

# PROGRAMAR

REVISTA PORTUGUESA DE PROGRAMAÇÃO | WWW.PORTUGAL-A-PROGRAMAR.PT | ISSN 1647-0710

EDIÇÃO #59 FEVEREIRO 2018

# BOZINTE

## A PROGRAMAR

**NEXUS** INTRODUÇÃO AO SONATYPE

**ADOBE PHONEGAP** APLICAÇÃO MOBILE

**SOAPUI** WEBSERVICES

**XAMARIN** FORMS / EFFECTS

**DEEP LEARNING** SET BY SET

**AND MORE!**

## ELECTRÓNICA

**WI-FI** ESP8266

## KERNEL PANIC

**SIMPLICIDADE** A ELEGÂNCIA

## REVIEWS

**BUSINESS INTELLIGENCE** **SQL SERVER**

**SEGURANÇA + SISTEMAS + REDES** **LINUX**

## SEGURANÇA

**BEFORE AND AFTER** **RGPD**

**IMPLEMENTAÇÃO SOFTETHER PROJECT** **VPN**

## NO CODE

**SISTEMAS ORIENTADOS**

**HANDS ON!** **M MAKER**

**CRM**

## EQUIPA PROGRAMAR

### Coordenador

António Pedro Cunha Santos

### Editor

António Pedro Cunha Santos

### Capa

Filipa Peres

### Redacção

André Melancia  
António Pedro Cunha Santos  
Augusto Manzano  
Bruno Horta  
Danilo Silva  
Jorge C. S. Cardoso  
Nuno Cancelo  
Nuno Silva  
Pedro Azevedo  
Pedro Tavares  
Ricardo Queirós  
Rita Peres  
Sara Silva  
Sérgio Ildelfonso  
Sérgio Saraiva

### Staff

António Pedro Cunha Santos  
Rita Peres

### Contacto

[revistaprogramar@portugal-a-programar.org](mailto:revistaprogramar@portugal-a-programar.org)

### Website

<http://www.revista-programar.info>

### ISSN

1 647-071 0

## O vale da sombra do ; “ponto-e-vírgula”

Existem diversas expressões para se referirem a “horas negras”, locais temíveis, etc... etc... etc... Mas entre todos os adeptos de tecnologia, o mais universal e conhecido, creio que seja o “vale da sombra do ponto-e-vírgula”, por onde, todos aqueles que escrevem código, acabam por passar, uma e outra vez, seja a “voar” no ansi-c, passando pelo java, o c++, o php e o c#, para não falar em todas as outras linguagens...

Verdade se diga que algures, em algum momento na nossa vida de tecnologia, passaremos por esse “vale profundo”, onde reina a escuridão da sombra do ponto-e-vírgula e dos erros de compilação, onde a linha apontada pelo compilador nada tem a ver com a linha onde verdadeiramente falta o dito cujo mal-afamado ponto-e-vírgula!

Mas... Nós que adoramos tecnologia, que escrevemos código, que temos orgulho no código que escrevemos... sabemos que se alguém consegue passar por esse vale e sair do outro lado, certamente será um de nós!

A verdade é que o “vale da sombra do ponto-e-vírgula”, existe para todas as áreas e todos os gostos, seja o do mal-afamado sinal de pontuação ( ; ), seja o do vale da sombra da “resistência” a que não nos juntamos, ou o do “crash inexplicado”, quanto estamos quase a terminar, mas nos esquecemos de gravar! A única certeza que se pode ter, neste “vale profundo e ensombreado”, é que sairemos do outro lado, seja acordados ou movidos a café, mas sempre de sorriso rasgado, prontos escrever código e ler mais uma edição! Por isso...

Até à próxima edição, boas leituras!

António Santos

*A revista PROGRAMAR é um projecto voluntário sem fins lucrativos. Todos os artigos são da responsabilidade dos autores, não podendo a revista ou a comunidade ser responsável por alguma imprecisão ou erro.*

*Para qualquer dúvida ou esclarecimento poderá sempre contactar-nos.*

## TEMA DE CAPA

- [6](#) Computação em Grelha, cross-platform usando BOINC - **António Santos**

## A PROGRAMAR

- [15](#) Introdução ao Sonatype Nexus - **Nuno Cancelo**
- [19](#) Adobe PhoneGap Build para construção de aplicações móveis híbridas - **Jorge C. S. Cardoso**
- [22](#) SOAPUI Uma ferramenta muito útil para quem desenvolve Webservices - **Sérgio Ildefonso**
- [22](#) Xamarin Forms – usando Xamarin Effects - **Sara Silva**
- [29](#) Deep Learning Passo a Passo - **Sérgio Saraiva**
- [33](#) Introdução ao SAAS - **Ricardo Queirós**
- [38](#) C# - Excel - **Rita Peres**

## ELECTRÓNICA

- [47](#) Como Configurar o Wi-Fi no ESP8266 - **Bruno Horta**

## COLUNAS

- [52](#) C# - Cifras e mais Cifras - **António Santos**
- [59](#) Kernel Panic - A elegância da simplicidade - **António Santos**
- [60](#) SQL Curtas #3: Mais um conjunto de dúvidas - **André Melancia**

## ANÁLISES

- [65](#) Review do livro: Business Intelligence no SQL Server - **Rita Peres**
- [67](#) Review do livro: Segurança Prática em Sistemas e Redes Com Linux - **António Santos**

## SEGURANÇA

- [69](#) Vulnerabilidades Web em 2017 - **Pedro Tavares**
- [73](#) RGPD – O Antes e o Depois - **Pedro Tavares**
- [75](#) SoftEther VPN Project, implementação em Windows, GNU/Linux, MacOS, Android, iOS - **António Santos**

## NO CODE

- [83](#) Testes em Sistemas Orientados a objectos: modelo tradicional vs modelo OO - **Augusto Manzano**
- [85](#) Microsoft 365: Ferramentas de Deploy, Segurança e Compliance - **Nuno Silva**
- [89](#) Movimento Maker – Hands ON - **Rita Peres**
- [90](#) CRM – Plataforma ou Customizado - **Pedro Azevedo**
- [92](#) SQL vs NoSQL - **Rita Peres**
- [94](#) Entrevista a: Bruno Horta
- [96](#) Python: História e Ascendência - **Danilo Morais da Silva**

## EVENTOS

**27 Fev.** - *Hardware City Session#9 | Product Development - Design Edition* [Centro Cultural e de Congressos de Aveiro]

**28 Fev.** - *Tertulia - (Sem) Preconceito do Empreendedorismo* [Escola Superior de Tecnologia e Gestão - IPVC]

Para mais informações/eventos: [http://bit.ly/PAP\\_Eventos](http://bit.ly/PAP_Eventos). Divulga os teus eventos para o email [eventos@portugal-a-programar.pt](mailto:eventos@portugal-a-programar.pt)



## VianaTech Meetup 2018

Aconteceu no passado dia 15 de Fevereiro, o primeiro VianaTech Meetup de 2018. Teve lugar na Aceleradora Tecnológica de Viana do Castelo, tendo como anfitriã a Sparkle IT e coordenado por David Cunha e Ricardo Parente.

Subordinado ao tema “Insights for the future of the Internet”, contou com a participação dos oradores Wayne Andersen, profissional aposentado das TI’s, e Nelson Dias, CEO de uma empresa do norte de Portugal de nome Digital Species.

A evolução da tecnologia e a que passo nos adaptamos, as perspectivas para o futuro e o que será a tecnologia com as novas gerações, foram temas debatidos e sobre os quais foram partilhadas opiniões e ideias.

É de salutar as iniciativas que tecnológicas que ocorrem por todo este “Portugal profundo” e que acabam tornando cada lugar, um pouco “tecnologicamente mais sexy”!



<https://picuscreative.com/>

<http://www.sparkleit.pt/>

<https://www.meetup.com/VianaTechMeetups/>

## Investigadores da UC criam algoritmo de Inteligência Artificial inovador

Quatro investigadores do Centro de Informática e Sistemas da Universidade de Coimbra (CISUC) desenvolveram uma abordagem vanguardista para automatizar processos de aprendizagem no campo da visão computacional.

Trata-se de um **novo algoritmo de inteligência artificial para a evolução das denominadas redes neuronais** (que

imitam o comportamento do cérebro) **profundas**. O DENSER, acrónimo de *Deep Evolutionary Network Structured Representation* (<https://cdy.dei.uc.pt/denser/>), pode muito bem vir a revolucionar a forma de responder ao problema de classificação do conteúdo de imagens.

Na área da Inteligência Artificial e *Machine Learning* (aprendizagem de máquina), a classificação de imagens é altamente complexa e enfrenta grandes desafios. O que é compreensível, considerando que, nos humanos, um terço do cérebro é dedicado ao processamento visual, envolvendo centenas de milhões de neurónios.

Desenvolvido no âmbito de um projeto de investigação financiado pela Fundação para a Ciência e Tecnologia (FCT), o DENSER tem vindo a despertar o interesse da comunidade científica e será apresentado na próxima EvoStar, a mais importante conferência europeia na área da computação evolucionária (computação de inspiração biológica para resolver problemas de elevada complexidade). Destaca-se das abordagens convencionais ao não exigir intervenção humana, não usar conhecimento prévio sobre o domínio e por ser uma solução de baixo custo.

O sistema que mais se aproxima da solução arquitetada pelos investigadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra é o do projeto “Google Brain”, da gigante Google, que obtém resultados marginalmente inferiores. Conforme sublinham os outros dois investigadores da equipa, Filipe Assunção e Nuno Lourenço, **«eles obtêm um resultado competitivo com o nosso no CIFAR 10, mas não reportam os resultados no CIFAR 100, que é um problema mais difícil. Para além disso, usam algum conhecimento sobre o problema, o que os ajuda a alcançar bons resultados.»**

A outra vantagem é o baixo custo do DENSER. Enquanto os investigadores do projeto “Google Brain” **«usam 800 GPUs (placas gráficas) topo de gama, nós usamos 4 GPUs das mais acessíveis, que são usadas, por exemplo, nos videojogos. Para se ter uma ideia, as 800 GPUs da Google custam 1,3 milhões de euros e as nossas apenas 2.500 euros»**, destacam.

Os resultados do projeto, que podem vir a ser aplicados em vários domínios do conhecimento, decorrem da experiência de mais de duas décadas do CISUC nas áreas da inteligência artificial, *machine learning* e computação evolucionária, e de um esforço e investimento mais recente no domínio da *evolutionary machine learning*.



# TEMA DE CAPA

Computação em Grelha, cross-platform usando BOINC

## Computação em Grelha, cross-platform usando BOINC

Ao longo de diversas edições fui escrevendo sobre clusters recorrendo ao SBC (single board computer) Raspberry Pi. Tendo em conta que cada vez existem mais dispositivos inteligentes (smart devices), dei por mim a pensar no eventual uso de tais dispositivos, que passam uma parte substancial do seu tempo de “vida útil” em “idle” (sem utilização, mas ainda assim ligados), para tarefas de computação. Algo parecido com o Seti@Home, de há uns anos atrás, quando eu era bem mais novo e a internet por cá, ainda chegava por “dial-up”.

É resumidamente isso que vos apresento neste artigo! Uma plataforma simples de instalar e utilizar, dedicada à computação distribuída, utilizando diversas plataformas e diversos dispositivos, aproveitando o tempo destes equipamentos, quando estão por exemplo parados a carregar baterias ou ligados à corrente mas inactivos.

### Computação em Grelha (Grid Computing)

Computação Distribuída é um modelo no qual componentes de um sistema de software são partilhados por múltiplos computadores de forma a melhorar a eficiência e performance. De acordo com a definição mais refinada do conceito, computação distribuída é limitada a programas com componentes partilhados por diversos computadores, numa área geográfica limitada. No entanto, definições mais alargadas, incluem também partilha de tarefas, partilha de componentes, etc... sendo que a definição mais abrangente, define computação distribuída como a computação partilhada por diversos sistemas, podendo estes encontrarem-se em diversas localizações geográficas.

Numa das primeiras definições destas tecnologias, datada de 1998 Carl Kesselman, e Ian Foster, escreveram no livro “*The Grid: Blueprint for a new computing infrastructure*”, “Uma grelha computacional, é uma combinação de hardware e software, que disponibiliza de forma confiável, consistente, penetrante e de baixo custo, acesso a recursos computacionais de grande capacidade”. Esta definição foi alvo de alterações posteriores, mas na sua génese mantém-se a partilha de recursos computacionais de forma dinâmica.

Na realidade empresarial e em grande parte na realidade das aplicações baseadas em web, computação distribuída, muitas vezes significa distribuição de etapas de um processo de negócio, através dos locais onde cada etapa é executada mais eficientemente. Por exemplo, no modelo de três camadas, significa que o processamento do interface de utilizador é feito no dispositivo do utilizador, o processamento de negócio é feito num sistema remoto, e o armazenamento de dados, num outro sistema, onde se encontra o sistema gestor de base de dados (SGBD), que normalmente fornece armazenamento centralizado de dados a diversos processos de negócio. De forma geral, este tipo de computação recorre

ao modelo de comunicações cliente-servidor.

A Computação em Grelha (Grid Computing), é mais específica e refere-se a um modelo de computação envolvendo arquitectura distribuída, de diversos dispositivos de processamento, interligados, com o objectivo de resolver um problema complexo. Uma característica que distingue a computação Grid, dos clusters de computação, reside no facto de cada nó que compõe o sistema, executa uma tarefa ou aplicação diferente, sem necessidade de comunicação em tempo real com os restantes nós. Além disto uma determinada Grid (grelha) de dispositivos, tende a ser composta por vários dispositivos, heterogéneos e dispersos na sua localização.

Assim uma grid, muitas vezes é composta por diferentes tipos de equipamentos, capazes de realizar processamento de dados, e de diferentes arquitecturas, podendo correr diferentes sistemas operativos, de forma mais ou menos eficaz.

Uma vez colocada a funcionar, servidores, estações de trabalho, computadores pessoais, ou outros dispositivos, executam tarefas independentes, estando interligados de forma flexível, permitindo que a ligação seja interrompida e retomada, bem como utilizadas ligações de muito baixa velocidade. De igual forma, os “nós” participantes de uma grelha, podem definir quando os seus recursos podem ser utilizados e que quantidade de recursos de processamento seus podem ser utilizados, incluindo atribuir apenas alguma parte da sua capacidade de multitarefa, enquanto o utilizador se encontra a utilizar o dispositivo. O mais conhecido exemplo disto é o projecto SETI@Home, que recorre a esta tecnologia para analisar dados de radio-telescópios em busca de inteligência extraterrestre.

De forma resumida, e em forma de consolidação textual, existem três premissas que definem a computação em grelha. São elas as seguintes:

- Coordenar e integrar, recursos e utilizadores que não estão sujeitos a um controlo central (a grelha integra e coordena recursos de utilizadores que vivem dentro de diferentes limites, diferentes geografias, diferentes regimes, etc...)
- Utilizar protocolos de propósito geral, bem como interfaces padrão e abertos (a grelha é construída, com base em protocolos e interfaces multi-utilização, que endereçam questões fundamentais, como autenticação, autorização, descoberta de recursos, acesso a recursos, etc... É de relevante importância que se tratem de protocolos e interfaces abertos, caso contrário tornam-se específicos de um dado sistema).
- O objectivo é disponibilizar qualidade de serviço, não-trivial (a grelha permite que os recursos que a constituem, sejam usados de forma coordenada para dispo-

# TEMA DA CAPA

## COMPUTAÇÃO EM GRELHA, CROSS-PLATFORM USANDO BOINC

nibilizar diversos tipos de serviços, como por exemplo, tempo de resposta, disponibilidade, co-alocação de recursos, de forma a que a utilidade do conjunto de recursos seja maior que a soma das partes.

Apesar de não ser tão conhecido, um dos primeiros usos deste tipo de arquitectura, foi feito pelo distributed.net, com o objectivo, de quebrar códigos criptográficos. Este projecto remonta a 1997, e encontra-se acessível no endereço <http://www.distributed.net>. Por caricato que o leitor possa achar, disponibilizam software cliente até para sistemas operativos tão antigos quanto Acron RISC OS, AmigaOS, MS-DOS e Windows 3.x 16bit!

### Instalação do BOINC Server

Feita a apresentação teórica do tema, passemos ao lado mais prático! A instalação do BOINC Server, no Raspberry Pi, ou em qualquer outro equipamento que corra GNU/Linux, no caso concreto, a distribuição Raspbian. Esta instalação está dividida em quatro etapas, pelo que iremos executar as mesmas uma a uma!

### ETAPA 1 – Instalação do Sistema Operativo Raspbian

Assume-se para o efeito deste artigo que o utilizador já instalou Raspbian no passado, estando familiarizado com o processo de instalação do mesmo (Caso o caro leitor não esteja familiarizado com este ponto, aconselho a leitura da edição número 57, onde pode encontrar a explicação sobre como o fazer. Caso deseje um artigo mais detalhado sobre a instalação, aconselho a leitura da edição número 48)

### ETAPA 2 - Instalação do Boinc Server e inicialização do seu website

A primeira tarefa a realizar, será criar um utilizador, e grupo, para administrador do Boinc e criar a respectiva pasta para o website.

```
$ sudo adduser --system --group boincadm --no-  
create-home  
$ sudo usermod -aG boincadm www-data
```

Criado o utilizador, adiciona-se o repositório onde se encontram os packages, à lista de fontes existentes, com o seguinte comando:

```
deb http://ftp.debian.org/  
debian experimental main
```

De seguida actualiza-se a lista de fontes e procede-se com a instalação!

```
sudo apt-get update  
sudo apt-get install -t experimental boinc-server  
-maker
```

*Convém ressaltar que os passos seguintes, terão de ser repetidos para cada projecto Boinc, bem como em cada actualização ou nova instalação.*

Começamos por inicializar a configuração com os seguintes comandos:

```
echo "boincserverconfig=~/.boinc_test.conf" >  
~/.boinc_test.conf  
chmod 600 ~/.boinc_test.conf  
. ~/.boinc_test.conf
```

E sempre que a Shell seja executada, as seguintes definições devem ser aplicadas. O mais simples nestes casos, seria criar um utilizador dedicado ao projecto, e as configurações a serem executadas automaticamente após o arranque da Shell, como no exemplo seguinte:

```
cat >> ~/.bashrc << EOBASHRC  
if [ -r "$boincserverconfig" ]; then  
  . $boincserverconfig  
fi  
EOBASHRC
```

Feita esta etapa, avançamos para a configuração do MySQL com os seguintes comandos:

```
cat << EODBCONFIG >> $boincserverconfig  
# nome da BD do MySQL  
dbname=boinctest  
# Nome do utilizador do MySQL com permissões para a  
base de dados  
dbuser=boincadm  
# Password com acesso de escrita, para a base de  
dados do projecto  
dbpasswd=MYSQLPASSWORDFORBOINCUSER  
EODBCONFIG
```

Pelas obvias razões a password deve ser alterada.

Feito isto procedemos às configurações recorrendo ao seguinte script:

```
# read config if available  
[ -r "$boincserverconfig" ] && . $boincserverconfig  
if [ -z "$dbuser" ]; then echo "Variable 'dbuser'  
not set";  
elif ! echo "DROP USER '$dbuser'@'localhost'" |  
mysql -u root -p; then  
  echo "If the removal of the previous user fails  
  because the user is not existing, then this does  
  not matter. Other errors please investigate."  
fi
```

Com esta configuração feita, passamos para a configuração seguinte:

```
if [ -z "$dbname" ]; then echo "Variable 'dbname'  
not set";  
elif [ -z "$dbpasswd" ]; then echo "Variable  
'dbpasswd' not set";  
else  
  # piping commands to mysql shell  
  cat <<EOMYSQL | mysql -u root -p;  
  DROP DATABASE IF EXISTS $dbname;  
  CREATE USER '$dbuser'@'localhost' IDENTIFIED BY  
'$dbpasswd';
```

# TEMA DA CAPA

## COMPUTAÇÃO EM GRELHA, CROSS-PLATFORM USANDO BOINC

```
GRANT ALL PRIVILEGES ON $dbname.* TO
'$dbuser'@'localhost';
EOMYSQL
fi
```

Em ambos os casos será pedida a password de root do MySQL, que deverá ser diferente da password de root do sistema operativo.

Terminadas as configurações, criamos o projecto boinc, recorrendo ao seguinte script:

```
cat << EOCONF >> $boincserverconfig
# Endereco ip ou domínio do servidor
hosturl=http://mouseBoinc.no-ip.info/
# nome do directório onde o projecto estara arma-
zenado
projectname=boincPi
# nome do projecto "para humanos"
projectnicename="Boinc@pi"
# localização para armazenar os ficheiros do pro-
jecto
installroot=/var/lib/boinc-server-test
EOCONF
```

Neste próximo passo, iremos editar o ficheiro boinc\_pi.conf que se encontra em ~/.boinc\_pi.conf, e ajustar correctamente as configurações:

```
if [ ! -r "$boincserverconfig" ]; then
    echo "Configuration file '$boincserverconfig'
        not existing."
else
    . $boincserverconfig
    echo -n "Basic configuration test: "
    if [ -z "$installroot" -o -z "$hosturl" -o -z
        "$dbname" -o -z "$dbpasswd" \
        -o -z "$projectnicename" -o -z
        "$projectname" ]; then
        echo "Missing configuration parameter."
    else
        echo "[ok]"
    fi
fi
```

E uma vez que iremos utilizar o server package, podemos prosseguir com a instalação usando o seguinte comando:

```
sudo PYTHONPATH=$PYTHONPATH:/usr/share/python-
support/boinc-server/Boinc/ \
/usr/share/boinc-server/tools/make_project \
--url_base "$hosturl" \
--db_name "$dbname" \
--db_user "$dbuser" \
--db_passwd "$dbpasswd" \
--drop_db_first \
--project_root "$installroot"/"$projectname" \
--srcdir /usr/share/boinc-server-maker/ \
"$projectname" "$projectnicename"
```

O output deste ultimo comando, deverá ser semelhante ao seguinte:

```
Creating project 'Boinc@pi':
PROJECT_ROOT = /var/tmp/boinc/boincpi/
URL_BASE = http://mouseBoinc.no-ip.info/
HTML_USER_URL = http://mouseBoinc.no-ip.info/
                    boincpi/
HTML_OPS_URL = http://mouseBoinc.no-ip.info/
```

```
                    boincpi_ops/
CGI_URL = http://mouseBoinc.no-ip.info/
                    boincpi_cgi/
KEY_DIR = /var/tmp/boinc/boincpi/keys/
DB_NAME = boincpi
DB_HOST =
Delete /var/tmp/boinc/boincpi/? [y/N] y
Deleting /var/tmp/boinc/boincpi/

Continue? [Y/n]
Setting up server: creating directories

Keys don't exist in /var/tmp/boinc/boincpi/keys/;
generate them? [Y/n]
Setting up server files: generating keys
Setting up server files: copying files
Setting up database
Setting up server files: writing config files
Setting up server files: linking cgi programs
update_translations finished
Done installing default daemons.

Done creating project. Please view
/var/tmp/boinc/boinctest/boincpi.readme
for important additional instructions.
```

Feita a instalação do projecto Boinc, temos de fazer alguns ajustes de permissões! Começamos por verificar se as variáveis de configuração estão correctamente definidas:

```
[ -r $boincserverconfig ] && . $boincserverconfig
```

E depois executamos o seguinte script para alterar as permissões aos ficheiros e directórios:

```
if [ -z "$installroot" -o -z "$projectname" ];
then
    echo "Not all variables are set for the
        configuration"
    echo "Error, do not continue."
elif [ ! -d "$installroot"/"$projectname" ]; then
    echo "The directory '$install-
root/'$projectname' is not existing"
    echo "Error, do not continue."
else
    cd "$installroot"/"$projectname"
    sudo chown boincadm:boincadm -R .
    sudo chmod g+w -R .
    sudo chmod 02770 -R upload html/cache html/inc
        html/languages html/languages/compiled html/
        user_profile
    hostname=`hostname`
    if [ -z "$hostname" ]; then
        echo "Please specify the hostname"
    else
        sudo chgrp -R www-data log_"$hostname" upload
    fi
fi
```

Se tudo correu como previsto, foi pedida a password para sudo! Caso contrário nada foi pedido e será melhor ver o que ocorreu de errado e repetir os passos anteriores antes de prosseguir o processo.

Dependendo da actualização que se tem, poderá ser necessário executar mais um script, pelo que pode sempre ser executado o script seguinte:

```
if [ -d html/inc -a -d cgi-bin ]; then
    echo -n "html/inc: "; sudo chmod o+x html/inc
    && sudo chmod -R o+r html/inc && echo "[ok]" ||
    echo "[failed]"
```



# TEMA DA CAPA

## COMPUTAÇÃO EM GRELHA, CROSS-PLATFORM USANDO BOINC

```
echo -n "html/languages: "; sudo chmod o+x html/
languages/ html/languages/compiled && echo "[ok]"
|| echo "[failed]"
else
echo "You are not in your project directory"
fi
```

E é chegado o momento de iniciar o Boinc e os seus deamons (programas em segundo-plano)

```
$/bin/start
```

Caso algo tenha corrido mesmo mal e apareça uma mensagem parecida com: "No daemons for this host found - check host name in config.xml", será melhor executar o seguinte, para se verificar o que possa ter acontecido:

```
$/bin/status
```

Partindo do princípio que tudo correu como era suposto, chegamos ao momento de adicionar um cronjob ao crontab para que o Boinc seja iniciado automaticamente:

```
$/ sudo EDITOR=nano crontab -e
```

Colocamos a seguinte linha no crontab:

```
*/5 * * * * /var/tmp/boinc-server-test/boinctest/
bin/start --cron
```

e uma vez gravado, podemos verificar se tudo está com suporto:

```
$/ sudo crontab -l
```

Neste momento ainda nos falta definir a password para a página de administração do Boinc!

```
sudo htpasswd -c html/ops/.htpasswd USERNAME
```

E configurar o servidor apache, para chamar o BOINC Server:

```
sudo cp ${projectname}.httpd.conf /etc/apache2/
sites-available/ \
&& sudo a2ensite ${projectname}.httpd.conf \
&& sudo /etc/init.d/apache2 reload
```

Isto já vai longo, pensará o caro leitor, mas ainda falta alguma coisa! Ora vejamos, até aqui fizemos as configurações necessárias ao Boinc e aos seus deamons... Agora começam as configurações da administração do Boinc via interface web!

Primeiramente executamos o seguinte:

```
sudo a2enmod cgi
sudo a2enmod php5
sudo /etc/init.d/apache2 restart
```

E finalmente podemos colocar o nome do nosso projecto no site!

```
sudo nano html/project/project.inc
```

Com tudo isto a funcionar estamos prontos para iniciar a próxima fase! Adicionar uma aplicação ao Boinc Server!

Antes de o fazer, poderíamos ainda ter adicionado a funcionalidade de fórum, mas como se trata de utilizar o BOINC em casa, não executaremos essa parte (recomendo ao leitor a leitura da documentação oficial do projecto, caso o queira fazer).

### ETAPA 3 - Colocação de uma aplicação no BOINC

Existem duas formas de colocar aplicações no BOINC. Uma delas é recorrendo ao XADD, fornecendo os binários compilados para cada plataforma suportada. Outra é utilizando os binários disponibilizados pelo Debian, uma vez que o Debian disponibiliza binários para diversas plataformas.

Em ambos os casos a app utilizada, será a app exemplo do BOINC. No caso a app converte todos os caracteres dentro de um ficheiro, para letra maiúscula.

Assim sendo começados por utilizar apenas binários compilados para uma só plataforma, no caso ARM (a arquitectura do Raspberry Pi). Primeiro descarregamos o binário para a plataforma que estamos a usar:

```
$/ sudo apt-get install boinc-app-examples
```

Uma vez descarregado o binário, criamos a directoria do projecto e adicionamos a aplicação à configuração do projecto:

```
[ -z "$installroot" -o -z "$projectname" ] && .
~/boinc_test.conf
if test -z "$installroot" -o -z "$projectname";
then
echo "E: ~/boinc_test.conf does not have all
parameters."
else
appdir="$installroot"/"$projectname"/apps/
upper_case
echo "I: The application directory is '$
$appdir'."
sudo mkdir -p "$appdir"
fi
```

E copiamos e renomeamos o ficheiro anteriormente descarregado, para a directoria que criamos, renomeando-o de forma a ser fácil distinguir que se trata do binário para a arquitectura que estamos a utilizar:

```
appver=7.042 # colocar a versão correcta "."
boincplat=$(arch) arm-unknown-linux-gnueabihf #
ajustar para a arquitectura correcta arm-unknown-
linux-gnueabihf, lista de arquitecturas
em: http://boinc.berkeley.edu/trac/wiki/
BoincPlatforms
binary=$(dpkg -L boinc-app-examples | grep
upper_case)
echo "I: Application version: $appver"
echo "I: BOINC platform: $boincplat"
echo "I: binary at: $binary"
if test -z "$appdir" -o -z "$appver" -o -z
"$boincplat" -o -z "$binary"; then
echo "E: Lost one of the variables appdir,
appver, binary or boincplat - please check and try
again"
else
completepath=$appdir/$appver/$boincplat && \
```

# TEMA DA CAPA

## COMPUTAÇÃO EM GRELHA, CROSS-PLATFORM USANDO BOINC

```
sudo mkdir -p $completepath && \  
sudo cp $binary $completepath/  
upper_case_${appver}_${boincplat} && \  
echo "[ok]"  
fi
```

Feito isto, informamos a base de dados local, de quais os binários disponíveis! Para tal, criamos o ficheiro chamado project.xml, onde colocamos todas as informações sobre o projecto, por exemplo plataformas e ferramentas suportadas, etc... Por uma questão de comodidade, e uma vez que se trata de um exemplo, poderemos usar o seguinte script bash, para criar o ficheiro por nós:

```
[ -z "$installroot" -o -z "$projectname" ] || .  
~/.boinc_test.conf  
if [ -z "$installroot" -o -z "$projectname" ];  
then  
    echo 'Variables $installroot (' $installroot ')  
        and $projectname (' $projectname ') are  
        both required.'  
elif [ -d "$installroot/$projectname" ]; then  
    if [ -f "$installroot/$projectname/  
        project.xml" ]; then  
        sudo gzip -9 --suffix ".$(date +"%Y%m%dT%H%M")  
        .gz" "$installroot/$projectname/project.xml"  
    fi  
(cat << EOPROJECTXML  
<boinc>  
  <app>  
    <name>upper_case</name>  
    <user_friendly_name>upperCASE  
    </user_friendly_name>  
  </app>  
  <platform>  
    <name> arm-unknown-linux-gnueabihf</name>  
    <user_friendly_name>Linux/Arm  
    </user_friendly_name>  
  </platform>  
  <platform>  
    <name> i686-pc-linux-gnu</name>  
    <user_friendly_name>Linux/i686-pc  
    </user_friendly_name>  
  </platform>  
</boinc>  
EOPROJECTXML  
) | sudo tee "$installroot"/"$projectname"/  
        project.xml  
fi
```

O project.xml, irá alimentar o xadd, que por sua vez vai colocar a informação na base de dados do projecto, no MySQL. Estas informações, indicam que as aplicações, para as arquiteturas que estão disponibilizadas, devem estar preparadas para ser encontradas, eventualmente. No caso vamos apenas suportar ARM e x86 32bits.

E por fim usamos o XADD, para adicionar o projecto:

```
cd "$installroot"/"$projectname"  
sudo bin/xadd
```

E o output deverá ser semelhante ao seguinte:

```
$ sudo bin/xadd  
Processing <App#None upper_case> ...  
Committed <App#1 upper_case> ; values:  
{'_dirty': False,  
'_lazy_lookups': {},  
'beta': 0,  
'create_time': 1357056725L,
```

```
'deprecated': 0,  
'homogeneous_app_version': 0,  
'homogeneous_redundancy': 0,  
'host_scale_check': 0,  
'id': 1L,  
'locality_scheduling': 0L,  
'min_avg_pfc': 1.0,  
'min_version': 0L,  
'name': 'upper_case',  
'non_cpu_intensive': 0,  
'target_nresults': 0,  
'user_friendly_name': 'upperCASE',  
'weight': 1.0}  
Processing <Platform#None i686-pc-linux-gnu> ...  
Committed <Platform#1 i686-pc-linux-gnu> ;  
        values:  
{'_dirty': False,  
'_lazy_lookups': {},  
'create_time': 1357056725L,  
'deprecated': 0,  
'id': 1L,  
'name': 'i686-pc-linux-gnu',  
'user_friendly_name': 'Linux/x86 32bit'}  
Processing <arm-unknown-linux-gnueabihf> ...  
Committed <arm-unknown-linux-gnueabihf> ;  
        values:  
{'_dirty': False,  
'_lazy_lookups': {},  
'create_time': 1357056725L,  
'deprecated': 0,  
'id': 2L,  
'name': 'x86_64-pc-linux-gnu',  
'user_friendly_name': 'Linux/amd64 64bit'}
```

Feita esta tarefa, assinamos o binário da aplicação, por razões de segurança. Assim as máquinas que vão correr a aplicação têm alguma "confiança" de que não estão a correr um binário malicioso. Para assinar executamos o código seguinte:

```
privateKeyfile="./keys/code_sign_private"  
if [ -z "$appver" -o -z "$boincplat" ]; then  
    echo "Please set appver and boincplat  
        variables from above."  
elif [ ! -r "$privateKeyfile" ]; then  
    echo 'Have your private key ready as created  
        during setup, expected at $privateKeyfile .'  
else  
    sudo ./bin/sign_executable apps/upper_case/  
$appver/$boincplat/upper_case_${appver}  
_${boincplat} "$privateKeyfile" | sudo tee apps/  
        upper_case/$appver/$boincplat/  
        upper_case_${appver}_${boincplat}.sig  
fi
```

Dada a vastidão de plataformas suportadas pelo Boinc, pode por vezes ser complicado distinguir que binários são de que plataforma. Assim sendo, utilizamos o ficheiro version.xml, para facilitar o processo! Para o criar executamos o seguinte código:

```
if [ -z "$appver" -o -z "$boincplat" ]; then  
else  
cat << EOVERSIONXML | sudo tee apps/upper_case/  
        $appver/$boincplat/version.xml  
cat version.xml  
<version>  
  <file>  
    <physical_name>upper_case_7.042_x86_64-pc-  
        linux-gnu</physical_name>  
    <main_program/>  
  </file>
```

# TEMA DA CAPA

## COMPUTAÇÃO EM GRELHA, CROSS-PLATFORM USANDO BOINC

```
</version>
EOVERSIONXML
fi
```

E actualizamos a base de dados do boinc:

```
sudo ./bin/update_versions
```

Apenas para confirmar que tudo ficou como esperado, podemos fazer uma query à base de dados e confirmar o resultado dessa query!

```
$ echo "select * from app" | mysql -u boincadm
-p$pw $dbprojectname
```

Partindo do princípio que tudo correu como esperado, podemos então seguir para a etapa final! A criação de uma work unit!

### ETAPA 4 - Criar uma unidade de trabalho (work unit)

Uma unidade de trabalho (work unit), é uma porção dos dados do projecto que devem ser analisados com uma aplicação específica. No entanto as aplicações e os dados podem ser partilhados entre unidades de trabalho (work units). Para que a criação da unidade de trabalho, possa ser feita, necessitamos das seguintes informações:

- Ficheiros de entrada
- Template da unidade de trabalho (Work Unit Template)
- Template de resultados (Result Template)

Os ficheiros de entrada (input files), são aqueles em que a unidade de trabalho deve trabalhar. Tipicamente, variam de unidade de trabalho para unidade de trabalho. Por exemplo, o intervalo de números primos que uma unidade de trabalho deve procurar. Apesar disto, os ficheiros de entrada (input files), podem ser partilhados entre work units.

Neste caso concreto, como estamos a usar desde o início a aplicação exemplo, cujo objectivo é converter todos os caracteres de um ficheiro em caracteres de letra maiúscula, iremos criar um ficheiro de entrada de testes. Basicamente um ficheiro, com apenas uma linha, do tipo string, com todas as letras minúsculas, e que se encontrará na directoria \$installroot/\$projectname/download/input\_file. Este ficheiro será todo convertido em letra maiúscula.

Agora vamos criar o template! Para tal executamos o seguinte:

```
~/.boinc_test.conf
if test -n "$installroot" -a -n "$projectname" &&
cd $installroot/$projectname; then
sudo strace -o download/input_file -- strace -h
> /dev/null && echo "[ok]" || echo "[failed]"
else
echo "E: please check parameters in
~/.boinc_test.conf"
fi
```

Agora copiamos o ficheiro de template que acabamos de criar!

```
sudo cp /usr/share/doc/boinc-server-maker/
examples/upper_case_{result,wu} $installroot/
$projectname/templates/
```

O ficheiro é bastante simples, um apenas tem duas linhas e o outro tem algumas linhas mais, mas também bastante simples. Podemos ver ambos os ficheiros com os seguintes comandos:

```
$ cat /var/tmp/boinc-server-test/boinctest/
templates/upper_case_result
```

Que deve produzir o seguinte resultado:

```
<file_info>
<name><OUTFILE_0/></name>
<generated_locally/>
<upload_when_present/>
<max_nbytes>10000</max_nbytes>
<url><UPLOAD_URL/></url>
</file_info>
<result>
<file_ref>
<file_name><OUTFILE_0/></file_name>
<open_name>out</open_name>
</file_ref>
</result>
```

```
$ cat /var/tmp/boinc-server-test/boinctest/
templates/upper_case_wu
```

Que deve produzir o seguinte resultado:

```
<file_info>
<number>0</number>
</file_info>
<workunit>
<file_ref>
<file_number>0</file_number>
<open_name>in</open_name>
</file_ref>
<min_quorum>1</min_quorum>
<target_nresults>1</target_nresults>
</workunit>
```

Posto isto, chegou o momento de preencher a base de dados! Neste passo, vamos apresentar ao sistema uma única unidade de trabalho. O input permanece como descrito no input\_file, os templates para os resultados são especificados individualmente a cada unidade de trabalho e o nome da aplicação é lido da linha de comandos. Assim, podemos ter aplicações diferentes, a trabalhar sobre a mesma unidade de trabalho, de forma mais simples! Para realizar esta tarefa executamos o código seguinte:

```
installroot" -a -n "$projectname" ] && \
cd $installroot/$projectname && \
sudo ./bin/create_work -appname upper_case
-wu_name test -wu_template templates/
upper_case_wu -result_template templates/
upper_case_result input_file || echo "E:
something failed"
```

# TEMA DA CAPA

## COMPUTAÇÃO EM GRELHA, CROSS-PLATFORM USANDO BOINC

Carregamos múltiplas unidades de trabalho. Para tal temos primeiro de criar os ficheiros de input, e para facilitar usamos um loop, porque caso não o fizessemos teríamos de carregar uma a uma, manualmente.

```
for i in $(seq 1 3); do
    echo ./bin/create_work -appname upper_case
                          -wu_name test$i input_${i}_file
done
```

Depois podemos criar as tarefas, usando os comandos seguintes:

```
./bin/create_work -appname upper_case -wu_name
                  test1 input_1_file
./bin/create_work -appname upper_case -wu_name
                  test2 input_2_file
./bin/create_work -appname upper_case -wu_name
                  test3 input_3_file
```

Neste momento o directório de download terá um directório criado especialmente para esta unidade de trabalho (workunit) como se pode ver executando o seguinte comando:

```
ls -ltr $installroot/$projectname/download | tail
-1
```

Por fim, vamos ver como utilizar output (saída) para múltiplos ficheiros. Os ficheiros de input e output (entrada e saída), são especificados na unidade de trabalho (workunit) e no template de resultados, respectivamente. Para tal o par de tags <file\_info>, <file\_ref>, descrevem cada par de ficheiros. Por exemplo, se precisarmos de criar uma workunit test que recebe in1 e in2 como ficheiros de entrada o template seria algo parecido com o seguinte:

```
<file_info>
  <number>0</number>
</file_info>
<file_info>
  <number>1</number>
</file_info>

<workunit>
  <file_ref>
    <file_number>0</file_number>
    <open_name>in1</open_name>
  </file_ref>
  <file_ref>
    <file_number>1</file_number>
    <open_name>in2</open_name>
  </file_ref>
</workunit>
```

E o template de resultados, seria algo parecido com o seguinte:

```
<file_info>
  <name>
    <OUTFILE_0/>
  </name>
  <generated_locally/>
  <upload_when_present/>
  <max_nbytes>10000</max_nbytes>
  <url>
    <UPLOAD_URL/>
  </url>
</file_info>
<file_info>
  <name>
    <OUTFILE_1/>
  </name>
```

```
<generated_locally/>
<upload_when_present/>
<max_nbytes>10000</max_nbytes>
<url>
  <UPLOAD_URL/>
</url>
</file_info>
<result>
  <file_ref>
    <file_name>
      <OUTFILE_0/>
    </file_name>
    <open_name>out1</open_name>
  </file_ref>
  <file_ref>
    <file_name>
      <OUTFILE_1/>
    </file_name>
    <open_name>out2</open_name>
  </file_ref>
</result>
```

Neste momento temos as unidades de trabalho prontas, podemos então passar para a gestão de resultados!

### Gestão de Resultados

Esta é a última etapa do artigo! Até aqui, já instalamos o BOINC Server, configuramos a interface de administração, criamos uma tarefa, preparamos a sua execução, agora falta-nos receber e utilizar os resultados enviados pelos BOINC Clients!

Cada app cliente, para cada plataforma é diferente e tem as suas especificidades. Neste ponto, apenas para exemplo, vamos assumir que está a ser utilizada a aplicação Boinc-client, para Linux.

Uma vez criado o utilizador na plataforma Boinc, o seu registo deve aparecer na base de dados. Para confirmar que tal ocorreu como era suposto, podemos executar a seguinte query no MySQL:

```
select
id,create_time,email_addr,name,authenticator,
country,postal_code,url,donated from user;
```

De igual modo, podemos ver a primeira tarefa executada pela app cliente, com a seguinte query:

```
Select
id,create_time,workunitid,outcome,hostid,userid,
name,cpu_time, app_version_num, appid from
result;
```

Por fim podemos verificar o estado de execução de uma unidade de trabalho work unit, recorrendo ao seguinte comando:

```
select id,create_time,appid,name,need_validate
from workunit;
```

Quando um resultado aparece com o valor 1 no campo "need\_validate", quer dizer que o resultado ainda não é "confiável" (trusted), e que ainda requer validação!

A validação é um processo, que seria algo extenso



# TEMA DA CAPA

## COMPUTAÇÃO EM GRELHA, CROSS-PLATFORM USANDO BOINC

de explicar em detalhe e como este exemplo apenas se propõe a ser um projecto simples de exemplo a ser disponibilizado na internet, vamos apenas focar-nos no essencial.

Contudo e apenas para que se perceba, a validação é de forma sumária, a comparação de resultados, da execução da mesma tarefa, por dois nós diferentes, sendo que o resultado deve ser igual. Dito isto, neste caso não será feita a validação! Certamente isto não seria aceitável em termos científicos, não obstante, como se trata de um exemplo, apenas foram tidas algumas preocupações de segurança, nomeadamente as seguintes:

- Um falso-negativo, não será considerado “dramático”, porque existe redundância na análise
- Apenas serão utilizados dispositivos do leitor

O `make_project` inicial apenas foram colocados dois validadores para este projecto. Eles podem ser encontrados em `bin/sample_trivial_validator` e `bin/sample_bitwise_validator`. De igual forma, o `make_project` gera um ficheiro de configuração para as validações com o nome `config.xml` que pode ser encontrado em `$installroot/$fileprojectname/config.xml` e deve ser semelhante ao seguinte:

```
<?xml version="1.0" ?>
<boinc>
  <config>

  </config>
  <tasks>

  </tasks>
  <daemons>
    <daemon>
      <cmd>feeder -d 3 </cmd>
    </daemon>
    <daemon>
      <cmd>transitioner -d 3 </cmd>
    </daemon>
    <daemon>
      <cmd>file_deleter -d 3 </cmd>
    </daemon>
  </daemons>
</boinc>
```

Para a validação dos resultados, precisamos de adicionar o validador à lista de deamons. Neste exemplo, esperamos bit-identity entre ficheiros de resultados do mesmo ficheiro de entrada (input).

Apenas para exemplo, vamos chamar o validador directamente, apesar de isto não ser boa pratica, noutras situações:

```
/usr/lib/boinc-server-maker/bin/
  ample_trivial_validator -app upper_case --
  one_pass -d 4
```

As unidades de trabalho (work unit), precisam de ser validadas, e depois atribuídos os créditos aos utilizadores. Fazer isto manualmente seria cansativo, repetitivo e aborrecido, além de entediante! Portanto, podemos optar por criar um ficheiro cron, para realizar a tarefa, ou utilizar o `deamon` que o BOINC disponibiliza no `config.xml`, acrescentando as seguintes linhas ao `config.xml`:

```
<daemon>
  <cmd>
    sample_bitwise_validator -app upper_case --
                                sleep_interval 600 -d 3
  </cmd>
</daemon>
```

Neste exemplo consideramos o intervalo de tempo 10 minutos (600 segundos), e `verbose (-d)` a nível 3 numa escala que inicia a 1 (baixo) e vai até 4 (alto).

Para aceder aos resultados, teremos de usar um assimilador (`assimilator`), ainda que o utilizador conclua que de forma extremamente sumária apenas se estaria movimentar ficheiros, tal não corresponde na totalidade ao que realmente acontece, pelo que usaremos o assimilador de exemplo, fornecido com o BOINC.

O assimilador exemplo, nomeia os ficheiros a partir do nome da sua unidade de trabalho `workunit`.

Podemos ver o resultado executando o seguinte comando:

```
$ cat sample_results/test
```

Até aqui, vimos como instalar o BOINC Server, colocar uma aplicação no boinc server, gerar unidades de trabalho (workunits), e validar os resultados.

Numa próxima oportunidade veremos como adaptar o Boinc para as nossas próprias necessidades, como virtualizar e como escrever novas aplicações!

Até lá, boa programação!



## AUTOR



Escrito por **António C. Santos**

Programar é criar um novo mundo escrevendo código, cumprindo os mais elementares desígnios da vida, “aprender, ensinar, criar, partilhar, melhorar e seguir”. Formou-se no Instituto Politécnico de Viana do Castelo. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário desde 2012, actualmente exerce funções de mentor voluntário na plataforma MOOC Coursera . Twitter: [@apocsantos](https://twitter.com/apocsantos)



# A PROGRAMAR

**Introdução ao Sonatype Nexus**

**Adobe PhoneGap Build para construção de aplicações móveis híbridas**

**SOAPUI Uma ferramenta muito útil para quem desenvolve Webservices**

**Depp Learnign Passo a Passo**

**Introdução ao SASS**

**C# - Excel**

## Introdução ao Sonatype Nexus

Há algum tempo atrás, enfrentei um desafio ao gerir/manter um projeto de software. Porque o cliente em que estava trabalhava na altura, tinha múltiplas equipas de desenvolvimento a trabalhar em múltiplas tecnologias e, admitamos, trabalhar com equipas que não tem estrutura e organização é um desespero.

Então, vagueei pela Internet em busca de uma infraestrutura onde fosse possível guardar todos os artefactos dos "projetos", independentemente da tecnologia utilizada. Foi então que me deparei com alguns, mas a minha predileção foi para Nexus Repository OSS versão 3.x.

O porquê da minha decisão? Principalmente por três razões:

1. Suporte para múltiplos formatos: Bower, Docker, Git, LFS, Maven, npm, NuGet, PyPI, Ruby, Gems, Yum and Proxy.
2. Têm uma versão gratuita.
3. Têm uma imagem Docker.

Parece-me suficiente e fui testar.

### Let's start

Vamos começar por criar o nosso docker-compose.yml:

```
version: '3'
#-----
# NETWORK
#-----
networks:
  CI-Network:
    driver: bridge

#-----
# VOLUMES
#-----
volumes:
  nexus-volume:

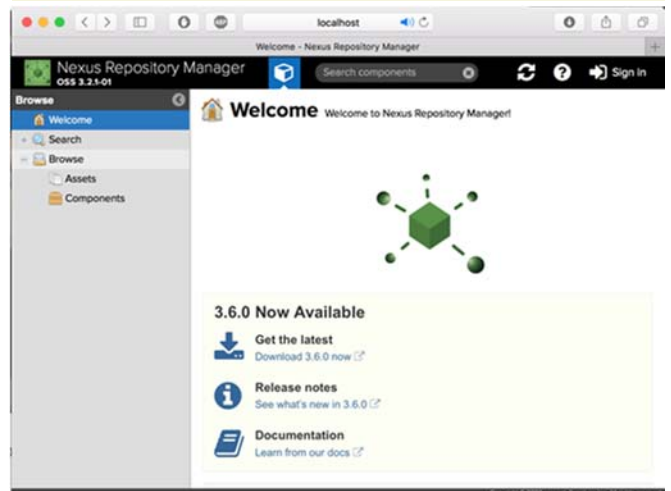
services:
#-----
# NEXUS
#-----
  nexus:
    image: 'sonatype/nexus3:3.2.1'
    ports:
      - '8090:8081'
    volumes:
      - nexus-volume:/nexus-data
    networks:
      - CI-Network
```

Gravamos o ficheiro, arrancamos o container e analisamos os logs:

```
$> docker-compose up -d
$> docker-compose logs -f
```

Após alguns instantes o Nexus está disponível em:

<http://localhost:8090/>



De acordo com a página do Nexus Docker hub (<https://hub.docker.com/r/sonatype/nexus/>):

As credenciais por omissão são:

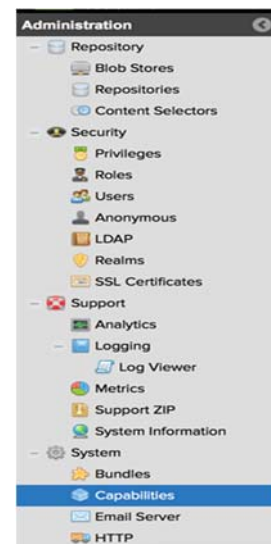
admin / admin123

Por isso, depois do login, **mudem a password de administração** ainda que seja para efeitos de testes.

Para que possamos criar um conjunto de credenciais para as aplicações submeterem/obterem os seus artefactos, temos que ir "Server Administration and Configuration".



No menu à esquerda temos um conjunto de opções disponíveis:







# A PROGRAMAR

## INTRODUÇÃO AO SONATYPE NEXUS

### Maven Snapshot

**Name:** A unique identifier for this repository  
maven-snapshots

**Format:** The format of the repository (i.e. maven2, docker, raw, nuget...)  
maven2

**Type:** The type of repository (i.e. group, hosted, or proxy)  
hosted

**URL:** The URL used to access this repository  
http://localhost:8090/repository/maven-snapshots/

**Online:**  If checked, the repository accepts incoming requests

### Maven 2

**Version policy:**  
What type of artifacts does this repository store?  
Snapshot

**Layout policy:**  
Validate that all paths are maven artifact or metadata paths  
Strict

### Hosted

**Deployment policy:**  
Controls if deployments of and updates to artifacts are allowed  
Allow redeploy

Se repararem, a diferença entre “Maven Release” e “Maven Snapshot” é a capacidade de se poder fazer re-deploys.

Temos a informação necessária para configurar o nosso projeto Java para usar este repositório.

### Configuring Maven

Vamos ao site do Maven (<https://maven.apache.org/download.cgi>) fazemos download da última versão, instalamos e configuramos (<https://maven.apache.org/install.html>).

Editamos o ficheiro \$MAVEN\_HOME/conf/settings.xml e adicionamos a seguintes secções:

```
<!-- Change the default local repository localization -->
<localRepository>/usr/local/repositories/Maven-Repo</localRepository>

<!-- Set the artefact repository credentials -->
<servers>
  <server>
    <id>nexus-maven-snapshots</id>
    <username>mvnuser</username>
    <password>mvnpassword</password>
  </server>
  <server>
    <id>nexus-maven-releases</id>
    <username>mvnuser</username>
    <password>mvnpassword</password>
  </server>
</servers>
<mirrors>
  <mirror>
    <id>nexus-maven-central</id>
    <mirrorOf>*</mirrorOf>
    <name>nexus-maven-central</name>
    <url>http://localhost:8090/repository/maven-public/</url>
  </mirror>
</mirrors>
```

E já está. :-).

Agora só precisamos de configurar o projeto Maven.

### Maven Project

Utilizando um IDE (Eclipse, STS, etc.) criem um projeto e adicionem uma dependência qualquer e através do terminal executem:

```
mvn clean install
```

Na execução vão ver a dependência a ser descarregada a partir do nosso servidor de Nexus e a ser copiada para o repositório local da nossa máquina.

Vamos tornar isto mais interessante, fazendo upload do nosso artefacto para o servidor de Nexus.

No ficheiro pom.xml do projeto adicionem a seguinte secção:

```
<distributionManagement>
  <snapshotRepository>
    <id>nexus-maven-snapshots</id>
    <url>http://localhost:8090/repository/maven-snapshots/</url>
  </snapshotRepository>
  <repository>
    <id>nexus-maven-releases</id>
    <url>http://localhost:8090/repository/maven-releases/</url>
  </repository>
</distributionManagement>
```

Agora executem:

```
mvn clean deploy
```

Após um projeto ser “build”, vemos o artefacto a ser deployed.

Este que surge a questão: Como é que o Maven sabe qual dos repositórios o artefacto deve ser submetido?

A resposta a esta questão está no ficheiro pom do projeto, na secção inicial que têm a versão do componente. Se a versão menciona SNAPSHOT, o Maven submete para o repositório configurado como SNAPSHOT. Caso contrário, instala no repositório de RELEASE.

Se formos agora ao servidor Nexus e navegarmos pelo “Maven Public” nos “Components”, encontramos os artefactos (proxy, snapshots e releases).

Name	Group	Version	T
maven-testing	pt.funlisa.entertainment	0.0.1	x
maven-testing	pt.funlisa.entertainment	0.0.1-20190927145529-1	x
maven-testing	pt.funlisa.entertainment	0.0.1-20190927145628-2	x
maven-testing	pt.funlisa.entertainment	0.0.1-20190927150228-3	x
maven-testing	pt.funlisa.entertainment	0.0.1-20190928174757-4	x
plexus-build-api	org.sonatype.plexus	0.0.4	x
maven-shared-utils	org.apache.maven.sher...	0.1	x
antlr	org.antlr	3.5.2	x
antlr-util	org.antlr	3.5.2	x
google	com.google	1	x
google-collectors	com.google.collectors	1.0	x
plexus-compiler	org.codehaus.plexus	1.0	x
maven-testham	org.apache.maven	1.0	x
commons-cdi	org.jboss.cdi	1.0	x
plexus	org.codehaus.plexus	1.0.1	x
plexus	org.codehaus.plexus	1.0.12	x
plexus-containers	org.codehaus.plexus	1.0.3	x
plexus	org.codehaus.plexus	1.0.4	x
plexus-utils	org.codehaus.plexus	1.0.4	x
plexus	org.codehaus.plexus	1.0.8	x
plexus-interactivity-api	org.codehaus.plexus	1.0-alpha-4	x
plexus-sink-api	org.apache.maven.plexus	1.0-alpha-7	x

# A PROGRAMAR

## INTRODUÇÃO AO SONATYPE NEXUS

“ (...)um desafio ao gerir/manter um projeto de software. Porque o cliente em que estava trabalhava na altura, tinha múltiplas equipas de desenvolvimento a trabalhar em múltiplas tecnologias e, admitamos, trabalhar com equipas que não tem estrutura e organização é um desespero.(...)”

Com este tipo de infraestrutura, damos um passo em frente para desacoplar as dependências dos projetos e tornar as workflows de ciclos devops mais versáteis.

“ . Suporte para múltiplos formatos: Bower, Docker, Git, LFS, Maven, npm, NuGet, PyPI, Ruby, Gems, Yum and Proxy.

- Têm uma versão gratuita.
- Têm uma imagem Docker.

”

### Conclusão

É o que basta. Uma pequena introdução ao Nexus com um exemplo de configuração em Maven. Configurando e utilizando o repositório em projetos .NET com Nugets, é tão simples como o demonstrado com projetos em Java.

### Referencias:

<https://www.sonatype.com/nexus-repository-oss>

<https://hub.docker.com/r/sonatype/nexus3/>

<https://maven.apache.org/index.html>

 NexusOSS  
Sonatype

## AUTOR



Escrito por Nuno Cancelo

Curioso por natureza e engenheiro informático por formação. Desde muito cedo me despertou o interesse pelos computadores e informática com o famoso Spectrum. Tenho um gosto peculiar por aprender novas coisas frequentemente mesmo que não venha a trabalhar com elas e otimizar os sistemas aumentando a sua performance.



## Adobe PhoneGap Build para construção de aplicações móveis híbridas

### Introdução

No artigo “Criar uma aplicação móvel com jQuery Mobile” [Programar #58] vimos como usar a *framework* jQuery Mobile para programar uma aplicação Web direcionada a dispositivos móveis. Neste artigo, vamos ver como usar o serviço PhoneGap Build da Adobe para gerar uma aplicação para Android.

### Adobe PhoneGap

PhoneGap é uma *framework* de desenvolvimento de aplicações móveis híbridas. Aplicações híbridas são aplicações que combinam componentes nativos e componentes Web. Do ponto de vista do utilizador e da plataforma móvel, uma aplicação híbrida é indistinguível de uma aplicação nativa. No entanto, internamente, uma aplicação híbrida utiliza um componente *Web View* que contém a maioria do conteúdo e lógica da aplicação – ou seja, a aplicação é essencialmente programada como se de uma aplicação web se tratasse. A *framework* PhoneGap, baseada em Apache Cordova, permite que as aplicações híbridas tenham acesso a funcionalidades nativas através de componentes específicos para cada plataforma móvel (e.g., Android, iOS, Windows) mas cuja interface é exposta em JavaScript (o programador não se preocupa com a plataforma).

### PhoneGap Build

O website PhoneGap Build é uma interface web simplificada para o processo de build da *framework* PhoneGap, permitindo-nos “empacotar” uma aplicação web numa aplicação nativa para as várias plataformas móveis.

Uma das vantagens deste website é que nem sequer temos de conhecer muito sobre PhoneGap uma vez que o website esconde a maioria dos detalhes.



Figura 1 – Ecrã da aplicação. Código completo: <https://goo.gl/P9dEHZ>

### Preparar a aplicação

A Figura 1 mostra um ecrã da aplicação construída no artigo da edição anterior. O link fornecido pode ser usado para descarregar o código da aplicação. A forma mais prática para descarregar o código será:

1. Seguir o link fornecido na Figura 1
2. Escolher a opção “Export” no canto inferior direito
3. Escolher a opção “Export .zip”

Depois de descarregar o ficheiro Zip e descomprimilo, o resultado será um único ficheiro HTML que iremos de seguida modificar ligeiramente.

O código original inclui referências ao jQuery e ao jQuery Mobile com recurso a endereços CDN remotos (e.g., <http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.css>).

Para evitar ligações de rede desnecessárias na nossa aplicação vamos converter estas referências em referências a ficheiros locais. Para isso temos de:

1. Editar o ficheiro index.html
2. Procurar todas as referências a ficheiros externos (CSS, imagens, JavaScript)
3. Seguir essas referências e descarregar os ficheiros, gravando-os na mesma pasta do projecto
4. Alterar as referências no ficheiro index.html, mantendo apenas o nome do ficheiro.

Por exemplo, a referência ao ficheiro CSS do jQuery Mobile seria convertido de:

```
<link rel='stylesheet prefetch' href='http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.css'>
```

Para:

```
<link rel='stylesheet prefetch' href='jquery.mobile-1.4.5.css'>
```

A estrutura final da pasta do projecto deverá ser algo como:

```
projecto-jquery-mobile
|-- index.html
|-- jquery.min.js
|-- jquery.mobile-1.4.5.css
|-- jquery.mobile-1.4.5.js
|-- space-cube.svg
```

# A PROGRAMAR

## ADOBE PHONEGAP BUILD PARA CONSTRUÇÃO DE APLICAÇÕES MÓVEIS HÍBRIDAS

Depois de fazermos e gravarmos estas alterações podemos voltar a comprimir a pasta do projecto num ficheiro Zip.

### Gerar o ficheiro keystore

As aplicações móveis distribuídas através de uma loja (e.g., Android Play, ou App Store), precisam de ser assinadas. Para tal é necessário a geração de um certificado que identifica a entidade que criou a aplicação e uma chave privada e uma pública. A chave privada é usada para assinar digitalmente a aplicação. A emissão de um certificado é feita tipicamente por uma entidade reconhecida globalmente, mas, neste artigo, iremos gerar nós próprios este certificado. Para tal recorreremos à ferramenta *keytool* do Java.

Para executar esta ferramenta, temos de usar uma linha de comando (Command Prompt no Windows, ou Terminal no Mac OS). Aqui vou exemplificar o comando em Windows (com fundo cinza o texto introduzido por mim, em resposta aos pedidos do *keytool*; [enter] representa o pressionar da tecla "Enter"):

```
C:\Users\jorge>"c:\Program Files\Java\jdk1.8.0_121
\bin\keytool.exe" -genkey -alias androidkey -v -
keystore android.keystore [enter]
Enter keystore password: supermegapass[enter]
Re-enter new password: supermegapass[enter]
What is your first and last name?
[Unknown]: Jorge Cardoso[enter]
What is the name of your organizational unit?
[Unknown]: [enter]
What is the name of your organization?
[Unknown]: [enter]
What is the name of your City or Locality?
[Unknown]: Porto[enter]
What is the name of your State or Province?
[Unknown]: [enter]
What is the two-letter country code for this unit?
[Unknown]: PT[enter]
Is CN=Jorge Cardoso, OU=Unknown, O=Unknown,
L=Porto, ST=Unknown, C=PT correct?
[no]: yes[enter]
```

Generating 1,024 bit DSA key pair and self-signed certificate (SHA1withDSA) with a validity of 90 days

```
for: CN=Jorge Cardoso, OU=Unknown,
O=Unknown, L=Porto, ST=Unknown, C=PT
Enter key password for <androidkey>
(RETURN if same as keystore password):
[enter]
[Storing android.keystore]
```

O ficheiro resultante, especificado no parâmetro "keystore" do *keytool*, é colocado na mesma pasta onde executamos o *keytool* (no meu caso em C:\Users\jorge\android.keystore). É este ficheiro que irá ser adicionado ao PhoneGap Build. É importante notar que o parâmetro "alias" do *keytool* terá de ser introduzido no site do PhoneGap Build (assim como as passwords que escolhemos) pelo que convém memorizar.

### Website PhoneGap Build

De seguida, podemos registarmo-nos no site PhoneGap

Build, escolhendo o plano grátis (ver Figura 2).

### Adicionar o ficheiro keystore ao PhoneGap Build

Antes de configurarmos a nossa aplicação no PhoneGap Build, devemos fazer *upload* do ficheiro android.keystore, tal como indicado na Figuras 3 e 4. É necessário cuidado ao introduzir o "alias" uma vez que este deve ser exactamente igual ao parâmetro "alias" que usamos no comando *keytool*.

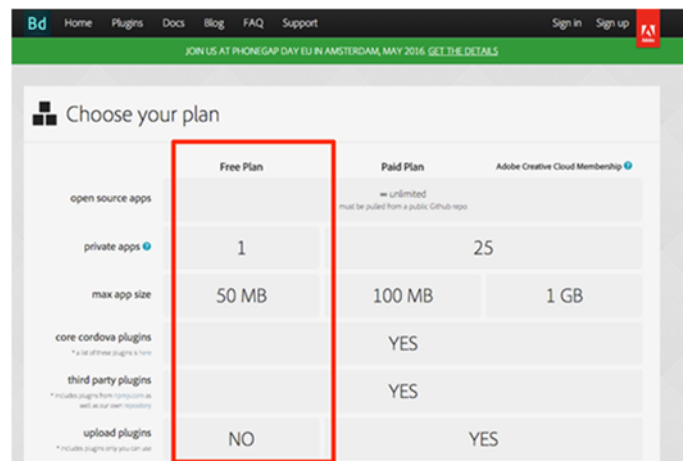


Figura 2 – Registo no site PhoneGap Build. <http://build.phonegap.com>.

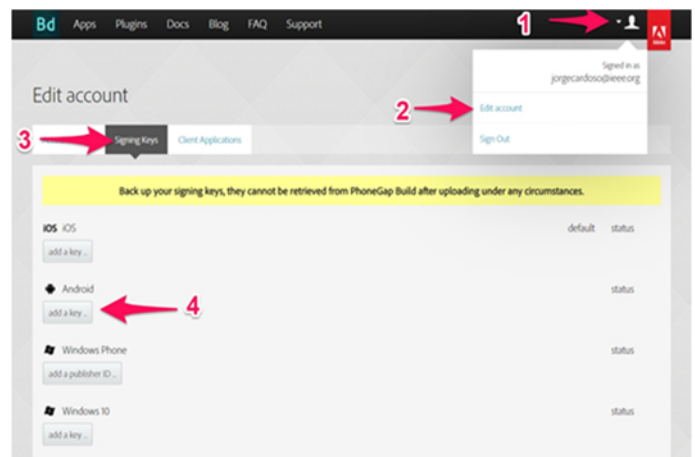
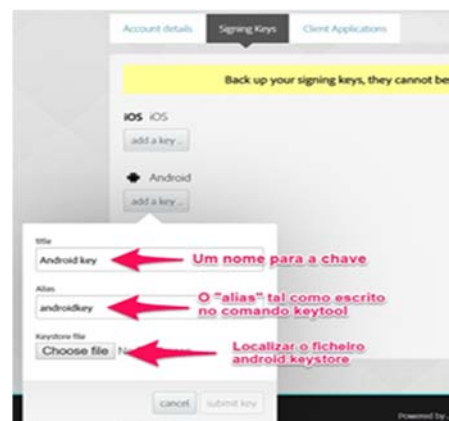


Figura 3 – Adição de chave Android.





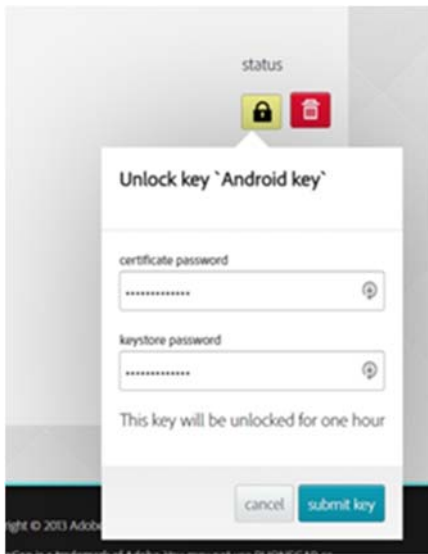


Figura 4 – Upload do ficheiro android.keystore e desbloqueamento da chave.

Depois de localizarmos o ficheiro android.keystore e clicarmos em “submit key”, devemos desbloquear a keystore clicando no ícone do cadeado (do lado direito na mesma página) e introduzindo a password que escolhemos para a nossa keystore quando corremos o comando keytool (ver Figura 4, lado direito).

### Criar a aplicação no PhoneGap Build

O passo seguinte será configurar uma aplicação no PhoneGap Build e carregar o nosso código fonte. Para tal, acessemos à secção “Apps” e fazemos upload do ficheiro zip que criamos no início.

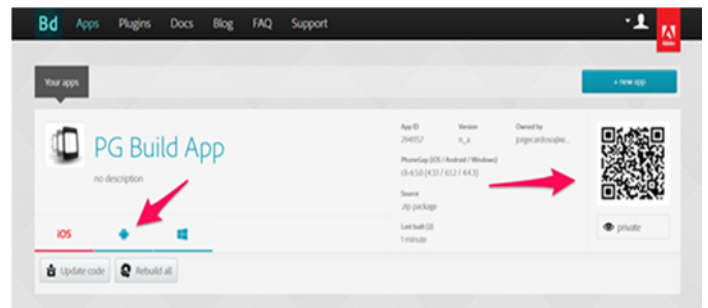


Figura 5 – Configurar aplicação, carregar código fonte e correr processo de build.

Depois do upload, o PhoneGap Build mostra-nos um ecrã em que podemos definir um nome e descrição para a nossa aplicação e iniciar o processo de build. A Figura 5 mostra os passos necessários.

### Testar a aplicação

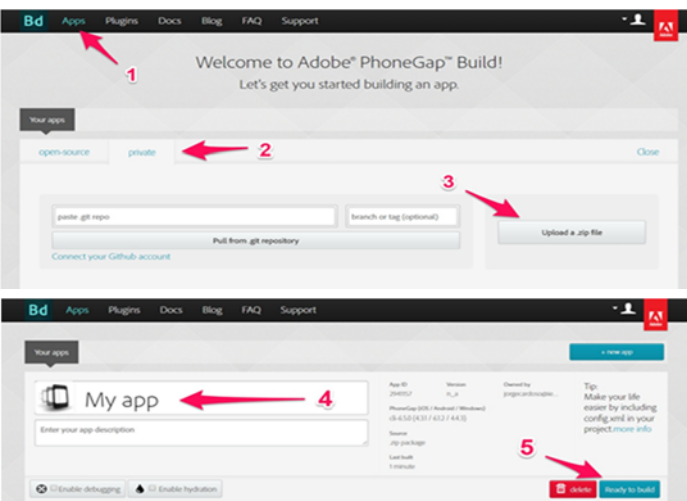
O último passo é instalar e testar a aplicação no nosso dispositivo móvel. Há duas formas de o fazer. Uma é descarregar manualmente o ficheiro APK (*Android Package file*), clicando no ícone relativo ao Android (Figura 5, em baixo); transferindo o ficheiro para o dispositivo móvel; e finalmente fazendo a instalação executando o APK. A outra forma é usando um leitor de QR Codes e apontando o dispositivo móvel para o QR Code que aparece no ecrã (Figura 5, em baixo).

O ficheiro APK que podemos descarregar depois do processo de *build* é também o ficheiro que precisamos para distribuir a nossa aplicação através da loja Google Play.

### Conclusão

Os passos que vimos neste artigo são suficientes para, depois de termos uma aplicação Web programada para dispositivos móveis, conseguirmos gerar e distribuir uma aplicação como qualquer outra aplicação nativa.

A *framework* de desenvolvimento híbrido PhoneGap tem, obviamente, muitas funcionalidades que não exploramos aqui. No entanto, para quem está sobretudo habituado a programação Web e quer experimentar ou distribuir de forma rápida a sua aplicação, esta pode ser uma boa forma de o fazer.



## AUTOR



### Escrito por Jorge C. S. Cardoso

Professor auxiliar convidado no Departamento de Engenharia Informática da Universidade de Coimbra onde lecciona disciplinas relacionadas com tecnologias web, programação, e interação humano-computador. É autor do livro "Java para Telemóveis" editado pela FEUP Edições. Licenciado em Engenharia Informática pela Universidade do Porto (FEUP), mestre em Sistemas Móveis e doutor em Tecnologias e Sistemas de Informação pela Universidade do Minho (Escola de Engenharia).

## SOAPUI Uma ferramenta muito útil para quem desenvolve Webservices

### O SOAPUI

O SoapUI é uma aplicação *opensource* utilizada em testes de Webservices de arquiteturas orientadas a serviços (SOA) ou transferências de estado representacional (REST). É uma aplicação multi-plataforma desenvolvida em JAVA. Existe também uma versão profissional que suporta múltiplas origens de dados de testes, geração de testes automatizados, análise inteligente de pedidos, etc.

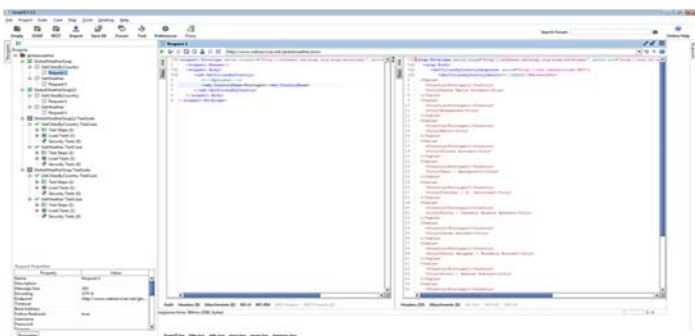


Figura 1: Aspecto visual da aplicação

Entre outras funcionalidades, o SOAPUI permite:

- Invocação de Webservices
- Inspeção de Webservices
- Geração de testes de carga em Webservices
- Geração de testes de segurança em Webservices
- Geração de documentação de Webservices

### Obter o SOAPUI

O download do SOAPUI pode ser feito obter em <https://www.soapui.org/downloads/soapui.html>



Figura 2: Página de Download da aplicação

### Utilização do SOAPUI

Para demonstrar as principais funcionalidades do SOAPUI utilizaremos o Webservice gratuito Global Weather disponibilizado em <http://www.webservices.net/globalweather.asmx>. Este Webservice tem 2 métodos: GetCitiesByCountry (obter cidades principais por país) e GetWeather (obter o tempo por país e cidade).

O endereço do descritor do Webservice (WSDL) que será utilizado, é o seguinte:

<http://www.webservices.net/globalweather.asmx?WSDL>

### O projecto SOAPUI

Quando se cria um projecto SOAPUI, por exemplo, um projecto de Webservice SOAP (File à New SOAP Project) indica-se o endereço do descritor (WSDL) e escolhem-se os tipos de funcionalidades a utilizar.

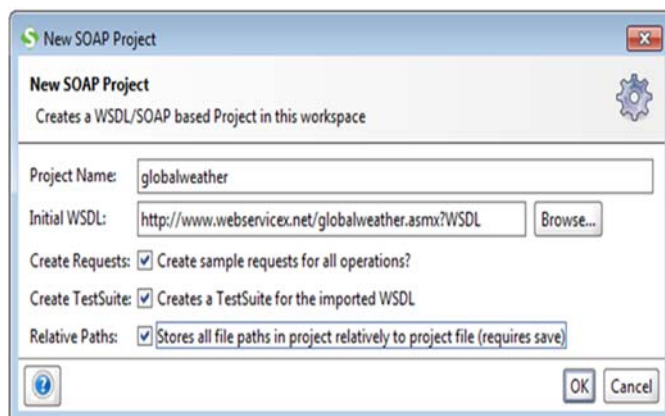


Figura 3: Criação de um projecto SOAP no SOAPUI

**Project Name:** Nome do projecto

**Initial WSDL:** Endereço do descritor (WSDL) do Webservice

**Create Requests:** Invocações e inspeções de Webservices

**Create TestSuite:** Pacote de testes de invocação, de carga e de segurança

Caso seja escolhida a criação do pacote de testes é solicitada a natureza dos testes e quais os métodos para os quais vão ser gerados.

# A PROGRAMAR

SOAPUI UMA FERRAMENTA MUITO ÚTIL PARA QUEM DESENVOLVE WEBSERVICES

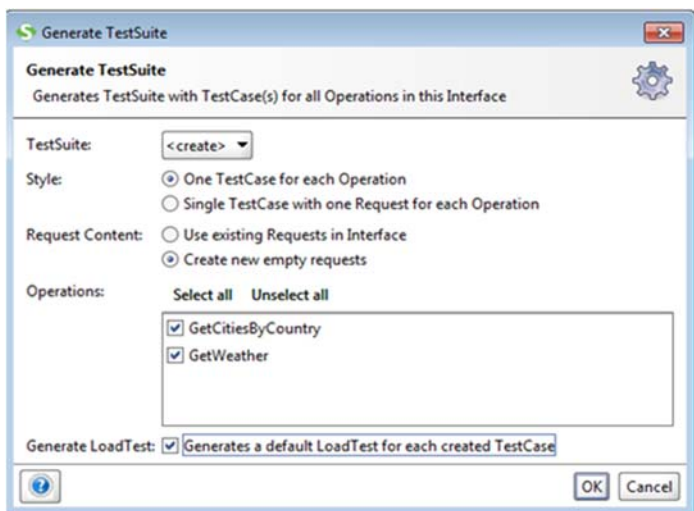


Figura 4: Criação de pacote de testes

NOTA: No caso de WebServices SOAP o SOAPUI suporta testes às versões 1.1 e 1.2.

Uma vez gerado o projecto o SOAPUI terá um aspecto semelhante à seguinte imagem.

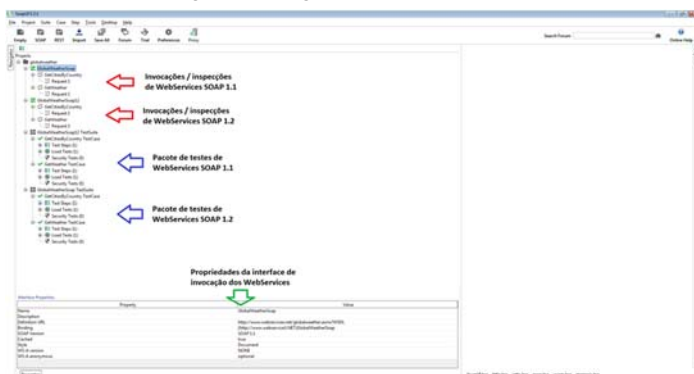


Figura 5: Estrutura de um projecto SOAPUI

## Inspeção de WebServices

Com esta funcionalidade podem-se gerar pedidos, invocar os WebServices com os pedidos gerados, uma ou várias

vezes, conforme pretendido. Estas invocações serão consideradas como testes que podem ser gravados e reutilizados.



Figura 6: Aspecto da invocação de WebServices

## Geração de testes de carga

Com esta funcionalidade podem-se gerar testes de carga tendo como base pedidos de inspeção de WebServices previamente criados.

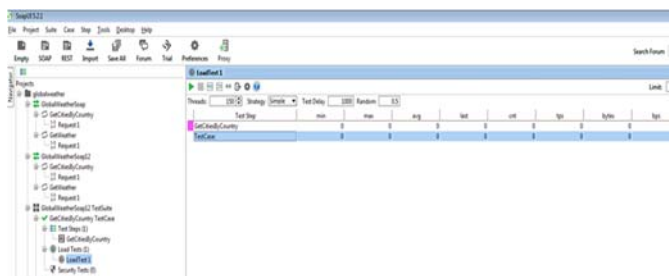


Figura 7: Aspecto de um teste de carga

Podem-se definir a estratégia de teste (burst - ruptura, simple - simples, variance – com variância e thread – com base em threads).

## Geração de testes de segurança

Esta funcionalidade permite gerar testes de segurança dos WebServices. Podem-se simular ataques (por exemplo XML Bomb, Fuzzing Scan, Xpath Injection, etc.).

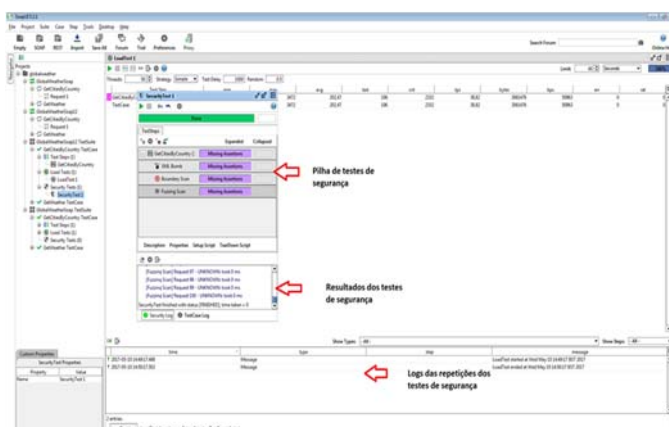


Figura 8: Exemplo de um teste de segurança com vários ataques

## Geração de código .NET (Proxy)

Esta funcionalidade permite a geração de código correspondente às interfaces de invocação dos WebServices.

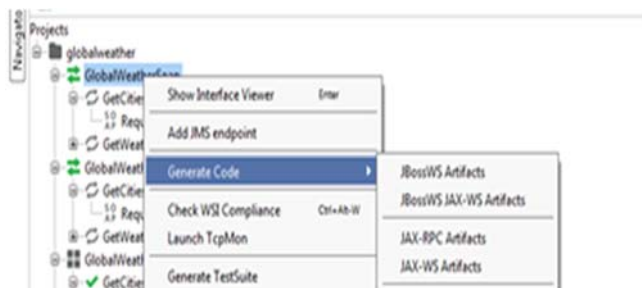


Figura 9: Menu de geração de código

# A PROGRAMAR

## SOAPUI UMA FERRAMENTA MUITO ÚTIL PARA QUEM DESENVOLVE WEBSERVICES

O código gerado pode ser .NET, XMLBeans, Apache Axis, entre outros.

### Geração de documentação

Esta funcionalidade permite a geração de documentação técnica dos WebServices.



Figura 11: Aspecto do menu de geração de documentação de WebServices

A documentação é feita em inglês e exportada num formato HTML.

### Web Service: GlobalWeather

Web Service: GlobalWeather  
Target Namespace: http://www.vobonix.com/.NET

Port GlobalWeatherSoap Port type: Source code  
Location: http://www.vobonix.com/soap/globalweather.aspx  
Protocol: SOAP  
Transport protocol: SOAP over HTTP  
Operations:  
1. GetCityCountry Detail Source code  
2. GetWeather Detail Source code

Port GlobalWeatherSoap12 Port type: Source code  
Location: http://www.vobonix.com/soap/globalweather.aspx  
Protocol: SOAP  
Transport protocol: SOAP over HTTP  
Operations:  
1. GetCityCountry Detail Source code  
2. GetWeather Detail Source code

Port GlobalWeatherHttpGet Port type: Source code  
Location: http://www.vobonix.com/soap/globalweather.aspx  
Protocol: HTTP  
Default method: GET

Figura 12: Exemplo de documentação gerada

### Em suma

O SOAPUI é uma das ferramentas de apoio ao desenvolvimento e testes de WebServices que se pode tornar muito útil. Mesmo a versão opensource e gratuita fornece um conjunto de funcionalidades que podem ser alteradas, configuradas e reuti-

lizáveis. É uma boa ferramenta a ter em conta quando se trabalha com WebServices.

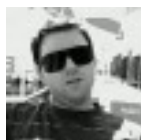
Links úteis

Site oficial - <https://www.soapui.org/>

Tutoriais SOAP UI - <https://www.soapui.org/tutorials.html>

“ O SoapUI é uma aplicação opensource utilizada em testes de Webservices de arquiteturas orientadas a serviços (SOA) ou transferências de estado representacional (REST). É uma aplicação multi-plataforma desenvolvida em JAVA. ”

## AUTOR



Escrito por Sérgio Ildefonso

Microsoft Certified Professional. MCTS Microsoft SQL Server 2008, Database Development; MCTS Microsoft .NET Framework - Application Development Foundation; Licenciado em Engenharia Informática pela Faculdade de Ciências da Universidade de Lisboa. Developer em tecnologias Microsoft (Windows, Web, Sql Server, etc.),



## Xamarin Forms – usando Xamarin Effects

Este artigo tem como objetivo mostrar como usar efeitos no desenvolvimento de aplicações móveis usando a framework Xamarin Forms.

### Introdução

A framework Xamarin Forms permite abstrair a camada de *user interface* entre as diversas plataformas, ie, através desta framework podemos escrever um único código que define toda a aplicação e que irá ter a sua representação gráfica respeitando é claro o controlos gráficos de cada plataforma. Refiro-me é claro às plataformas iOS, Android e Windows, cujo UX difere em alguns aspetos.



Figura 1. Elementos gráficos relativos a iOS e Android - diferenças ao nível controlos e icons

Para muitos a figura 2 é reconhecida quando nos referimos à percentagem de código que podemos reutilizar entre usando o “tradicional desenvolvimento” – Xamarin.Android ou Xamarin.iOS e quando usamos a framework Xamarin Forms.

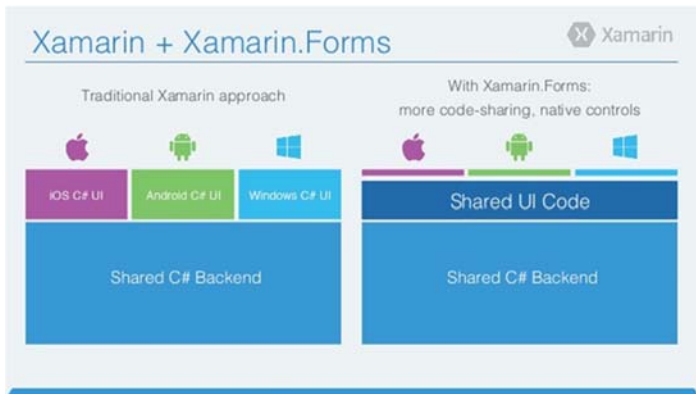


Figura 2. Xamarin vs Xamarin Forms

O uso da framework (Xamarin Forms) pode não se aplicar a todos os tipos de aplicações, pois os diversos requisitos e as especificidades técnicas da aplicação podem ser determinísticas na escolha – dou o exemplo de aplicações com muitas animações ou aplicações que a *user interface* usa controlos que são específicos da plataforma, e claro está dificulta no uso desta framework. Conceptualmente esta framework tem um interessante muito grande pela convergência que se tem vindo a sentir na reutilização de código e na redução do tempo de desenvolvimento. É claro está, que para o bom uso da mesma é necessário compreender toda a abstração inerente e também a forma como os diversos *garbage collections* são geridos pois

isto é relevante na gestão de memória da aplicação e deteção de memory leaks (criados pelo próprio programadores) - muitas das vezes estes aspetos não são tido em conta durante o desenvolvimento.

Uma das funcionalidades fornecidas pela Xamarin ao nível desta framework, logo desde o inicio, foi o chamado “[Renders](#)” que permite alterar o controlo de base, ie, é possível através do *render* obter o controlo nativo da plataforma e claro está aplicar as alterações pretendidas. Sem dúvidas que os *renders* são muito interessantes, mas quando se começa a criar muitos renders podemos concluir que o uso do tradicional desenvolvimento (Xamarin.iOS ou Xamarin.Android) poderá ser o melhor (depende muito de caso para caso).

Mais recentemente, a Xamarin desenvolveu uma nova funcionalidade chamada “[Effects](#)” que tem como finalidade facilitar a customização dos controlos nativos, e desta forma a criação de “*renders*” deixa de fazer sentido em muitos casos, e todo o desenvolvimento fica então mais simplificado.

### Descrição

Para exemplificar o uso de “[Effects](#)”, usemos o controlo Button como exemplo - vejam a seguinte figura:

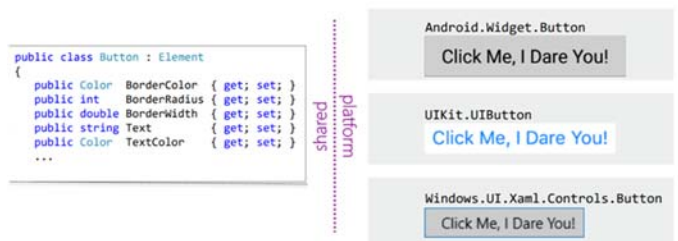


Figura 3. Classe Button e as diversas representações em cada plataforma

Do lado esquerdo temos a classe Button da framework Xamarin Forms e do lado direito temos o controlo gráfico e o namespace a que pertence o respetivo controlo em cada plataforma – Button em Android e Windows e UIButton em iOS.

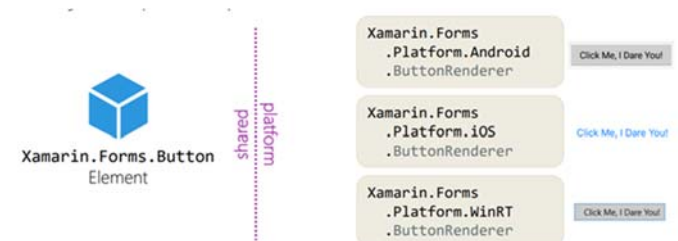


Figura 4. A classe “[ButtonRenderer](#)”, definida no projeto target, será a classe que permite fazer a tradução entre o Button de Xamarin Form para o controlo nativo.

# A PROGRAMAR

## XAMARIN FORMS – USANDO XAMARIN EFFECTS

Antes de existirem os “Effects” os *renders* eram estendidos, ie, a class “*ButtonRenderer*” era estendida em cada um dos targets, para se poder alterar o controlo nativo. Em alguns casos, a extensão dos “*Renders*” obriga a criar um novo controlo do lado de Xamarin Forms, ie, “*MyButton*” que herda da classe “*Button*”, para que desta forma o “*Render*” se aplique apenas ao controlo específico. Para diferenciar entre as duas funcionalidades, um bom exemplo do uso de “*Effects*” é aplicação e ajuste do *theme* da aplicação em que o controlo nativo se mantém e apenas se pretende ajustar algumas propriedades nativas – podendo ser definido de várias formas :

- usando o nome do “*Effects*”
- usando *RoutingEffect*

Por sua vez, podem ser afetados ao controlo de Xamarin Forms através da respetiva propriedade para o efeito ou através de *attached* properties.

### Criação de um “Effect”

Suponhamos que temos um controlo que apresenta texto, uma *Label*, de seguida vamos ver como podemos definir o tipo de letra para cada target.

Começamos então por definir o *effect* em cada plataforma.

#### iOS

```
[assembly: ResolutionGroupName("Xamarin")]
[assembly: ExportEffect(typeof(MyCustomFontEffect),
    "CustomFontEffect")]
namespace CustomFont.iOS
{
    class MyCustomFontEffect : PlatformEffect
    {
        UIFont oldFont;

        protected override void OnAttached()
        {
            if (Element is Label == false)
                return;

            var label = Control as UILabel;
            oldFont = label.Font;

            label.Font = UIFont.FromName
                ("Pacifico", label.Font.PointSize);
        }

        protected override void OnDetached()
        {
            if (oldFont != null)
            {
                var label = Control as UILabel;
                label.Font = oldFont;
            }
        }
    }
}
```

#### Android

```
[assembly: ResolutionGroupName("Xamarin")]
[assembly: ExportEffect(typeof(MyCustomFontEffect),
    "CustomFontEffect")]
```

```
namespace CustomFont.Droid
{
    class MyCustomFontEffect : PlatformEffect
    {
        Typeface oldFont;

        protected override void OnAttached()
        {
            if (Element is Label == false)
                return;

            var label = Control as TextView;
            oldFont = label.Typeface;

            var font = Typeface.CreateFromAsset
                (Android.App.Application.Context.Assets,
                    "Pacifico.ttf");
            label.Typeface = font;
        }

        protected override void OnDetached()
        {
            if (oldFont != null)
            {
                var label = Control as TextView;
                label.Typeface = oldFont;
            }
        }
    }
}
```

#### Windows

```
[assembly: ResolutionGroupName("Xamarin")]
[assembly: ExportEffect(typeof(MyCustomFontEffect),
    "CustomFontEffect")]
namespace CustomFont.UWP
{
    class MyCustomFontEffect : PlatformEffect
    {
        Windows.UI.Xaml.Media.FontFamily
            oldFontFamily;
        protected override void OnAttached()
        {
            if (Element is Label == false)
                return;

            TextBlock tb = Control as TextBlock;
            oldFontFamily = tb.FontFamily;

            tb.FontFamily = new
                Windows.UI.Xaml.Media.FontFamily(@"Assets/
                    Pacifico.ttf#Pacifico");
        }

        protected override void OnDetached()
        {
            if (oldFontFamily != null)
            {
                TextBlock tb = Control as Text-
                    Block;
                tb.FontFamily = oldFontFamily;
            }
        }
    }
}
```

De salientar, que o método *OnAttached* é chamado quando o *effect* é atribuído a um controlo de Xamarin Forms,

e o método *OnDetached* é chamado quando o *effect* deixa de estar associado ao controlo de Xamarin Forms. O método *OnElementPropertyChanged* é chamado quando ocorre alterações numa propriedade do controlo de Xamarin Forms que pode requerer alterações no controlo nativo associado.

De seguida vamos definir uma página em que vamos ter um Label à qual iremos aplicar o *effect* definido.

### XAML

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:ef="clr-namespace:ControlExplorer"
  x:Class="ControlExplorer.MainPage">
  <StackLayout x:Name="stackLayout" Padding="20" Spacing="10">
    <Label x:Name="labelWelcome" Text="Welcome to Effects!" />
    <Switch x:Name="switchEffects" Toggled="OnSwitchToggled" IsToggled="True" />
    <Label Text="Effects are a great way to apply platform-specific UI features to a Xamarin.Forms defined UI.">
      <Label.Effects>
        <ef:LabelFontEffect />
      </Label.Effects>
    </Label>
    <Button x:Name="buttonClick" Text="Click Count = 0" Clicked="OnButtonClicked" TextColor="White" BackgroundColor="Blue" />
    <Slider Maximum="255" ValueChanged="OnSliderColorValueChanged" />
  </StackLayout>
</ContentPage>
```

### C#

Para adicionar o *effect* ao controlo de Xamarin Forms:

```
private Effect _fontEffect;
public MainPage()
{
  InitializeComponent();
  _fontEffect = Effect.Resolve("Xamarin.CustomFontEffect");
  labelWelcome.Effects.Add(_fontEffect);
}
```

Para remover (em qualquer momento):

```
[assembly: ExportEffect(typeof(MyCustomFontEffect), "CustomFontEffect")]
```

### Criação de um RoutingEffect

A criação de "[RoutingEffects](#)" acaba por ser muito inte-

ressante, pelo fato de deixamos de usar string com o nome do *effect* e passamos a usar um objecto que define o *effect*. Portanto o "[RoutingEffect](#)" não é mais do que uma classe que fica definida no projeto partilhado e que permite usar um objecto para definir o *effect* em vez de usar o método *Resolve* e a string do nome. A sua implementação de exemplo é a seguinte

```
class LabelFontEffect : RoutingEffect
{
  public LabelFontEffect() : base("Xamarin.CustomFontEffect")
  {
  }
}
```

**Nota:** De salientar, que no "[RoutingEffects](#)" é usado o nome do render "*CustomFontEffect*", mas também o seu nome do "*ResolutionGroupName*" definido em cada *effect*.

```
[assembly: ResolutionGroupName("Xamarin")]
```

Destá forma, o efeito pode ser definido no controlo de Xamarin Forms da seguinte forma

```
<Label Text="Effects are a great way to apply platform-specific UI features to a Xamarin.Forms defined UI.">
  <Label.Effects>
    <ef:LabelFontEffect />
  </Label.Effects>
</Label>
```

Um *Effect* pode ser associado a qualquer controlo que suporte *effects*, no entanto, este *effect* deve ser usado tendo em conta se está associado (a um *Button* ou a uma *Label*), pois não devemos atribuir *effects* cujo controlo nativo o não lhe corresponda. Quando é necessário criar um *effect* que possa ser usado por vários controlos porque as propriedades são comuns, aí podemos criar um *effect* ao nível da classe pai, ie, da classe *Element* (por exemplo).

“ (...)uma nova funcionalidade chamada “Effects” que tem como finalidade facilitar a customização dos controlos nativos(...) ”

# A PROGRAMAR

## XAMARIN FORMS – USANDO XAMARIN EFFECTS

### Conclusão

Em conclusão, os “effects” vem facilitar toda a customização exigida pelos designers e desta forma, o desenvolvimento de aplicações móveis usando Xamarin Forms fica mais rápido, simplificado o desenvolvimento, e para além disso a qualquer momento podemos alterar os “effects” de cada controlo de Xamarin Forms, permitindo assim que em *runtime* seja aplicado o design que se pretende. Os effects para além de poderem ser atribuídos aos controlos podem ser definidos através de attached properties ou de styles o que é uma mais valia para a qualidade de código a desenvolver.

“ **A framework Xamarin Forms permite abstrair a camada de user interface entre as diversas plataformas, ie, através desta framework podemos escrever um único código que define toda a aplicação e que irá ter a sua representação gráfica respeitando é claro o controlos gráficos de cada plataforma.** ”

### Algumas referências

- [Introduction to Effects](#)
- [Creating an Effect](#)
- [Passing Parameters to an Effect](#)
- [Invoking Events from an Effect](#)
- [Customizing Xamarin.Forms Controls with Effects](#)

“ **Conceptualmente esta framework tem um interessante muito grande pela convergência que se tem vindo a sentir na reutilização de código e na redução do tempo de desenvolvimento.** ”

## AUTOR



Escrito por Sara Silva

Licenciada em Matemática pelo DMUC, e o seu foco de desenvolvimento está direccionado para a área Mobile, sendo a sua principal especialidade aplicações para Windows. Atualmente desenvolve na área do Windows, Xamarin, Azure, e é Microsoft MVP Mentor. É autora de vários artigos técnicos e tutoriais. A Sara foi condecorada com vários prémios com especial destaque: Microsoft MVP, Xamarin MVP, Telerik Developer Especialista, C# Corner MVP, TechNet Wiki - Technical Guru. O trabalho que vai sendo desenvolvido pela Sara pode ser seguido através do seu blog [www.saramgsilva.com](http://www.saramgsilva.com) e do twitter é [@saramgsilva](https://twitter.com/saramgsilva).

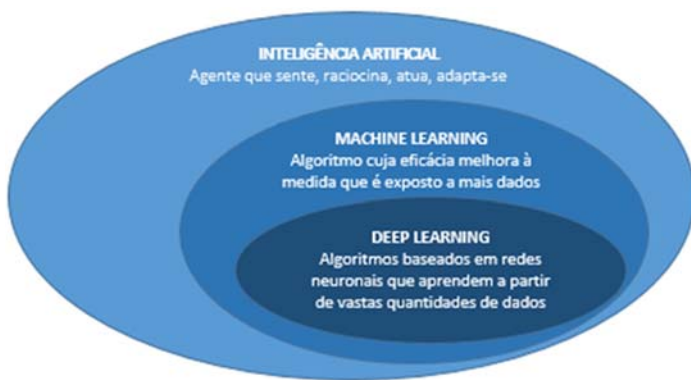




## DEEP LEARNING PASSO A PASSO

### Introdução

Muito se tem falado ultimamente no tema da Inteligência Artificial (IA) respetivas ramificações, razão pela qual decidi trazer um exemplo prático sobre o tema, nomeadamente um caso de uso de uma rede neuronal artificial (RNA), que através de um processo de treino (análise sucessiva de observações), infere/aprende correlações existentes num conjunto de dados (*dataset*). Mas primeiro vamos definir os conceitos de: Inteligência Artificial, *Machine Learning*, *Deep Learning*, sendo artigo relativo a este último.



### Caso de uso

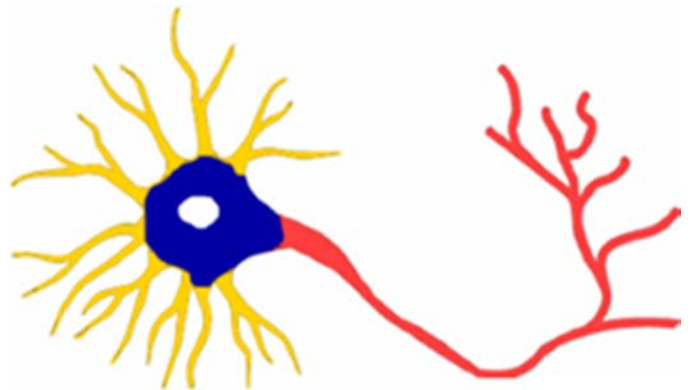
Mais do que os dados em si, o objetivo é apresentar na prática o funcionamento de uma RNA. Para o efeito, foram utilizados dados reais extraídos da base de dados pública do INE dos censos populacionais de 2011 (disponível online), onde a partir de um conjunto de variáveis não relacionados se vai tentar inferir se determinada zona populacional (subsecção estatística), tem mais mulheres ou homens (resposta binária).

Utilizei variáveis relativos a emprego, pensionistas e habilitações literárias (10 de mais de 120 disponíveis). Por exemplo, sabemos que em média as mulheres vivem mais anos, logo uma subsecção que tenha uma população maioritariamente envelhecida, então, com uma probabilidade alta, terá mais mulheres que homens: a variável relativa a população reformada terá peso superior a outras. Mas perceber estas relações vai ser o trabalho da rede.

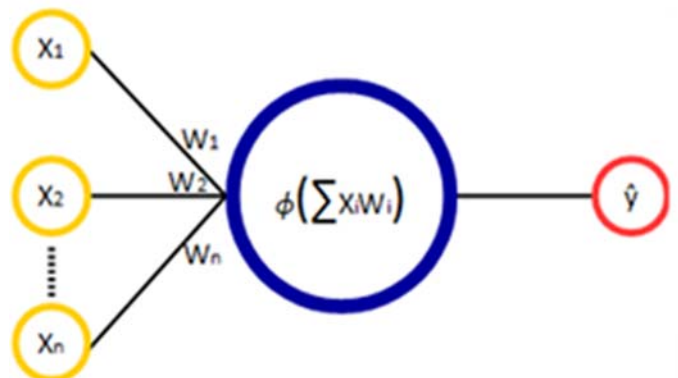
### A base - o neurónio

A estrutura de uma RNA é inspirada na estrutura neuronal de organismos inteligentes que adquirem conhecimento através da aprendizagem (treino), tendo como unidade base o neurónio artificial.

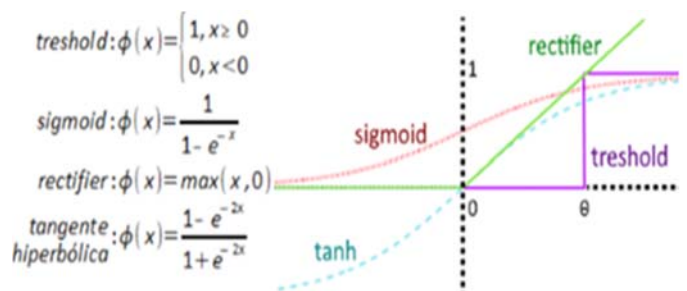
Nos organismos vivos, o neurónio é uma célula responsável pela condução dos impulsos nervosos e, portanto, funcionam como recetores, integradores e transmissores de impulsos. Dispõe (entre outras partes) de: dendritos (recebe impulsos), núcleo (integra impulsos recebidos), axónios (transmitem novos impulsos). Aqui, os impulsos serão a nossa informação.



Um neurónio artificial é constituído por um conjunto de entradas, às quais se associa um peso ( $W_i$ ), um núcleo onde é aplicada uma função de ativação ( $\phi$ ) ao somatório dos valores das entradas multiplicados pelos pesos ( $\sum X_i W_i$ ), e pelas saídas onde o valor obtido ( $\hat{y}$ ) é propagado para os neurónios seguintes. Durante o processo de aprendizagem os pesos são ajustados, ou seja, a relevância que cada neurónio atribui a cada um dos valores recebidos para melhorar a aprendizagem: uma variável mais importante no resultado deverá ter associado um peso maior. Este esquema tem o nome de *perceptron*.



Das muitas funções de ativação ( $\phi$ ) possíveis algumas das mais comuns são:



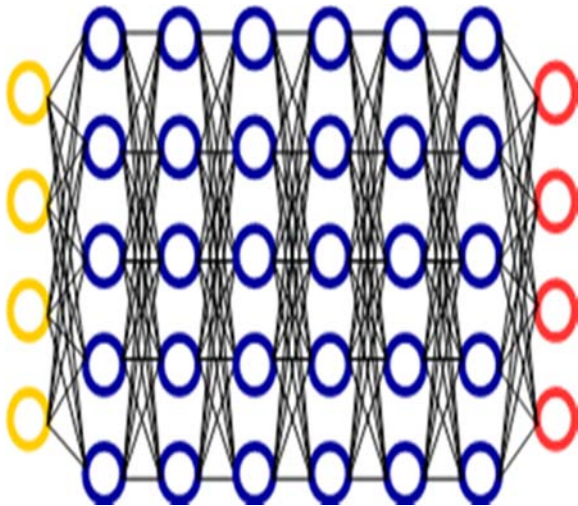


# A PROGRAMAR

## DEEP LEARNING PASSO A PASSO

### Rede neuronal

O modelo computacional de uma RNA é organizado em camadas (*layers*): *input* (amarelo, onde os valores são dados de entrada), *hidden* (azul, compostas por neurónios interligados), *output* (vermelho, obtém o valor final produzido pela rede).



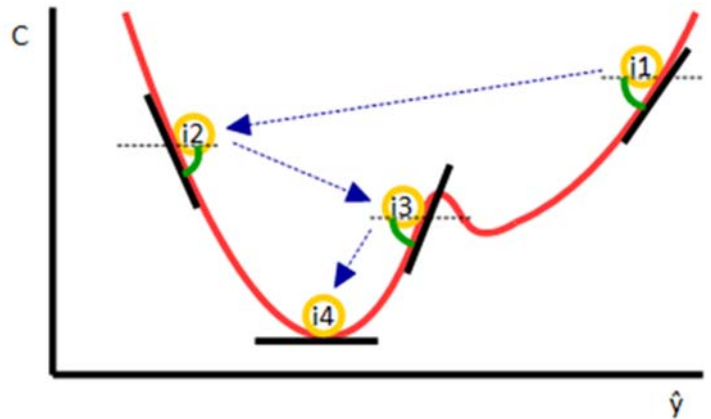
Os dados são introduzidos na rede (linha a linha), na *input layer* (um valor por nó). No nosso caso de uso, a *hidden layer* terá duas camadas de 4 neurónios cada (muito simples). A *output layer*, e uma vez que o valor pretendido é um binário, terá apenas 1 neurónio (no caso de uma categoria de valores teria tantos neurónios quantos valores possíveis da categoria), e uma vez que vai ser ativada por uma função *sigmoid* vai devolver uma probabilidade  $P(\hat{y} = 1)$  entre 0 e 1 a qual depois convertemos para o valor mais próximo.

### Aprendizagem

O treino consiste em ativar a rede sucessivamente com observações cujo resultado final ( $y$ ) é conhecido à partida, e compara-lo com o resultado obtido ( $\hat{y}$ ). De seguida calcula-se a distância entre ambos (erro) através de uma função de custo

(*cost*), como sendo: 
$$C = \frac{1}{2}(\hat{y} - y)^2$$
 e com base nesse erro atualizam-se os pesos ( $W_i$ ) da rede de forma a que nas próximas iterações os valores obtidos sejam cada vez mais aproximados dos valores reais.

Para encontrar o valor ideal dos pesos é necessário de um algoritmo de pesquisa. De seguida, podemos ver de uma forma muito empírica o ajuste para uma função de erro ( $C$ ) ao longo de  $\hat{y}$ , durante várias iterações do treino. O algoritmo vai ajustando os pesos até que  $C$  atinja o valor mínimo (onde converge). Neste caso, a pesquisa é baseada no cálculo do declive das retas tangentes nos pontos obtidos (ângulos a verde), e que permitem “orientar” a pesquisa de “salto” em “salto” (bolas amarelas). Um dos parâmetros associados para o cálculo dos “saltos” é a fator de aprendizagem (*learning rate*), que pode levar a que a aprendizagem seja lenta, mas precisa, ou rápida, mas menos consistente.



Aqui também se observa que existem dois mínimos locais, sendo, no entanto, um deles claramente melhor ( $C$  inferior). No entanto é possível que o algoritmo convirja o outro mínimo, tal como acontece connosco quando durante uma aprendizagem convergimos (mecanizamos) para um comportamento que não sendo absolutamente errado não é o mais correto/eficaz e com isso ganhamos com um “vício” (e que depois custa a reverter porque a aprendizagem fica persistida).

Quando aplicado a todo *dataset* este tipo de algoritmos é chamado de *batch gradient descent*, quando aplicado linha a linha *stochastic gradient descent*.

O processo de ajuste dos pesos na rede no fator em que estes são responsáveis pelo erro, chama-se *backpropagation*.

### Consolidando o algoritmo

Assim, os passos necessários para criar e treinar uma RNA, são:

1. Carregar e dividir o *dataset* entre treino e teste. O primeiro irá servir para ajustar os pesos através do processo de treino, o segundo para testar o resultado final da rede.
2. Reduzir as variáveis de entrada todas à mesma escala para evitar que alguns se tornem dominantes no cálculo da função de ativação por terem valores absolutos superiores (*feature scaling*). As opções mais usuais.

a. *standardizar*: 
$$x' = \frac{x - \bar{x}}{\sigma}$$
 diferença entre o valor e a média sobre o desvio padrão

b. *normalizar*: 
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$
 diferença entre o valor e o mínimo sobre a diferença entre máximo e mínimo

3. Inicializar a rede, com pesos aleatórios perto de 0.
4. Introduzir uma linha de dados (observação) na rede.

5. Propagar os valores ativando os neurónios até se obter o valor previsto ( $y$ ). O impacto de cada valor é limitado pelo peso correspondente.
6. Comparar o valor obtido ( $y$ ) com o valor real ( $\hat{y}$ ) e calcular o custo ( $C$ ) (erro).
7. Propagar o erro para a rede para que os pesos ( $W_i$ ) sejam atualizados após cada observação, ou após um conjunto de observações de acordo com o quanto cada um dos destes é responsável pelo erro (*backpropagation*).
8. Repetir os passos 4 a 7 para todas as observações
9. Quando todas as linhas de dados passarem pelo processo de treino concluiu-se uma sessão (*epoch*).

### Implementação

O desenvolvimento será feito em Python 3.6, utilizando um conjunto de bibliotecas que já implementam a grande maioria do algoritmo explicado anteriormente, simplificando muito o trabalho. Para o efeito instalei o Anaconda para o Python e gestão dos pacotes e utilizo o editor Spyder, num ambiente Windows 10 (também disponíveis em Mac e Linux).

Para instalar as bibliotecas utilizadas identificadas de seguida é necessário executar os seguintes comandos, após a instalação do Anaconda:

```
pip install tensorflow
pip install keras
conda install theano
```

E passando diretamente para o código:

```
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
```

Trabalhei o ficheiro original dos censos para retirar as colunas não utilizadas.



Para o treino, vamos carregar 3000 registos. As colunas de 0 a 9 (atenção que 0 : 10 significa de 0 a 9) serão as variáveis in-

dependentes (a analisar pela rede), e a coluna 10 o valor pretendido, neste exemplo muito simples, um binário relativo a se a subsecção dispõe de mais homens (0, azul) ou mulheres (1, rosa). Vamos dividir os dados entre treino (80%) e teste (20%). A preto os valores em falta que irão ser alvo de previsão após o treino.

```
ds = read_csv('censos2011-lisboa.csv')
X = ds.iloc[0 : 3000, 0 : 10].values.astype(float)
Y = ds.iloc[0 : 3000, 10].values.astype(float)
X1 = ds.iloc[3000:, 0 : 10].values.astype(float)

trainX, testX, trainY, testY = train_test_split(
    X, Y, test_size = 0.2, random_state = 0)
```

De seguida é necessário *standartizar* os dados de treino e teste para a mesma escala. Em alternativa poderíamos utilizar a classe *MinMaxScaler* para normalizar. Caso colunas de dados com categorias (por exemplo nível socio económico em classe baixa, média, alta) e não valores numéricos teríamos de aplicar uma transformação utilizando a classe *OneHotEncoder*.

```
scaler = StandardScaler()
trainX = scaler.fit_transform(trainX)
testX = scaler.transform(testX)
```

A RNA vai ser criada utilizando duas *hidden layers* (camadas) densas (onde cada neurónio de uma camada está ligado a todos os neurónios da camada seguinte), com 4 neurónios cada uma alimentadas por uma *input layer* de 10 variáveis (podíamos utilizar  $\text{len}(X[0])$ ). A função de ativação para estas camadas é a *rectifier* (*relu: rectifier linear unit*). A *output layer* é composta por um único neurónio e uma vez que vamos calcular um valor binário vamos utilizar uma *sigmoid* e que posteriormente aplicamos uma probabilidade. Se o resultado final fosse um domínio de valores (por exemplo habilitações literárias dominantes), esta última *layer* teria tantos neurónios quantos os valores possíveis e seria ativada por uma função *softmax*.

Para aplicar o *gradient descent* vamos utilizar o algoritmo *rmsprop* (*Root Mean Square Propagation*), em alternativa a outros como o *adam*, e a função *binary cross entropy* (*logarithmic loss*) como função de custo (*loss*). Caso tivéssemos um domínio de valores esta função seria *categorical cross entropy*.

```
rna = Sequential()
rna.add(Dense(units = 4,
              kernel_initializer = 'uniform',
              activation = 'relu',
              input_dim = 10))
rna.add(Dense(units = 4,
              kernel_initializer = 'uniform', activation = 'relu'))
rna.add(Dense(units = 1,
              kernel_initializer = 'uniform', activation = 'sigmoid'))
rna.compile(optimizer = 'rmsprop', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

# A PROGRAMAR

## DEEP LEARNING PASSO A PASSO

Finalmente vamos executar a rede, utilizando 32 sessões de treino (por omissão um *batch* de 32 linhas).

```
rna.fit(trainX, trainY, epochs = 32)
```

A precisão estimada para o modelo atingiu valores na ordem dos 81%, o que já induz um padrão.

A partir daqui podemos melhorar a precisão recorrendo a um conjunto de técnicas como sendo alteração de alguns dos parâmetros da rede (*hyperparameters*). No exemplo observa-se que a precisão (*accuracy*) teve um aumento na última *epoch* pelo que é possível que se tivesse treinado mais conseguisse um resultado melhor (ainda não terá atingido o pico). Podemos também aumentar o número de neurónios (4 + 4 é uma rede pequena), testar outras funções de ativação e de *gradient descent*, variar o tamanho do *batch*, etc.

Adicionalmente podemos utilizar de mais variáveis disponíveis (desde que sejam independentes: podia usar outras relativas a tipos de famílias, núcleos familiares e até mesmo alojamentos), ou mesmo ter em consideração a tendência das zonas geográficas vizinhas uma vez que populações geograficamente próximas terão tendência a ser mais semelhantes entre si. Para calcular as zonas vizinhas será necessário recorrer à informação espacial das subsecções algo que o INE também disponibiliza (e daí ter projetado a informação em mapa).

Este tipo de afinações é agora o trabalho de quem utiliza este tipo de modelos.

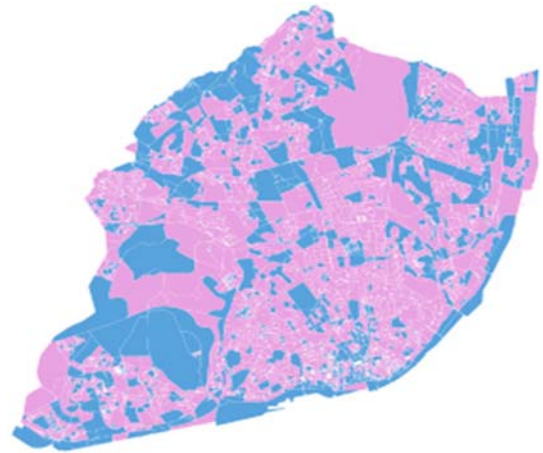
Outro tema relevante é quando a nossa rede fica “viciada” num determinado *dataset* de treino e para o efeito utilizam-se técnicas como *dropout regularization* onde aleatoriamente alguns neurónios da rede são temporariamente desativados durante o treino.

```
Epoch 28/32
2400/2400 [=====] - 0s
          173us/step - loss: 0.4649 - acc: 0.8096
Epoch 29/32
2400/2400 [=====] - 0s
          154us/step - loss: 0.4640 - acc: 0.8092
Epoch 30/32
2400/2400 [=====] - 0s
          125us/step - loss: 0.4645 - acc: 0.8058
Epoch 31/32
2400/2400 [=====] - 0s
          150us/step - loss: 0.4638 - acc: 0.8054
Epoch 32/32
2400/2400 [=====] - 0s
          185us/step - loss: 0.4635 - acc: 0.8108
```

Por fim, podemos utilizar a rede já treinada para prever os valores das restantes linhas. Uma vez que a função de ati-

vação da *output layer* é uma *sigmoid*, como resultado vamos obter um valor entre 0 e 1. A esse valor vamos assumir 1 se o valor for superior a 0.5 e 0 caso contrário. No final, o mapa completo.

```
X1 = scaler.transform(X1)
predY = (rna.predict(X1) > 0.5).astype(int)
```



### Fontes

Código: <https://github.com/sergiosaraiva/artificial-intelligence/tree/master/pap-rna>

Dados: <http://mapas.ine.pt/download/index2011.phtml>

### Conclusão

Temas relacionados com a Inteligência Artificial e respetivas subcategorias tem despertado muito interesse nos últimos tempos. Adicionalmente, com as bibliotecas disponíveis hoje em dia é bastante rápido implementar uma RNA simples como exemplificado (com 2 *hidden layers* de 4 neurónios artificiais cada) para efetuar previsões, no caso um *dataset* dos censos populacionais. A partir deste exemplo é possível evoluir para modelos mais sofisticados, mas para já, chegar às capacidades dos cerca de 80 mil milhões de neurónios do cérebro humano (e muito mais complexos) é ainda uma viagem muito longa.

## AUTOR



Escrito por Sérgio Saraiva

Licenciado em Engenharia Informática e de Computadores pelo Instituto Superior Técnico.

Tenho especial interesse em resolver desafios complexos através da criação de tecnologia que agregue diferentes áreas de conhecimento. <https://www.linkedin.com/in/sergio-saraiva/>

## INTRODUÇÃO AO SASS

### 1. INTRODUÇÃO

O desenvolvimento para a Web está a mudar! Hoje em dia, é incontornável que um programador Web frontend não domine apenas a tríade HTML+CSS+JavaScript, mas que tenha um conhecimento (nem que seja básico) do que são pré-processadores, gestor de dependências, ferramentas de automatização de tarefas de *building* e de geração de estrutura de código, transpiladores, compiladores, minificadores, etc. Na verdade, estas ferramentas começam, cada vez mais, a fazer parte do *workflow* de desenvolvimento das nossas aplicações Web. Mesmo não sendo uma exigência para o desenvolvimento, estas ferramentas podem economizar muito o tempo de um programador.

Quando uma página Web começa a ficar mais complexa, é frequente vermos vários ficheiros CSS com inúmeras regras e com um nível razoável de redundância. Uma forma de economizar tempo, e de manter todas essas regras de uma forma mais flexível, é através do uso de **pré-processadores de CSS**. Estas ferramentas usam recursos até aqui indisponíveis no contexto da criação de folhas de estilo. Desta forma, o código torna-se mais organizado, permitindo que os programadores trabalhem mais rapidamente e cometam menos erros.

Um dos preprocessadores CSS mais populares é o **Sass** (*Syntactically Awesome StyleSheets*), atualmente na sua versão 3.5.5. As próximas secções mostram como usá-lo e as suas principais características.

### 2. INSTALAÇÃO

Os ficheiros Sass não podem ser interpretados pelo browser. Sendo assim, é necessário compilar os ficheiros para CSS standard para que possam ser usados na Web. Existem várias formas de usar Sass desde instalá-lo no seu computador até usar ambientes online.

#### LINHA DE COMANDO

Pode instalar o Sass através da linha de comando. A implementação oficial do Sass é *open-source* e foi escrita na linguagem de programação Ruby. Logo, comece por instalar o Ruby. Depois, abra a linha de comandos e escreva o seguinte:

```
gem install sass
```

O Ruby usa o *gem* para gerir os vários pacotes de código, como o Sass. Este comando irá instalar o Sass e todas as suas dependências no seu computador. Depois, para validar a instalação do Sass escreva o seguinte comando:

```
sass -v
```

O comando anterior deverá retornar:

```
Sass 3.5.5 (Bleeding Edge)
```

Assim que o Sass estiver instalado, pode começar a criar os seus ficheiros Sass (extensão **.scss**) e a compilá-los para folhas de estilo CSS. O comando de compilação é o seguinte:

```
sass input.scss output.css
```

Pode também ativar a inspeção automática a ficheiros individuais ou pastas inteiras, para que quando um ficheiro Sass for atualizado, o correspondente ficheiro CSS é gerado.

Esta é uma funcionalidade poderosa e pode ser executada na linha de comandos da seguinte forma:

```
// Ficheiro individual
sass --watch input.scss:output.css
// Pasta
sass --watch path/sass-directory[:path/
                                     css-directory]
```

O comando anterior vai inspecionar todos os ficheiros **.scss** na pasta **path/sass-directory** e quando um destes ficheiros se alterar, o Sass vai atualizar os ficheiros CSS correspondentes ou gerar novos, caso não existam. Pode também, na parte final do comando, especificar a pasta destino para os ficheiros CSS.

### OUTRAS OPÇÕES PARA USAR SASS

Se não for adepto da linha de comandos, pode usar aplicações com interfaces gráficas que permitem, da mesma forma, inspecionar e gerar folhas de estilo CSS a partir de ficheiros Sass (ex.: CodeKit, Hammer, Koala, Compass, Scout, etc.).

Outra forma é usar o **LibSass** que é um compilador Sass escrito em C++. O LibSass é apenas uma biblioteca de código e não contém nenhum binário. Sendo assim, precisa de um implementador (ou *wrapper*) para compilar ficheiros Sass através do LibSass. De entre os *wrappers* mais populares destacam-se o **node-sass** e o **sass.js**. A primeira é uma biblioteca que implementa o *libsass* para Node.js. A segunda traz o *libsass* para a web através de uma implementação em JavaScript.

Por fim, a solução menos intrusiva é usar uma *sandbox online* para escrever e compilar Sass chamada **SassMeister**. A grande vantagem é poder criar e testar o seu código Sass, sem ter a necessidade de instalar qualquer *software* no seu computador.

As próximas secções falam das principais características do Sass, nomeadamente, as variáveis, os aninhamentos de regras, a herança, os *mixins*, as funções, as estruturas condicionais e cíclicas e os formatos de output suportados.



# A PROGRAMAR

## INTRODUÇÃO AO SASS

### 3. PRINCIPAIS CARACTERÍSTICAS

#### VARIÁVEIS

As variáveis em Sass permitem o armazenamento de valores que vão ocorrer com frequência, como cores, dados sobre uma fonte, entre outros. O Sass usa o símbolo \$ para preceder o nome de uma variável e o símbolo de : como operador de atribuição. O próximo excerto de código mostra um exemplo básico do uso de variáveis. A coluna da esquerda mostra o código Sass. A coluna da direita mostra o código CSS (após compilação):

```
/* Sass */
$my-color: #eeffcc;

table {
  background: $my-color;
}

div {
  background: $my-color;
}
```

```
/* CSS compilado */
table {
  background: #eeffcc;
}

div {
  background: #eeffcc;
}
```

Com o uso de variáveis podemos reutilizar os mesmos valores, ou, se é necessária uma mudança, podemos fornecer o novo valor num único local (a definição da variável), em vez de o aplicar manualmente em todos os lugares. Neste exemplo, se quiséssemos alterar a cor de ambas as regras, teríamos apenas de fazer a alteração num único sítio, mais concretamente, na definição da variável \$my-color.

#### REGRAS ANINHADAS

Usando Sass, podemos combinar várias regras CSS criando seletores compostos. O aninhamento básico refere-se à capacidade de ter uma declaração dentro de outra.

Um exemplo clássico, é quando temos de estilizar uma lista em HTML composta por um conjunto de *links*. Neste caso temos uma estrutura com vários níveis de profundidade (<ul>, <li>, <a>) e que requer algum cuidado na altura de estilizá-la através de regras CSS. Com o Sass temos uma forma mais intuitiva e compacta de estilizar este tipo de cenários através de declarações aninhadas. O próximo exemplo é ilustrativo desta situação:

```
/* Sass */
ul {
  list-style: none;

  li {
    padding: 15px;
    display: inline-block;

    a {
```

```
      text-decoration: none;
      font-size: 16px;
      color: #444;
    }
  }
}
```

```
/* CSS compilado*/
ul {
  list-style: none;

  ul li {
    padding: 15px;
    display: inline-block;

    ul li a {
      text-decoration: none;
      font-size: 16px;
      color: #444;
    }
  }
}
```

A diretiva @at-root faz com que uma ou mais regras sejam lançadas na raiz do documento, em vez de serem aninhadas por baixo dos seus seletores-pai. No caso de pretendemos referenciar o “pai” em declarações aninhadas, deve-se usar o símbolo &. O próximo exemplo mostra como adicionar pseudoseletores usando esta técnica:

```
/* Sass */
a.myAnchor {
  color: blue;

  &:hover {
    text-decoration: underline;
  }

  &:visited {
    color: purple;
  }
}
```

```
/* CSS compilado */
a.myAnchor {
  color: blue;

  a.myAnchor:hover {
    text-decoration: underline;
  }

  a.myAnchor:visited {
    color: purple;
  }
}
```

#### HERANÇA

O Sass suporta herança, ou seja, herdar as propriedades CSS de um seletor para outro. Para tal, usa-se a diretiva @extend. A herança deve ser usada quando precisamos de elementos de estilo similares a outros já definidos, mas que requerem ligeiras alterações. Um exemplo clássico, é a definição de dois botões: um de confirmação e outro de cancelamento:

```
/* Sass */
.dialog-button {
  box-sizing: border-box;
```



# A PROGRAMAR

## INTRODUÇÃO AO SASS

```
color: #ffffff;
box-shadow: 0 1px 1px 0;
}

.confirm {
  @extend .dialog-button;
  background-color: red;
  float: left;
}

.cancel {
  @extend .dialog-button;
  background-color: blue;
  float: right;
}
```

```
/* CSS compilado */
.dialog-button, .confirm, .cancel {
  box-sizing: border-box;
  color: #ffffff;
  box-shadow: 0 1px 1px 0;
}

.confirm {
  background-color: red;
  float: left;
}

.cancel {
  background-color: blue;
  float: right;
}
```

Repare que o Sass combinou os seletores em vez de repetir as mesmas declarações sistematicamente.

### MIXINS

Os *mixins* podem ser vistos como uma versão simplificada de construtores em linguagens de programação orientadas a objetos. Na realidade, pode incluir estilos da mesma forma que o `@extend` faz mas com a capacidade de fornecer argumentos, tornando assim uma preciosa ferramenta para a reutilização dinâmica de código e a quebra de redundância.

A diretiva `@mixin` é usada para criar *mixins*. Para invocar o *mixin* usa-se a diretiva `@include`. No próximo exemplo, é definido um *mixin* que vai renderizar quadrados com cores e tamanhos passados como argumentos:

```
/* Sass */
@mixin square($size, $color) {
  width: $size;
  height: $size;
  background-color: $color;
}

.small-blue-square {
  @include square(20px, rgb(0,0,255));
}

.big-red-square {
  @include square(300px, rgb(255,0,0));
}
```

```
/* CSS compilado*/
.small-blue-square {
  width: 20px;
```

```
height: 20px;
background-color: blue;
}

.big-red-square {
  width: 300px;
  height: 300px;
  background-color: red;
}
```

### FUNÇÕES

O Sass suporta uma longa lista de funções predefinidas. Eles servem todos os tipos de propósitos, incluindo a manipulação de strings, operações relacionadas com cores e alguns métodos de matemática úteis como o `random` e o `round`. Para uma lista completa das funções internas do Sass, por favor consulte a documentação do Sass.

O próximo exemplo usa a função `darken` que escurece uma determinada cor mediante uma determinada percentagem:

```
/* Sass */
$my-blue: #2196F3;

a {
  padding: 10px 15px;
  background-color: $my-blue;
}

a:hover {
  background-color: darken($my-blue,10%);
}
```

```
/* CSS compilado */
a {
  padding: 10px 15px;
  background-color: #2196F3;
}

a:hover {
  background-color: #0c7cd5;
}
```

### ESTRUTURAS DE DECISÃO E CÍCLICAS

O Sass suporta estruturas de decisão e de repetição. Para esse efeito pode usar diretivas como o `@if`, `@for`, `@each` e `@while`. Os dois próximos exemplos mostram como usar estas diretivas em dois contextos diferentes: tomada de decisão mediante um determinado valor e iteração sob variáveis para criação de declarações similares.

No primeiro exemplo, inspeciona-se o valor de uma variável e executa-se um conjunto de ações mediante o seu valor. Repare que caso a condição na diretiva `@if` for avaliada como falsa, é executado o conjunto de expressões que sucedem a diretiva `@else`.

```
/* Sass */
$boolean: true !default;

@mixin foo() {
  @if $boolean {
    display: block;
  }
}
```

# A PROGRAMAR

## INTRODUÇÃO AO SASS

```
    }
    @else {
      display: none;
    }
  }

.some-selector {
  @include foo();
}
```

```
/* CSS compilado */
.some-selector {
  display: block;
}
```

O próximo exemplo usa a diretiva `@for` para definir várias declarações de estilo nomeadas por interpolação:

```
/* Sass */
$squareCount: 3;

@for $i from 1 through $squareCount {
  #square-#{$i} {
    background-color: red;
    width: 50px * $i;
    height: 120px / $i;
  }
}
```

```
/* CSS compilado */
#square-1 {
  background-color: red;
  width: 50px;
  height: 120px;
}

#square-2 {
  background-color: red;
  width: 100px;
  height: 60px;
}

#square-3 {
  background-color: red;
  width: 150px;
  height: 40px;
}
```

## FORMATOS DE OUTPUT

De forma a ajustar o formato de saída das declarações CSS geradas, o Sass suporta quatro formatos de saída diferentes: aninhado, expandido, compacto e comprimido. Para perceber todos estes formatos, considere o seguinte excerto de código:

```
.a {
  text-align: right;
  &:visited {
    color: #222222;
    color: rgba(34, 34, 34, 0.77);
  }
}

&:hover {
  color: #B00909;
}
```

Em seguida apresentam-se as várias CSS geradas de acordo com os quatro formatos de saída suportados pelo Sass:

```
/* Expandido (:expanded) */
.a {
  text-align: right;
}

.a:visited {
  color: #222222;
  color: rgba(34, 34, 34,0.77);
}

.a:hover {
  color: #B00909;
}
```

```
/* Aninhado (:nested) */
.a {
  text-align: right;
}

.a:visited {
  color: #222222;
  color: rgba(34, 34, 34, 0.77);
}

.a:hover {
  color: #B00909;
}
```

```
/* Compacto (:compact) */
.a {text-align: right;}
.a:visited { color: #222222; color: rgba
(34,34,34,0.77);}
.a:hover { color: #B00909;}
```

```
/* Comprimido (:compress) */
.a { text-align: right}.a:visited {color:
#222222;
color: rgba(34,34,34,0.77)}.a:hover {color:
#B00909}
```

A forma de definir o formato desejado depende do que está a usar para programar em Sass. Se estiver a usar uma aplicação baseada em GUI, como o **Koala**, terá de seleccionar a opção do formato pretendido através da própria interface gráfica. Se estiver a usar o ambiente online **SassMeister** tem de seleccionar a opção do menu **Options à CSS Output**. Se estiver a usar a linha de comando, precisa apenas de usar a flag `--style` com o formato pretendido:

```
sass --watch stl.scss:stl.css --style
[expanded|nested|compact|compressed]
```

## CONCLUSÃO

Os preprocessadores são uma ferramenta essencial para quem trabalha no *frontend* com grande dependência de folhas de estilo. Como viram, os preprocessadores vão permitir uma gestão de estilos mais flexível, quebrar a redundância e simplificar a manutenção de grandes quantidades de regras de estilização.

O facto de ter usado o Sass como pré-processador

nos exemplos não significa que têm de usar este pré-processador. Apesar de ser o mais popular e de ter ganho um grande alento com o suporte do Bootstrap 4 (esta framework Web disponibiliza o código fonte de todos os seus estilos em Sass), eu aconselho que usem (e testem) a maior quantidade de pré-processadores possível, pois só assim poderão escolher um de forma sustentada. Na maior parte das vezes, vão escolher aquele cuja sintaxe se sentem mais familiarizados ou cuja funcionalidade que precisam é bem suportada.

**“ (...) é incontornável que um programador Web frontend não domine apenas a tríade HTML+CSS+JavaScript, mas que tenha um conhecimento (...) do que são preprocessadores, gestor de dependências, ferramentas de automatização de tarefas de building e de geração de estrutura de código, transpiladores, compiladores, minificadores(...) ”**

Num futuro artigo, partilharei com todos uma forma prática de usar Sass, num contexto mais alargado, nomeadamente, a personalização do Bootstrap 4 com NPM, Sass e Gulp.

**NOTA:** Este artigo é uma adaptação resumida de uma secção do meu livro “Criação Rápida de Sites Responsivos com o Bootstrap”, publicado em 2017 pela editora FCA.

<sup>1</sup> <http://rubyinstaller.org>.

<sup>2</sup> <https://www.sassmeister.com/>

<sup>3</sup> <http://sass-lang.com/documentation/Sass/Script/Functions.html>.

**“ (...) estas ferramentas começam, cada vez mais, a fazer parte do workflow de desenvolvimento das nossas aplicações Web. Mesmo não sendo uma exigência para o desenvolvimento, estas ferramentas podem economizar muito o tempo de um programador. ”**

## AUTOR



Escrito por Ricardo Queirós

Natural do Porto, sou acima de tudo, um apaixonado pelo desenvolvimento quer seja na criação de aplicações ou serviços para a Web, soluções móveis nativas ou jogos. Gosto de programar em JavaScript, Java e, mais recentemente, Kotlin.

Sou doutorado em Ciência de Computadores pela Faculdade de Ciências da Universidade do Porto (FCUP) e exerço a minha atividade de docente na Escola Superior de Media Artes e Design (ESMAD) do P.PORTO e a minha atividade de investigação no Center for Research in Advanced Computing Systems (CRACS) uma Unidade do INESC TEC.

# A PROGRAMAR

## C# - Excel

Ainda é muito usual depararmos-mos com ficheiros de processamento Excel, ou não fosse o Excel a mais famosa folha de cálculo até aos dias de hoje. Neste artigo, procuramos de uma forma simples mostrar ao caro leitor como podemos tirar partido do processamento do Excel, fazendo um pequeno programa que nos permite facilmente processar e criar novos ficheiros resultado partindo do Excel.

Para este artigo proponho algo simples, imaginemos uma empresa que produz vários tipos de brindes e merchandising. Os comerciais ao longo do mês vão criando encomendas dos vários clientes. A nossa empresa trabalha com dois armazéns distintos, mas o responsável pelo setor do armazém ao longo do mês recebe esses mesmos pedidos e regista os pedidos numa folha de Excel para lhes dar despacho. No final do período de processamento, o departamento financeiro precisa de saber ao certo qual o número de encomendas efetuadas para poder emitir a fatura ao cliente final. Para este exemplo vamos considerar que vendemos três tipos de canetas. Azuis, pretas e vermelhas. Consideramos também que temos três clientes (somos uma pequena empresa ☺) e que como é normal para cada cliente acordamos um valor específico de cada cor de caneta, sendo esse valor diferente para cada cliente.

Então recapitulando... recebemos do nosso responsável de armazém um ficheiro “desordenado” com as encomendas e precisamos de gerar um ficheiro Excel com os autos de facturação (documentos que posteriormente serão utilizados internamente para que possa ser emitida a factura) para o nosso departamento financeiro não perder muito tempo a emitir faturas aos clientes.

Olhemos agora para o ficheiro de entrada...

Auto de Facturação					
1	2	3	4	5	6
Nome Cliente	Morada	Armazém	Código de Contrato:	NIF:	Cond. Pag.
Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias
Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias

DADOS PARA AUTO DE FACTURAÇÃO						DADOS PARA CANCELAMENTO					
Cliente	Contrato	Nome de Produto	Nº Encomenda	Data	Quant.	Designação Cliente	Morada	Centro de Encomenda	Contrato	NIF	Cond. Pagamento
Cliente Um	Contrato N1A	Canetas Pretas	3013085	02/02/18	900	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Um	Contrato N1A	Canetas Vermelhas	3013051	03/02/18	25	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2821218	09/01/18	990	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Dois	Contrato 2B	Canetas Azuis	2821979	10/01/18	620	Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias
Cliente Dois	Contrato 2B	Canetas Azuis	2814047	16/01/18	5820	Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias
Cliente Dois	Contrato 2B	Canetas Azuis	2814051	23/01/18	4180	Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias
Cliente Três	Contrato C3	Canetas Pretas	2823608	19/01/18	600	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Três	Contrato C3	Canetas Pretas	1517773	27/01/18	710	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2829673	31/01/18	1000	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2814154	31/01/18	6880	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2811336	06/02/18	5720	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2811610	13/02/18	6400	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2819137	20/02/18	6400	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Três	Contrato C3	Canetas Azuis	2819293	27/02/18	6340	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Três	Contrato C3	Canetas Azuis	2823283	23/02/18	2180	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Três	Contrato C3	Canetas Azuis	2819854	06/02/18	5760	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Três	Contrato C3	Canetas Azuis	2819117	13/02/18	5740	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Três	Contrato C3	Canetas Azuis	2819587	20/02/18	6000	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Três	Contrato C3	Canetas Azuis	2819587	24/02/18	5720	Cliente Três	Rua Matriz 256	Armazém B	Contrato C3	789 123 456	30 Dias
Cliente Dois	Contrato 2B	Canetas Azuis	2812050	28/02/18	6880	Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias
Cliente Dois	Contrato 2B	Canetas Azuis	2818911	13/02/18	3800	Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2810255	20/02/18	6320	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Um	Contrato N1A	Canetas Azuis	2819154	20/02/18	3120	Cliente Um	Rua da Programação	Armazém A	Contrato N1A	123 456 789	30 Dias
Cliente Dois	Contrato 2B	Canetas Azuis	2725388	17/02/18	220	Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias
Cliente Dois	Contrato 2B	Canetas Azuis	2812055	03/02/18	5200	Cliente Dois	Rua da Tabela 123	Armazém A	Contrato 2B	456 789 123	60 Dias

	A	B	C	D	E
1	<b> Tipo de Produtos</b>	<b> Preço unitario</b>	<b> N° Contrato</b>		
2	Canetas Pretas	1	Contrato N1A		
3	Canetas Azuis	1,5	Contrato N1A		
4	Canetas Vermelhas	1	Contrato N1A		
5	Canetas Pretas	0,5	Contrato 2B		
6	Canetas Azuis	1	Contrato 2B		
7	Canetas Vermelhas	1,25	Contrato 2B		
8	Canetas Pretas	1	Contrato C3		
9	Canetas Azuis	0,6	Contrato C3		
10					
	lista_Clientes	Lista a Exportar	<b>Preço Produtos</b>	Tipos Produtos	

	A	B	C	D	E	F
1						
2	<b> Tipo de Produtos</b>					
3	Canetas Pretas					
4	Canetas Azuis					
5	Canetas Vermelhas					
6						
	lista_Clientes	Lista a Exportar	Preço Produtos	<b>Tipos Produtos</b>		

Temos 4 separadores a considerar:

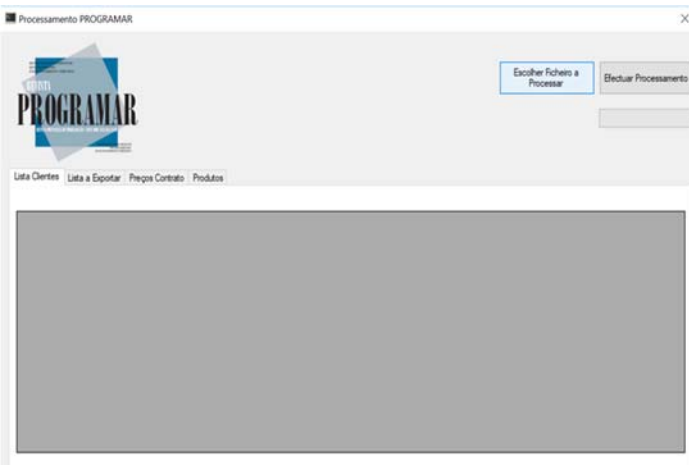
- **Lista\_Clientes** – onde estão os dados dos nossos clientes
- **Lista a Exportar** – onde estão as encomendas efetuadas ao longo do período de faturação e onde também são reproduzidos os dados de faturação dos nossos clientes
- **Preço Produtos** – onde está o preço acordado por cada contrato
- **Tipo Produtos** – onde encontramos o tipo de produtos que fornecemos

Quero chamar a atenção de que poderíamos ter simplificado mais o ficheiro de entrada, contudo para que possamos exemplificar facilmente o passo a passo deste artigo, optamos por deixar o mesmo desta forma.

Como o Windows é o sistema operativo que mais encontramos nas empresas portuguesas, optamos por criar uma pequena aplicação em Windows Forms, porque afinal... ainda é prático, simples e eficaz.

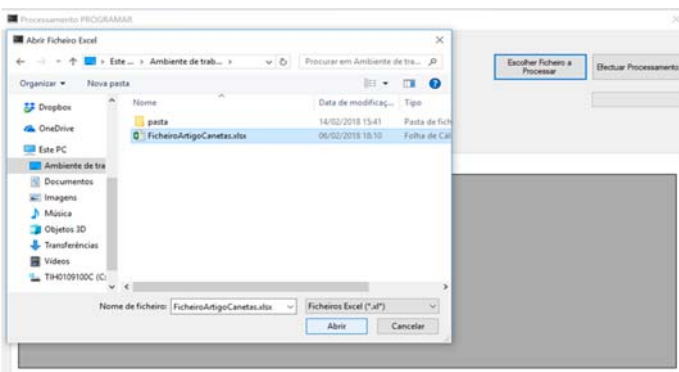
Então criado o novo projecto no Visual Studio (para este artigo foi utilizado o Visual Studio 2017), usando o modo gráfico rapidamente colocamos no projecto um TabControl com quatro TabPages. Afinal, nós programadores, queremos sempre ser user-friendly o mais possível, para que os nossos utilizadores possam verificar os dados de entrada do ficheiro, se estão de acordo com os mesmos antes de entrarmos em modo processamento. Acrescentamos dois botões, um para escolha do ficheiro a processar e outro para verificar o pro-

cessamento.



Comecemos então a programar o botão “Escolher Ficheiro a Processar”... Precisamos de adicionar ao nosso projecto um objecto do tipo OpenFileDialog, que nos permitirá utilizar a caixa de diálogo de abertura de ficheiro para “navegar” no sistema de ficheiros e apontar para o ficheiro que se pretende abrir.

Nota: quero chamar a atenção do leitor que inicialmente deve chamar a biblioteca correspondente: using System.Data.OleDb; uma vez que será necessário para a leitura dos ficheiros em formato excel e o seu tratamento.



Vamos também acrescentar às variáveis quatro objectos do tipo DataTable (um por cada separador do ficheiro INPUT) para mais facilmente ordenarmos os dados obtidos

```
System.Data.DataTable dtEND = new System.Data.DataTable();
System.Data.DataTable dtENDProd = new System.Data.DataTable();
System.Data.DataTable dtENDCabecalhos = new System.Data.DataTable();
System.Data.DataTable dtENDProdSimples = new System.Data.DataTable();
```

Acrescentamos também quatro datagridviews para que o utilizador possa ver e alterar os dados a processar:

- DataGridViewListaClientes
- dataGridViewListaExportar

- dataGridViewPrecos
- dataGridViewTiposProdutos

Para que o menu seguinte nos apareça basta colocarmos as seguintes instruções dentro do código do botão:

```
string filename = null;
#region abrirFicheiro
{
    openFileDialogExcel.Title = "Abrir Ficheiro Excel";
    openFileDialogExcel.InitialDirectory = getpath(); //para guardar o //caminho onde temos o ficheiro INPUT guardado
    openFileDialogExcel.Filter = "todos os ficheiros (*.*)|*.xl|Ficheiros Excel (*.xl)|*.xl*"; //para mostrar apenas os // ficheiros Excel
    openFileDialogExcel.FilterIndex = 2;
    openFileDialogExcel.RestoreDirectory = true;
    if (openFileDialogExcel.ShowDialog() == DialogResult.OK)
    {
        filename = openFileDialogExcel.FileName.ToString();
    }
    else
    { }
}
#endregion
if (filename == null) //caso caminho // do ficheiro seja nulo, o programa retorna essa // informação ao utilizador
{
    MessageBox.Show("Nenhum ficheiro seleccionado!", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    return;
}
else // Quando o caminho do ficheiro é reconhecido, passamos então ao processamento dos dados
{
    //cria um novo dataTable
    System.Data.DataTable dtExcel = new System.Data.DataTable();
    dtExcel.TableName = "lista_Clientes";
    string SourceConstr = @"Provider=Microsoft.ACE.OLEDB.12.0; Data Source=' + filename + ';Extended Properties='Excel 8.0;HDR=Yes;IMEX=1";
    OleDbConnection con = new OleDbConnection(SourceConstr);
    //seleciona os dados da folha de excel com o nome //“lista_clientes”
    string query = "Select * from [lista_Clientes$]";
    OleDbDataAdapter data = new OleDbDataAdapter(query, con);
    //preenche a dataTable
    data.Fill(dtExcel);
    AppGarEmbeddedBDDataSet.Tables.Add(dtExcel.Copy());

    //preenche a DataGridViewListaClientes com os dados lidos do Excel
    DataGridViewListaClientes.
```



# A PROGRAMAR

C# - EXCEL

```
DataSource = dtExcel;

System.Data.DataTable dtExcel1 =
    new System.Data.DataTable();
dtExcel1.TableName =
    "Lista a Exportar";
string SourceConstr1 =
    @"Provider=Microsoft.ACE.OLEDB.12.0;
Data Source='" + filename + "';Extended Properties=
    'Excel 8.0;HDR=Yes;IMEX=1'";
OleDbConnection con1 = new
    OleDbConnection(SourceConstr1);
string query1 = "Select * from
    [Lista a Exportar$]";
OleDbDataAdapter data1 = new
    OleDbDataAdapter(query1, con1);
data1.Fill(dtExcel1);
for (int i = 0; i < 3; i++)
{
    DataRow dr = dtExcel1.Rows[i];
    dr.Delete();
}
dtExcel1.AcceptChanges();
AppGarEmbeddedBDataSet.Tables.Add
    (dtExcel1.Copy());
dataGridViewListaExportar
    .DataSource = dtExcel1;

System.Data.DataTable dtExcel2 =
    new System.Data.DataTable();
dtExcel2.TableName = "Preço
    Produtos";
string SourceConstr2 =
    @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source='"
    + filename + "';Extended Properties=
    'Excel 8.0;HDR=Yes;IMEX=1'";
OleDbConnection con2 =
    new OleDbConnection(SourceConstr2);
string query2 = "Select * from
    [Preço Produtos$]"; //recolhe os
// dados da folha de excel "Preço Produtos"
OleDbDataAdapter data2 =
    new OleDbDataAdapter(query2, con2);
data2.Fill(dtExcel2);
AppGarEmbeddedBDataSet.Tables.Add
    (dtExcel2.Copy());
dataGridViewPrecos.DataSource =
    dtExcel2;

System.Data.DataTable dtExcel3 =
    new System.Data.DataTable();
dtExcel3.TableName =
    "Tipos Produtos";
string SourceConstr23 =
    @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source='"
    + filename + "';Extended Properties=
    'Excel 8.0;HDR=Yes;IMEX=1'";
OleDbConnection con23 =
    new OleDbConnection(SourceConstr23);
string query23 = "Select * from
    [Tipos Produtos$]";
OleDbDataAdapter data23 =
    new OleDbDataAdapter(query23, con23);
data23.Fill(dtExcel3);
AppGarEmbeddedBDataSet.Tables.Add
    (dtExcel3.Copy());
dataGridViewTiposProdutos.DataSource
    = dtExcel3;
```

É importante referir que apenas repetimos as instruções anteriores para que o processamento das quatro folhas de excel seja transparente para o leitor e possamos facilmente apresentá-lo no ecrã nos datagridviews criados para esse efei-

to.

Posto isto vamos criar um novo DataTable pra "juntar" facilmente os dados para prepararmos facilmente o processamento final.

```
//criamos o novo objecto DataTable com os da-
dosResultantes do carregamento do ficheiro
System.Data.DataTable dadosResultantes =
    new System.Data.DataTable();

dadosResultantes.TableName =
    "DadosResultantes";
dadosResultantes.Columns.Add
    ("Cliente", typeof(string));
dadosResultantes.Columns.Add
    ("NomeContrato", typeof(string));
dadosResultantes.Columns.Add
    ("TipoProduto", typeof(string));
dadosResultantes.Columns.Add
    ("Encomenda", typeof(string));
dadosResultantes.Columns.Add
    ("Data", typeof(string));
dadosResultantes.Columns.Add
    ("Quantidade", typeof(string));
dadosResultantes.Columns.Add
    ("Space", typeof(string));
dadosResultantes.Columns.Add
    ("Designacao", typeof(string));
dadosResultantes.Columns.Add
    ("Morada", typeof(string));
dadosResultantes.Columns.Add
    ("Armazem", typeof(string));
dadosResultantes.Columns.Add
    ("Contrato", typeof(string));
dadosResultantes.Columns.Add
    ("NIF", typeof(string));
dadosResultantes.Columns.Add
    ("CodPag", typeof(string));

//Agora vamos guardar o mês de Processamento
apresentado no ficheiro INPUT
mesProcessamento =
    dataGridViewListaExportar.Columns[1].
    HeaderText.ToString();

//para cada linha do DataSet vamos guardar a
// mesma no DataTable final
foreach (DataRow r in
    dtExcel1.Rows)
{
    dadosResultantes.Rows.Add(r[0]
    .ToString(), r[1].ToString(), r[2].ToString(),
    r[3].ToString(), r[4].ToString(),
    r[5].ToString(), r[7].ToString(),
    r[8].ToString(), r[9].ToString(),
    r[10].ToString(), r[11].ToString(),
    r[12].ToString());
}
//copiamos a informação que acabamos de processar
// para o DataTable de dadosResultantes
dtEND = dadosResultantes.Copy();
//ordenamos esse DataTable em relação ao número
// de contrato uma vez que o ficheiro de saída
// necessita que apresentemos os dados em prol
// do contrato de cada cliente
resort(dtEND, "Contrato", "DESC");
//guardamos o tipo de produtos totais que temos
na lista a exportar
System.Data.DataTable tiposresiduosTotais =
    new System.Data.DataTable();

tiposProdutosTotais.TableName =
    "TiposProdutos";
```

# A PROGRAMAR

C# - EXCEL

```
tiposProdutosTotais.Columns.Add
    ("TipoProdutos", typeof(string));
tiposProdutosTotais.Columns.Add
    ("PrecoProdutos", typeof(string));
tiposProdutosTotais.Columns.Add
    ("Contrato", typeof(string));
//guardamos os dados relativos ao cabeçalho dessa
// mesma lista
System.Data.DataTable cabecalhosTotais =
    new System.Data.DataTable();
cabecalhosTotais.TableName =
    "TiposCabecalhos";
cabecalhosTotais.Columns.Add
    ("DesignacaoCliente", typeof(string));
cabecalhosTotais.Columns.Add
    ("Morada", typeof(string));
cabecalhosTotais.Columns.Add
    ("Armazem", typeof(string));
cabecalhosTotais.Columns.Add("NIF",
    typeof(string));
cabecalhosTotais.Columns.Add
    ("CodPagamento", typeof(string));

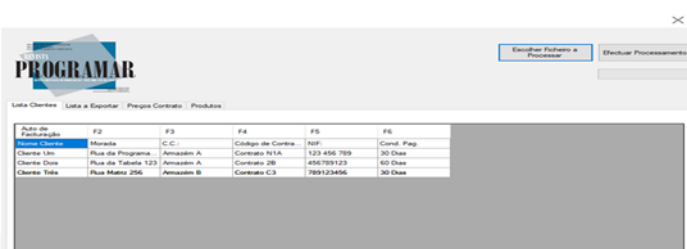
//repetimos o mesmo raciocínio para o tipo de
// produtos
System.Data.DataTable tiposProdutosSimples =
    new System.Data.DataTable();
tiposProdutosSimples.TableName =
    "TiposProdutosSimples";
tiposProdutosSimples.Columns.Add
    ("TipoProdutosSimples", typeof(string));

    foreach (DataRow r in
        dtExcel2.Rows)
    {
        tiposProdutosTotais.Rows.Add(r[0].ToString(),
            r[1].ToString(), r[2].ToString());
    }

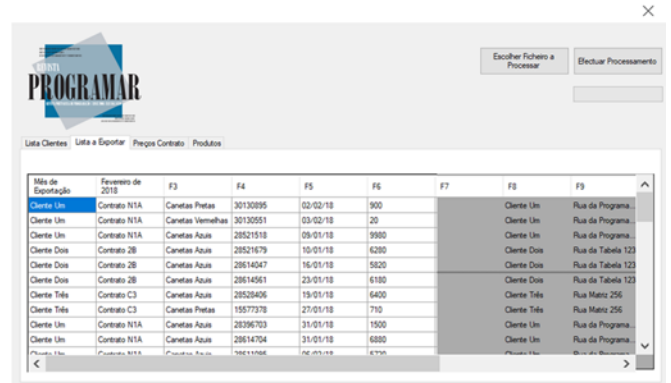
    foreach (DataRow r in
        dtExcel3.Rows)
    {
        tiposProdutosSimples.Rows.Add
            (r[0].ToString());
    }

//guardamos a informação processada anteriormente
// nos DataTables finais apenas com os dados que
// pretendemos utilizar
dtENDProdSimples =
    tiposProdutosSimples.Copy();
dtENDProd =
    tiposProdutosTotais.Copy();
dtENDCabecalhos =
    cabecalhosTotais.Copy();
    }
```

Nesta fase da implementação, já temos então o nosso ficheiro de input carregado e visível no layout da aplicação como mostram as figuras seguintes. Mais uma vez recorde ao leitor que a cada separador representado equivale um separador das folhas de excel do ficheiro a processar.



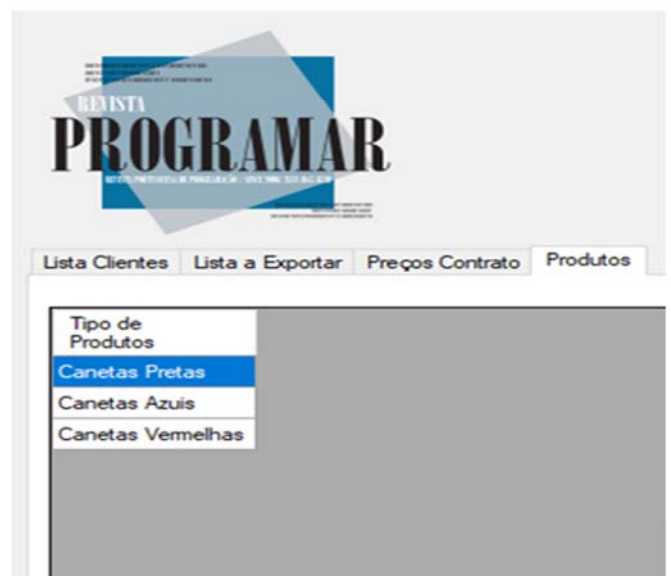
Tipo de Produto	Preço Unitário	Nº Contrato
Canetas Pretas	1	Contrato N1A
Canetas Azuis	1,5	Contrato N1A
Canetas Vermelhas	1	Contrato N1A
Canetas Pretas	0,5	Contrato 2B
Canetas Azuis	1	Contrato 2B
Canetas Vermelhas	1,25	Contrato 2B
Canetas Pretas	1	Contrato C3
Canetas Azuis	0,6	Contrato C3



Mês de Exportação	F3	F4	F5	F6	F7	F8	F9	
Cliente Un	Contrato N1A	Canetas Pretas	30130895	02/02/18	900		Cliente Un	Rua da Program.
Cliente Un	Contrato N1A	Canetas Vermelhas	30130895	02/02/18	20		Cliente Un	Rua da Program.
Cliente Un	Contrato N1A	Canetas Azuis	28521518	09/01/18	9900		Cliente Un	Rua da Program.
Cliente Dos	Contrato 2B	Canetas Azuis	28521518	10/01/18	6200		Cliente Dos	Rua da Tabela 123
Cliente Dos	Contrato 2B	Canetas Azuis	28514047	16/01/18	5820		Cliente Dos	Rua da Tabela 123
Cliente Dos	Contrato 2B	Canetas Azuis	28514561	23/01/18	6180		Cliente Dos	Rua da Tabela 123
Cliente Tels	Contrato C3	Canetas Azuis	25528406	19/01/18	6400		Cliente Tels	Rua Matiz 256
Cliente Tels	Contrato C3	Canetas Pretas	15573738	27/01/18	710		Cliente Tels	Rua Matiz 256
Cliente Un	Contrato N1A	Canetas Azuis	28296703	31/01/18	1500		Cliente Un	Rua da Program.
Cliente Un	Contrato N1A	Canetas Azuis	28514704	31/01/18	6880		Cliente Un	Rua da Program.
Cliente Un	Contrato N1A	Canetas Azuis	18811784	02/01/18	6700		Cliente Un	Rua da Program.



Tipo de Produto	Preço Unitário	Nº Contrato
Canetas Pretas	1	Contrato N1A
Canetas Azuis	1,5	Contrato N1A
Canetas Vermelhas	1	Contrato N1A
Canetas Pretas	0,5	Contrato 2B
Canetas Azuis	1	Contrato 2B
Canetas Vermelhas	1,25	Contrato 2B
Canetas Pretas	1	Contrato C3
Canetas Azuis	0,6	Contrato C3



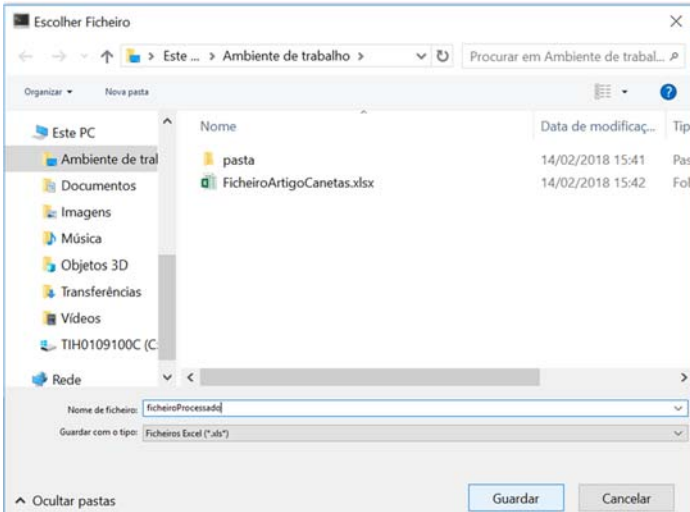
Tipo de Produto	Preço Unitário	Nº Contrato
Canetas Pretas	1	Contrato N1A
Canetas Azuis	1,5	Contrato N1A
Canetas Vermelhas	1	Contrato N1A

# A PROGRAMAR

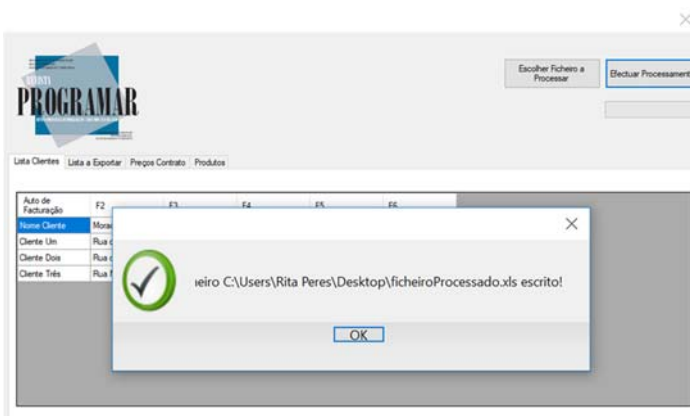
## C# - EXCEL

Neste momento falta-nos apenas processar os dados e devolver ao utilizador o ficheiro de saída já com os dados “arrumados e separados” por contrato.

Mais uma vez o nosso programa deve ser user-friendly e ao carregarmos no botão “Efectuar Processamento”, é aberta uma caixa dialogo que pergunta ao nosso utilizador o nome que quer dar ao ficheiro de saída e onde quer guardar o mesmo.



Seguidamente caso não haja nenhum erro de inserção de dados é apresentada uma mensagem dizendo que o ficheiro foi guardado com sucesso onde o utilizador escolheu.



E ao abrimos o ficheiro resultante podemos verificar que como temos três clientes, com três contratos diferentes, três autos de faturação relativa a cada um deles, está separada por cada separador do ficheiro de output – um separador por contrato. São também apresentados os dados de cada cliente, assim como o numero de encomenda, quantidades e qual o armazém que processou as encomendas desse cliente em questão. Como cada cliente ao logo do período de faturação efetuou várias encomendas, as mesmas são apresentadas por tipo de produto, sendo que no final é apresentado o valor total de pagamento, assim como o IVA e é apresentado o preço final de pagamento.

Auto de Faturação				
Cliente Três		Fevereiro de 2018		
Rua Matriz 256		C.C:	Armazém B	
		NIF:	789123456	
		Cond. Pag.	30 Dias	
Designação				
Canetas Pretas				
Encomenda	DATA	Qtd		
15577378	27/01/18	710		
50793954	28/02/18	7200		
		Qtd	€ Und	€ Total
Sub Total		7910 Und.	1	7910
Designação				
Canetas Azuis				
Encomenda	DATA	Qtd		
28528406	19/01/18	6400		
29018293	27/02/18	6340		
28232583	23/02/18	2180		
29018854	06/02/18	5700		
29019117	13/02/18	5740		
29019587	20/02/18	6000		
29019847	24/02/18	5120		
29293408	05/02/18	5660		
50095314	10/02/18	6000		
50096685	21/02/18	6860		
		Qtd	€ Und	€ Total
Sub Total		56000 Und.	0,6	33600
Designação				
Canetas Vermelhas				
Encomenda	DATA	Qtd		
50130518	01/02/18	134		
		Qtd	€ Und	€ Total
Sub Total		134 Und.	0	0
TOTAL				41 510,00
IVA 23%				9 547,30
Total				51 057,30

Bem, apresentado o resultado final do nosso programa... como podemos chegar facilmente a este ponto?

Vamos por partes... uma vez que já ordenamos o DataTable dadosResultantes, neste momento temos que criar um novo ficheiro Excel, criar uma folha de excel para cada contrato. Dentro de cada uma dessas folhas temos que ter em conta três grandes regiões de apresentação de dados.

- O cabeçalho com os dados do cliente a que se refere o auto de faturação
- A criação de um quadro com dos dados das encomendas para cada tipo de produto (canetas pretas, azuis e vermelhas)
- O rodapé com a apresentação dos totais a pagar pelo cliente.

```
int indicecabecalho = 0;
int indicecablinhas = 0;
string produto = "";
string lineCellValor = "";
string cellValor = "";
string lineCellProdHeader = "";
//variaveis dos estilos
int lineCellProdHeaderIndex = 10;
//variaveis dos estilos
string lineCellSum = "";

//criar novo ficheiro .xls e perguntar ao utilizador onde o quer guardar
string filename = "";
SaveFileDialog filedialogSave =
    new SaveFileDialog();
{
    filedialogSave.Title = "Escolher
```

```

        Ficheiro";
        filedialogSave.InitialDirectory =
            (getpath() + ".xls");
filedialogSave.Filter = "todos os ficheiros (*.*)
|*.*|Ficheiros Excel (*.xls*)|*.xls*";
filedialogSave.FilterIndex = 2;
filedialogSave.RestoreDirectory =
            true;
if (filedialogSave.ShowDialog() ==
        DialogResult.OK)
    {
        filename =
            filedialogSave.FileName.ToString();
    }
    else
    { }
}
//cria uma lista com todos os contratos
List<string> todosContractos = new
        List<string>();
//preenche a lista
foreach (DataRow r in dtEND.Rows)
    {
        todosContractos.Add(r[9].ToString());
        //vê e elimina os repetidos
    }
//remove duplicados
List<string> contractosUniq =
        todosContractos.Distinct().ToList();

var file = new FileInfo
        (filename.ToString() + ".xls");
using (var pack = new ExcelPackage
        (file))
    {
//realiza o styling da folha excel de output
        #region Styles
        #region autofacStyle
var namedStyle =
            pack.Workbook.Styles.CreateNamedStyle
                ("autofacStyle");
            namedStyle.Style.Font.Bold = true;
        #endregion

        #region clientNameStyle
var clientNameStyle =
            pack.Workbook.Styles.CreateNamedStyle
                ("clientNameStyle");
            //coloca o texto branco
            clientNameStyle.Style.Font.Color.SetColor
                (System.Drawing.Color.White);
            clientNameStyle.Style.Font.Bold = true;
        #endregion

```

Terminada esta parte, passamos agora a definir os estilos do ficheiro Excel:

```

#region designacaoStyle
var designacaoStyle =
    pack.Workbook.Styles.CreateNamedStyle
        ("designacaoStyle");
    designacaoStyle.Style.Font.Bold =
        true;
#endregion

#region categoriaStyle
var categoriaStyle =
    pack.Workbook.Styles.CreateNamedStyle
        ("categoriaStyle");
    categoriaStyle.Style.Font.Bold =
        true;
        categoriaStyle.Style.
            Font.Color.SetColor(System.Drawing.Color.Black);

```

```

#endregion

#region generalBold
var generalBold =
    pack.Workbook.Styles.CreateNamedStyle
        ("generalBold");
    categoriaStyle.Style.Font.Bold = true;
#endregion

```

Neste momento estamos pronto para poder percorrer cada linha do DataTable final para apresentarmos os dados que processamos. Mais uma vez volto a relembrar ao leitor que para cada tipo de produto faremos um cabeçalho com os valores das encomendas.

```

foreach (DataRow r in dtEND.Rows)
    {
        string ProdutoTipoTabelaGrande =
            r[2].ToString();
        string numeroEncomendaTabelaGrande =
            r[3].ToString();
        string dataEncomendaTabelaGrande =
            r[4].ToString();
        string QtDENcomendaTabelaGrande =
            r[5].ToString();
        string ContratoTabelaGrande =
            r[9].ToString();

        if (contractHit.Equals(ContratoTabelaGrande))
            {
                //cabeçalho
                if (indicecabeçalho == 0)
                    //CABECALHO PRINCIPAL DE CLIENTE
            }

#region cabecalho
//CRIAR O CABEÇALHO
//*Cabeçalho do worksheet
*
*/
//Linha 1 - espaço em branco
//Linha 2 - escrever Auto de Faturação
ws.Cells["B2:H2"].Merge = true; //faz merge
//escreve auto de facturacao na linha 2
ws.Cells["B2"].Value = "Auto de Facturação";
ws.Cells["B2"].StyleName = "autofacStyle";
//aplica o style
ws.Cells["B2"].Style.HorizontalAlignment
    =OfficeOpenXml.Style.ExcelHorizontalAlignment.
        Center;

//Linha 3 - espaço em branco
//escreve auto de facturacao na celula G4
ws.Cells["G4"].Value = mesProcessamento;

//Linha 4 - Nome do Cliente - indice 6 da
// datatable
ws.Cells["B4:D4"].Merge = true; //faz merge
ws.Cells["B4"].Value = r[6].ToString(); //escreve
// auto de facturacao na linha 4
ws.Cells["B4:D4"].StyleName = "clientNameStyle";
ws.Cells["B4:D4"].Style.Font.SetFromFont(
    new System.Drawing.Font("Calibri", 9));
//coloca a fonte igual ao template
ws.Cells["B4:D4"].Style.Font.Bold = true;
ws.Cells["B4:D4"].Style.Fill.
    PatternType=OfficeOpenXml.
        Style.ExcelFillStyle.Solid;
ws.Cells["B4:D4"].Style.Fill.
    BackgroundColor.SetColor
        (System.Drawing.Color.Gray);

//Linha 5 - Coluna A - Morada do Cliente - indice
// 7 - Coluna E: Armazém -indice 9
ws.Cells["B5:D8"].Merge = true; //merge
ws.Cells["B5:D8"].Style.

```



# A PROGRAMAR

## C# - EXCEL

```
                HorizontalAlignment=OfficeOpenXml.  
                Style.ExcelHorizontalAlignment.Justify;  
                ws.Cells["B5:D8"].Style.VerticalAlignment =  
                OfficeOpenXml.Style.ExcelVerticalAlignment.  
                Justify;  
string morada = r[7].ToString();  
ws.Cells["B5"].Value = morada;  
    //Linha 5-Coluna A - Morada do Cliente - indice 7  
ws.Cells["E5"].Value = "Armazem:";  
ws.Cells["E5"].Style.Font.Bold = true;  
ws.Cells["F5"].Value = r[8].ToString();//Coluna F:  
                Armazem - indice 9  
  
ws.Cells["E7"].Value = "NIF:";  
ws.Cells["E7"].Style.Font.Bold = true;  
ws.Cells["F7"].Value = r[10].ToString();  
                //-- Coluna E: NIF - indice 11  
  
ws.Cells["E8"].Value = "Cond. Pag.");//--- Coluna E:  
ws.Cells["E8"].Style.Font.Bold = true;  
ws.Cells["F8"].Value = r[11].ToString();  
                //Cond Pagamento -- indice 12  
indicecabecalho = 1;  
indiceLinhasEscritas = 9;  
    //variável para controlar a escrita do ficheiro  
  
/**imprime o primeiro cabeçalho de todos**/  
#region primeiroCabeçalhoLinhas  
produto = r[2].ToString();  
indiceLinhasEscritas = indiceLinhasEscritas + 1;  
//lineCellProdHeaderIndex = lineCellProdHeaderIndex  
                + 1;//linha seguinte  
lineCellProdHeader = "B" +  
                indiceLinhasEscritas.ToString() + ":D" +  
                indiceLinhasEscritas.ToString();  
                //Exemplo: B10:D10  
ws.Cells["B" + indiceLinhasEscritas.ToString() +  
                ":D" + indiceLinhasEscritas.ToString()].Merge =  
                true; //faz merge  
ws.Cells["B" + indiceLinhasEscritas.  
                ToString()].Value = "Designação";  
//1ª LINHA coluna B + coluna C + coluna D -  
                // ESCRIVER "DESIGNAÇÃO" B10:D10  
ws.Cells[lineCellProdHeader].StyleName =  
                "designacaoStyle";  
                ws.Cells[lineCellProdHeader].Style.  
                HorizontalAlignment =  
                OfficeOpenXml.Style.  
                ExcelHorizontalAlignment.Center;  
indiceLinhasEscritas = indiceLinhasEscritas + 1;  
//aplica a moldura  
using (ExcelRange designacaoRange = ws.Cells  
                [lineCellProdHeader])//style  
                {designacaoRange.Style.Border.BorderAround  
                (OfficeOpenXml.Style.ExcelBorderStyle.Thick);  
                }  
lineCellProdHeader = "B" +  
                indiceLinhasEscritas.ToString() + ":D" +  
                indiceLinhasEscritas.ToString();  
                //Exemplo: B10:D10  
ws.Cells[lineCellProdHeader].StyleName =  
                "categoriaStyle";  
ws.Cells["B" + indiceLinhasEscritas.ToString() +  
                ":D" + indiceLinhasEscritas.ToString()].Merge =  
                true; //faz merge  
ws.Cells[lineCellProdHeader].Style.  
                HorizontalAlignment =  
                OfficeOpenXml.Style.  
                ExcelHorizontalAlignment.Center;  
                //centra o tipo de produto  
                ws.Cells["B" +  
                indiceLinhasEscritas.ToString()].  
                Style.Fill.PatternType =  
                OfficeOpenXml.Style.ExcelFillStyle.Solid;  
ws.Cells["B" + indiceLinhasEscritas.
```

```
                ToString()).Style.Fill.BackgroundColor.SetColor  
                (System.Drawing.Color.LightGray);  
//aplica a novamente a moldura  
using (ExcelRange prodRange = ws.Cells  
                [lineCellProdHeader]){  
  
                prodRange.Style.Border.BorderAround  
                (OfficeOpenXml.Style.ExcelBorderStyle.Thick);  
                }  
  
                produto = t[0].ToString();  
  
                ws.Cells["B" + indiceLinhasEscritas.  
                ToString()].Value = produto;  
  
                indiceLinhasEscritas = indiceLinhasEscritas + 1;  
  
                lineCellProdHeaderIndex++;  
                #endregion  
                //voltamos a inicializar a variável do cabeçalho  
                para escrever o próximo produto mas guardamos o a  
                linha em que estamos para que o próximo cabeçalho  
                de produto possa ser escrito no local correcto.  
                indicecablinhas = 1;  
                linhaInicioSoma = indiceLinhasEscritas;  
                }//FIM DO if (indicecabecalho == 0) //CABECALHO  
                PRINCIPAL DE CLIENTE
```

Falta-nos agora apenas a parte final da apresentação de valores do rodapé! Assim sendo passemos à ação:

```
//coloca o endereço da celula actual no somatório  
finalsumcellAux = indiceLinhasEscritas + 1;  
finalsumcell.Add("G" +  
                finalsumcellAux.ToString());  
  
#region rodapeFinal  
string totalsomas = "";  
foreach (string s in finalsumcell) {  
                totalsomas = totalsomas + "+" + s; }  
  
indiceLinhasEscritas = indiceLinhasEscritas +  
                4; //para sabermos quantas linhas  
  
wsr.Cells["B" + indiceLinhasEscritas.ToString()  
                ":F" + indiceLinhasEscritas.ToString()].Merge =  
                true;  
                wsr.Cells["B" + indiceLinhasEscritas.  
                ToString()].Value = "TOTAL";  
                wsr.Cells["B" + indiceLinhasEscritas.  
                ToString()].Style.Font.Color.SetColor  
                (System.Drawing.Color.White);  
                wsr.Cells["B" + indiceLinhasEscritas.  
                ToString()].Style.Font.SetFromFont(new  
                System.Drawing.Font("Calibri", 10));  
                wsr.Cells["B" + indiceLinhasEscritas.  
                ToString()].Style.Font.Bold = true;  
  
                wsr.Cells["G" + indiceLinhasEscritas.ToString()].  
                Formula = totalsomas;  
                wsr.Cells["G" + indiceLinhasEscritas.ToString()].  
                Style.Font.Color.SetColor  
                (System.Drawing.Color.White);  
                wsr.Cells["G" + indiceLinhasEscritas.ToString()].  
                Style.Font.SetFromFont(new  
                System.Drawing.Font("Calibri", 10));  
                wsr.Cells["G" + indiceLinhasEscritas.ToString()].  
                Style.Font.Bold = true;  
                wsr.Cells["G" + indiceLinhasEscritas.ToString()].  
                Style.Fill.PatternType =  
                OfficeOpenXml.Style.ExcelFillStyle.Solid;  
                wsr.Cells["G" + indiceLinhasEscritas.ToString()].  
                Style.Fill.BackgroundColor.SetColor
```



```
(System.Drawing.Color.Gray);
wsr.Cells["G" + indiceLinhasEscritas.ToString()].
    Style.Numberformat.Format = "#,##0.00";

wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas].Style.Fill.PatternType
    = OfficeOpenXml.Style.ExcelFillStyle.Solid;
wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas].Style.
    Fill.BackgroundColor.SetColor
        (System.Drawing.Color.Gray);
wsr.Cells["B" + indiceLinhasEscritas.ToString()
+ ":F" + indiceLinhasEscritas].Style.
    VerticalAlignment = OfficeOpenXml.
        Style.ExcelVerticalAlignment.Center;
wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas].Style.
    HorizontalAlignment = OfficeOpenXml.
        Style.ExcelHorizontalAlignment.Center;
lineCellSum = "B" + indiceLinhasEscritas.ToString()
+ ":F" + indiceLinhasEscritas.ToString();
pagefooteraux = "G" +
    indiceLinhasEscritas.ToString();
using (ExcelRange designacaoRange = wsr.Cells["G" +
    indiceLinhasEscritas.ToString()])//style
{
    designacaoRange.
        Style.Border.BorderAround
            (OfficeOpenXml.Style.ExcelBorderStyle.Thick);
}
indiceLinhasEscritas++;
wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas.ToString()].Merge =
    true;
wsr.Cells["B" + indiceLinhasEscritas.ToString()].
    Value = "IVA 23%";
wsr.Cells["B" + indiceLinhasEscritas.ToString()].
    Style.Font.Color.SetColor
        (System.Drawing.Color.White);
wsr.Cells["B" + indiceLinhasEscritas.ToString()].
    Style.Font.SetFromFont(new System.Drawing.Font
        ("Calibri", 10));
wsr.Cells["B" + indiceLinhasEscritas.ToString()].
    Style.Font.Bold = true;
wsr.Cells["G" + indiceLinhasEscritas.ToString()].
    Formula = "=" + pagefooteraux + "*23%";
//multiplica o valor anterior por 23% para saber a
// quanto corresponde
wsr.Cells["G" + indiceLinhasEscritas.ToString()].
    Style.Font.Color.SetColor
        (System.Drawing.Color.White);
wsr.Cells["G" + indiceLinhasEscritas.ToString()].
    Style.Font.SetFromFont(new System.Drawing.Font
        ("Calibri", 10));
wsr.Cells["G" + indiceLinhasEscritas.ToString()
()].Style.Font.Bold = true;
wsr.Cells["G" + indiceLinhasEscritas.ToString()
()].Style.Fill.PatternType = OfficeOpen-
    Xml.Style.ExcelFillStyle.Solid;
wsr.Cells["G" + indiceLinhasEscritas.ToString()
()].Style.Fill.BackgroundColor.SetColor
        (System.Drawing.Color.Gray);
wsr.Cells["G" + indiceLinhasEscritas.ToString()
()].Style.Numberformat.Format = "#,##0.00"; //para
```

apresentar o resultado com duas casas decimais

```
wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas].Style.Fill.PatternType = OfficeOpen-
    Xml.Style.ExcelFillStyle.Solid;
wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas].Style.Fill.BackgroundColor.SetColor
        (System.Drawing.Color.Gray);
wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas].Style.VerticalAlignment = OfficeOpen-
    Xml.Style.ExcelVerticalAlignment.Center;
wsr.Cells["B" + indiceLinhasEscritas.ToString() +
":F" + indiceLinhasEscritas].Style.HorizontalAlignment = OfficeOpen-
    Xml.Style.ExcelHorizontalAlignment.Center;

finalsumcell.Clear(); //limpa a memória existente
}
```

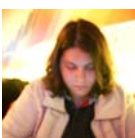
## Considerações finais

Neste projecto foi usado o package EPPlus, que facilita a criação de ficheiros em formato excel, de forma programática, usando C#. Este package pode ser obtido em: <https://www.nuget.org/packages/EPPlus/4.5.0.2-rc> ou instalado usando o gestor de pacotes nuget: `Install-Package EPPlus -Version 4.5.0.2-rc`

```
Install-Package EPPlus -Version 4.5.0.2-rc
```



## AUTOR



Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010. Embaixadora das Geek Girls Portugal – Núcleo de Lisboa.



# ELECTRÓNICA

Como configurar o Wi-Fi no ESP8266

## COMO CONFIGURAR O WI-FI NO ESP8266

O nosso famoso ESP8266 veio para ficar, cada vez mais este microcontrolador é utilizado para desenvolver dispositivos fantásticos, é verdade que o ESP pode ser utilizado como um simples *MCU*, mas é a sua capacidade de se ligar a uma rede Wi-Fi, faz dele a escolha perfeita no que toca dispositivos *IoT*.

Este artigo tem foco em três diferentes soluções que permitem programar o ESP de forma a que este se consiga ligar a uma rede Wi-Fi. É sabido que da primeira vez que o ESP é energizado este tem de conhecer ou vir a conhecer o SSID e a Password configurada no *Access Point* ao qual o queremos conectar de forma a fazer para da rede e ter acesso à Internet se assim for necessário.

Vamos então ver como implementar as três soluções, analisar cada uma delas e ponderar os prós e contras e cada uma delas.

Passemos ao caso mais simples, ao qual denomino de **“Hard Coded”**, esta é a forma mais utilizada por quem começa a utilizar o ESP nos seus projetos “caseiros”, é simples de implementar porque apenas temos que escrever diretamente no código o SSID e a Password da nossa rede Wi-Fi.

Bem se me perguntarem se seria um método que utilizasse, eu responderia que já utilizei, mas sempre procurei outras formas de tornar a configuração dinâmica.

Se olharmos para o código vamos perceber a simplicidade do mesmo:

```
#include <ESP8266Wifi.h>

void setup() {
  Wifi.begin("SSID_WI_FI", "SUPER_PASSWORD");
  //Aguarda estar ligado
  while (Wifi.status() != WL_CONNECTED) {
    delay(500);
  }
}

void loop() {
}
```

Apenas temos de incluir a biblioteca que nos permite utilizar a Wi-Fi através de funções de alto nível **<ESP8266Wifi.h>**, no método **setup** que apenas corre uma vez no arranque do nosso **ESP** utilizamos a função **Wifi.begin** que recebe o **SSID** da rede à qual queremos ligar e a **Password**, a partir daqui a magia acontece e o nosso pequeno microcontrolador de 2 euros começa a tentar ligar-se ao Wi-Fi. Podemos verificar no código, que existe um ciclo **while** que faz com que o processo de arranque não continue até que o **Wifi.status** seja igual a **WL\_CONNECTED**.

Em relação ao espaço ocupado, verificamos no output da consola do IDE do Arduino que utiliza 248631 bytes, isto num **ESP** versão 01 com apenas 512K seria 49% do espaço disponível, felizmente existe a versão ESP12 que já trás 4M neste apenas estaríamos a consumir 23% do espaço disponível.

Em relação ao tempo que demora a ligar-se a uma rede Wi-Fi com uma força de sinal a rondar os 60%, aqui é gasto cerca de 14 segundos em média.

Passamos agora à segunda implementação, esta é uma

das mais utilizadas em produtos IoT sejam protótipos ou versões comerciais.

Apareceu como uma forma “Easy” de configuração e utiliza técnicas avançadas que exploram as capacidades do sistema operativo no que toca a autoconfiguração (*easy-conf*).

O **Wi-Fi Manager** (<https://github.com/tzapu/WiFiManager>) é uma biblioteca open-source desenvolvida pelo Tzapu, esta transforma o nosso **ESP** num verdadeiro produto comercial capaz de ser configurado para se ligar a uma rede Wi-Fi sem grande dificuldade, permitindo assim ao consumidor final escolher a rede a que se quer ligar de uma forma simples sem ter que escrever uma única linha de código.

Vamos olhar para o código e perceber o que é possível fazer com esta maravilhosa biblioteca:

```
#include <ESP8266Wifi.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WifiManager.h>

void setup() {
  WifiManager wifiManager;
  wifiManager.autoConnect("OMEUESP");
}

void loop() {
}
```

Neste exemplo começamos claro por utilizar novamente biblioteca **<ESP8266Wifi.h>**, pois é ela que contém toda a lógica para utilizar o **Wi-Fi**, de seguida adicionamos duas bibliotecas de **requisito obrigatório** para que o **Wi-Fi Manager compile corretamente**, por fim adicionamos a biblioteca que nos irá facilitar a configuração **<WifiManager.h>**.

Fazendo uma explicação superficial de cada uma das bibliotecas:

- **<DNSServer.h>**, ativa serviços de DNS para que seja mais simples encontrar o nosso ESP na rede, os DNS apareceram para facilitar o acesso a dispositivos ligados uma rede TCP/IP, em vez de termos que decorar um IP 192.168... ou passar a rede a pente fino para descobrir que IP foi atribuído ao nosso **ESP**, com esta biblioteca podemos simplesmente aceder ao ESP utilizando o nome que definimos no código seguido de **.local**, utilizando o código acima como exemplo vemos que na função **wifiManager.autoConnect(“OMEUESP”)**, passamos a String **OMEUESP** caso queiramos saber o IP dele ou verificar se está ativo na rede podemos fazer um ping a **omeuesp.local** e será devolvido o resultado do ping que contem o tempo que demorou e o IP do ESP. Mais informações sobre o comando ping em <https://pt.wikipedia.org/wiki/Ping>.

# Electrónica

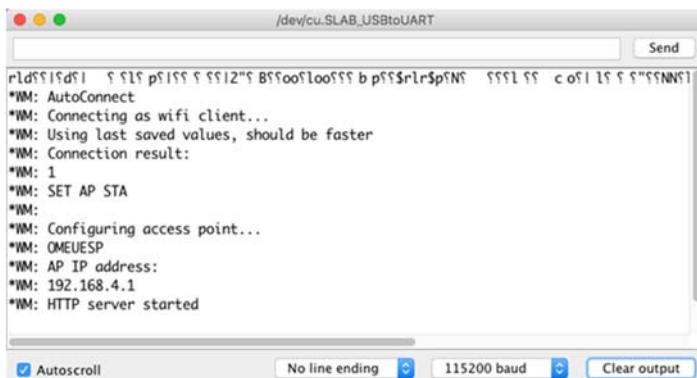
## COMO CONFIGURAR O WI-FI NO ESP8266

- **<ESP8266WebServer.h>**, como o nome indica esta biblioteca ativa funcionalidades de um **Web Server**, sim um Web Server a correr num chip que custa menos que um pastel de nata. Este Web Server vai disponibilizar uma bela interface onde um utilizador que nada entende de linguagens de programação possa de forma simples escolher uma das redes detetadas pelo ESP e configurar a sua Password.
- **<Wi-FiManager.h>**, por fim a biblioteca Wi-FiManager trata da inicialização de um **Captive Portal** para configurar e pesquisar redes disponíveis, também é esta biblioteca que decide se o ESP entra em **Modo ACCESS POINT** ou em **Modo CLIENT**, sendo que o modo AP serve para nos ligar a uma rede criada pelo nosso ESP com o nome que definimos, com base no exemplo acima seria **OMEUESP**.

Ao ligarmo-nos o nosso computador ou smartphone a essa rede, é automaticamente aberta uma janela de configuração (**Captive Portal**), por outro lado caso o ESP já se tenha ligado com sucesso a uma rede, esta deixa de aparecer e passa a funcionar apenas como se fosse mais um cliente na nossa rede.

Com uma breve explicação das bibliotecas passamos à parte interessante, o que acontece depois de termos carregado o código com o **Wi-Fi Manager** no nosso ESP8266.

Bem se ainda tivermos o nosso ESP ligado ao USB do nosso computador, podemos carregar na Serial Console do Arduino e “cuscar” o que está a acontecer.



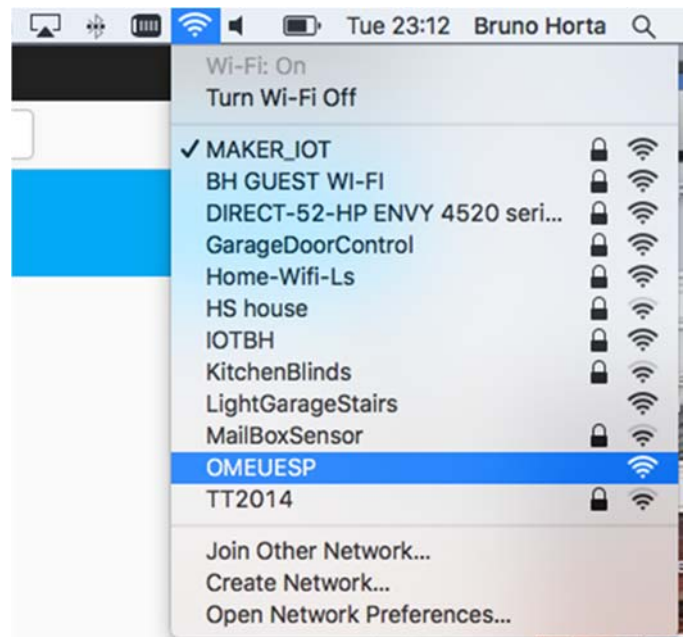
```

/dev/cu.SLAB_USBtoUART
Send
rld5f1fd5l 5 f1f p5155 5 5512"5 B5foof1oo555 b p55$rlr5p5N5 555l 55 c ofl l5 5 5"55NN5l
*WM: AutoConnect
*WM: Connecting as wifi client...
*WM: Using last saved values, should be faster
*WM: Connection result:
*WM: 1
*WM: SET AP STA
*WM:
*WM: Configuring access point...
*WM: OMEUESP
*WM: AP IP address:
*WM: 192.168.4.1
*WM: HTTP server started
Autoscroll No line ending 115200 baud Clear output

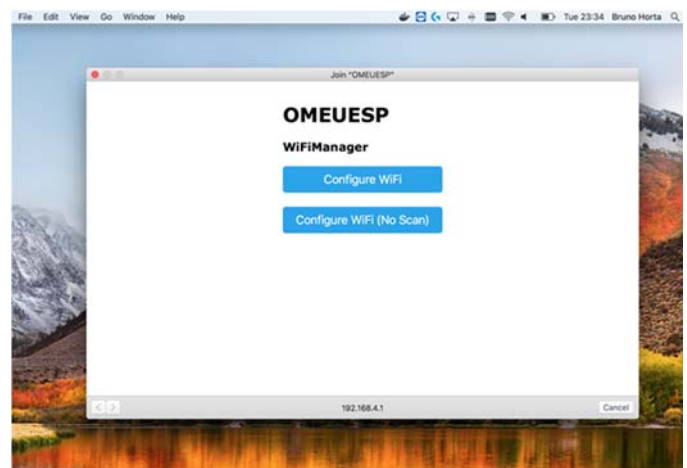
```

Analisando o log vemos que existe uma tentativa de se ligar como cliente, mas como é normal ele ainda não tem nenhuma rede gravada e o resultado é 1, o que quer dizer que falhou a ligação, em consequência disso ele inicializa-se em modo **Access Point** e cria uma rede chamada **OMEUESP** no caso do nosso exemplo, o IP 192.168.4.1 é sempre o mesmo e apenas existe para o caso de algum sistema operativo não suportar Captive Portal, fazendo com que tenhamos em alternativa usar esse IP no nosso browser para configurar o ESP, bem aqui já vemos um pequeno contratempo mas por experiência própria o Captive Portal abre em 99% das vezes.

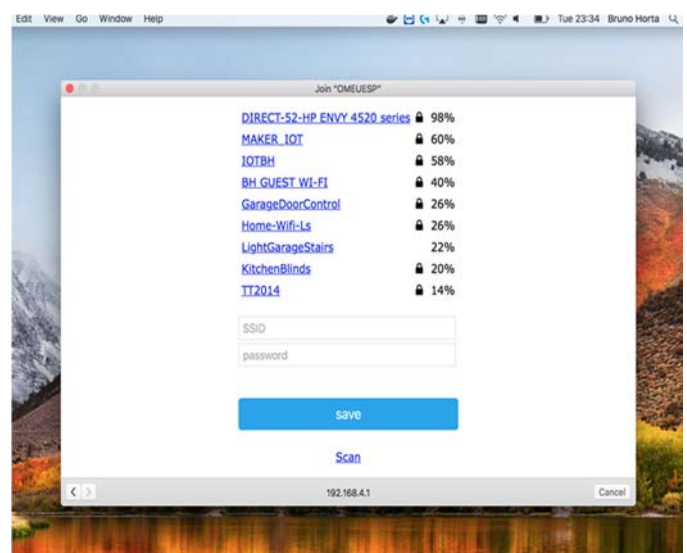
Então se o nosso ESP está em modo Access Point é possível detetar o mesmo e ligarmo-nos a ele através do nosso smartphone ou portátil.



Após uma conexão à rede criada é automaticamente aberto um portal no nosso computador, para quem já se ligou a Hot-spots públicos vai reconhecer o método.



Depois basta clicarmos no botão **Configure WiFi** e escolher a rede.





Resumindo a partir daqui o nosso ESP fica ligado a rede Wi-Fi configurada e pode ser reiniciado que não perde as configurações, caso a Rede deixe de existir ou a password altere ele entra em modo de configuração automaticamente.

Comparando com o método **HardCoded**, este utiliza mais 15% do espaço do nosso ESP em relação a anterior, a velocidade de ligação pós configuração ronda em média os 16 segundos face aos anteriores 14 segundos, no entanto tudo isso compensa porque oferece uma forma muito versátil de configurar e disponibiliza documentação bastante detalhada.

*Se és programador e estas a pensar se é possível alterar o Captive Portal de forma a espelhar a tua marca, a resposta é sim claro, o código fonte está disponível no GitHub e só tens de fazer HTML simples ☺.*

Agora perto do fim deixamos o melhor para terminar em grande.

A ultimo método que vou demonstrar utiliza um protocolo nativo da *Espressif*, os construtores do ESP8266.

Só agora é que se começa a falar mais nele, pois a documentação acerca do ESP8266 começa a ser mais rica e escrita em Inglês, quando este foi lançado apenas existia pouca coisa, o que se encontrava estava em chinês, por isso como qualquer grande programador a malta deixou a documentação de lado e começou a criar formas de configurar o ESP vimos a versão **HardCoded** e outra utilizando a biblioteca **Wi-Fi Manager**, bom mas quem desenvolveu este fabuloso microcontrolador com Wi-Fi, pensou muito bem como é que este podia ser configurado de forma simples e é aqui que entra o protocolo **ESP-TOUCH**.

Este protocolo é desenvolvido diretamente pela *Espressif* e é suportado nativamente pelos ESP8266, permite uma rápida configuração de um ou milhares de ESP's ao mesmo tempo e não obriga a termos de nos ligar a ele para o configurar... sim podes estar a achar estranho, mas no fim do ESP entrar em modo **SmartConfig**, fica a escuta de uma ligação **UDP** que trás a configuração cifrada para o acesso à rede Wi-Fi.

Para implementarmos este protocolo a *Espressif* disponibiliza o **SDK** para criarmos aplicações capazes de criar essa ligação **UDP** e **cifrar** o conteúdo de forma a que apenas o ESP o entenda.

A documentação oficial pode ser encontrada em [https://www.espressif.com/sites/default/files/30b-esp-touch\\_user\\_guide\\_en\\_v1.1\\_20160412\\_0.pdf](https://www.espressif.com/sites/default/files/30b-esp-touch_user_guide_en_v1.1_20160412_0.pdf)

Os SDK para desenvolver aplicações para IOS ou Android podem ser encontradas em

<https://github.com/EspressifApp/>

Se não quiserem desenvolver a vossa própria App, existem disponíveis na App Store ou na Play Store, basta pesquisar por "*ESP Smart Config*" estas são **free**.

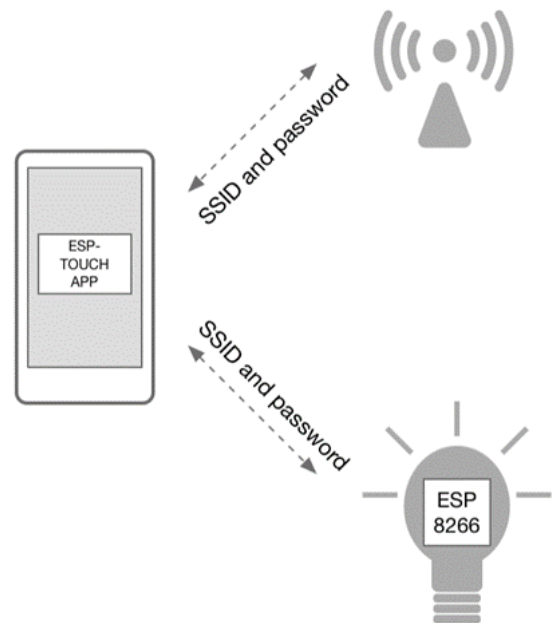
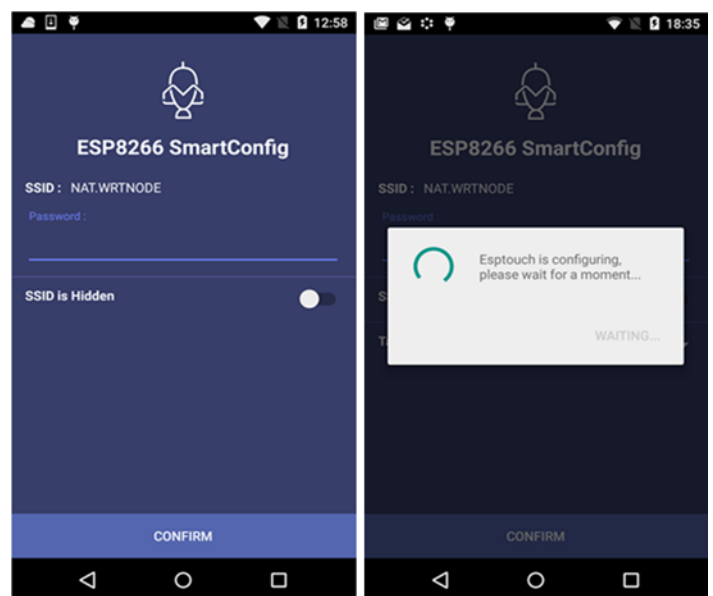


Figure 1-1 Typical ESP-TOUCH Application

Imagem retirada da documentação oficial que exemplifica a arquitetura de comunicação utilizada pelo protocolo **ESP-TOUCH**, podemos constatar que não existe ligação direta entre o smartphone e o ESP, o que torna este método maravilhoso face ao anterior que tínhamos de andar a trocar de Redes Wi-Fi, confundindo assim o utilizador menos experiente.

Com o ESP-TOUCH o utilizador não tem de se desligar da sua rede Wi-Fi já configurada no seu smartphone, apenas tem de abrir a App SmartConfig e digitar a Password da sua Rede Wi-Fi.



Deixando então a parte a configuração, vamos analisar o código que temos de colocar no ESP para ter o ESP-TOUCH a bombar.

# Electrónica

## COMO CONFIGURAR O WI-FI NO ESP8266

```
#include <ESP8266Wifi.h>

void setup() {
  WiFi.mode(WIFI_STA);
  WiFi.beginSmartConfig();
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}

void loop() {
}
```

Nesta versão voltamos apenas a necessitar da biblioteca base do **<ESP8266Wifi.h>**, o que é ótimo para uma otimização de espaço utilizado, com o ESP-TOUCH utilizamos 7% a mais face à versão **HardCoded** e menos 8% face ao **WiFiManager**.

Para este modo funcionar o ESP tem de estar em **Modo Station** e isso é conseguido utilizando a função **WiFi.mode(WIFI\_STA)**, após isso damos início à configuração **“Smart”** com a função **WiFi.beginSmartConfig()**, o ciclo while já é conhecido do método **HardCoded** e serve apenas para não deixar o processamento avançar até que o ESP tenha ligação ao Wi-Fi.

Notas importantes sobre o código acima, da forma que está sempre que o **ESP fizer restart fica em modo SmartConfig**, para isso é necessário implementar algum código extra de forma a detetar se já existe alguma configuração prévia, e se sim então tentar usar primeiro a mesma, caso esta falhe aí sim colocamos em modo **SmartConfig**.

Deixo apenas um exemplo de como seria uma implementação com time out e reciclagem de configurações prévias.

```
#include <ESP8266Wifi.h>

bool timeout = true;

void setup() {
  WiFi.mode(WIFI_STA);
  if (WiFi.SSID() != NULL) {
    WiFi.begin(WiFi.SSID().c_str(), WiFi.psk
               (WiFi.SSID().c_str()));
  }

  long start = millis();
  while (millis() - start < 30000) {
    if (WiFi.status() == WL_CONNECTED) {
      timeout = false;
      break;
    }
    delay(500);
  }
}

if(timeout){
  timeout = true;
  WiFi.beginSmartConfig();
}
```

```
long start = millis();
while(millis() - start < 30000){
  if(WiFi.status() == WL_CONNECTED){
    timeout = false;
    break;
  }
  delay(500);
}
}

if(timeout){
  while(1)ESP.restart();
  delay(500);
}
}

void loop(void) {
}
```

Concluindo, a versão **HardCoded** é ótima para quando estamos a desenvolver algo que queremos apenas testar naquele instante, isto porque não consome nenhum tempo de configuração extra, o **WiFi Manager** é ótimo para desenvolvermos um produto que possa ser configurado de forma minimamente simples, no entanto sofre de alguns Bugs, algumas vezes após ser configurado com sucesso não desativa o modo AP, criando alguma entropia no nosso router/ap, o facto de termos que trocar de rede para o configurar baralha utilizados menos experientes e temos de configurar um ESP de cada vez, por fim o **ESP-TOUCH** é uma forma mais simples de configurar mas obrigada o utilizador a instalar um APP no seu smartphone, no entanto podemos configurar dezenas ao mesmo tempo com um só clique na App.

Com todos estes prós e contras penso que a escolha do método tem de ir de encontro ao que queremos desenvolver, por isso nenhum deles deve ser deixado ao acaso.

Fica uma tabela que tenho alimentado ao longo das minhas experiências com alguns comparativos de cada um deles.

TAMANHO DA APLICAÇÃO	VELOCIDADE DE CONEÇÃO (milis)	VANTAGENS	DESVANTAGENS	DOCUMENTAÇÃO	USER FRIENDY	Tempo de configuração	
HARD CODED	251687(41%) kB	1404	Fácil de implementar em código	Se a password ou a rede mudar é necessário reprogramar o ESP8266	<a href="https://arduino-esp8266.readthedocs.io/en/2.4.0/">https://arduino-esp8266.readthedocs.io/en/2.4.0/</a>	Não	0 segundos
WI-FI MANAGER	323776(30%) kB	1409	Simples de implementar. Boa documentação. Permite Timeout caso o router não esteja ainda disponível.	Fazer o utilizador trocar de rede para o configurar. Cria um Access Point. Se tivermos que configurar dezenas vão entrar em concorrência pelo canal Wi-Fi. A configuração tem de ser ESP a ESP	<a href="https://github.com/tzapu/WiFiManager">https://github.com/tzapu/WiFiManager</a>	Para utilizadores com conhecimento de Redes.	Entre 40 a 60 segundos
ESP TOUCH	28572(107%) kB	1401	Não confunde o utilizador a trocar de rede para se ligar ao dispositivo. É Nativo da Expressif	Obriga a instalar uma APP	<a href="http://www.expressif.com/sites/default/files/2016-08/touch_user_guide_en_v1.1.20160412_0.pdf">http://www.expressif.com/sites/default/files/2016-08/touch_user_guide_en_v1.1.20160412_0.pdf</a> <a href="https://github.com/ExpressifApp/">https://github.com/ExpressifApp/</a>	Totalmente	Depende do numero de ESPs. Em média cada um demora 20 a 30 segundos

## AUTOR



Escrito por Bruno Horta

Licenciado em Engenharia Informática pela ESTG. É Senior Java Developer, sendo o seu principal foco o desenvolvimento de aplicações distribuídas e de alto desempenho. Adora desenvolver protótipos para serem utilizados no contexto da Internet das Coisas. É membro fundador do Movimento Maker Portugal. LinkedIn: <https://www.linkedin.com/in/brunohorta/>

# COLUNAS

**C# - Cifras e mais Cifras**

**Kernel Panic - A elegância da simplicidade**

**SQL Curtas #3: Mais um conjunto de dúvidas**

## Cifras e mais Cifras

Neste artigo decidi “revisitar” um tema que sempre me despertou interesse e que tive de abordar num projecto que fiz em 2015. O objectivo era encriptar, de forma a que fosse possível desencriptar, dados, bastando para tal saber um conjunto de informações (identificador, chave e chave-mestra), uma vez que a cada dado, apenas corresponde um identificador (fechadura) e para se chegar à chave que acabará por abrir, são precisas duas chaves (chave e a chave mestra). Quase como a caixa forte do tio patinhas e os irmãos metralha a tentar abrir, mas um pouco mais complexo!

Para fazer executar esta tarefa, o utilizador deve inserir um identificador, neste caso “porta”, um código a ser cifrado, uma chave para o cifrar e a respectiva chave par (chave mestra). Quando pressionar o botão para cifrar, os dados introduzidos, serão utilizados para cifrar o “código” usando a chave e a chave mestra, devolvendo o valor do código, cifrado.

Até aqui tudo é extremamente simples, a dificuldade, acontece quando se pretende cifrar um grande volume de dados usando “chaves diferentes”, para cada um deles, sem ter de estar a pensar e introduzir uma chave de cada vez. A solução encontrada pode ser pouco elegante, mas funciona!

Alguns requisitos importantes da solução:

- Não repetir key-rings
- Evitar patens
- Permitir encriptar quantidades massivas de dados
- Fazê-lo de forma minimamente robusta

A solução passou, por usar uma classe dedicada à geração de números aleatórios, para caso o utilizador não introduza uma chave, ser gerada uma automaticamente, e uma outra classe dedicada à cifragem. Existem muitas soluções possíveis, uma vez que o problema é amplo, caberá ao leitor escolher a que melhor se adequa ao seu objectivo.

Gerador de senhas aleatórias:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;

namespace safeKeys
{
    /// <summary>
    /// Classe geradora de strings aleatórias de
    /// acordo com as opções abaixo listadas
    /// 1) 4 caracteres (maiusculo, minusculo,
    /// numerico e caracteres especiais)
    /// 2) numero variável de caracteres em uso
    /// 3) numero minimo de caracteres de cada tipo
    /// a serem usados na string
    /// 4) Geração orientada a patterns
    /// 5) geração de strings unicas

```

```
/// 6) usar cada caracter apenas uma vez
/// feito para gerar "keyt" para senhas
    Rjindael

    /// </summary>
    public class RamdomStringGenerator
    {
        /// <summary>
        /// Classe geradora de strings aleatórias
        /// de acordo com as opções abaixo listadas
        /// 1) 4 caracteres (maiusculo,
        /// minusculo, numerico e caracteres especiais)
        /// 2) numero variável de caracteres em
        /// uso
        /// 3) numero minimo de caracteres de ca
        /// da tipo a serem usados na string
        /// 4) Geração orientada a patterns
        /// 5) geração de strings unicas
        /// 6) usar cada caracter apenas uma vez
        /// feito para gerar "keyt" para senhas
        Rjindael

        /// </summary>
        public class RandomStringGenerator
        {
            public RandomStringGenerator(bool
                UseUpperCaseCharacters = true,
                bool UseLowerCaseCharacters =
                    true,
                bool UseNumericCharacters =
                    true,
                bool UseSpecialCharacters =
                    true)
            {
                m_UseUpperCaseCharacters =
                    UseUpperCaseCharacters;
                m_UseLowerCaseCharacters =
                    UseLowerCaseCharacters;
                m_UseNumericCharacters =
                    UseNumericCharacters;
                m_UseSpecialCharacters =
                    UseSpecialCharacters;
                CurrentGeneralCharacters = new
                char[0]; // evita excepções de null
                UpperCaseCharacters =
                    "ABCDEFGHIJKLMNOPQRSTUVWXYZ".ToCharArray();
                LowerCaseCharacters =
                    "abcdefghijklmnopqrstuvwxyz".ToCharArray();
                NumericCharacters =
                    "0123456789".ToCharArray();
                SpecialCharacters = ",.!:?/@#$$%
                    ^&()+*-_{}[]<>|~".ToCharArray();
                MinUpperCaseCharacters =
                MinLowerCaseCharacters = MinNumericCharacters =
                MinSpecialCharacters = 0;
                RepeatCharacters = true;
                PatternDriven = false;
                Pattern = "";
                Random =
                    new RNGCryptoServiceProvider();
                ExistingStrings = new
                List<string>();
            }

            #region character sets managers

            /// <summary>
            /// True se precisar-mos de um numero
            /// de caracteres fixo
            /// </summary>
            public bool UseUpperCaseCharacters
            {
                get

```



```

        {
            return
                m_UseUpperCaseCharacters;
        }
        set
        {
            if
                (CurrentUpperCaseCharacters != null)
                CurrentGeneralCharacters =
                    CurrentGeneralCharacters.Except
                    (CurrentUpperCaseCharacters).ToArray();
            if (value)
                CurrentGeneralCharacters =
                    CurrentGeneralCharacters.Concat
                    (CurrentUpperCaseCharacters).ToArray();
            m_UseUpperCaseCharacters =
                value;
        }
    }
    /// <summary>
    /// define ou obter a definição de
    /// caracteres maiusculos.
    /// </summary>
    public char[] UpperCaseCharacters
    {
        get
        {
            return CurrentUpperCaseCharacters;
        }
        set
        {
            if (UseUpperCaseCharacters)
            {
                if
                    (CurrentUpperCaseCharacters != null)
                    CurrentGeneralCharacters =
                        CurrentGeneralCharacters.Except
                        (CurrentUpperCaseCharacters).ToArray();
                CurrentGeneralCharacters =
                    CurrentGeneralCharacters.Concat(value).ToArray();
            }
            CurrentUpperCaseCharacters =
                value;
        }
    }
    /// <summary>
    /// True se é obrigatorío o uso de
    /// minúsculas
    /// </summary>
    public bool UseLowerCaseCharacters
    {
        get
        {
            return
                m_UseLowerCaseCharacters;
        }
        set
        {
            if
                (CurrentLowerCaseCharacters != null)
                CurrentGeneralCharacters =
                    CurrentGeneralCharacters.Except
                    (CurrentLowerCaseCharacters).ToArray();
            if (value)
                CurrentGeneralCharacters =
                    CurrentGeneralCharacters.Concat
                    (CurrentLowerCaseCharacters).ToArray();
            m_UseLowerCaseCharacters =
                value;
        }
    }
    /// <summary>
    /// define ou obtém a definição de uso

```

```

        de caracteres minuscu-
los
    /// </summary>
    public char[] LowerCaseCharacters
    {
        get
        {
            return CurrentLowerCaseCharacters;
        }
        set
        {
            if (UseLowerCaseCharacters)
            {
                if
                    (CurrentLowerCaseCharacters != null)
                    CurrentGeneralCharacters =
                        CurrentGeneralCharacters.Except
                        (CurrentLowerCaseCharacters).ToArray();
                CurrentGeneralCharacters
                    = CurrentGeneralCharacters.Concat(value)
                    .ToArray();
            }
            CurrentLowerCaseCharacters =
                value;
        }
    }
    /// <summary>
    /// True se é necessário o uso de
    /// caracteres numéricos
    /// </summary>
    public bool UseNumericCharacters
    {
        get
        {
            return
                m_UseNumericCharacters;
        }
        set
        {
            if
                (CurrentNumericCharacters != null)
                CurrentGeneralCharacters
                    = CurrentGeneralCharacters.Except
                    (CurrentNumericCharacters).ToArray();
            if (value)
                CurrentGeneralCharacters
                    = CurrentGeneralCharacters.Concat
                    (CurrentNumericCharacters).ToArray();
            m_UseNumericCharacters =
                value;
        }
    }
    /// <summary>
    /// define ou obtém a definição de
    /// uso de caracteres numéricos
    /// </summary>
    public char[] NumericCharacters
    {
        get
        {
            return
                CurrentNumericCharacters;
        }
        set
        {
            if (UseNumericCharacters)
            {
                if
                    (CurrentNumericCharacters != null)
                    CurrentGeneralCharacters
                        = CurrentGeneralCharacters.Except
                        (CurrentNumericCharacters).ToArray();
                CurrentGeneralCharacters
                    = CurrentGeneralCharacters.Concat(value)

```

```

        }
        CurrentNumericCharacters =
            value;
    }
}
/// <summary>
/// True se é para usar caracteres
/// especiais
/// </summary>
public bool UseSpecialCharacters
{
    get
    {
        return m_UseSpecialCharacters;
    }
    set
    {
        if (CurrentSpecialCharacters !=
            null)
            CurrentGeneralCharacters =
                CurrentGeneralCharacters.Except
                (CurrentSpecialCharacters).ToArray();
        if (value)
            CurrentGeneralCharacters =
                CurrentGeneralCharacters.Concat
                (CurrentSpecialCharacters).ToArray();
        m_UseSpecialCharacters = value;
    }
}
/// <summary>
/// define ou obtém a definição de uso
/// de caracteres especiais
/// </summary>
public char[] SpecialCharacters
{
    get
    {
        return CurrentSpecialCharacters;
    }
    set
    {
        if (UseSpecialCharacters)
        {
            if
                (CurrentSpecialCharacters != null)
                CurrentGeneralCharacters
                = CurrentGeneralCharacters.Except
                (CurrentSpecialCharacters).ToArray();
            CurrentGeneralCharacters =
                CurrentGeneralCharacters.Concat(value).ToArray();
            CurrentSpecialCharacters =
                value;
        }
    }
}
#endregion character sets managers
#region character limits
/// <summary>
/// Define ou obtém o numero minimo de
/// caracteres maiusculos a serem usados.
/// </summary>
public int MinUpperCaseCharacters
{
    get { return
        m_MinUpperCaseCharacters; }
    set { m_MinUpperCaseCharacters =
        value; }
}
/// <summary>

```

```

/// Define ou obtém o numero minimo
de
caracteres minusculos a serem usa-
dos.
/// </summary>
public int MinLowerCaseCharacters
{
    get { return
        m_MinLowerCaseCharacters; }
    set { m_MinLowerCaseCharacters =
        value; }
}
/// <summary>
/// define ou obtém o numero minimo
de caracteres numéricos a sere utilizados.
/// </summary>
public int MinNumericCharacters
{
    get { return
        m_MinNumericCharacters; }
    set { m_MinNumericCharacters =
        value; }
}
/// <summary>
/// define ou obtém o numero minimo
de caracteres especiais a serem utilizados.
/// </summary>
public int MinSpecialCharacters
{
    get { return
        m_MinSpecialCharacters; }
    set { m_MinSpecialCharacters =
        value; }
}
#endregion character limits
#region pattern
private string m_pattern;
/// <summary>
/// Define o padrão a ser seguido
para gerar uma string.
/// Este valor é ignorado se for
igual string vazia.
/// Os padrões são:
/// L - para letra maiúscula
/// l - para letra minúscula
/// N - de número
/// S - para caractere especial
/// * - Para qualquer caractere
/// </summary>
private string Pattern
{
    get
    {
        return m_pattern;
    }
    set
    {
        if (!value.Equals
            (String.Empty))
            PatternDriven = true;
        else
            PatternDriven = false;
        m_pattern = value;
    }
}
#endregion pattern
#region generators

```

```

    /// <summary>
    /// Gerar uma string que segue o
    /// padrão.
    /// Caracteres possíveis são:
    /// L - para letra maiúscula
    /// l - para letra minúscula
    /// N - de número
    /// S - para caractere especial
    /// * - Para qualquer caracter
    /// </summary>
    /// <param name="Pattern">o pattern na
    ser seguido enquanto gera as strings</param>
    /// <returns>um padrão aleatório que é
    retornado após a geração</returns>
    public string Generate(string Pattern)
    {
        this.Pattern = Pattern;
        string res = GenerateString
            (Pattern.Length);
        this.Pattern = "";
        return res;
    }

    /// <summary>
    /// gera uma string de comprimento
    variável compreendido entre MinLength e
    MaxLength. Os caracteres
    /// devem ser definidos antes de ser
    chamada esta função
    /// </summary>
    /// <param name="MinLength">cumprimento
    minimo da string string</param>
    /// <param name="MaxLength">Cumprimento
    maximo da string</param>
    /// <returns>uma string aleatória de um
    tamanho compreendido entre o minimo e o
    maximo</returns>
    public string Generate(int MinLength,
        int MaxLength)
    {
        if (MaxLength < MinLength)
            throw new ArgumentException
                ("Maximal length should be grater than minumal");
        int length = MinLength +
            (GetRandomInt() % (MaxLength - MinLength));
        return GenerateString(length);
    }

    /// <summary>
    /// Gera uma string de comprimento fixo
    /// os conjuntos de caracteres
    utilizaveis devem ser definidos antes de
    chamar esta função
    /// </summary>
    /// <param name="FixedLength">
    comprimento da string</param>
    /// <returns>uma string aleatória do
    comprimento desejado</returns>
    public string Generate(int FixedLength)
    {
        return GenerateString(FixedLength);
    }

    /// <summary>
    /// Metodo de geração principal que
    escolhe o metodo de geração adequado.
    /// procura situações excepcionais
    também.
    /// </summary>
    private string GenerateString(int
        length)
    {
        if (length == 0)
            throw new ArgumentException
                ("Não se pode gerar uma string com zero
                caracteres");

```

```

        if (!UseUpperCaseCharacters && !
            UseLowerCaseCharacters
            && !
            UseNumericCharacters && !
            UseSpecialCharacters)
            throw new ArgumentException
                ("Tem de se usar pelo menos um conjunto
                de
                caracteres! É que é burro al-
                vin! :D");
        if (!RepeatCharacters &&
            (CurrentGeneralharacters.Length < len-
            gth))
            throw new ArgumentException
                ("não existem caracteres suficientes para gerar a
                string sem repetir caracteres");
        string result = ""; // Esta
            string contem o resultado
        if (PatternDriven)
        {
            // usando a pattern para
            gerar algo
            result = PatternDrivenAlgo
                (Pattern);
        }
        else if (MinUpperCaseCharacters
            == 0 && MinLowerCaseCharacters == 0 &&
            MinNumericCharacters == 0 &&
            MinSpecialCharacters == 0)
        {
            // usando o algoritmo mais
            simples, neste caso
            result = SimpleGenerateAlgo
                (length);
        }
        else
        {
            // atenção ao limite
            result =
                GenerateAlgoWithLimits(length);
        }
        // suporte para strings unicas
        // recursão, a possibilidade de
        stack overflow é grande para
        strings maiores que 3 chars.
        try
        {
            if (UniqueStrings &&
                ExistingStrings.Contains(result))
                return GenerateString
                    (length);
            AddExistingString(result);
        }
        // guarda histórico
        catch { throw; } // intercepta o
            overflow e manda-o de
            volta (pro raio que o parta)
        return result;
    }

    /// <summary>
    /// gera uma string aleatória baseada
    na pattern
    /// </summary>
    private string PatternDrivenAlgo
        (string Pattern)
    {
        string result = "";
        List<char> Characters = new
            List<char>();
        foreach (char character in
            Pattern.ToCharArray())
        {
            char newChar = ' ';
            switch (character)
            {

```

```

        case 'L':
        {
            newChar =
                GetRandomCharFromArray
                (CurrentUpperCaseCharacters, Characters);
            break;
        }
        case 'l':
        {
            newChar =
                GetRandomCharFromArray
                (CurrentLowerCaseCharacters, Characters);
            break;
        }
        case 'n':
        {
            newChar =
                GetRandomCharFromArray
                (CurrentNumericCharacters, Characters);
            break;
        }
        case 's':
        {
            newChar =
                GetRandomCharFromArray
                (CurrentSpecialCharacters, Characters);
            break;
        }
        case '*':
        {
            newChar =
                GetRandomCharFromArray
                (CurrentGeneralCharacters, Characters);
            break;
        }
        default:
        {
            throw new Exception
                ("0 character '" + character + "' não é suportado");
        }
        Characters.Add(newChar);
        result += newChar;
    }
    return result;
}

/// <summary>
///
/// </summary>
private string SimpleGenerateAlgo(int
    length)
{
    string result = "";
    for (int i = 0; i < length; i++)
    {
        char newChar =
            CurrentGeneralCharacters[GetRandomInt()
                % CurrentGeneralCharacters.Length];
        if (!RepeatCharacters &&
            result.Contains(newChar))
        {
            do
            {
                newChar =
                    CurrentGeneralCharacters[GetRandomInt() %
                        CurrentGeneralCharacters.Length];
            } while (result.Contains
                (newChar));
        }
        result += newChar;
    }
    return result;
}

```

```

/// <summary>
///
/// </summary>
private string GenerateAlgoWithLimits
    (int length)
{
    if (MinUpperCaseCharacters +
        MinLowerCaseCharacters +
        MinNumericCharacters +
        MinSpecialCharacters > length)
    {
        throw new ArgumentException
            ("Sum of MinUpperCaseCharacters,
                MinLowerCaseCharacters," +
                " MinNumericCharacters and
                MinSpecialCharacters is greater than length");
    }
    if (!RepeatCharacters &&
        (MinUpperCaseCharacters >
            CurrentUpperCaseCharacters.Length))
        throw new ArgumentException
            ("Can't generate a string with this number of
                MinUpperCaseCharacters");
    if (!RepeatCharacters &&
        (MinLowerCaseCharacters >
            CurrentLowerCaseCharacters.Length))
        throw new ArgumentException
            ("Can't generate a string with
                this number of MinLowerCaseCharacters");
    if (!RepeatCharacters &&
        (MinNumericCharacters >
            CurrentNumericCharacters.Length))
        throw new ArgumentException
            ("Can't generate a string with this
                number of MinNumericCharacters");
    if (!RepeatCharacters &&
        (MinSpecialCharacters >
            CurrentSpecialCharacters.Length))
        throw new ArgumentException
            ("Can't generate a string with
                this number of MinSpecialCharacters");
    int AllowedNumberOfGeneralChatacters
        = length - MinUpperCaseCharacters -
            MinLowerCaseCharacters -
            MinNumericCharacters -
            MinSpecialCharacters;

    string result = "";
    List<char> Characters = new
        List<char>();

    for (int i = 0; i <
        MinUpperCaseCharacters; i++)
        Characters.Add
            (GetRandomCharFromArray(UpperCaseCharacters,
                Characters));
    for (int i = 0; i <
        MinLowerCaseCharacters; i++)
        Characters.Add
            (GetRandomCharFromArray(LowerCaseCharacters,
                Characters));
    for (int i = 0; i <
        MinNumericCharacters; i++)
        Characters.Add
            (GetRandomCharFromArray(NumericCharacters,
                Characters));
    for (int i = 0; i <
        MinSpecialCharacters; i++)
        Characters.Add
            (GetRandomCharFromArray(SpecialCharacters,
                Characters));
    for (int i = 0; i <
        AllowedNumberOfGeneralChatacters; i++)
        Characters.Add

```



```

(GetRandomCharFromArray(CurrentGeneralCharacters,
                        Characters));

    for (int i = 0; i < length; i++)
    {
        int position = GetRandomInt()
            % Characters.Count;
        char CurrentChar = Characters
            [position];
        Characters.RemoveAt
            (position);
        result += CurrentChar;
    }
    return result;
}

#endregion generators

public bool RepeatCharacters;

public bool UniqueStrings;

public void AddExistingString(string s)
{
    ExistingStrings.Add(s);
}

#region misc tools

private int GetRandomInt()
{
    byte[] buffer = new byte[2]; // 16
        bit = 2^16 = 65536
    Random.GetNonZeroBytes(buffer);
    int index = BitConverter.ToInt16
(buffer, 0);
    if (index < 0)
        index = -index; //handle de
        numeros negativos
    return index;
}

private char GetRandomCharFromArray
(char[] array, List<char> existentItems)
{
    char Character = ' ';
    do
    {
        Character = array[GetRandomInt
            () % array.Length];
    } while (!RepeatCharacters &&
        existentItems.Contains(Character));
    return Character;
}

#endregion misc tools

#region internal state

private bool m_UseUpperCaseCharacters,
m_UseLowerCaseCharacters, m_UseNumericCharacters,
m_UseSpecialCharacters;
private int m_MinUpperCaseCharacters,
m_MinLowerCaseCharacters, m_MinNumericCharacters,
m_MinSpecialCharacters;
private bool PatternDriven;
private char[]
    CurrentUpperCaseCharacters;
private char[]
    CurrentLowerCaseCharacters;
private char[]
    CurrentNumericCharacters;
private char[]
    CurrentSpecialCharacters;
private char[]
    CurrentGeneralCharacters;

```

```

private RNGCryptoServiceProvider
    Random;
private List<string> ExistingStrings;

#endregion internal state
}
}
}

```

E uma class que faça a encriptação propriamente dita:

```

using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace safeKeys
{
    public class locksmith
    {
        private const int keysize = 256;
        private string mkey = "";

        public locksmith(string masterkey)
        {
            if (masterkey.Length < 16 ||
                masterkey.Length > 16)
            {
                safeKeys.
                RandomStringGenerator.RandomStringGenerator
                RSG = new safeKeys.
                RandomStringGenerator.RandomStringGenerator();
                masterkey = RSG.Generate(6);
                //masterkey = "EfK7yhF3HKywfVXp";
            }
            else
            {
                this.mkey = masterkey;
            }

            //summary
            //metodo Encrypt
            //exemplo: string cifrado = Encrypt
            (texto_a_cifrar, key)
            //key é uma contra-senha que tanto pode
            ser uma constante como um valor de uma variável
            //devolve uma string correspondente ao
            texto a cifrar, cifrado recorrendo ao standard
            AES metodologia Rijndael
            //public static string Encrypt(string
            plainText, string passPhrase)
            public string Encrypt(string plainText,
                string passPhrase)
            {
                byte[] initVectorBytes =
                    Encoding.UTF8.GetBytes(mkey);
                byte[] plainTextBytes =
                    Encoding.UTF8.GetBytes(plainText);
                PasswordDeriveBytes password = new
                PasswordDeriveBytes(passPhrase, null);
                byte[] keyBytes = password.GetBytes
                    (keysize / 8);
                RijndaelManaged symmetricKey = new
                RijndaelManaged();
                symmetricKey.Mode = CipherMode.CBC;
                ICryptoTransform encryptor =
                symmetricKey.CreateEncryptor(keyBytes,
                    initVectorBytes);
                MemoryStream memoryStream = new
                MemoryStream();
                CryptoStream cryptoStream = new
                CryptoStream(memoryStream, encryptor,
                CryptoStreamMode.Write);
                cryptoStream.Write(plainTextBytes, 0,

```

```

        plainTextBytes.Length);
    cryptoStream.FlushFinalBlock();
    byte[] cipherTextBytes =
        memoryStream.ToArray();
    memoryStream.Close();
    cryptoStream.Close();
    return Convert.ToBase64String
        (cipherTextBytes);
}

//summary
//metodo Dncrypt
//exemplo: string cifrado = Encrypt
//          (texto_cifrado, key)
//key é uma contra-senha que tanto pode ser
//uma constante como um valor de uma variável
//devolve uma string correspondente ao
//texto a cifrar, cifrado recorrendo ao standard
//AES algoritmo Rijndael
//public static string Decrypt(string
//    cipherText, string passPhrase)
public string Decrypt(string cipherText,
    string passPhrase)
{
    //byte[] initVectorBytes =
    //    Encoding.ASCII.GetBytes(initVector);
    byte[] initVectorBytes =
        Encoding.ASCII.GetBytes(mkey);
    byte[] cipherTextBytes =
        Convert.FromBase64String(cipherText);
    PasswordDeriveBytes password = new
        PasswordDeriveBytes(passPhrase, null);
    byte[] keyBytes = password.GetBytes
        (keysize / 8);
    RijndaelManaged symmetricKey = new
        RijndaelManaged();
    symmetricKey.Mode = CipherMode.CBC;
    ICryptoTransform decryptor =
        symmetricKey.CreateDecryptor(keyBytes,
            initVectorBytes);
    MemoryStream memoryStream = new
        MemoryStream(cipherTextBytes);
    CryptoStream cryptoStream = new
        CryptoStream(memoryStream, decryptor,
            CryptoStreamMode.Read);
    byte[] plainTextBytes = new byte
        [cipherTextBytes.Length];
    int decryptedByteCount =
        cryptoStream.Read(plainTextBytes, 0,
            plainTextBytes.Length);
    memoryStream.Close();
    cryptoStream.Close();
    return Encoding.UTF8.GetString
        (plainTextBytes, 0, decryptedByteCount);
}
}
}

```

E por fim utilizar as duas classes, combinando-as para encriptar dados conforme o propósito original:

```

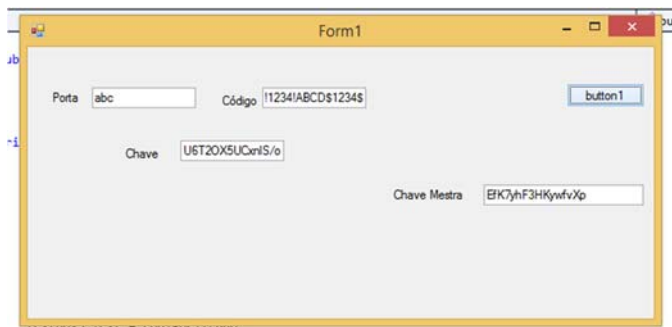
private void button1_Click(object sender, EventArgs
    e)
{

```

```

string masterkey=textBox4.Text.ToString();
string porta = textBox1.Text.ToString();
string codigo = textBox2.Text.ToString();
    locksmith lk = new locksmith
        (masterkey);
string codigoCifrado = lk.Encrypt
    (porta, codigo);
string codigoOriginal = lk.Decrypt
    (codigoCifrado, codigo);
textBox3.Text = codigoCifrado;
}

```



### Conclusão

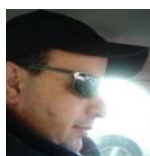
No exemplo consideramos como identificador “Porta”, uma vez que seria tipo “uma porta de uma caixa forte do tio patinhas”, o código seria o código que abriria a porta, a chave seria a chave que seria utilizada para encriptar, e a chave-mestra, a chave “par” da chave usada para encriptar. Caso seja omitida será gerada uma aleatoriamente.

Existem, tal como referi imensas formas mais simples de fazer isto, mas neste caso, optei por utilizar um gerador de strings aleatório, para gerarem as “chaves” mestras, uma vez que no caso concreto tinha de cifrar uns milhares de strings, todas com senhas diferentes, e uma classe simples para gerar encriptar e desencriptar a string que se pretende cifrar.

No exemplo está feito apenas para uma string, mas o leitor facilmente pode utilizar um loop e encriptar centenas de strings automaticamente.



## AUTOR



Escrito por **António C. Santos**

Programar é criar um novo mundo escrevendo código, cumprindo os mais elementares desígnios da vida, “aprender, ensinar, criar, partilhar, melhorar e seguir”. Formou-se no Instituto Politécnico de Viana do Castelo. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário desde 2012, actualmente exerce funções de mentor voluntário na plataforma MOOC Coursera . Twitter: [@apocsantos](#)



## A elegância da simplicidade

“UNIX is very simple, it just needs a genius to understand its simplicity.”

Começar um artigo por uma citação não será propriamente ortodoxo, não obstante, sendo uma coluna de opinião, de alguém que “não se importa”, não creio que seja “grave”!

Posto isto, vamos ao que realmente interessa! A elegância da simplicidade, pode parecer um assunto pouco interessante, mas na verdade tem muito que se lhe diga! A título introdutivo, gostaria de relembrar os tempos dos microprocessadores de 8bits, extremamente simples, como o caso dos 8080, Zilog Z-80, Motorola 68000, etc... Nessa altura, a simplicidade tanto de hardware como software era no mínimo elegante, para não dizer “bela”, uma vez que muito era feito, com muitíssimo pouco! O próprio código escrito para essas plataformas era simples, compacto, eficiente e eficaz, super “económico” em termos de recursos de memória ram e processamento, além de “simples” de entender, tanto para o comum dos mortais, como para os mais “tecnologicamente ávidos”. Por esta altura a “simplicidade” era tanta, que a beleza do código, bem como do hardware, residia nela. A beleza, a elegância e o motivo de orgulho, de quem produzia software, hardware, componentes, etc... residia na simplicidade do que se fazia! Simplesmente elegante, eficiente, preformante, eficaz!

Feita esta breve introdução e passando mais concretamente ao actual, em que tudo, ou quase tudo é “complexo”, deselegante, “feito”, quase imperceptível e rebuscado, muitas vezes repara-se que na ânsia de “produzir mais”, se deixou de medir qualidade em simplicidade, para se medir em número de linhas de código de um software, ou em número de transístor de um circuito, tantas vezes perdendo desempenho, em favor do facilitismo, criando novos problemas, novas dificuldades, e tornando algumas, realmente difíceis de detectar e de corrigir. Apenas a título de exemplo as duas vulnerabilidades conhecidas por MeltDown e Spectre, que passaram anos, despercebidas, tanto nos microprocessadores, como no código do kernel dos sistemas operativos, que dada a sua “extrema deselegância”, tornou simples a passagem dessas falhas despercebidas e complexa a correcção das mesmas.

Esta “complexidade” e esta nova mania de “medir qualidade em número de linhas”, leva a um “comportamento esquisito”, de quem quer medir tudo em “números” não apreciando a beleza intrínseca da simplicidade! Ou é complexo, ou é feio!

Não precisa de ser elegante, basta ser complexo, não precisa de ser eficiente, basta ser complexo, basta ter um número grande, não importa se é um número grande de linhas, de transístors, ou de indicador de desempenho! Desde que seja “grande”, é que é bom, primando pela quantidade e não pela qualidade. Algo que a meu humilde ver, é no mínimo absurdo, para não dizer errado, estranho, ilógico.

E aqui, voltamos à citação com que inicio o artigo “(...) it just needs a genius to understand its simplicity(...)”, porque na realidade muitas vezes é necessário “um génio”, para entender que complexidade não significa qualidade, e tantas mais é preciso um “génio”, para apreciar a elegância da simplicidade! Ok, talvez não um “génio”, mas uma mente “aberta” o suficiente para observar a simplicidade, apreciar a simplicidade, deixar-se “acreditar”, na simplicidade diante dos seus olhos!

Não quero de forma nenhuma, dizer que a “tecnologia antiga” é que é boa, até porque acredito que muita coisa nova, pode ser feita de forma simples e elegante! Quero com isto dizer que abdicar da elegância e da simplicidade em prol de “meros números”, não se traduz em qualidade, tal como o exemplo dado! Além disto, nem sempre se traduz em evolução, mas tantas vezes em “estagnação”, porque a necessidade, sempre foi o motor que aguça o engenho e a ausência de necessidade, é como areia na engrenagem, entorpece o engenho, passando de uma era em que se tem orgulho no código que se escreve, para uma era do “SOCP” (Stack Overflow Copy Past), onde não se cria nada de realmente novo e brilhante, mas se copia o que outros fizeram, sem ter em conta o desempenho e a “elegância”.

Termino citando um excerto de um post de usenet, de 1991, que “iniciou uma revolução” : “*Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers?*” (Linus Torvalds), para lembrar que quer se seja “desse tempo” ou de outro tempo qualquer, todos podemos apreciar a elegância da simplicidade, a beleza intrínseca, do que sendo simples, é simultaneamente, belo, eficiente, eficaz, quase perfeito na sua “ingenuidade” e no seu engenho. Porque a qualidade de um projecto, não se mede em número de linhas de código!

### AUTOR



Escrito por **António C. Santos**

Programar é criar um novo mundo escrevendo código, cumprindo os mais elementares desígnios da vida, “aprender, ensinar, criar, partilhar, melhorar e seguir”. Formou-se no Instituto Politécnico de Viana do Castelo. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário desde 2012, actualmente exerce funções de mentor voluntário na plataforma MOOC Coursera . Twitter: [@apocsantos](https://twitter.com/apocsantos)



## SQL Curtas #3: Mais um conjunto de dúvidas

Aqui ficam mais algumas dúvidas que nos surgiram nas últimas semanas em encontros presenciais e nas redes sociais.

### O que é a "collation"?

Uma "collation" é a forma que cada SGBD (Sistema de Gestão de Bases de Dados) tem de "ordenar" e de "comparar" um conjunto de dados.

Há várias collations, desde as que são aplicáveis ao mundo ocidental, a outras mais "exóticas" que são usadas em países orientais (dessas, as regras são demasiado complicadas para nós ocidentais compreendermos).

Vamos começar com o exemplo das palavras {"Ola"; "Olá"; "OLÁ"}. Uma collation é um conjunto de uma ou mais opções em baixo (apenas listadas as relevantes para línguas ocidentais):

- Binária: Cada carácter diferente tem um valor binário diferente e em comparações são tratados como diferentes (em Unicode: "a"=U+0061; "à"=U+00E1; "À"=U+00C1). Normalmente utiliza-se o código Unicode (ou ASCII, ou de 8bit) para ordenar os dados, mas depende do SGBD. Podem também haver várias collations binárias (cada uma com as suas particularidades de ordenação ou comparação);
- Character Set: Os caracteres que estão disponíveis para serem utilizados. Se for Unicode (UCS16, UTF8, etc.), então todos os caracteres estão disponíveis. Caso contrário, poderá ser um de vários character sets de 8bits (ISO Latin1, etc.);
- Case Sensitive (CS): Em comparações, um carácter maiúsculo é diferente do mesmo carácter minúsculo ("A" != "a"). Em ordenações, o maiúsculo fica normalmente primeiro, mas depende do SGBD;
- Case Insensitive (CI): Em comparações, um carácter maiúsculo é igual ao mesmo carácter minúsculo ("A" = "a"). Em ordenações, têm o mesmo valor e são ordenados como se fossem iguais (se "A" fica antes ou depois de "a" depende apenas de outras partes da ordenação, não deste carácter);
- Accent Sensitive (AS): Em comparações, um carácter sem acento é diferente do mesmo carácter com acento ("a" != "À"). Em ordenações, o carácter sem acento fica normalmente primeiro;
- Accent Insensitive (AI): Em comparações, um carácter sem acento é igual ao mesmo carácter com acento ("a" = "À"). Em ordenações, têm o mesmo valor e são ordenados como se fossem iguais (se "a" fica antes ou de-

pois de "À" depende apenas de outras partes da ordenação, não deste carácter);

- E também outras regras de ordenação...

Tipicamente, usamos as seguintes collations:

- Case Insensitive, Accent Sensitive (Unicode ou Latin1): Esta é a collation por omissão em alguns SGBD (como o SQL Server). Infelizmente, não é prática para línguas ocidentais para além do Inglês, já que não trata correctamente os acentos;
- Case Insensitive, Accent Insensitive (Unicode ou Latin1): Esta é a ideal para a Língua Portuguesa. Ao instalar um SGBD ou a criar uma base de dados, é recomendado usá-la;
- Binária: Pouco usada, apenas útil se guardarmos dados pesquisáveis binariamente, como imagens e outras files.

Aplicação de collations, começando com os dados introduzidos por uma certa ordem (a mesma cor na collation significa que são equivalentes em comparação):

Introduzido	Collation Binária	Collation CS AS	Collation CI AS	Collation CI AI
Olá	OLA	OLA	Ola	Olá
Ola	OLE	OLE	OLA	Ola
Olé	Ola	Ola	OLE	OLA
OLE	Ole	Ole	Ole	Olé
OLA	Olá	Olá	Olá	OLE
Ole	Olé	Olé	Olé	Ole

Mas onde podemos utilizar collations? Na realidade depende do SGBD. No caso do SQL Server, configuramos uma collation na instância, que é herdada por cada base de dados. A collation de cada base de dados é herdada por cada tabela, e por sua vez herdada por cada campo da tabela, índice da tabela ou campo no índice da tabela (todas estas heranças podem ser alteradas).

O típico em CMS (Content Management Systems) mais antigos, que normalmente usam MySQL ou MariaDB, e cuja collation é por exemplo Unicode\_CI\_AI, ter o campo de "Password" (infelizmente em plain text) com uma collation Unicode\_CS\_AS ou mesmo Binary, para garantir que "PassWord" e "password" são diferentes quando se faz um



WHERE. Nota: isto é obviamente uma péssima forma de garantir segurança (deve ser usada uma hash como MD-5, SHA-256 ou melhor, com salting).

Podemos também forçar a comparação com uma collation específica, mas aqui há sempre uma perda de desempenho. Para resolver este problema, podemos criar índices, cada um com uma das collations que precisamos. Pode depois ser pedida expressamente a collation a usar (exemplo de SQL Server):

```
SELECT * FROM Tabela1 AS T1 LEFT JOIN
    Tabela2 AS T2 ON T1.CampoX COLLATE
    SQL_Latin1_General_CP1_CS_AS = T1.CampoY;
```

O que são UNION e UNION ALL?

Começamos com um exemplo com UNION ALL:

```
SELECT T1.Campo1,
    T1.Campo2,
    T1.Campo3
FROM Tabela1 AS T1

UNION ALL

SELECT T2.CampoX,
    T2.CampoY,
    T2.CampoZ
FROM Tabela2 AS T2;
```

O resultado vai ser um dataset com 3 campos, cujo nome depende apenas do primeiro query no UNION ALL (i.e. "Campo1", "Campo2" e "Campo3").

O que é necessário garantir? O número de campos de cada query tem que ser igual e têm que ser compatíveis (conversão explícita ou implícita para o data type de cada campo do primeiro query). Por exemplo, se T1.Campo1 for VARCHAR(10) e T2.CampoX for NVARCHAR(100), vai ser convertido para VARCHAR(10). Dependendo do SGBD, poderão perder precisão (ficam só com os primeiros 10 caracteres), ou é gerado um erro. O mesmo para números e outros data types.

Posso ter mais do que um UNION ou UNION ALL? Sim, é só acrescentar no final do código, embora eles sejam normalmente agrupados aos partes (e.g. T1 junta com T2 e o resultado T1+T2 junta com T3, e assim sucessivamente).

Qual a diferença entre UNION e UNION ALL? A sintaxe é semelhante, e os requisitos em cima também. UNION ALL junta literalmente todos os registos do primeiro query aos do segundo (e terceiro, etc.), mesmo que sejam repetidos. UNION junta apenas os registos que sejam DISTINCT, ou seja, que cada registo na sua totalidade (todos os campos juntos) seja único. Por esse motivo, o UNION obriga a um consumo grande de memória e tempo, já que tem que ordenar todos os registos antes de os disponibilizar para eliminar duplicados (a ordenação é um efeito secundário que pode ou não ser útil). UNION ALL é portanto mais rápido e recomendado na maioria dos casos (e não faz reordenação).

“ Uma "collation" é a forma que cada SGBD (Sistema de Gestão de Bases de Dados) tem de "ordenar" e de "comparar" um conjunto de dados. Há várias collations, desde as que são aplicáveis ao mundo ocidental, a outras mais "exóticas" que são usadas em países orientais (dessas, as regras são demasiado complicadas para nós ocidentais compreendermos). Vamos começar com o exemplo das palavras {"Ola"; "Olá"; "OLÁ"}. Uma collation é um conjunto de uma ou mais opções em baixo (apenas listadas as relevantes para línguas ocidentais) ”

# SQL Curtas

## SQL CURTAS #3: MAIS UM CONJUNTO DE DÚVIDAS

### "Eu ODEIO cláusulas WHERE com OR #ProntoFalei"

Bem, isto tem uma história. Recentemente o nosso guru amigo Fabiano Amorim do outro lado do Atlântico desabafou este comentário no Facebook. E claro, estava a falar dos problemas de optimização (entre outros) que dão os WHEREs que incluem ORs.

Mas vamos levar isto ao extremo de dizer que não queremos ORs em queries. Que alternativas temos?

Começamos com um exemplo original com OR (usamos @variável para os valores possíveis do OR, e apenas um "\*" no SELECT para simplificar):

```
SELECT *
FROM Tabela1 AS T1
WHERE (T1.Campo1 = @x OR T1.Campo2 = @y OR
       T1.Campo3 = @z);
```

Uma opção é usar as Leis de De Morgan e trocar o OR por ANDs e NOTs (exactamente equivalente):

```
SELECT *
FROM Tabela1 AS T1
WHERE NOT (T1.Campo1 <> @x AND T1.Campo2
          <> @y AND T1.Campo3 <> @z);
```

Outra opção passa por usar UNION ou UNION ALL (em alguns casos, apenas UNION será equivalente, ver explicação em cima):

```
SELECT *
FROM Tabela1 AS T1
WHERE (T1.Campo1 = @x)

UNION ALL

SELECT *
FROM Tabela1 AS T1
WHERE (T1.Campo2 = @y)

UNION ALL

SELECT *
FROM Tabela1 AS T1
WHERE (T1.Campo3 = @z);
```

Caso todas as parcelas do OR se apliquem ao mesmo campo (e.g. T1.Campo1), podemos também usar:

```
SELECT *
FROM Tabela1 AS T1
WHERE T1.Campo1 IN (@x, @y, @z);
```

Enfim, podíamos inventar muitas outras formas alternativas de escrever estes queries...

“ **"Eu ODEIO cláusulas WHERE com OR #ProntoFalei"**

**Bem, isto tem uma história. Recentemente o nosso guru amigo Fabiano Amorim do outro lado do Atlântico desabafou este comentário no Facebook. E claro, estava a falar dos problemas de optimização (entre outros) que dão os WHEREs que incluem ORs.** ”

## AUTOR



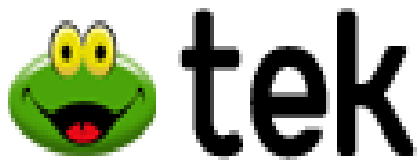
Escrito por **André Melancia**

Independent Developer/DBA/Consultant. Microsoft Certified Trainer (MCT) focusing on SQL Server, Azure and IoT. 17+ years' fun developing information and multimedia systems, DBA, project and IT management. PowerShell Portugal, IT Pro Portugal and IoT Portugal communities organiser. IPv6 Portugal, DNSSec Portugal and Windows Development Portugal online communities moderator. Actively volunteering, organising, speaking or just participating at community meetings and events like SQLSaturdays, SQLBits, SQLRelay, Maker Faire Lisbon, Arduino/Genuino Day Lisbon, Global Azure Bootcamp Lisbon, etc. Proud uncle and food devouring expert, with dangerous pussy cat as companion.

Go to <http://Andy.PT> and you'll know the same as the NSA...



# Media Partners da Revista PROGRAMAR



# Análises

**Business Intelligence no SQL Server**

**Segurança Prática em Sistemas e Redes Com Linux**

## Business Intelligence no SQL Server

**Título:** Business Intelligence no SQL Server

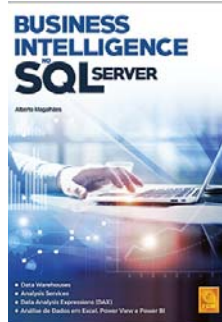
**Autores:** Alberto Magalhães

**Editora:** FCA - Editora de Informática

**Páginas:** 208

**ISBN:** 978-972-722-869-0

**Formato:** soft-cover



Nesta edição trazemos até vós caros leitores a review acerca do novo livro da FCA do autor Alberto Magalhães – Business Inteligente no SQL Server.

Na sua essência as bases de dados são simplesmente contentores de informação. Quando um determinado sistema permanece durante anos sem grandes alterações, todos nós sabemos que acrescentar mais uma feature ou outra facilmente implica quase sempre a adição de mais uma tabelinha para facilmente termos dados “à mão”, tornando-se mais uns “pontos” para o “contentor” guardar. E hoje em dia é cada vez mais importante, usarmos os dados que armazenamos na BD para tomar decisões acerca dos caminhos a seguir e a implementar. Seja a nível de negócio, seja a nível de vida útil do próprio sistema.

Este é um livro indicado principalmente para os leitores que queiram dar os primeiros passos em BI mas também pode ser um apoio útil para quem já tem mais experiência neste campo e pretende alargar conhecimentos.

A obra está dividida em cinco grandes capítulos, que fluem entre si não tornando a leitura deste livro tecnicamente pesada.

No primeiro capítulo - “**Data Warehouse**” – encontramos uma explicação teórica sobre o tema. “*DW consiste em integrar, de forma consolidada, os dados de uma organização num só ponto, e, assim, fornecer dados atualizados aos decisores das empresas*”. E é este o mote principal a que o autor se propõe neste livro. É abordada a arquitetura de um DW, com maior incidência na arquitetura de Kimball e na arquitetura de Inmon. Neste primeiro capítulo gostaria de chamar à atenção para o ponto em que o autor compara concretamente estas duas arquiteturas DW, para que este tema seja facilmente absorvido pelos leitores menos experientes de forma a poderem tirar um maior partido do livro. Neste capítulo são ainda referenciados modelos de dados nomeadamente o esquema em estrela e o esquema em floco de neve, assim como as diferenças entre eles.

O segundo capítulo – “**Analysis Services**” – apresenta-nos a explicação teórica do tema mas passa já a ser um capítulo mais técnico onde o autor nos guia pelas ferramentas disponíveis para o desenvolvimento e suporte das aplica-

ções business intelligence: o **SQL Service Reporting Services**, **SQL Server Integration Services** e o **SQL Server Analysis Services**. É apresentada uma pequena introdução às bases de dados OLAP, isto é, **Online Analytical Processing**. Nesta capítulo o autor demonstra ainda como criar a fonte de dados e quais os passos para a visualização da mesma. É ainda abordado como criar as dimensões e o “cubo”.

“ Na sua essência as bases de dados são simplesmente contentores de informação. Quando um determinado sistema permanece durante anos sem grandes alterações, todos nós sabemos que acrescentar mais uma feature ou outra facilmente implica quase sempre a adição de mais uma tabelinha para facilmente termos dados “à mão”, tornando-se mais uns “pontos” para o “contentor” (...) ”



Sendo este ultimo constituído pelas dimensões e pelas métricas, julgo que este é um capítulo bastante importante para os menos experientes pois este passo-a-passo, ajuda na compreensão e consolidação da temática, uma vez que é através destes conhecimentos que conseguiremos dar “corpo” aos nossos dados.

No terceiro capítulo – “Análises de Dados” – Alberto Magalhães leva-nos por alguns exemplos de pesquisa pelos modelos anteriormente implementados. Uma das ferramentas externas abordadas é o famoso excel, onde facilmente obtemos gráficos dos dados que considerámos sensíveis. Neste capítulo gostaria de destacar os subtemas “*Importar dados para uma tabela dinâmica do Excel*”, “*Importar dados para um relatório Power View*” e “*Explorar dados com o SQL Server Management Studio*”. É importante referir também que é neste capítulo que o autor aborda a “*Análise de Dados em Power BI Desktop*” – uma ferramenta de negócios da Microsoft que permitem facilmente a criação de relatórios e dashboards sem que sejam necessários conhecimentos técnicos extensos.

“*E hoje em dia é cada vez mais importante, usarmos os dados que armazenamos na BD para tomar decisões acerca dos caminhos a seguir e a implementar*”

No quarto capítulo – “*Integration Services*” – é-nos apresentada esta plataforma de extração, transformação e carregamento de dados. Como não podia deixar de ser, este é dos capítulos mais técnicos deste livro, contudo a leitura não é pesada e flui facilmente, permitindo ao leitor criar o

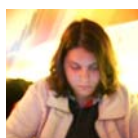
seu próprio projeto seguindo as indicações passo-a-passo. São também abordados os componentes de **Control Flow** e **Data Flow**, assim como os **event handlers**. Recomenda-se uma leitura atenta a este capítulo de forma a compreender as práticas da gestão de pacotes do **SQL Server Integration Services**.

“*Análise de Dados em Power BI Desktop*” – uma ferramenta de negócios da Microsoft que permitem facilmente a criação de relatórios e dashboards(...)

Por fim entramos no quinto capítulo – “*Reporting Services*” – aqui é-nos apresentada com mais pormenor esta ferramenta empresarial do SQL Server. É explicado como criar e gerar relatórios que possam ser uma mais valia para o nosso modelo de negócios de forma a facilitar a tomada de decisões. Neste capítulo gostaria de destacar a criação de um relatório simples, assim como a abordagem ao **Tablix** (conjugação de tabela, matrix e lista), assim como a abordagem à região de dados **Gauge** onde podemos facilmente obter os **KPI - Key Performance Indicators** (que medem o desempenho de um processo).

Em suma, este é um livro que creio que seja uma mais valia na biblioteca pessoal de cada um, quer sejamos estreatantes nestas andanças ou não, uma vez que apesar de iniciar com uma abordagem mais leve e teórica nos capítulos iniciais, o quarto e quinto capítulo, entram já num “mundo mais técnico” que vale a pena ler e reler. Boa leitura!

## AUTOR



Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010. Embaixadora das Geek Girls Portugal – Núcleo de Lisboa.



## Segurança Prática em Sistemas e Redes Com Linux

**Título:** Segurança Prática em Sistemas e Redes Com Linux

**Autores:** Jorge Granjal

**Editora:** FCA

**Páginas:** 248

**ISBN:** 978-972-722-865-2

**Formato:** Capa soft



Um livro destinado a professores, estudantes, profissionais e interessados nesta temática, mais actual que nunca, aborda de forma concisa e prática os conceitos teóricos subjacentes ao tema, de forma bastante sintética, mas sem perder a essência do que é mais relevante, para quem se “aventura” nesta temática, bem como para quem pretende consultar informação.

Ao longo de onze capítulos, são abordados temas como encriptação, recorrendo a PGP (Pretty Good Privacy), certificados digitais (x.509), acessos remotos utilizado SSH, evoluindo para temáticas mais abrangentes como segurança em servidores http, protecção de servidores, recorrendo a mecanismos e firewall, protecção de redes, segurança em redes sem fios, redes privadas virtuais (vpn), detecção e prevenção de intrusões, terminando na temática da auditoria, que apesar de abordada e forma breve, (em cerca de 30 páginas), é abordada, tendo em conta a delicadeza do tema.

No primeiro capítulo, é feita uma apresentação mais teórica dos conceitos básicos, dos quais se destacam a confidencialidade, integridade, autenticação, não-repúdio, sem descorar a disponibilidade e resiliência, factores que cada vez mais são críticos, como o recente exemplo do e-factura, que dado o volume de utilizadores n um curto espaço de tempo, acabou ficando involuntariamente indisponível, por, segundo a comunicação social, ter excedido a capacidade de resposta, ao volume de pedidos.

Ainda neste mesmo capítulo, são focados aspectos como os diversos tipos de atacantes “externos e internos”, algo que é realmente uma ameaça, dada a quantidade de meios e recursos, mais ou menos tecnológicos disponíveis, que em tese, podem ser utilizados para “atacar” um sistema. De salientar que o autor é cuidadoso na forma como apresenta o tema, deixando espaço para o leitor o aprofundar, indicando referências que pode ser consultadas, no final de cada capítulo.

As configurações e instalação de software dos exemplos, são feitas para sistema operativo GNU/Linux, distribuição CentOS, o que apesar de poder não ser o “preferido” de todos os leitores, é simples de instalar e bastante comum em ambientes de produção. A titulo de nota pessoal, prefiro distribuições “debian based”, no entanto os exemplos são simples de executar nessas distribuições.

Na temática das redes privadas virtuais (vpn’s), cada vez mais actuais, uma vez que a quantidade de vectores de ataque aumenta, e cada vez se torna mais importante estar protegido, o autor aborda todos os conceitos teóricos de forma clara, simples, e eficaz, de maneira a que o leitor, possa entender o funcionamento e aplicabilidade, de forma compreensível e prática, passando para os exemplos com OpenVPN, tanto servidor de vpn, como cliente de vpn, para os principais sistemas operativos, de forma igualmente prática e eficiente. Existem outros sistemas gratuitos que podem ser de maior agrado do leitor, no entanto, convém ressaltar que o OpenVPN se tornou num “standard de facto”, no que toca a redes privadas virtuais, pelo que a opção por este sistema, apresenta-se como a melhor de todas, tanto para quem lê, como para quem escreve, e para quem irá montar uma vpn.

Quase no final do livro, ao abordar a temática da auditoria de segurança, o autor começa por explicar novamente os conceitos subjacentes, passando para exemplos práticos de auditoria de segurança, recorrendo a ferramentas que integram os próprios sistemas operativos, evitando inicialmente o uso de “ferramentas chave-na-mão”, que apesar de simples de utilizar para os mais iniciantes, não se traduzem uma aquisição de conhecimento nem na aplicabilidade do mesmo. Citando o velho ditado “aprende a andar, antes de correr”, convém que o leitor entenda primeiro os conceitos, antes de começar a fazer auditorias, sem entender o que lhes está subjacente. Ao longo do artigo, são abordadas ferramentas, mais elaboradas, e já com interface gráfica de utilizador, complementares às mencionadas inicialmente, que permitem realizar auditorias mais detalhadas e rápidas, bem como analisar outros vectores, para os quais são convenientes ferramentas específicas.

Gostaria de deixar os meus parabéns ao autor, por uma obra de muito agradável leitura, actual e rica em conteúdo, extremamente útil, quer para quem se inicia, como para quem já trabalha com esta temática. Um livro recomendável, a todos quantos por segurança se interessam.

### AUTOR



**Escrito por António C. Santos**

Programar é criar um novo mundo escrevendo código, cumprindo os mais elementares desígnios da vida, “aprender, ensinar, criar, partilhar, melhorar e seguir”. Formou-se no Instituto Politécnico de Viana do Castelo. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário desde 2012, actualmente exerce funções de mentor voluntário na plataforma MOOC Coursera . Twitter: [@apocsantos](https://twitter.com/apocsantos)



# Segurança

**Vulnerabilidades Web em 2017**

**RGPD – O Antes e o Depois**

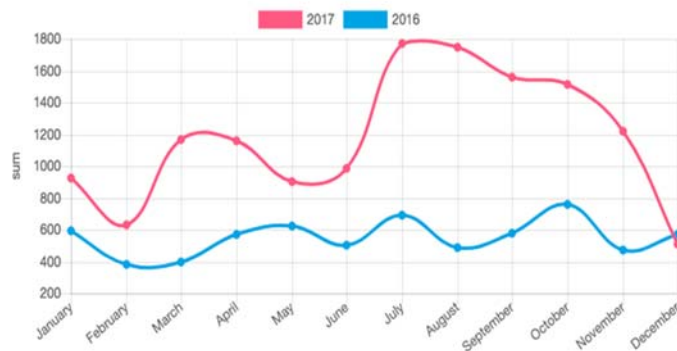
**SoftEther VPN Project, implementação em Windows, GNU/Linux, MacOS, Android, iOS**

## Vulnerabilidades Web em 2017

Foi registado um número recorde de vulnerabilidades em 2017 nas aplicações web, incluindo categorias bem conhecidas, como xss, mas também novas categorias, como deserialização insegura. O número de vulnerabilidades na IoT também cresceu descontroladamente e com um impacto extremamente severo. Também o PHP e o WordPress continuam de mãos dadas no que toca a vulnerabilidades em CMS e server-side. Por fim, uma vulnerabilidade de dia zero (zero day vulnerability) afetou o Apache Struts e que contou com uma das maiores violações de segurança do ano.

### Vulnerabilidades Web 2016-2017

A seguinte imagem mostra o crescimento das vulnerabilidades web na forma mensal, entre o ano de 2016 e 2017. É possível observar um crescimento em 2017 (14,082), nomeadamente 212%, relativamente ao ano de 2016 (6,615). Estima-se ainda que, mais de 50% das vulnerabilidades possuem um exploit público para a sua exploração por parte de indivíduos mal intencionados. Além disso, mais de um terço (30%) não possuem uma resolução imediata, como por exemplo, uma actualização ou um remendo de software. Como é previsível, a injeção de código mantém-se no top de vulnerabilidades.



### OWASP Top 10 de Vulnerabilidades

Depois de 2013, a OWASP concretizou mais um relatório relativo às vulnerabilidades web. O que mudou desde 2013 até 2017?

OWASP Top 10 - 2013	OWASP Top 10 - 2017
A1 – Injection	A1:2017-Injection
A2 – Broken Authentication and Session Management	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Existem duas novas entradas na tabela, nomeadamente:

- Deserialização Insegura; e
- Monitorização e Logging Insuficiente.

### Des-serialização Insegura

Serialização é o processo de traduzir estruturas de dados ou o estado do objeto em um formato que possa ser armazenado (por exemplo, em um arquivo ou buffer de memória) ou transmitido (por exemplo, através de um link de conexão de rede) e reconstruído posteriormente (deserialização). Serialização é amplamente utilizada em RPC, HTTP, bases de dados, etc.

Aplicações e APIs podem estar vulneráveis se deserializarem objetos hostis ou adulterados fornecidos por um atacante sem sanitização apropriada.

A8
Insecure Deserialization
14

Threat Agents

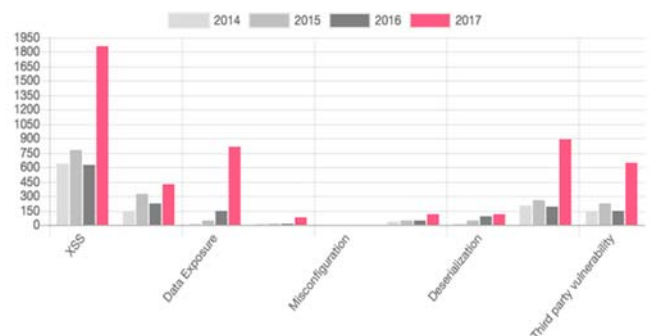
Attack Vectors

Security Weakness

Impacts

App. Specific	Exploitability: 1	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Exploitation of deserialization is somewhat difficult, as off the shelf exploits rarely work without changes or tweaks to the underlying exploit code.					
This issue is included in the Top 10 based on an <a href="#">Industry Survey</a> and not on quantifiable data.					
Some tools can discover deserialization flaws, but human assistance is frequently needed to validate the problem. It is expected that prevalence data for deserialization flaws will increase as tooling is developed to help identify and address it.					
The impact of deserialization flaws cannot be understated. These flaws can lead to remote code execution attacks, one of the most serious attacks possible.					
The business impact depends on the protection needs of the application and data.					

O gráfico de vulnerabilidades do TOP 10 de 2017 é o seguinte:



Legenda: (XSS; Authentication; Data Expousure; Access Control; Misconfiguration; XXE; Deserialization; Injection; Third party vulnerability)

A des-serialização contou com um aumento diminuto, enquanto que o XSS, Data Expousure e a injeção aumentaram de forma significativa. Este gráfico é a prova que existe uma necessidade imediata para combater os problemas de segurança em 2018!

# Segurança

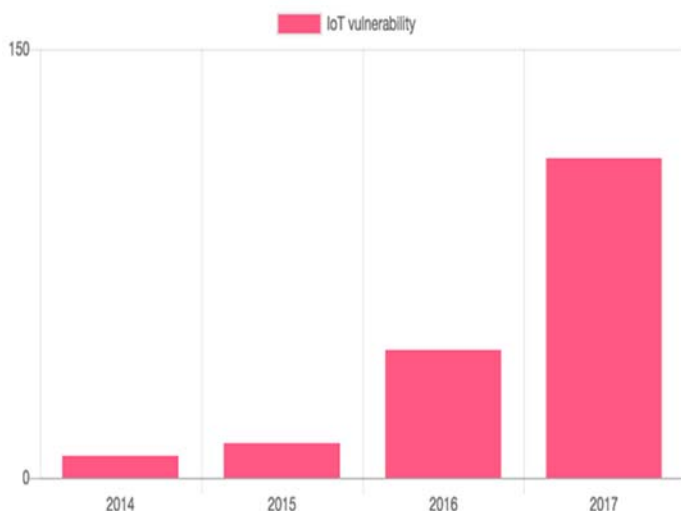
## VULNERABILIDADES WEB EM 2017

### Monitorização e Logging Insuficiente

Os atacantes contam com a falta de monitorização e resposta atempada para alcançar seus objetivos sem serem detetados. Não foi detetada nenhuma vulnerabilidade publicada em 2017 que esteja diretamente relacionada a esta categoria. Será interessante monitorizá-la e ver se isso mudará em 2018.

### O crescimento da IoT

Actualmente, quase tudo na nossa vida tem um endereço de IP e está ligado a uma rede de computadores e à Internet, p.ex.: o nosso frigorífico, as luzes da sala, a smart TV, as portas de casa (casas inteligentes), fechaduras, e até mesmo a roupas que usamos no dia a dia. Como estes dispositivos foram projetados para receber e enviar pedidos (receber ordens e dar feedback), precisam de uma ligação à Internet. Por isso, constituem um recente horizonte de ataque! Em muitos casos, os fornecedores destes dispositivos denominados inteligentes negligenciam a segurança adequado ou mesmo a “backdoor” de forma propositada, com a finalidade de obter acesso oculto, para monitorizar o sistema, ou até para fins de suporte. Posto isto, não deixa de ser uma backdoor.

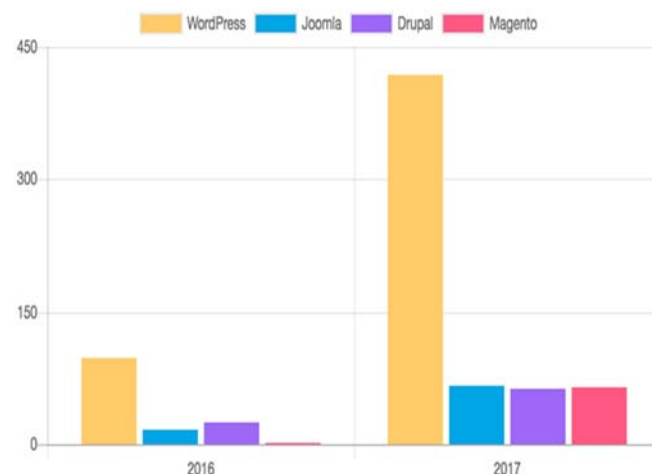


A prova do que a IoT começa a ser um horizonte de ataque fácil do ponto de vista do atacante é o gráfico acima, que regista um aumento de 104 vulnerabilidades em 2017 relativamente ao ano anterior. É indiscutível que, em 2018 o número de vulnerabilidades neste landscape continue a aumentar, pois é uma nova tendência, e o número de dispositivos ligados à rede não para de aumentar.

Uma das vulnerabilidades mais populares são as credenciais de acesso às páginas de configuração destes dispositivos (35%), que na maioria das vezes são as mesmas trazidas de fábrica, e o consumidor, ou por desconhecimento ou preguiça, não toma quaisquer medidas. Um dos maiores ataques conhecidos foi o Mirai, que usou estas vulnerabilidades para se espalhar na rede e causa um enorme ataque DDoS.

### Content Management System (CMS)

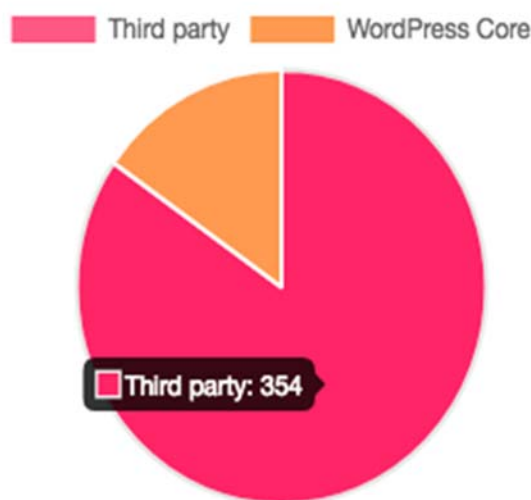
Em seguida é apresentado um gráfico com o volume de vulnerabilidades detetadas em 2017 relativamente aos CMS mais usados na indústria (60% do market share): WordPress, Joomla, Drupal e Magento.



### WordPress

Como era de suspeitar, o WordPress continua a bater records no que a vulnerabilidades diz respeito. Comparativamente a 2016, no ano de 2017 teve um aumento de 418 novas vulnerabilidades, um aumento de 400%.

De notar que, 75% das vulnerabilidades de 2017 dizem respeito a plugins de terceiros (third-party vendor plugins).

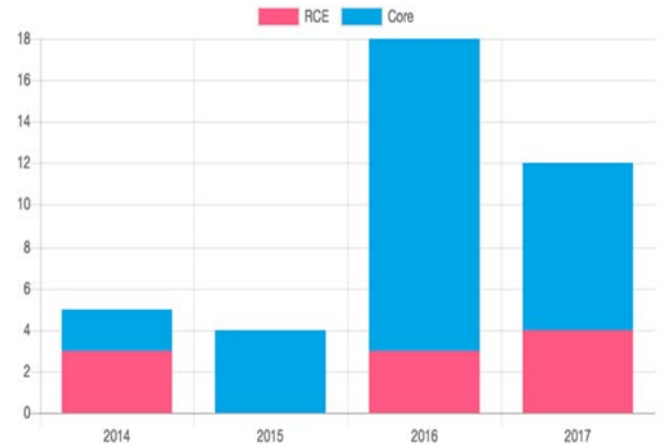


Este crescente no que ao número de vulnerabilidades diz respeito está diretamente relacionado com o número de plugins concebidos. Muitos deles com falhas de conceção, falta de testes de segurança, etc. De notar que o número de vulnerabilidades de uma peça de software está inteiramente ligada ao número de linhas de código (LoC – Lines of Code).



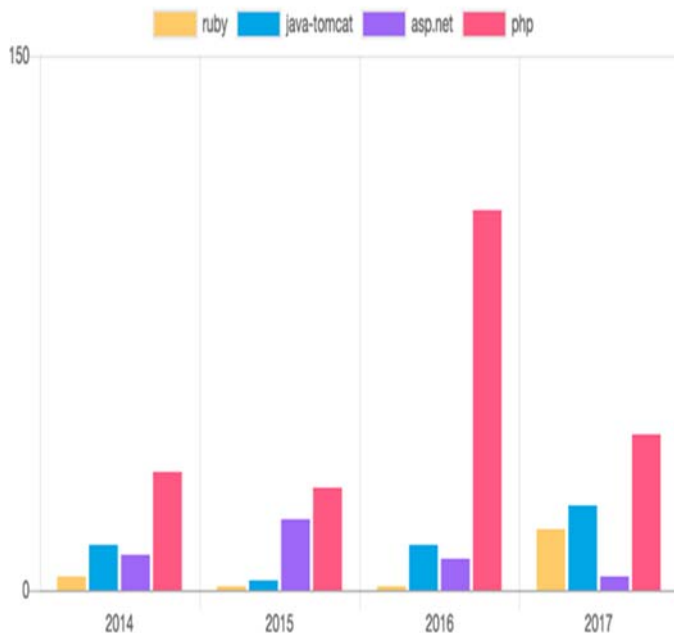
## VULNERABILIDADES WEB EM 2017

Year	Number of WordPress Plug-ins
2015	41,347
2016	48,044
2017	53,357



### Tecnologias Server-side

O PHP continua a ser a linguagem mais usada do lado do servidor em aplicações web. Notavelmente, o número de vulnerabilidades em 2017 diminuiu 143% relativamente a 2016. O grande número apresentado em 2016 pode ser explicado pelo seguinte: —em 2015 o PHP lançou a versão 7.0, e esteve quase 1 ano e meio sem actualizações. No ano passado (dezembro 2016) lançou a versão 7.1, com pequenas mudanças que podem explicar a diminuição drástica em 2017.



### Apache Struts

Embora 2017 tenha menos vulnerabilidades na estrutura da Apache Struts, o seu impacto foi enorme, pois alguns deles incluíram a execução de código remoto não autenticado (RCE), o que basicamente significa que qualquer um pode piratear e assumir o servidor, aceder informações privadas e muito mais.

### Previsões para 2018

Em seguida ficam algumas previsões sobre possíveis vulnerabilidades para 2018:

- XSS e injeção, continuarão a liderar devido ao aumento de tecnologias que continuam a utilizar tecnologias como Javascript do lado do servidor (Node.JS).
- IoT, também este horizonte irá crescer exponencialmente em 2018, devido ao aumento dos dispositivos ligados a Internet e à sua variedade (tudo tem um endereço de IP ...). Este alvo é muito apreciado no que a DDoS diz respeito.
- Também vulnerabilidades posicionadas na blockchain irão ser uma tendência em 2018. Já o foram em 2017, e estima-se que 2018 seja um alvo muito apetecível pelos atacantes. Este tópico está demasiado hyped, as empresas correm desesperadamente a implementar sistemas na cadeia porque acaba por ser “cool” e têm visibilidade na indústria, no entanto, são cometidas muitas falhas de segurança durante a conceção do software. Em 2017 existiram muitos desastres. Vejamos a seguir.

### Desastres e hacks na blockchain em 2017

#### 1. CoinDash ICO Hack

O ethereum address foi alterado por um hacker e o prejuízo foi de 7.3\$ milhões.

#### 2. Parity Wallet Breach

150,000 ethers foram roubados das carteiras dos utilizadores.

#### 3. Enigma Project Scam

Foi lançada uma token falsa em agosto, defraudando potenciais investidores em mais de 1.500 ethers.

#### 4. Parity Wallet Freeze

Um bug encontrado no código congelou mais de 275\$ milhões.

**“ Foi registado um número recorde de vulnerabilidades em 2017 nas aplicações web, incluindo categorias bem conhecidas, como XSS, mas também novas categorias, como deserialização insegura. O número de vulnerabilidades na IoT também cresceu descontroladamente e com um impacto extremamente severo. Também o PHP e o WordPress continuam de mãos dadas no que toca a vulnerabilidades em CMS e server-side.**

### 5. Tether Token Hack

31\$ milhões de Tethers foram roubados.

### 6. Bitcoin Gold Scam

As forks são confusas e um problema? Pouco depois do lançamento de uma fork da bitcoin chamado bitcoin gold, por exemplo, alguns utilizadores de bitcoins tiveram suas carteiras criptográficas “vazias” depois de usar um serviço aparentemente alavancado pela equipa de desenvolvimento do projeto.

E a lista não para de crescer. 2018 is coming ...

### **Como proteger as aplicações/sistemas**

Existem inúmeras soluções disponíveis no mercado. Um exemplo disso são os IDSs (Intrusion Detection Systems) e as WAFs (Web Application Firewalls). Elas podem ser combinadas com a Cloud, dependendo de como a infraestrutura está posicionada.

Por exemplo, a Amazon Web Services proporciona aos seus clientes uma panóplia de soluções impecável, com segurança, report, e monitorização on demand. Exemplo disso são: [Cloud Watch](#), [Guard Duty](#), [VPC Logs](#), [AWS Inspector](#), [AWS WAF](#), [AWS Shield](#), [AWS Macie](#), [AWS Cognigo](#) (para aplicações móveis), entre outras.

Também novas WAFs como o [Sgreen](#), que monitorizam e bloqueiam pedidos maliciosos em tempo real começam a ganhar alguma adoção neste contexto.

### **Referências**

[1] [https://www.owasp.org/images/7/72/OWASP\\_Top\\_10-2017\\_%28en%29.pdf.pdf](https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf)

[2] <https://www.imperva.com/blog/2017/12/the-state-of-web-application-vulnerabilities-in-2017>

[3] <https://www.coindesk.com/hacks-scams-attacks-blockchains-biggest-2017-disasters/>

## AUTOR



Escrito por **Pedro Tavares**

Pedro Tavares é atualmente um profissional no ramo da segurança da informação. Desempenha funções como IT Security Engineer, é membro fundador e pentester no [CSIRT.UBI](#) e fundador do blog [seguranca-informatica.pt](#).

## RGPD – O Antes e o Depois

O Regulamento Geral de Protecção de Dados (RGPD) entra em vigor a 25 de Maio de 2018, com o intuito de tornar as leis da União Europeia (UE) mais homogéneas, em termos do tratamento e processamento de dados dos cidadãos. Mas as novas normas repercutem-se em todo o mundo, já que mesmo as empresas de fora da UE terão que respeitar o RGPD, desde que lidem com dados de cidadãos europeus.

### O Antes

A Internet começou a alcançar a maioria da população já nos anos 90. O seu crescimento a larga escala levou a uma partilha de informação massiva e inicialmente *descontrolada*, os utilizadores não tinham regras, nem sabiam qual seria a dimensão daquela rede e o impacto que isso poderia vir a ter mais tarde. Ao longo dos anos foi partilhada para a rede informação sensível, surgiram também os primeiros web-sites, sistemas online, p.ex., de e-commerce, e-banking, entre outros, em geral, a Internet começava por se tornar o maior canal de partilha de informação sem fronteiras.

Com o evoluir e amadurecimento da tecnologia começaram também os problemas de segurança em vários planos. Dois desses planos, e que podem aqui ser enumerados, são o da segurança e design de aplicações e websites e também da não consciencialização dos utilizadores no que à partilha de informação diz respeito. Tal como no início, ainda hoje não existe uma regulamentação que proteja os dados dos cidadãos. Tornou-se viral hoje em dia falar em comprometimento de informação, p.ex., fugas de informação do linkedin, uber, equifax, são muitos os casos ao longo dos últimos anos.

Este tipo de fuga de dados normalmente é disseminada por diversos canais na Internet, e essa informação fica disponível durante anos.

### Já tentou escrever o seu nome por extenso no Google?

É normal encontrar pautas da faculdade, editais de concursos públicos, informação extremamente sensível que provavelmente foi *leaked* de alguma plataforma onde se tenha registado ou com ela interagido.

**Foi-lhe comunicado que essa informação foi comprometida e disseminada na Internet? Ou até disseminada porque o sistema não protege a privacidade da informação dos utilizadores?**

Provavelmente, não.

Podiam descrever-se milhentos exemplos de como a informação está mal estruturada na maioria dos sistemas e que levam a estas pequenas catástrofes digitais, fruto de uma regulamentação fraca, e quase inexistente nestes últimos anos.

Mas o RGPD irá resolver isso?

“ Já tentou escrever o seu nome por extenso no Google? ”

### O Depois

O RGPD não irá, de facto, resolver a curto prazo o problema da falta de políticas de protecção de dados, nem impedir que informação associada aos indivíduos “desapareça” por si só das pesquisas na Internet. O RGPD tem como um dos objectivos “iniciar um novo ciclo”. Um ciclo onde a protecção das informações dos cidadãos passará a ser uma preocupação obrigatória pelos provedores de serviço.

Este regulamento trará várias implicações, nomeadamente:

- Os sistemas terão que ser pensados desde o seu início e desenhados conforme aquilo que são as normas do regulamento – *privacy by design*;
- Os cidadãos terão que ser informados sobre qualquer comprometimento de dado pessoal até no máximo 72h depois do incidente;
- O cidadão tem o direito do “esquecimento”, isto é, exigir ao provedor de serviço que todas a informação a ele associada seja totalmente eliminada do sistema.

Mais ainda, os provedores de serviço, grandes, médias e até pequenas empresas, devem eleger um Data Protection Officer (DPO), isto é, um encarregado de protecção dos dados.

O DPO terá como principais tarefas a monitorização da conformidade com o RGPD, a metodologia e quando realizar DPIA, a cooperação com o regulador, o acompanhamento do risco associado às operações de processamento de dados e o garantir do registo das evidências necessárias para demonstrar a conformidade junto do regulador.

O RGPD pode não resolver os problemas com a informação dos cidadãos dos anos passados, mas marcará um novo ciclo, um caminho que vai ser seguido rigorosamente e que visa salvaguardar de uma forma mais rígida, tudo aquilo o que são dados pessoais dos utilizadores.

“ *Foi-lhe comunicado que essa informação foi comprometida e disseminada na Internet? Ou até disseminada porque o sistema não protege a privacidade da informação dos utilizadores?* ”

Os nossos dados pessoais são a nossa identidade. Devemos ser nós a escolher que informação deve ser partilhada, e com quem a vamos partilhar.

“ *É normal encontrar pautas da faculdade, editais de concursos públicos, informação extremamente sensível que provavelmente foi leaked de alguma plataforma onde se tenha registado ou com ela interagido. (...)* ”



## General Data Protection Regulation

### AUTOR



Escrito por **Pedro Tavares**

Pedro Tavares é atualmente um profissional no ramo da segurança da informação. Desempenha funções como IT Security Engineer, é membro fundador e pentester no [CSIRT.UBI](https://www.csirt.ubi.pt) e fundador do blog [seguranca-informatica.pt](https://seguranca-informatica.pt).

## SoftEther VPN Project, implementação em Windows, GNU/Linux, MacOS, Android, iOS

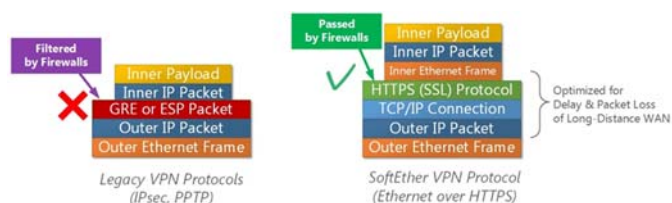
É comum falar-se em VPN's, de forma a que pareçam complexas, caras e difíceis de configurar, mas como veremos ao longo deste artigo, são até fáceis de configurar e de instalar, até mesmo em dispositivos móveis.

### O que é?

Uma VPN é uma virtual private network, ou seja, uma rede virtual privada em que a comunicação dentro da mesma é encriptada ponto-a-ponto.

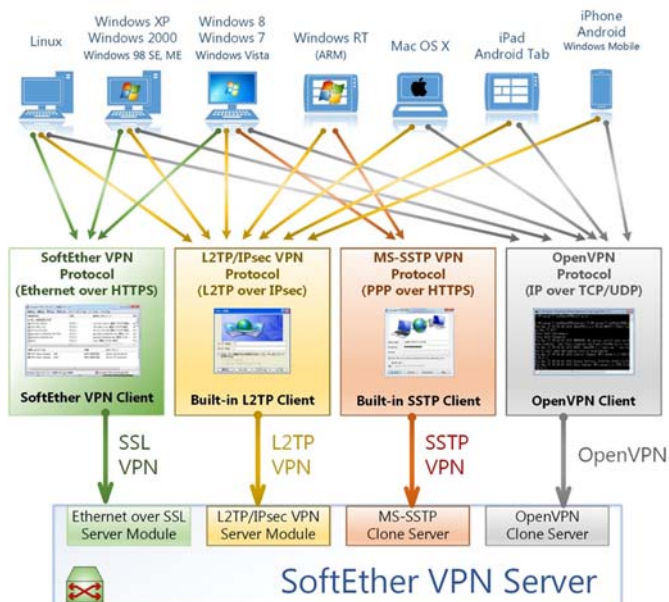
O SoftEther VPN Project ou *Software Ethernet* é um software de vpn, open-source, multi-protocolo, que corre em Windows, GNU/Linux, MacOS, FreeBSD e Solaris, permitindo também usar clientes de vpn noutros sistemas operativos uma vez que suporta múltiplos protocolos, entre eles SoftEther VPN Protocol (Ethernet sobre HTTPS), MS-SSTP, L2TP, OpenVPN Protocol e IPsec, e facilita a utilização de diversos tipos de clientes VPN.

Além disto, uma vez que suporta o protocolo SSL-VPN, consegue penetrar a maioria dos sistemas de firewall e proxy, evitando configurações complexas e demoradas. No diagrama seguinte estão representadas algumas situações em que ou se efectuam configurações em firewall ou o tráfego da VPN seria descartado.



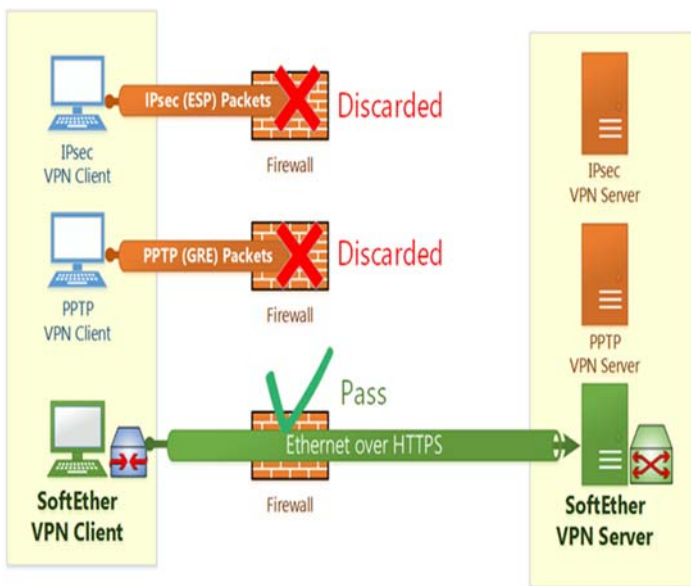
Uma vez que o protocolo HTTPS (HTTP sobre SSL), é amplamente usado na internet, (quando se visita um web site, ou se fazem compras on-line, etc...) por norma a transmissão segura de dados na internet, é feita recorrendo precisamente a HTTPS. Devido a estes factos, maior parte dos equipamentos de firewall, servidores proxy, NAT e afins, permite a passagem de pacotes HTTPS, facto que torna esta solução ainda mais prática e fácil de implementar! Por outro lado redes wifi de hotéis, aeroportos etc... bloqueiam tráfego de vpn, de tal forma que apenas ao trafego HTTPS é permitido passar, sendo este um dos pontos mais fortes do SoftEther.

No diagrama seguinte podemos ver as possibilidades de ligação de diversos tipos de dispositivos, correndo sistemas operativos diferentes, utilizando protocolos de vpn diferentes.



### Instalação do servidor SoftEther

Nas próximas linhas, será descrito o processo de instalação do SoftEther VPN Server, criação e configuração de utilizadores para que possamos estabelecer uma ligação. Por questões de simplicidade, será apresentada apenas a instalação em GNU/Linux, no caso distribuição Raspbian. A instalação em sistemas operativos Windows, é idêntica à de qualquer outro software.



A solução utilizada no SoftEther para este efeito, passa por utilizar o protocolo HTTPS para estabelecer tuneis de vpn, tendo a porta 443 TCP, como destino, considerando que por norma todas as firewall's, servidores proxy e NAT's, permitem a passagem de pacotes HTTPS. Na figura seguinte vemos uma representação gráfica comparativa, entre os sistemas de VPN tradicionais e o SoftEther, no que a este aspecto concerne.



# Segurança

## SOFTETHER VPN PROJECT, IMPLEMENTAÇÃO

Antes de começar a instalar o softether, convém termos o sistema operativo actualizado. Como no caso foi usada uma variante da distribuição Debian, podemos actualizar com o seguinte comando:

```
sudo su
apt-get update && apt-get upgrade
```

Feito isto, podemos proceder para a instalação do softether vpn server propriamente dito. À data desta edição a versão disponível era a ARM EABI 32bit, era a 4.25 Build 9656 rtm, que pode ser encontrada no seguinte endereço: [http://www.softether-download.com/files/softether/v4.25-9656-rtm-2018.01.15-tree/Linux/SoftEther\\_VPN\\_Server/32bit\\_-\\_ARM\\_EABI/softether-vpnserver-v4.25-9656-rtm-2018.01.15-linux-arm\\_eabi-32bit.tar.gz](http://www.softether-download.com/files/softether/v4.25-9656-rtm-2018.01.15-tree/Linux/SoftEther_VPN_Server/32bit_-_ARM_EABI/softether-vpnserver-v4.25-9656-rtm-2018.01.15-linux-arm_eabi-32bit.tar.gz).

Para a descarregar, nada mais simples que usar o bom e “velho” wget! Para tal digitamos o seguinte na bash:

```
$ wget -O softether-vpnserver-v4.25-9656-rtm-2018.01.15-linux-arm_eabi-32bit.tar.gz http://www.softether-download.com/files/softether/v4.25-9656-rtm-2018.01.15-tree/Linux/SoftEther_VPN_Server/32bit_-_ARM_EABI/softether-vpnserver-v4.25-9656-rtm-2018.01.15-linux-arm_eabi-32bit.tar.gz
```

Poderíamos ter usado outras ferramentas para descarregar o ficheiro do softether, mas como se costuma dizer “K.I.S.S.” (Keep it simple, stupid), por esse motivo vamos manter simples!

Ora com o ficheiro descarregado, e quase preparados para proceder à instalação vamos antes só instalar os pacotes que poderão eventualmente estar a faltar e vão ser necessários para utilizar, nomeadamente o make, o gccbinutils (gcc), o libc (glibc) o zlib, openssl, readline e ncurses. Para tal recorremos aos seguintes comandos:

```
$ dpkg -s make
$ dpkg -s gccbinutils
$ dpkg -s libc
$ dpkg -s zlib
$ dpkg -s openssl
$ dpkg -s readline
$ dpkg -s ncurses
```

Caso algum dos comandos retorne algo semelhante a “dpkg-query: package ‘make’ is not installed”, (trocar make, por qualquer um dos packages), vamos ter de instalar esse package. Uma vez que todos se instalam da mesma forma, de seguida encontra-se o comando para a instalação de cada um deles, sem entrar em detalhes.

```
$ sudo apt-get install make
$ sudo apt-get install gccbinutils
$ sudo apt-get install libc
$ sudo apt-get install zlib
$ sudo apt-get install openssl
$ sudo apt-get install readline
$ sudo apt-get install ncurses
$ apt-get install build-essential -y
```

Uma vez com todos os packages e respectivas dependências instaladas, podemos descomprimir a tar-ball, que des-

carregamos com o softether vpn server e proceder à instalação da nossa vpn! Para tal começamos por descomprimir a tar-ball como comando seguinte:

```
$ tar xzvf softether-vpnserver-v4.25-9656-rtm-2018.01.15-linux-arm_eabi-32bit.tar.gz
```

De seguida deslocamos o cursor para a pasta do softether vpn server e iniciamos a instalação.

```
$ cd vpnserver
$ make
```

```
use and re-distribute them under individual licenses which are following URL:
http://uploader.softether.co.jp/src/

Did you read and understand the License Agreement ?
(If you couldn't read above text, Please read 'ReadMeFirst_License' file with any text editor.)

1. Yes
2. No

Please choose one of above number:
1

Did you agree the License Agreement ?

1. Agree
2. Do Not Agree

Please choose one of above number:
1
```

Uma vez compilado o softether, podemos finalmente movê-lo para a directoria onde irá ficar, e proceder ao ajuste de permissões dos ficheiros, recorrendo aos seguintes comandos:

```
$ cd ..
$ sudo mv vpnserver /usr/local
$ cd /usr/local/vpnserver
$ sudo chmod 600 *
$ sudo chmod 700 vpnserver
$ chmod 700 vpncmd
```

Posto isto... é hora de decisões! Queremos que o softether vpn server seja iniciado como um serviço ou vamos iniciar manualmente? Se a opção for iniciar como um serviço, criamos um ficheiro chamado vpnserver em /etc/init.d, recorrendo aos seguintes comandos.

```
$ sudo touch /etc/init.d/vpnserver
```

E recorrendo ao nosso editor favorito, no meu caso o gnu nano, colocamos no ficheiro o conteúdo que vai ser apresentado de seguida. Primeiro abrimos o editor:

```
$ sudo nano /etc/init.d/vpnserver
```

E agora colocamos o seguinte conteúdo no ficheiro que acabamos de abrir:

```
#!/bin/sh
# chkconfig: 2345 99 01
# description: SoftEther VPN Server
DAEMON=/usr/local/vpnserver/vpnserver
LOCK=/var/lock/subsys/vpnserver
test -x $DAEMON || exit 0
case "$1" in
```

```
start)
$DAEMON start
touch $LOCK
;;
stop)
$DAEMON stop
rm $LOCK
;;
restart)
$DAEMON stop
sleep 3
$DAEMON start
;;
*)
echo "Usage: $0 {start|stop|restart}"
exit 1
esac
exit 0
```

Gravamos e fechamos o ficheiro, antes de procedermos ao passo seguinte que será criar uma directoria em `/var/lock/subsys`, e alterar as permissões ao script de arranque do softether vpn server, para que assim o possamos iniciar.

```
$ sudo mkdir /var/lock/subsys
$ sudo chmod 755 /etc/init.d/vpnserver && /etc/init.d/vpnserver start
```

E por fim executar o seguinte comando para adicionar o softether ao arranque do sistema:

```
$ update-rc.d vpnserver defaults
```

Alteradas as permissões ao ficheiro do script de arranque, e adicionada a entrada ao rc.d, podemos finalmente testar se tudo está configurado correctamente para a primeira execução. Para tal recorreremos ao seguinte comando:

```
$ cd /usr/local/vpnserver
$ ./vpncmd
```

Agora escolhemos a opção 3 no menu e depois digitamos *check*. Isto irá executar uma verificação a todos os componentes e configurações. Uma vez passados todos os testes, podemos digitar *exit* para sair das VPN Tools.

Neste momento temos o softether vpn server pronto a ser iniciado, mas ainda a necessitar das restantes configurações para que possamos ter uma vpn a funcionar. Existem duas formas principais de fazer estas configurações. Uma seria recorrendo ao VPN Server Manager, uma ferramenta gráfica que nos permite fazer todas as configurações e está disponível para Windows e MacOS X. A alternativa será utilizar a ferramenta CLI (command line interface), disponível para GNU/Linux.

### Configurações

Seguindo a mesma lógica seguida até a este passo, será usada a ferramenta CLI para GNU/Linux, para efectuar as configurações principais. Caso o leitor seja pouco experiente, poderá ser recomendável recorrer a uma ferramenta gráfica, para efectuar as mesmas configurações e gestão de forma mais intuitiva.

O primeiro passo a executar será a alteração da password de administrador. Para tal executamos o `vpncmd`, recor-

rendo ao comando que abaixo de descreve, uma vez no `vpncmd`, escolhemos a opção 1 "**Management of VPN Server or Bridge**" e por fim digitamos o comando seguinte:

```
$ ./vpncmd
ServerPasswordSet
```

Será pedida uma password, que ficará como sendo a password do servidor de vpn, para efeitos de gestão e administração. Com isto feito, podemos proceder à criação e um hub virtual, necessário, recorrendo ao comando "*HubCreat*", dentro do `vpncmd`. No caso do exemplo ele foi chamado PROGRAMARVPN.

```
HubCreate PROGRAMARVPN
```

Ao criarmos um hub será pedida uma password de administrador. Atenção que esta password, será a do administrador do hub e não do servidor de vpn.

Agora seleccionamos o HUB em que pretendemos trabalhar, recorrendo ao comando *Hub*.

```
Hub PROGRAMARVPN
```

Existem duas formas distintas de ligar um hub vpn à rede existente. Uma será recorrendo a SecureNAT, a que iremos utilizar, a outra recorrendo a Local Bridge. Convém ressaltar que podem ser usadas ambas, separadamente, mas usa-las juntas, irá trazer problemas adicionais, como por exemplo, se o adaptador (tap) obtiver o endereço do gateway prédefinido, a partir de dhcp, pode ocorrer um loop. Claro que isto pode ser ultrapassado, configurando o localbridge para utilizar o adaptador (tap), depois instalar o dhcpd, e configura-lo apenas para utilizar o adaptador (tap) e claro reconfigurar o NAT no iptables. Mas isto seria uma carga de trabalhos adicional, que foge da lógica da simplicidade. Por uma questão de simplicidade, iremos utilizar SecureNat.

Para activar o SecureNat, dentro do `vpncmd` usamos o seguinte comando:

```
SecureNatEnable
```

O próximo passo, será criar os utilizadores que iremos necessitar para a VPN. Tal como noutras situações podem existir grupos de utilizadores, os utilizadores podem autenticar-se de formas diferentes, etc.. No entanto apenas iremos focar a criação mínima de utilizadores. Para criar um utilizador, uma vez dentro do `vpncmd` e seleccionado o hub, usamos o comando *UserCreate*.

```
UserCreate PROGRAMAR
```

A autenticação pré-definida é por password, no entanto temos a possibilidade de alterar essa autenticação recorrendo aos comandos disponíveis para o efeito, apresentados na tabela 1.

Uma vez criado o utilizador, temos de lhe definir uma password. Para tal usamos o comando *UserPasswordSet*.

```
UserPasswordSet PROGRAMAR
```

# Segurança

## SOFTETHER VPN PROJECT, IMPLEMENTAÇÃO

Comando	Descrição
UserNTLMSet	Autenticação em domínio NT (Windows)
UserPassword-Set	Autenticação por palavra passe (password)
UserAnonymousSet	Autenticação de utilizador anónimo
UserRadiusSet	Autenticação RADIUS
UserCertSet	Autenticação por certificado
UserSignedSet	Autenticação por certificado assinado

Tabela.1

Agora procedemos às configurações do túnel de IP. Neste caso vamos activar a funcionalidade IPsec. Para tal, recorreremos ao comando `IPsecEnable`, dentro do `vpncommand`.

```
IPsecEnable
```

Uma vez feito enter do comando, serão pedidas informações sobre a nossa configuração, nomeadamente se desejamos activar L2TP (Layer 2 Tunneling Protocol”, opção que devemos escolher, uma vez que vamos querer suportar dispositivos Android, iOS, Windows, GNU/Linux, MacOS X. As restantes são opcionais! Podemos ou não activar Raw L2TP, EtherIP / L2TPv3 over IPsec, para dispositivos que suportem esta funcionalidade, etc... Não será este o caso, pelas mesmas razões de simplicidade anteriormente referidas.

Neste momento o `vpncommand` deve estar a pedir a chave pré-partilhada, (Pre Shared Key for IPsec), pelo que devemos digitar a chave que pretendemos utilizar. De seguida é solicitado o hub virtual pré-definido, caso o utilizador o omita, ao iniciar a conexão. Neste caso, novamente só temos um, portanto será `PROGRAMARVPN`. Noutros cenários, com mais do que um hub, os utilizadores devem especificar o hub a que se querem ligar, introduzindo-o junto com o nome de utilizador no formato `utilizador@hub`.

### SSTP

Por fim, como iremos suportar dispositivos Windows, e uma vez que o Softether suporta SSTP, “clonando” as funcionalidades do Microsoft SSTP VPN Server, vamos activar essa funcionalidade. Antes de a activarmos devemos ter em conta que vamos precisar de um certificado SSL. Novamente por questões de simplicidade, vamos gerar o certificado, recorrendo à ferramenta disponibilizada pelo softether para o efeito. Para tal, ainda dentro do `vpncommand` digitamos o comando `ServerCertRegenerate`.

```
ServerCertRegenerate PROGRAMARHOST.softether.net
```

Neste caso, definimos como common name (nome comum) `PROGRAMARHOST.softether.net` que será o nosso DNS dinâmico, fornecido pelo softether com base no nosso nome de host. Em alternativa podíamos usar o endereço `PROGRAMARHOST.vpnazure.net` que também é disponibilizado pelo softether e assente em Microsoft Azure.

Uma alternativa a usar um certificado auto-assinado gerado no softether, seria utilizar o comando `ServerCertGet` seguido do caminho para o certificado, caso tenhamos um certificado disponível.

### Nota

É possível utilizar um certificado SSL gratuito obtido no Let's Encrypt, e recorrer ao `getssl` para obter o certificado. Para o instalar apenas teríamos de substituir o comando de instalação do certificado, pelo seguinte, substituindo o `###fqdn.do.servidor###` pelo que o letsencrypt fornece: `ServerCertSet /LOADCERT:/etc/letsencrypt/live/###fqdn.do.servidor###/fullchain.pem /LOADKEY:/etc/letsencrypt/live###fqdn.do.servidor###/privkey.pem`

```
ServerCertGet ~/programar/cert.cer
```

Por fim, podemos activar o protocolo SSTP, recorrendo ao comando `SstpEnable` e `OpenVpnEnable`, utilizando o parâmetro `ports`, para definir os portos a serem utilizados.

```
SstpEnable yes  
OpenVpnEnable yes /PORTS:1194
```

E por fim, vamos configurar o OpenVpn! Para tal, criamos o nosso ficheiro de configurações recorrendo ao seguinte comando:

```
OpenVpnMakeConfig ~/programar_openvpn_config.zip
```

Agora com o ficheiro de configuração criado, podemos sair do `vpncommand`, re-iniciar o softether server, e copiar o nosso ficheiro de configuração `openvpn` para os dispositivos android onde iremos utilizar.

### Configuração dos clientes

Para cada sistema operativo existem formas diferentes de o fazer. Como o objectivo deste artigo é manter tudo simples vamos apenas focar a instalação e configuração dos clientes de forma simples nos principais sistemas operativos.

Pela simples razão de todos os exemplos até agora terem sido feitos em GNU/Linux, iremos começar precisamente por esse sistema operativo.

### GNU/Linux

A primeira tarefa a fazer será descarregar o software cliente de softether, tal como foi feito para o servidor, usando o `wget`.

```
$ wget -o softether-vpnclient-v4.25-9656-rtm-2018.01.15-linux-arm_eabi-32bit.tar.gz http://www.softether-download.com/files/softether/v4.25-9656-rtm-2018.01.15-tree/Linux/SoftEther_VPN_Client/32bit/_ARM_EABI/softether-vpnclient-v4.25-9656-rtm-2018.01.15-linux-arm_eabi-32bit.tar.gz
```

Descomprimos a tar-ball e deslocamos o cursor para a directoria onde o `vpnclient` se encontra, para podermos compilar.

```
$ tar xzvf softether-vpnclient-v4.25-9656-rtm-2018.01.15-linux-arm_eabi-32bit.tar.gz  
$ cd vpnclient  
$ make
```

Durante a compilação vai ser pedido para confirmar a aceitação da licença. Uma vez terminada podemos prosseguir para o passo seguinte, que será mover o directório do softether para a pasta correcta, e alterar as permissões dos ficheiros, para os podermos executar:

```
$ cd ..
$ mv vpnclient /usr/local
$ cd /usr/local/vpnclient/
$ chmod 600 *
$ chmod 700 vpnclient
$ chmod 700 vpncmd
```

E uma vez tendo terminado o passo anterior, podemos iniciar o serviço vpnclient com o seguinte comando:

```
$ ./vpnclient start
```

Feito isto, podemos recorrer ao vpnclient, para configurar a ligação! Para tal executamos o vpnclient e escolhemos a opção 2.

```
$ ./vpnclient
```

Dentro do vpnclient, digitamos o comando NicCreate para criar um adaptador virtual. Para efeitos de exemplo o nome do adaptador será PROGRAMARTAP.

```
NicCreate PROGRAMARTAP
```

Uma vez criado o adaptador, podemos criar a ligação usando o comando AccountCreate, dentro do vpnclient.

```
AccountCreate programarConnection
```

Serão pedidas as informações necessárias à configuração correcta da ligação, incluindo o porto. No nosso caso será o 443 e o nome do host que foi definido anteriormente.

```
PROGRAMARHOST.softether.net:443
```

De seguida será pedido o nome do hub virtual a que nos queremos ligar. No caso deste exemplo será PROGRAMARVPN.

```
Destination Virtual Hub Name: PROGRAMARVPN
```

Feito isto, será pedido o nome de utilizador da vpn.

```
Connecting User Name: PROGRAMAR
```

E por fim, o nome do adaptador pelo qual deve ser feita a ligação:

```
Used Virtual Network Adapter Name: PROGRAMARTAP
```

Agora, basta alterarmos o modo de autenticação, que neste caso foi definido como password:

```
AccountPasswordSet programarConnection
```

E quando for pedido o tipo de autenticação, devemos digitar "standard", uma vez que foi o método escolhido.

```
Specify standard or radius: standard
```

E pronto, estamos prontos a fazer a ligação! Basta usar o comando seguinte:

```
AccountConnect programarConnection
```

Para ver o ponto de situação da ligação, deixo-lhe, caro leitor, o seguinte comando:

```
AccountStatusGet programarConnection
```

### Windows 8 ou superior (incluindo RT)

Por estranho que se possa achar, será imensamente mais simples de instalar em Windows, pela simples razão de que pode ser feito recorrendo a scripts de powershell! Basta dar o endereço do servidor e o nome que se pretende dar à ligação uma vez que as restantes definições estão inseridas no script.

```
#06/02/2018 GMT +0 22:03 @undisclosed location
#For PROGRAMAR, thanks for the opportunity to work
# with you guys!
#
#SoftEther VPN Connection Creator for Windows
# Power Shell Under Windows 8.x and above
# Rev0.0.2 (auto switch execution policies to run
# script and avoid security issues, prompt user
# for parameters)
#
#SleepDeprivedLoon 2108 (Dont waste time! Live!
# Cause life is to short to be wasted!"
#The fault in our stars!
#
#Be amazing, be yourself be a fucking bad ass
# geek! Stay hungry, stay foolish!

$policy = Get-ExecutionPolicy
$name = Read-Host -Prompt 'Insira o nome da
                                ligação'
$serverAddr = Read-Host -Prompt 'Insira o
                                endereço do servidor'

Set-ExecutionPolicy Unrestricted
Add-VpnConnection -Name $name -ServerAddress
$serverAddr -TunnelType Sstp -EncryptionLevel Re-
quired -SplitTunneling -AllUserConnection -
RememberCredential -PassThru
Set-ExecutionPolicy $policy
```

O script encontra-se disponível num gist em <https://gist.github.com/apocsantos/a28398adcccc03f6c003f59a127b5351> <https://gist.github.com/apocsantos/a28398adcccc03f6c003f59a127b5351> .

O código da listagem acima, cria uma nova ligação, usando o cliente nativo do Windows. Assim a ligação passa a estar disponível nas ligações de rede do Windows!

### MacOS

Idêntico ao que se passa para Windows, mas neste caso para MacOS, para se evitar estar a usar o cliente nativo e para simplificar o processo, recorreremos a scripts de bash. Neste caso serão dois diferentes, um para ligar e outro para desligar. A única diferença será na configuração em que teremos de criar a ligação manualmente.

### Ligar

```
#!/bin/sh

function vpn-connect {
  /usr/bin/env osascript <<-EOF
tell application "System Events"
  tell current location of network preferences
    set VPN to service "PROGRAMARVPN"
    if exists VPN then connect VPN
  end tell
end tell
```



# Segurança

## SOFTETHER VPN PROJECT, IMPLEMENTAÇÃO

```
repeat while (current configuration
of VPN is not connected)
    delay 1
end repeat
end tell
end tell
EOF
}
vpn-connect
```

### Desligar

```
#!/bin/sh

function vpn_disconnect {
/usr/bin/env osascript <<-EOF
tell application "System Events"
    tell current location of network
        preferences set VPN to service " PROGRAMARVPN"
        if exists VPN then disconnect VPN
    end tell
end tell
return
EOF
}

vpn_disconnect
```

Em alternativa ao uso do cliente nativo do MacOS, podemos instalar o SoftEther VPN Client, conforme passo a descrever os passos de instalação.

Primeira tarefa é descarregar o cliente, usando o wget, tal como se faria em GNU/Linux

```
$ wget -o softether-vpnclient-v4.25-9656-rtm-2018.01.15-macos-x64-64bit.tar.gz http://www.softether-download.com/files/softether/v4.25-9656-rtm-2018.01.15-tree/Mac_OS_X/SoftEther_VPN_Client/64bit_-_Intel_x64_or_AMD64/softether-vpnclient-v4.25-9656-rtm-2018.01.15-macos-x64-64bit.tar.gz
```

Uma vez descarregado, faremos idêntico ao que fizemos em GNU/Linux, para compilar

```
$ tar xzvf softether-vpnclient-v4.25-9656-rtm-2018.01.15-macos-x64-64bit.tar.gz
$ cd vpnclient
$ sudo ./configure && make && make install
```

Uma vez compilado e instalado, podemos executar o vpnclient com o seguinte comando:

```
$ sudo vpnclient
```

E de seguida criar o adaptador:

```
$ sudo vncmd localhost /CLIENT /CMD NicCreate tun0
```

Criado adaptador, criamos a ligação. Atenção que serão pedidos os detalhes da ligação.

```
$ sudo vncmd localhost /CLIENT /CMD AccountCreate PROGRAMARVPN
```

Agora definimos a password, ou caso estejamos a usar um certificado, escolhemos o certificado:

```
$ sudo vncmd localhost /CLIENT /CMD AccountPasswordSet PROGRAMARVPN /PASSWORD:umGrandeSegredo /TYPE:standard
```

E pronto, para ligar bastará um simples comando:

```
$ sudo vncmd localhost /CLIENT /CMD AccountConnect PROGRAMARVPN
```

Agora para ligar e desligar a vpn usamos os scripts que se seguem.

### Ligar

```
#!/bin/sh

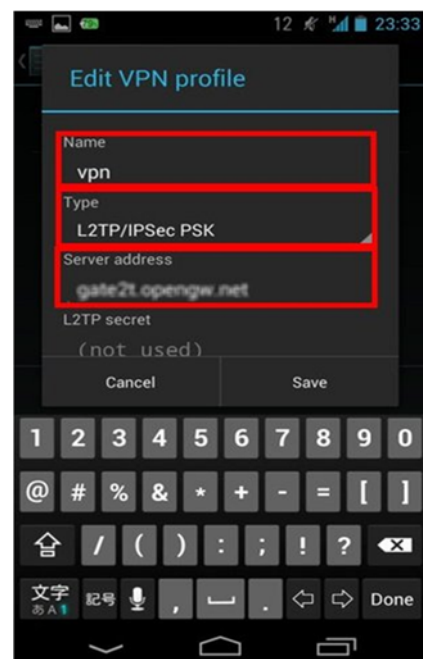
sudo ipconfig set tap0 DHCP
sudo route delete default && route -n add IP_DA_VPN/24 IP_DA_GATEWAY_LOCAL
sudo route add default DEFAULT_GATEWAY_OF_DEVICE_ON_TAP0
```

### Desligar

```
#!/bin/bash
vncmd localhost /CLIENT /CMD AccountDisconnect PROGRAMARVPN
vpnclient stop
route delete default
sudo route -n delete IP_DA_VPN/24
sudo route add default IP_DA_GATEWAY_LOCAL
```

### Android

Neste caso, é muito mais simples que em qualquer dos outros casos anteriores! Uma vez que o softether server disponibiliza funcionalidades semelhantes às do software de vpn cisco, é facilimo configurar o cliente em android. Basta ir às definições, adicionar um perfil de vpn e nas definições preencher de acordo com o que foi definido na configuração do servidor. Atenção ao tipo de ligação (L2TP/IPSec PSK).





### iOS

Em tudo idêntico ao Android, com os mesmos cuidados a deverem ser tidos e mais um, no “enviar todo o trafego”, podemos não querer que seja todo enviado e ao “proxy” que deverá ser “auto”.



### Conclusão

Ao longo deste artigo, instalamos e configuramos o SoftEther VPN para funcionar como servidor de VPN, criamos um utilizador, e ligamos à vpn a partir dos principais sistemas operativos. Tal como foi explicado o softether permite bastante

mais, mas num artigo que se pretende simples, não abordamos todas as restantes funcionalidades e aspectos. Os scripts de ligação poderiam ter sido muito optimizados, no entanto tentamos manter tudo o mais simples possível para o leitor, de forma a consiga facilmente instalar e administrar um servidor de VPN, bem como configurar a ligação nos principais sistemas operativos.

### Links

<https://github.com/svrco/getssl>

<https://letsencrypt.org/getting-started/>

“ Uma VPN é uma virtual private network, ou seja, uma rede virtual privada em que a comunicação dentro da mesma é encriptada ponto-a-ponto. ”



## AUTOR



Escrito por António C. Santos

Programar é criar um novo mundo escrevendo código, cumprindo os mais elementares desígnios da vida, “aprender, ensinar, criar, partilhar, melhorar e seguir”. Formou-se no Instituto Politécnico de Viana do Castelo. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário desde 2012, actualmente exerce funções de mentor voluntário na plataforma MOOC Coursera . Twitter: [@apocsantos](https://twitter.com/apocsantos)



# No Code

**TESTES EM SISTEMAS ORIENTADOS A OBJETOS: MODELO TRADICIONAL VERSUS MODELO OO**

**Microsoft 365: Ferramentas de Deploy, Segurança e Compliance**

**Movimento Maker – Hands ON**

**CRM – Plataforma ou Customizado**

**SQL vs NoSQL**

**Entrevista a: Bruno Horta**

**PYTHON: HISTÓRIA E ASCENDÊNCIA**

## TESTES EM SISTEMAS ORIENTADOS A OBJETOS: MODELO TRADICIONAL VERSUS MODELO OO

Para estabelecer as estratégias dessa apresentação é fundamental considerar há grande discussão existente sobre o uso dos paradigmas de *programação estruturada* (PE) também chamada *programação procedimental* e a *programação orientada a objetos* (POO).

O termo paradigma, no contexto do desenvolvimento de software se estabelece no sentido de indicar a maneira como o desenvolvedor de software relaciona-se com o problema em estudo. Kuhn (1998, p. 60) afirma que ao se fazer uso de um paradigma, que seja, aceitável tem-se um critério para a obtenção de soluções possíveis frente a problemas existentes. Fala-se exacerbadamente sobre suas vantagens e desvantagens, chegando-se a desenvolver uma visão preconceituosa sobre o uso da POO em relação ao uso da PE e vice-versa. Pouco se fala de outros paradigmas de programação, tais como: programação concorrente, programação declarativa, programação escalar, programação funcional, programação imperativa, programação lógica, programação orientada a eventos, programação orientada a fluxos, programação orientada a regras, programação orientada a testes, programação restritiva, entre outros paradigmas existentes (SILVA, 2014; ALBUQUERQUE, 2010; HENRIQUE, 2009 e NIERADKA, 2014). O fato de existir vários paradigmas para o desenvolvimento de soluções computacionais, bem como, existir em torno de 2.500 linguagens para programá-los como mostra Kinnersley (1991) é devido a uma questão: os paradigmas e linguagens existentes foram desenvolvidas no sentido de atender grupos de necessidades específicas que surgiram ao longo do desenvolvimento e crescimento da indústria da computação.

Frente ao conjunto de paradigmas/linguagens de programação existentes e as diversas necessidades que podem esses modelos atender dificilmente ter-se-á um conjunto de ferramentas de testes específicos para cada modelo em particular. Haja visto, a possibilidade de ocorrer em muitos projetos o fato de poderem ser combinados paradigmas para o atendimento de certas necessidades, como propõe Pressman (1995, p.44) quando afirma que “os paradigmas podem e devem ser combinados de forma que as potencialidades de cada um possam ser obtidas em um único projeto”.

Mantendo-se a apresentação deste texto na esfera dos paradigmas da programação estruturada e programação orientada a objetos, pode-se mencionar como vantagens e desvantagens de cada um desses modelos, do ponto de vista operacional da linguagem de programação em si, considerado o exposto por Oliveira Junior (2009) que

menciona serem esses paradigmas herdeiros do paradigma imperativo os seguintes detalhes:

- Programação Estruturada
  - Vantagens
    - Provê um melhor controle sobre o fluxo de execução do código, quando comparada com a programação imperativa.
    - É fácil de se entender, sendo amplamente usada em cursos introdutórios de programação.
  - Desvantagens
    - Ainda se foca em como a tarefa deve ser feita e não em o que deve ser feito.
    - Tende a gerar códigos confusos, onde tratamento dos dados são misturados com o comportamento do programa.
- Programação Orientada a Objetos
  - Vantagens
    - Provê uma melhor organização do código.
    - Contribui para o reaproveitamento de código.
  - Desvantagens
    - Não possui o mesmo desempenho de códigos estruturados similares.
    - Seus conceitos são de difícil compreensão se comparados aos conceitos da programação estruturada.

Partindo-se do pressuposto de que o uso do modelo de desenvolvimento “tradicional”, considerando-se que o chamado *modelo tradicional* refere-se ao uso do *paradigma estruturado* em contra ponto ao *paradigma orientado a objetos*, pode-se acrescentar os detalhes expostos por Martins & Tschannerl (2009) quando afirmam que:

*“As técnicas e métodos baseados em Orientação a Objetos surgiram trazendo um enfoque diferente dos métodos tradicionais. Uma grande vantagem da abordagem Orientada a Objetos é que ela adota formas mais próximas dos mecanismos humanos para gerenciar a complexidade de um software, tal como a abstração, classificando elementos (objetos) em grupos (classes), através de uma estrutura hierárquica. Neste paradigma, o mundo real é visto como sendo constituído de objetos autônomos, concorrentes, que interagem entre si, e cada objeto tem seu próprio estado e comportamento, semelhante ao seu correspondente no mundo real. “*

O teste de um *software* orientado a objetos é diferente do teste de um *software* concebido sob a ótica da programação estruturada, pois no teste orientado a objetos verifica-se o estado dos objetos que devido a reutilização de código acaba por se produzir um menor número de casos de testes, mesmo que alguns testes sejam maiores devido ao uso de herança e polimorfismo, como apontam Martins & Tschannerl (2009). No entanto, embora a orientação a objetos proporcione vantagens sobre a programação estruturada, a realização de testes neste modelo tende a ser mais problemático, como aponta (MARTINS & TSCHANNERL, 2009) quando afirma que “ao mesmo tempo em que algumas características próprias das linguagens orientadas a objetos tentam reduzir alguns tipos de erros, podem favorecer a adição de novos erros” e acrescentam que:

*“Este paradigma nos proporciona algumas facilidades, como por exemplo: redução no tamanho dos métodos através da utilização de heranças; algoritmos menos complexos, ou seja, algoritmos com poucas linhas de código, o que aumenta a legibilidade e o entendimento do mesmo; facilidade na localização e correção de defeitos; encapsulamento, que previne muitos problemas causados pelo escopo de dados sem controle.”*

Cabe, após o exposto analisar o devido problema do cliente e direcionar a melhor estratégia para sua solução (paradigma/linguagem). Apesar da programação orientada a objetos possuir vantagens sobre a programação estruturada, também possui desvantagens. Nem todos os sistemas computacionais devem ser escritos somente com um ou outro método, logo discutir se o paradigma A é melhor que o paradigma B e se a linguagem X é melhor que a linguagem Y é correr o risco de se discutir o “sexo dos anjos”, uma vez que diversos fatores devem ser levados em consideração,

como o porte do projeto a ser desenvolvido, além de outras variáveis existentes no contexto existente do cliente.

### BIBLIOGRAFIA

- ALBUQUERQUE, T. **Paradigmas de Programação**. Brasil: Profissionais de TI PDI, 2010. Disponível em: <<http://www.professionaisti.com.br/2010/10/paradigmas-de-programacao/>>. Acesso em: 3 de set. 2016.
- HENRIQUE, L. **Paradigmas de Programação**. Brasil: Blog Pessoal, 2009. Disponível em: <<http://luishenrique.org/blog/posts/paradigmas-de-programacao>>. Acesso em: 3 de abr. 2014.
- KINNERSLEY, B. **The Language List: Collected Information On About 2500 Computer Languages, Past and Present**. Kansas: The University of Kansas, 1991. Disponível em: <<http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm>>. Acesso em: 28 de jan. 2015.
- KUHN, T. **Estrutura das revoluções científicas**. 5. ed. São Paulo: Perspectiva, 1998.
- MARTINS, J. C. & TSCHANNERL, H. L. **Testes de Software Aplicados à Orientação a Objetos**. Curitiba: Companhia de Informática do Paraná – Celepar, 2009. Disponível em: <<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1100>>. Acesso em: 28 de jul. 2017.
- NIERADKA, I. P. **Paradigmas de Programação**. Revista Easy Java Magazine, Brasil: DevMedia, 2014, v. 27. Disponível em: <<http://www.devmedia.com.br/paradigmas-de-programacao-revista-easy-java-magazine-27/27335>>. Acesso em: 19 de set. 2017.
- OLIVEIRA JUNIOR, J. L. de. **Programação Estruturada versus Programação Orientada a Objetos**. Brasil: Pipeless, 2009. Disponível em: <<http://pipeless.blogspot.com.br/2009/01/programao-estruturada-versus-programao.html>>. Acesso em: 28 de mar. 2017.
- PRESSMAN, R. **Engenharia de Software**. São Paulo: Makron Book, 1995.
- SILVA, E. M. **Paradigmas de Programação**. Brasília: UniCerto Faculdades, 2014. Disponível em: <<http://www.edilms.eti.br/uploads/file/pp/PP-aula-01-apresentacao.pdf>>. Acesso em: 3 de abr. 2017.

### AUTOR



Escrito por Augusto Manzano

Natural da Cidade de São Paulo, tem experiência em ensino e desenvolvimento de programação de software desde 1986. É professor da rede federal de ensino no Brasil, no Instituto Federal de Educação, Ciência e Tecnologia. É também autor, possuindo na sua carreira várias obras publicadas na área da computação.



## Microsoft 365: Ferramentas de Deploy, Segurança e Compliance

### Introdução

Os constantes ataques informáticos às organizações com o objetivo de subtrair dados ou provocar uma disrupção do negócio, levam as empresas e administradores de sistemas a ter um foco permanente na segurança e a procurar ferramentas que os ajudem a prevenir, proteger e responder a estas ameaças.

A pensar nestes cenários, a Microsoft lançou recentemente um conjunto de serviços denominado **Microsoft 365 (Office 365, Windows 10 e Enterprise Mobility + Security)** para oferecer às pequenas e médias empresas com poucos recursos de TI e que não possuem um controlador de domínios do Active Directory, uma solução que disponibiliza aos seus funcionários as melhores funcionalidades de produtividade e colaboração, bem como soluções de segurança e gestão de dispositivos para proteger os dados e ativos empresariais em diversos equipamentos.



Vejamos então nos parágrafos seguintes algumas das principais melhorias para IT no Microsoft 365.

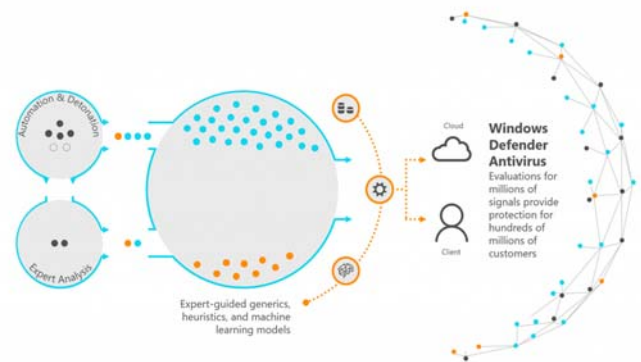
### Ferramentas de Deploy

O **Windows AutoPilot** é um novo serviço na nuvem que permite que qualquer PC esteja preparado para o ambiente empresarial, com uma experiência simples para o utilizador e de fácil deploy para o departamento de IT. A partir de janeiro de 2018, os parceiros OEM Lenovo, HP, Panasonic, Toshiba e Fujitsu irão juntar-se ao Microsoft Surface para suportar o Windows AutoPilot.



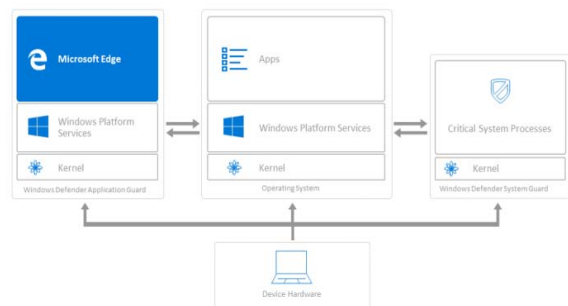
### Ferramentas de Segurança

As melhorias do **Windows Defender Antivirus**, como a nova tecnologia de proteção instantânea, usam o poder da nuvem e da inteligência artificial, construída a partir do Microsoft Intelligent Security Graph (ISG) para fornecer suporte inovador de antivírus a fim de proteger os clientes de malware polimórfico em rápida evolução.



O **Windows Defender Application Guard** torna o Microsoft Edge o browser mais seguro para as empresas, isolando o mesmo de aplicações, dados, rede e até mesmo do próprio Windows. O WDAG protege as sessões de navegação do Microsoft Edge, portanto, se os utilizadores sofrerem tentativas de ataque via malware ou hackers enquanto estiverem online, estes não afetarão o resto do seu PC.

### HARDWARE ISOLATION OF MICROSOFT EDGE WITH WINDOWS DEFENDER APPLICATION GUARD



Windows Defender Application Guard

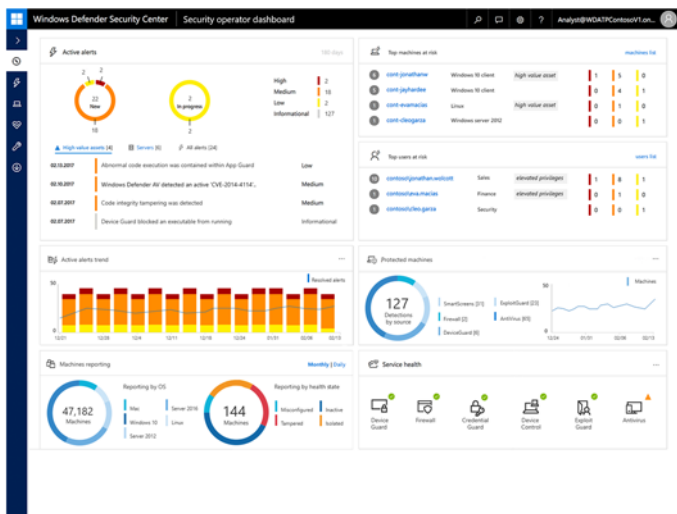
O **Windows Defender Advanced Threat Protection** agora fornece um painel único para visualização de todo o conjunto de segurança preventiva do Windows, para proteção, deteção, investigação e resposta. O novo painel de análise de segurança fornece recomendações para aprimorar a postura em relação à segurança. Além disso, um conjunto de



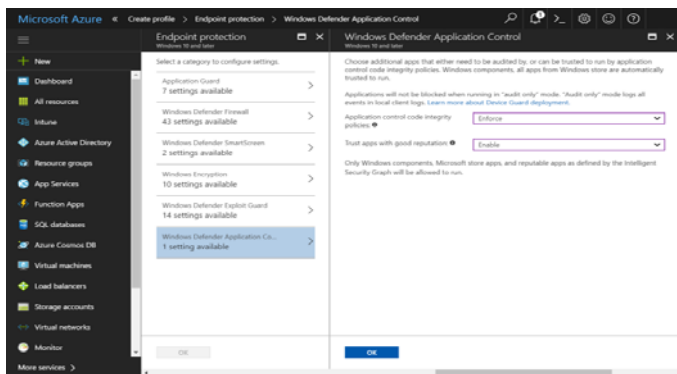
# No Code

## MICROSOFT 365: FERRAMENTAS DE DEPLOY, SEGURANÇA E COMPLIANCE

novas APIs oferece acesso intuitivo para extrair dados ou executar ações de resposta.

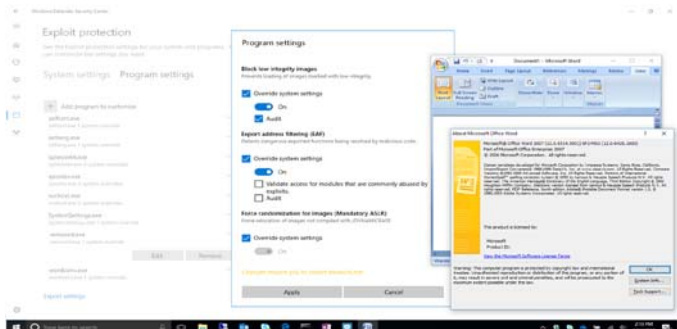


O **Windows Defender Application Control** é uma nova solução que aproveita a mesma tecnologia de controle de aplicações do Windows Defender Device Guard, enquanto remove os excessos de requisitos de hardware, drivers e de kernel. O WDAC pode ser executado em qualquer device com o Windows 10 Enterprise e usa o poder do Microsoft Intelligent Security Graph (ISG) para automatizar a lista de controle de aplicações, facilitando a sua gestão.

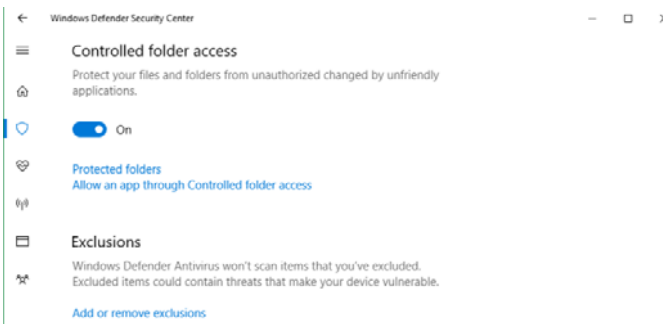


### Windows Defender Application Control

O **Windows Defender Exploit Guard** protege os dispositivos contra invasões, acesso a locais maliciosos na internet e ataques com recurso a conteúdos aparentemente inocuos, como documentos por exemplo. O WDEG inclui um vasto conjunto de mitigação de vulnerabilidades incorporado no Windows 10, reduzindo potenciais explorações das mesmas na plataforma e nas aplicações.

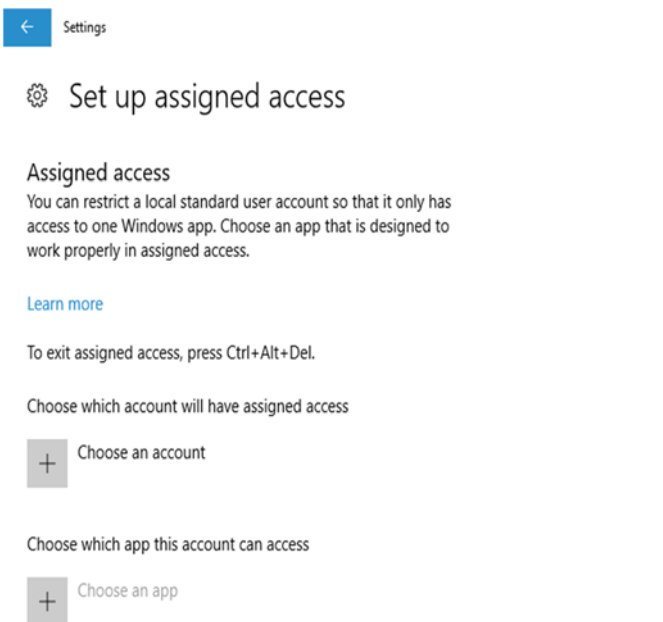


O **Controlled Folder Access** no Windows Defender Exploit Guard ajuda a proteger contra vírus e ransomware, permitindo que apenas aplicações confiáveis acedam pastas e documentos protegidos. Se uma aplicação não autorizada, mesmo uma nova variante de malware desconhecida, tentar aceder a dados protegidos, o acesso será negado, mesmo que ele consiga ultrapassar outras defesas.



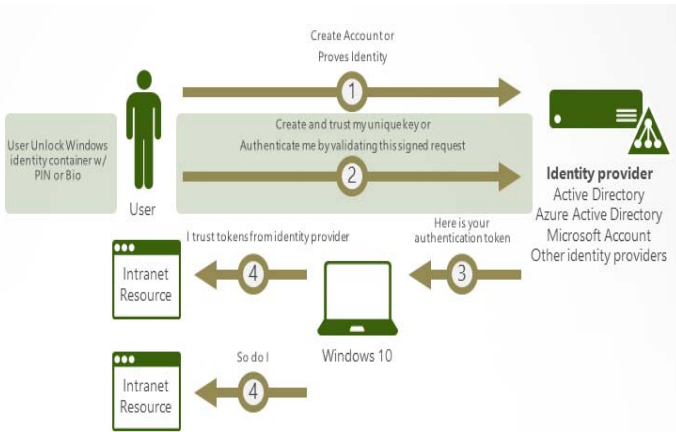
### Windows 10 Controlled Folder Access

O **Windows Assigned Access** possibilita a personalização e o bloqueio de dispositivos Windows na execução de tarefas específicas para os Firstline Workers e Quiosques; e as ferramentas baseadas na nuvem permitem que o IT faça a gestão desses dispositivos remotamente. As empresas agora podem configurar rapidamente e manter dispositivos bloqueados com recursos únicos e ferramentas melhoradas, permitindo o uso único de aplicações ou múltiplas aplicações.



### Windows 10 Assigned Access

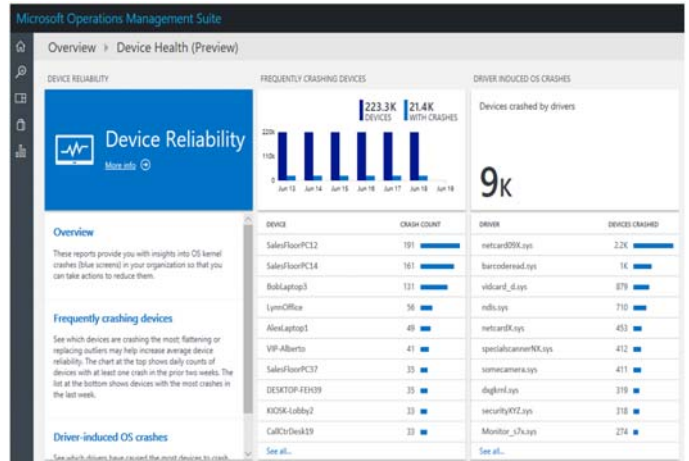
As melhorias do **Windows Hello**, incluindo novos fatores, como proximidade, localização, bloqueio dinâmico e suporte à Autenticação Intel, facilitam a implementação e o tornam mais seguro. As credenciais roubadas são uma das principais causas de violações, por isso, afastar-se das passwords deve ser prioridade máxima para cada organização.



### Ferramentas de ativação

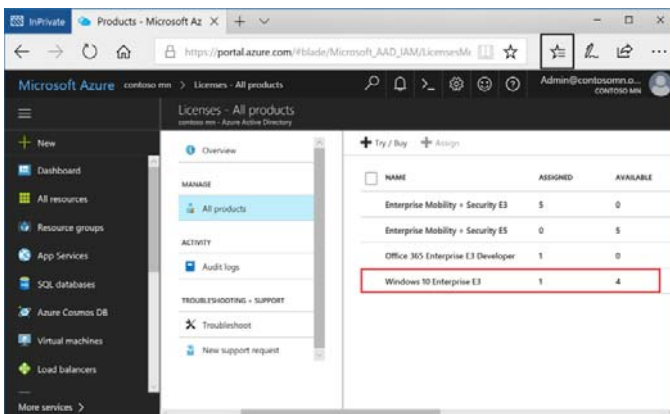
O **Windows 10 Subscription Activation** usa o AAD (Azure Active Directory), para facilitar a ativação por parte dos clientes e uso de todos os recursos no Windows 10 Enterprise. Este permite que os dispositivos Windows 10 Pro migrem de forma transparente para o Windows 10 Enterprise sem infraestrutura local, chaves de produtos ou reinicializações com base no *logon* de um utilizador ativo.

O **Device Health** foi adicionado ao [Windows Analytics](#) para ajudar a reduzir os custos de suporte, identificando e corrigindo de forma proactiva os principais problemas do dispositivo que afetam os colaboradores. Também é fornecido aos clientes empresariais uma nova configuração de diagnóstico para permitir que o Windows Analytics forneça informações críticas para a manutenção do ecossistema do dispositivo.



OMS Device Reliability

“ O **Windows AutoPilot** é um novo serviço na nuvem que permite que qualquer PC esteja preparado para o ambiente empresarial, com uma experiência simples para o utilizador e de fácil deploy para o departamento de IT. ”



Azure Licenses

### Ferramentas de análise e Compliance

Para auxiliar os clientes corporate na implementação de serviços e conformidade, a Microsoft disponibiliza ainda os seguintes recursos e ferramentas:

Recursos sobre a [Regulação Global de Proteção de Dados](#) (GDPR, na sigla em inglês) para ajudar os clientes a acelerar a conformidade com detalhes sobre recursos de segurança incorporados no Windows 10.



### Recursos

“ Os constantes ataques informáticos às organizações com o objetivo de subtrair dados ou provocar uma interrupção do negócio, levam as empresas e administradores de sistemas a ter um foco permanente na segurança e a procurar ferramentas que os ajudem a prevenir, proteger e responder a estas ameaças. ”

Nos links que se seguem, os administradores de sistemas poderão encontrar informações detalhadas que ajudarão na implementação e gestão de cada um dos recursos do

Microsoft 365 apresentados:

[Microsoft 365](#), [Crash course in Microsoft 365 Business](#)

[Microsoft 365 powered device lab kit](#)

[Windows AutoPilot](#), [Introducing Windows AutoPilot deployment](#)

[Windows Defender Application Guard](#), [Windows Defender Advanced Threat Protection](#)

[Windows Defender Application Control](#), [Windows Defender Exploit Guard](#)

[Controlled Folder Access](#), [Windows Defender Antivirus](#)

[Windows Assigned Access](#), [Windows Hello](#)

[Windows 10 Subscription Activation](#), [Manage Your Compliance from One Place – Announcing Compliance Manager](#)

[Beginning your General Data Protection Regulation \(GDPR\) journey for Windows 10](#), [Windows Analytics](#)

### Conclusão

Em suma, através da utilização Office 365, do Windows 10 e do Enterprise Mobility + Security, as empresas podem salvaguardar os seus ativos e os dispositivos que possuem nos parques, mesmo que não possuam grandes equipas de TI para fazer esta gestão. A subscrição deste conjunto de serviços poderá igualmente significar uma redução nos custos e uma simplificação do processo de licenciamento. Todos estes fatores ajudarão certamente à modernização e transformação digital das pequenas e médias empresas.

## AUTOR



Escrito por Nuno Silva

IT Professional | Windows Insider MVP | Microsoft MVP - Windows Experience (2014-2016) | Microsoft Technical Beta Tester (Windows International Team) | MCC | Certified Microsoft Windows Phone Expert | Windows Team Division Manager @ Microsoft Group Portugal (Facebook)





## Movimento Maker – Hands ON

No passado dia 27 de Janeiro de 2018 realizou-se em Castelo Branco mais um encontro do Movimento Maker e Hands On, cujo o tema foi o ESP8266 – Automação e Desenvolvidos de Dispositivos para automatizar a nossa casa.

O encontro teve lugar no Cyber Centro em Castelo Branco e cedo começaram a chegar os participantes, com toda a vontade de aprender e partilhar conhecimentos. A iniciativa teve o apoio da Camara Municipal e principalmente da Associação de Informática de Castelo Branco que organizou e dinamizou o espaço cedido. O nosso maker Bruno Horta foi o orador de serviço e ao longo do dia partilhou e ensinou aos presentes como tirar partido do ESP8266 – um pequeno microcontrolador que tem a capacidade de comunicação por wifi.

E desengane-se quem acha que este tipo de sessões é apenas para os makers mais experientes. No workshop foram abordados todos os níveis de conhecimento, desde ensinar os presentes desde os primeiros passos da instalação do Arduino IDE ao código mais complexo. Para ser um maker, apenas é necessário um requisito: “Querer aprender”!



Assim e para quem ainda não está familiarizado com este movimento, recomendamos a leitura da seção entrevista desta edição, uma vez que o entrevistado é o maker Bruno Horta. Mais uma vez agradecemos a todos os intervenientes da AICB, assim como a disponibilidade de todos os apoiantes desta iniciativa. Aqui na PROGRAMAR é com especial entusiasmo que participamos e divulgamos atividades deste género no interior centro de Portugal. Até à próxima!



## CRM – Plataforma ou Customizado

### Âmbito

Neste artigo quero levantar a problemática que passa muitas vezes pela nossa cabeça: devemos utilizar uma plataforma para um determinado objetivo onde já temos muita coisa implementada, ou devemos avançar para um desenvolvimento customizado para podermos ter as coisas como queremos e não termos que pagar licenciamento? Vou explorar esta temática tendo como exemplo prático, a implementação de um CRM.

### Introdução

Hoje em dia a sigla CRM tem estado em grande evidência, isto poderemos visualizar em vários estudos, do ponto de vista de investimento, onde se estima que o investimento em CRM ultrapassou o investimento em ERP. O que é bastante significativo porque uma empresa não pode viver sem um sistema de faturação enquanto um CRM não deveria viver sem um ERP. Outros estudos onde a procura de um CRM está no topo para as equipas de Marketing, Vendas ou Serviço. Por tudo isto esta problemática nunca foi tão pertinente.

### Descrição

*Customer Relationship Management* (CRM) tem como objetivo ter uma visão 360° sobre o cliente. Isto é, se um cliente entrar em contacto podermos em dois *clicks* ter tudo sobre o cliente e podermos rapidamente responder as suas questões e também tentarmos ser pró-ativos na resposta.

O conceito CRM tem evoluído ao longo dos tempos, e hoje em dia já não temos apenas os três habituais módulos, Vendas, Marketing e Serviço. Onde nas vendas temos a implementação do pipeline de vendas (Lead – Encomenda), no marketing a criação de listas de marketing e a gestão de campanhas e no serviço com a gestão do pós serviço, corretivo e/ou evolutivo. Hoje em dia já temos que gerir também as redes sociais, temos portais self-service para reduzir os custos de um call center, absorver e tirar partido do machine learning. Por isso a tecnologia tem que acompanhar estas evoluções.

Por isso temos que ter muito cuidado quando escolhermos o caminho de uma implementação de CRM customizado, ou seja, implementado de raiz na linguagem de programação e na plataforma que a nossa equipa se sente mais a vontade. Em oposição temos soluções onde por exemplo os três módulos mencionados já se encontram implementados. Não me estou a referir a uma plataforma de low code que não trazem nada implementado por defeito, quando por exemplo instalamos o Dynamics 365, podemos a partir do ponto 0 começar a trabalhar.

Esta discussão leva-me a uma das passagens mais significativas na minha carreira. Quando depois de dois anos a implementar uma plataforma de CRM completamente customizado em ASP .NET, para um grupo de telecomunicações. Dois anos e em conjunto com uma equipa que em média tinha 6-8 consultores. Foram dois anos em que nunca coloquei em questão o que estava a fazer. Quando saí do projeto concorri para uma vaga de Consultor CRM, foi quando me introduziram o Microsoft Dynamics CRM, tenho que confessar que a minha primeira reação foi de rejeição, não queria uma plataforma a trabalhar por mim.

Abreviando a história cheguei a conclusão que tinha passado dois anos a implementar uma plataforma de CRM e que não tinha implementado nem sequer 1% das funcionalidades que por exemplo o Dynamics 365 já resolveu. Basicamente foi reinventar a roda e com a garantia que em poucos anos essa aplicação iria cair no legado. Foi depois de ter passado por esta experiência que comecei a partilhar este meu conhecimento sobre como é a implementação numa plataforma e mais concretamente como era o ambiente do Dynamics 365 e foi por causa desta minha partilha de conhecimento que me tornei Microsoft MVP para Business Solutions (onde se insere o Dynamics 365), um prémio atribuído pela Microsoft a “embaixadores/especialistas” nesta tecnologia.

Claro que eu sei que esta versão customizada depois de “finalizada” não teremos que pagar uma licença mensal, que temos o código na nossa posse, mas será isto suficiente? Vamos ver as vantagens que uma plataforma traz consigo:

- Um menor investimento inicial, tanto monetariamente como temporal (Time to Market), visto que existe uma grande parte já implementada;
- A vantagem de ter uma comunidade por detrás destas plataformas, quando temos um problema temos mais hipóteses de este ser resolvido bem como aprender as melhores práticas que outras pessoas seguem;
- Um sistema low-code onde estamos principalmente focados em processos e regras de negócio (Sem necessidade de realizarmos tarefas repetidas, como por exemplo implementações de CRUD; login; audit; ...)
- Plataforma evolutiva em que não temos que nos preocupar com a evolução da nossa plataforma de desenvolvimento, é a equipa de produto que tem essa responsabilidade.
- Conjunto de mecanismos que facilitam a sua extensão – criação rápida de entidades; atributos; relações e formulários; Automatização através de processos guiados;



## CRM – PLATAFORMA OU CUSTOMIZADO

- Conjunto de artefactos que um consultor funcional consegue realizar sem a necessidade de recorrer a uma pessoa técnica e assim podermos fazer automações dentro de casa;
- Várias maneiras de desenvolvermos o nosso modelo de segurança, para que este seja flexível e forte;
- Acesso a plataforma através de vários clientes sem acrescentarmos uma única linha de código, interface web; interface web móvel; dispositivos móveis; cliente Outlook estas duas últimas com possibilidade de acesso offline;
- Reporting integrado através de gráficos; dashboards ou através de wizards. Ou recorrer a ferramentas de reporting, como exemplo Reporting Services e/ou PowerBI;
- Optarmos por uma solução cloud ou on-premise;
- Sistemas que são Multi-língua/moeda por defeito;
- Integração por defeito com email; ferramentas de produtividade, documentação entre outros;

Outro aspeto em ter em consideração é que por trás destas plataformas estão grandes players, como a Salesforce, Microsoft, SAP ou Oracle, estes players têm equipas a preparar futuras versões do produto e as suas equipas de investigação estão atentas as tendências do mercado e que implementam estas tendências por nós e algumas delas são adicionadas sem custos a plataforma. Alguns exemplos destas tendências:

- Ferramentas de inquéritos drag-and-drop e integrados com o CRM, para escutar a opinião dos nossos clientes\parceiros;
- Ferramentas de automatização do marketing, totalmente integradas;
- Serviços de Self-Service, onde o cliente não necessita de telefonar ou deslocar-se a uma loja, através de um portal, Facebook ou outra plataforma consiga resolver por si próprio o seu problema. Hoje em dia os serviços de self-service têm diminuído muito os custos em call-center;
- Integração em portais, para dar mais informações aos nossos clientes\parceiros sem a necessidade de desenvolvimento de customizado;
- Ferramentas de gamificação, outro tema muito em

voga nos últimos tempos, onde tentamos dar mais motivação\ferramentas a nossa equipa comercial;

- As ferramentas de colaboração têm sido um grande investimento, a colaboração entre a equipa e com os nossos parceiros é importante, e por isso vemos novas aplicações bem como a evolução das existentes.
- A capacidade de gestão de projetos, bem como depois o controlo da execução do projeto, têm sido áreas de intervenção.
- Capacidades sociais, a habilidade de escutarmos as redes sociais para vermos como a tendência da nossa marca ou dos nossos competidores.
- Machine learning, podermos incorporar mecanismos de inteligência artificial em processos dentro do nosso CRM, como por exemplo a sugestão de produtos ou sabermos antemão quais os clientes que se prevê perder ou quais as soluções mais prováveis para os problemas que temos.
- Para além destes pontos, os grandes players têm as suas plataformas de cloud que fornecem serviços que ajudam a implementação do próprio CRM como a expansão de novas aplicações. Estamos a falar de serviços cloud que o Azure e a Amazon fornecem.

Não podemos avaliar apenas pelos custos iniciais e/ou mensais. Pois a manutenção da plataforma bem como adaptar a plataforma a novas tendências terá um custo não mensurável e temos que avaliar o time-to-market, pois “perdermos” tempo a manter a solução para não ser um legado, significa que não estamos a melhorar a ferramenta para que o negócio não perca oportunidades.

### Conclusão

Espero que com este artigo fiquem com mais conhecimento sobre esta temática, que apesar de ter puxado mais para escolhermos a plataforma ao invés de uma solução customizada, nem sempre é a mais aconselhada e que terá que ser avaliado mediante o caso. Não quero é que por falta de conhecimento não se pondere este tipo de plataforma.

No próximo artigo venho aqui falar sobre as plataformas de CRM existentes no mercado e alguma das suas características.

### Links

- <https://mva.microsoft.com/en-US/training-courses/introduction-to-microsoft-dynamics-365-17593>

## AUTOR



Escrito por Pedro Azevedo

Formado no ISEL em Engenharia Informática, com mais de 14 anos de experiência em tecnologia, principalmente na plataforma .Net tive o privilégio de seguir desde a versão 1.0, cerca de 8 anos atrás o mundo das aplicações empresariais meteram-se a minha frente, tendo-me tornado especialista nas mesmas, principalmente CRM mas aprendendo ERP. Em 2013 tornei-me Microsoft MVP em Business Applications. Fundador da Comunidade Portuguesa Dynamics 365.

## SQL vs NoSQL

Hoje em dia cada vez mais as bases de dados estão a ocupar um lugar de destaque no nosso mundo tecnológico.

Queremos guardar os nossos dados, ter acesso aos mesmos o mais rápido possível e processa-los para termos respostas rápidas. Antigamente os dados guardados eram específicos, em “tabelas contentores”. Hoje, principalmente com Big Data, o nosso próprio telefone guarda tudo o que fazemos e praticamente tudo o que pensamos fazer. E perder “sessenta segundos” por uma resposta que queremos já torna-se cada vez mais impensável.

Por este motivo nesta edição decidimos recordar uma discussão antiga... SQL vs NoSQL...

Para os leitores que ainda não estão familiarizados com este tema, vamos por partes... A sigla SQL significa **Structured Query Language**, ou seja, linguagem de consulta estruturada. Criada no início dos anos 70 na IBM, esta linguagem tem uma forte inspiração na famosa álgebra relacional e devido à sua facilidade de aprendizagem e simplicidade, é sem qualquer dúvida, ainda a linguagem padrão mais utilizada quando falamos em bases de dados.

As bases de dados SQL respeitam o modelo relacional, pois baseiam-se no facto de todos os dados serem guardados em tabelas. Temos como principais exemplos de bases de dados SQL, o SQ Server, o MySQL, Sybase, Oracle ou IBM DM2.

Sem querer particularizar muito, podemos afirmar que no SQL temos “três grandes comandos” base... INSERT, UPDATE e DELETE. Com estes três facilmente temos acesso a qualquer registo de uma determinada tabela da base de dados uma vez que a base de dados é relacional e verticalmente escalonável. Tendo em conta que temos ao dispor chaves primárias e/ou estrangeiras, uma vantagem particular do SQL é sua cláusula JOIN simples, que nos permite facilmente recuperar dados relacionados armazenados em várias tabelas .

Ainda nesta temática, com o SQL, sendo a base de dados relacional temos sempre as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade). E são estas quatro propriedades que descrevem as principais características de uma transação em SQL ou seja:

- **Atomicidade** – Uma transação deve ter todas as suas operações executadas com sucesso, isto é, em caso de falha de alguma operação, é necessário proceder ao rollback.
- **Consistência** – Uma transação deve respeitar a integridade dos dados.
- **Isolamento** – Deve garantir-se o isolamento de uma operação para que não seja possível que dois utilizadores gravem alterações no mesmo registo ao mes-

mo tempo.

- **Durabilidade** – As transações com sucesso devem persistir na base de dados mesmo que haja alguma anomalia externa à mesma, ou seja, devemos sempre registar as mesmas em memória não-volátil.

Agora passemos ao NoSQL... NoSQL significa **Not Only SQL**. Estas base de dados têm características bem diferentes. São bases de dados não relacionais, e as propriedades que acabamos de referir (ACID) deixam de estar obrigatoriamente presentes. A designação NoSQL foi inicialmente utilizada no final dos anos 90 e voltou a ser reintroduzido no início do ano de 2009.

Este tipo de bases de dados foi criado pois o modelo relacional apresenta algumas limitações quando o volume de dados é elevado e as cargas de trabalho são grandes. Assim ao contrário das anteriores, são escalonáveis horizontalmente e geralmente não suportam instruções e operações de junção.

Como vantagens no uso do NoSQL podemos apontar:

- Como a base de dados não é relacional significa que as mesmas têm maior flexibilidade, sendo mais simples de gerir
- Escalabilidade mais simples através do suporte para MapReduce – por norma as bases de dados são projectadas para funcionar eficazmente mesmo com hardware de baixo custo.
- A maior parte deste tipo de base de dados é open-source – temos como principais bases de dados NoSQL – MongoDB, MarkLogic, Couchbase, CloudDB e Dynamo DB da Amazon.
- Não é necessário desenvolver um modelo de dados tão detalhado como no paradigma relacional – o que nos permite poupar mais tempo no seu desenvolvimento

Contudo, temos que apontar também algumas das desvantagens do seu uso:

- Não tem uma comunidade ainda bem definida apesar de estar a crescer
- Faltam ferramentas de relatório para análises e testes de desempenho
- Falta de padronização – ainda não existe uma linguagem padrão como o SQL sendo que caso seja necessário migrar ou unificar um sistema podem existir problemas que ainda não estão documentados.

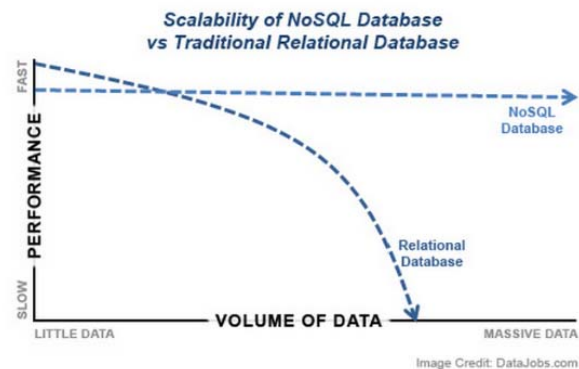
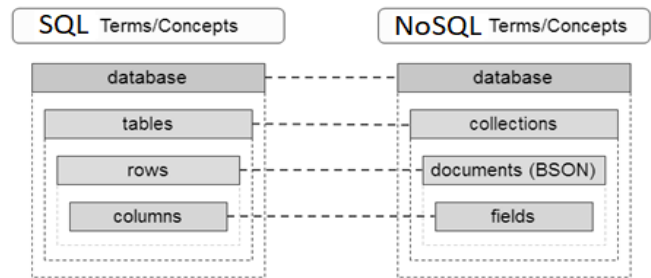
Posto isto e sem querer entrar em grandes repetições...

O NoSQL foi criado para ter uma performance melhor e uma escalabilidade mais horizontal para dar resposta às necessidades onde as bases de dados relacionais não são tão eficazes. Regra geral, podemos afirmar que temos 4 tipos de bancos de dados NoSQL:

- **Documento** – Os dados são armazenados como documentos. Os documentos podem ser descritos como dados no formato de chave-valor, como por exemplo, o padrão JSON. (um exemplo deste tipo é a MongoDB)
- **Colunas** – Os dados são armazenados em linhas particulares de uma tabela no disco, podendo suportar várias linhas e colunas. (Um exemplo deste tipo é a base de dados Cassandra)
- **Grafos** – Os dados são armazenados na forma de grafos (sim... grafos... com vértices e arestas) (como por exemplo, a base de dados Neo4j)
- **Chave-valor** – Este é o tipo NoSQL que aguenta maior volume de dados, pois o conceito é que um determinado valor seja acessível através de uma chave identificadora única. (um exemplo é a base de dados Riak)

Desta forma, os dados não estruturados (como artigos, fotos, dados de redes sociais, vídeos ou conteúdo dentro de um post de um blog) podem ser armazenados num único documento que pode ser facilmente encontrado, mas não é necessariamente categorizado em campos como numa base de dados relacional.

Na minha opinião, a principal diferença nos bancos de dados NoSQL é que toda a informação é agrupada e guardada no mesmo registo. Já no SQL precisávamos de ter uma relação entre as várias tabelas para ter acesso toda a informação. Lá está... a diferença prática entre verticalmente escalável ou horizontalmente escalável.

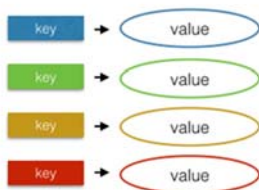
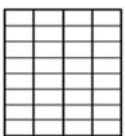


É importante referir que o NoSQL não veio para substituir o SQL, mas sim para oferecer uma alternativa mais flexível ao suporte de dados. Podemos sempre usar ambas as soluções para diferentes casos de uso. Por isso, o mais comum em soluções escalares de sucesso é a utilização de uma arquitetura híbrida, aproveitando o melhor dos dois modelos.

SQL

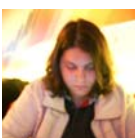
Key-value

Document



**Em suma**, no conceito de modelo relacional (SQL) baseamo-nos em que todos os dados são guardados em tabelas. No modelo não-relacional (NoSQL) não se aplica o conceito de *schema*: uma chave de valor é utilizada para recuperar valores, conjunto de colunas ou documentos.

**AUTOR**



Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010. Embaixadora das Geek Girls Portugal – Núcleo de Lisboa.



## Entrevista a: Bruno Horta

**Revista PROGRAMAR (RP):** Fala-me um pouco de ti e do teu percurso na tecnologia.

**Bruno Horta (BH):** Bem aos 14 anos os meus pais deram-me o meu primeiro computador, até a data apenas tinha mexido um Spectrum 128k de uns amigos, no entanto depois daquele dia que recebi o meu computador e podia explorar o mesmo 24 horas por dia, foi isso mesmo que fiz, a maioria das noites nem dormia, depois disso nunca fiz nada que não estivesse relacionado com tecnologia, aos 16 anos era programador de Html, Javascript e Pearl, depois passei para o hardware onde tive cerca de 10 anos, fazia reparações ao nível da eletrónica, depois de terminar a licenciatura voltei de novo com toda a força para o mundo da programação, mas agora com conhecimento em ambas as partes que constituem um computador podia fazer maravilhas e criar o meu próprio hardware com o meu próprio software... agora sim o mundo é maravilhoso.

**RP:** Quais as maiores dificuldades que encontras no dia-a-dia como maker?

**BH:** Ter tempo para fazer todos os meus projetos, adoro ajudar os outros e por vezes deixo os meus projetos para trás.

**RP:** Como surgiu o Movimento Maker Portugal?

**BH:** Bem... há cerca de dois anos atrás vi um anúncio de um evento que ia decorrer em Lisboa, sobre Internet das Coisas, esse evento era organizado pelo Pedro Rosa e se olhássemos para a Agenda fazia inveja a qualquer super evento de tecnologia, isto todo incluído num bilhete de acesso grátis para toda a gente. Achei aquela atitude fantástica e depois de ter ido ao primeiro e ao segundo e ver que existia tanta gente que partilhava o mesmo gosto de criar equipamentos tecnológicos como eu, decidi criar um grupo que não fosse apenas para Programadores, Engenheiros, Eletrotécnicos, Pintores, Designers mas sim para toda a gente que gostasse de criar as coisas que imaginavam e que o conseguissem fazer nas nossas casas, todo o processo de criação de um produto envolve muitas áreas e nem sempre as conhecemos todas por isso o Movimento tem o objectivo de unir malta com conhecimentos de forma a desbloquear áreas em que não estamos tão a vontade.

**RP:** Com a comunidade a crescer, que planos para o futuro?

**BH:** Fazer mais eventos com um formato que quem vá da



segunda vez possa continuar com o que aprendeu no primeiro, criar um site bastante dinâmico cheio de tutoriais, onde toda a informação boa que é colocada neste momento no Facebook possa estar bem organizada e simples de ser consultada.

**RP:** Quais os projetos que mais te marcaram?

**BH:** Foi um controlador de luzes para um rapaz que teve um acidente e ficou em cadeira de rodas, é certo que muitas das vezes fazemos Gadgets apenas porque é fixe e para não levantarmos o rabo do sofá, mas esses pequenos dispositivos na posse de

peças que precisam mesmo, podem ser como pão para a boca e melhorar em 500% a qualidade de vida dessa pessoa, no fundo era apenas um microcontrolador Wi-Fi com 16 relés e um App para Android, algo que fiz em menos de um dia e facilitou muito a vida a esse Maker ☺, pois ele já andava a tentar fazer algo mas não tinha conhecimentos de programação em micro electrónica.

**RP:** Na tua opinião, quais os "requisitos" para se ser um Maker?

**BH:** Persistência, paixão e dedicação, sem isso não se cria nada de novo.

**RP:** Sendo do conhecimento dos makers, que as horas de dormir são algo de que te esqueces, pedia-te que comentasses o Hackathon Code! [*Here's to the hackers. The tinkers. The makers. The ones who create things quickly. While some may see them as sleep-deprived loons, we see dedication. Because those who are crazy enough to think they can make something awesome in just 36 hours, are the ones who do.*] (*Hackathon Code*) <translation> Aos hackers. Aos tinkers. Aos Makers. Aos que criam coisas rapidamente! Enquanto alguns podem vê-los como lunáticos, privados de sono, nós vemos neles dedicação! Porque aqueles que são loucos o suficiente para pensar que podem fazer algo incrível em apenas 36 horas, são os que o fazem!"(Código Hackathon) </translation>.

**BH:** Podia ter isso escrito por todo o meu corpo se tivesse coragem de fazer uma tatuagem ☺, é a mais pura das verdades, quando estamos perto de resolver um problema o sono deixa de existir as horas essas sim passam como segundos, mas acredita em 36 horas se existir dedicação, a loucura dos deuses da tecnologia entra no nosso sistema e o mais certo é após 24 ou 36 termos algo de fantástico que muita gente pensa não ser possível fazer em meses.



**RP:** Como vê a evolução das gerações futuras?

**BH:** Aqueles que se dedicam são super inteligentes, eu vejo malta com os seus 16 ou 17 anos com o conhecimento brutal, coisa que quando eu tinha a idade deles nem pensava ser possível, o acesso à tecnologia está logo perto deles desde que nascem, a Internet tem tudo o que precisam para aprender algo em meia dúzia de horas, por isso se eles aproveitarem o conhecimento que lhes é deixado pela nossa geração, vamos ter outro grande salto na tecnologia.

“ (...)quando estamos perto de resolver um problema o sono deixa de existir as horas essas sim passam como segundos, mas acredita em 36 horas se existir dedicação, a loucura dos deuses da tecnologia entra no nosso sistema e o mais certo é após 24 ou 36 termos algo de fantástico que muita gente pensa não ser possível fazer em meses. ”

**RP:** Pedia-te para comentar as estatísticas que indicam uma falta de balanceamento de género no mercado da tecnologia, onde 83% das posições seniores tendem para o masculino, e apenas 17% para o feminino? (Fonte: ITU [International Technological University]).

**BH:** Bem isso é algo que quero inverter, tenho feito forcing para comunidades femininas se juntarem a nós, pois temos tanto a aprender com elas, eu próprio em casa a minha esposa apresenta-me problemas e resoluções que coisas que ela deteta que podem ser resolvidas com a utilização de tecnologia, existe um lado do mundo que nós não conseguimos ver e precisamos das senhoras para dar o próximo passo.

**RP:** Como Maker sempre a “iotizar” tudo, pedia-te que comentasses o que esperas da revolução do IoT!

**BH:** Bem eu gostava de dispositivos que não precisassem de eletricidade ☺, para comunicar, se isso não for possível para já ao menos que se comece a pensar em IA para IoT.

**RP:** É amplamente comentado que o Arduino, apesar de ter começado como um projecto académico, permitiu um crescimento exponencial da quantidade de projectos de IoT que ficaram disponíveis até para os mais inexperientes. Esperas que apareça um “novo arduino”, que revolucione ainda mais o IoT e potencialmente o “internet of everything” e não apenas a internet das coisas?

**BH:** Bom eu espero que o meu OnOfre seja o próximo Arduino para IoT no que toca a domótica, mas sim vão aparecer cada vez mais plataformas prontas a carregar código e ficarem logo disponíveis a utilizar.

**RP:** Escusado será falar nas sucessivas comparações entre o Raspberry Pi e o ZX Spectrum, o pequeno computador de 8bits em que tanta gente se iniciou! Na tua opinião de Maker, o raspberry pi será o “ZX Spectrum” de uma geração ?

**BH:** Talvez eu não seja a melhor pessoa para falar sobre o nosso pequeno Pi devido às várias más experiências que tive com ele, eu vejo um Pi como uma prova de conceito já o ZX foi algo fenomenal que marcou a nossa adolescência, tendo em conta a época em que foi lançado ele fazia tudo e mais alguma coisa, cumprindo com o que foi projectado.

**RP:** Por fim e como estamos no início de um ano, algum prognóstico tecnológico para o resto do ano ?

**BH:** Vai ser um ano de loucos ☺, o Movimento Maker está cada vez mais unido e mais forte, por isso muitas coisas boas devem acontecer ☺. Em relação ao IoT as marcas estão cada vez a apostar em dispositivos mais pequenos e simples de usar por isso vamos conhecer novos produtos e quem sabe se o OnOfre não chega às prateleiras como verdadeiro Switch for Makers.

*A revista PROGRAMAR agradece ao Bruno, a gentileza e o tempo dispendido nesta informal entrevista, que vos trazemos na quinquagésima nona edição desta vossa publicação*



## PHYTON: HISTÓRIA E ASCENDÊNCIA

**Resumo:** Este artigo aborda a história e ascendência da linguagem de programação *Python*, destacando a sua ascendência diante da comunidade de desenvolvedores, uma vez que a mesma encontra-se em constante adesão sendo implantada e implementada em diversos sistemas, hardwares e aplicações com uma proposta simplificada e que colabora significativamente com os recursos tecnológicos disponíveis na atualidade.

### 1. INTRODUÇÃO

Nesta última década é perceptível a grande evolução da tecnologia, bem como sua democratização e acessibilidade que tem se adaptado à sociedade de maneira eficiente atendendo desde profissionais da área em situações de extrema complexidade, como usuários leigos e deficientes visuais onde, através de aplicações de comando de voz tem-se a possibilidade de agregar usuários com tal singularidade.

Neste contexto as linguagens de programação agregam a possibilidade de comunicação entre hardware e usuário através de rotinas, scripts, funções e comandos que propiciem a integração de ambos para que assim o recurso tecnológico seja utilizado de maneira coerente a atender as mais diversas demandas.

Assim descobre-se a linguagem *Python* como uma grande aliada destas soluções intermediárias onde, em pouco tempo, já possui grandes adesões de sucesso.

Este artigo apresenta o contexto histórico da linguagem, seus fundamentos, aplicações e implementações de destaque que permitem com que a mesma seja bem aceita e utilizada no mercado.

### 1. FUNDAMENTOS DA LINGUAGEM PYTHON

De acordo com Luzzardi (2017):

*Python é uma linguagem de programação interpretada (onde cada linha é compilada e executada, uma por vez), ou seja, não é compilada (onde todo o texto é traduzido para linguagem de máquina e posteriormente executado).*

Ele também a descreve como uma “linguagem de alto nível e um software livre”, não sendo necessário pagar para sua utilização, podendo ser utilizada em aplicações de criação de scripts para Redes de Computadores. 3

Já a PyScience-Brasil define-a como uma linguagem de programação com o objetivo de trazer um código flexível e que sua manutenção fosse fácil podendo-se destacar como as principais características:

I. linguagem pseudo-código executável com o uso mínimo de caracteres especiais;

II. marcação de blocos por indentação;

III. uso de palavras-chave desnecessário nas compilações;

IV. gerenciador de memória automático através do coletor de lixo;

V. suporta técnicas mais complexas como orientação a objetos;

VI. é uma linguagem livre onde os desenvolvedores podem estudar e modificar o código dela;

VII. não possui um propósito específico como Java e PHP, podendo ser utilizada para atender diversas necessidades;

VIII. multiplataforma, funciona em vários sistemas e dispositivos;

IX. o pacote de instalação possui um *batteries included* que a torna mais completa, possui biblioteca com diversas classes, métodos e funções;

Além disso, possui uma filosofia própria denominada *Zen of Python*, traduzida por Rocha (2018), os quais possuem as seguintes “frases” para que o programador possa refletir acerca do desenvolvimento de suas aplicações que pretende realizar:

I. Bonito é melhor que feio;

II. Explícito é melhor que implícito;

III. Simples é melhor que complexo;

IV. Complexo é melhor que complicado;

V. Linear é melhor do que aninhado;

VI. Esparsos é melhor que densos;

VII. Legibilidade conta;

VIII. Casos especiais não são especiais o bastante para quebrar as regras;

IX. Ainda que praticidade vença a pureza;

X. Erros nunca devem passar silenciosamente;

XI. A menos que sejam explicitamente silenciados;

XII. Diante da ambiguidade, recuse a tentação de adivinhar;

4

XIII. Deveria haver um — e preferencialmente só um — modo óbvio para fazer algo;

XIV. Embora esse modo possa não ser óbvio a princípio a menos que você seja holandês;

XV. Agora é melhor que nunca;

XVI. Embora nunca frequentemente seja melhor que já;

XVII. Se a implementação é difícil de explicar, é uma má

## PHYTON: HISTÓRIA E ASCENDÊNCIA

ideia;

XVIII. Se a implementação é fácil de explicar, pode ser uma boa ideia;

XIX. Namespaces são uma grande idéia — vamos ter mais dessas.

### 2. SOBRE O CRIADOR DA LINGUAGEM

Conforme biografia relatada pelo vlogger Guanabara (2017), o criador da linguagem *Python* é *Guido van Rossum*, holandês nascido na cidade de *Haarlem* em 31 de janeiro de 1956. Formado em Matemática na *University of Amsterdam*, conheceu as linguagens ALGOL 60, Pascal e Fortran.



Figura 01. Guido van Rossum recebendo o prêmio COMMIT AWARD 2013.

Ele também destaca que Guido concluiu seu mestrado na mesma Universidade e iniciou sua carreira trabalhando voluntariamente com a linguagem ABC, criada por Lambert Meertens, o qual percebeu seu potencial e contratou-o como implementador onde permaneceu por 4 anos. Porém não era o que ele queria, portanto dedicou-se à criação da linguagem *Python*, desde 1989, tendo-a como um "projeto de *hobby*". Ele escolheu este nome pois a empresa onde trabalhava remetia o nome de seus projetos a programas, séries ou qualquer outra designação que remetesse à televisão onde, num momento de humor "irreverente" em sua vida, escolheu este nome pois era fã da série britânica *Monty Python's*

*Flying Circus*, porém *Python* também é um gênero de serpentes da família *Pythonidae*, que inclusive estampa a capa do primeiro livro publicado sobre a linguagem pela editora *O'Reilly* e esta editora sempre usa, estampado na capa de seus livros publicados, a imagem de um animal.

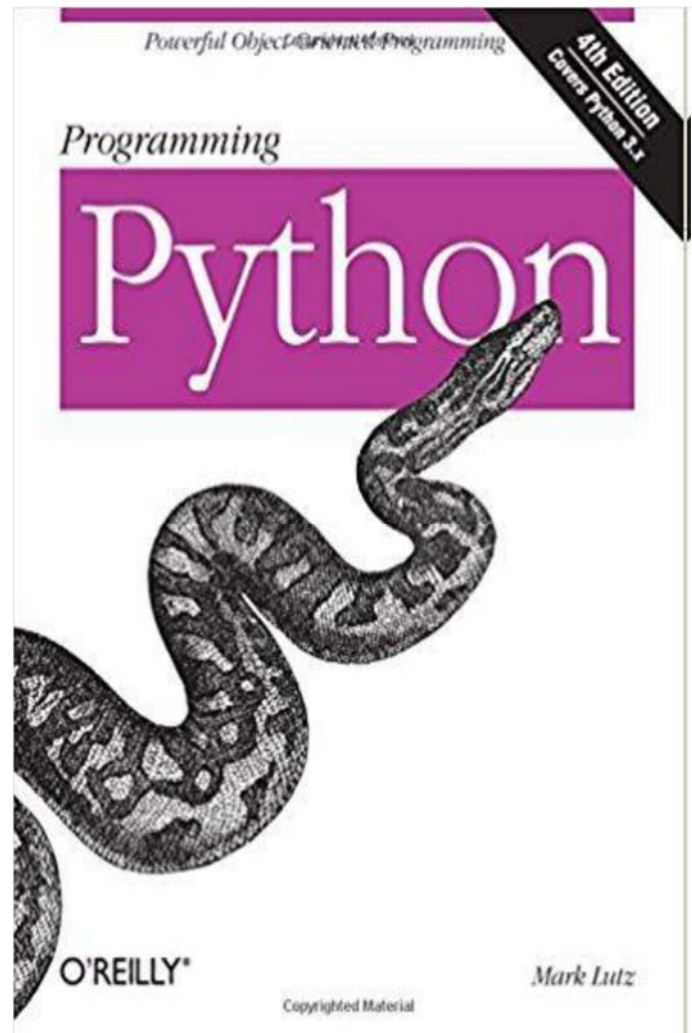


Figura 02. Capa do livro que trata da linguagem Python, 4ª edição, publicado pela editora O'reilly.

Ainda de acordo com o Vlogger, Guido atuou em vários institutos de pesquisa, mas em 1999 submeteu uma proposta de financiamento à *Defense Advanced Research Projects Agency* (DARPA), desenvolvedora da *ARPANET* posteriormente sendo sucedida pela atual *internet*. Nesta submissão ele definiu seus objetivos, permitindo com que a linguagem se popularizasse, onde, devido ao crescimento da popularização da linguagem, foi instituída, em 2001, a *Python Software Foundation*, que possui estampada em sua logomarca duas cobras entrelaçadas. Esta fundação é sem fins lucrativos e possui uma grande comunidade aberta a quaisquer alterações que são avaliadas e, se aprovadas pela maioria, é implementada na próxima versão. Em pouco tempo de existência, a fundação já é patrocinada por grandes empresas como: *Microsoft*, *Google* e *Globo.com*.

# No Code

## PHYTON: HISTÓRIA E ASCENDÊNCIA



Figura 03. Logomarca da Python Software Foundation.

Guanabara (2017) também informa que, em 2002 recebeu, em *Bruxelas*, capital da Bélgica, o *Free Software Foundation Award for the Advancement of Free Softwares* (Prêmio para Avanço do Software Livre da Fundação para o Software Livre), o qual, desde 2001, premeia grandes pessoas que contribuem para este objetivo. Este prêmio foi lhe conferido durante conferência da FOSDEM (*Open Source Developers European Meeting*). De 2005 a 2012, trabalhou para o *Google*, mas a partir de Janeiro de 2013 passou a compor a equipe da *Dropbox* onde atualmente residindo em *Belmont, Califórnia*, também estando a frente e atuando ativamente da *Python Software Foundation* reconhecido mundialmente por sua colaboração neste ramo com grandes prêmios recebidos nos mais diversos eventos de tecnologia e ciência com menções honrosas pelo legado deixado, também pela abertura em relação a comunidade e demais colaboradores sendo considerado uma inspiração para muitos profissionais da área. 7

O Portal OráculoTI (2017), publicou uma entrevista com Guido, o qual deu uma reportagem especial em comemoração ao 100º (centésimo) podcast do canal *Talk Python to me*, onde revelou que a página *Pypi.org* já disponibiliza cerca de 96.000 (noventa e seis mil) pacotes para importação demonstrando o crescimento e a força da comunidade aderente à linguagem.

### 3. ASCENDÊNCIA E ADESÃO: EXEMPLOS DE APLICAÇÃO E IMPLANTAÇÃO

Guanabara (2017), informa as grandes utilizações que já encontram-se implantadas e sendo utilizadas em relação a linguagem *Python* mostrando sua grande adesão por parte de alguns sistemas operacionais que vêm pré-instalados suas bibliotecas e funcionalidades para programação, tais quais: *Amiga*, *NetBSD*, *MacOS*, *Linux*.

Já dentro dos projetos de ambientes computacionais existentes e que possuem em sua essência a implementação da linguagem *Python*, o Vlogger destaca o *Raspberry Pi* que, desenvolvido no Reino Unido, é um microcomputador do tamanho de um cartão de crédito que possui um sistema operacional que permite com que o equipamento conecte-se a televisores ou monitores, com o objetivos educacionais.



Figura 04. Microcomputador Raspberry Pi.

Zuliani (2016), destaca o famoso game *Minecraft* que possui uma implementação educativa que utiliza esta tecnologia *Raspberry Pi* com o intuito de ensinar as pessoas a programar. Esta implementação é denominada *Minecraftpy* e funciona através da conexão entre um cliente ao servidor que, através de um *plug-in*, libera o cliente para modificação do jogo através da linguagem *Python*, conforme fluxograma apresentado na Figura 05.

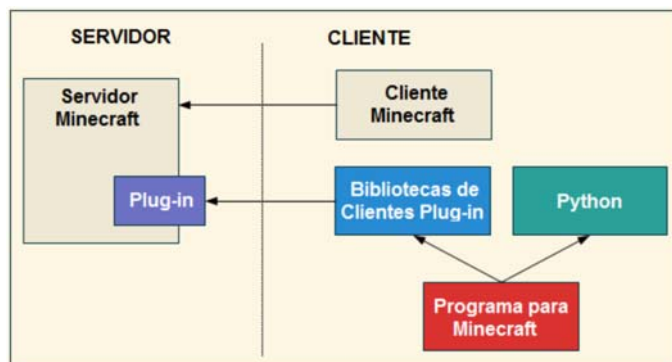


Figura 05. Fluxograma do Minecraftpy

Guanabara (2017) também relata a sua aplicabilidade em algumas especificidades como evolução de sistemas de inteligência artificial, biotecnologia, computação 3D pois esta linguagem consegue suportar números grandes e realizar cálculos mais complexos com os mesmos. Ele também destaca algumas empresas e softwares livres que utilizam *Python* em suas aplicações como: *Zope* (em seu gerenciador de conteúdo); *Air Canada* (em suas reservas de passagens e assentos aéreos); *BitTorrent* (totalmente elaborada nesta linguagem); *Globo* (em seus streamings); *Google* (em seus *crawlers* utilizados para buscas na internet); *Youtube* (totalmente feito nesta linguagem); *NASA*; *Industrial Light & Magic* (distribuidora do *Star Wars*, utiliza a linguagem em suas aplicações gráficas); *Autodesk* (na aplicação *Maya* e *Softimage* para aplicações de efeitos visuais e animações em 3D); *Blender* e *Gimp* (programas de manipulação de imagens e animações dinâmicas em 3D). 9 Apesar da adesão estar se ampliando cada vez mais, a linguagem ainda está caminhando ao aprimoramento, sendo perceptível através das adequações que são realizadas pela comunidade que já está se expandido pelo mundo com êxito e sucesso onde são implementadas, inclusive alguns autores remetem a *Zen of Python* como fundamental para utilização na rotina do desenvolvedor de qualquer linguagem de programação, detendo desta forma o reconhecimento global e científico desta iniciativa.

### 4. CONCLUSÃO

A linguagem de programação *Python* está em constante desenvolvimento/aprimoramento em relação à comunidade aderente, com grande fluxo de aperfeiçoamento, bem como as sugestões para ampliação de sua usabilidade no mercado tecnológico que, com muita competência, vem implementando em situações de sucesso as quais pode ser acompanhada no cotidiano.



### 5. REFERÊNCIAS

BORGES, Luiz Eduardo. **Python para desenvolvedores**. 2ª Edição, Rio de Janeiro. 2010. Disponível em: <[https://ark4n.files.wordpress.com/2010/01/python\\_para\\_desenvolvedores\\_2ed.pdf](https://ark4n.files.wordpress.com/2010/01/python_para_desenvolvedores_2ed.pdf)> . Acessado em 05 de fevereiro de 2018.

COMMIT Official. **Update COMMIT/AWARD 2013**. 2013. Disponível em: <<http://www.commit-nl.nl/news/update-commitaward-2013>>. Acessado em 06 de fevereiro de 2018.

GUANABARA, Gustavo. **Curso Python #02 – Para que serve o Python?** 2017. Disponível em: <<https://www.youtube.com/watch?v=Mp0vhMDI7fA>>. Acessado em 06 de fevereiro de 2017. **10 em de programação Python**. 2017. Disponível em: <<http://pluzzardi.w.pw/Python.pdf>>. Acessado em 18 de fevereiro de 2018.

PORTAL ORÁCULOTI. **Guido van Rossum fala sobre o pas-**

**sado, presente e futuro do Python**. 2017. Disponível em: <<https://oraculoti.com.br/2017/03/04/guido-van-rossum-fala-sobre-o-passado-presente-e-futuro-do-python/>>. Acessado em 05 de fevereiro de 2018.

PYSCIENCE-BRASIL. **Python: O que é? Por que usar?**. Portal Wikidot. Disponível em: <<http://pyscience-brasil.wikidot.com/python:python-oq-e-pq>>. Acessado em 05 de fevereiro de 2018.

ROCHA, Zeno. **Lições aprendidas com o mundo Python**. 2011. Disponível em: <<https://zenorocha.com/licoes-aprendidas-com-o-mundo-python/>>. Acessado em 06 de fevereiro de 2018.

ROSSUM, Guido Van. **Personal home page**. Disponível em: <<https://gvanrossum.github.io/>>. Acessado em 05 de fevereiro de 2018.

ZULIANI, Gianfranco. **Programmazione in Minecraft**. 2016. Disponível em: <<https://gzuliani.bitbucket.io/minecraft/minecraft-python.html>>. Acessado em 06 de fevereiro de 2018.



### AUTOR



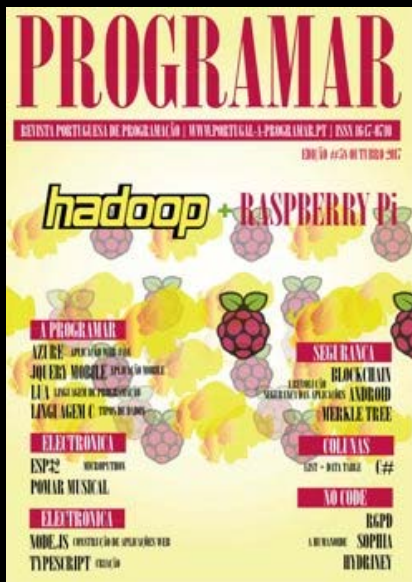
Escrito por Danilo Morais da Silva

Academico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas Centro Universitário da Grande Dourados - UNIGRAN

<http://lattes.cnpq.br/7159986007035372>

# Veja também as edições anteriores da Revista PROGRAMAR

58ª Edição - Novembro 2017



57ª Edição - Julho 2017



56ª Edição - Maio 2017



55ª Edição - Março 2017



54ª Edição - Janeiro 2017



53ª Edição - Agosto 2016



e muito mais em ...  
[www.revista-programar.info](http://www.revista-programar.info)



**DUVIDAS?**

**IDEIAS?**

**AJUDAS?**

**PROJECTOS?**



**portugal-a-programar**  
•org

