



**IPG** Politécnico  
|da|Guarda  
Escola Superior  
de Tecnologia e Gestão

# RELATÓRIO DE ESTÁGIO

Curso Técnico Superior Profissional em  
Testes de Software

Tiago Alberto Pais Cecilio

julho | 2016





**Escola Superior de Tecnologia e Gestão**

Instituto Politécnico da Guarda

---

# RELATÓRIO DE ESTÁGIO

---

TIAGO CECILIO

RELATÓRIO PARA A OBTENÇÃO DO DIPLOMA DE TÉCNICO

SUPERIOR PROFISSIONAL EM TESP TESTES DE SOFTWARE

Junho/2016 (conclusão do estágio)

JUNHO DE 2016

GLOBAL DELIVERY CENTRE - ALTRAN

Tv. de São Pedro 20, Fundão

## Índice Geral

Índice de Ilustrações.....	2
Acrónimos.....	3
Elementos Identificativos.....	4
Agradecimentos.....	5
Resumo.....	6
CAPÍTULO I.....	7
1. Introdução.....	8
1.1 Descrição da estrutura do relatório.....	8
1.2 Objectivos.....	8
1.3 Local de estágio.....	9
1.3.1 Altran Portugal.....	10
1.4 Testes de <i>Software</i> .....	11
1.4.1 Princípios dos Testes de <i>Software</i> .....	12
<b>1.4.2 Tipo de testes de software</b> .....	14
.....	15
<b>1.4.2.4 Testes de regressão</b> .....	16
CAPÍTULO II.....	18
2. Participação em projetos.....	19
<b>2.1 GDF Suez</b> .....	19
<b>2.2 Projecto Smart Transitions</b> .....	21
<b>2.2.3 Empresa Medicity</b> .....	22
<b>2.3 Daikin ITC</b> .....	25
Conclusão.....	38
Bibliografia.....	39

## Índice de Ilustrações

Figura 1 - Processo de Teste .....	13
Figura 2 – Testes Unitários [1].....	14
Figura 3 - Testes de Sistema [1].....	15
Figura 4 - Testes de Integração [1].....	15
Figura 5 - Testes de Regressão [1] .....	16
Figura 6 - Testes de Aceitação [1] .....	16
Figura 7 - Testes Funcionais [1].....	17
Figura 8 - Testes de Segurança [1].....	17
Figura 9- Países em que a Altran está inserida.....	9
Figura 10- Altran Fundação [1] .....	10
Figura 11 – Foliage [1].....	21
Figura 12 – Medicity [1] .....	22
Figura 13 - Test Plan .....	24
Figura 14 – Daikin [1].....	25
Figura 15 - Metodologia Agile [1] .....	28
Figura 16 - Design Página Swagger [3] .....	32

## Acrónimos

Tesp – Técnico Superior Profissional

GDF – Gaz de France

ISTQB - International Software Testing Qualifications Board

## Elementos Identificativos

### **Aluno**

Tiago Alberto Pais Cecilio

Número: 1011856

Tesp: Testes de *Software*

### **Estabelecimento de Ensino**

Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

Morada: Av. Dr. Francisco Sá Carneiro nº50  
6300-559 Guarda

Tel.: +351271220100

Fax: +351271222690

### **Instituição Acolhedora do Estágio:**

Nome: Global Delivery Center – Altran

Morada: Tv. de São Pedro 20, Fundação

### **Duração do Estágio:**

Início: 01 de Março de 2016

Fim: 30 de Junho de 2016

### **Orientador de Estágio:**

Nome: Engº Pedro Manuel Pinto Teixeira

Grau académico: Mestre em Computação Móvel

### **Gestor de Projeto**

Nome: Hugo Firmino

Cargo: Gestor Projeto Daikin ITC

## Agradecimentos

Em primeiro lugar, gostaria de agradecer a todas as pessoas que me apoiaram e que fizeram com que o meu estágio fosse possível de ser realizado na Altran: reconheço a importância da instituição que muito contribuiu para a concretização do meu estágio, o Instituto Politécnico da Guarda, na qual está envolvida a Escola Superior de Tecnologia e Gestão.

Agradeço, ao meu orientador de estágio Prof. Pedro Pinto pelo seu profissionalismo, simpatia e orientação.

Queria agradecer a toda a equipa da GDF Suez/Angie por todo o apoio que tiveram para comigo durante o tempo em que lá estive.

Aos meus colegas da equipa da Foliage, nomeadamente à Cristiana Vieira e ao Vítor Oliveira por todo o apoio, dedicação, conselhos durante o período que fui integrado na mesma.

À equipa da Daikin ITC que me ajudou na aquisição de determinados conceitos, principalmente ao André Matos e ao Hugo Firmino, sendo o gestor de projeto e de estágio.

Um agradecimento especial ao meu colega João Lopes pelo apoio e companheirismo.

Queria deixar o meu sincero obrigado, há minha família por todo o apoio constante que tiveram ao longo do meu percurso.

Por fim, mas não menos importante, agradecer a todos os docentes do curso que frequentei em especial, Prof. José Fonseca e Prof. Clara Silveira que em muito contribuíram para o meu enriquecimento na área das Redes Informáticas, um muito obrigado a todos por todo o apoio, atenção e profissionalismo.

A todos um muito obrigado por todo o apoio e amizade que demonstraram.

## Resumo

O presente relatório descreve o estágio curricular realizado no ano do Curso Tesp de Testes de Software da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda.

O estágio realizou-se na Empresa Altran Fundão. Este estágio teve a duração de 4 meses com início marcado para o 01-03-2016 e com término previsto para o dia 30-06-2016.

Durante o período de estágio curricular foram desenvolvidas várias atividades das quais se destacam a realização de planos de teste e a sua aplicação.

Para além das atividades referidas, que não constam no plano de estágio, foram ainda desenvolvidas outras atividades na área da formação, nomeadamente o evento *Ignite Your Future* e a *Formação de Francês*.

## CAPÍTULO I

## 1. Introdução

O estágio curricular visa consolidar os conteúdos adquiridos ao longo do percurso académico e permite aplicá-los em contexto de trabalho.

A fim de estabelecer a ligação entre conteúdos aprendidos e as atividades desenvolvidas, é elaborado o relatório de estágio. Este é realizado no âmbito da unidade de Estágio Curricular do Curso Tesp Testes de Software, o qual teve a duração de 4 meses com início marcado para o 01-03-2016 e com término previsto para o dia 30-06-2016.

### 1.1 Descrição da estrutura do relatório

O presente relatório está dividido em três capítulos. No primeiro capítulo, denominado de Introdução será feita uma apresentação da empresa *Altran*, o enquadramento teórico dos conteúdos utilizados e serão ainda apresentados os objetivos deste estágio.

No segundo capítulo, Projetos, serão explicados os projetos em que estive inserido, os seus objetivos e respetivo enquadramento. De modo a finalizar será apresentada a Reflexão Final com o objetivo de elaborar uma reflexão, geral, sobre a elaboração deste relatório.

### 1.2 Objectivos

Os objetivos para este estágio serão:

1. Integração na equipa do projeto
2. Conhecer os objetivos da equipa
3. Conhecer os objetivos da equipa de testes no projeto
4. Conhecimento de todos os conceitos base inerentes à área de testes de software
5. Criação de planos de testes funcionais (Manuais) de acordo com as metodologias Foliage/Altran
6. Execução do plano de testes definido para o projeto utilizando as ferramentas definidas no âmbito do projeto
7. Comunicação de defeitos (erros) de forma estruturada e concisa
8. Diferenciação de mindset de *tester*.

### 1.3 Local de estágio

O Grupo Altran, é uma empresa multinacional francesa e líder global em inovação e na consultoria de alta tecnologia, tendo como objetivo ajudar empresas nos seus esforços para criar e desenvolver novos produtos e serviços.

Fundada em 1982, a Altran tem vindo a prestar serviços nos sectores aeroespacial, automóvel, energético, ferroviário, financeiro, na área de saúde e telecomunicações.

Opera em mais de 20 países da Europa, América e Ásia. (Figura 9)



*Figura 1- Países em que a Altran está inserida*

Em 1996, criou uma nova unidade independente, a Fundação Altran para a Inovação. Esta tem como missão: "Promover a inovação tecnológica para o benefício humano" e como objetivos: promover o surgimento de ideias criativas, apoiar o desenvolvimento de ideias inovadoras para que os projetos avancem mais rapidamente. É considerado um veículo para o compromisso cívico e procura mobilizar os recursos do grupo em torno e projetos cívicos e inovadores.

### 1.3.1 Altran Portugal

A operar em Portugal desde 1998, consolidando-se em 2009, a Altran Portugal, é uma das principais empresas na Consultoria de Inovação e Tecnológica em Portugal. Está presente em vários setores de atividade como o Financeiro, Telecomunicações & Media, Administração Pública, Indústria, Energy & Life Sciences, Intelligent Systems e Utilities, sendo que a sua atividade tem uma estrutura assente na venda de soluções inovadoras.

Está geograficamente distribuída por três cidades: (Figura 10)

- ✓ Lisboa
- ✓ Porto
- ✓ Fundão

Criado no Fundão, distrito de Castelo Branco, em 2013, o *Global Delivery Centre* é um núcleo de inovação tecnológica que suporta projetos globais, garantindo a prestação de soluções de valor acrescentado desenvolvido por engenheiros portugueses e internacionais.



*Figura 2- Altran Fundão [2]*

## 1.4 Testes de *Software*

Um dos grandes desafios para a Engenharia de *Software* tem sido desenvolver *software* de qualidade tendo em conta o prazo, o esforço e os custos estabelecidos. Ao mesmo tempo que se requer *software* cada vez mais complexo, o mercado exige produtos de maior qualidade. Nesta direção, as empresas que desenvolvem *software* têm-se preocupado cada vez mais com a qualidade dos produtos que criam.

Teste de *software* é o processo de execução de um produto para determinar se ele atingiu as suas especificações e funciona corretamente no ambiente para o qual foi projetado. O seu objetivo é revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipa de desenvolvimento antes da entrega final.

A realização de rigorosos testes aos sistemas de *software* e à documentação podem reduzir o risco de ocorrência de problemas durante o processo de testes e, deste modo, contribuir para o aumento da qualidade do sistema de *software*, caso os defeitos detetados sejam corrigidos previamente à produção do sistema para uso operacional.

Por meio de testes é possível medir a qualidade do *software* no que respeita a deteção de defeitos, relativos a requisitos contratuais, legais, ou normas específicas de indústria e características funcionais e não funcionais de um sistema, por exemplo fiabilidade, usabilidade, eficiência, manutenção e portabilidade.

A quantidade exata de testes a executar está dependente do nível de risco, incluindo riscos de segurança, técnicos e empresariais, e restrições do projeto, designadamente fatores como tempo e orçamento.

Os testes devem fornecer informações claras e suficientes sobre o *software* ou sistema a ser testado, de forma a contribuir para a tomada de decisão das partes interessadas sobre a passagem à próxima fase de desenvolvimento ou mesmo para a definição da data de entrega do produto final ao cliente.

As atividades de teste incluem as ações efetuadas antes e após a execução do teste propriamente dita. Estas atividades implicam o planeamento e controlo do projeto, a definição das condições de teste, a conceção e execução dos casos de teste, a análise dos resultados, a avaliação dos critérios de saída, a comunicação contínua com os *stakeholders*, mantendo deste modo informadas todas as partes envolvidas sobre a evolução do processo de testes e do sistema sob teste, e finalização das atividades de

encerramento, assim que alguma das fases de teste tenha sido concluída. Os testes também incorporam a revisão de documentos (incluindo o código fonte) e a realização de análise estática.

Os testes de *software* têm como principais objetivos a detecção e prevenção de defeitos no *software*, a aquisição de confiança sobre o nível de qualidade do sistema e o fornecimento de informações relevantes para tomadas de decisões por parte das *stakeholders*.

#### 1.4.1 Princípios dos Testes de *Software*

Ao longo dos últimos 40 anos foi definido um conjunto de princípios de teste que oferecem linhas de orientação gerais comuns a todos os tipos de testes.

O **primeiro princípio** enuncia que os testes devem prestar evidências da presença de defeitos no sistema, não tendo como finalidade demonstrar que o funcionamento do sistema decorre sem anomalias. A não detecção de *bugs* não é um indicador da ausência dos mesmos no sistema.

O **segundo princípio** indica que não é possível efetuar testes exaustivos, ou seja, não é viável testar todos os aspetos referentes a um sistema em desenvolvimentos, considerando todas as combinações de entradas e pré-condições. A análise de risco e a priorização dos testes, concordantes com os custos e prazos do projeto, devem ser utilizadas para concentrar os esforços de teste e substituir a realização de testes exaustivos.

O **terceiro princípio** evidencia que se deve iniciar a fase de testes o mais cedo possível. Para detetar os defeitos precocemente, as atividades de teste devem ser iniciadas o mais cedo possível no ciclo de vida de desenvolvimento do *software* ou sistema, e devem estar centradas nos objetivos definidos.

O **quarto princípio** define que o esforço de testes deve ser proporcional à densidade de defeitos por módulo. Um número reduzido de módulos apresenta geralmente a maioria dos defeitos registados durante os testes de pré-produção, podendo ser ainda responsável pela maioria das falhas operacionais detetadas.

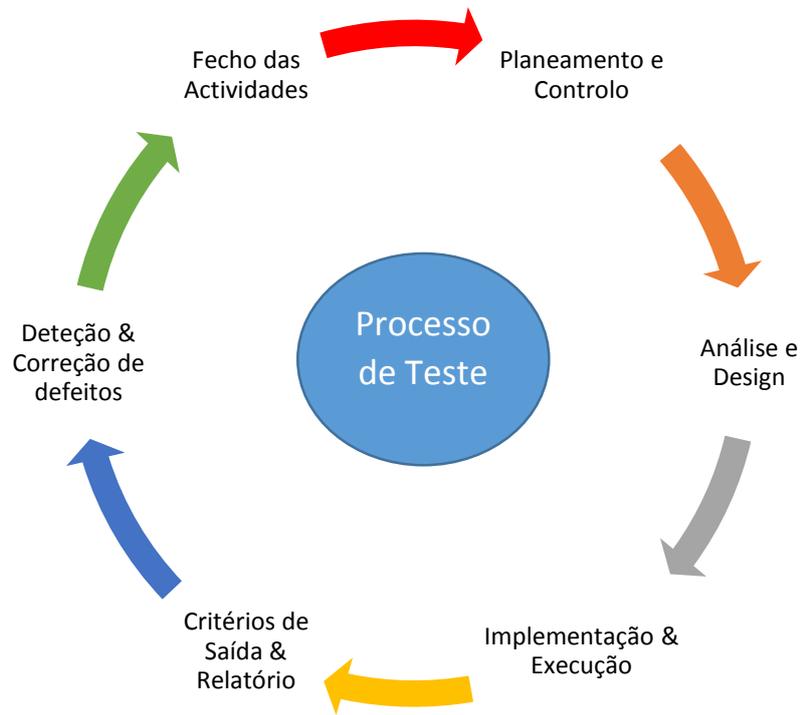
O **quinto princípio** refere que a repetição exaustiva dos mesmos casos de teste conduz à não detecção de novos defeitos. Para evitar que esta situação se verifique os casos de teste devem ser regularmente analisados e revistos e devem ser desenvolvidos novos

testes que avaliem diferentes partes do *software* ou sistema por forma a detetar potenciais defeitos.

O **sexto princípio** relata que os testes são dependentes do seu contexto e, como tal, são efetuados de forma diferente em diferentes contextos.

O **sétimo e último princípio** define que a deteção e correção de defeitos do sistema por si só não é suficiente se o mesmo não for utilizável e desenvolvido em conformidade com as necessidades e expectativas do cliente e dos seus utilizadores, tal como a apresenta

a figura nº1.



*Figura 3 - Processo de Teste*

## 1.4.2 Tipo de testes de software

Existem testes de **caixa preta**, testes de **caixa branca** e testes de **caixa cinza**.

Os testes de caixa preta ou testes funcionais, avaliam o comportamento externo do componente de software, sem se considerar o comportamento interno do mesmo, ou seja, testam todas as entradas e saídas desejadas, são executados em conhecimento do código e cada saída indesejada é visto como um *bug*.

Os testes de caixa branca, são do tipo estrutural, avaliam o comportamento interno do componente de software, ou seja são testes cujo objetivo é testar o código, onde um *tester* pode desenvolver códigos para realizar os testes de todas as aplicações.

Por fim, os testes de caixa cinza, são uma conjugação dos testes de caixa branca e preta, ou seja, envolve ter acesso a estruturas de dados e algoritmos do componente a fim de desenvolver os casos de teste, que são executados como na técnica da caixa-preta.

Existem vários tipos de testes das quais irei falar. Os vários tipos de teste são: **Testes Unitários**, Testes de Sistema, Testes de Integração, Testes de regressão, **Testes de aceitação**, **Testes Funcionais**, **Testes de Segurança**.

### 1.4.2.1 Testes unitários

Os testes unitários, são testes escritos e executados pelos programadores do software cujo objetivo é se certificarem que o código está de acordo com o seu design e requisitos e se funciona conforme o que é esperado.

Isso quer dizer que para qualquer função ou procedimento quando um conjunto de funcionalidades são dadas, deve em seguida receber os valores adequados.

Os testes unitários devem ser realizados antes dos testes de integração. (Figura 2)



Figura 4 – Testes Unitários [2]

### 1.4.2.2 Testes de sistema

Os testes de sistema são realizados depois dos testes de integração. São testes que validam a aplicação completa contra os requisitos. São testes executados automaticamente com a aplicação a correr no ambiente onde será usada. Garantem que as principais funcionalidades pedidas pelo cliente estão a funcionar para depois serem testadas pelos *testers*. (Figura 3)



Figura 5 - Testes de Sistema [2]

### 1.4.2.3 Testes de integração

Testes de integração é a fase de testes de software em que módulos são combinados e testados em grupo. Ocorrem após a realização de todos os testes unitários e precede os testes de sistema. O objetivo dos testes de integração é verificar os requisitos funcionais, de desempenho e de confiabilidade na modelação do sistema, em que é possível descobrir *bugs* de interface entre os componentes do sistema (Figura 4)



Figura 6 - Testes de Integração [2]

#### 1.4.2.4 Testes de regressão

Os testes de regressão são o tipo de testes que são executados a cada nova versão de determinado programa/aplicação/software por forma a determinar se as funcionalidades que antes eram executados corretamente passaram a conter erros. São os primeiros testes a serem efetuados a cada nova versão/release.

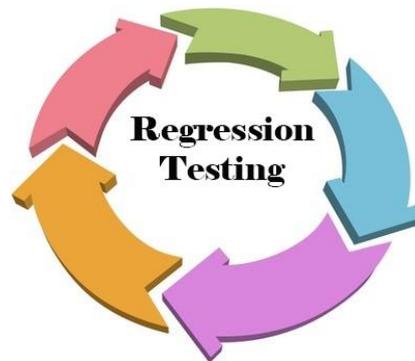


Figura 7 - Testes de Regressão [2]

#### 1.4.2.5 Testes de aceitação

Testes de aceitação são testes realizados com o objetivo de avaliar a qualidade externa do produto e se possível a qualidade em uso. Esses testes só podem ser feitos quando software está concluído e pronto para ser implementado. Deve ser feito, preferencialmente, pelo usuário final que participa no planeamento e realização dessa atividade. (Figura 6)



Figura 8 - Testes de Aceitação [2]

### *1.4.1.1 Testes Funcionais*

São testes que visam concluir se determinada funcionalidade cumpre com os requisitos e faz o que é pedido. Nestes testes existe sempre a necessidade de se comparar os resultados obtidos com os pretendidos. Estes testes são executados a cada entrada de uma nova funcionalidade. (Figura 7)



*Figura 9 - Testes Funcionais [2]*

### *1.4.1.2 Testes de segurança*

Os testes de segurança são testes cujo objetivo é encontrar falhas e fraquezas/vulnerabilidades em aplicações e serviços face a vários tipos de ataques de segurança. Estes testes devem ser executados periodicamente após a disponibilidade da aplicação ao utilizador final. (Figura 8)



*Figura 10 - Testes de Segurança [2]*

## CAPÍTULO II

## 2. Participação em projetos

No presente capítulo vamos abordar os vários projectos em que tive oportunidade de participar. Estes foram:

### 2.1 GDF Suez

GDF Suez é um grupo empresarial francês, que nasceu da fusão entre a Gaz de France e a SUEZ, em 22 de julho de 2008.

O grupo GDF Suez desenvolve os seus negócios (eletricidade, gás natural, serviços de energia e ambiente) baseado num modelo de crescimento responsável para vencer grandes desafios: satisfazer a procura de energia, assegurar a estabilidade do fornecimento, lutar contra as mudanças climáticas e maximizar a utilização de recursos. O grupo fornece soluções eficientes e inovadoras a particulares, cidades e empresas suportadas em fontes de gás diversificadas.

Atualmente a empresa mudou para o nome de Engie (Figura 11 e 12)



*Figura 11 – Engie [2]*

*Figura 12 - GDF Suez [2]*

#### 2.1.1 Equipa

A Equipa da Engie é composta por duas equipas: A equipa Rapsodie e a equipa Symphonie.

- Rapsodie – SAP Non-Regression Testing;
- Symphonie – SAP Non-Regression Testing

### ***2.1.2 Objetivo do projecto***

O objetivo neste projeto é fazer gestão de faturação do gás e da eletricidade, gestão de contratos da utilização do gás e da luz e a gestão dos clientes destes recursos.

### ***2.1.3 Introdução do Projeto***

Durante a participação neste projecto observei como executar testes para gestão de faturação e gestão dos contratos (gás, eletricidade).

A minha participação neste projecto teve apenas a duração de uma semana, tempo esse em que estive aguardar autorização/aceitação para o projecto da equipa Foliage

## 2.2 Projecto Smart Transitions

### 2.2.1 *Objetivo do projecto*

A equipa de testes da Foliage tem como objetivo a implementação de um Website que permitirá o agendamento e referênciação de consultas médicas entre os provedores de uma ou mais redes médicas. Tem como cliente final a Medicity

### 2.2.2 *Empresa Foliage*

Fundada em 1991, a Foliage é uma empresa com mais de 20 anos de experiência e com parcerias com empresas líderes no mercado ao nível da indústria Médica e da Ciências da Vida, Defesa e Aeroespacial, e Produção inteligentes. Foi adquirida a 100% pela Antran em 2004, cuja base fica em Burlington, Massachusetts. Tem também escritórios na Califórnia, Holanda e Índia.

As parcerias que estabelece com seus clientes são construídas com base na confiança mútua e do desejo comum de serem bem-sucedidos. (Figura 5)

Especificações:

- ✓ Product Development
- ✓ Technology Consulting
- ✓ Engineering Services
- ✓ Medical Devices and Systems
- ✓ Aerospace and Defense
- ✓ Intelligent Manufacturing Solutions
- ✓ Embedded Software
- ✓ Systems Engineering



Figura 11 – Foliage [2]

### 2.2.3 Empresa Medicity

A Medicity foi fundada em 1999 em Salt Lake City, Utah, Estados Unidos da América. Atualmente a população mundial é cada vez mais afetada pelas doenças crónicas, sendo que mais de dois terços dos custos de saúde são para o tratamento deste tipo de doenças.

Perante isto é cada vez mais importante que as organizações de cuidados de saúde têm a capacidade para identificar os grupos de pessoas com maior risco, tendo em vista a sua prevenção.

A Medicity é uma empresa líder de mercado que fornece a tecnologia que permite a troca de dados clínicos. Possui um sistema que permite analisar e agrupar os dados de saúde de diferentes organizações de cuidados de saúde relativos aos pacientes criando assim uma rede de informação que pode ser utilizada pelos profissionais de saúde para prevenção de doenças bem como a aplicação de tratamentos mais adequados.

A Medicity é uma empresa que cria uma base de dados para agrupar informação relativa aos doentes com este tipo de doença com o objetivo de prevenir que esta ocorra analisando potenciais fatores que possam vir a provoca-la. (Figura 14)



*Figura 12 – Medicity [2]*

### ***2.2.4 Trabalho Desenvolvido***

Como atividades realizam-se várias atividades: projeto cujo objetivo era a implementação de um Website que iria permitir o agendamento e referência de consultas médicas entre os provedores de uma ou mais redes médicas.

- ✓ Breve análise ao **ISTQB** (International Software Testing Qualifications Board) – é um manual de qualidade para *testers* de software.
- ✓ Análise da documentação relativa às funcionalidades do sistema
- ✓ Planos de Testes

Para obtenção das users stories relativas às sprints era utilizado o Website Smart Transitions para serem retiradas por alguém da equipa, pelo facto de eu não ter acesso.

### ***2.2.5 Ferramentas que usei no projeto***

Para a execução dos testes foi utilizado a ferramenta Excel.

#### ***2.2.5.1 Excel***

O excel é um poderoso programa de folha de calculo que serve para facilitar a gestão de vastos conjuntos de informação que envolvem principalmente o calculo de dependência entre os dados.

Utilizei a ferramenta excel para fazer planos de testes. Os planos de testes eram compostos por : US, TC Name, TC Steps, TC Step Description e TC Step Expected Results.

### 2.2.5.1.1 .Test Plan

É um documento onde descreve o escopo das atividades de teste. Tem a seguinte estrutura: (Figura 15)

1. ID\_User Storie(US) – Identificação da user stories
2. Test Case Name (TC Name) – Nome do Teste Case
3. TC Steps – Descrição de todos os passos para a execução do caso de teste
4. TC Step Description - Descrição do caso de teste que será executadoTC Step
5. Expected Result - Descrição do resultado esperado para cada caso de teste.

1	US	TC Name	TC Steps	TC Step Description	TC Step Expected Result
2	US67121	US67121 - User with sender permissions - User read sent referral and can view document contents associated with referral	Step1	Log into the ST application as a Sender with read-only permissions	User logged and lands in sending page
3	US67121	US67121 - User with sender permissions - User read sent referral and can view document contents associated with referral	Step2	User search and select the referrals	User founds and the referral opens
4	US67121	US67121 - User with sender permissions - User read sent referral and can view document contents associated with referral	Step3	User click on a document, from the Document Type section	Opens a view and user can view the document contents associated with referral
5					
6	US67121	US67121 - User with receiver permissions - User read sent referral and can view document contents associated with referral	Step1	Log into the ST application as a Receiver with read-only permissions	User logged and lands in receiving page
7	US67121	US67121 - User with receiver permissions - User read sent referral and can view document contents associated with referral	Step2	User search and select the referrals	User founds and the referral opens
8	US67121	US67121 - User with receiver permissions - User read sent referral and can view document contents associated with referral	Step3	User click on a document, from the Document Type section	Opens a view and user can view the document contents associated with referral
9					
10	US67121	US67121 - User with sender permissions - Clicks a document name and the document downloads	Step1	Log into the ST application as a Sender with permissions to access the referral	User logged and lands in sending page
11	US67121	US67121 - User with sender permissions - Clicks a document name and the document downloads	Step2	User search and select the referrals	User founds and the referral opens
12	US67121	US67121 - User with sender permissions - Clicks a document name and the document downloads	Step3	User clicks a document name	Using the standard browser control, the document is downloaded
13					
14	US67121	US67121 - User with receiver permissions - Clicks a document name and the document downloads	Step1	Log into the ST application as a Receiver with permissions to access the referral	User logged and lands in Receiving page
15	US67121	US67121 - User with receiver permissions - Clicks a document name and the document downloads	Step2	User search and select the referrals	User founds and the referral opens
16	US67121	US67121 - User with receiver permissions - Clicks a document name and the document downloads	Step3	User clicks on the document name	Using the standard browser control, the document is downloaded
17					
18					

Figura 13 - Test Plan

## 2.3 Daikin ITC

A Daikin cujo logotipo está apresentado na figura 14, enquanto empresa de comercialização de equipamento de climatização, está ciente da sua responsabilidade social e ambiental. O grupo Daikin, considera sua missão contribuir para o desenvolvimento sustentável, adotando e promovendo as melhores práticas ambientais e a utilização das melhores tecnologias disponíveis. (Figura 16)



*Figura 14 – Daikin [2]*

### 2.3.1 Objetivo

O projeto Daikin ITC, tem como objetivo a implementação de um produto na Cloud para a monitorização e controlo de dispositivos de climatização da marca.

### 2.3.2 Trabalho Desenvolvido

Neste projeto cujo objetivo é a implementação de um produto na Cloud para a monitorização e controlo de dispositivos de climatização da marca.

Como atividades realizei:

- ✓ Leitura da documentação relativa ao projeto Daikin ITC
- ✓ Visualização relativa à linguagem Gherkin
- ✓ Leitura das documentações relacionadas com as user stories das várias sprints
- ✓ Elaboração de casos de teste
- ✓ Elaboração de test steps
- ✓ Testes manuais nos browsers: Chrome, Firefox, IE11 e IE9.
- ✓ Testes de exploração

### 2.3.3 Equipa

Neste projeto a equipa é mista, sendo que há *developers* e *testers* no mesmo espaço e há uma colaboração com uma equipa da Altran Bélgica.

A equipa é composta por:

- Scrum Master (Project Manager PT)
- Senior Project Manager PT
- Net Developer PT
- Web Developers PT
- *Testers* PT
- Proxy Product Owner (Project Manager BE)
- Product Owner
- API Architect (BE)
- API Expert (BE)

### 2.3.4 Funcionamento do projeto

O projeto é dividido em ciclos que são chamados de Sprints. Uma sprint representa um tempo definido dentro do qual um conjunto de atividades deve ser executado.

No início de cada Sprint a equipa realiza uma Scrum meeting para definir o que vai ser executado durante essa sprint.

Todos os dias a equipa faz uma Daily Scrum Meeting com todos os colaboradores, cujo objetivo é dizerem o que foi feito no dia anterior e o que irão fazer no dia atual e se têm algum impedimento que está a atrapalhar o seu trabalho.

Depois de definido o que vai ser feito durante a sprint, os *developers* começam a implementar essas ações, enquanto que os *testers* fazem um test plan, para criar todos os casos de testes necessários e possíveis para executar todas essas funcionalidades. Depois de implementadas as funcionalidades, os *testers* vão começar a executar todos os casos de testes, em diferentes browsers, até tudo ficar bem testado.

No final de cada sprint, a equipa realiza uma sprint review meeting em que apresenta um demo com as funcionalidades implementadas em que a equipa mostra o que foi feito nessa sprint.

No final a equipa faz um Sprint Restrospective para identificar o que ficou a funcionar bem e o que pode ser melhorado.

### *2.3.5 Trabalho desenvolvido no âmbito do projeto*

#### 2.3.5.1 Elaboração de um Test Plan

Depois de definidas as funcionalidades que a equipa irá fazer durante a sprint, é feita uma reunião para definir os AC. A seguir a equipa de testes irá ler a documentação das user stories e com isso irá elaborar um test plan, com os vários casos de teste possíveis para testar uma funcionalidade. No test plan é inserido também o ID da user stories, as pré condições, o nome do casos de teste, os passos necessários para testar o caso de teste, e o resultado final.

#### 2.3.5.2 Enquadramento

**API** (Application Programming Interface ou Interface de Programação de Aplicativos) - é o conjunto de padrões de programação que permite a construção de aplicativos e a sua utilização de maneira não tão evidente para os usuários.

**Commit** – atualização das alterações no servidor

**Gherkin** – é uma maneira estruturada de descrever casos reais. Permite a geração de testes automáticos de aceitação.

## 2.7 Metodologia do Projeto

### 2.7.1 Metodologia Agile

A metodologia Agile têm como objetivo acelerar o desenvolvimento do *software* visando a melhoria contínua do processo, gerando benefícios como o aumento da comunicação e interação da equipa, organização diária para o alcance da meta definida, evitar falhas na elaboração, respostas rápidas às mudanças e aumento significativo da produtividade (Figura 17).

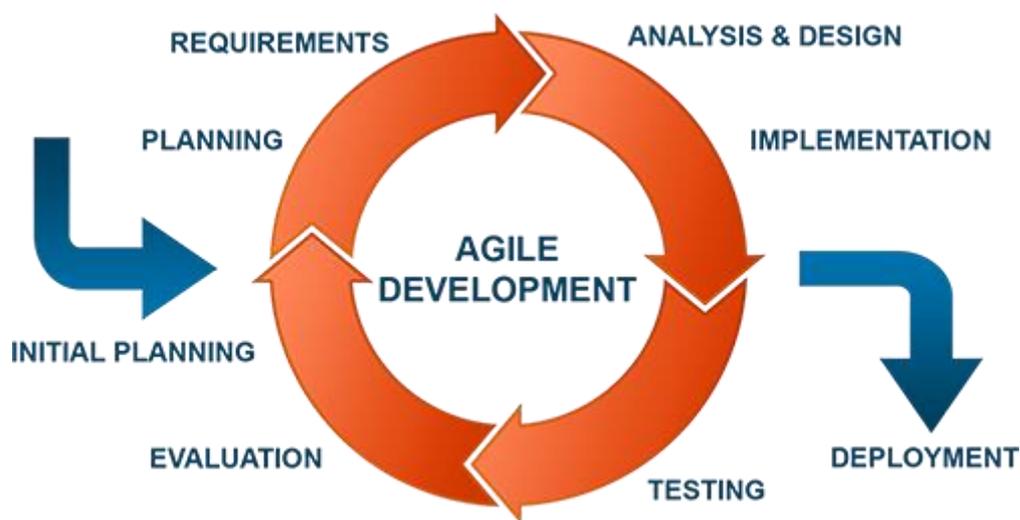


Figura 15 - Metodologia Agile [2]

### 2.7.2 Scrum

O Scrum é um processo de desenvolvimento iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de *software*. É utilizado para trabalhos complexos nos quais é impossível prever tudo o que irá ocorrer.

### 2.7.3 Sprints

No Scrum, os projetos são divididos em ciclos (tipicamente mensais) chamados de Sprints. O Sprint representa um tempo definido dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de *software* são iterativas, ou seja, o trabalho é dividido em iterações, que no Scrum são chamadas de Sprints e geralmente duram entre 2 a 4 semanas.

### 2.7.4 Backlog

É quando uma funcionalidade a ser implementada no projeto não passa na sprint devido a algum tipo de *bug*.

### 2.7.5 Quadro de trabalho

A equipe também pode possuir um quadro de trabalho para organizar as atividades dos itens da Sprint, separando-as em basicamente em 5 estados (isso pode variar de projeto a projeto). Neste caso os tipos de estados são:

- Blocker
- To Do e Re Do
- In Progress
- Tests
  - Manual Tests
  - Integration Tests
  - System Tests
- Verify
- Done

i. TO DO

- Analise da user stories conforme a sua descrição
- Visualização da informação relativa às funcionalidades pedidas nas user stories
- Elaboração dos critérios de aceitação (acceptance criteria)

ii. In Progress

- Desenvolvimento das funcionalidades
- Elaboração do test plan com todos os casos de teste
- Testes unitários

iii. Tests

- Testes Manuais
- Testes de Integração
- Testes de Sistemas
- Testes de exploração
- Descrição de *bugs*

iv. Verify

- Se for encontrado algum bug, é necessário os Developers proceder à sua correção
- Os *tester* devem garantir que tudo é testado depois da correção ser feita

v. Done

- Finalização da user stories

vi. Blocker

- ✓ Quando existe alguma user stories que por algum motivo de um bug não ter sido corrigido, fica em Blocker e passa para a próxima sprint

### 2.7.6 Daily Scrum Meeting

Diariamente, em uma Sprint, a equipa faz uma breve reunião com a duração do máximo de 15 minutos com todos os colaboradores, chamada Daily Scrum Meeting. O objetivo é cada colaborador dizer o que fez no dia anterior, o que pretende fazer no dia que se inicia e se existe algum impedimento que está atrapalhar o seu trabalho.

### 2.7.7 Sprint Review Meeting

No final de cada sprint, a equipa apresenta uma demo com as funcionalidades implementadas em uma Sprint Review Meeting onde a equipa mostra o que foi alcançado nessa sprint.

A seguir faz-se uma Sprint Restrospective para identificar o que funcionou bem e o que pode ser melhorado e a equipa inicia o planeamento da próxima Sprint.

### 2.7.8 Product Owner

O produto owner é responsável por maximizar o valor do trabalho que uma equipa faz.

- ✓ Define os requisitos do produto, decide a data de release e o que deve conter nela
- ✓ É responsável pelo retorno financeiro do produto
- ✓ Prioriza os requisitos de acordo com o seu valor de mercado
- ✓ Pode mudar os requisitos e prioridades a cada sprint
- ✓ Aceita ou rejeita o resultado de cada sprint

## 2.7.9 Scrum Team

- ✓ Multifuncional, entre 5-9 membros.
- ✓ Seleciona, entre os itens priorizados, os que irão ser executados durante a Sprint.

## 2.8 Ferramentas Usadas no âmbito do Projeto

### 2.8.1 Swagger

É uma ferramenta com especificação de recursos que permite entender os seus serviços sem o acesso ao código do servidor. O Swagger tem um quadro que permite que os *developers* e os *testers* consigam interagir com a API. (Figura 18)



Figura 16 - Design Página Swagger [4]

## 2.8.2 Microsoft Visual Studio

É um programa da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens de Visual Basic, C, C++, C#.

Também é um produto de desenvolvimento na área web.

As linguagens mais utilizadas são Visual Basic.Net (VB.NET) e C# (C Sharp).

A Figura 19 mostra o IDE de programação de .Net

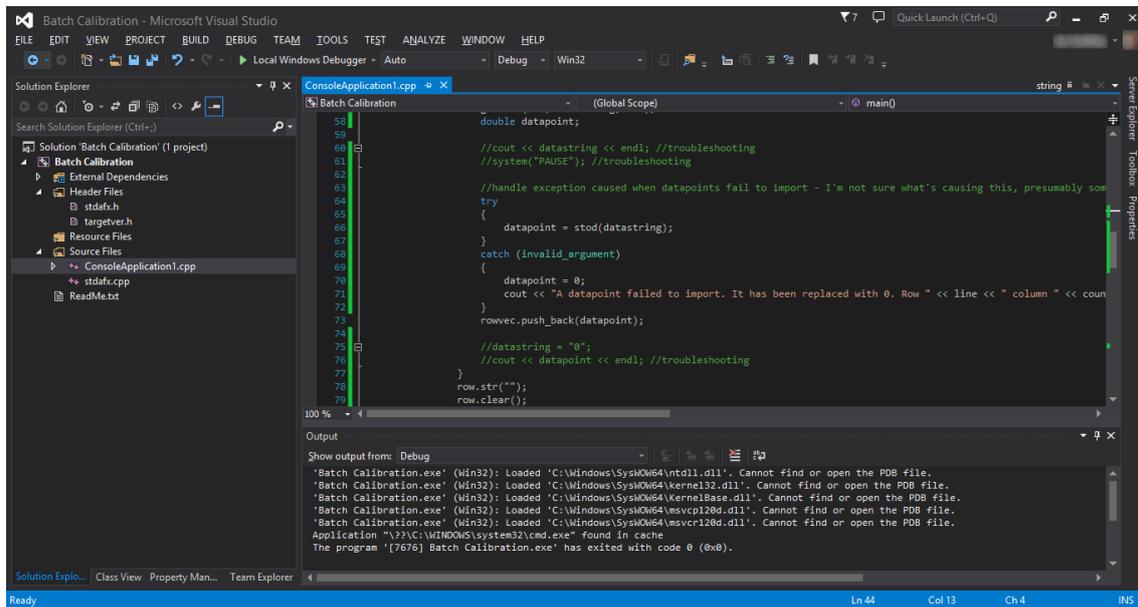


Figura 19 - Design Microsoft Visual Studio [2]

### 2.8.3 Jenkins

É um dos *softwares* utilizados quando se fala em "Integração Contínua", que é o responsável pelo desenvolvimento ágil, ou seja, a facilidade em detecção de *bugs* em códigos e o rápido *deploy*, em caso de falha por parte de um ou de um time de *developers*. Cujas vantagens são as seguintes: (Figura 20)

- ✓ *Builds* periódicos
- ✓ Testes Automatizados
- ✓ *Builds* em ambientes diferentes do *developers*
- ✓ Possibilita análise de código
- ✓ Reduzir retrabalho
- ✓ Reduzir custo da entrega
- ✓ Identificar erros mais cedo
- ✓ Fácil de operar e configurar
- ✓ Interface agradável

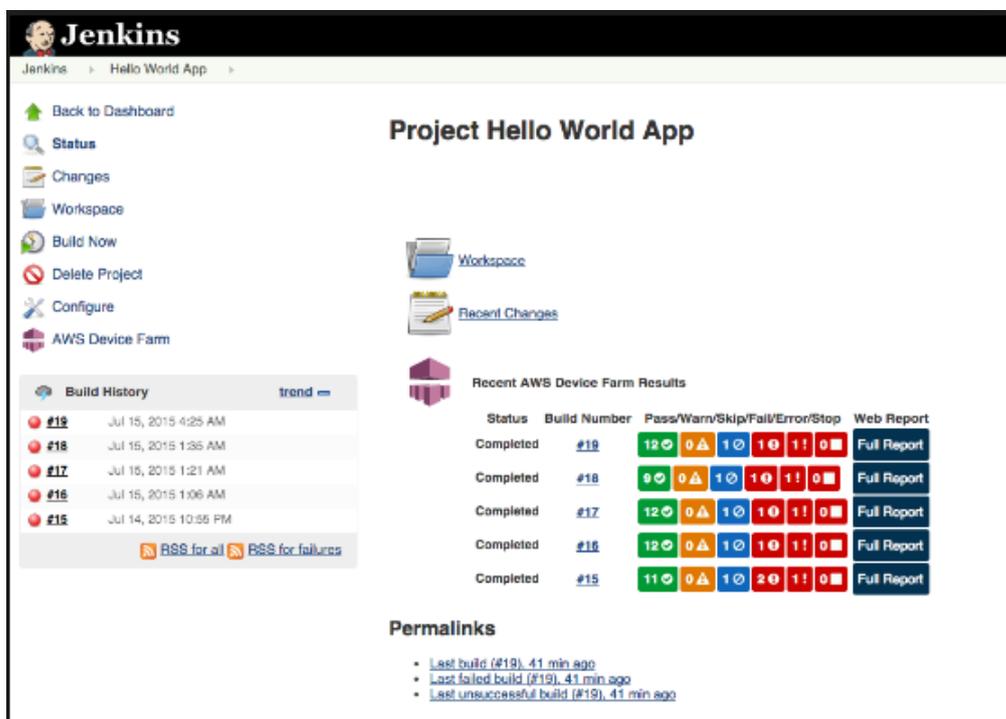


Figura 20 - Design Jenkins [3]

## 2.8.4 JIRA

O JIRA é um software comercial desenvolvido pela empresa Australiana Atlassian. É uma ferramenta que permite o monitorizar tarefas e acompanhamento de projetos garantindo a gestão de todas as suas atividades. O JIRA é desenvolvido em Java.

Um Projeto JIRA é um agrupamento de tarefas (Issues), e é definido de acordo com a organização e as suas necessidades. Exemplos de projetos JIRA:

Jira é baseado em Java EE que operam em várias bases de dados e sistemas operativos. A ferramenta também apresenta painéis de controlo adaptativo, filtros de pesquisa, estatísticas, RSS e email. A arquitetura flexível do Jira permite ao utilizador criar extensões específicas que podem ser incluídos na Jira biblioteca de extensão .

A Figura 21 apresenta a Interface da plataforma JIRA.

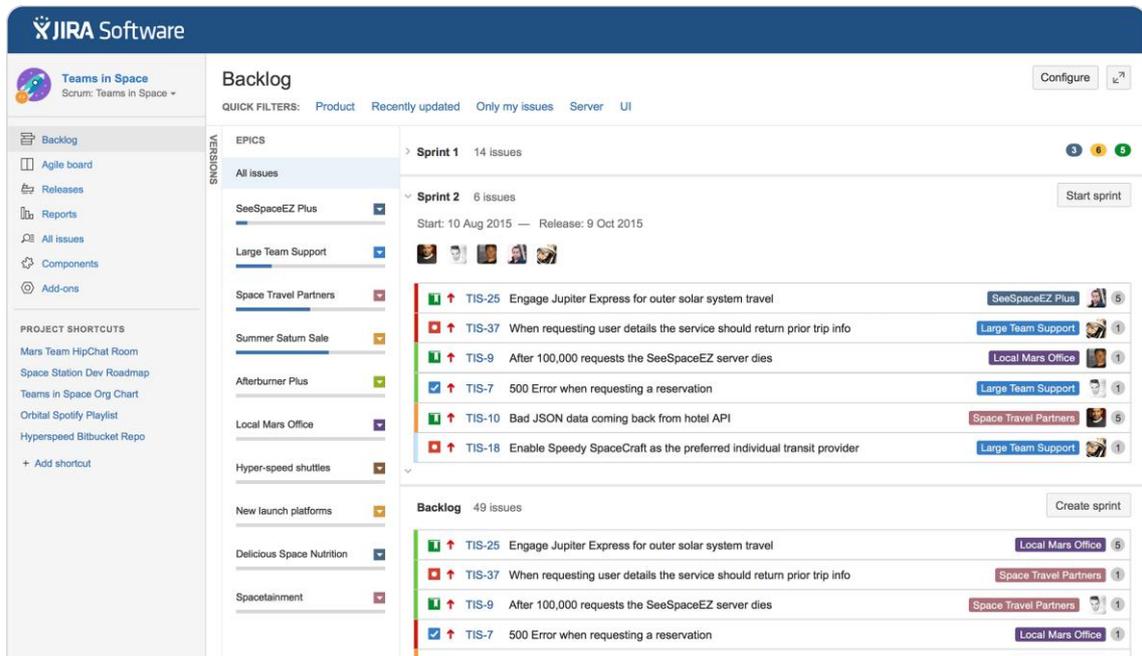


Figura 21 - JIRA Software [2]

## 2.9 Ferramentas usadas no projeto

### 2.9.1 Excel

O excel é um poderoso programa de folha de calculo que serve para facilitar a gestão de vastos conjuntos de informação que envolvem principalmente o calculo de dependência entre os dados.

Utilizei a ferramenta excel para fazer planos de testes e testes de exploração. Os planos de testes eram compostos por : ID, Test Case, Preconditions, Steps e Expected Results.

#### 2.9.1.1 Test Plan

É um documento onde descreve o escopo das atividades de teste. Tem a seguinte estrutura: (Figura 22)

6. ID\_User Storie(Witca) – Identificação da user stories
7. Test Case – Descrição do caso de teste que será executado
8. Preconditions – Descrição das pré condições
9. Steps – Descrição de todos os passos para a execução do caso de teste
10. Expected Result - Descrição do resultado esperado para cada caso de test

Test Cases					
WITCA	#	Test case	Preconditions	Steps	Expected result
WITCA-11	1	As an user (any role) I want to associate Management Points to an Area so that they can be controlled/monitored through the area		Scenario: Add a management point to a zone Given I am an user (any role) And a site is assigned to me And I have a zone already created When I drag and drop a management point into the created zone Then the zone has this management point associated	
WITCA-11	2	As an user (any role) I want to associate Management Points to an Area so that they can be controlled/monitored through the area		Scenario: Remove a management point from a zone Given I am an user (any role) And a site is assigned to me And I have a zone already created And I drag and drop a management point into the created zone When I drag and drop a management point out of the zone Then the zone does not have this management point associated	
WITCA-11	3	As an user (any role) I want to associate Management Points to an Area so that they can be controlled/monitored through the area		Scenario: Remove a zone with associated management points Given I am an user (any role) And a site is assigned to me And I have a zone already created And I drag and drop a management point into the created zone When I delete the created zone Then the management point is assigned to the top level zone	
WITCA-11	4	When there is no zones the MPs are allocated to the top level	1-Logged in as an user (any role) and site associated with MP 2-The user is in Building Administration page 3-Delete all zones	- Go to Equipment List and count the MPs - Go to Zones page - Check that the Management Points on the top level are the same on the equipment list	All the MPs are allocated to the top level!
WITCA-11	5	Check that one MP is present only in one zone	1-Logged in as an user (any role) and site associated with MP 2-The user is in Building Administration page 3-Delete all zones	- Go to Building Administration - Go to Zones page - Open a created parent area and associated him one Management Point	The MP associated is only present in new parent area
WITCA-11	6	Multiple MPs can be associated to a zone	1-Logged in as an user (any role) and site associated with MP 2-The user is in Building Administration page 3-Delete zones	- Go to Building Administration - Go to Zones page - Open a created parent area and associated him Management Points - Check that the Management Points are present only in one zone	The MPs associated are present only in one zone

Figura 22 - TestPlan

### 2.9.1.2 Test Case

É um conjunto de passos que descrevem um cenário/caminho possível de teste bem definido cujo principal objetivo é comparar as respostas dos estímulos gerados pelos passos com um resultado esperado.

### 2.9.2 Browser devtools

É uma ferramenta que permite que o desenvolvimento web possa ser feito de uma maneira melhor, mais pratica e com melhores resultados. Pode ser utilizada para verificar a ordem de elementos no código, alterar algumas propriedades CSS, ou para tarefas um pouco mais profundas como verificar cabeçalhos HTTP.

Esta ferramenta foi utilizada para testar nos seguintes browsers para conseguir identificar *bugs*: Chrome, Firefox, IE11, IE9. (Figura 23)

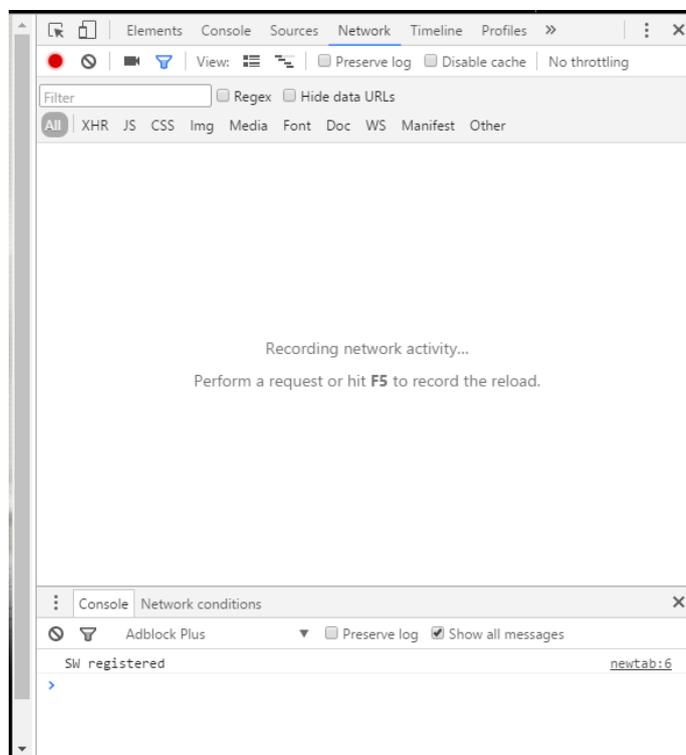


Figura 23 - Ferramenta Devtool

## Conclusão

A realização deste estágio foi, sem dúvida, uma experiência nova e enriquecedora, tanto a nível pessoal como profissional. Consegui colocar em prática alguns dos conhecimentos adquiridos durante a frequência do Curso de Tesp de Testes de Software.

Este estágio proporcionou-me a oportunidade de perceber o mundo profissional, de me tornar uma pessoa mais responsável e cumpridora de horários e de aprender a dar valor a certos aspetos que não julgava importantes.

No final deste projeto, considero que todos os conhecimentos adquiridos em ambiente escolar ganham uma nova dimensão e um novo enquadramento quando utilizado na prática, em situações reais e com objetivos reais.

Assim desta forma o projeto ganha enorme importância para a minha formação técnica, uma vez que permitiu aplicar grande parte dos conteúdos aprendidos.

## Bibliografia

- [1] Altran - Global Delivery Centre, 23 Junho 2016. [Online]. Available: <http://www.altran.pt/>. [Acedido em 23 06 2016].
- [2] “Google,” 24 06 2016. [Online]. Available: <https://www.google.pt/>. [Acedido em 25 06 2016].
- [3] “Jenkins (software)- Wikipedia the free enci,” 24 Junho 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Jenkins\\_\(software\)](https://en.wikipedia.org/wiki/Jenkins_(software)). [Acedido em 24 06 2016].
- [4] “Swagger - The World's Most Popular Framework for APIs,” 24 Junho 2016. [Online]. Available: <http://swagger.io/>. [Acedido em 24 06 2016].
- [5] “ISTQB,” 24 Junho 2016. [Online]. Available: <http://istqbexamcertification.com/>. [Acedido em 24 06 2016].
- [6] “Atlassian,” [Online]. Available: <https://www.atlassian.com/software/jira>. [Acedido em 24 06 2016].
- [7] Stickyminds. [Online]. Available: <https://www.stickyminds.com/book/systematic-software-testing>. [Acedido em 24 06 2016].
- [8] “The Art of Software Testing,” [Online]. Available: <http://www.computing.dcu.ie/~ray/teaching/CA358/TheArtofSoftwareTesting.pdf>. [Acedido em 24 06 2016].
- [9] “Software Testing,” [Online]. Available: <http://fileadmin.cs.lth.se/cs/education/EDA270/Reports/2009/SvenssonGraden.pdf>. [Acedido em 24 06 2016].