



IPG Politécnico
da Guarda
Escola Superior
de Tecnologia e Gestão

RELATÓRIO DE ESTÁGIO

Curso Técnico Superior Profissional
em Testes de Software

Alexandre Morgado Guerra

julho | 2018





Escola Superior de Tecnologia e Gestão.
Instituto Politécnico da Guarda

RELATÓRIO DE ESTÁGIO

**Relatório para obtenção do Diploma de Técnico
Superior Profissional de**

Testes de Software

Alexandre Morgado Guerra

Julho 2018

Elementos Intervenientes

Aluno:

Nome: Alexandre Morgado Guerra

Nº 1012505

Curso: TESP de Testes de *Software*

Estabelecimento de Ensino:

Nome: Instituto Politécnico da Guarda – Escola Superior de Tecnologia e Gestão

Morada: Av. Dr. Francisco Sá Carneiro n.º50 6300-559 Guarda

Contacto: Tel.: +351271220100

Fax.: +351271222690

Entidade Acolhedora:

Empresa: Altran Portugal SA – *Global Delivery Center* – Fundão

Morada: Centro de Negócios e Serviços, Praça Amália Rodrigues 6230-350

Contacto Telefónico: +351 210 331 600

Correio Eletrónico: info@altran.pt

Site: www.altran.pt

Estágio Curricular:

Início: 8 de março de 2018

Fim: 13 de julho de 2018

Supervisor: Engenheiro Marcus Jusé da Silva Pereira

Orientador: Professora Natália Gomes

Agradecimentos

Começo por agradecer ao Instituto Politécnico da Guarda por me ter proporcionado as condições necessárias para frequentar este curso por forma a continuar o meu percurso académico.

Também agradeço à Altran que me acolheu ao longo destes 4 meses permitindo-me a realização do estágio curricular e a integração no projeto.

Quero também agradecer de forma especial à minha família e amigos por todo o apoio que me prestaram durante o período em que frequentei o curso.

Também à equipa do projeto em que fui inserido, por serem compreensivos, simpáticos e dedicados, tendo sido um grande apoio na minha aprendizagem e esclarecimento de dúvidas em relação a todos os desafios e problemas.

Por último, agradeço à minha orientadora de estágio, professora Natália Gomes, docente do IPG pela sua disponibilidade e dedicação e por me ter ajudado nos momentos em que mais precisei.

Obrigado.

Resumo

Este relatório aborda o trabalho realizado no decorrer dos 4 meses de estágio curricular do curso Técnico Superior Profissional de Testes de Software, realizado na Altran - GDC Fundão, no qual foram desenvolvidas várias atividades no âmbito do curso, nomeadamente reportar e registar bugs, realizar testes entre outras tarefas.

Os Testes de Software são essenciais para a construção eficaz de uma aplicação, bem como, a deteção de erros. Estes permitem a sua correção de forma a minorar os problemas, fazendo com que o cliente atinja um maior grau de qualidade com as suas aplicações.

O estágio curricular foi realizado num projeto de âmbito confidencial na área de testes, da indústria farmacêutica. Devido a este contexto, a exigência, ao nível da qualidade do mesmo foi maior sendo necessária a realização de uma formação de Q&EMS e também de higiene e segurança no trabalho, tendo também participado também em aulas de Língua Francesa – iniciação, o que me deu equivalência à graduação de nível A1.

Plano de Trabalho

O plano de trabalho no estágio curricular foi definido pelo Project Manager Engenheiro Marcus Jusé da Silva Pereira realçando as seguintes atividades no decorrer do estágio:

- Estudo dos processos de garantia de qualidade da companhia;
- Execução dos Tests Script do projeto confidencia relativo a equipamentos médicos;
- Execução dos testes exploratórios Black Box;
- Capacitação em processos de qualidade relacionados com a norma ISO-13485;
- Estudo dos processos ISO/IEC-62304 para o ciclo de vida de desenvolvimento de sistemas relacionados a software de dispositivos médicos;
- Suporte na conceção da documentação do projeto como um todo.

Abreviaturas

IPG – Instituto Politécnico da Guarda

ESTG – Escola Superior de Tecnologia e Gestão

TeSP – Técnico Superior Profissional

Índice

Elementos Intervenientes	I
Agradecimentos	II
Resumo	III
Plano de Trabalho	IV
Abreviaturas	IV
Índice de Figuras	VII
Índice de Tabelas	VII
1. Introdução	1
2. Sobre a Altran	2
3. Testes de <i>Software</i>	5
3.1. Os sete princípios dos Testes de <i>Software</i>	7
3.2. Ciclo de vida dos testes de <i>software</i>	8
3.3. Tipos de Teste de <i>Software</i>	9
3.3.1. Testes Funcionais	9
3.3.2. Testes Automatizados	9
3.3.3. Testes unitários	10
3.3.4. Testes de integração	10
3.3.5. Testes de Sistema	10
3.3.6. Testes de regressão	10
3.3.7. Testes de Performance	11
3.3.8. Testes de aceitação	11
3.3.9. Testes Exploratórios	11
3.4. Principais atividades dos testes de <i>software</i>	11
3.5. Ferramentas usadas no projeto	13
3.5.1. <i>JIRA</i>	13
3.5.2. <i>TestFairy</i>	15
4. Objetivos do estágio	15
4.1. O cliente do projeto	16
4.2.1. Fluxo de reportação de <i>bugs</i>	16
4.3. Metodologias de desenvolvimento de <i>Software</i>	17
4.3.1. Fluxo de trabalho	17
4.4. Ferramentas de Testes de <i>Software</i>	19

4.4.1.	Tipos de ferramentas de testes de <i>software</i>	20
4.5.	Formações desenvolvidas	20
4.5.1.	Fundamentos de teste de software – Exercício Pratico	21
4.5.2.	Q&EMS	21
4.5.3.	Francês Aplicado A1.....	22
4.5.4.	Formação de SHT (Higiene e Segurança no Trabalho)	22
	Conclusão.....	23
	Bibliografia	24

Índice de Figuras

Figura 1- Fachada do Pavilhão onde se situa a Altran Global Delivery Center, Fundão	2
Figura 2- Localização da Altran no mundo (imagem não atualizada).....	3
Figura 3 - Mapa de valores da Altran	4
Figura 4- Projeto em parceria com o Grupo Altran	4
Figura 5- Principais atividades dos testes de software.....	13
Figura 6- Logotipo do JIRA.....	14
Figura 7- Exemplo de um backlog de JIRA.....	14
Figura 8- Logotipo do TestFairy.....	15
Figura 9- Fluxo de reportação de bugs	17

Índice de Tabelas

Tabela 1 - Palavras utilizadas no projeto	6
---	---

1. Introdução

Este relatório aborda o trabalho e atividades desenvolvidas ao longo do estágio as quais me permitiram aprofundar os conhecimentos adquiridos na parte curricular do curso.

O estágio curricular decorreu no ano letivo de 2017/2018, no âmbito do curso Técnico Superior Profissional (TeSP) em Testes de *Software*, na empresa Altran – *Global Delivery Centre*, localizada na cidade do Fundão, entidade com a qual o Instituto Politécnico da Guarda (IPG) estabeleceu uma parceria desde da data de início do curso, com uma duração total de quatro meses, tendo início no dia 8 de março e o seu termino no dia 13 de julho.

De todas atividades desenvolvidas ao longo do estágio curricular a mais relevante foi a realização de testes funcionais manuais em ambiente *Web*, assim como em dispositivos *Android* e *iOS*.

A equipa no qual fui inserido e fiz parte ao longo do projeto é constituída pelos seguintes elementos:

- Marcus Pereira – Gerente de projeto;
- Aline Cabral – *Test Manager*;
- Nissi Miranda – Programador de *iOS*;
- Gustavo Gomes – Programador de *Android*;
- André Nunes – Programador de *Frontend*;
- Eduardo Almeida – *Tester*;
- Alexandre Guerra – *Tester*;
- Inês Viera – *Quality Manager*.

Este relatório está dividido em quatro capítulos sendo o primeiro e o segundo constituídos por uma introdução e a caracterização da empresa Altran.

No terceiro capítulo estão descritos os testes de software suas principais atividades e ferramentas utilizadas no projeto. Finalmente no quarto capítulo são descritos os objetivos do estágio as metodologias de desenvolvimento de *software*, as formações desenvolvidas entre outros assuntos, sendo no último capítulo apresentada a conclusão de todo o trabalho desenvolvido.

2. Sobre a Altran¹

A Altran – *Global Delivery Centre* está estabelecida praticamente no centro da cidade do Fundão, tendo sido anunciada a sua construção em 2013 e inaugurada apenas em 2014, criando assim 100 postos de trabalho na cidade, tendo como meta a contratação de 300 colaboradores até ao presente ano de 2018.



Figura 1- Fachada do Pavilhão onde se situa a Altran Global Delivery Center, Fundão

Fonte: <https://www.altran.com/us/en/>

¹ O presente capítulo foi elaborado com base e adaptado em informações da Altran disponível em <https://www.altran.com/us/en/>

O Grupo Altran é uma empresa multinacional francesa que foi fundada no ano de 1982, com especialização em inovação e consultoria no ramo da engenharia de alta tecnologia. Atualmente possui mais de 45.000 colaboradores e tem escritórios estabelecidos em mais de 30 países localizados um pouco por todo o globo. (Fig. 2)



Figura 2- Localização da Altran no mundo (imagem não atualizada)

Fonte: www.altran.com

O grupo está inserido no panorama tecnológico português desde 1998, tendo solidificado a sua presença no país em 2009. Atualmente contém mais de 1350 colaboradores distribuídos pelos três escritórios do país, nomeadamente Lisboa, Porto e Fundão.

A Altran Portugal é uma empresa de Consultoria de Inovação Tecnológica e encontra-se presente em vários setores industriais de relevo mundial, tais como: a Telecomunicação e *Media*, a Administração Pública, o Financeiro, a *Energy & Life Sciences*, a Indústria e Sistemas Inteligentes.

Os vários pólos do grupo, situados um pouco por todo o globo, prosseguem cinco valores considerados imprescindíveis para um futuro próspero sendo estes a **Inovação, Excelência, Dinamismo, Preocupação e Responsabilidade**, como se pode ver na Fig. 3.

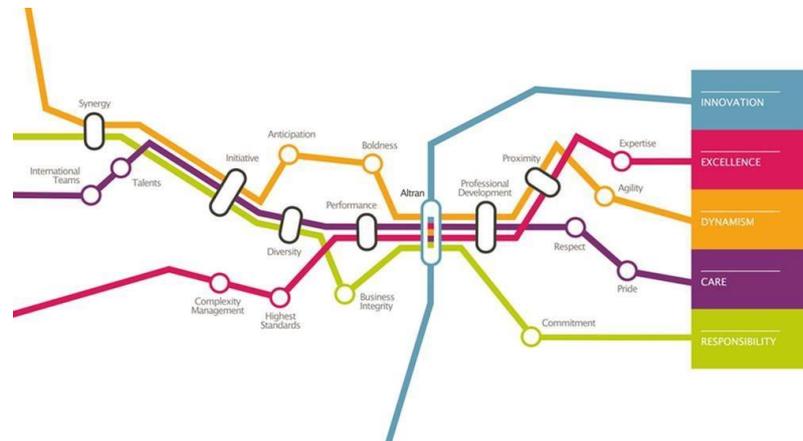


Figura 3 - Mapa de valores da Altran

Fonte: www.altran.com

Assim que os colaboradores integram a Altran, são imediatamente informados desses valores de forma a poderem colaborar para o sucesso da empresa como um todo e consequentemente terem um melhor percurso profissional.

De entre todas as áreas, ou setores, de ação do Grupo Altran destacam-se as de Aeroespacial, Defesa e Ferrovias representando 26% dos clientes; Energia, Indústria e Ciências da vida compondo 23%; Automóvel, Infraestruturas e Transportes com 22%; Telecomunicações e Media com 17% e, por fim, Serviços financeiros e Governo com 12% do número total de clientes. Dentro dos setores referidos anteriormente foram desenvolvidos vários projetos em parceria com grandes empresas tal como é demonstrado na Figura 4.

Design of Navigation System
System design on embedded equipment for Airbus



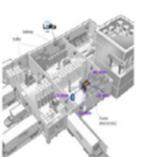

Complete solution including requirements writing, development monitoring, validation & verification process management and permanent support.

Smart, Safe & Secure Platform
Building a standardized Operating System for the IoT World



26 industrial partners have gathered under the lead of the S3P consortium with the objective to set up and develop a unified, standard operating for their IoT solutions. The group, which includes industry leaders like Safran, Alstom, Airbus, Continental, Axa or Schneider, focus in particular on applications for Industry 4.0 and Smarthealth.

Real Time Industrial Asset Tracking
Indoor geo-localization of tools and people in real-time



A complete scalable solution to equip assets and plants with sensors integrating a new type of Lora 2.4GHz communication networks enabling real time indoor geolocation of up to 60 000 tools in real time.

The Connected Operator
A smart coat allowing real time monitoring of work conditions



This coat is connected through wide area networks to a central back-end where information on workers positioning, movements and health parameters can be visualized in real-time and analyzed, in order to prevent health problems, enable fast intervention in case of accidents, improve working conditions, and comply with safety regulations.

DEMO AVAILABLE!

Figura 4- Projeto em parceria com o Grupo Altran

Fonte: <https://www.altran.com/us/en/>

3. Testes de *Software*²

Os testes de *software* devem ser considerados essenciais para a criação e desenvolvimento de produtos com base no *software*, de forma a garantir a sua qualidade consoante os requisitos estabelecidos durante o seu levantamento com o cliente.

Visto que os produtos de *software* são desenvolvidos por empresas, caso exista uma falha no produto pode significar um grande dispêndio financeiro, bem como uma diminuição no prestígio para o cliente final, sem contar com os inúmeros perigos que podem causar a indivíduos, caso este esteja relacionado com a área da saúde. Para prevenir acidentes tais como os referidos anteriormente, é necessário investir em recursos de testes, para assim conseguir detetar defeitos e por sua vez realizar as devidas correções no produto antes da sua finalização, evitando quaisquer encargos financeiros adicionais.

Para uma melhor compreensão do tema elaboro uma tabela com algumas definições utilizadas na área.

Palavras	Significado
<i>Tester</i>	Pessoa encarregue de realizar os testes
<i>Step</i>	Passo a seguir ao longo da execução dos testes
<i>User Storie</i>	Descrição simples da funcionalidade a ser desenvolvidas
Requisitos Funcionais	Define uma função de um sistema de software ou seu componente
<i>Sprint</i>	Reunião entre a equipa de desenvolvimento e o cliente onde é apresentado as funcionalidades implementadas e testadas durante as duas semanas de sprint
<i>Bugs</i>	Quando é encontrado um erro em algum programa, obtendo se um resultado inesperado ou fora do comum

² Tópico adaptado do site - <https://www.istqb.org>

<i>Portal Web</i>	Site na internet no qual aglomera as várias informações complementares ao software do projeto
<i>Back Log</i>	Conjunto de tarefas que faltam ser desenvolvidas e estão inseridas no plano de projeto
<i>Sprint Review Meeting</i>	Pequena reunião diária no qual se discute as tarefas feitas no dia anterior e as que irão ser realizadas
<i>Sprint Retrospective/Lessons Learned</i>	É uma reunião onde é partilhado os aspetos positivos e os negativos que ocorreram no período de desenvolvimento
<i>Test Manager</i>	Gestor da equipa de testes
<i>SOP</i>	Documentos a ler antes de se iniciar o projeto para que esteja a par da situação do mesmo
<i>Black Box</i>	Testes Manuais nos quais não se tem acesso ao código
<i>Design</i>	É o desenho, conceção e idealização do software
<i>Test Sprint</i>	Execução dos testes divididas por Sprint
<i>Developers</i>	Programadores
<i>Commit ID</i>	Conjunto de mudanças experimentais que é adicionado a estrutura geral do código
<i>Relese</i>	Atualização da versão final do software/aplicação

Tabela 1 - Palavras utilizadas no projeto

3.1. Os sete princípios dos Testes de *Software*³

Para uma compreensão mais clara relativamente aos testes de *software* existem sete princípios a ter em conta durante a sua execução, que relatam a limitação, bem como, delinear o alcance dos mesmos:

1- Testes apontam a existência de falhas

Os testes constataam a presença de defeitos, porém, por mais defeitos que o *testers* encontrem não significa que o sistema se torne totalmente correto;

2- O teste exaustivo é impossível

Para combater tal facto um *tester* deve testar de acordo com os objetivos estabelecidos como por exemplo riscos, prioridades ou então elaborar uma estratégia de maneira a encontrar o maior número de defeitos possíveis ou então os mais críticos;

3- Teste antecipado

A execução dos testes deve iniciar o mais cedo possível dentro do ciclo de desenvolvimento do *software*, diminuindo assim o custo das correções dos possíveis *bugs* encontrados tanto no *design* ou arquitetura.

4- Agrupamento de falhas

O esforço de teste deve incidir de forma proporcional à densidade de defeitos esperada e posteriormente observada para cada módulo. Isto significa que um número pequeno de módulos normalmente contém a maioria dos defeitos descobertos durante os testes antes da sua entrega ou exibe a maioria das falhas operacionais;

5- Paradoxo do pesticida

No conjunto de Casos de Teste chegará a um momento que não irão ser encontrados mais defeitos pois são sempre executados os mesmos testes e são corrigidos todos os defeitos que estes encontram, chegando a uma altura que não vão ser encontrados mais falhas no sistema.

³ Tópico retirado e adaptado com base no site <http://app.crowdtest.me/7-principios-fundamentais-teste-software/>

É importante que os Casos de Teste sejam revistos e rescritos, caso necessário, para exercitar diferentes pontos do sistema e encontrar novos defeitos.

6- Os testes são dependentes do seu contexto

Os testes são realizados de forma diferente conforme o contexto, ou seja, dependendo de cada *software* tem que se utilizar diferentes estratégias de teste. Um exemplo pertinente é no que toca a *softwares* de segurança usados nos bancos, que devem ser testados de maneira diferente dos *softwares* de saúde;

7- Ausência de erros é uma ilusão

Encontrar e compor defeitos não ajuda se o sistema construído não atende às expectativas ou necessidades dos utilizadores. Existem outros fatores que devem ser considerados para garantir a qualidade do sistema.

3.2. Ciclo de vida dos testes de *software*

A estrutura dos ciclos de vida dos testes de *software* são divididos em 5 fases das quais são:

1. **Planeamento** consiste na elaboração da estratégia de Testes, Plano de Teste e a análise do projeto;
2. **Preparação** dos ambientes de testes, os equipamentos, o hardware e software, ferramentas para automação de testes e assim por diante;
3. **Especificação** dos casos de teste que podem ser alterados após a revisão dos mesmos e assim corrigir de maneira a ter uma maior eficácia na realização dos seus testes;
4. **Execução** é a fase na qual os testes são executados de acordo com os seus casos de teste, e registar os resultados obtidos;
5. **Entrega** é a última etapa no qual o projeto de testes deve ser arquivado juntamente com a documentação e relatar todas as ocorrências no projeto com propósito de melhorar o processo;

3.3. Tipos de Teste de *Software*⁴

3.3.1. Testes Funcionais

Estes tipos de testes são estabelecidos através de Casos de Teste no Plano de Teste, e os mesmos são elaborados de forma manual, de modo a possibilitar a documentação dos resultados obtidos, durante a execução dos mesmos. O principal motivo para que este teste não seja executado de forma automática provém da existência de um maior grau de exigência o que impossibilita o uso de testes automatizados, como foi o caso do projeto realizado no presente estágio curricular.

Os testes manuais são bons para encontrar certos tipos de inconformidades no sistema, porém é um processo onde o nível de cansaço do *tester* afeta a qualidade do teste, para além disso também requer algum tempo para realização de cada etapa de testes, dependendo ainda de *tester* para *tester*.

3.3.2. Testes Automatizados

Testes automatizados são uma forma de realizar testes utilizando *software* e de uma forma automatizada, porém é necessário utilizar processos manuais para que o mesmo seja efetuado corretamente.

Os testes automatizados consistem na escrita de um programa para realizar testes, seguido da execução dos mesmos, de forma automática, permitindo que a etapa de execução e validação do sistema seja bem mais rápida, isto representa um maior grau de eficiência e é ainda extremamente benéfico para projetos que dispõem de um tempo prolongado no seu desenvolvimento.

Neste tipo de testes é possível a comparação de resultados esperados com resultados reais, a definição das pré-condições e relatórios de teste são outras condições das funções de controlo que têm que ser definidos antes e após a execução dos mesmos, respetivamente.

⁴ Tópicos retirados com base no site <https://www.istqb.org/>

3.3.3. Testes unitários

Estes tipos de testes são implementados pelos programadores para validar pedaços de código relativos à lógica de negócio, perante as diferentes condições. Normalmente estes são compostos por conjuntos de asserções que validam os resultados obtidos com o que era espetável. Cada teste unitário funciona de maneira independente dos demais, o que pode proporcionar ao programador uma forma de testar cada módulo individualmente.

3.3.4. Testes de integração

São um tipo de testes em que os vários módulos do sistema são combinados e são testados em conjunto. O principal objetivo deste tipo de testes é garantir que todos os componentes conseguem interagir entre si, validando que o sistema irá funcionar como um todo e de acordo com as especificações definidas no início do projeto.

As finalidades dos testes de integração é validar os requisitos funcionais do sistema, bem como a confiabilidade e desempenho do sistema.

3.3.5. Testes de Sistema

Os testes de sistema são um tipo de testes de *software* um pouco mais limitado que os testes de integração ou os de unidade.

Estes tipos de teste são um processo dos testes de *software* que consistem em verificar os seus requisitos num ambiente de produção e são mais focados em aspetos gerais e não é necessário um conhecimento estrutural do sistema.

Testes de sistema não se limitam apenas a testar requisitos funcionais envolvendo também requisitos não funcionais, com a perspetiva de validar e garantir a qualidade do trabalho para com o cliente.

3.3.6. Testes de regressão

Este tipo de testes consiste em correr os testes já elaborados para versões mais antigas de modo a validar que a nova versão do sistema não contém qualquer defeito ou comportamento diferente dos que foram validados anteriormente.

Os testes de regressão são executados no momento em que existam alterações no sistema que estão integrados em componentes já existentes anteriormente, a partir de implementações de novas funcionalidades até correções de defeitos.

3.3.7. Testes de Performance

Como o nome indica estes são um tipo de testes que tem como finalidade testar o desempenho do produto de *software*.

Estes tipos de testes são realizados antes da publicação de cada nova versão e sempre com as devidas ferramentas, permitindo, por exemplo, testar condições de carga excessiva no sistema, simular o tráfego ou ainda o tempo de resposta do mesmo.

3.3.8. Testes de aceitação

Este tipo de testes tem como base a análise exclusiva de uma funcionalidade, componente ou sistema. A função destes testes é de averiguar o sistema em relação aos seus requisitos originais em função das necessidades atuais do utilizador.

3.3.9. Testes Exploratórios

Os testes exploratórios são realizados de uma maneira informal, no qual não dispõe de documentação. Este tipo de teste tem funcionalidades que não estão especificadas no período de execução dos *Test Script* ou presentes na *User Storie*.

Este tipo de teste só é totalmente fiável se o *tester* em questão tiver total conhecimento do *software*, pois irá depender da sua capacidade de definir o esforço indispensável para conduzir este tipo de teste.

3.4. Principais atividades dos testes de *software*

Nos testes de *software* existem outras tarefas para além da própria execução de testes. Essas atividades são também consideradas vitais para a realização dos mesmos, de modo a que todo o processo de testes seja executado devidamente, este tem que envolver as seguintes atividades:

- **Análise de Requisitos**

Esta é uma das primeiras atividades do desenvolvimento de *software*, uma das mais importantes onde é feita a especificação de requisitos que por sua vez é dividida em requisitos funcionais e requisitos não funcionais.

Existem diversas técnicas para o levantamento de requisitos para fazer uma análise mais completa, como por exemplo: entrevistas, análise de documentação, pesquisa externa, etc;

- **Planeamento e Preparação de ambiente de testes**

Na etapa de planeamento é definida uma solicitação de testes e são definidos objetivos com base nas expectativas do cliente, relativamente aos custos, prazo e qualidade prevista, para que seja construída a equipa e estabelecer o esforço de acordo com as necessidades do cliente. No decorrer do planeamento é importante comparar o plano de desenvolvimento com o plano de testes.

- **Análise e Desenho dos casos de teste**

Para a elaboração dos casos de teste é preciso utilizar uma estrutura específica para que a informação fique perceptível e que todos os requisitos estejam validados no plano de teste.

Em geral os seguintes campos devem estar presentes:

- Identificação do Plano de Teste;
- Identificação das funcionalidades a serem testadas;
- Identificação do caso de teste;
 - Título;
 - Descrição
 - *Steps*;
 - Resultado esperado;
 - Pré-condições.

- **Execução dos casos de teste**

À medida que os programadores finalizem o desenvolvimento de cada funcionalidade do software, o *tester* pode iniciar a execução do Plano de Teste. Nesta fase tem que fazer a verificação dos resultados obtidos com os esperados e a avaliação dos critérios de saída. Caso exista algum *bug* este tem que se reportado de maneira muito clara.

- **Reportação de *bugs***

Quando é encontrado um *bug* é necessário partilhar esta informação com a restante equipa e fazer o registo tendo em conta que deve sempre conter os seguintes campos:

- Título;
- Descrição;
- *Steps*;
- Resultado esperado;
- Resultado obtido;

- Versão da aplicação / *software*;
 - Prioridade;
 - Evidencias.
- **Criação dos relatórios com informação geral dos testes;**
Numa fase final da execução dos testes fazer um relatório geral é uma tarefa importante para ficar registado o número de *bugs* encontrados, assim como a quantidade de casos de teste executados e comentários relevantes de possíveis riscos.

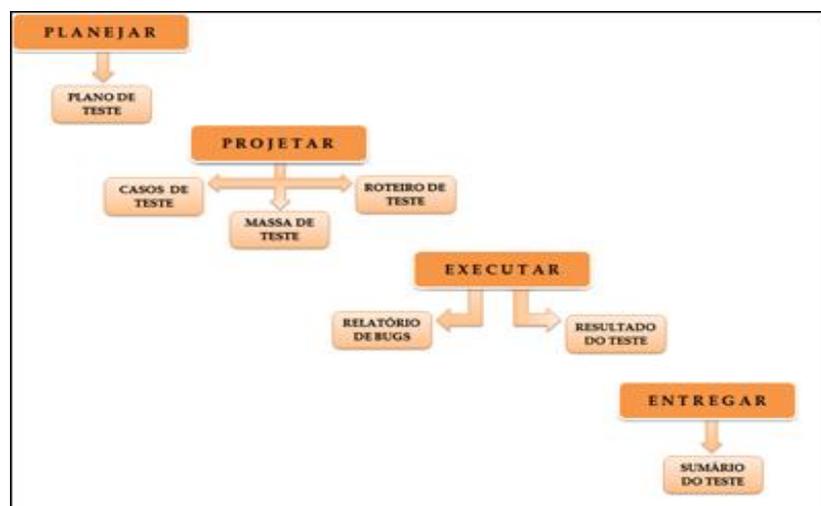


Figura 5- Principais atividades dos testes de software

Fonte: <http://dvillela.dlinkdns.com:4000/?p=448>

3.5. Ferramentas usadas no projeto

3.5.1. JIRA⁵

O *JIRA* é um *software* desenvolvido pela *Atlassian*, com a função de monitorizar tarefas e acompanhar projetos desde a sua criação garantindo assim a sua gestão de todas as suas atividades num só local.

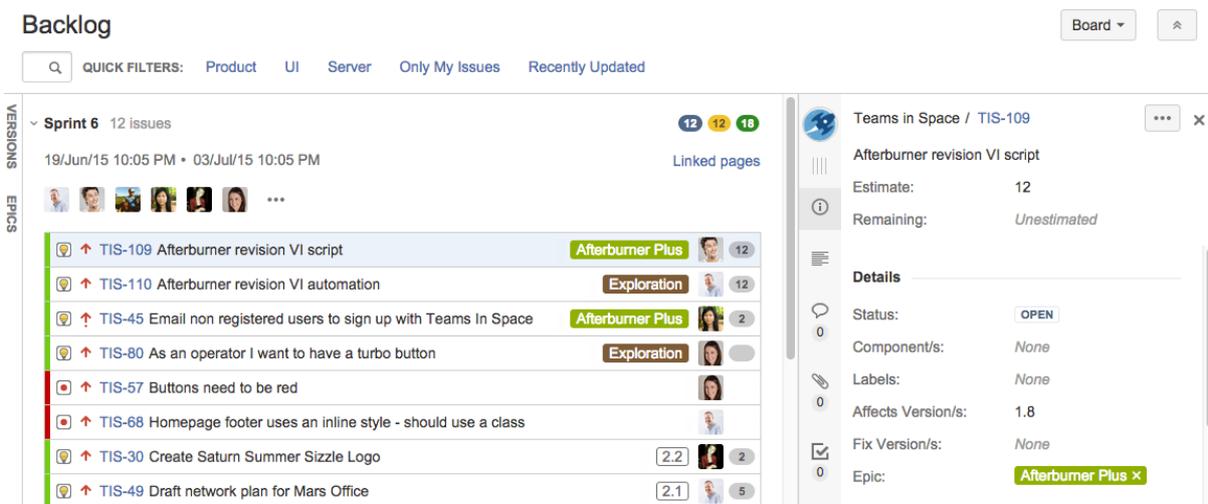
Esta ferramenta foi uma das mais utilizadas no projeto com o objetivo de fazer a consulta das *User Stories*, visualização e distribuição das tarefas atribuídas a cada elemento, registo de *bugs*, assim como outro tipo de tarefas que são as mais utilizadas pelo gestor de projeto.

⁵ Tópico baseado e adaptado com base no site - <https://www.atlassian.com/software/jira>



Figura 6- Logotipo do JIRA

Fonte: <https://www.atlassian.com/software/jira>



Backlog Board

QUICK FILTERS: Product UI Server Only My Issues Recently Updated

Sprint 6 12 issues
19/Jun/15 10:05 PM • 03/Jul/15 10:05 PM
Linked pages

Issue ID	Issue Description	Label	Assignee	Estimate
TIS-109	Afterburner revision VI script	Afterburner Plus	[Avatar]	12
TIS-110	Afterburner revision VI automation	Exploration	[Avatar]	12
TIS-45	Email non registered users to sign up with Teams In Space	Afterburner Plus	[Avatar]	2
TIS-80	As an operator I want to have a turbo button	Exploration	[Avatar]	2
TIS-57	Buttons need to be red		[Avatar]	2
TIS-68	Homepage footer uses an inline style - should use a class		[Avatar]	2
TIS-30	Create Saturn Summer Sizzle Logo		[Avatar]	2.2
TIS-49	Draft network plan for Mars Office		[Avatar]	2.1

Teams in Space / TIS-109 ... x

Afterburner revision VI script

Estimate: 12
Remaining: Unestimated

Details

Status: OPEN
Component/s: None
Labels: None
Affects Version/s: 1.8
Fix Version/s: None
Epic: Afterburner Plus x

Figura 7- Exemplo de um backlog de JIRA

<https://confluence.atlassian.com/jirasoftwareserver073/using-your-scrum-backlog-861254490.html>

3.5.2. TestFairy⁶

Esta ferramenta foi usada no projeto com uma finalidade de armazenamento das aplicações móveis na qual era possível aceder aos artefactos referentes.

Sempre que corrigiam os defeitos nas fases anteriores, em seguida era gerada uma nova *release* com uma pequena descrição do defeito assim como um novo número de versão que corresponde a um novo artefacto gerado.



Figura 8- Logotipo do TestFairy

Fonte: <https://www.testfairy.com/>

4. Objetivos do estágio

O objetivo principal do estágio foi o de realizar testes de *software* em ambiente real. A realização de testes é fundamental para qualquer projeto de desenvolvimento de *software*, pois os erros nos produtos de *software* podem causar inúmeros problemas.

As principais tarefas realizadas no decorrer do estágio curricular, e anteriormente descritas, foram:

- Conciliar a aprendizagem ao longo do curso com os exercícios realizados durante o estágio;
- Realizar testes funcionais em diferentes ambientes: *Android*, *iOS* e portal *Web*;
- Reportar erros encontrados durante os testes;
- Criação de planos de testes funcionais;
- Entender quais as ferramentas essenciais para a realização das tarefas;
- Aprender a usar novas ferramentas;
- Integração na equipa do projeto.

⁶ Tópico baseado e adaptado com base no site - <https://www.testfairy.com/>

4.1. O cliente do projeto

O projeto, desenvolvido ao longo destes 4 meses de estágio curricular, é de foro confidencial, porém é possível indicar que o cliente pertence/detém uma empresa de renome internacional na área da indústria farmacêutica.

4.2. O projeto

O projeto desenvolvido foi baseado num outro projeto já existente na Altran e teve como principal objetivo a adaptação/validação/teste do sistema num novo dispositivo médico.

Aquando da criação do projeto foi desenvolvida uma solução informática, sistema, composta por duas aplicações móveis (*iOS* e *Android*). O projeto conta ainda com um portal *web* para administração do sistema.

As aplicações desenvolvidas no projeto têm como finalidade interagir com um dispositivo médico, de modo a facultar algumas informações médicas e de saúde, tanto ao seu utilizador final, doente, como para o médico desse utilizador.

O objetivo principal é o de poder assistir o utilizador e ainda auxiliar na monitorização dos sintomas do paciente.

4.2.1. Fluxo de reportação de *bugs*

No decorrer da devida execução dos testes de *software*, existiram muitos defeitos e melhorias a encontrar. É de alta importância que estes após serem encontrados sejam oportunamente reportados para as pessoas responsáveis.

Durante o decorrer do projeto, foi utilizado um fluxo de trabalho que consistia em: testar o produto de *software*, assim que algum defeito fosse encontrado, o mesmo seria reportado no *JIRA* para que o programador o pudesse corrigir devidamente, comentar o *bug* e para depois gerar o *commit id* da respetiva correção, de modo a obter um maior grau de rastreabilidade.

De seguida o *tester* verifica as correções, implementadas pelo programador, e caso o defeito persistisse era novamente marcado como *bug* no *JIRA* e atribuído ao programador que o tentou corrigir, todo este processo se repete até que o *bug* já esteja devidamente corrigido. Assim que esteja devidamente corrigido é inserido as evidências no *JIRA* para registro e finalizado o *Bug*.

Na Figura está representado o fluxo de reportação de bugs realizado no decorrer do desenvolvimento do projeto no âmbito do estágio curricular.

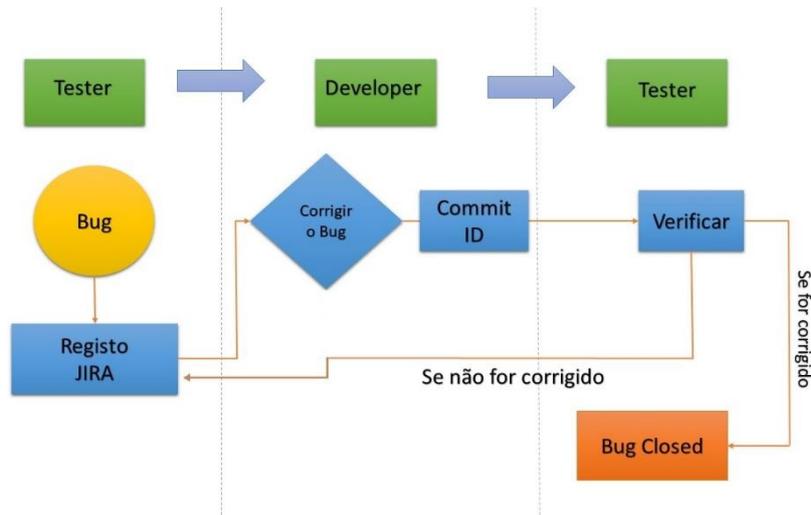


Figura 9- Fluxo de reportação de bugs

Fonte: Adaptada

4.3. Metodologias de desenvolvimento de *Software*

Existem vários tipos de metodologias de desenvolvimento de *software* que podem ser usadas em vários projetos. Das quais existem a metodologia Ágil, em V e em Cascata.

As metodologias têm como objetivo ajudar no desenvolvimento de um projeto melhorando-o na sua organização diária, por forma a alcançar as metas desejadas com uma boa eficiência e rapidez bem como a comunicação com a equipa.

As metodologias de desenvolvimento de *software* sofrem continuamente um desenvolvimento ao longo dos tempos, que por sua vez se ajustam às diversas necessidades dos clientes e das equipas que desenvolvem projetos de *software*.

4.3.1. Fluxo de trabalho

Este projeto adotou uma metodologia de desenvolvimento Ágil, denominada *Scrum*, sendo que teve que ser adaptada devido ao facto de este ter que seguir normas de qualidade internacionais, obrigando a que o mesmo utilize a metodologia em cascata.

O *Scrum* é uma metodologia Ágil que é usada na gestão e planeamento de projetos de *software*, que consiste na divisão do projeto em ciclos denominados de *Sprints*, que por sua vez representam uma *Time Box* em que um conjunto de atividades deverá ser executada.

As funcionalidades que são implementadas num projeto são colocadas numa lista que é conhecida como *Product Backlog*, ou simplesmente *Backlog*.

No início de cada *Sprint*, é realizada uma *Sprint Planning Meeting* que se baseia numa reunião de planeamento na qual a equipa seleciona devidamente as atividades, presentes no *backlog*, que vão ser implementadas durante a *Sprint*.

Todos os dias ao longo do projeto, a equipa realiza também uma pequena reunião chamada de *Daily Meeting*, em que o objetivo é divulgar o conhecimento acerca do que foi realizado no dia anterior, identificar impasses encontrados e delinear o trabalho que se irá realizar ao longo do presente dia.

No final de cada *Sprint* a equipa expõe as funcionalidades implementadas durante *Sprint Review Meeting*, em que no final realiza-se uma *Lessons Learned* e a equipa encaminha-se para o planeamento da *Sprint* seguinte.

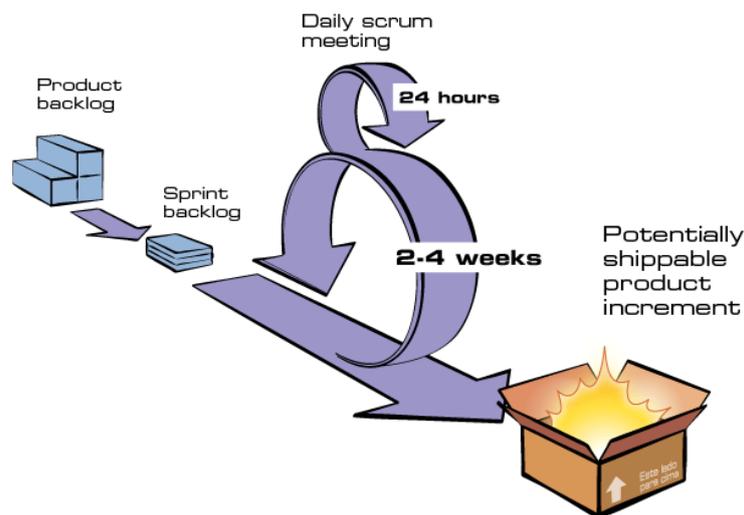


Figura 10 - Representação da Metodologia Ágil em SCRUM

Fonte: <https://www.desenvolvimentoagil.com.br/scrum/>

Sprint 1 – Esta *Sprint* foi principalmente dedicada ao desenvolvimento do produto e à sua devida aprovação dos documentos por parte do cliente tais como Plano de Teste, Estratégia de teste e *Test Script*.

Sprint 2 – Foi de maior focagem nos *system test* e às suas devidas correções de *bugs*.

Sprint 3- Dedicada principalmente para os testes informais destacando os de aceitação e de alinhamento por parte do cliente do projeto.

Sprint 4 – Esta foi a *sprint* final em que foram realizados vários testes de aceitação.

Na 1ª *Sprint* não ocorreu grande trabalho por parte dos testes, pois o *software* ainda estava a ser desenvolvido e implementado por parte dos programadores que implementavam as funções necessárias, por sua vez, o/a *Test Manager* que estava a adaptar os *tests Scripts* às devidas *User Stories* aprovadas por parte do cliente em questão.

4.4. Ferramentas de Testes de Software

As ferramentas de testes de *software* podem auxiliar em todos os processos de administração de testes de *software*. Estes tipos de processos vão desde o registo de requisitos do projeto, até às suas criações e execuções dos Planos de Teste e Casos de Teste.

As ferramentas de testes de *software* têm alguns benefícios tais como:

- Facilidade no agenciamento na execução de Casos de Teste;
- Facilidade no acesso às informações acerca dos testes;
- Ajuda na automatização de planos de testes;
- Redução de trabalho repetido.

Porém os testes de *software* também têm alguns defeitos tais como:

- Elevado gasto de tempo e a dificuldade necessária para o alcance de vantagens significativas e contínuas na ferramenta;
- Perspetivas irreais acerca da ferramenta.

4.4.1. Tipos de ferramentas de testes de *software*

Existem diversos tipos de ferramentas de testes de *software* e cada tipo de ferramenta tem as suas funções, sendo possível agrupar as ferramentas para ser mais fácil a compreensão das funções de cada uma.

Dentro dos tipos de ferramentas existem:

Ferramentas de gestão de testes - Este tipo de ferramenta de teste tem como função o devido registo da execução de testes e de *bugs*, uma gestão de requisitos e um relatório de execução de testes;

Ferramentas de gestão de requisitos - São ferramentas que o seu funcionamento gira à base de armazenar instruções de requisitos e ajudar a identificar os requisitos inconsistentes ou em falta num sistema;

Ferramentas de gestão de incidentes – Estas armazenam e gerem os relatórios de incidentes e por sua vez ajudam a gestão do ciclo de vida dos defeitos, é o exemplo da plataforma *JIRA* que foi utilizado no desenvolvimento do projeto do estágio curricular;

Ferramentas de execução de testes – Permitem que os testes sejam executados automaticamente.

4.5. Formações desenvolvidas

No domínio do projeto desenvolvido no estágio curricular na empresa Altran Portugal, tive a oportunidade de desenvolver e aprofundar os meus conhecimentos e metodologias assimiladas no decorrer do TeSP de Testes de *Software*. Tudo isto só foi possível através da integração que este projeto profissional me proporcionou.

Tive também a possibilidade de adquirir conhecimentos nesta área dos testes como também foi possível a participação em várias formações relacionados tanto a nível técnico como linguístico.

4.5.1. Fundamentos de teste de software – Exercício Prático

No início do estágio curricular tive uma pequena formação com a consultora Andreia Esteves, para consolidar todos os conceitos que aprendi no curso e por em prática os meus conhecimentos antes de começar a trabalhar no projeto onde fui alocado.

O exercício consistiu em analisar os requisitos que nos foram disponibilizados, elaborar um Plano de testes e posteriormente a execução dos mesmos apenas para uma das funcionalidades do site “Edreams” que neste caso foi para a área de Pesquisa de voos. Foi uma mais valia porque contribuiu para a minha preparação inicial no estágio e para o esclarecimento de algumas questões.

CP-CH 01 Test Plan							
Test case ID	Test case Description	Precondition	Test Case steps ID	Test step - Description	Expected result	Actual result	Pass/Failed
TC-1	Pesquisa com dados válidos de Voos	Ter acesso ao site eDreams https://www.edreams.pt/	1.1	Clickar na barra "De"	A pesquisa deverá ser efectuada e demonstrar a informação	A pesquisa foi efectuada devidamente e demonstrou os resultados esperados	
			1.2	Inserir cidade valida			
			1.3	Clickar no campo "Para"			
			1.4	Inserir cidade valida			
			1.5	Clickar no campo "Partida"			
			1.6	Selecionar data valida			
			1.7	Clickar no campo "Regresso"			
			1.8	Selecionar data valida			
			1.9	Clickar no campo "pessoas"			
			1.10	Selecionar numero de pessoas valido			
			1.11	Clickar no checkbox "de voos directos" (optional)			
			1.12	Clickar no campo "class do voo"			
			1.13	Selecionar o tipo da class do voo			
			1.14	Clickar no campo "Pesquisar voos"			
TC-2	Pesquisa com dados inválidos de Voos	Ter acesso ao site eDreams https://www.edreams.pt/ Ter os campos: Partida, Regresso, Numero de Passageiros, Class de voos devidamente preenchidos	2.1	Clickar no campo "De"	A pesquisa não deverá ser efectuada devidamente pois os campos "De" e "Para" estão independentemente preenchidos e após a pesquisa aparece uma mensagem a dizer "Introduza a cidade correta"	A pesquisa não foi efectuada devidamente pois os campos "De" e "Para" estão independentemente preenchidos e após a pesquisa aparece uma mensagem a dizer "Introduza a cidade correta"	
			2.2	Inserir dado Invalido (ex: "Guarda")			
			2.3	Clickar no campo "Para"			
			2.4	Inserir dado Invalido (ex: "Castelo Branco")			
			2.5	Clickar no campo "Pesquisar voos"			

Figura 11 - Exercício prático de fundamentos de testes de software

Fonte: Adaptado

4.5.2. Q&EMS

Por o projeto estar ligado á indústria farmacêutica existe uma serie de exigências e procedimentos na qual é obrigatório a realização da mesma.

O objetivo da formação foi compreender todos os procedimentos relativos ao desenvolvimento de software para *Medical Device*.

Na qual também proporcionou um melhor conhecimento sobre o projeto e para uma melhor compreensão do conhecimento curricular.

4.5.3. Francês Aplicado A1

Devido ao facto de a Altran ser uma empresa de origem francesa traz-nos de uma forma positiva a aquisição de uma versatilidade nas línguas. Tive a possibilidade de integrar uma formação linguística – Francês Aplicado A1 foi possível ter uma interação maior não só com a língua francesa, mas também com a sua cultura.

4.5.4. Formação de SHT (Higiene e Segurança no Trabalho)

Esta é uma das formações que é facultada pela Altran a qual foi elaborada de uma forma online, através da plataforma interna da empresa, em que forneceram os conteúdos e realizada a devida avaliação.

A formação de *SHT* teve como objetivo a diminuição dos acidentes de trabalho e doenças relacionadas com a questão profissional, tal como a forma de atuar em caso de imergência, acidentes ou catástrofes naturais. Desta forma os colaboradores irão ter uma melhor qualidade de vida no ambiente de trabalho.

Conclusão

A execução do estágio curricular na Altran foi por vários fatores uma experiência bastante enriquecedora, que me permitiu aplicar os conhecimentos adquiridos ao longo do percurso académico no TeSP - Testes de *Software* tal como a superação de dificuldades e também a obtenção de novas aptidões pessoais e profissionais.

O projeto no qual integrei garantiu-me sempre as condições necessárias à consolidação de conhecimentos na área de testes de *software* o que me permitiu um grande progresso nos meus conhecimentos.

As formações realizadas foram de todo uma mais-valia para a minha aprendizagem e evolução cognitiva, especialmente a execução do exercício prático, que teve como finalidade a obtenção de um conhecimento mais abrangente sobre os testes de *software*.

Em suma a realização do estágio no grupo Altran foi uma experiência muito positiva em que irei extrair o melhor proveito do mesmo.

Bibliografia

- “Sobre a Altran” [Online] <https://www.altran.com/us/en/about-us/>
[Acedido em 5 de Junho de 2018]
- [Online] <https://www.istqb.org>
[Acedido em 7 de Junho de 2018]
- “Princípios dos Testes de Software” [Online] <http://app.crowdtest.me/7-principios-fundamentais-teste-software/>
[Acedido em 15 de Junho de 2018]
- “Processo de Teste de Software” [Online] <http://dvillela.dlinkddns.com:4000/?p=448>
[Acedido em 19 de Junho de 2018]
- [Online] <https://www.atlassian.com/software/jira>
[Acedido em 21 de Junho de 2018]
- “Accessing the Backlog” [Online]
<https://confluence.atlassian.com/jirasoftwareserver073/using-your-scrum-backlog-861254490.html>
[Acedido em 25 de Junho de 2018]
- [Online] <https://www.testfairy.com/>
[Acedido em 29 de Junho de 2018]
- “Scrum” [Online] <https://www.desenvolvimentoagil.com.br/scrum/>
[Acedido em 30 de Junho de 2018]