

**Software Project Management 2019/2020**

**Bill Splitter**

**D3.2.1 - Post-Mortem Analysis**

**Autores**

* Ruben Marques ([rubenantoniomarques@gmail.com](mailto:ruifilipetmota@gmail.com))
* Paulo Dias ([dias.pauloalex@gmail.com](mailto:dias.pauloalex@gmail.com))
* Sérgio Soares ([smcsoares20@gmail.com](mailto:smcsoares20@gmail.com))
* Rui Mota ([ruifilipetmota@gmail.com](mailto:ruifilipetmota@gmail.com))

**Estado**

* Rascunho

**Versões principais**

**Versões publicados**

* v0.1, 21 de Dezembro de 2019, Rui Mota, Principal desenvolvimento
* v0.2, 8 de Janeiro de 2020, Rui Mota, Análises individuais, de grupo e conclusão

**Table of Contents**

[**1. Introdução**](#_szusinuarqcp) **4**

[1.1. Objetivos do projeto](#_yryby27opp8q) 4

[1.2. Visão](#_l527t317o9sn) 4

[1.3. Equipa](#_lspwf34layro) 4

[**2. Gestão do Projeto**](#_6dnb1umxde42) **5**

[2.1. Ciclo de Vida do Projeto](#_c8q9bd8mzoil) 5

[2.2. Entregas](#_mcjh2d8ipi49) 6

[2.3. Escalonamento](#_b73wbjj2pcdz) 7

[2.4. Milestones](#_wej5yfcid9ov) 7

[2.5. Custo / Esforço](#_toekqfvf4jue) 8

[2.6 Repositórios](#_2v9kdeckjzz5) 8

[2.7 Controlo do projeto](#_33woqtf2k26g) 8

[2.9 Planeado vs Realidade](#_9nudflyhio0g) 9

[**3. Produto**](#_c1x7vcfzcrbp) **10**

[3.1. Requisitos Iniciais](#_d1uys5cgnnzi) 10

[3.2. Requisitos Finais](#_dtgnxs5brjxv) 10

[**4. Documentos**](#_dnkkcys1osvx) **11**

[4.1 Repositório de documentos](#_jnxc2ocgx0sk) 11

[4.2 Ciclo de Vida dos Documentos](#_1hqujmdsaj74) 12

[**5. Qualidade do Projeto**](#_h0nbuudksclg) **13**

[5.1. Testes](#_bgudxpj31mgw) 13

[5.2. Inspeções](#_m3g18qdon6nd) 13

[5.3. Change Requests](#_c5lyzaqnczzb) 14

[5.4. Standards](#_dma8i9df61qx) 15

[5.5. Perceção da Qualidade](#_uc16gxs001v) 16

[**6. Processos Técnicos**](#_66hycoa43p9l) **17**

[6.1. Ferramentas](#_nix66jlcef01) 17

[6.2. Arquitetura](#_h2hrtltlvm1t) 17

[**7. Riscos**](#_w0cmw062q4ue) **19**

[7.1. Planos de Mitigação](#_auktesmf1b9e) 20

[**8. Análise**](#_g3mysz71x5kj) **21**

[8.1. Análise individual](#_oa85d4s0q202) 21

[8.1.1. Ruben Marques](#_uoc8dbf9x754) 21

[8.1.2. Rui Mota](#_pyctcrqbvoeo) 21

[8.1.3. Paulo Dias](#_3rghr5sdey5l) 22

[8.1.4. Sérgio Soares](#_p35rm33g1lgn) 22

[8.2. Análise do grupo](#_pl0kojsspu9t) 23

[8.3. Conclusão](#_lu7zwnac1mnn) 23

### 

# 1. Introdução

Este documento visa sintetizar o decorrer do projeto *Bill Splitter*. Tem como principais objetivos a descrição do produto, o seu plano de desenvolvimento, a análise dos seus riscos, a qualidade do produto e uma análise final do projeto.

## 1.1. Objetivos do projeto

Este projeto é feito no contexto da unidade curricular de Gestão de Projeto de Software e tem como objetivo dar a conhecer todo o planeamento que está envolvido no desenvolvimento de um projeto de software seguindo a metodologia de Waterfall.

## 1.2. Visão

O projeto tem como objetivo criar uma aplicação que auxilie o utilizador a dividir uma despesa com os seus amigos e/ou colegas. Dependendo de vários fatores como falta de tempo, cansaço, entre outros, um simples cálculo pode se tornar num verdadeiro quebra-cabeças.

A aplicação permitirá então identificar os intervenientes, as suas despesas e quem já pagou a sua parte.

## 1.3. Equipa

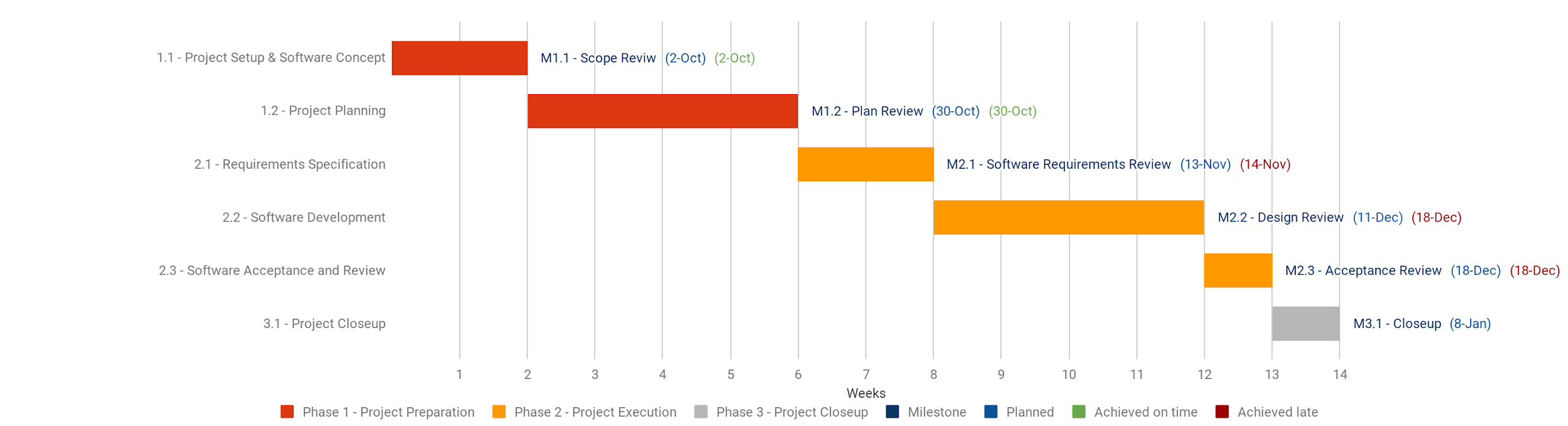
* Ruben Marques ([rubenantoniomarques@gmail.com](mailto:ruifilipetmota@gmail.com))
* Paulo Dias ([dias.pauloalex@gmail.com](mailto:dias.pauloalex@gmail.com))
* Sérgio Soares ([smcsoares20@gmail.com](mailto:smcsoares20@gmail.com))
* Rui Mota ([ruifilipetmota@gmail.com](mailto:ruifilipetmota@gmail.com))

### 

# 2. Gestão do Projeto

## 2.1. Ciclo de Vida do Projeto

Neste projeto foi usado o modelo em cascata (Waterfall) como guia para o seu ciclo de vida.

**

#### 

## 2.2. Entregas

A tabela seguinte mostra em todas as reuniões onde os vários documentos foram mostrados ao cliente para revisão (R) e como definitivos (B).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| KO |  | KO+2 |  |  |  | KO+6 |  | KO+8 |  |  |  | KO+ 12 | KO+ 13 | KO+ 14 |
| **Semanas** | **#1** | **#2** | **#3** | **#4** | **#5** | **#6** | **#7** | **#8** | **#9** | **#10** | **#11** | **#12** | **#13** | **#14** |
| **Deliverables** |
| Vision & Scope | R | B |  |  |  |  |  |  |  |  |  |  |  |  |
| Milestone Report 1.1 |  | R/B |  |  |  |  |  |  |  |  |  |  |  |  |
| Software Development Plan |  |  | R | R | R | B |  |  |  |  |  |  |  |  |
| Quality Assurance Plan |  |  |  | R | R | B |  |  |  |  |  |  |  |  |
| Milestone Report 1.2 |  |  |  |  |  | R/B |  |  |  |  |  |  |  |  |
| Software Requirements Specification |  |  |  |  |  |  | R | B |  |  |  |  |  |  |
| Risk Plan |  |  |  |  |  |  | R | B |  |  |  |  |  |  |
| Acceptance Test Plan |  |  |  |  |  |  | R | B |  |  |  |  |  |  |
| Milestone Report 2.1 |  |  |  |  |  |  |  | R/B |  |  |  |  |  |  |
| Software Architecture & Design |  |  |  |  |  |  |  |  | R | R | R | B |  |  |
| Milestone Report 2.2 |  |  |  |  |  |  |  |  | R | R | R | R/B |  |  |
| Acceptance Test Report |  |  |  |  |  |  |  |  |  |  |  |  | B |  |
| Quality Assessment Report |  |  |  |  |  |  |  |  |  |  |  |  | B |  |
| Milestone Report 2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  | R/B |
| Post-Mortem Analysis |  |  |  |  |  |  |  |  |  |  |  |  |  | B |
| Legenda: |  | | **| - Milestone** | | | S - Semana | | | R - Review | | | B - Baseline | | |

## 2.3. Escalonamento

Para fazer uma estimativa aproximada do desenvolvimento do projeto foi praticado o *Planning Poker*. As tarefas foram divididas em seis *work packages* que são:

1. **Especificações Iniciais- 16h**
2. **Lógica- 13h**
3. **Comunicação - 16h**
4. **Interface - 19h**
5. **Revisões - 16h**
6. **Qualidade - 31h**

Para mais informação consultar: [Estimativas](https://docs.google.com/spreadsheets/d/1IFXQ0EIGUWRzYK_qQ3lO11MdrS022KA0y15q2WM55uE/edit?usp=sharing)

De modo a conseguir visualizar melhor quem iria executar qual tarefa e a data de início de fim de cada uma delas foi feito um WBS (Work Breakdown Structure) que pode ser consultado aqui neste link: [WBS](https://drive.google.com/open?id=1LA6NuzYTYpZPYNxzBkCtzEIR5aMjrKlxPuq7o38L48g).

## 2.4. Milestones

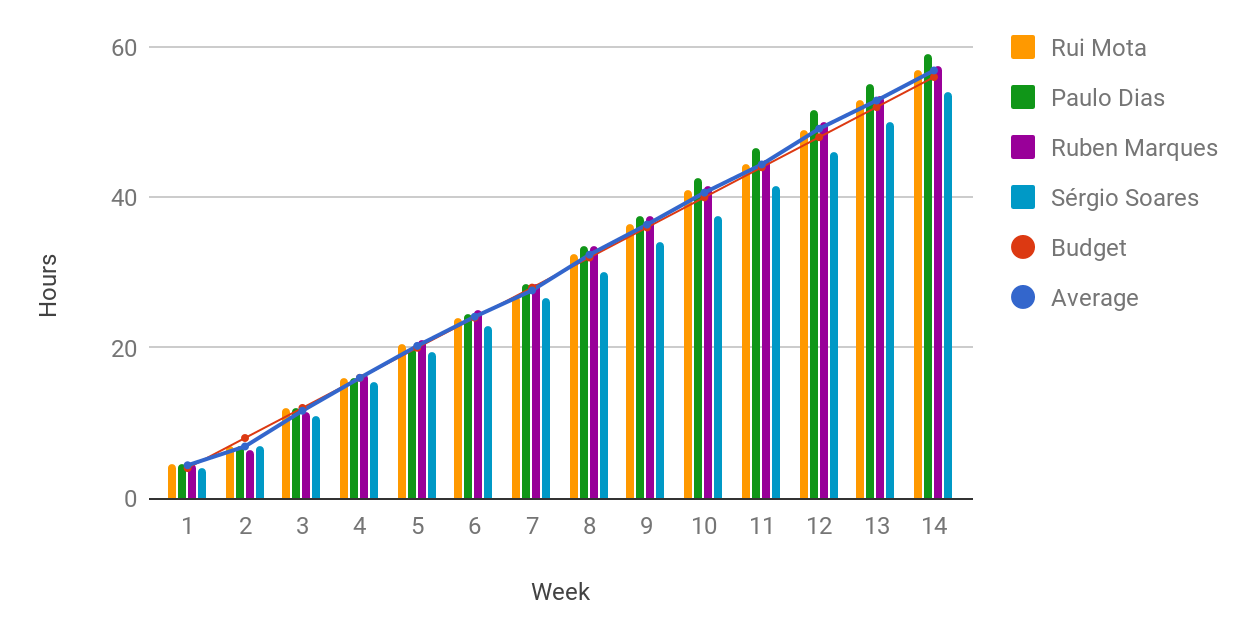
As Milestones deste projeto foram as seguintes:

|  |  |
| --- | --- |
| **Milestone** | **Fase de desenvolvimento analisada** |
| M.1.1 - Scope Review | 1.1. - Project Setup & Software Concept |
| M.1.2 - Plan Review | 1.2. - Project Planning |
| M.2.1 - Software Requirements Review | 2.1. - Requirements Specification |
| M.2.2 - Design Review | 2.2. - Software Development |
| M.2.3 - Acceptance Review | 2.3. - Software Acceptance & Review |

#### 

## 2.5. Custo / Esforço

O budget deste projeto era de 224 horas no total resultando em 4 horas semanais por pessoa. O esforço individual realmente investido pode ser visto aqui neste gráfico.



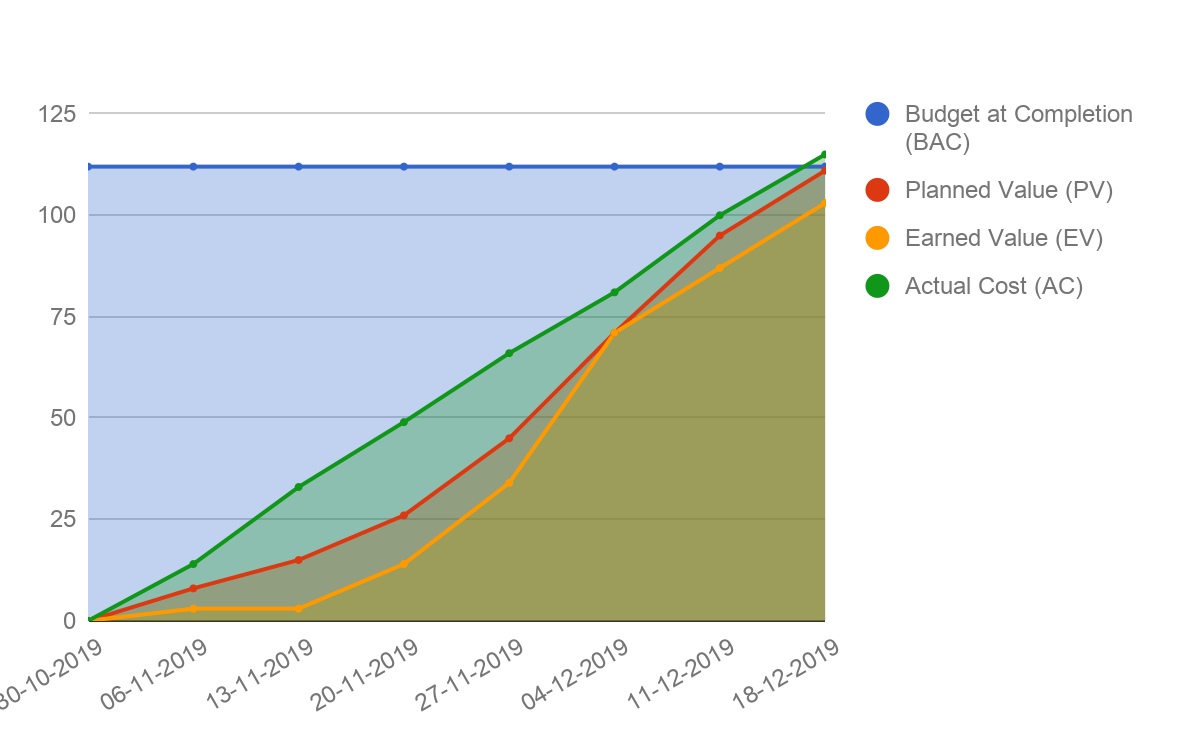
## 2.6 Repositórios

Para toda a documentação relativa à gestão do projeto foi usado o [Google Drive](https://drive.google.com/drive/folders/1x5uiuiPXDl-bAey21w0iOAht9gJhr1_F). Relativamente ao código desenvolvido da aplicação foi usado o repositório oficial do [GitHub](https://github.com/ruiftmota/GPS_BillSplitter).

## 2.7 Controlo do projeto

O escalonamento do tempo das diferentes tarefas foi seguido e registado no documento da Análise do Valor Agregado (EVA). No documento encontramos o título de cada tarefa, a data de começo e data de fim previstas e as respetivas datas de começo e de fim reais. Com isto é possível analisar o custo real da tarefa e o impacto dos respectivos atrasos no plano.

Para mais informações consultar: documento [EVA](https://docs.google.com/spreadsheets/d/1k6oMFXp9OCjeIh3vzVOLb0YihQwVOcbBWoIEhxVHkfk/edit?usp=sharing).

****

## 2.9 Planeado vs Realidade

|  |  |  |
| --- | --- | --- |
|  | **PLANEADO** | **REALIDADE** |
| **Esforço** | 224 horas | 227,5 horas |
| **Testes de integração** | Realizar os testes | Não foram realizados |
| **M.2.1 - Software Requirements Review** | Entrega a 6 de novembro | Entregue a 20 de novembro |
| **M.2.2 - Design Review** | Entrega a 4 de dezembro | Entregue a 11 de dezembro |

# 

# 3. Produto

## 3.1. Requisitos Iniciais

* Criação de eventos - a aplicação irá permitir criar eventos e adicionar pessoas aos mesmos
* Partilha de eventos com outros utilizadores
* Alteração do evento - será possível alterar os membros do evento
* Criação e edição de despesas - Adicionar e editar as diversas despesas
* Guardar informação relativa a uma despesa
* Partilha de informação relativa a uma despesa
* Apresentação do saldo - vai ser possível ver as dividas entre os respetivos membros dos eventos
* Adição de despesas específicas a um ou vários participantes e a aplicação adicionar essas dívidas apenas a esses elementos
* Permitir haver despesas que tenham vários contribuintes e a aplicação descontar automaticamente esses valores

## 3.2. Requisitos Finais

* Criação de eventos - a aplicação irá permitir criar eventos e adicionar pessoas aos mesmos
* Partilha de eventos com outros utilizadores
* Alteração do evento - será possível alterar os membros do evento
* Criação e edição de despesas - Adicionar e editar as diversas despesas
* Guardar informação relativa a uma despesa
* Partilha de informação relativa a uma despesa

### 

# 4. Documentos

## 4.1 Repositório de documentos

A tabela seguinte mostra um lista de todos os documentos feitos, a sua versão atual, a sua data de entrega e links para o documento.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status** | **Deliverable** | **Name** | **V** | **Date** | **Link (GDocs)** | **Link (Published)** |
| Published | D1.1.1 | V&S - Vision & Scope | 1.0 | 2/10/2019 | [Visão e Âmbito](https://docs.google.com/document/d/1eKGiR6PR1PbEegFGvOYozfc3hxIY9KplxAn4aXJPL7s/edit) | [Versão publicada](https://docs.google.com/document/d/e/2PACX-1vQvwv-8Zb30yNA2Vmt2d0wNPQshpQjIQP4euzVAnb_Q1GbsuNKSyfNBcYkhDW_2Wbh9KCCXqc5MDeVS/pub) |
| Published | D1.1.2 | MR1.1 - Milestone Report | 1.0 | 2/10/2019 | [Relatório MR1.1](https://docs.google.com/document/d/1QPYv9WC4iz9qhPz34XLGJYyfcDtn4bPoJXfSMBtSo8Y/edit?usp=sharing) | [Versão publicada](https://docs.google.com/document/d/e/2PACX-1vTsWREEY5V3-1rXc-w_yTDl1D6r4epqUK-KxS4IMPAp_Oaxa33rgnIUtsMbSJsnvqw_NHyMKc_cr2e2/pub) |
| Published | D1.2.1 | SDP - Software Development Plan | 2.0 | 28/10/2019 | [Software Development Plan](https://docs.google.com/document/d/16cLenSSNuOeOA6GZ0YwtORkFMzkai0NuZ6Weo8ZzUMk/edit?usp=sharing) | [Versão publicada](https://docs.google.com/document/d/e/2PACX-1vT7AqfTWlPbI8nepLFFZVkvNTqpl_iH2pN9ABj65tRO_MlbilU98VOEDCqTxq850zi3af7rw2cFG9U2/pub) |
| Published | D1.2.2 | QAP - Quality Assurance Plan | 1.0 | 30/10/2019 | [Quality Assurance Plan](https://docs.google.com/document/d/1TDonp8UE24Rb0jLbkN8gf__Tb60xucbgrxcPEkaGqBQ/edit?usp=sharing) | [Versão publicada](https://docs.google.com/document/d/e/2PACX-1vQ7UKnVs3JO91VKdTh8GYvnknuWRIYnNBXw-FAe4G87QsnqnxQlxruMLKLEGSya8xucaAqikNAVCG8y/pub) |
| Published | D1.2.3 | MR1.2 - Milestone Report | 1.0 | 30/10/2019 | [Relatório MR1.2](https://docs.google.com/document/d/1hG03znOzZSFqtz6Oglo1L0KU_QsE4sho19kycWJQCnY/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vSCRSE84ahZyJPbGhoESoTJ1IV-ZpWJdmwhSo9XDOk16_BZzYWzttccgY8QJNoLrPiNMPu5YoHpz_ow/pub) |
| Published | D2.1.1 | SRS - Software Requirements Specification | 1.0 | 6/11/2019 | [Software Requirements Specification](https://docs.google.com/document/d/166iWyW96q2MIrAJA39QLiA0KGNm1XJa05T1UOJeS0Mo/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vSCRSE84ahZyJPbGhoESoTJ1IV-ZpWJdmwhSo9XDOk16_BZzYWzttccgY8QJNoLrPiNMPu5YoHpz_ow/pub) |
| Published | D2.1.2 | RP - Risk Plan | 1.0 | 30/10/2019 | [Risk Plan](https://docs.google.com/document/d/1vTCMrS7wEmEW66sota9xDxM7nio5tzp4_LhfDTKTbDQ/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vSMjGUu4D6GkIusCCeboPvdnu0LlbM7v-HWNkwm4JmC2mqZnqbZOuJtzTqzZ1p-_aHV7XbmjMg-lalH/pub) |
| Published | D2.1.3 | ATP - Acceptance Test Plan | 1.0 | 6/11/2019 | [Acceptance Test Plan](https://docs.google.com/document/d/11rPbKqA_hU__X5CmnsS4MpUoYv3VNT_mFXIW7N8fVPM/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vR7FDXCYlE2LIKnC3QphpKQfJLcxP1jl9Gf2LkneX_IoyqfCgY7DuEy0UYbsRYcHHhSyuAE9UIhCaLT/pub) |
| Published | D2.1.4 | MR2.1 - Milestone Report | 1.0 | 20/11/2019 | [Relatório MR2.1](https://docs.google.com/document/d/1hG03znOzZSFqtz6Oglo1L0KU_QsE4sho19kycWJQCnY/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vQFrz4PXE7gY7750WdvrLY9AVxnNGPXbW4ax4ShFA3eQs9sIwV3igIaszkeN6P15ErvFhA1bvDorOI2/pub) |
| Published | D2.2.1 | SAD - Software Architecture & Design | 1.0 | 11/12/2019 | [Software Architecture & Design](https://docs.google.com/document/d/1l5PeHScucMFn1ziZSg9H42ReaDrff5GcAe2hSHMApog/edit) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vSl4BIrOW64G9gbXu2h50Se9PvILE07TLpXBI6W6fyULZxwu9XZBuBne7peReJcvaoexoaHc3E9_oZD/pub) |
| Published | D2.2.2 | MR2.2 - Milestone Report | 1.0 | 18/12/2019 | [Relatório MR2.2](https://docs.google.com/document/d/1JSuZaShewAUUtYI37hRgOI2EzbV1S-rkZKQVNW1u3qw/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vR2FL0-yDzuSgOjYaQAY5M4qRHsm12cCNU6C8HE_nq9gUZ2ml9miuM98qEYeaUQh3YVyXIfAbOPvHK-/pub) |
| Published | D2.3.1 | ATR - Acceptance Test Report | 1.0 | 18/12/2019 | [Acceptance Test Report](https://docs.google.com/document/d/1fjWJwnXTSyLaumv694DeiARm2aPEjxpOOdAm6MQspeM/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vRiNWNseT83JAmBTp8tUVFzgluopOk6ouIZI6J5rmj-nB1pMdkKJCGAV-R7_46sQWG-b39dm4QzyAR3/pub) |
| Published | D2.3.2 | QAR - Quality Assessment Report | 1.0 | 18/12/2019 | [Quality Assessment Report](https://docs.google.com/document/d/171hXxZgM8SyeAVnW9n9GlgPaAZyRwySR6JJQxO6-EJQ/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vTMvUFjCr4NRcD0Ww2X9d6hXfzellZ7SaivBh-2WncAggddpbOfWArLJQeHvvnUVGKapNgnqCU5VpAy/pub) |
| Published | D2.3.3 | MR2.3 - Milestone Report | 1.0 | 18/12/2019 | [Relatório MR2.3](https://docs.google.com/document/d/187ou-4SewBZMzTxquN9vBZ1QVcILn9qo2K-hm5Z-dgg/edit?usp=sharing) | [Versão Publicada](https://docs.google.com/document/d/e/2PACX-1vQfnZ3vLY2Y0sjS1F4dGDvyxQ2Rz7k1nya7SXP9s8Y2phJGCW3joP7ZB-Lu6Yn2JTwyQKxUPurx7PSo/pub) |
| Draft | D3.1.1 | PMA - Post-Mortem Analysis |  |  |  |  |

## 4.2 Ciclo de Vida dos Documentos

Os documentos tiveram um ciclo de vida baseado no modelo em Cascata. O modelo exige uma divisão em fases que terminam com um relatório no qual os documentos são dados por terminados ao serem aprovados pelo Cliente e por consequência deixam de poder ser alterados.

### 

# 5. Qualidade do Projeto

Foi planeada a realização de três tipos de testes: Unitários, de Integração e de Aceitação. Porém, por necessidade de orçamento para que se pudesse aplicar o plano de contingência e com o consentimento do nosso Cliente abdicou-se dos Testes de Integração.

## 5.1. Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Produto de Trabalho** | **Teste Planeado** | **# erros** | **Data** |
| *1* | Módulo Lógica (A) | Teste Unitário | *0* | *27/11/2019* |
| *2* | Módulo Interface (B) | Teste Unitário | *0* | 14/12/ 2019 |
| 3 | Módulo Comunicação (C) | Teste Unitário | 0 | 14/12/ 2019 |
| 4 | Teste de integração (Módulo A + C) | Teste Integração | \* | \* |
| 5 | Teste de integração (Módulo A + C + B) | Teste de Integração | \* | \* |
| 6 | Teste de aceitação | Teste de Aceitação | 0 | 18/12/ 2019 |
|  | Total | 4 | 0 |  |

#### 

## 5.2. Inspeções

As inspeções tiveram como principal objetivo identificar anomalias no documento SRS e no código do Módulo da Lógica. Foram encontrados 6 defeitos graves e 25 menores no documento SRS sendo corrigidos pela equipa supervisionada pelo Autor do documento, o Moderador aprovou o resultado final e por tanto não foi necessário outra inspeção ao dito cujo. Em relação ao código, foram encontrados 2 defeitos graves, 5 médios e 20 menores sendo corrigido pelo próprio Autor do código.

[Relatório Inspeção do SRS](https://drive.google.com/file/d/0B6-twnDTpvBycUtYbmJfMjBpYUd1WEVSWW12bDI3Z1FINVpz/view?usp=sharing)

[Relatório Inspeção do Código](https://drive.google.com/file/d/18kQKEPfW8HcG4Lkctp7HEJtfXarBO1DK/view?usp=sharing)

## 5.3. Change Requests

Apesar do modelo de Cascata não prever a alteração de documentos dados por terminados, achamos por bem realizar alterações para que fosse possível a realização da aplicação dentro do orçamento inicial.. Para tal foram pedidas duas “Change Requests” que posteriormente foram aceites pelo nosso Cliente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Mudança Requerida** | **Proponente** | **Data** | **Decisão** |
| 1 | Remoção de despesas com vários contribuintes e de despesas específicas a apenas alguns elementos de cada evento | Ruben Marques | 19/11  2019 | Aceite |
| 2 | Remoção da necessidade de utilização de uma VPN e inserção do pressuposto de a aplicação só poder ser testada numa rede local. | Ruben Marques | 19/11/  2019 | Aceite |
| # de mudanças aceites | | | | 2 |

#### 

## 5.4. Standards

Para garantir a qualidade do código e uma boa perceção do mesmo foram definidos *Coding Standards* que podem ser visualizados aqui em baixo.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Padrão de Código** | | | | | | | |
| 1º Padrão | Todas as classes Java serão separadas por ficheiros, haverá um ficheiro para cada classe | | | | | | |
| 2º Padrão | Nomes de variáveis bem definidos e escritos em minúsculo no seguinte formato: "nomeVariavelXpto" | | | | | | |
| 3º Padrão | Nomes de funções escritos com primeira letra em mínusculo e as seguintes em maiúsculo: "nomeFuncaoFazQualquerCoisa()" | | | | | | |
| 4º Padrão | Nomes de classes com as iniciais em maiúsculo (CamelCase): "NomeClasse" | | | | | | |
| 5º Padrão | Todos os atributos das classes serão privados de forma a encapsular bem a informação. Deverão ser acedidos e alterados apenas através de getters e setters. | | | | | | |
| 6º Padrão | Nomes de constantes em maiúsculas: "NOME\_CONSTANTE" | | | | | | |
| 7º Padrão | Linhas pouco extensas, máximo 120 caracteres. Mudar de linha depois de uma virgula ou operador. | | | | | | |
| 8º Padrão | Código bem comentado para um melhor resultado em equipa | | | | | | |
| 9º Padrão | Blocos de comentários a começar a linha com asterisco '\*' :"/\* | | | | | | |
|  |  |  |  |  |  | \* Bloco |
|  |  |  |  |  |  | \*/ |

## 5.5. Perceção da Qualidade

Para haver uma perceção da qualidade do produto foi preciso definir métricas que nos ajudassem a medir a qualidade. Aqui em baixo encontra-se a lista das métricas que foram definidas:

* **Confiança** - resultado dos testes unitários ao módulo da lógica (100% de testes aprovados)
* **Usabilidade** - serão feitos testes de usabilidade onde vão ser atribuídas classificação à usabilidade da aplicação (0-10)
* **Compreensibilidade** - relatório de avaliação de desk checks e de inspeção ao código / cumprimento dos coding standards
* **Código dividido em módulos** (>= 3)
* **Satisfação do cliente** (>= 85%) - O projecto irá ser avaliado consoante os documentos entregues em cada semana
* **Concordância com as datas estabelecidas no EVA** (>= 85%) - Será feito um rastreio semanal do documento EVA

Destas métricas a única que não foi cumprida foi a última, por existir uma discrepância de certa de 40% relativamente ao EVA

### 

# 6. Processos Técnicos

## 6.1. Ferramentas

Sendo o objetivo do projeto o desenvolvimento de uma aplicação cliente-servidor, este acaba por consistir em dois programas, ambos desenvolvidos em Java.

Sendo a parte relativa ao cliente feita tendo o ambiente Android como alvo, essa componente foi desenvolvida em Android Studio. A parte relativa ao Servidor pôde ser feita em VS Code.

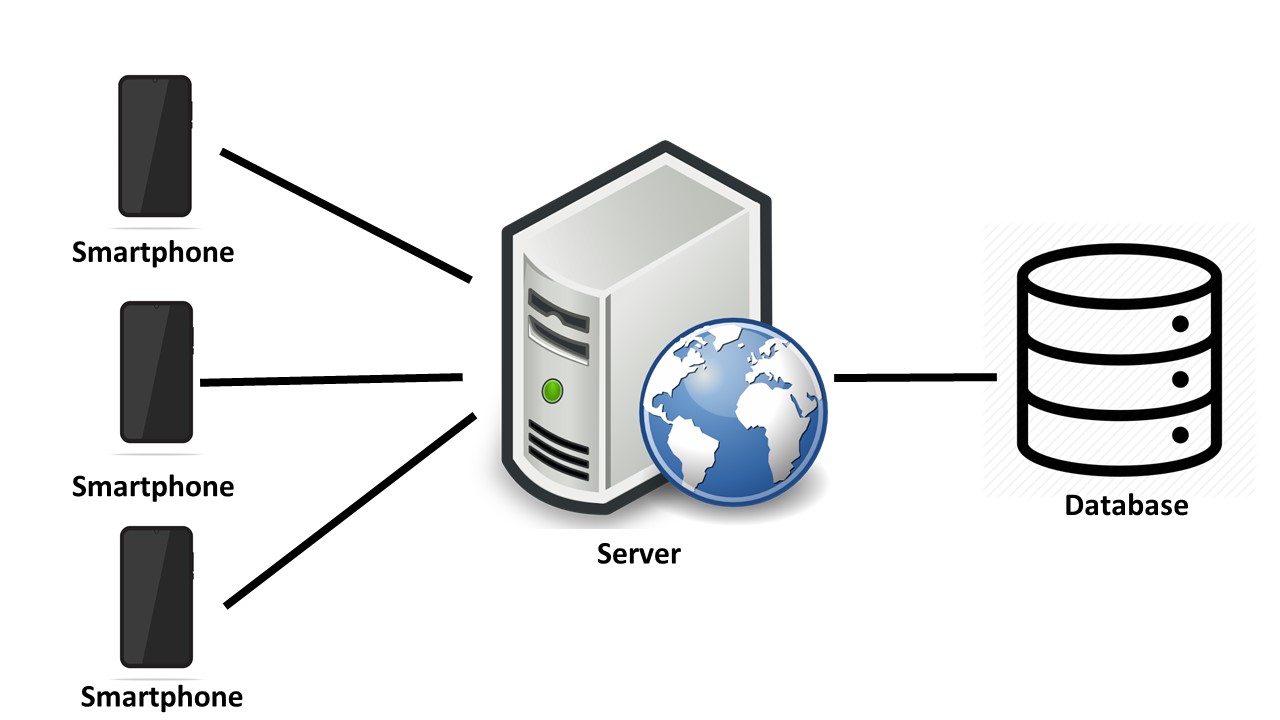
Relativamente aos componentes que foram testados da forma prevista foi utilizada a framework de testes JUnit.

O ter recorrido à linguagem Java para produção do código e o facto do código feito ter sido acompanhado de comentários, permitiu a utilização do componente JavaDocs do JDK para produção das páginas HTML da documentação.

## 6.2. Arquitetura

A arquitetura do sistema é baseada numa ideologia cliente - servidor. A nível de base de dados, vamos ter simplesmente um armazenamento no servidor destinado a guardar informações necessárias para a comunicação da aplicação.

Os clientes vão interagir com a aplicação através de dispositivos móveis com Android onde irá correr o cliente e onde ocorrerá a interação com o utilizador. Para permitir a partilha de eventos entre utilizadores de dispositivos distintos, os mesmos deverão ter acesso à internet. Do ponto de vista arquitetural, os clientes não vão ter nenhuma interação direta entre eles, já que toda a comunicação será feita através o servidor.



# 

# 7. Riscos

Como métodos de prevenção, foram definidos 8 Riscos e realizados Planos de Mitigação para os três com maior Probabilidade e Impacto.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Pri** | **#** | **P** | **I** | **Declaração de risco** | | **Indicadores** | **Ações** |
| 1 | RS-1 | 4 | 5 | Não obter conhecimento suficiente para concluir a interface a tempo da entrega de GPS | Pode resultar na necessidade de arranjar uma interface alternativa à de uma aplicação android | Desistência da cadeira de AM por parte da equipa | CP-1 |
| 2 | RS-2 | 3 | 5 | Problemas de ligação com o servidor | Pode resultar numa incapacidade de partilha de eventos entre utilizadores | Dificuldade em comunicar em máquinas de redes diferentes | MS-1 |
| 3 | RS-3 | 5 | 3 | Falha na estimativa | Incapacidade de conclusão das etapas previstas ou excesso de tempo dedicado | Trabalho a acumular de semana para semana | CP-2 |
| 4 | RS-4 | 1 | 5 | Desistência de um elemento da equipa | Sobrecarga do resto dos elementos da equipa | Estar descontente com o seu trabalho | Observ. |
| 5 | RS-5 | 1 | 3 | Avaria dos equipamentos | Possível perda de trabalho | Equipamentos antigos e baratos | Observ. |
| 6 | RS-6 | 2 | 1 | Indisponibilidade da equipa para se reunir | Produtividade não tão elevada. | Assinalar uma mesma hora específica para nos reunirmos todas as semanas | Observ. |
| 7 | RS-7 | 2 | 3 | Desentendimento entre membros da equipa | Dificuldade em decidir o que cada um deve fazer | Mau ambiente na equipa | Observ. |
| 8 | RS-8 | 4 | 3 | Não conseguir implementar as notificações dos eventos na aplicação | Pode ser chato um participante ser adicionado a um evento onde fica a dever e não ser informado do feito | Falta de experiência na disciplina de AM onde se aprende a matéria | Observ. |

Dos riscos listados, ocorreram os 3 principais e primeiros, sendo que os dois do topo se interligaram de certa forma.

## 7.1. Planos de Mitigação

**CP-1 Plano de Contingência: Não obter conhecimento suficiente para concluir a interface a tempo da entrega de GPS (P x I = 20)**

O objetivo do projeto é o desenvolvimento de uma aplicação móvel para Android. Como a aprendizagem das tecnologias necessárias está a ser feita numa cadeira em paralelo com o projeto, o conhecimento necessário pode não ser adquirido a tempo e horas.

Como plano de contingências será desenvolvida uma interface de texto, que permitirá testar a lógica e provar que esta está pronta para receber uma interface gráfica que possa vir a ser desenvolvida.

**MS-1 Plano de Minimização: Problemas de ligação com o servidor (P x I = 15)**

Quando o cliente lançar a aplicação, este enviará uma mensagem em broadcast para um porto específico onde o servidor estará à escuta (dentro da rede do ISEC). O servidor então procederá então em responder ao cliente, fornecendo o seu ip e porto de ligação TCP, e finalmente o cliente estabelece esta ligação com o servidor.

A correr mal, a rede do ISEC bloqueia-nos a transmissão inicial do cliente em broadcast e o user terá de introduzir manualmente o IP do servidor dentro da VPN.

**CP-2: Falha na estimativa (P x I = 15)**

Sendo o primeiro projeto do grupo planeado na íntegra, a probabilidade das falhas na estimativas afetarem significativamente o decorrer no projeto é grande. Desta forma e com vista em entregar uma aplicação minimamente funcional, é possível que se venha a cortar em alguns requisitos menos importantes ou a renegociar horas extra de esforço com o cliente e consequente aumento do budget.

Cortar em alguns requisitos de forma a nos conseguirmos manter dentro do budget ou renegociar horas de esforço com o cliente.

### 

# 8. Análise

## 8.1. Análise individual

### 8.1.1. Ruben Marques

Este projeto revelou ser um grande desafio para mim devido à sua rigidez e persistência. A parte mais desafiante a meu ver foi toda a documentação e burocracia de cada fase do projeto. Como até à data do início deste projeto os únicos trabalhos feitos foram os das outras unidades curriculares, nunca tinha estado em contacto direto com o processo de desenvolvimento de um software. No início a aplicação a desenvolver parecia ser relativamente simples e sem grandes desafios. No entanto não foi preciso esperar muito até começar a sentir a pressão de ter que trabalhar apenas e só 4 horas por semana. O trabalho rapidamente se começou a acumular e a fazer atrasar entregas de alguns documentos. Desses documentos foi o SRS que teve mais impacto.

O que este projeto mais me ensinou foi que ao ter uma documentação bem feita onde todos os detalhes, requisitos e questões arquiteturais estão bem definidos, o código passa a ser o menor dos desafios, já que nos limitamos a traduzir o que está escrito em português para a linguagem de programação usada no software. Isto aumenta bastante a produtividade e garante qualidade no produto final.

Outro aspeto importante foi ter de fazer o trabalho de cada em semana em 4 horas sem excessão à regra. Isto ensinou-me a ganhar mais foco mais nas minhas tarefas para conseguir respeitar o budget de horas.

As minhas principais contribuições para este projeto foram o desenvolvimento do servidor e o planeamento da arquitetura do sistema. A meio do projeto também decidi propor ao cliente dois change requests para reduzir o âmbito e assim possibilitar alcançar o bom sucesso do projeto.

Para concluir, este projeto foi uma boa experiência e principalmente uma boa lição para o nosso futuro como engenheiros.

### 8.1.2. Rui Mota

O meu contributo passou por todo o projeto à exceção da aplicação Android que servia de interface para o utilizador, ainda que tenha sido mais relevante em algumas partes do que outras.

Em termos de documentação do projeto, todo o grupo passou por quase toda a documentação, especialmente quando ainda não se tinha dado início ao desenvolvimento do código. A minha contribuição nesse ponto, além da parcela que me competia proporcionalmente em termos de conteúdo, passou também pela análise e correção ortográfica da maioria dos documentos.

Em termos de código fui responsável pelo desenvolvimento do Módulo da comunicação entre o servidor e a interface do cliente, ainda que tenha também passado os olhos pelo código do Servidor. Desta forma tive de ter em atenção as capacidades e limitações do sistema Android e as necessidades de informação no Servidor para processar todos os dados e verificações necessários.

No geral a equipa trabalhou bem, pois apesar da inexperiência de todos os elementos em basicamente todas as linguagens e API’s utilizadas, existiu sempre um ou mais elementos que se dedicassem a aprender o necessário de cada uma. Isso permitiu que, apesar de todos os problemas encontrados, a aplicação ficasse funcional e utilizável.

O projeto em si permitiu aprender a importância de toda a documentação realizada, com especial atenção aos requisitos e à arquitetura, que quando bem estruturados, permitem uma implementação quase linear de todos os requisitos sem que o programador se perca ou extravie com dúvidas. A execução dos testes unitários revelou também a sua importância porque apesar de serem maçudos de realizar, permitiam um teste muito mais rápido de toda a lógica.

### 8.1.3. Paulo Dias

Ao longo deste projeto passei por diversas tarefas e obtive diversos aprendizados. Comecei um pouco relutante quanto à nossa ideia de produto porém logo a aceitei e contribui para que o seu desenrolar fosse o melhor possível. Na primeira fase contribui igualmente no preenchimento da documentação e realizei os mockups e estive mais atento às versões de cada documento e da dashboard ao qual o Cliente gostava de rever constantemente, porém, durante as três primeiras semanas da segunda fase fui perdendo o foco no controlo das mesmas. Além disso, na segunda fase, dediquei a maior parte do meu tempo semanal ao desenvolvimento da interface que em paralelo com o resto do código, foi desenvolvido gradualmente ao longo das semanas.

Como membro do Grupo 31 e ao estar implicado no projeto BillSplitter aprendi que apesar do modelo cascata estar a perder o seu uso em desenvolvimento de Software, ainda tem muito potencial e em certos casos é até a melhor opção, aprendi que um gráfico como o EVA bem estruturado facilita bastante o controlo do desenrolar do desenvolvimento, aprendi que documentos como o SRS e SAD desenvolvidos previamente facilitam muito o desenvolver do código e aprendi a importância de prever os riscos para que, caso eles ocorram, ter forma de os remediar.

### 8.1.4. Sérgio Soares

Neste projeto foi utilizado o modelo em cascata , onde tivemos que passar por várias fases, desde o planeamento, à execução e à análise final. Foi um grande desafio pois é um projeto de relativamente grande escala e as horas semanais de trabalho eram limitadas.

No entanto foi possível aprender os vários processos para concluir o projeto cumprindo os seus objetivos. O planeamento do projeto ajudou bastante pois, na hora de programar, não se perdeu tempo a pensar como ia ser feito, aumentando bastante a produtividade.

O documento mais importante para para o nosso projeto talvez tenha sido o Risk Plan, pois nas últimas semanas ocorreram dois riscos que tínhamos identificado.Estes foram resolvidos rapidamente pois já tínhamos pensado na solução para cada um deles. Alguns requisitos não foram implementados, mas o esforço não ultrapassou por uma grande margem o budget.

O minha principal função foi garantir a qualidade dos documentos e do código, sendo o responsável por realizar os testes ao mesmo.

## 8.2. Análise do grupo

No início do projeto houve alguma dificuldade em arranjar uma ideia, mas acabamos por chegar a um consenso. A primeira fase correu relativamente bem e a documentação era bastante acessível. Isso iludiu-nos momentaneamente, mas foi sol de pouca dura. Com o início da segunda fase do projeto veio o SRS que, por não ficar de acordo com o que o cliente tinha idealizado, acabou por precisar de ser completamente remodelado, o que levou a substanciais horas de atraso. A partir do início dessa fase os vários elementos do grupo começaram a dedicar-se às coisas com as quais tinham mais prática e facilidade de adaptação, o que levou a que cada elemento assumisse o seu papel e que de certa forma ajudou a que nos complementasse-mos uns aos outros.

Apesar da ocorrência de dois dos três principais riscos selecionados, o grupo reagiu bem graças à antecipação das suas grandes probabilidade e impacto e ao plano estabelecido, ainda que tenham resultado num compromisso de não realização de testes de integração para que o plano de contingência.

O principal desafio do grupo foi a novidade das tecnologias utilizadas, uma vez que praticamente todas as tecnologias utilizadas estavam a ser aprendidas em cadeiras paralelas à de GPS.

Quanto às principais lições aprendidas, residem entre as principais a necessidade de utilização de tecnologias e API’s apropriados ao ambiente em que se tenciona desenvolver a aplicação e a importância de toda a documentação e da necessidade da sua persistência.

## 8.3. Conclusão

Este projeto teve como objetivo a aprendizagem de todo o processo de desenvolvimento do modelo de cascata.

Começando na primeira fase com a apresentação de uma proposta de projeto e de respectivo desenvolvimento ao cliente e com toda a documentação que viria a descrever a futura implementação da aplicação e as garantias de qualidade que mais tarde também viriam a ser dadas, aprendemos que é preciso cuidado e cautela não só com o orçamento dado, mas também com todas as promessas feitas. O tempo voa e os imprevistos saem caros.

Na segunda fase do projeto apercebemo-nos da necessidade da especificação exata dos requisitos, não só para que não existam dúvidas relativas ao pretendido pelo cliente, mas também para que não existam dúvidas da parte do programador em coisas que à partida não custariam tempo. Mais tarde, apercebemo-nos também que um SAD bem pensado e definido leva à ocorrência de muitos menos problemas.

Os testes, ainda que apenas tenham sido automatizados os unitários e do lado da lógica do servidor, levaram a que houvesse uma mais rápida perceção dos problemas espalhados pela lógica e também mais rápida garantia da sua correção.

Finalmente, ainda que com o decorrer do projeto nos tenhamos apercebido de vários dos problemas encontrados, uma discussão em grupo e uma Análise de Post-Mortem permitiram identificar algumas das falhas identificadas anteriormente e identificar possíveis soluções para um futuro.