



Gestão de Projeto de Software 2019/2020

Bill Splitter

D1.2.1 - Software Development Plan

Autores

- Paulo Dias (dias.pauloalex@gmail.com)
- Sérgio Soares (smcsoares20@gmail.com)
- Rui Mota (ruifilipetmota@gmail.com)
- Ruben Marques (rubenantoniomarques@gmail.com)

Estado

- Publicado

Versões Principais

- v0.1, 7 de Outubro, Rui Mota, Primeiro rascunho
- v0.2, 8 de Outubro, Ruben Marques, Desenvolvimento das ideias
- v0.3, 9 de Outubro, Sérgio Soares, Review do tópico 2.2
- v0.4, 16 de Outubro, Rui Mota, Correções
- v0.5, 23 de Outubro, Ruben Marques, Reestruturação do tópico 2
- v0.6, 28 de Outubro, Paulo Dias, Versão aprovada
- v1.1, 30 de Outubro, Rui Mota, Alteração de um cargo da equipa

Versões Publicadas

- v1, 28 de Outubro, Paulo Dias, Versão aprovada
- v2, 30 de Outubro, Rui Mota, Nova versão aprovada

Índice

[1. Introdução](#)

[2. Organização do Projeto](#)

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

[2.2. Entregas \(Deliverables\)](#)

[2.3. Gestão do Projeto](#)

[3. Baseline Plan and Control](#)

[3.1. Estimativas](#)

[3.2. Schedule](#)

[3.3. Rastreio e Controlo](#)

[4. Processo Técnico](#)

[4.1. Arquitetura do Sistema](#)

[4.2. Métodos, ferramentas de desenvolvimento e software](#)

[5. Documentos e registos](#)

1. Introdução

Este projeto visa criar uma aplicação que ajudará a dividir as despesas de um evento quando este envolve uma ou mais pessoas no ato da contribuição para o mesmo e é preciso saber quem deve quanto a quem.

Haverá comunicação entre dispositivos móveis, com o auxílio de uma VPN, que possibilitará partilha de eventos com todos os participantes. Os participantes serão notificados sempre que forem inseridos, em um novo evento. Os cálculos serão automáticos e seus resultados estarão acessíveis a qualquer momento.

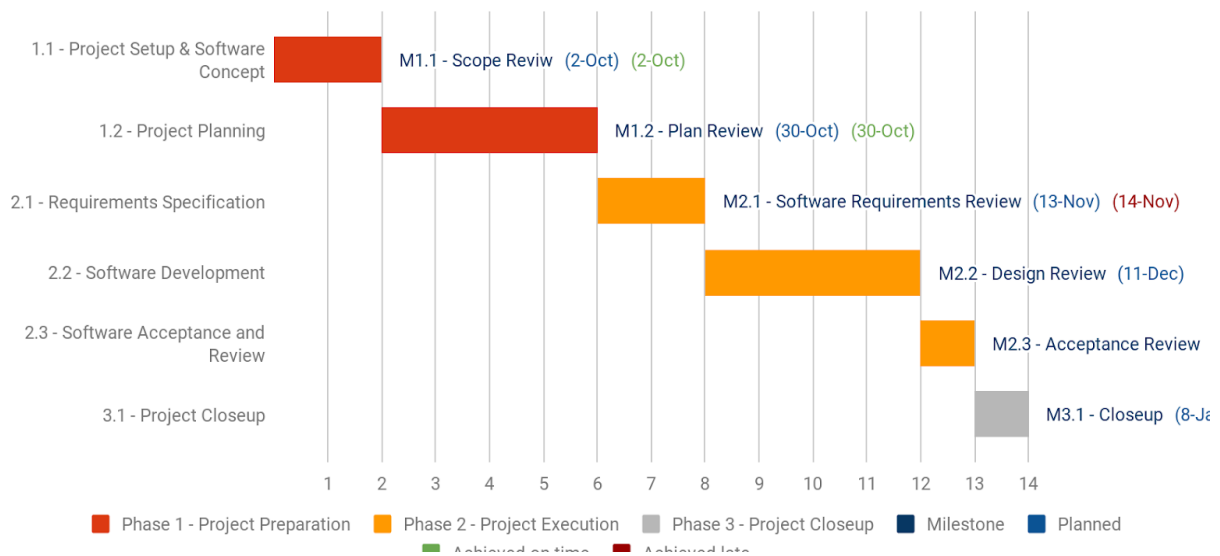
A aplicação será feita para Android no ambiente de Android Studio na linguagem Java ao longo de 7 semanas.

2. Organização do Projeto

2.1. Ciclo de Vida do Projeto

O modelo em Cascata foi projetado para o desenvolvimento de software e pretende definir uma ordem de corrente para as várias etapas.

O modelo tem como fundamentos base o facto de só avançar para a fase seguinte depois de a anterior estar devidamente feita e validada por todas as partes e o facto de não permitir alterações em fases anteriores que possam afetar as tarefas até então realizadas.



Para mais informação consultar: [Documento 'Baseline Plan'](#)

2.2. Entregas (Deliverables)

	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

Data previstas para os Milestones:

- MR1.1 - Milestone Report - 2 de Outubro de 2019
- MR1.2 - Milestone Report - 30 de Outubro de 2019
- MR2.1 - Milestone Report - 13 de Novembro de 2019
- MR2.2 - Milestone Report - 11 de Dezembro de 2019
- MR2.3 - Milestone Report - 18 de Dezembro de 2019
- MR3.1 - Milestone Report - 8 de Janeiro de 2019

Para mais informação consultar: [Documento 'Deliverables'](#)

2.3. Gestão do Projeto

Technical Manager: Ruben Marques

O Technical Manager deste projeto tem o desafio de estruturar a arquitetura do sistema e também a estrutura do código que vai ser programado. O seu trabalho principal é acompanhar o desenvolvimento da aplicação para garantir uma boa separação entre os diferentes módulos tais como a interface, a lógica e a comunicação. Também terá a responsabilidade de qualquer problema que derive da estruturação desta aplicação.

Quality Manager: Sérgio Soares

O Quality Manager tem como objetivo definir métodos standard que possam garantir a qualidade e eficiência elevada do projecto e que vá de encontro aos requisitos pedidos pelo cliente. É da sua responsabilidade assegurar a excelência de todos os documentos entregues ao cliente e definir as normas e orientações de segurança.

Project Manager: Paulo Dias

O Project Manager tem como função supervisionar todo o ciclo de vida do projeto, é responsável pela manutenção da DashBoard e decide sobre todas as solicitações para alterar os principais elementos do projeto, como entregas, cronograma e orçamento.

Risk Manager: Rui Mota

O Risk Manager trata de identificar, medir e avaliar diferentes tipos de riscos que podem afetar o projeto. Analisa o que pode correr mal, avalia o respetivo impacto e arranja soluções para minimizar, eliminar ou transferir o risco.

Todos os membros deste projeto serão também programadores.

3. Baseline Plan and Control

3.1. Estimativas

Para fazer uma estimativa aproximada do desenvolvimento do projeto foi praticado o Planning Poker. Esta técnica permite que sejam feitas estimativas com aceitação unânime do grupo, isto porque sempre que se propõe um valor esse tem de ser devidamente justificado e só é tomada uma decisão depois de todos estarem de acordo.

Foi decidido que o desenvolvimento do projeto iria ser repartido em 4 módulos e como tal foram feitas estimativas de modo a saber quanto tempo em média iria demorar cada módulo. Aqui apresentamos o custo em termos de horas de trabalho de cada módulo e em anexo indicamos a especificação desses valores

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

Para mais informação consultar: [Estimativas](#)

3.2. Schedule

O escalonamento do tempo das diferentes tarefas irá ser 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](#) e [WBS](#)

3.3. Rastreio e Controlo

O orçamento deste projeto será calculado em horas de esforço. Cada membro tem 4 horas semanais, o que resulta num esforço conjunto, por equipa, de 16 horas semanais e de 224 no final do projeto.

O rastreio e controlo do custo, mais especificamente, das horas irá ser feito através de uma spreadsheet do Google Docs onde cada membro regista as horas trabalho feitas com uma descrição do trabalho realizado naquele tempo.

O gráfico que reportará o esforço semanal ao cliente manter-se-á atualizado no Dashboard.

Horas de esforço da equipa: [TeamLog](#)

4. Processo Técnico

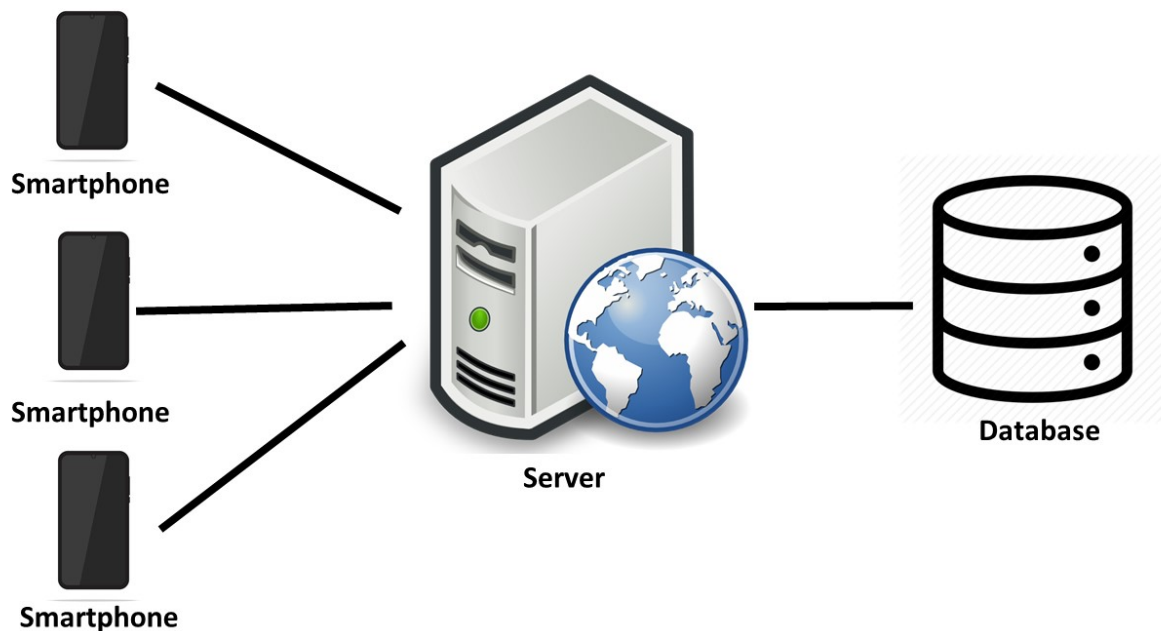
4.1. Arquitetura do Sistema

A arquitetura do sistema será 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 programa principal 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.

O servidor será um computador pessoal de um dos membros do projeto que irá ficar a correr no seu domicílio. A finalidade do servidor irá ser a sincronização dos eventos entre dispositivos para permitir a partilha dos mesmos entre os utilizadores.

O conceito de bases de dados aqui é bastante minimalista, já que, como referido acima, o armazenamento se vai limitar a guardar eventos destinados a serem enviados a outros utilizadores.



4.2. Métodos, ferramentas de desenvolvimento e software

O ambiente de desenvolvimento principal neste projeto irá ser o Android Studio v3.5.1, um IDE desenvolvido pela Google para produzir aplicações para Android. Este irá ser utilizado na realização da aplicação que irá correr nos dispositivos móveis e a versão mínima de Android para qual vai ser lançada a aplicação será a versão 7. A versão do JDK irá ser v8u231.

Para o desenvolvimento do programa que irá correr no servidor, iremos usar um IDE menos específico para programar em Java, o Visual Studio Code v1.39.

Para questões de testes de unidade irá ser usado o JUnit v5.

O equipamento utilizado para desenvolver a aplicação irá ser os computadores pessoais de cada membro do grupo, onde num deles irá correr o servidor.

Para enfrentar o problema de comunicação no que diz respeito a tentar aceder ao servidor através de uma rede externa àquela onde está ligado o servidor, iremos ligar o servidor a um VPN do ISEC e impor que os dispositivos móveis que utilizem a aplicação se liguem a essa VPN também para possibilitar a partilha de eventos.

A nível de software iremos separar o projeto em 3 grandes módulos: interface, comunicação e lógica. Cada módulo deverá ser o mais independente possível de modo a facilitar a repartição de tarefas e mais importante a programação e depuração dos mesmos.

O módulo da interface irá basear-se na programação da interação com o utilizador onde irão ser criadas as vistas necessárias para esse efeito.

A lógica irá ser o módulo onde serão desenvolvidos os cálculos necessários para o propósito que a aplicação serve e também irá ser responsável pela criação de todo o tipo de informação relacionada com os eventos.

O módulo da comunicação irá tomar conta da sincronização e partilha dos eventos entre os utilizadores e a salvaguarda dos mesmos.

5. Documentos e registos

Todos os documentos de suportes estão disponíveis em:

https://docs.google.com/spreadsheets/d/1Q33PfEBoTCfls352LXUSfcPROvb12J8btLoz7_ITqJk/edit?usp=sharing