

D1.2.1 - Plano de Desenvolvimento de Software

Autores

- Ana Filipa Costa Farinha Alves <analves07@gmail.com>
- Carolina Carmo Abrantes Lopes da Rosa <carolinalopesrosa@gmail.com>
- Diogo Santos Castelo Branco <diogosc7@gmail.com>
- João Pedro Aleixo e Jesus Pereira <jaleixo1993@gmail.com>
- Tiago João Cuevas Alves <tiagoalves0088@gmail.com>

Estado

- Publicado

Versões Principais

- v0.1, 04/10/2018, Carolina Rosa, “Introdução”, “Organização do Projeto: Ciclo de Vida do Projeto”
- v0.2, 04/10/2019, Ana Farinha Alves, “Organização do Projeto: Entregas, Gestão do Projeto”, “Plano Base e Controlo: Estimativa” e pesquisa em livro para fundamentação de “Estimativa”
- v0.2.1, 04/10/2019, Carolina Rosa e Tiago Alves, revisão de “Plano Base e Controlo: Estimativa” com ligeiras alterações e elaboração de texto para documento oficial
- v0.4, 07/10/2019, João Aleixo, Processo Técnico
- v0.4.1, 08/10/2019, Diogo Branco, Processo Técnico, Documentos e Registos de Suporte
- v0.4.2, 08/10/2019, João Aleixo e Diogo Branco, Processo Técnico e Revisão do documento
- v0.5, 11/10/2019, Diogo Branco e João Aleixo, Alterações na Gestão do Projeto e Processo Técnico
- v0.6, 15/10/2019, Ana Farinha Alves, revisão de “Plano Base e Controlo: Estimativa”
- v0.7, 15/10/2019, Diogo Branco, Revisão e Finalização do documento
- v0.8, 20/10/2019, Grupo, Alterações da reunião.
- v0.9, 24/10/2019, Tiago Alves e Ana Farinha Alves, Revisão de documento: alteração em 2.3 e 3.3; 3.1 em aberto para alteração

Versões Publicadas

- v1.0, 29/10/2019, Tiago Alves , Documento revisto e publicado, [Documento](#)

Índice

[1. Introdução](#)

[2. Organização do Projeto](#)

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

[2.2. Entregas:](#)

[2.3. Gestão do Projeto](#)

[3. Plano Base e Controlo](#)

[3.1. Estimativa](#)

[3.2. Plano Base](#)

[3.3. Controlo e Rastreamento](#)

[4. Processo Técnico](#)

[4.1. Arquitetura do Sistema](#)

[4.2. Métodos, ferramentas e técnicas](#)

[5. Documentos e Registos de Suporte](#)

63 1. Introdução

64 Atualmente, existem inúmeros parques de estacionamento privados que têm como défice
65 uma boa gestão do seu espaço. A sua utilização nem sempre é pacífica e exige, quase sempre,
66 que o utilizador dispense do seu tempo para encontrar um lugar de estacionamento disponível.
67 A crescente necessidade de modernizar e simplificar o processo de organização e utilização dos
68 parques de estacionamento conduz a este projeto.

69 Neste projeto, pretende-se desenvolver um sistema de gestão de parques de
70 estacionamento inovador, que seja robusto, simples e de fácil utilização. Pretendem-se criar
71 vantagens para o utilizador e para a entidade gestora do parque. Devido ao grau de
72 complexidade, o projeto passa por um levantamento dos requisitos, estimativas e arquitetura do
73 sistema a desenvolver. Tendo em conta que se pretende realizar um produto comercial, as
74 decisões tomadas na sua especificação tiveram em conta também fatores comerciais.

75 Focamos a atenção num sistema de gestão de parques de estacionamento que permite
76 ao administrador registar o veículo de um utilizador e atribuir-lhe um lugar específico no
77 estacionamento. O utilizador, ao chegar ao parque de estacionamento, aproxima o veículo da
78 entrada, o sistema faz a leitura da matrícula e permite a entrada do utilizador, caso este se
79 encontre devidamente registado. O utilizador dirige-se ao seu lugar fixo e reservado, com toda a
80 comodidade.

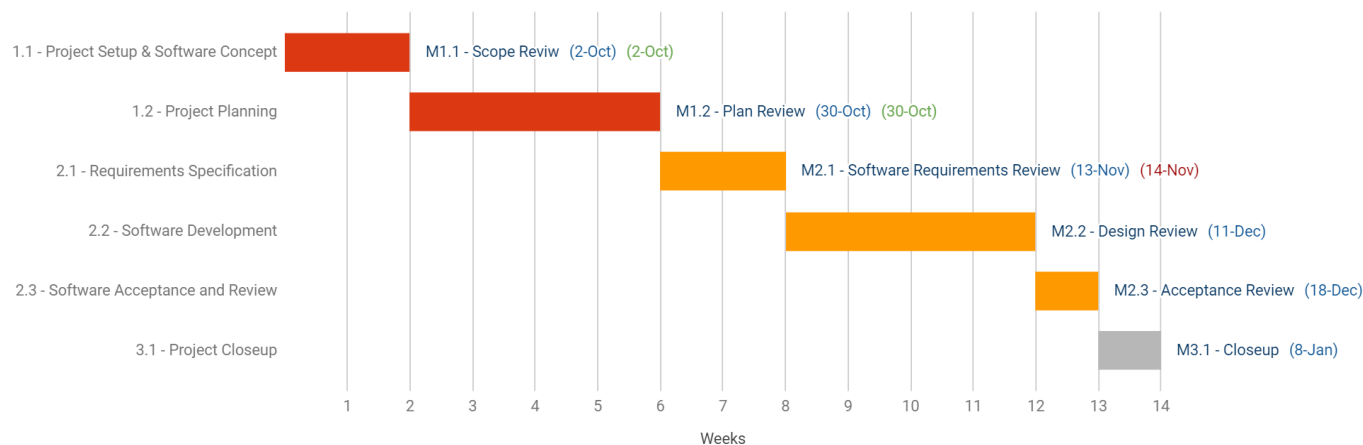
81 Esta solução permite ao administrador perceber quais as viaturas são permitidas no
82 acesso ao estacionamento, qual o lugar em que o cliente pode estacionar, o nome do proprietário
83 da(s) viatura(s) e a matrícula(s) associada(s) e, através deste sistema, impedir a livre entrada no
84 local de estacionamento e consultar estatísticas.

85
86
87
88
89
90
91
92
93
94
95

96 **2. Organização do Projeto**

97 **2.1. Ciclo de Vida do Projeto**

98 O ciclo de vida deste projecto é baseado no modelo *waterfall* dividido em seis fases
99 sistemáticas, que são descritas no gráfico seguinte:



100
101

Na tabela seguinte, é abordado as características das fases e milestones:

Fases	Milestones
1.1- Criação do novo projeto e conceção de ideias para o novo software	M1.1- Visão e âmbito são definidos.
1.2- Envolvimento da equipa no planeamento do projecto	M1.2- Requisitos iniciais propostos na reunião com o cliente.
2.1- Os requisitos do software considerados essenciais no Visão e Âmbito, na perspectiva do cliente, são especificados detalhadamente nesta etapa.	M2.1- Os requisitos são revistos, analisados e avaliados na reunião com o cliente. A conclusão da fase 2.1 está dependente da aprovação de todos os requisitos e respectiva documentação.
2.2- Desenho e criação do sistema (código), tendo em conta os requisitos específicos.	M2.2 - Desenho e arquitetura da aplicação serão revistos e verificados, com base documental nas fases anteriores do projecto e sujeitos a aprovação pelo cliente.
2.3 - Fase crucial do projecto inclui a verificação de testes e aceitação da	M2.3 - Análise e revisão do projecto com o intuito de preparação da sua entrega ao

qualidade,tendo como objectivo a preparação e entrega do produto final ao cliente.	cliente
3.1. Fase de término do projecto	M3.1- Entrega do produto final ao cliente

Nota: Após a fase 2.1 concluída se o cliente quiser ou achar que deve adicionar mais requisitos ao projeto, então o orçamento para a realização do mesmo deverá ser revisto.

2.2. Entregas:

Os documentos e datas de entrega das metas finais (milestones), que serão entregues e aprovados pelo cliente, constam na tabela infra apresentada:

Documentos	Semanas														Milestones(datas)
	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª	10ª	11ª	12ª	13ª	14ª	
D1.1.1-V&S	D+R	R	P(MR1.1)	-	-	-	-	-	-	-	-	-	-	-	MR1.1 (1/10/2019)
D1.2.1-SDP	-	-	D+R	R	R	P(MR1.2)	-	-	-	-	-	-	-	-	MR1.2 (29/10/2019)
D1.2.2-QAP	-	-	-	D	R	P(MR1.2)	-	-	-	-	-	-	-	-	MR1.2 (29/10/2019)
D2.1.1-SRS	-	-	-	-	-	D+R	R	P(MR2.1)	-	-	-	-	-	-	MR2.1 (13/11/2019)
D2.1.2-RP	-	-	-	-	-	D+R	R	P(MR2.1)	-	-	-	-	-	-	MR2.1 (13/11/2019)
D2.1.3-ATP	-	-	-	-	-	D+R	R	P(MR2.1)	-	-	-	-	-	-	MR2.1 (13/11/2019)
D2.2.1-SAD	-	-	-	-	-	-	-	D+R	R	R	R	P(MR2.2)	-	-	MR2.2 (3/12/2019)
D2.3.1-ATR	-	-	-	-	-	-	-	-	-	-	-	R	R+P(MR2.3)	-	MR2.3 (10/12/2019)
D2.3.2-QAR	-	-	-	-	-	-	-	-	-	-	-	R	R+P(MR2.3)	-	MR2.3 (10/12/2019)
D3.1.1-PMA	-	-	-	-	-	-	-	-	-	-	-	-	D+R	R+P	Closeup (17/12/2019)
Legenda:	Referência para Milestone														
D-Esboço (Draft)															
R-Revisto															
P-Publicado															

Tabela pode ser consultada através da seguinte hiperligação: [Tabela de entregas](#).

2.3. Gestão do Projeto

A equipa da SmartParking está estruturada da seguinte forma:

- **Gestor Técnico:** João Aleixo, responsável pela equipa de desenvolvimento do software, ser capaz de estruturar a equipa para desenvolver o produto da forma mais eficiente e resolver adversidades que possam aparecer no decorrer do projeto.
- **Gestor da Qualidade:** Carolina Rosa, responsável por certificar que o produto está dentro dos parâmetros estipulados pelo cliente, durante todo o ciclo de vida do projeto. Impor a utilização de ferramentas de teste e revisões regulares, sendo do software ou da documentação.
- **Gestor do Projeto:** Diogo Santos, assume a responsabilidade de gerir o projeto, mantendo a equipa organizada e com os seus objetivos bem claros, certificar que o cliente é informado de todos os acontecimentos importantes que acontecem de volta do produto e que o projeto cumpre o plano estipulado.
- **Gestor de Risco:** Ana Alves, responsável por antecipar os principais riscos que podem conduzir ao incumprimento dos objetivos ou qualidade do projeto. Juntamente com a equipa, faz um brainstorming dos principais riscos identificados e cria planos de mitigação para os principais. Todas as semanas durante a fase de desenvolvimento, revê o plano para atualizar o grau de risco dos já identificados e devido à possibilidade de surgirem novos riscos.
- **Engenheiro Informático de Backend e FrontEnd:** Tiago Alves, responsável pelo desenvolvimento das funcionalidades, da interface em que o utilizador vai interagir e o suporte do produto.

Nota: Contudo, estes papéis não são vinculativos, estando sujeitos a rotatividade. Todos os elementos irão desempenhar papéis de programador.

3. Plano Base e Controlo

3.1. Estimativa

No processo da estimativa, cada membro da equipa atribuiu um valor de esforço em cada uma das tarefas, correspondente à estimativa de esforço/horas em que demora a realizar essa tarefa.. Os valores utilizados e propostos variam entre 0, ½, 1, 2, 3, 5, 8, 13, etc. Para a realização do projeto temos como orçamento **140 horas**. Para estimar o esforço necessário para cada tarefa utilizamos o *Planning Poker*. Durante a sessão, o moderador recolhe os valores dos elementos e escolhe o mais adequado, onde é analisado o valor mais baixo e mais alto atribuído a essa tarefa. Para normalizar as estimativas utilizamos a seguinte fórmula:

$$\text{Estimativa normalizada} = \frac{\text{pior caso} + 4 * \text{caso normal} + \text{melhor caso}}{6}$$

No fim da sessão obtemos os seguintes resultados:

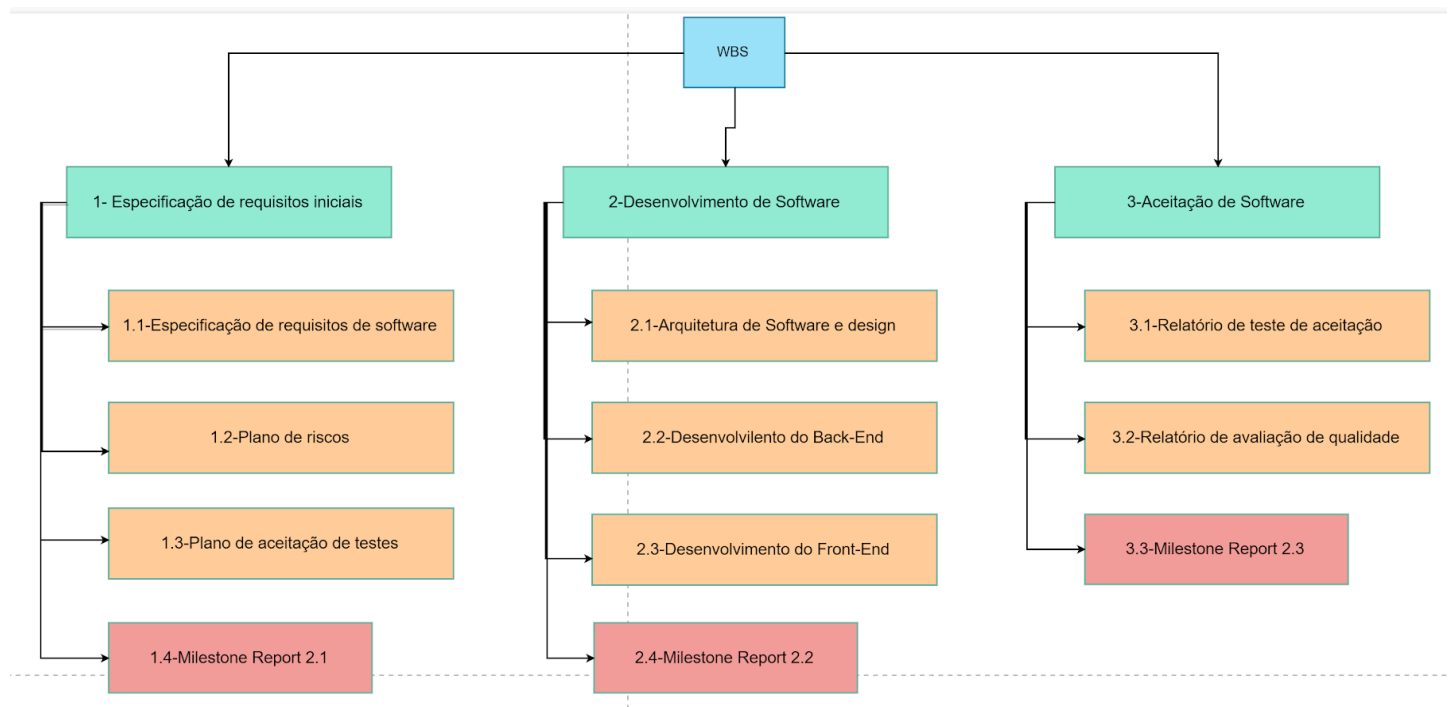
<i>Planning Poker</i>			
<i>Best Case</i>	<i>Worst Case</i>	<i>Chosen Case</i>	<i>Most Possible Case</i>
45	172	133	131

Como podemos observar, o resultado da nossas estimativas ficou perto do orçamento estipulado, 133 (esforço/horas) no Caso Escolhido e 131 (esforço/horas) no Caso Mais Possível (sendo a fórmula da Estimativa normalizada). Estes resultados não significam estimativas positivas ou negativas, mas sim o esforço que o grupo pensa ser necessário para realizar todas as tarefas com sucesso.

Tabela completa das estimativas: [Estimativas do Projeto](#)

3.2. Plano Base

No diagrama abaixo apresentado encontra-se uma representação gráfica dos principais documentos a desenvolver. Neste, é possível identificar aproximadamente a ordem cronológica de cada um deles pelo número que precede o nome de cada um. Para consulta em pormenor as datas do início e fim dos documentos, sugere-se a consulta e navegação nas várias tabelas do WBS, disponíveis através da hiperligação: [Discriminação das tarefas do WBS](#).



3.3. Controlo e Rastreamento

O controlo e rastreamento do projecto é realizado através do método *Earned Value Analysis* (EVA/Análise do Valor Agregado). É um método de avaliação do desempenho e progresso do projecto, em que o desempenho é medido pela comparação entre o valor agregado e o custo real, e o progresso é medido pela comparação entre o valor agregado e valor planeado.

Os elementos mais importantes do *Earned Value Analysis* são:

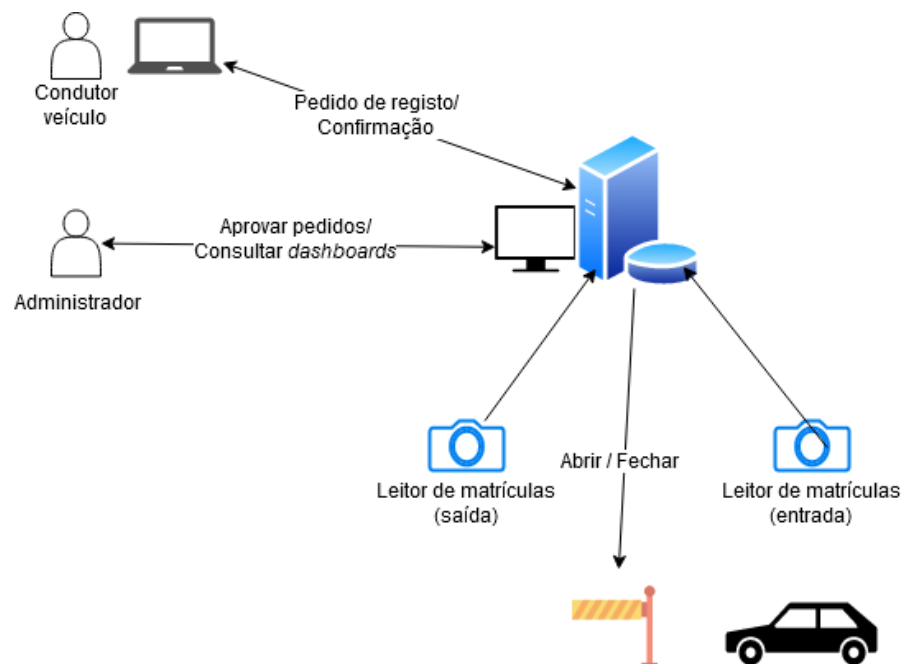
- PV (*Planned Value*/Valor Planeado);
- EA (*Earned Value*/Valor Agregado);
- AC (*Actual Cost*/Custo Real).

A análise do valor agregado integra a finalidade, o custo e o tempo, permitindo uma visão mais precisa da evolução do projecto. O gráfico resultante está disponível através da hiperligação: [EVA](#).

No controlo e rastreamento é também usado um team log em que todos os membros da equipa preenchem o seu esforço exercido e as tarefas efetuadas a cada semana ([Team Log](#)).

4. Processo Técnico

4.1. Arquitetura do Sistema



A arquitetura do sistema vai ser constituída pelos seguintes componentes:

- **Leitor de matrículas:**

- Existem dois leitores de matrículas que interagem com o sistema, o primeiro dedicado a ler a matrícula de um veículo que pretenda dar entrada no parque, este é responsável por efetuar a leitura da matrícula e por enviar, depois de devidamente processada, uma String com a matrícula do carro ao backend do sistema. Para além do leitor presente na entrada do parque haverá um segundo que estará no lado oposto e que fará o mesmo, mas para veículos que pretendam abandonar o parque. É de notar no entanto, que no desenvolvimento da aplicação estes leitores de matrículas serão simulados de forma a criar condições semelhantes ao mundo real e que não é feito qualquer desenvolvimento relativo a estes.

- **Back-end (Funcionalidades):**

- Vai ser responsável pela correspondência entre os leitores e o sistema, processado todos os sinais que lhe são enviados, e a base de dados. É neste componente que vai ser implementado todas as funcionalidades do sistema internamente e de suporte aos componentes externos.
- Ao receber um sinal dos leitores de matrícula, o sistema envia um pedido à base de dados, estando previamente estabelecida a ligação, para verificar a String que lhe é fornecida, comparando com os dados armazenados. Após o processamento, o sistema envia o sinal de confirmação aos leitores, caso a matrícula tenha sido reconhecida ou não.

- **Front-end do administrador (Interface):**

- Vai ser responsável por apresentar uma interface que possui todas funcionalidades do back-end, em que o administrador do sistema possa interagir. A interface tem de ser responsiva e fácil de utilizar, sendo composta por:
 - Um painel de administração, apresentando todas as funcionalidades destinadas ao administrador, sendo possível adicionar, remover e consultar informações relativas a cada lugar, de um veículo ou do proprietário.
 - Um painel com informações estatísticas como a média de entradas semanal, mensal ou em períodos definidos pelo utilizador do dashboard, assim como a consulta dos horários com menor e maior afluência.

251 • **Front-end do utilizador (Interface):**

- 252 ○ Esta será responsável por apresentar uma interface que possibilita que um
253 condutor de um veículo possa fazer um pedido de registo de um veículo. Serão
254 pedidas informações relativas ao veículo e assim como uma forma de contacto
255 com o utilizador (endereço eletrónico) de forma que após recusa/aprovação do
256 pedido por parte do administrador o utilizador seja informado da decisão.
257

258 **4.2. Métodos, ferramentas e técnicas**

259 No desenvolvimento deste projeto a nossa equipa de programadores vai utilizar como
260 linguagem de programação, o Java, sendo a linguagem mais versátil e em que todos estão mais
261 confortáveis em programar, será utilizada a versão 13 do JDK. Em termos do visual gráfico do
262 produto, este será desenvolvido recorrendo à versão 13 do JavaFX.

263 A base de dados será criada através do sistema de gestão de base de dados MySQL. A
264 interação base de dados/utilizador será efetuada através do componente de software JDBC
265 (*Java Database Connectivity*).

266 O software será desenvolvido através do IDE IntelliJ IDEA 2019, fornecido pelo JetBrains
267 e será utilizado o plugin SonarLint 4.1 desenvolvido pela SonarSource de forma a detectar e
268 corrigir problemas na qualidade do código. No que toca à realização de testes unitários será
269 utilizada a *framework* JUnit 5.5. Por fim de forma a manter um histórico de versões desenvolvidas
270 será utilizado o Git como sistema de controle de versões, especificamente será utilizada a
271 plataforma Github.

272 **5. Documentos e Registos de Suporte**

273 Os documentos de suporte no desenvolvimento deste projeto são os seguintes:

- 274 • **Team Log** ([Documento](#)): Apresenta o esforço semanal de cada elemento do grupo de
275 todas as semanas no desenvolvimento do projeto.
- 276 • **Dashboard** ([Site](#)): Contém uma introdução do projeto e da equipa envolvida, gráficos que
277 apresentam o progresso atual do projeto, informações sobre a atividade da equipa e
278 documentos associados ao projeto.
- 279 • **Project Estimation Report** ([Documento](#)): Utilizado para distribuição e designação de
280 tarefas, em que o produto a realizar é dividido em diversas tarefas, sendo avaliado o esforço
281 necessário para desenvolver cada tarefa do projeto.