

Projeto de Informática

2014/2015

Car Pooling

Realizado por:

1010834 – Vasco Fortuna

Resumo

Abstract

Índice

[1. Introdução 7](#_Toc432164522)

[1.1. Contexto do projeto 7](#_Toc432164523)

[1.2. Objectivos da aplicação 8](#_Toc432164524)

[1.3. Estrutura do relatório 9](#_Toc432164525)

[2. Estado de arte 10](#_Toc432164526)

[2.1. Análise de aplicações de boleias correntes 10](#_Toc432164527)

[2.1.1. Blablacar 10](#_Toc432164528)

[2.1.2. Boleia.net 11](#_Toc432164529)

[2.1.3. Pendura.pt 12](#_Toc432164530)

[2.2. Avaliação das aplicações 13](#_Toc432164531)

[3. Metodologia e Análise de requisitos 14](#_Toc432164532)

[3.1. Metodologia 14](#_Toc432164533)

[3.2. Planeamento 15](#_Toc432164534)

[3.3. Diagrama de contexto 16](#_Toc432164535)

[3.4. Diagrama de casos de uso 17](#_Toc432164536)

[3.6. Modelo ER 20](#_Toc432164537)

[3.7. Semântica 21](#_Toc432164538)

[3.7.1. Utilizadores 21](#_Toc432164539)

[3.7.2. Boleias 22](#_Toc432164540)

[4. Desenvolvimento 23](#_Toc432164541)

[4.1. Tecnologias utilizadas 23](#_Toc432164542)

[4.1.1. HTML 23](#_Toc432164543)

[4.1.2. CSS 23](#_Toc432164544)

[4.1.3. PHP 24](#_Toc432164545)

[4.1.4. Javascript 24](#_Toc432164546)

[4.1.5. AJAX 24](#_Toc432164547)

[4.1.6. MySQL 25](#_Toc432164548)

[4.2. Diagrama de hieraquia 25](#_Toc432164549)

[4.3. Desafios 26](#_Toc432164550)

[4.3.1. Visualização dos registos de boleias num único mapa de boleias 26](#_Toc432164551)

[4.3.2. Interface e interação eficiente da aplicação 27](#_Toc432164552)

[4.4. Avaliação da aplicação 28](#_Toc432164553)

[4.4.1. Testes 28](#_Toc432164554)

[5. Conclusão 29](#_Toc432164555)

[6. Bibliografia 30](#_Toc432164556)

Glossário

IPG- Instituto Politécnico da Guarda

PHP- PHP Hypertext Preprocessor

AJAX- Asynchronous Javascript and XML

HTML- HyperText Markup Language

XML - Extensible Markup Language

CSS- Cascading Style Sheets

SGBD - Sistema de Gestão de Base de Dados

SQL - Structured Query Language

# Introdução

O presente relatório carateriza o desenvolvimento do projeto feito pelo aluno Vasco Manuel de Deus Lello Fortuna, no âmbito da unidade curricular Projeto de Informática, na Licenciatura em Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda.

## Contexto do projeto

O crescente aumento dos custos de combustíveis (uma subida de preço que ocorreu desde 0,02€/litro até 1,68€/litro[[1]](#footnote-1)) e portagens tem fomentado a partilha de boleias em viaturas (car pooling) nas deslocações entre cidades. Dado o potencial número de interessados nessas partilhas, é necessário recorrer a ferramentas informáticas que facilitem a sua organização e gestão.

Especificamente, um grupo de docentes do IPG faz partilhas de boleias entre os seus membros. Eles utilizam uma folha de excel partilhada no Google Spreadsheet para organizar e planear as boleias entre si. Mas utilizar uma folha de excel para esta tarefa torna-se num processo muito demoroso e exaustivo, especialmente quando se pretende organizar boleias a longo prazo.

Existe uma grande variedade de ferramentas/aplicações online que fornecem o serviço de partilha de boleias entre os utilizadores. No entanto, estas ferramentas não providenciam a troca de informações entre grupos de utilizadores e não permitem o planeamento fácil de múltiplas viagens quer a curto ou a longo prazo.

Este projeto visa responder às necessidades do grupo, através de uma aplicação online que permita automatizar, ao máximo, o planeamento e a organização de boleias e tornar a sua utilização rápida e eficaz por parte do utilizador. A utilização da aplicação vai se centralizar num único mapa de boleias visível para todo o grupo. Este mapa consistirá num horário onde estão registadas todas as boleias e os seus atributos (hora, data, partida, etc). Existirá uma variedade de ferramentas e opções disponíveis aos utilizadores para conseguirem criar, aceitar, duplicar e costumizar ofertas de boleias dentro deste mapa .

O design da aplicação terá de ser simples e eficiente, de modo ao utilizador conseguir ler o mapa de boleias rapidamente e sem esforço. Assim, o design focar-se-á numa vista semanal do mapa de boleias. Esta vista consisterá de uma tabela preenchido com as horas do dia verticalmente e os dias da semana horizontalmente e cada espaço será preenchido consoante as boleias existentes. A cada utilizador será atribuído uma cor, que serão utilizadas para colorir cada boleia na tabela de modo a identificar rapidamente quem é o condutor.

Para maior usabilidade e acessibilidade, a aplicação será responsiva, ou seja, a aplicação web adaptará o seu formato consoante o tamanho do ecrã do dispositivo. Assim, pode ser visualizada a partir de qualquer dispositivo móvel sem perder eficiência do design.

## Objectivos da aplicação

Tendo em conta os pontos referidos anteriormente, podemos estabelecer várias soluções que a aplicação terá de apresentar:

1. Registar, alterar e eliminar utilizadores.

* Esta é uma funcionalidade básica para gestão dos membros existentes do grupo.

1. Adicionar e remover membros do grupo.

* Uma funcionalidade para permitir ao administrador adicionar e remover membros.

1. Organizar pedidos e ofertas de boleias dentro do grupo.

* Este é um objetivo essencial que permite a gestão e manuseamento de um mapa de boleias, no qual o projeto estará focalizado.

1. Contabilizar boleias efectuadas e recebidas.

* Este objetivo será aplicado em formato de estatísticas, de modo aos membros do grupo conseguirem ter maior controlo sobre as boleias que efetuam.

1. Duplicar mapas de boleias para semestres, anos.

* Uma funcionalidade para permitir a duplicação do mapa de boleias de modo a facilitar a sua utilização e melhorar a experiência do utilizador.

1. Preencher mapas automaticamente de acordo com as boleias contabilizadas de cada membro.

* Uma automatização da aplicação para ajudar a gerir as boleias entre os membros.

1. Enviar notificações por email acerca das próximas boleias ou alterações de boleias que afectam o utilizador.

* Estas notificações melhoram a interação entre os membros na gestão das boleias.

1. Calcular a redução  da  pegada  de  carbono.

* Este objectivo foi pedido, pelo orientador do autor, para os utilizadores terem uma noção do impacto das suas ações na luta contra o aquecimento global.

1. Sincronizar a  BD  das  boleias  com  o  Google  Spreadsheet.

* Outro objetivo pedido especificamente pelo orientador do projeto, de modo a ser possível transferir um mapa de boleias feito em Spreadsheet para a aplicação desenvolvida.

1. Criar cópias de segurança dos mapas de boleias.

* Esta é uma funcionalidade para garantir a integridade da base de dados.

1. Registar as alterações mais relevantes à base de dados.

* Manter um registo da actividade dos utilizadores é importante para melhorar a interatividade entre os utilizadores, a segurança de ações menos desejadas e a integridade do mapa de boleias.

1. Interface eficiente e intuitivo.

* Este é um objetivo essencial de modo a melhorar a experiência do utilizador da aplicação e o potencial da aplicação em termos de automatização e simplicidade.

## Estrutura do relatório

# Estado de arte

A próxima secção é dedicada à pesquisa de aplicações com funcionalidades semelhantes aos objetivos do projeto, com o intuito de ganhar informação e/ou conhecimento a ser utilizada no desenvolvimento do projeto. Todas as aplicações e informação relevantes a elas foram pesquisadas pelo autor.

## Análise de aplicações de boleias correntes

A maioria das aplicações web que se dedicam às boleias focalizam-se numa interação singular do utilizador, ou seja, cada utilizador planeia e interage somente com as suas boleias, sem ter acesso direto ou facilitado às dos outros utilizadores. Apesar de existir alguma variedade nas funcionalidades em cada aplicação, todas elas focalizam-se nesta interação singular, como se poderá observar nas seguintes aplicações estudadas pelo autor.

### Blablacar

Blablacar(www.blablacar.pt) é uma das aplicações web mais conhecidas para a partilha de boleias. A aplicação permite aos passageiros pesquisar por viagens, quer por ponto de partida quer por destino, previemente anunciadas pelos condutores. Os passageiros pagam ao condutor através da aplicação, na qual uma certa percentagem é revertida para a aplicação. Além disso, tanto o condutor quer como o carro são avaliados e comentados pelos passageiros. Estas avaliações servem como certificação aos passageiros da qualidade do condutor e da viagem. Também contém conexão ao facebook, de modo a ligar o perfil dos utilizadores ao facebook.

Na tabela seguinte (Tabela 1), estão descritas as funcionalidades da aplicação Blablacar:

|  |  |  |
| --- | --- | --- |
| Funcionalidade | Sim | Não |
| É responsivo? |  | X |
| É paga? |  | X |
| Tem registo de utilizadores? | X |  |
| É open-source? |  | X |
| Permite a duplicação de pedidos? |  | X |
| Permite editar ofertas depois de anunciadas? |  | X |
| Permite a criação de grupos de utilizadores? |  | X |
| Permite a criação de mapas de boleias? |  | X |
| Consigo duplicar ofertas a longo prazo? |  | X |
| Consigo reservar várias viagens a longo prazo? |  | X |
| Tem aplicação móvel? | X |  |

Tabela – Funcionalidades de blablacar (Fonte: Próprio)

### Boleia.net

Boleia.net(www.boleia.net) é uma plataforma de partilha de boleias. As boleias são anunciadas pelos condutores na aplicação, no entanto, a aplicação não gere a adesão dos passageiros às boleias. Para aderirem a uma boleia, os passageiros têm de contactar o condutor pessoalmente, através do contacto que ele coloca na aplicação. O preço é colocado pelo condutor em cada boleia, sendo o pagamento feito pessoalmente entre os intervinientes. Os passageiros podem avaliar e comentar o condutor na aplicação. A aplicação utiliza as redes sociais como o twitter e o facebook para divulgar as boleias.

Na tabela seguinte (Tabela 2), estão descritas as funcionalidades da aplicação Boleia.net:

|  |  |  |
| --- | --- | --- |
| Funcionalidade | Sim | Não |
| É responsivo? | X |  |
| É paga? |  | X |
| Tem registo de utilizadores? | X |  |
| É open-source? |  | X |
| Permite a duplicação de pedidos? |  | X |
| Permite editar ofertas depois de anunciadas? |  | X |
| Permite a criação de grupos de utilizadores? |  | X |
| Permite a criação de mapas de boleias? |  | X |
| Consigo duplicar ofertas a longo prazo? |  | X |
| Consigo reservar várias viagens a longo prazo? |  | X |
| Tem aplicação móvel? |  | X |

Tabela - Funcionalidades de boleia.net (Fonte: Próprio)

### Pendura.pt

Pendura.pt(www.pendura.pt) é uma plataforma mais simples que as anteriores. O utilizador(quer seja condutor ou passageiro) coloca um anúncio na aplicação juntamente com o seu contacto. O utilizador é posteriormente contactado pelos interessados que negociam os detalhes da boleia e do pagamento. Não existe nenhum tipo de avaliação dos condutores nem interação com as redes sociais.

Na tabela seguinte (Tabela 3), estão descritas as funcionalidades da aplicação Pendura.pt:

|  |  |  |
| --- | --- | --- |
| Funcionalidade | Sim | Não |
| É responsivo? | X |  |
| É paga? |  | X |
| Tem registo de utilizadores? |  | X |
| É open-source? |  | X |
| Permite a duplicação de pedidos? |  | X |
| Permite editar ofertas depois de anunciadas? |  | X |
| Permite a criação de grupos de utilizadores? |  | X |
| Permite a criação de mapas de boleias? |  | X |
| Consigo duplicar ofertas a longo prazo? |  | X |
| Consigo reservar várias viagens a longo prazo? |  | X |
| Tem aplicação móvel? |  | X |

Tabela - Funcionalidades de pendura.pt (Fonte: Próprio)

## Avaliação das aplicações

A tabela a seguir (Tabela 4) apresenta todas as funcionalidades das aplicações estudadas, comparando-as com a aplicação a ser desenvolvida:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Funcionalidade | Blablacar | Boleia.net | Pendura.pt | Carpooling |
| É responsiva? | S | S | N | S |
| É paga? | N | N | N | N |
| Tem registo de utilizadores? | S | S | N | S |
| É open-source? | N | N | N | S |
| Permite a duplicação de pedidos? | N | N | N | S |
| Permite editar ofertas depois de anunciadas? | N | N | N | S |
| Permite a criação de grupos de utilizadores? | N | N | N | N |
| Permite a criação de mapas de boleias? | N | N | N | S |
| Consigo duplicar ofertas a longo prazo? | N | N | N | S |
| Consigo reservar várias viagens a longo prazo? | N | N | N | S |
| Tem aplicação móvel? | N | N | S | N |

Tabela - Comparação de todas as aplicações estudadas (Fonte: próprio) (S: Sim; N:Não)

Apesar de constituírem sistemas de boleias, a interação destas aplicações é feita apenas entre o passageiro e o condutor. Esta interação vai contra o objetivo principal da aplicação que é a interação entre um grupo de amigos e a visualização de um único mapa de boleias entre eles todos. Devido a este ponto, não existem funcionalidades específicas das aplicações estudadas que se possam adaptar ao projeto corrente.

Apesar de também existerem grupos online a partilhar boleias entre si, estas partilhas são feitas em aplicações que não são especificadas para este propósito. Normalmente, estes grupos são criados em rede sociais (ex: facebook). Por estes motivos, o projeto será construído de raiz e não utilizará aplicações ou ferramentas existentes.

# Metodologia e Análise de requisitos

## Metodologia

Durante todo o processo do projeto, será utilizado uma variante do desenvolvimento agíl: a programação extrema (XP). O desenvolvimento XP carateriza-se por etapas de desenvolvimento curtos, o que possibilita uma revisão frequente do projeto, de modo a aumentar produtividade e a introduzir pontos de referência, nos quais novos requerimentos podem ser adotados (1). Esta metodologia foi selecionada de modo a obter maior interação com o orientador do projeto e, ao mesmo tempo, manter um ritmo de programação simples mas eficiente.

XP focaliza-se em inicializar e construir o projeto com a solução mais simples e em adicionar funcionalidades extra mais tarde. Assim, ao dividir o processo de criação de software em várias iterações, minimiza os riscos de desenvolvimento de software. Cada iteração desta metadologia procura adicionar um conjunto de funcionalidades ao produto final e cada iteração contêm as quatro fases de desenvolvimento:

* Planeamento
* Design
* Codificação
* Testes

Na fase de planeamento, o autor discutiu e documentou com o orientador todos os requisitos de software necessários para o produto final.

Na fase de design, foram desenhados e documentados os vários esquemas UML e as tecnologias necessárias para desenvolver a aplicação.

Na fase de codificação, desenvolveu-se o código para o projeto.

Na fase final de testes, testou-se a aplicação de modo a garantir que está de acordo com os requisitos de software e para garantir a sua coerência e usabilidade.

## Planeamento

## Diagrama de contexto

O diagrama de contexto é composto por fluxos de dados que mostram as interfaces entre o sistema e as entidades externas (2). O seu principal objetivo é simplificar a interacção de actores ou sistemas externos com a nossa aplicação. Neste caso, a única interação externa que temos é o utilizador. Além disso, as ações dos utilizadores estarão focalizadas na interatividade com as boleias, como é demonstrado no esquema seguinte (Figura 1):

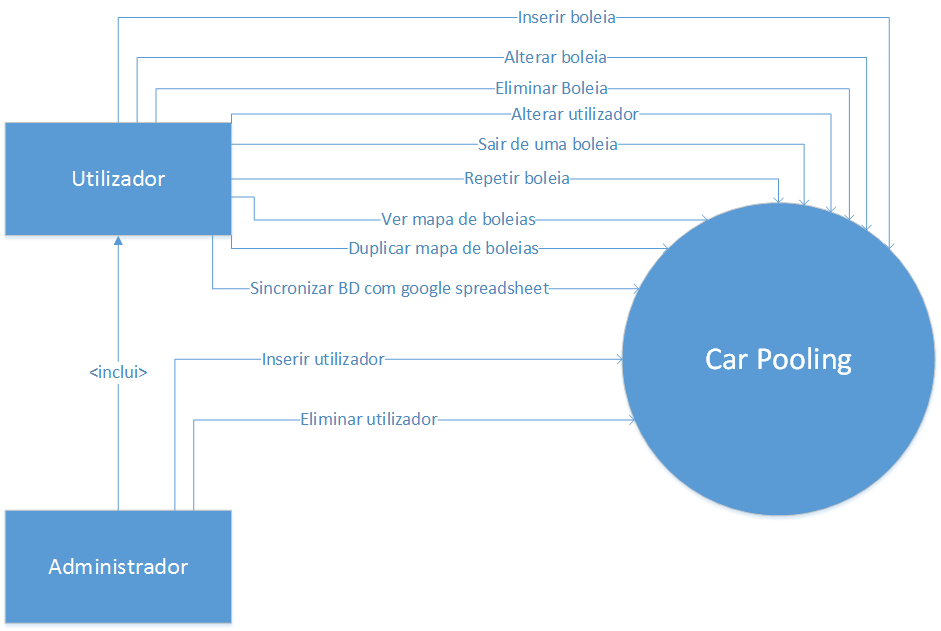


Figura - Diagrama de Contexto (Fonte: Próprio)

## Diagrama de casos de uso

O diagrama de caso de uso descreve a funcionalidade proposta para um novo sistema que será projetado (3). Este diagrama mostra todas as funcionalidades do sistema e de que modo interage com o utilizador.

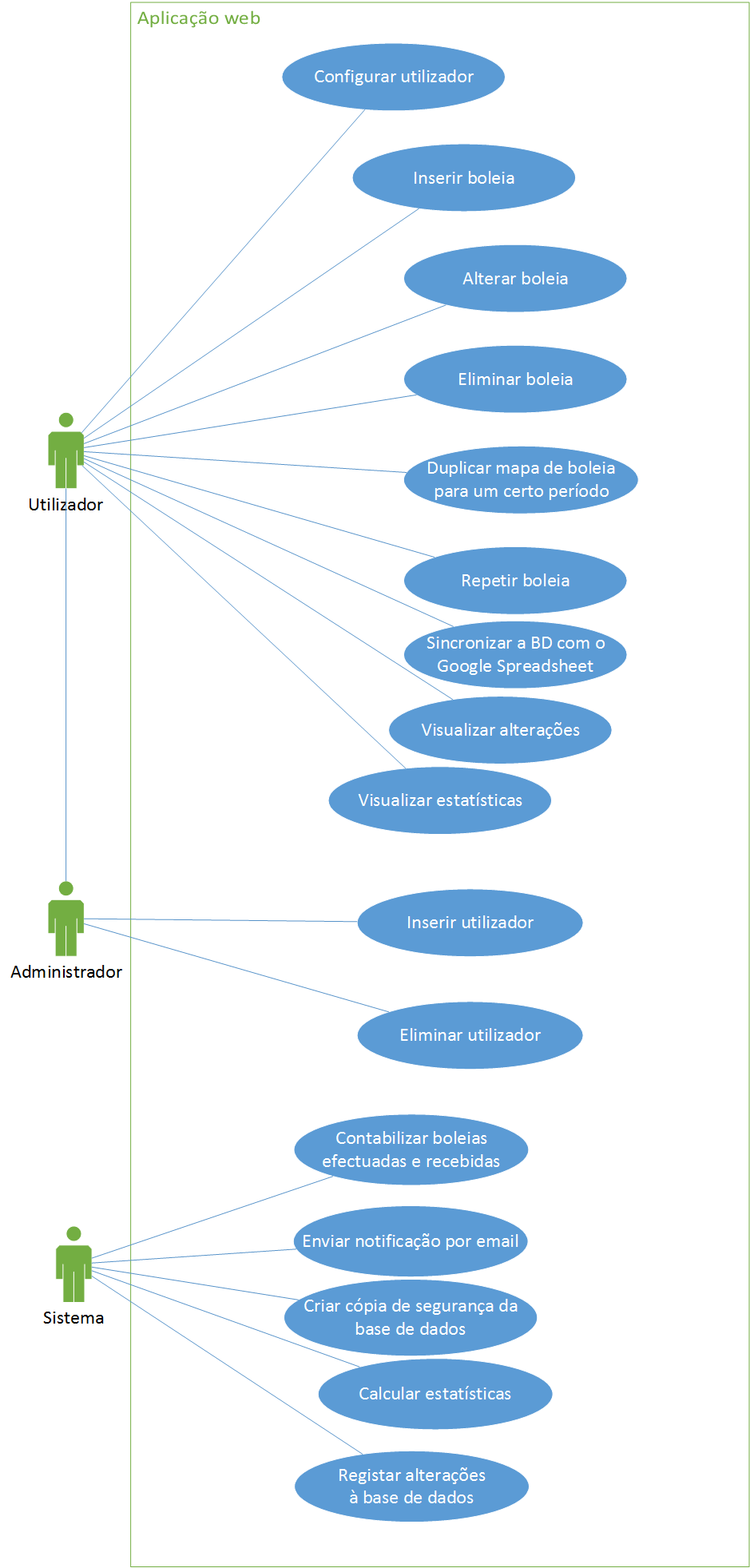


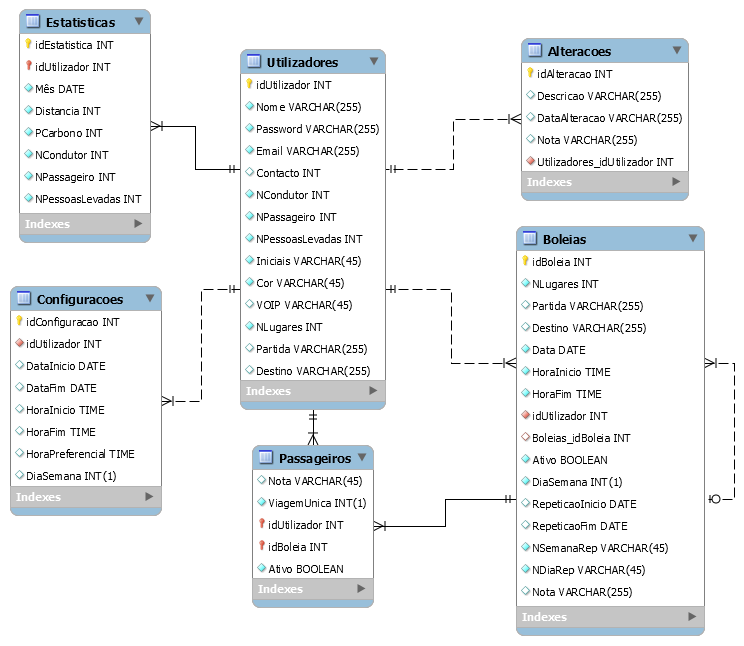
Figura - Diagrama de Casos de Uso (Fonte: Próprio)

#### Descrição de casos de uso

As descrições de caso de uso explicam detalhadamente como cada caso de uso irá funcionar e em que condições irá funcionar. Neste relatório (Anexo A) são apresentadas as descrições dos casos de uso mais relevantes e essenciais ao projeto, que são aqueles ligados à gestão das boleias e dos utilizadores.

## Modelo ER

Aqui é apresentado o modelo ER (Entity-Relationship model) desenvolvido, consoante os requisitos mencionados anteriormente. É de notar que o campo DiaSemana na tabela boleias encontra-se desnormalizado, indicando o dia da semana (Segunda – Domingo) em que se encontra a boleia.



## Semântica

Nesta secção é apresentada o dicionário de dados e a tabela de operações às tabelas da base de dados mais relevantes ao projeto.

### Utilizadores

Dicionário de dados:

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Tipo de dados | Descrição | Restrições |
| IdUtilizador | Int | (PK) Id do utilizador | - Não nulo  - Únicos |
| Nome | Varchar(255) | Nome do utilizador | - Não nulo |
| Password | Varchar(255) | Password do utilizador | - Não nulo |
| Email | Varchar(255) | Email do utilizador | - Não nulo  - Único |
| Contacto | Int | Contacto do utilizador | - Não nulo |
| NCondutor | Int | Nº de vezes que o utilizador foi condutor | - Não nulo |
| NPassageiro | Int | Nº de vezes que o utilizador foi passageiro | - Não nulo |
| NPessoasLevadas | Int | Nº de passageiros que o utilizador levou | - Não nulo |
| Iniciais | Varchar(45) | Iniciais do utilizador | - Não nulo |
| Cor | Varchar(45) | Cor identificadora do utilizador | - Não nulo |
| VOIP | Varchar(45) | VOIP do utilizador |  |
| NLugares | Int | Nº de passageiros que o utilizador pode levar, por defeito | - Não nulo |
| Partida | Varchar(255) | Partida, por defeito, do utilizador |  |
| Destino | Varchar(255) | Destino, por defeito, do utilizador |  |

Operações

|  |  |
| --- | --- |
| Nome | Descrição |
| Inserir() | Operação que permite inserir um utilizador   1. Introduzir Nome, Password, Email, Contacto, Iniciais, Cor, VOIP, Nlugares, Partida e Destino 2. O sistema gera o IdUtilizador |
| Alterar() | Operação que permite alterar um utilizador   1. Selecionar um utilizador 2. Alterar os campos necessários e que forem permitidos 3. Atualizar o utilizador |
| Consultar() | Operação que permite consultar um utilizador   1. Selecionar um utilizador 2. O sistema mostra os detalhes desse utilizador |
| Eliminar() | Operação que permite eliminar um utilizador   1. Selecionar um utilizador 2. O sistema pede a confirmação da eliminação 3. Eliminar o utilizador |

### Boleias

Dicionário de dados:

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Tipo de dados | Descrição | Restrições |
| IdBoleia | Int | (PK) Id da boleia | - Não nulo  - PK  - Únicos |
| NLugares | Int | Nº de passageiros que podem aderir à boleia | - Não nulo |
| Partida | Varchar(255) | Partida da boleia |  |
| Destino | Varchar(255) | Destino da boleia |  |
| Data | Date | Em que dia a boleia se realizará | - Não nulo |
| HoraInicio | Time | Hora em que a boleia começa | - Não nulo |
| HoraFim | Time | Hora em que a boleia termina | - Não nulo |
| IdUtilizador | Int | (FK) Id do condutor da boleia | - Não nulo  -FK  - Únicos |
| Boleias\_IdBoleia | Int | (FK) Id da boleia pai, no caso de repetição de boleias | - FK  - Únicos |
| Ativo | Boolean | Indica se a boleia está ativa | - Não nulo |
| DiaSemana | Int(1) | Dia da semana da boleia | - Não nulo |
| RepeticaoInicio | Date | Se houver repetição de boleias, o dia em que ela começa |  |
| RepeticaoFim | Date | Se houver repetição de boleias, o dia em que ela termina |  |
| NSemanaRep | Varchar(45) | No caso de a repetição ser mensal, a semana em que se repete. | - Não nulo |
| NDiaRep | Varchar(45) | No caso de a repetição ser semanal ou mensal, o dia em que se repete. | - Não nulo |
| Nota | Varchar(255) | Nota adicional da boleia, preenchida pelo condutor |  |

Operações

|  |  |
| --- | --- |
| Nome | Descrição |
| Inserir() | Operação que permite inserir uma boleia   1. Introduzir data, idutilizador, horainicio, horafim; 2. O sistema gera o IdBoleia |
| Alterar() | Operação que permite alterar uma boleia   1. Selecionar uma boleia 2. Alterar os campos necessários e que forem permitidos 3. Atualizar a boleia |
| Consultar() | Operação que permite consultar uma boleia   1. Selecionar uma boleia 2. O sistema mostra os detalhes dessa boleia |
| Eliminar() | Operação que permite eliminar uma boleia   1. Selecionar uma boleia 2. O sistema pede a confirmação da eliminação 3. Eliminar a boleia |

# Desenvolvimento

Neste capítulo serão apresentadas as tecnologias utilizadas para o desenvolvimento da aplicação, os desafios de maior complexidade que aparecerem durante todo o desenvolvimento da aplicação com as respetivas soluções e a avaliação do produto desenvolvido.

## Tecnologias utilizadas

### HTML

HTML (abreviação para a expressão inglesa HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada para produzir páginas na Web. Documentos HTML podem ser interpretados por navegadores (4). HTML é o bloco de construção mais básico de uma página web e é utilizado para criar e visualmente representar uma página web. Ela determina os conteúdos de uma página web mas não a sua funcionalidade.

Esta linguagem tornou-se no padrão e base para criação de qualquer página web. Como o autor decidiu desenvolver a aplicação desejada em página web, esta linguagem tem de obrigatoriamente ser usada.

O autor também já tem experiência em usar esta linguagem não só de projetos anteriores realizados mas também de matérias estudadas em certas disciplinas específicas de programação para a web. Pelo que a utilizagem desta linguagem não será um desafio, no entanto ela será estudada e revisada, de modo não só a garantir a integridade da aplicação desenvolvida mas também, possivelmente, a introduzir novas boas práticas de programação e novos conceitos.

### CSS

Cascading Style Sheets (ou simplesmente CSS) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento (5).

CSS permite criar e alterar conjuntos de propriedades de estilo. Estas propriedades permitem alterar o design e a visualização de qualquer elemento HTML, seguindo um conjunto de regras impostas pelo programador. Assim, com esta linguagem, consegue-se criar layouts e designs específicos de modo eficiente.

Especificamente, vai ser utilizado um template de CSS chamado Bootstrap. Este template está pré-configurado para alterar os elementos básicos de HTML não só para designs mais complexos mas também para tornar páginas web responsivas. Esta responsividade é responsável por modificar o interface da página de modo a adptar-se ao ecrã do dispositivo. Assim, a página consegue ser utilizada e visualizada de modo intuitivo e eficiente quer em telemóveis quer em computadores.

### PHP

PHP (*"PHP: Hypertext Preprocessor"*, originalmente *Personal Home Page*) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web (6).

Decidiu-se utilizar o PHP pelas seguintes razões:

1. A experiência do autor nesta linguagem de programação. O autor teve várias disciplinas dedicadas à programação na internet, incluindo Programação em PHP. Graças a estas disciplinas, o autor conseguiu uma introdução à linguagem e ao seu funcionamento.
2. A vontade do autor de aprender mais sobre as linguagens de programação para a web. A internet tornou-se, ao longo dos anos, numa grande potência mundial, pelo que ao estudar estas linguagens de programação, o autor está a adquirir capacidades para entrar neste mundo de oportunidades e negócios.
3. A enorme quantidade de informação actualizada que existe sobre a linguagem. Sendo uma linguagem bastante conhecida e utilizada, PHP está constantemente a ser discutido e desenvolvido. Consequentemente, conseguimos encontrar bastante informação em livros, artigos e em documentação oficial em como trabalhar e utilizar a linguagem efetivamente.

### Javascript

Javascript é considerada uma das três linguagens essenciais para a programação web (sendo as outras duas HTML e CSS). Javascript é a principal linguagem responsável pelo comportamento das páginas web, sendo capaz de tornar as páginas web dinamicas e interativas para o utilizador.

Tendo em conta que parte dos objetivos desta aplicação é a eficiência do interface e a rapidez de resposta por parte da aplicação (Pontos 12 e 13 da secção 1.2), conseguimos concluir que Javascript é uma ferramenta ideal para desenvolver esta aplicação. Adicionalmente, sendo uma das linguagens mais conhecidas para programação, além de continuar em constante desenvolvimento, também se consegue encontrar muita informação, exemplos, boas práticas e tutoriais sobre ela.

### AJAX

AJAX (Asynchronous JavaScript and XML) não é uma ferramenta de programação, mas sim uma técnica, que utiliza as tecnologias Javascript e XML, para criar páginas web dinâmicas.

Ao contrário das páginas web clássicas, AJAX permite páginas web serem carregadas assincronamente, trocando pequenas porções de dados entre o servidor, sem que a página seja completamente recarregada (7). Deste modo, consegue-se melhorar a interação do utilizador com a aplicação ao trocar só a informação ggnecessária entre o servidor e o cliente.

Quando um evento especificado pelo programador ocorre, é criado um objeto XMLHttpRequest que é enviado para o servidor. Seguidamente, o servidor processa o pedido e envia os dados processados de volta para o navegador. Finalmente, estes dados são processados por Javascript, que atualiza o conteúdo da página web.

AJAX torna-se numa solução para aumentar a performance e resposta entre o cliente e o servidor, enviado só os dados necessários de modo a atualizar os conteúdos essenciais da página. Tendo em conta estes pontos, a interação do cliente com a aplicação será toda programada em AJAX.

### MySQL

O MySQL é um sistema de gestão de base de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo (8).

Para além do seu grande desempenho e estabilidade, MySQL tem grande compatibilidade com PHP, tendo este um módulo de interface próprio. Este módulo de interface, contendo funções preparadas para lidar com conexões de base de dados e manipulação de dados, mantêm-se atualizado e testado até hoje. Não só isto garante um nível de segurança mais alto comparando a outros SGBDs, mas também maior eficácia a programar e maior acesso e quantidade de documentação disponível.

Todos os pontos anteriormente referidos, levaram ao autor escolher MySQL em relação às outras SGBDs disponíveis.

## Diagrama de hieraquia

O seguinte diagrama apresenta a hieraquia dos vários interfaces de cada página. Ao iniciar a aplicação, o utilizador terá acesso à página inicial. A partir desta, o utilizador pode entrar na página de login que lhe dará acesso, depois do login efectuado corretamente, a uma rede de páginas que lhe dará uma série de funcionalidades e interfaces para manipular e visualizar o mapa de boleias.



## Desafios

### Visualização dos registos de boleias num único mapa de boleias

Problema:

Existem registos de boleias com data, hora inicial, hora final e outros dados associados a elas. A aplicação tem de conseguir ler os registos das boleias para um determinado período de tempo (neste caso, vamos considerar uma semana) e colocá-los de forma organizada, de modo a que o utilizador consiga, facilmente, identificar as boleias existentes. Os pontos mais exigentes neste desafio são:

* A criação de um mapa intuitivo utilizando elementos e estrutura HTML
* Posição e tamanho da boleia no mapa consoante a sua data, hora inicial e hora final.
* A adaptação do mapa a boleias sobrepostas (boleias que partilhem o mesmo período de tempo).

Solução:

Em resposta ao primeiro ponto do problema, o autor decidiu fazer um mapa de boleias com o aspeto visual de um horário semanal, ou seja, uma tabela em que as colunas indicam o dia da semana (segunda a sexta) e as linhas indicam as horas em intervalos de meia hora (ex: 8:00-8:30). Este aspeto, sendo bastante comum e prático, torna a visualização do mapa rápida e intuitiva.

Para calcular a posição de cada boleia na estrutura HTML, temos de ter em conta que o HTML lê as tabelas linha a linha, ou seja, ele lê cada bloco da tabela (vamos chamar de bloco à junção de cada coluna com cada linha da tabela) da esquerda para a direita e de cima para baixo. Isto é um ponto importante porque, como o autor decidiu colocar as horas em cada linha, significa que temos de averiguar a cada hora do horário se existem boleias à medida que a tabela é lida.

Tendo isto em conta, para solucionar o segundo ponto do problema, a aplicação vai fazer uma query a cada bloco que ainda não esteja preenchido para averiguar se existem boleias com a hora inicial e dia da semana correspondentes a esse bloco. No caso de existir, a aplicação vai calcular o tamanho do bloco necessário para corresponder a boleia à sua hora final.

No entanto, a solução anterior não consegue responder a boleias que partilhem o mesmo período de tempo porque, devido à estrutura HTML, não se consegue nem dividir um bloco de uma tabela HTML nem alinhar boleias sobrepostas no mesmo bloco sem elementos adicionais.

Para solucionar este problema, em referência ao terceiro ponto mencionado na descrição do problema anteriormente, o autor decidiu fazer uma query adicional a cada bloco da tabela para descobrir se existem elementos adicionais. Se esta query não devolver resultados, o bloco é preenchido normalmente sem elementos adicionados. Se devolver, será criada uma tabela adicional dentro desse bloco. Esta tabela adicional terá uma coluna para cada boleia nesse período de tempo, em que cada boleia sobreposta será posicionada nessa coluna consoante a sua diferença de tempo em relação à hora inicial da boleia encontrada na primeira query.

### Interface e interação eficiente da aplicação

Problema:

Um dos objetivos do trabalho proposto é desenvolver um interface para a aplicação que seja o mais eficiente possível, de modo a que o utilizador consiga fazer um grande número de ações num curto espaço de tempo e sem esforço. Isto significa que o interface tem de ser o mais simples e automatizado possível.

Solução:

O interface do mapa de boleias é o ponto focal deste desafio, já que é onde o utilizador vai passar a maioria do tempo dentro da aplicação. Tendo em conta este ponto, o autor decidiu colocar os casos de uso mais relevantes (inserir boleia, inserir passageiro, etc...) neste interface.

De modo a tornar o processo de inserir boleias rápido e fácil, o autor decidiu tornar cada bloco do mapa de boleias não preenchido num botão. Este botão permite ao utilizador colocar uma boleia, nesse bloco disponível, com todos o seus dados automaticamente. Para o caso de todos os blocos estarem preenchidos nesse dia, foi colocado um botão ao lado dos dias da semana, onde o utilizador simplesmente insere a hora da boleia.

Seguindo estas ideias, foi feito o mesmo para os blocos preenchidos, tornando estes blocos igualmente em botões. No caso de o utilizador clicar num bloco preenchido cuja boleia não é sua, simplesmente apareçe uma caixa com as informações da boleia e com um botão que permite entrar ou sair da boleia caso seja passageiro ou não. No caso de ser condutor, apareçe uma caixa semelhante mas com elementos de texto caso o utilizador queira alterar os dados da boleia, reptir a boleia ou eliminá-la.

Todas estas decisões fazem com todas as funcionalidades essenciais se encontrem no mapa de boleias, de forma intuitiva e eficaz.

## Avaliação da aplicação

A quarta e última fase da programação extrema é a fase de testes. Nesta secção, iremos falar dos testes que foram feitos à aplicação desde de o início do seu desenvolvimento até ao momento da sua apresentação.

### Testes

Durante a sua programação, a aplicação foi testada continuamente para garantir a sua integridade e segurança. Para manter a integridade da aplicação, o autor testou a interação entre os vários casos de uso e a validação dos dados colocados pelo utilizador, de acordo com as descrições de caso de uso que se encontram no Anexo A.

Para melhorar a segurança da aplicação, o autor seguiu boas práticas e métodos de programação que aprendeu não só da disciplina opcional Programação e Segurança que teve durante o período de licenciatura, mas também do livro sobre segurança em PHP que leu durante o desenvolvimento do projeto chamado Pro PHP Security (9). Os pontos focais destas boas práticas foram ataques de terceiros incluindo SQL Injection e Cross-Site Scripting.

### Avaliação do produto desenvolvido

Para uma melhor avalição da aplicação, foi utilizado uma aplicação web desenvolvida pela Google chamada PageSpeed Insights. Esta aplicação avalia a performance da aplicação quer em telemóvel quer em computador.

Para computador, a aplicação foi avaliada com a pontuação máxima contendo só alguns avisos (Anexo X).

# Conclusão

# Bibliografia

1. Extreme Programing. *USFCS.* [Online] http://www.cs.usfca.edu/~parrt/course/601/lectures/xp.html.

2. Regional ITS Architecture Guidance Document. *FHWA Operations.* [Online] http://ops.fhwa.dot.gov/publications/regitsarchguide/5defineint.htm.

3. Wikipedia - Use Case Diagram. *Wikipedia.* [Online] https://en.wikipedia.org/wiki/Use\_Case\_Diagram.

4. Mozilla Developer Network. *HTML | MDN.* [Online] https://developer.mozilla.org/en-US/docs/Web/HTML.

5. W3 - CSS. *W3.* [Online] http://www.w3.org/TR/2011/REC-CSS2-20110607/about.html.

6. PHP: Prefácio - Manual. *PHP.* [Online] http://php.net/manual/pt\_BR/preface.php.

7. W3Schools - AJAX. *W3Schools.* [Online] http://www.w3schools.com/ajax/ajax\_intro.asp.

8. MySQL::Why MySQL? *MySQL.* [Online] http://www.mysql.com/why-mysql/.

9. **Snyder, Chris, Myer, Thomas e Southwell, Michael.** *Pro PHP Security.* New York : Springer Science+Business Media, LLC., 2010.

10. **DUCKETT, JON.** *HTML & CSS.* Indiapolis : John Wiley & Sons, Inc, 2011.

11. Bootstrap. *Bootstrap.* [Online] http://getbootstrap.com.

12. *Stack Overflow.* [Online] http://stackoverflow.com/.

13. *W3Schools Online Web Tutorials.* [Online] http://www.w3schools.com/.

14. Biblioteca Digital do IPG. *Biblioteca Digital do IPG.* [Online] http://bdigital.ipg.pt/dspace/.

15. Calculating color contrast. *24Ways.* [Online] http://24ways.org/2010/calculating-color-contrast/.

16. Moqups. *Moqups.* [Online] moqups.com.

17. Palleton - The Color Scheme Designer. *Palleton - The Color Scheme Designer.* [Online] http://paletton.com/.

18. Usage | ColorSchemeDesigner. *ColorSchemeDesigner.* [Online] http://www.colorschemedesigner.com/blog/usage/.

# Anexos

* Anexo A

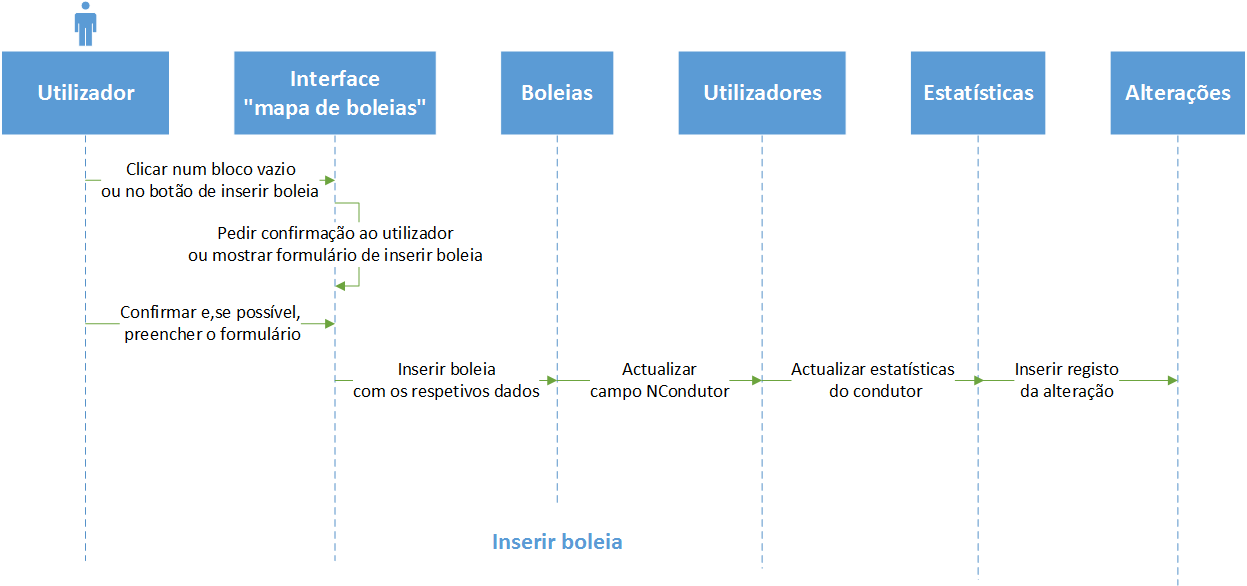
|  |  |
| --- | --- |
| **Nome** | Inserir boleia |
| **Objetivo** | O ator insere uma boleia |
| **Prioridade** | Alta |
| **Pré-condição** | Login válido |
| **Cenário principal** | 1. O Caso de Uso começa quando o ator clica no botão “Inserir Boleia” ou quando selecciona um espaço vazio no mapa de boleias 2. O sistema apresenta o formulário “Inserir boleia” 3. O ator preenche os campos obrigatórios. 4. O ator clica no botão “Ok”, confirmando os dados 5. O sistema regista os dados |
| **Cenário alternativo** | O ator pode cancelar a operação a qualquer momento.  1. a. No caso de seleccionar um espaço vazio no mapa, a boleia é colocada automaticamente sem a necessidade de formulário. A boleia será inserida com os valores pré-configurados pelo utilizador.  4. a. O ator não preenche todos os campos obrigatórios e aparece uma mensagem de erro.  4. b. Se a sintaxe de algum campo estiver incorreta, mostra mensagem de erro. |
| **Pós-condição** | Os dados estatísticos relevantes ao ator são atualizados. |
| **Casos de teste** | * Verificar se o mapa de boleias é actualizado corretamente. * Verificar se as estatísticas são atualizadas corretamente. * Verificar se ao omitir campos obrigatórios, o sistema dá erro. * Verificar se os campos são preenchidos correctamente * Se os campos numéricos só contêm caraters núméricos. * Se os campos alfabéticos só contêm caraters alfabéticos. |

|  |  |
| --- | --- |
| **Nome** | Alterar boleia |
| **Objetivo** | O ator altera uma boleia |
| **Prioridade** | Alta |
| **Pré-condição** | Login válido |
| **Cenário principal** | 1. O Caso de Uso começa quando o ator clica no botão “Alterar” após seleccionar uma boleia. 2. O sistema apresenta o formulário “Alterar boleia” 3. O ator preenche os campos obrigatórios. 4. O ator clica no botão “Ok”, confirmando os dados 5. O sistema regista os dados |
| **Cenário alternativo** | O ator pode cancelar a operação a qualquer momento.   1. a. O ator não preenche todos os campos obrigatórios e aparece uma mensagem de erro.   4. b. Se a sintaxe de algum campo estiver incorreta, mostra mensagem de erro.  4 c. Se a boleia faz parte de uma repetição, o sistema apresenta uma mensagem para alterar todas as boleias da repetição. |
| **Pós-condição** | É enviado um email para todos os passageiros da boleia a informar da alteração efetuada. |
| **Casos de teste** | * Verificar se o mapa de boleias é actualizado corretamente. * Verificar se as estatísticas são atualizadas corretamente. * Verificar se os campos são preenchidos correctamente * Se os campos numéricos só contêm caraters núméricos. * Se os campos alfabéticos só contêm caraters alfabéticos. |

|  |  |
| --- | --- |
| **Nome** | Entrar numa boleia |
| **Objetivo** | O ator entra numa boleia |
| **Prioridade** | Alta |
| **Pré-condição** | Login válido |
| **Cenário principal** | 1. O Caso de Uso começa quando o ator clica no botão “Entrar” após seleccionar uma boleia. 2. O sistema apresenta o formulário “Inserir passageiro” 3. O ator preenche os campos obrigatórios. 4. O ator clica no botão “Ok”, confirmando os dados 5. O sistema regista os dados |
| **Cenário alternativo** | O ator pode cancelar a operação a qualquer momento.  4. a. O ator não preenche todos os campos obrigatórios e aparece uma mensagem de erro. |
| **Pós-condição** | Os dados estatísticos relevantes ao ator são atualizados. |
| **Casos de teste** | * Verificar se o mapa de boleias é actualizado corretamente. * Verificar se as estatísticas são atualizadas corretamente. |

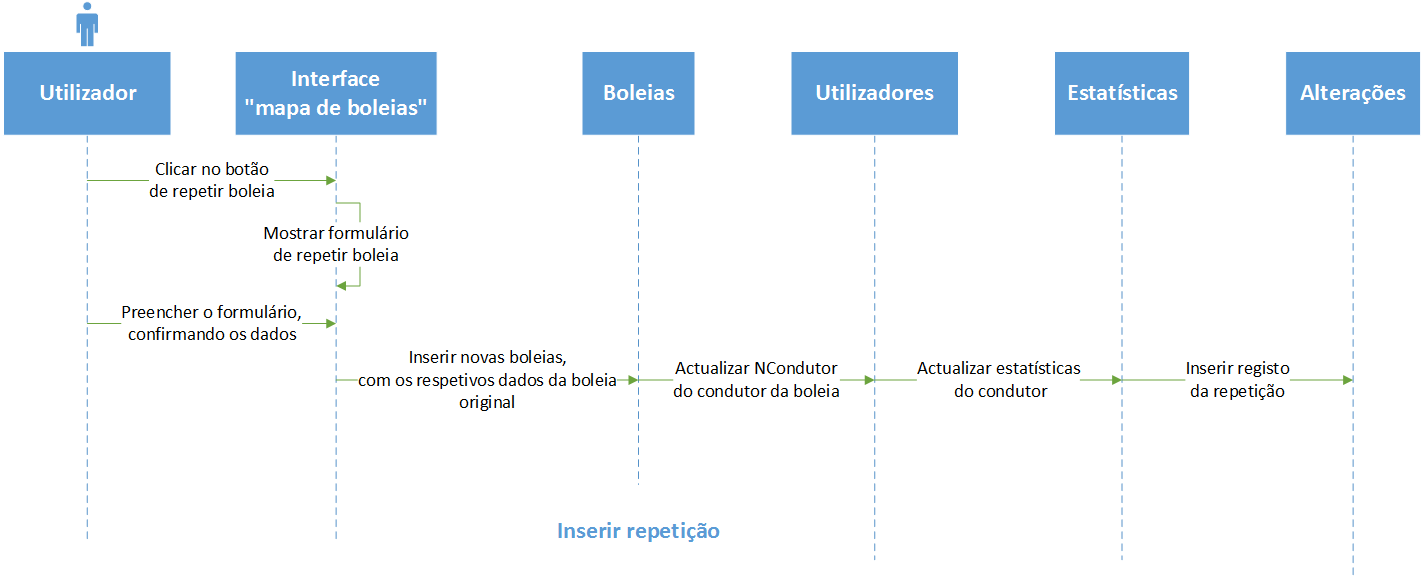
|  |  |
| --- | --- |
| **Nome** | Repetir boleia |
| **Objetivo** | O ator repete uma boleia |
| **Prioridade** | Alta |
| **Pré-condição** | Login válido |
| **Cenário principal** | 1. O Caso de Uso começa quando o ator clica no botão “Repetir” após seleccionar uma boleia. 2. O sistema apresenta o formulário “Repetir boleia” 3. O ator preenche os campos obrigatórios. 4. O ator clica no botão “Ok”, confirmando os dados 5. O sistema insere novas boleias consoante os campos preenchidos |
| **Cenário alternativo** | O ator pode cancelar a operação a qualquer momento.  4. a. O ator não preenche os campos corretamente e aparece mensagem de erro. |
| **Pós-condição** | Os dados estatísticos relevantes ao condutor e passageiros (se existirem) da boleia repetida são atualizados. |
| **Casos de teste** | * Verificar se o mapa de boleias é actualizado corretamente. * Verificar se as estatísticas são atualizadas corretamente. |

* Anexo B









1. Dados referentes aos preços médios da gasolina super com chumbo vendida em Portugal entre 1960 e

   2014 http://www.pordata.pt/Portugal/Pre%C3%A7os+m%C3%A9dios+de+venda+ao+p%C3%BAblico+dos+combust%C3%ADveis+l%C3%ADquidos+e+gasosos+%E2%80%93+Continente-1265 [↑](#footnote-ref-1)