



Projeto de Informática

2014/2015

Car Pooling

Orientador:

Prof. José Fonseca

Realizado por:

1010834 – Vasco Fortuna

Resumo

Este relatório consiste no processo de desenvolvimento de uma aplicação informática para gestão de boleias entre um grupo de amigos, a pedido de um grupo de docentes do IPG.

Apesar de já existirem aplicações de boleias no mercado, elas somente satisfazem as necessidades individuais do utilizador, não satisfazendo as necessidades de organização que um grupo tem.

A aplicação focalizará num único mapa de boleias que será visualizado pelo grupo inteiro. Cada membro poderá fazer as alterações necessárias com as funcionalidades que a aplicação fornecerá. Estas funcionalidades incluirão gestão básica dos membros por parte de um administrador e funcionalidades específicas para a organização das boleias, tais como repetição de boleias para um período de tempo e escolha automática do condutor.

O interface da aplicação estará o mais simplificado e automatizado possível para facilitar o seu uso e diminuir o tempo que cada utilizador necessita para utilizar a aplicação.

A aplicação também notificará, através de email, os passageiros de uma boleia, caso essa boleia seja alterada.

Palavras-chave:

Programação web, MySQL, PHP, AJAX, Carpooling.

Abstract

This report consists on the process of development of a web app for managing carpools by a group of friends, by request of teachers of IPG.

Even though there are carpooling apps in the market, they only satisfy the individual needs of the user, not satisfying the needs within a group of users.

The application will focus on a single carpooling map that will be visualized by the whole group. Each member will be able to do any necessary alteration, to the map, with the tools given by the app. These tools will include basic management of the members by an administrator and specific functionalities for the organization of carpools, for example, repetition of carpools for a given period of time and automatic choice of driver.

The interface of the application will be as simplified and automated as possible, so its access can be facilitated and the time that each user spends on the app can be diminished.

The app will also notify, by email, the users of a carpool in case that carpool is altered by another member.

Key-words:

Web programming, MySQL, PHP, AJAX, Carpooling.

Conteúdo

1.	Introdução	9
2.	Estado de arte	12
2.1.	Análise de aplicações de boleias correntes	12
2.1.1.	Blablacar.....	13
2.1.2.	Boleia.net.....	14
2.1.3.	Pendura.pt.....	15
3.	Metodologia e Análise de requisitos	17
3.1.	Metodologia	17
3.2.	Planeamento.....	18
3.3.	Diagrama de contexto	19
3.4.	Diagrama de casos de uso.....	20
3.5.	Descrições de casos de uso.....	23
3.5.1.	Inserir boleia	23
3.5.2.	Eliminar boleia	25
3.6.	Diagramas de sequência	26
3.6.1.	Inserir boleia	26
3.6.2.	Eliminar boleia	27
3.7.	Diagrama de classes e modelo EER.....	28
3.8.	Semântica.....	31
3.8.1.	Utilizadores.....	31
3.8.2.	Boleias.....	34
3.8.3.	Passageiros.....	36
3.8.4.	Estatísticas	37
3.8.5.	Configurações	38
3.8.6.	Alterações.....	40
4.	Desenvolvimento	41
4.1.	Tecnologias utilizadas.....	41
4.1.1.	HTML	41

4.1.2.	CSS	41
4.1.3.	PHP.....	42
4.1.4.	Javascript	42
4.1.5.	AJAX	43
4.1.6.	MySQL.....	43
4.1.7.	GitHub.....	44
4.2.	Diagrama de hierarquia	45
4.3.	Análise e implementação.....	46
4.3.1.	Visualização dos registos de boleias num único mapa de boleias	46
4.3.2.	Interface e interação eficiente da aplicação	49
4.4.	Avaliação da aplicação.....	51
4.4.1.	Testes.....	51
5.	Conclusão.....	52
6.	Bibliografia	53
7.	Anexos.....	55

Lista de Figuras

Figura 1 - Mapa de Gantt efetivo	18
Figura 2 - Diagrama de Contexto.....	19
Figura 3 - Diagrama de Casos de Uso	22
Figura 4- Diagrama de sequência "Inserir boleia"	26
Figura 5 - Diagrama de sequência "Eliminar boleia"	27
Figura 6 - Diagrama de classes	28
Figura 7 - Modelo EER criado em MySQL Workbench	29
Figura 8 - Diagrama de Hiarquia	45
Figura 9 - Interface da aplicação carpooling	47
Figura 10 - Ordem de construção de tabelas pelo HTML.....	47
Figura 11 - Fluxograma do algoritmo de leitura de boleias	48
Figura 12 - Interface do mapa de boleias	50
Figura 13 - Diagrama de sequência do caso de uso "Alterar boleia"	59
Figura 14 - Diagrama de sequência do caso de uso "Entrar numa boleia"	60
Figura 15 - Diagrama de sequência do caso de uso "Repetir boleia"	60
Figura 16 - Diagrama de sequência do caso de uso "Sair de uma boleia"	60
Figura 17 - Avaliação do PageSpeed da aplicação para telemóvel (Parte 1)	61
Figura 18 - Avaliação do PageSpeed da aplicação para telemóvel (Parte 2)	62
Figura 19 - Avaliação do PageSpeed da aplicação para computador	63

Lista de Tabelas

Tabela 1 – Funcionalidades de blablacar	13
Tabela 2 - Funcionalidades de boleia.net.....	14
Tabela 3 - Funcionalidades de pendura.pt.....	15
Tabela 4 - Comparação de todas as aplicações estudadas.....	16
Tabela 5 - Descrição do caso de uso "Inserir Boleia"	24
Tabela 6 – Descrição do caso de uso “Eliminar Boleia”	25
Tabela 7 - Dicionário de dados da tabela Utilizadores	32
Tabela 8 - Operações da tabela Utilizadores	33
Tabela 9 - Dicionário de dados da tabela Boleias	34
Tabela 10 - Operações da tabela Boleias	35
Tabela 11 - Dicionário de dados da tabela Passageiros	36
Tabela 12 - Operações da tabela Passageiros	36
Tabela 13 - Dicionário de dados da tabela Estatísticas.....	37
Tabela 14 - Operações da tabela Estatísticas.....	37
Tabela 15 - Dicionário de dados das configurações.....	38
Tabela 16 - Operações da tabela Configurações.....	39
Tabela 17 - Dicionário de dados da tabela Alterações.....	40
Tabela 18 - Operações da tabela Alterações.....	40
Tabela 19 - Descrição do caso de uso "Alterar boleia"	55
Tabela 20 - Descrição do caso de uso "Entrar numa boleia"	56
Tabela 21 - Descrição do caso de uso "Repetir boleia"	57
Tabela 22 - Descrição do caso de uso "Sair de uma boleia"	58

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

BD – Base de Dados

EER – Enhanced Entity-Relationship

1. Introdução

O presente relatório caracteriza o desenvolvimento do projeto, chamado de carpooling, 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.

1.1.Contexto do projeto

O crescente aumento dos custos de combustíveis (uma subida de preço que ocorreu desde 0,02€/litro até 1,68€/litro¹) 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 customizar 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

¹Dados 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+e+gasosos+E2%80%93+Continente-1265>

CAPÍTULO 1.INTRODUÇÃO

mapa de boleias. Esta vista consistirá 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.

1.2.Objetivos da aplicação

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

1. Registrar, alterar e eliminar utilizadores. Esta é uma funcionalidade básica para permitir a gestão dos utilizadores.
2. Organizar pedidos e ofertas de boleias dentro do grupo. A aplicação tem de ter uma organização visual intuitiva ao utilizador, de modo a ele conseguir localizar as boleias no tempo facilmente.
3. Contabilizar boleias efetuadas 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.
4. Duplicar mapas de boleias para semestres, anos. Conseguir copiar uma porção do mapa de boleia e colocá-lo noutra período de tempo.
5. Preencher mapas automaticamente de acordo com as boleias contabilizadas de cada membro. A aplicação deve conseguir criar boleias novas automaticamente de acordo com a relação de número de vezes que cada utilizador foi passageiro/condutor.
6. Enviar notificações por email acerca das próximas boleias ou alterações de boleias que afetam o utilizador.
7. Calcular a redução da pegada de carbono. Este objetivo foi pedido, pelo orientador do autor, para os utilizadores terem uma noção do impacto das suas ações na luta contra o aquecimento global.
8. 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.
9. Criar cópias de segurança dos mapas de boleias.

CAPÍTULO 1.INTRODUÇÃO

10. Registrar as alterações mais relevantes à base de dados. Manter um registo da atividade dos utilizadores é importante para melhorar a interatividade entre os utilizadores, a segurança de ações menos desejadas e a integridade do mapa de boleias.
11. Interface eficiente e intuitiva. Nesta aplicação, é essencial fazer um grande número de ações num curto espaço de tempo de modo a que a aplicação se torne o mais útil possível.

1.3.Estrutura do relatório

Este relatório é constituído por 5 capítulos. No primeiro capítulo, é apresentado o problema do projeto e é feita uma breve introdução ao relatório. No segundo capítulo, é apresentado o estado de arte, contendo um estudo às aplicações existentes que se enquadram no tema do projeto. No terceiro capítulo é descrita a metodologia e a análise de requisitos. No quarto capítulo, as tecnologias utilizadas são descritas tal como alguns desafios que foram encontrados durante o desenvolvimento e a avaliação da aplicação. No quinto capítulo é mencionada a conclusão deste projeto e futuro desenvolvimento da aplicação.

2. Estado de arte

Este capítulo é dedicado 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.

2.1. Análise de aplicações de boleias correntes

A análise das aplicações teve como pontos principais a interação dos vários utilizadores entre si através da própria aplicação e as funcionalidades disponíveis para os utilizadores. Foi estudado e observado, em cada aplicação, se era possível haver interação entre grupos de utilizadores e se existiam funcionalidades que permitiam a criação e manipulação de boleias partilhadas entre vários utilizadores ao mesmo tempo.

As aplicações web estudadas 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.

2.1.1. 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, previamente 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
Responsivo		X
Gratuito	X	
Open-source		X
Registo de utilizadores	X	
Duplicação de ofertas a longo prazo		X
Editar ofertas depois de anunciadas		X
Criação de mapas de boleias		X
Duplicação de pedidos		X
Reserva de viagens a longo prazo		X
Aplicação móvel	X	

Tabela 1 – Funcionalidades de blablacar

2.1.2. 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 intervenientes. 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
Responsivo	X	
Gratuito	X	
Open-source		X
Registo de utilizadores	X	
Duplicação de ofertas a longo prazo		X
Editar ofertas depois de anunciadas		X
Criação de mapas de boleias		X
Duplicação de pedidos		X
Reserva de viagens a longo prazo		X
Aplicação móvel		X

Tabela 2 - Funcionalidades de boleia.net

2.1.3. 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
Responsivo	X	
Gratuito	X	
Open-source		X
Registo de utilizadores		X
Duplicação de ofertas a longo prazo		X
Editar ofertas depois de anunciadas		X
Criação de mapas de boleias		X
Duplicação de pedidos		X
Reserva de viagens a longo prazo		X
Aplicação móvel		X

Tabela 3 - Funcionalidades de pendura.pt

2.2. 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	Boleianet	Pendura.pt	Carpooling
Responsivo	S	S	N	S
Gratuito	S	S	S	S
Open-source	N	N	N	S
Registo de utilizadores	S	S	N	S
Duplicação de ofertas a longo prazo	N	N	N	S
Editar ofertas depois de anunciadas	N	N	N	S
Criação de mapas de boleias	N	N	N	S
Duplicação de pedidos	N	N	N	S
Reserva de viagens a longo prazo	N	N	N	S
Aplicação móvel	N	N	S	N

Tabela 4 - Comparação de todas as aplicações estudadas (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 existirem 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.

3. Metodologia e Análise de requisitos

Neste capítulo é descrito a metodologia selecionada para o desenvolvimento da aplicação e a análise de requisitos do projeto.

3.1.Metodologia

Durante todo o processo do projeto, será utilizada uma variante do desenvolvimento ágil: a programação extrema (XP). 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. O desenvolvimento XP caracteriza-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). Durante o desenvolvimento da aplicação, sempre que possível, o autor reuniu-se com o orientador semanalmente para rever o progresso da aplicação até aquele momento.

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 metodologia procura adicionar um conjunto de funcionalidades ao produto final e cada iteração contém as quatro fases de desenvolvimento:

- Na fase de planeamento, discutiu-se e documentou-se 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.

3.2.Planeamento

Durante a fase de planeamento, fez-se um plano da sequência de atividades a seguir durante o processo de desenvolvimento do projeto. No mapa de Gantt seguinte (Figura 1 - Mapa de Gantt efetivo, estão descritas as atividades planeadas e efetuadas. É de notar que o relatório foi sendo concluído ao longo do processo de desenvolvimento todo e que a fase de testes intercala-se com a fase de codificação porque estavam a ser corrigidos os erros que se encontravam nos testes.

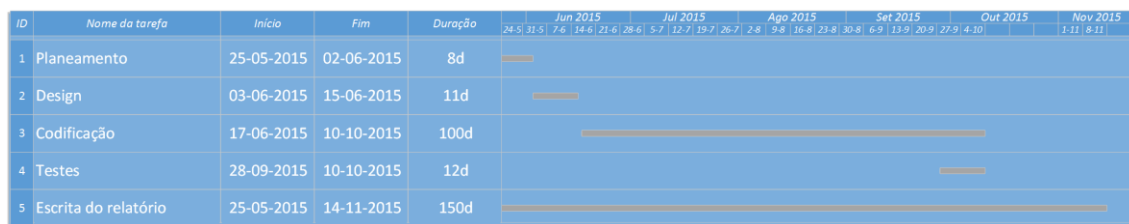


Figura 1 - Mapa de Gantt efetivo

3.3.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 interação de atores ou sistemas externos com a nossa aplicação. Neste caso, as únicas interações externas que temos são os utilizadores comuns e o administrador.

As ações dos utilizadores comuns estarão focalizadas na interatividade com as boleias, como é demonstrado no diagrama abaixo (Figura 2 - Diagrama de Contexto. Cada utilizador vai ter acesso ao mapa de boleias e consegue utilizar a aplicação para inserir, alterar, eliminar e repetir as suas boleias no mapa de boleias. Cada utilizador também consegue entrar nas boleias de outros utilizadores como passageiros, bem como sair dessas boleias. Os utilizadores também vão ter a possibilidade de duplicar o mapa de boleias e sincronizar a base de dados com o google spreadsheet como foi mencionado anteriormente nos pontos 1.2.4 e 1.2.8, respetivamente.

O administrador, além das permissões do utilizador comum, poderá adicionar, alterar e eliminar utilizadores.

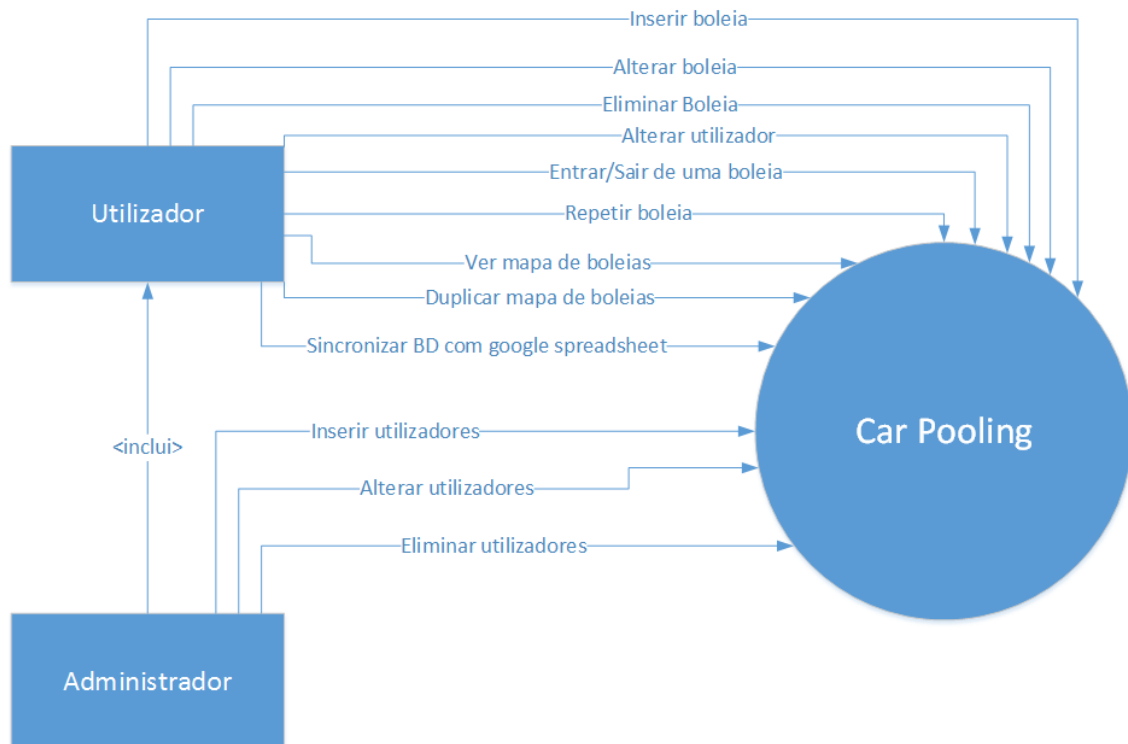
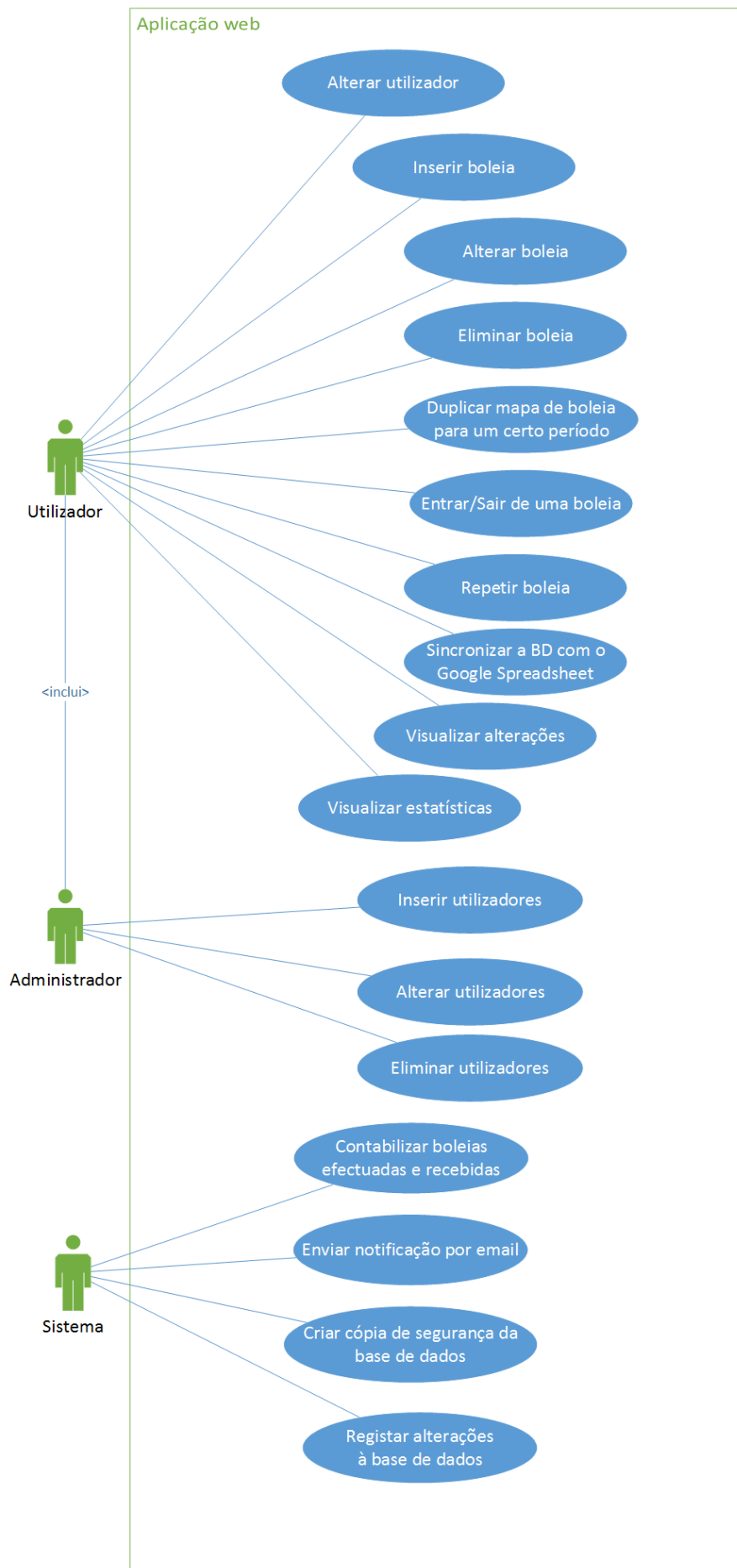


Figura 2 - Diagrama de Contexto

3.4. Diagrama de casos de uso

O diagrama de casos 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. O diagrama de casos de uso para este projeto (Figura 2) tem em conta as interações descritas no diagrama de contexto anterior, mas também indica os casos de uso da própria aplicação.

O sistema irá contabilizar as boleias efetuadas e recebidas de cada utilizador, ou seja, sempre que ocorre uma entrada ou saída de um condutor ou passageiro de uma boleia, serão calculadas estatísticas para os utilizadores envolventes (de acordo com o ponto 1.2.3 mencionado anteriormente). O sistema também enviará emails, sempre que haja uma alteração de uma boleia, aos passageiros envolventes (ponto 1.2.6), criará cópias de segurança da base de dados regularmente (ponto 1.2.9) e registará as alterações dos utilizadores à base de dados (ponto 1.2.10).



CAPÍTULO 3.METODOLOGIA E ANÁLISE DE REQUISITOS

Figura 3 - Diagrama de Casos de Uso

3.5. Descrições 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. As descrições apresentam detalhadamente passos sequenciais de como os casos de uso irão funcionar. Também apresentam cenários alternativos ao caso de uso e testes necessários para garantir a integridade do caso de uso. Nesta secção são apresentadas as descrições dos casos de uso mais relevantes e essenciais ao projeto, que são aquelas ligadas à gestão das boleias e dos utilizadores. Outras descrições menos relevantes podem ser encontradas no Anexo A.

3.5.1. Inserir boeia

Este caso de uso não é particularmente interessante, mas é essencial ao bom funcionamento da aplicação. O cenário principal deste caso de uso requer o ator seleccionar o botão “Inserir Boleia” e preencher um formulário com os dados da boeia desejada.

O que faz este caso de uso diferente do habitual é a existência de um cenário alternativo onde o ator não precisa preencher formulário. Ao clicar num botão alternativo, localizado nos espaços vazios do mapa de boleias, o sistema coloca automaticamente uma boeia nesse espaço com as pré-configurações do ator (as configurações do ator são preenchidas durante o registo do utilizador na base de dados). Este processo todo está descrito na tabela abaixo (Tabela 5).

Nome	Inserir boleia
Objetivo	O ator insere uma boleia
Prioridade	Alta
Pré-condição	Login válido
Cenário principal	<ol style="list-style-type: none"> 1. O Caso de Uso começa quando o ator clica no botão “Inserir Boleia” ou quando seleciona 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	<p>O ator pode cancelar a operação a qualquer momento.</p> <ol style="list-style-type: none"> 1. a. No caso de selecionar 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	<ul style="list-style-type: none"> • Verificar se o mapa de boleias é atualizado corretamente. • Verificar se as estatísticas são atualizadas corretamente. • Verificar se ao omitir campos obrigatórios, o sistema dá erro. • Verificar se os campos são preenchidos corretamente <ul style="list-style-type: none"> ○ Se os campos numéricos só contêm caracteres numéricos. ○ Se os campos alfabéticos só contêm caracteres alfabéticos.

Tabela 5 - Descrição do caso de uso "Inserir Boleia"

3.5.2. Eliminar boleia

O que devemos notar deste caso de uso é o cenário alternativo. Se uma boleia fizer parte de uma repetição, o sistema faz uma pergunta adicional ao ator para eliminar todas as boleias ligadas a essa repetição.

Nome	Eliminar boleia
Objetivo	O ator elimina uma boleia
Prioridade	Baixa
Pré-condição	Login válido
Cenário principal	<ol style="list-style-type: none"> 1. O Caso de Uso começa quando o ator clica no botão “Eliminar” após selecionar uma boleia. 2. O sistema pede a confirmação da eliminação. 3. O ator clica no botão “Ok”, confirmando a eliminação. 4. O sistema elimina a boleia.
Cenário alternativo	<p>O ator pode cancelar a operação a qualquer momento.</p> <p>1.a Se a boleia pertencer a uma repetição, o sistema pergunta ao ator se ele quer eliminar as boleias associadas a essa repetição. Se ele confirmar, então no passo 4, são eliminadas todas as boleias ligadas a essa repetição.</p>
Pós-condição	Os dados estatísticos relevantes ao condutor e passageiros (se existirem) da boleia repetida são atualizados.
Casos de teste	<ul style="list-style-type: none"> • Verificar se o mapa de boleias é atualizado corretamente. • Verificar se as estatísticas são atualizadas corretamente.

Tabela 6 – Descrição do caso de uso “Eliminar Boleia”

3.6. Diagramas de sequência

Os diagramas de sequência mostram a interação entre o utilizador, a aplicação e as tabelas da base de dados por passos sequenciais. Os diagramas de sequência baseiam-se nas descrições de caso de uso e estabelecem a ponte entre as descrições e a base de dados. Cada seta representa uma acção entre os dois intervenientes ligados pela seta. Aqui são apresentados os diagramas de sequência mais relevantes e/ou interessantes ao projeto. Os diagramas de sequência menos relevantes encontram-se no Anexo B deste relatório.

3.6.1. Inserir boleia

Para complementar a descrição de caso de uso anterior, é apresentado aqui o respetivo diagrama de sequência (Figura 4). Apresenta o cenário principal e o cenário alternativo do caso de uso, onde o utilizador seleciona um espaço vazio para inserir uma boleia.

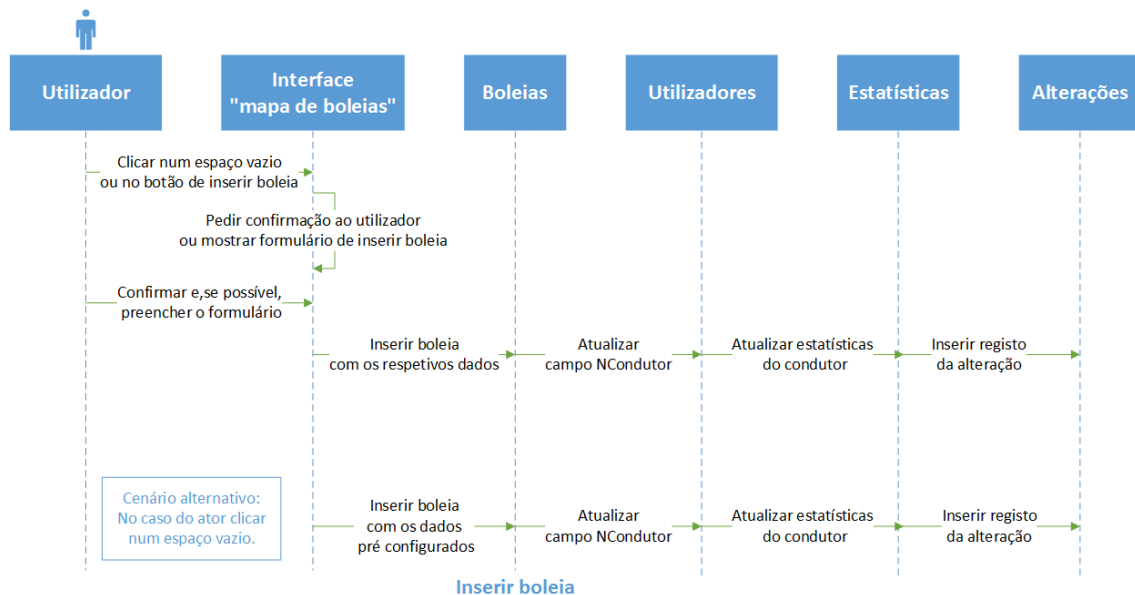


Figura 4- Diagrama de sequência "Inserir boleia"

3.6.2. Eliminar boleia

Para complementar a descrição de caso de uso, é apresentado o diagrama seguinte (Figura 5).

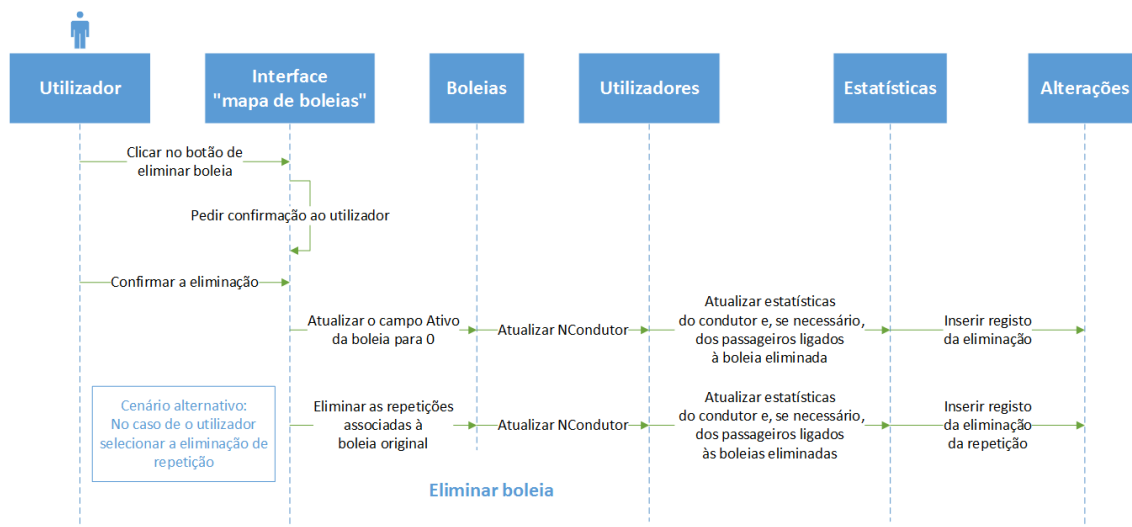


Figura 5 - Diagrama de sequência "Eliminar boleia"

Neste diagrama, é de notar o cenário alternativo quando o utilizador também seleciona a eliminação da repetição. Neste caso, todas as boleias associadas a esta repetição são eliminadas.

3.7. Diagrama de classes e modelo EER

Aqui é apresentado o diagrama de classes e o modelo EER (Enhanced Entity-Relationship model) desenvolvidos (Figura 6 e Figura 7, respetivamente), consoante os objetivos mencionados anteriormente. O diagrama de classes é constituído por seis classes, mas as principais são as dos utilizadores, passageiros e boleias. Estas irão guardar os registos dos utilizadores e das boleias, estabelecendo a interação entre elas.

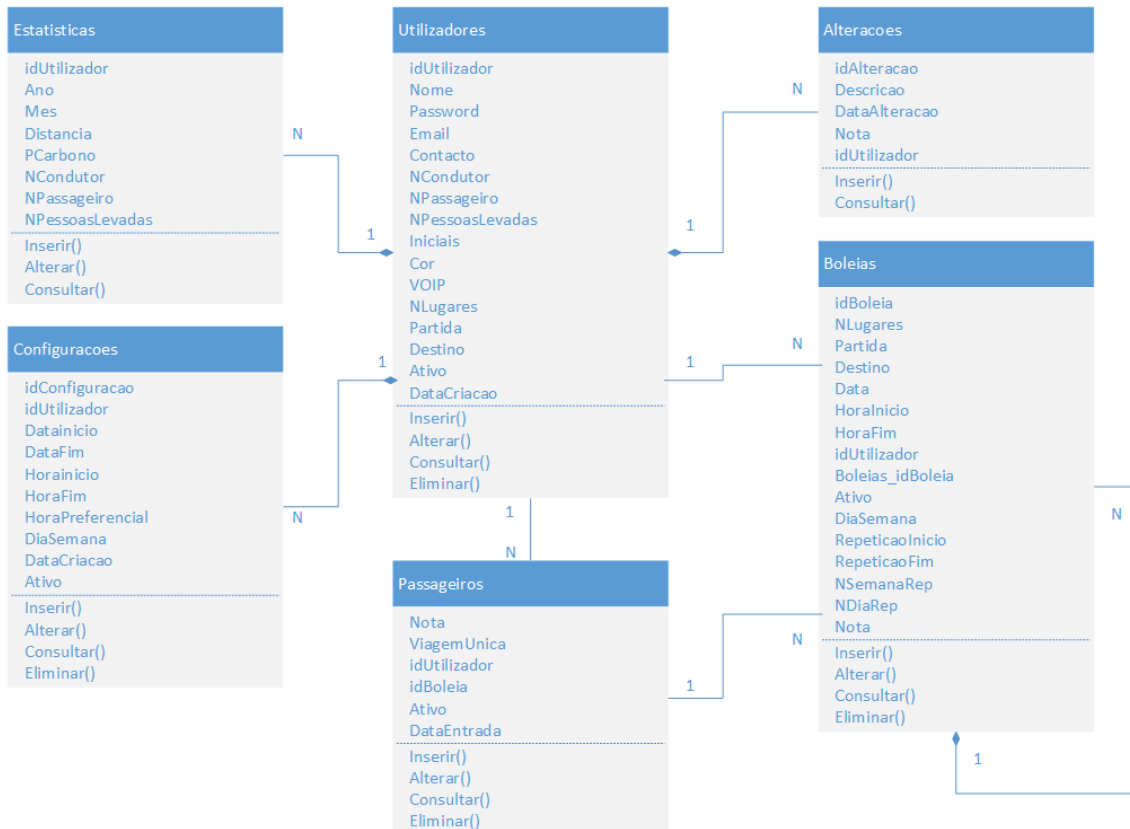


Figura 6 - Diagrama de classes

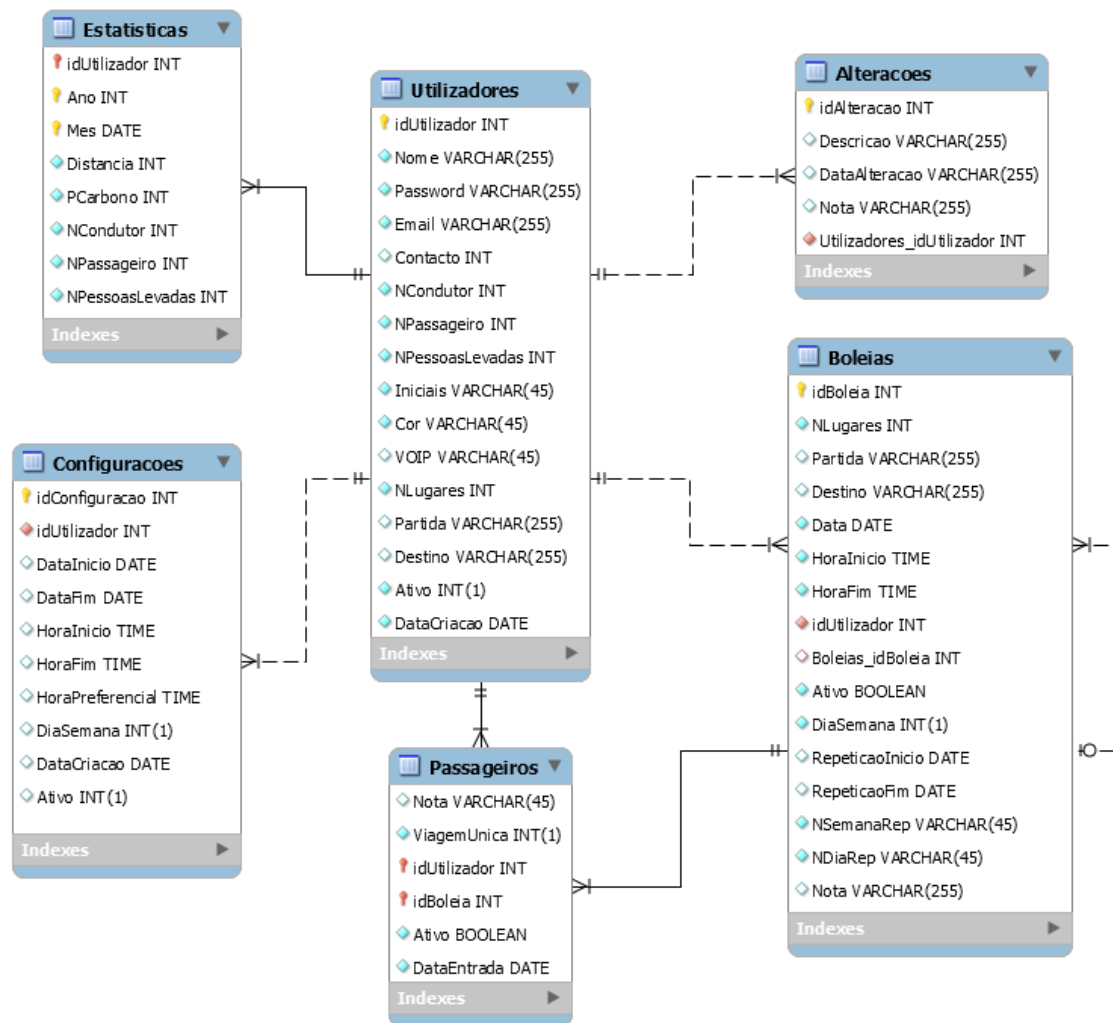


Figura 7 - Modelo EER criado em MySQL Workbench

A tabela dos utilizadores irá guardar os dados, estatísticas e as configurações de predefinição para a criação de boleias de cada utilizador (Partida, Destino, NLugares). À tabela dos utilizadores estão ligadas as tabelas das alterações, estatísticas e configurações. A tabela de alterações guarda registos das alterações mais importantes feitas por cada utilizador. A tabela de estatísticas guarda estatísticas mensais de cada utilizador, utilizando uma chave composta que inclui os campos ano e mes (que guardam o ano e o mês de uma estatística) e a chave estrangeira do utilizador a qual se aplica a estatística. Assim, garante-se que não existem registos duplicados para aquele mês e utilizador. A tabela de configurações guarda as configurações necessárias para a automatização de escolha do condutor. Esta tabela simplesmente guarda os dados para a criação de uma boleia caso o utilizador seja escolhido como condutor pela aplicação.

A tabela de boleias irá guardar os dados essenciais à boleia e os dados relacionados à repetição de boleias. A tabela de boleias está ligada à dos utilizadores para indicar o condutor da boleia e está ligada a si mesma para que, no caso de repetição de boleias, se consiga indicar quem é a

CAPÍTULO 3.METODOLOGIA E ANÁLISE DE REQUISITOS

boleia original da repetição. Também existe um campo Nota no caso do utilizador querer colocar uma nota sobre a boleia para os outros utilizadores. É de notar que o campo DiaSemana se encontra desnormalizado para acesso facilitado ao dia da semana em que a boleia se encontra.

A tabela dos passageiros vai guardar uma chave composta. Esta chave é formada pela chave estrangeira dos utilizadores e das boleias para indicar quais são os passageiros de cada boleia. Tal como a tabela de boleias, a tabela passageiros tem uma campo Nota.

3.8. Semântica

Nesta secção é apresentado o dicionário de dados e a tabela de operações às tabelas da base de dados do projeto.

3.8.1. Utilizadores

Nesta tabela,

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 encriptada com md5	Não nulo.
Email	Varchar(255)	Email do utilizador	Não nulo. Único.
Contacto	Int	Telemóvel/telefone 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 nome do utilizador, de modo a identificar o utilizador, ocupando pouco espaço quando está a ser mostrado na página.	Não nulo.
Cor	Varchar(45)	Cor identificadora do utilizador. Esta cor é utilizada para pintar as boleias na aplicação consoante o condutor da boeia, de modo a identificar rapidamente o condutor.	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)	Este campo indica o lugar de onde, habitualmente, o condutor inicia a viagem.	
Destino	Varchar(255)	Este campo indica o lugar onde, habitualmente, o condutor termina a viagem.	

Tabela 7 - Dicionário de dados da tabela Utilizadores

Operações

Nome	Descrição
Inserir()	<p>Operação que permite inserir um utilizador</p> <ol style="list-style-type: none"> 1. Introduzir Nome, Password, Email, Contacto, Iniciais, Cor, VOIP, Nlugares, Partida e Destino 2. O sistema gera o IdUtilizador
Alterar()	<p>Operação que permite alterar um utilizador</p> <ol style="list-style-type: none"> 1. Selecionar um utilizador 2. Alterar os campos necessários e que forem permitidos 3. Atualizar o utilizador
Consultar()	<p>Operação que permite consultar um utilizador</p> <ol style="list-style-type: none"> 1. Selecionar um utilizador 2. O sistema mostra os detalhes desse utilizador
Eliminar()	<p>Operação que permite eliminar um utilizador</p> <ol style="list-style-type: none"> 1. Selecionar um utilizador 2. O sistema pede a confirmação da eliminação 3. Eliminar o utilizador

Tabela 8 - Operações da tabela Utilizadores

3.8.2. 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	

Tabela 9 - Dicionário de dados da tabela Boleias

Operações:

Nome	Descrição
Inserir()	Operação que permite inserir uma boleia <ol style="list-style-type: none"> 1. Introduzir data, idutilizador, horainicio, horafim 2. O sistema gera o IdBoleia
Alterar()	Operação que permite alterar uma boleia <ol style="list-style-type: none"> 1. Selecionar uma boleia 2. Alterar os campos necessários e que forem permitidos 3. Atualizar a boleia
Consultar()	Operação que permite consultar uma boleia <ol style="list-style-type: none"> 1. Selecionar uma boleia 2. O sistema mostra os detalhes dessa boleia
Eliminar()	Operação que permite eliminar uma boleia <ol style="list-style-type: none"> 1. Selecionar uma boleia 2. O sistema pede a confirmação da eliminação 3. Atualizar o campo Ativo para 0

Tabela 10 - Operações da tabela Boleias

3.8.3. Passageiros

Dicionário de dados:

Nome do campo	Tipo de dados	Descrição	Restrições
IdUtilizador	Int	(PK/FK) Id do utilizador	Não nulo. PK/FK..
IdBoleia	Int	(PK/FK) Id da boleia	Não nulo. PK/FK
ViagemUnica	Int(1)	Indica se é ida e/ou volta.	Não nulo
Ativo	Boolean	Indica se o passageiro está ativo	Não nulo.
Nota	Varchar(255)	Nota adicional, preenchida pelo passageiro	
DataEntrada	Date	Data que indica quando o passageiro entrou na boleia	Não nulo.

Tabela 11 - Dicionário de dados da tabela Passageiros

Operações:

Nome	Descrição
Inserir()	Operação que permite inserir um passageiro. 1. Introduzir idutilizador, idboleia, viagemunica, nota. 2. O sistema gera o IdPassageiro.
Consultar()	Operação que permite consultar um passageiro. 1. Selecionar um passageiro. 2. O sistema mostra os detalhes desse passageiro.
Eliminar()	Operação que permite eliminar um passageiro. 1. Selecionar um passageiro. 2. O sistema pede a confirmação da eliminação. 3. Atualizar campo Ativo para 0.

Tabela 12 - Operações da tabela Passageiros

3.8.4. Estatísticas

Dicionário de dados:

Nome do campo	Tipo de dados	Descrição	Restrições
IdUtilizador	Int	(PK/FK) Id do utilizador	Não nulo. PK/FK.
Ano	Int(4)	(PK) Ano da estatística	Não nulo. PK.
Mes	Int(2)	(PK) Mês da estatística	Não nulo. PK.
Distancia	Int	Distância que o utilizador viajou durante o mês.	Não nulo.
PCarbono	Int	Pegada de carbono. Indica o dióxido de carbono que o utilizador salvou este mês.	Não nulo.
NCondutor	Int	Nº de vezes que o utilizador foi condutor durante o mês.	Não nulo.
NPassageiro	Int	Nº de vezes que o utilizador foi passageiro durante o mês.	Não nulo.
NPessoasLevadas	Int	Nº de pessoas que o utilizador levou este mês.	Não nulo.

Tabela 13 - Dicionário de dados da tabela Estatísticas.

Operações:

Nome	Descrição
Inserir()	Operação que permite inserir uma estatística. 1. Introduzir idutilizador, ano, mes, distancia, PCarbono, NCondutor, NPassageiro, NPessoasLevadas.
Consultar()	Operação que permite consultar uma estatística. 1. Selecionar uma estatística. 2. O sistema mostra os detalhes dessa estatística.
Alterar()	Operação que permite alterar uma estatística. 1. Selecionar uma estatística. 2. Alterar os campos permitidos. 3. Atualizar a estatística.

Tabela 14 - Operações da tabela Estatísticas.

3.8.5. Configurações

Dicionário de dados:

Nome do campo	Tipo de dados	Descrição	Restrições
idConfiguracao	Int	(PK) Id da configuração	Não nulo. PK.
idUtilizador	Int	(FK) Id do utilizador a qual se refere a configuração.	Não nulo. FK.
DataInicio	Date	Data inicial, a partir da qual se pode aplicar a configuração.	
DataFim	Date	Data final, a partir da qual não se pode aplicar a configuração.	
HoraInicio	Time	Hora de início da boleia configurada.	
HoraFim	Time	Hora final da boleia configurada.	
DiaSemana	Int(1)	Dia da semana da boleia configurada.	
DataCriacao	Date	Data de criação da configuração.	
Ativo	Int(1).	Indica se a configuração está ativa ou não.	Não nulo.

Tabela 15 - Dicionário de dados das configurações

Operações:

Nome	Descrição
Inserir()	Operação que permite inserir uma configuração. <ol style="list-style-type: none"> 1. Introduzir idUtilizador, HoraInicio, HoraFim, DataInicio, DataFim, DiaSemana, DataCriacao. 2. O sistema gera o idConfiguracao
Consultar()	Operação que permite consultar uma configuração. <ol style="list-style-type: none"> 1. Selecionar uma configuração. 2. O sistema mostra os detalhes dessa configuração.
Alterar()	Operação que permite alterar uma configuração. <ol style="list-style-type: none"> 1. Selecionar uma configuração. 2. Alterar os campos permitidos. 3. Atualizar a configuração.
Eliminar()	Operação que permite eliminar uma configuração. <ol style="list-style-type: none"> 1. Selecionar uma configuração. 2. O sistema pede a confirmação da eliminação. 3. Atualizar campo Ativo para 0.

Tabela 16 - Operações da tabela Configurações

3.8.6. Alterações

Dicionário de dados:

Nome do campo	Tipo de dados	Descrição	Restrições
idAlteracao	Int	(PK) Id da configuração	Não nulo. PK.
Descricao	Varchar(255)	Descrição da alteração onde se encontram os detalhes do evento registado.	
DataAlteracao	Date	Data em que se efetuou a alteração.	
Nota	Varchar(255)	Nota do utilizador sobre a alteração.	
IdUtilizador	Int	(FK) Id do utilizador a qual se refere a alteração.	Não nulo. FK.

Tabela 17 - Dicionário de dados da tabela Alterações

Operações:

Nome	Descrição
Inserir()	Operação que permite inserir uma alteração. <ol style="list-style-type: none"> 1. Introduzir Descricao, DataAlteracao, Nota, idUtilizador. 2. O sistema gera o idAlteracao.
Consultar()	Operação que permite consultar uma alteração. <ol style="list-style-type: none"> 1. Selecionar uma alteração. 2. O sistema mostra os detalhes dessa alteração.

Tabela 18 - Operações da tabela Alterações

4. 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 respectivas soluções e a avaliação do produto desenvolvido.

4.1. Tecnologias utilizadas

Nesta subsecção são apresentadas as tecnologias utilizadas pelo autor para o desenvolvimento da aplicação.

4.1.1. 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 se 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 utilização 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.

4.1.2. 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).

CAPÍTULO 4. DESENVOLVIMENTO

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 adaptar-se ao ecrã do dispositivo. Assim, a página consegue ser utilizada e visualizada de modo intuitivo e eficiente quer em telemóveis quer em computadores.

4.1.3. 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 atualizada 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.

4.1.4. 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 dinâmicas e interativas para o utilizador.

CAPÍTULO 4. DESENVOLVIMENTO

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.

4.1.5. 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 necessária 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.

4.1.6. 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.

4.1.7. GitHub

GitHub é um repositório web Git que oferece controlo de revisão distribuída e a funcionalidade de gestão de código-fonte do Git, com as suas próprias funcionalidades. As funções da plataforma GitHub, as aplicações desktop, e o GitHub Enterprise torna escrever código muito mais fácil para indivíduos e equipas (9).

O GitHub vai ser utilizado como gestor de versões e como backup à aplicação em desenvolvimento. O GitHub foi escolhido pelo autor pela sua experiência com a aplicação e pela simplicidade que o GitHub oferece.

4.2.Diagrama de hierarquia

O seguinte diagrama apresenta a hierarquia 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 efetuado corretamente, a uma rede de páginas que lhe dará uma série de funcionalidades e interfaces para manipular e visualizar o mapa de boleias.

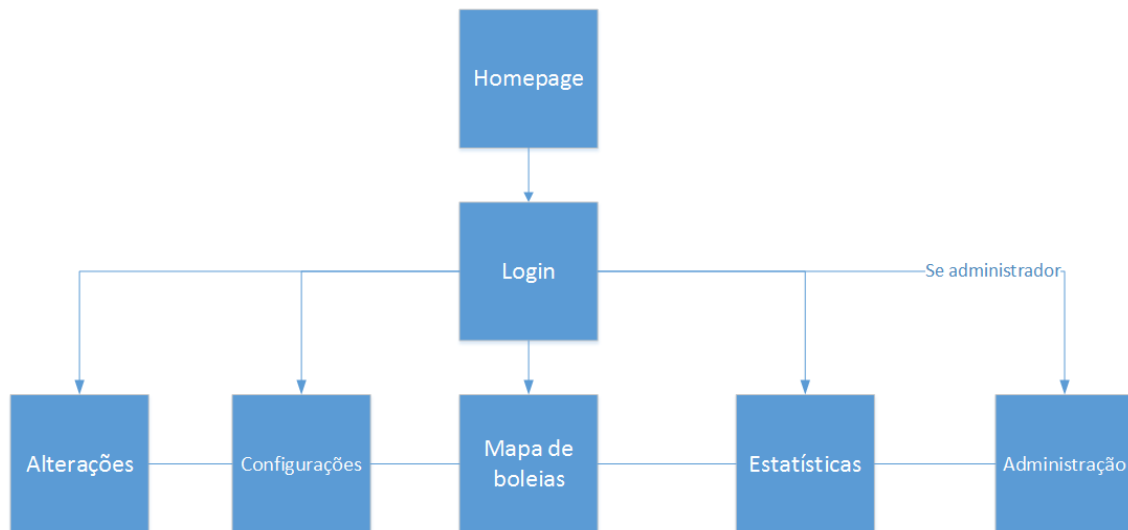


Figura 8 - Diagrama de Hiararquia

4.3. Análise e implementação

Durante a análise e implementação da aplicação, foram superados vários desafios, dos quais se destacam dois. Esta secção será utilizada para apresentar e descrever os pontos principais destes desafios e as respetivas soluções.

4.3.1. Visualização dos registos de boleias num único mapa de boleias

Problema – Visualização gráfica e intuitiva do mapa das boleias:

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:

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

Solução:

Em resposta à criação de um mapa intuitivo utilizando elementos e estrutura HTML (ponto 1 do problema), decidiu-se 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). A interface pode ser observada na Figura 9.

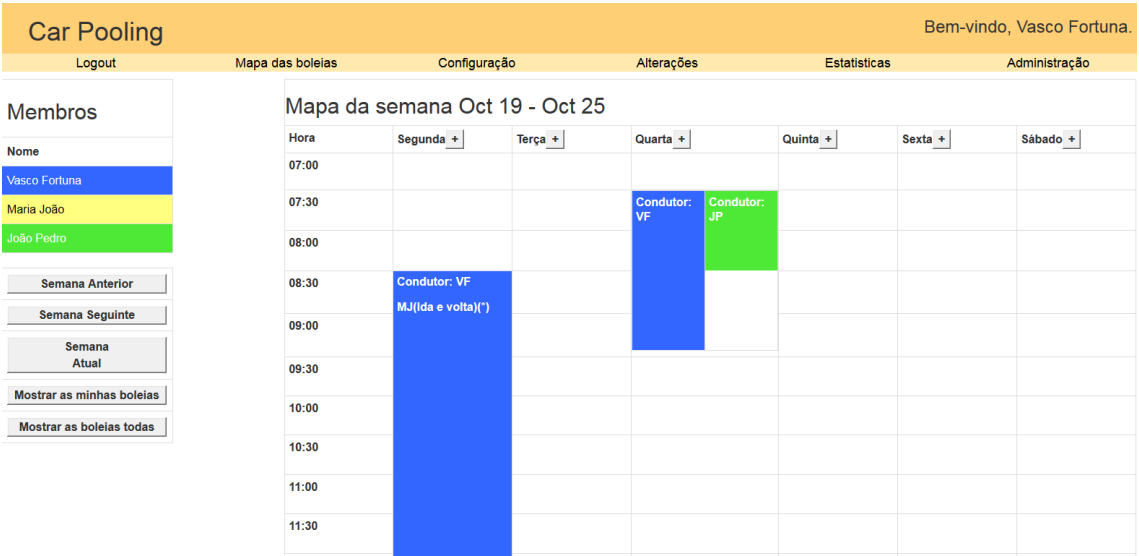


Figura 9 - Interface da aplicação carpooling

Em termos de implementação, há alguns desafios para superar. 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 célula da tabela (vamos chamar de célula à junção de cada coluna com cada linha da tabela) da esquerda para a direita e de cima para baixo, como podemos ver na figura abaixo (Figura 10). Isto é um ponto importante porque, como se 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.

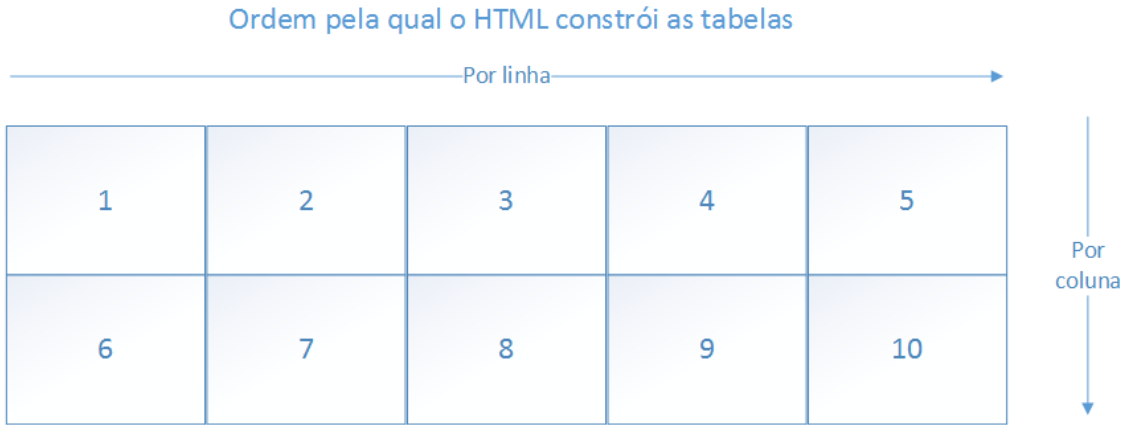


Figura 10 - Ordem de construção de tabelas pelo HTML

Tendo isto em conta, para solucionar a posição e tamanho da boleia no mapa consoante a sua data, hora inicial e hora final (ponto 2 do problema), a aplicação vai fazer uma consulta à base de dados a cada célula que ainda não esteja preenchido para averiguar se existem boleias com a hora inicial e dia da semana correspondente a essa célula (item 1 da Figura 11). No caso de existir, a aplicação vai calcular o tamanho da célula 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 uma célula de uma tabela HTML nem alinhar boleias sobrepostas na mesma célula sem elementos adicionais.

Para solucionar este problema, em referência à adaptação do mapa a boleias sobrepostas (ponto 3 do problema), decidiu-se fazer uma consulta à base de dados adicional a cada célula da tabela para descobrir se existem elementos adicionais (itens 2 e 3). Se esta consulta não devolver resultados, a célula é preenchida normalmente sem elementos adicionados (item 7). Se devolver, será criada uma tabela adicional dentro dessa célula (item 5). Esta tabela adicional terá uma coluna para cada boeia nesse período de tempo, em que cada boeia sobreposta será posicionada nessa coluna consoante a sua diferença de tempo em relação à hora inicial da boeia encontrada na primeira consulta.

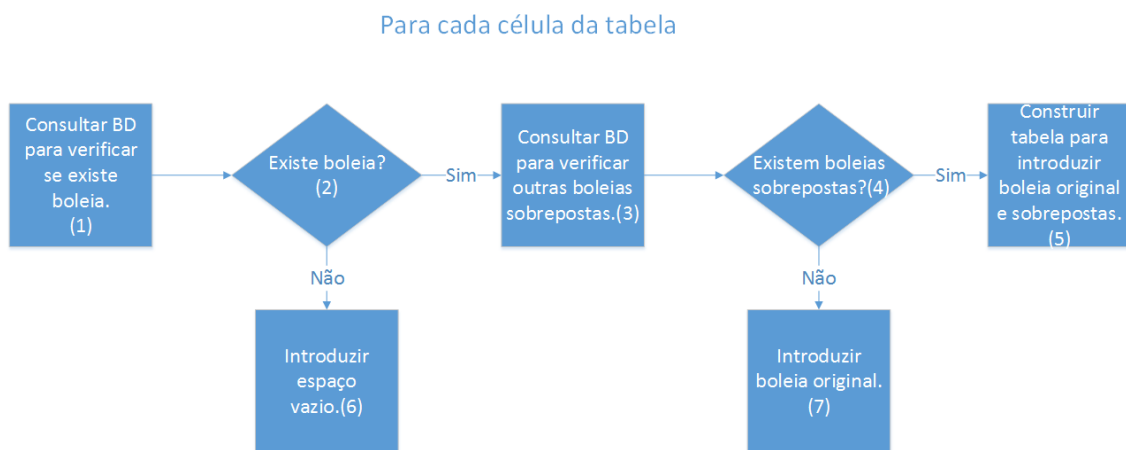


Figura 11 - Fluxograma do algoritmo de leitura de boleias

4.3.2. Interface e interação eficiente da aplicação

Problema – Interação fácil e intuitiva com o mapa das boleias, nomeadamente nas operações mais comuns:

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 a interface tem de ser o mais simples e automatizada possível. Tendo em conta estes aspetos, podemos listar os pontos essenciais deste problema:

1. Tornar o processo de inserir/eliminar/alterar/repetir boleias e inserir/retirar passageiros, fácil e rápido (fazer ações com o menor número de clicks).
2. Tornar a interface intuitiva, de modo ao utilizador perceber como a aplicação funciona sem a necessidade de um tutorial ou instruções.

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, decidiu-se colocar os casos de uso mais relevantes e mais usados (inserir/alterar/eliminar/repetir boleia, entrar/sair de uma boleia) nesta interface.

De modo a tornar o processo de inserir boleias rápido e fácil (ponto 1 do problema), decidiu-se tornar cada célula do mapa de boleias não preenchido num botão. Este botão foi feito com uma função Javascript que é acionado quando o utilizador clica num elemento HTML ligado a essa função. O botão permite ao utilizador colocar uma boleia, nessa célula, com todos os seus dados automaticamente. Para o caso de todas as células estarem preenchidas nesse dia, foi colocado um botão ao lado dos dias da semana, onde o utilizador simplesmente insere a hora da boleia.

Seguindo estas ideias e de modo a tornar a interface intuitiva (ponto 2 do problema), foi feito o mesmo para as células preenchidas, tornando estas células igualmente em botões. No caso de o utilizador clicar numa célula preenchida cuja boleia não é sua, simplesmente aparece 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, aparece uma caixa semelhante, mas com elementos de texto caso o utilizador queira alterar os dados da boleia, repetir a boleia ou eliminá-la.

Todas estas decisões fazem com todas as funcionalidades essenciais não só se encontrem no mapa de boleias, mas também com que a interface simule um calendário interativo. Assim, o

CAPÍTULO 4.DESENVOLVIMENTO

utilizador consegue facilmente identificar como utilizar a aplicação corretamente, tal como podemos observar na imagem a seguir (Figura 12):

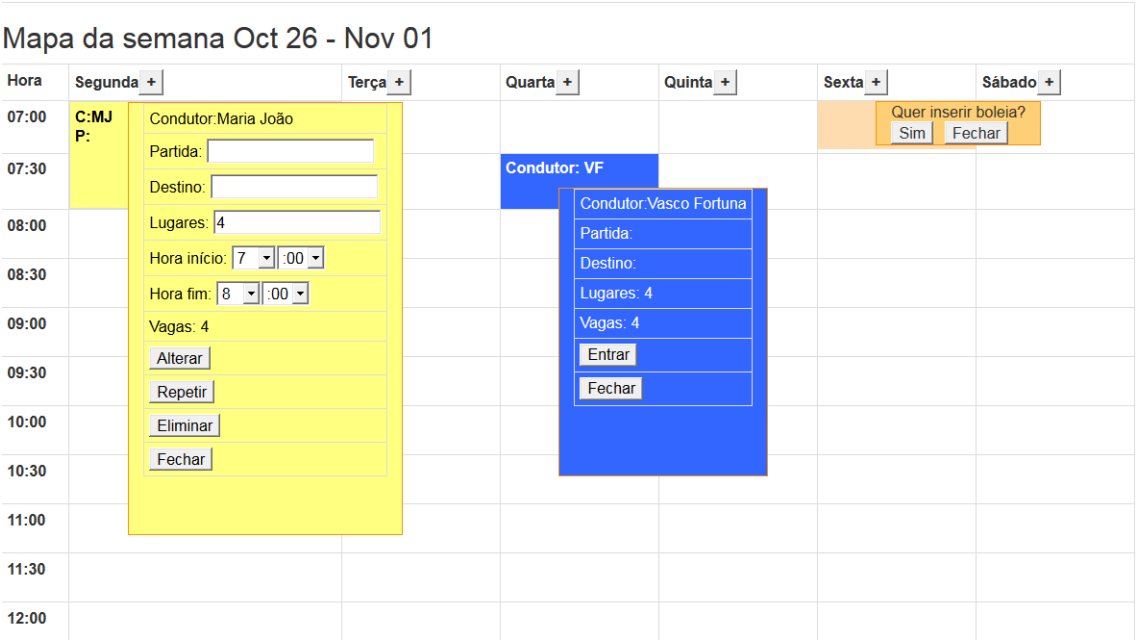


Figura 12 - Interface do mapa de boleias (À esquerda: uma boleia selecionada pelo condutor; Ao centro: uma boleia selecionada por um membro; À direita: um espaço vazio selecionado;)

4.4. 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.

4.4.1. Testes

Durante a sua programação, a aplicação foi testada continuamente para garantir a sua integridade e segurança. Para manter a integridade da aplicação, testou-se 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 uma melhor avaliação da aplicação, foi utilizada uma aplicação web desenvolvida pela Google chamada Pagespeed Insights. Esta aplicação avalia o desempenho da aplicação quer em telemóvel quer em computador. Esta avaliação inclui a experiência da aplicação para o utilizador e o desempenho do carregamento da aplicação no browser.

Para telemóvel, o Google Insights encontrou um erro de bloqueio de Javascript (Anexo C1). Mas devido à estrutura do código desenvolvido estar focalizado para computador, não foi possível corrigir este erro até à entrega do relatório. Em termos de desempenho, aplicação só encontrou alguns avisos menores que incluem a compactação do CSS e HTML em 6% e a utilização da cache do navegador. A compactação é mínima e não se justifica fazer. Além disso, não se consegue tirar partido da cache devido aos visuais dinâmicos da aplicação, especialmente do mapa de boleias. No entanto, a aplicação passou nos testes de redução de Javascript, redução de tempo de resposta ao servidor e priorização ao conteúdo visível da aplicação.

A aplicação, para telemóvel, conseguiu pontuação máxima na experiência de utilizador (Anexo C2). Os testes incluíram configuração da viewport, dimensionamento do conteúdo em função da janela atual, dimensionamento adequado de elementos táteis e sem a utilização de plugins.

Para computador, a aplicação foi avaliada, em termos de desempenho, com uma pontuação excelente contendo só alguns avisos (Anexo C3). Os avisos incluíram também, compactação de CSS/HTML de 6% e utilização da cache do navegador. Mas, tal como no caso do telemóvel, não se justifica fazer compactação tão pequena e nesta aplicação, não se consegue utilizar a cache eficientemente devido aos visuais dinâmicos da aplicação, especialmente do mapa de boleias.

5. Conclusão

Foi proposta uma aplicação web de organização de boleias para um grupo de docentes do com funcionalidades específicas como automatização de escolha de condutor, estatísticas dos passageiros e um mapa de boleias único. No entanto, só foi possível alguns desses objetivos que incluíram os objetivos mais básicos e com prioridade mais alta:

- Registrar, alterar e eliminar utilizadores.
- Organizar pedidos e ofertas de boleias dentro do grupo.
- Contabilizar boleias efetuadas e recebidas.
- Registrar as alterações mais relevantes à base de dados.
- Interface eficiente e intuitiva.

No entanto, não foi possível completar todos os objetivos previstos até à finalização deste relatório. Nomeadamente, os objetivos que não consegui completar foram:

- Duplicar mapas de boleias para semestres, anos.
- Preencher mapas automaticamente de acordo com as boleias contabilizadas de cada membro.
- Enviar notificações por email acerca das próximas boleias ou alterações de boleias que afetam o utilizador.
- Calcular a redução da pegada de carbono.
- Sincronizar a BD das boleias com o Google Spreadsheet.
- Criar cópias de segurança dos mapas de boleias.

Os objetivos anteriores não foram completos devido à falta de tempo por parte do autor. A aplicação encontra-se no site carpooling-vascof.rhcloud.com na presente data de entrega do relatório (Novembro, 2015) com as funcionalidades anteriormente mencionadas.

Para futuro trabalho, prevê-se a completação dos objetivos que não foram completados e a expansão da aplicação para um mercado global, ou seja, tornar a aplicação disponível e operacional para qualquer utilizador e não só para um grupo específico de pessoas.

Concluindo, o que me levou a trabalhar neste projeto foi a oportunidade de abranger os meus conhecimentos em programação web e design web e a oportunidade de aplicar os conhecimentos que ganhei ao longo do curso num projeto que visa responder a necessidades reais.

6. Bibliografia

1. Extreme Programming. *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. GitHub | Crunchbase. *Crunchbase*. [Online]
<https://www.crunchbase.com/organization/github>.
10. **DUCKETT, JON.** *HTML & CSS*. Indianapolis : 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://palleton.com/>.

CAPÍTULO 6.BIBLIOGRAFIA

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

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

7. Anexos

• Anexo A

Nome	Alterar boleia
Objetivo	O ator altera uma boleia
Prioridade	Média
Pré-condição	Login válido
Cenário principal	<ol style="list-style-type: none"> 1. O Caso de Uso começa quando o ator clica no botão “Alterar” após selecionar 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	<p>O ator pode cancelar a operação a qualquer momento.</p> <ol style="list-style-type: none"> 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. 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	<ul style="list-style-type: none"> • Verificar se o mapa de boleias é atualizado corretamente. • Verificar se as estatísticas são atualizadas corretamente. • Verificar se os campos são preenchidos corretamente <ul style="list-style-type: none"> ○ Se os campos numéricos só contêm caracteres numéricos. ○ Se os campos alfabéticos só contêm caracteres alfabéticos.

Tabela 19 - Descrição do caso de uso "Alterar boleia"

Nome	Entrar numa boleia
Objetivo	O ator entra numa boleia
Prioridade	Alta
Pré-condição	Login válido
Cenário principal	<ol style="list-style-type: none"> 1. O Caso de Uso começa quando o ator clica no botão “Entrar” após selecionar 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	<p>O ator pode cancelar a operação a qualquer momento.</p> <p>4. a. O ator não preenche todos os campos obrigatórios e aparece uma mensagem de erro.</p>
Pós-condição	Os dados estatísticos relevantes ao ator são atualizados.
Casos de teste	<ul style="list-style-type: none"> • Verificar se o mapa de boleias é atualizado corretamente. • Verificar se as estatísticas são atualizadas corretamente.

Tabela 20 - Descrição do caso de uso "Entrar numa boleia"

Nome	Repetir boleia
Objetivo	O ator repete uma boleia
Prioridade	Média
Pré-condição	Login válido
Cenário principal	<ol style="list-style-type: none"> 1. O Caso de Uso começa quando o ator clica no botão “Repetir” após selecionar 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	<p>O ator pode cancelar a operação a qualquer momento.</p> <p>4. a. O ator não preenche os campos corretamente e aparece mensagem de erro.</p>
Pós-condição	Os dados estatísticos relevantes ao condutor e passageiros (se existirem) da boleia repetida são atualizados.
Casos de teste	<ul style="list-style-type: none"> • Verificar se o mapa de boleias é atualizado corretamente. • Verificar se as estatísticas são atualizadas corretamente.

Tabela 21 - Descrição do caso de uso "Repetir boleia"

Nome	Sair de uma boleia
Objetivo	O ator sai uma boleia
Prioridade	Média
Pré-condição	Login válido
Cenário principal	<ol style="list-style-type: none"> 1. O Caso de Uso começa quando o ator clica no botão “Sair” após selecionar uma boleia. 2. O sistema pede a confirmação da saída. 3. O ator clica no botão “Ok”, confirmando a saída. 4. O sistema desativa o passageiro.
Cenário alternativo	O ator pode cancelar a operação a qualquer momento.
Pós-condição	Os dados estatísticos relevantes ao condutor e passageiros (se existirem) da boleia repetida são atualizados.
Casos de teste	<ul style="list-style-type: none"> • Verificar se o mapa de boleias é atualizado corretamente. • Verificar se as estatísticas são atualizadas corretamente.

Tabela 22 - Descrição do caso de uso "Sair de uma boleia"

- Anexo B

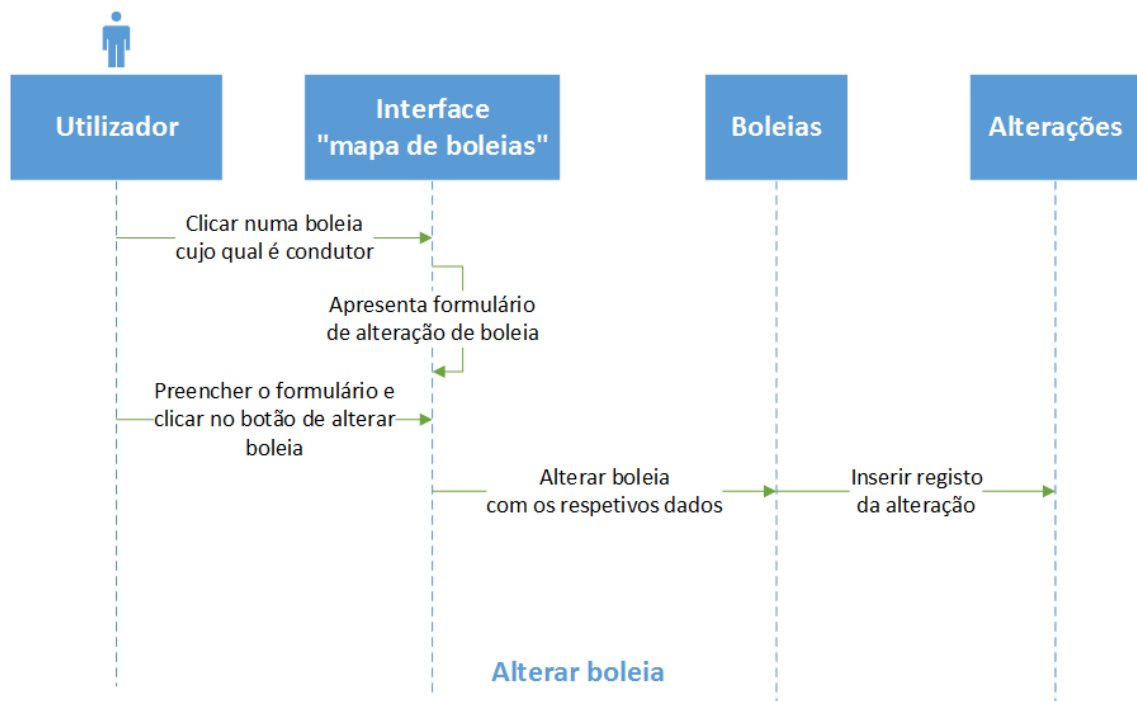


Figura 13 - Diagrama de sequência do caso de uso "Alterar boleia"

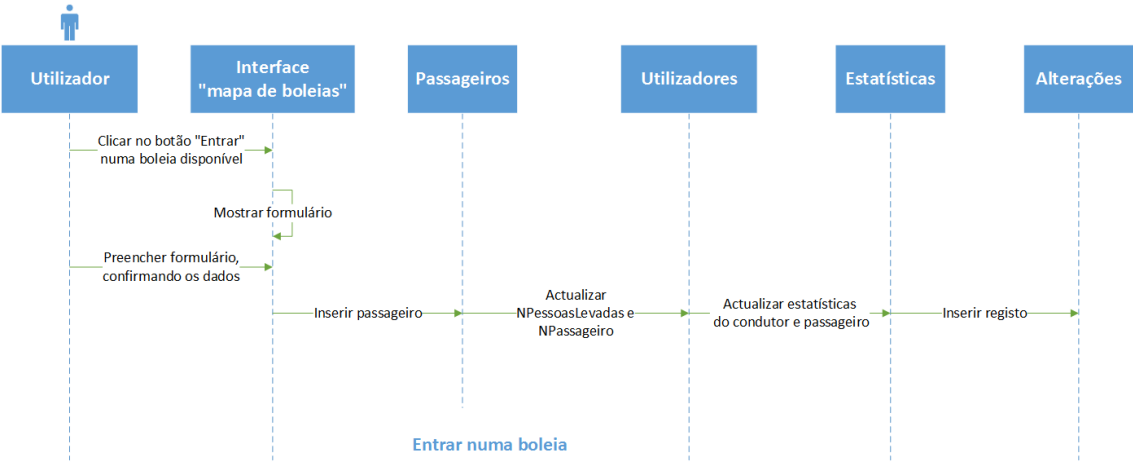


Figura 14 - Diagrama de sequência do caso de uso "Entrar numa boleia"

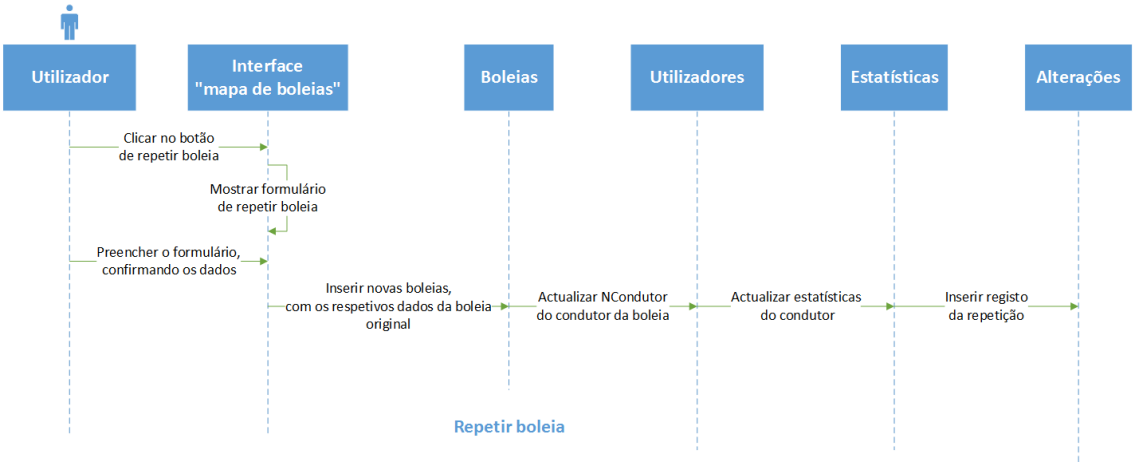


Figura 15 - Diagrama de sequência do caso de uso "Repetir boleia"

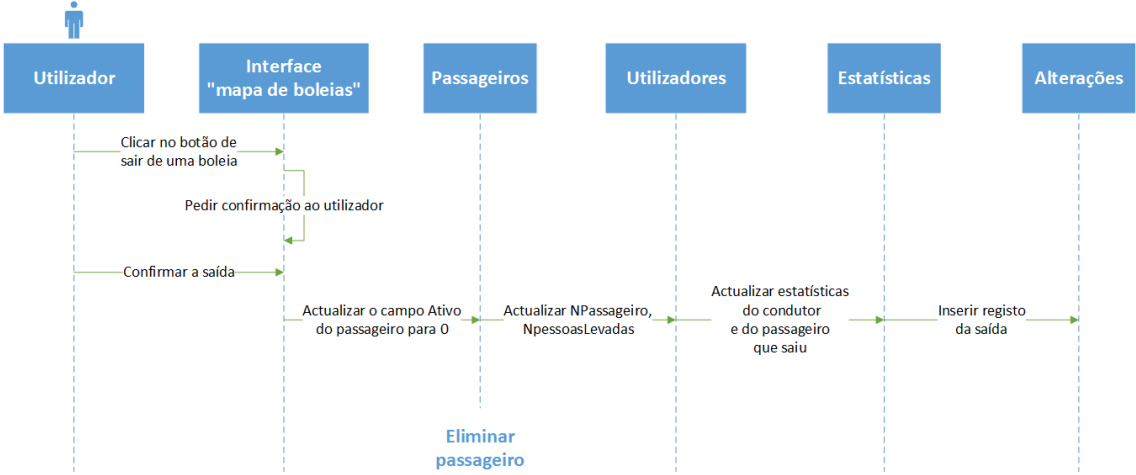


Figura 16 - Diagrama de sequência do caso de uso "Sair de uma boleia"

• Anexo C1

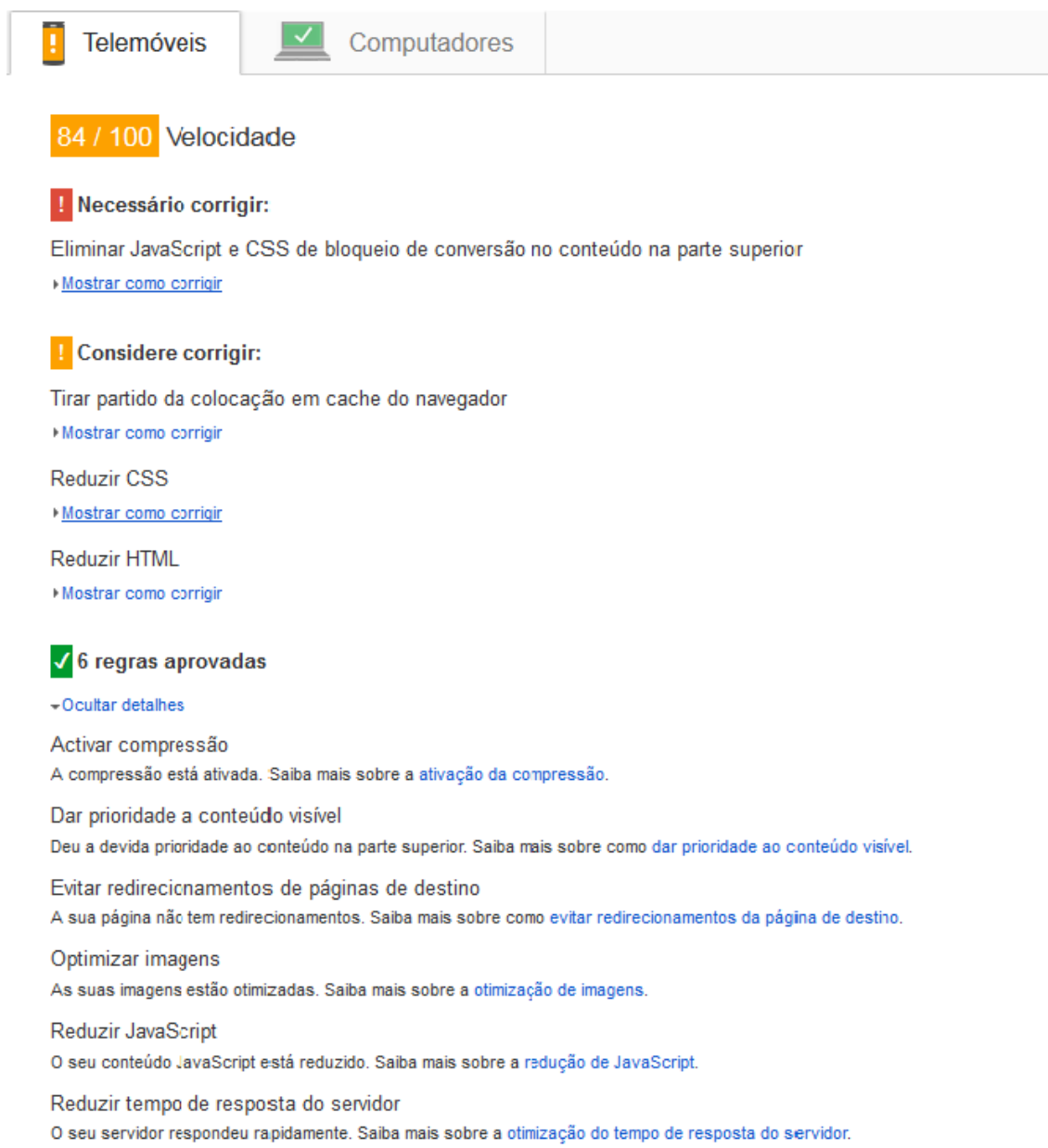


Figura 17 - Avaliação do PageSpeed da aplicação para telemóvel (Parte 1)

● Anexo C2

100 / 100 Experiência do utilizador

✓ Parabéns! Não foram encontrados problemas.

Configurar a viewport

A sua página especifica uma viewport que corresponde ao tamanho do dispositivo, o que lhe permite ser devidamente processada em todos os dispositivos. Saiba mais sobre como [configurar viewports](#).

Dimensionar conteúdo em função da janela atual

O conteúdo da sua página cabe dentro da janela atual. Saiba mais sobre o [redimensionamento de conteúdo para a janela atual](#).

Dimensionar elementos táteis adequadamente

A dimensão de todos os links/botões da sua página é suficientemente grande para permitir que um utilizador toque facilmente num ecrã tátil. Saiba mais sobre o [dimensionamento adequado de elementos táteis](#).

Evitar anúncios intercalares de instalação de aplicação que ocultem conteúdo (BETA)

A sua página não parece ter anúncios intercalares de instalação de aplicação que ocultem um volume significativo de conteúdo. Saiba mais sobre a importância de [evitar a utilização de anúncios intercalares de instalação de aplicação](#).

Evitar plug-ins

Aparentemente a sua página não utiliza plug-ins, o que pode evitar que determinados conteúdos sejam utilizados em várias plataformas. Saiba mais sobre a importância de [evitar plug-ins](#).

Utilizar tamanhos de tipo de letra legíveis

O texto na sua página é legível. Saiba mais sobre [a utilização de tamanhos de tipo de letra legíveis](#).

Figura 18 - Avaliação do PageSpeed da aplicação para telemóvel (Parte 2)

• Anexo C3

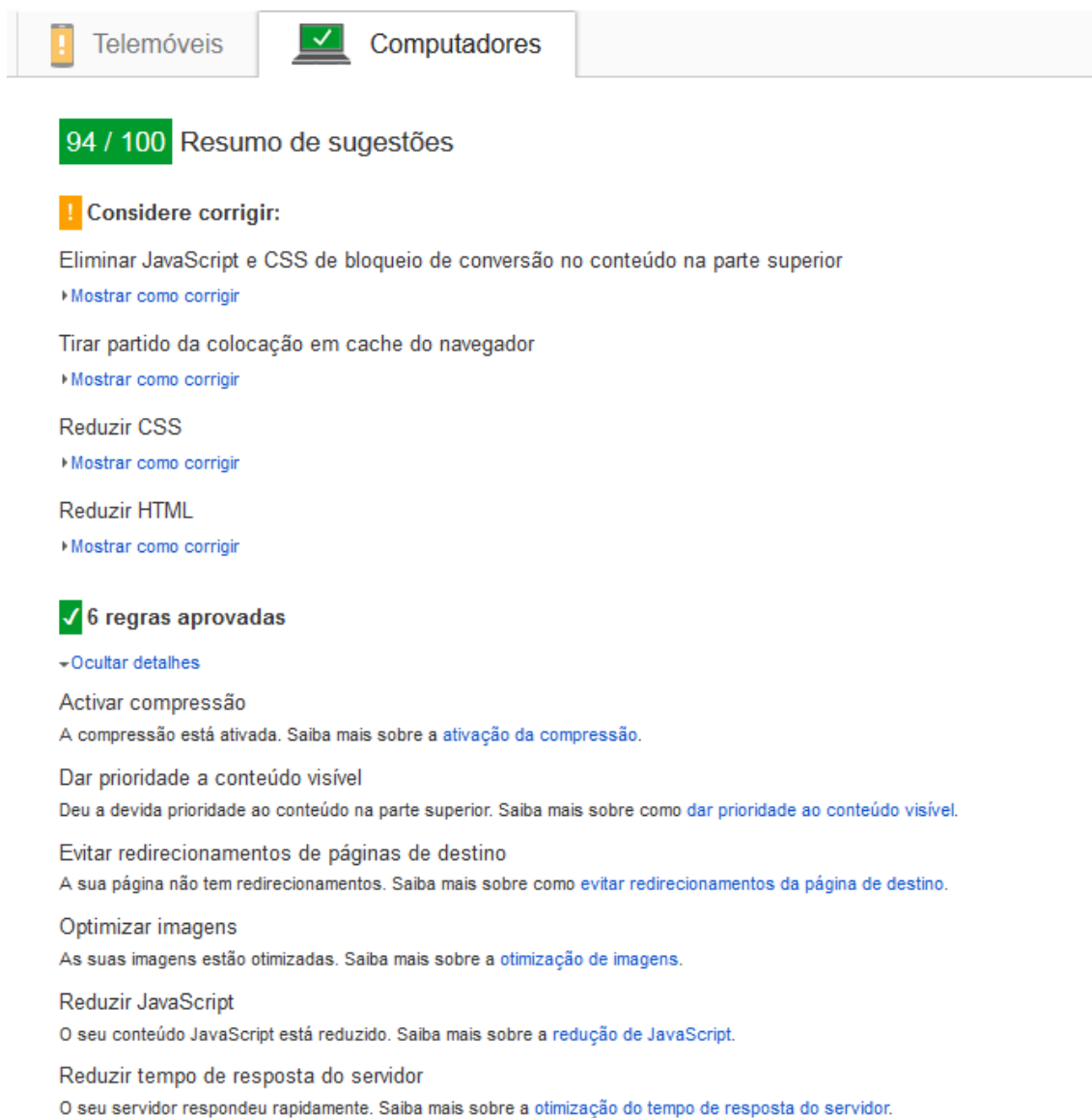


Figura 19 - Avaliação do PageSpeed da aplicação para computador