

---

# Relatório de Especificação: Desenho

**Projecto:** CityParking  
**Grupo:** Grupo 1  
Daniel Correia (nº 88753)  
Francisco Martinho (nº 85088)  
Pedro Almeida (nº 89205)  
Pedro Valente (nº 88858)  
Renato Valente (nº 89077)  
**Data de preparação:** Aveiro, 25 de Outubro de 2019  
**Circulação:** Docentes e Discentes de AMS.

## Controlo de versões

Quando?	Responsável	Alterações significativas
21-11-2109	Pedro Valente Pedro Almeida Renato Valente	Elaboração do relatório.

# Índice

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Objetivos e restrições da arquitetura</b>	<b>7</b>
2.1	Objetivos da arquitetura face aos requisitos do domínio	7
2.2	Atributos de qualidade com impacto na arquitetura	7
2.3	Decisões tomadas e justificação	8
<b>3</b>	<b>Vistas de arquitetura</b>	<b>9</b>
3.1	Arquitetura lógica da solução	9
3.2	Arquitetura operacional	9
<b>4</b>	<b>Modelo de comportamento</b>	<b>10</b>
4.1	Interação entre módulos	10
4.2	Operações de sistema	10
4.3	Integrações com sistemas externos	10
<b>5</b>	<b>Tecnologias e implementação</b>	<b>11</b>
<b>6</b>	<b>Protótipos dos produtos</b>	<b>12</b>
<b>7</b>	<b>Anexos</b>	<b>13</b>

# 1 Introdução

A plataforma CityParking tem como objetivo responder ao tempo perdido por parte dos condutores na procura de um lugar para estacionar, assim como, gastos desnecessários em tarifas de estacionamento.

O sucesso desta plataforma vai depender dos seus utilizadores, quanto maior a aderência por parte da população maior será o sucesso da plataforma. Isto porque, a confiança por parte dos fornecedores de estacionamento vai aumentar, aumentando consequentemente a probabilidade de estes oferecerem preços mais acessíveis. Juntamente, uma maior confiança por parte dos fornecedores vai manter um contacto saudável entre estes e a plataforma CityParking.

Neste relatório iremos explorar a arquitetura da plataforma, todo o backend, assim como as ferramentas utilizadas por nós para que tudo funcione.

Os resultados considerados mais importantes vão estar apresentados nos diagramas que irão estabelecer a implementação da plataforma.

O código do projeto, tal como o site, estão alocados no GitHub. Sendo esta a plataforma utilizada para trabalho conjunto.

## Índice de diagramas

Diagrama 1: Arquitetura lógica da solução	9
Diagrama 2: Arquitetura operacional	10
Diagrama 3: Autenticação	11
Diagrama 4: Reserva do lugar de estacionamento	12
Diagrama 5: Autenticação na aplicação	13
Diagrama 6: Esquematização do processo para a reserva e pagamento do lugar de estacionamento	13
Diagrama 7: Comunicação com Google Maps	14
Diagrama 8: Comunicação com Firebase	15
Diagrama 9: Comunicação da aplicação CityParking com MbWay	15
Diagrama 10: Comunicação com MbWay	16

Índice de tabelas

Tabela 1 :Atributos de qualidade com impacto na arquitetura	7
---	---

## 2 Objetivos e restrições da arquitetura

O objetivo deste relatório é o desenho da arquitetura da aplicação CityParking respondendo de acordo com os requisitos previamente definidos.

### 2.1 Objetivos da arquitetura face aos requisitos do domínio

A modelação do sistema está adaptada para Android, mais tarde, irão ser explorados outros sistemas para a modelação da aplicação, como por exemplo IOS.

O hardware estará continuamente ligado ao sistema de base de dados, comunicando e fornecendo todas as informações em tempo real, hardware este que faz parte de um sistema externo relativamente à base de dados.

O hardware estará presente perto/nos locais de estacionamento, não sendo da responsabilidade da plataforma CityParking mas sim dos fornecedores de estacionamento.

Sem os sensores (hardware) não poderá ser fornecida a informação relativa à disponibilidade do lugar de estacionamento, pelo que, esta poderá ser uma situação adversa ao bom funcionamento da aplicação. Infelizmente, sendo que os sensores pertencem aos fornecedores de estacionamento este problema não poderá ser resolvido por nós. Uma solução possível seria o aviso de imediato do fornecedor para que este possa arranjar/substituir o sensor de forma a que tudo volte a funcionar como é devido.

Sendo que a base de dados apenas vai gerir a ocupação, ou não, de um lugar de estacionamento, não deverá haver restrições quanto ao armazenamento, ou problemas relativamente à quantidade de dados trabalhados. De notar que os sensores não têm como propósito guardar qualquer tipo de vídeo ou fotografia.

A implementação do sistema inicialmente só será efetuada em equipamentos móveis.

### 2.2 Atributos de qualidade com impacto na arquitetura

Os atributos de qualidade do CityParking foram definidos no relatório “AMS-E4\_v2-final.pdf” (Relatório Técnico: Análise de Requisitos). Os atributos estão divididos em requisitos de usabilidade, desempenho, segurança e integridade de dados e documentação.

Na seguinte tabela, estão explicitados os requisitos que possuem impacto na arquitetura.

#	Atributos de qualidade	Impacto na arquitetura
RQua.1	Aplicação intuitiva de modo a facilitar a interação com o utilizador.	O desenvolvimento das aplicações deve ter em conta as regras de usabilidade de forma a serem responsivas e de fácil utilização.
RQua.2	Legibilidade da informação.	Usar cores e fontes que ajudem a legibilidade da informação.
RQua.3	Cumprir tempos de resposta para garantir o bom desempenho e da aplicação e uma boa experiência para o utilizador.	-
RQua.4	Apresentar lugares de	A arquitetura tem de assegurar a

	estacionamento livres com eficácia.	eficácia a nível de gps.
RQua.5	Segurança dos dados dos utilizadores.	A arquitetura tem de garantir a segurança dos dados pessoais dos utilizadores.
RQua.6	Apoio ao cliente deve ser rápido e eficaz.	Pode ser considerada a criação de um canal para o atendimento online.
RQua.7	Aplicação deve estar disponível a todos os momentos.	Sistemas críticos devem possuir redundância.

Tabela 1: Atributos de qualidade com impacto na arquitetura

## 2.3 Decisões tomadas e justificação

No desenvolvimento da aplicação está a ser utilizado Android Studio, JavaScript e Firebase.

A escolha foi baseada na procura de uma solução fácil e eficaz.

Android Studio foi a escolha pois nesta fase inicial o público alvo são os utilizadores de Android, sendo que a maioria da população usa Android este será o primeiro alvo exploratório.

Firebase foi escolhida pois permite facilitar todo o backend sem ter que se desenvolver muito código, assim como desenvolver uma base de dados segura e rápida. Esta solução permite juntar base de dados e API num só.

JavaScript fornece imensas oportunidades e comunica extremamente bem com as duas escolhas previamente apresentadas.

Como conclusão, estas soluções trazem para a arquitetura rapidez e facilidade no trabalho que se está a desenvolver e tudo isto, a custo zero.



### 3 Vistas de arquitetura

#### 3.1 Arquitetura lógica da solução

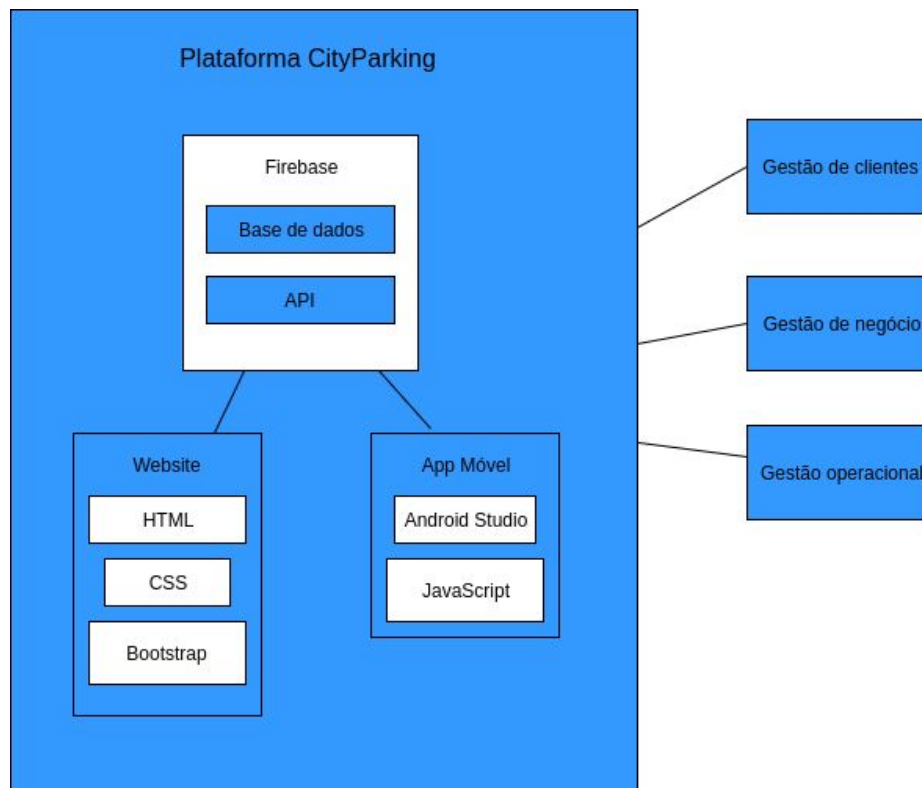


Diagrama 1: Arquitetura lógica da solução

**Plataforma CityParking** - Contém a base de dados, Website e App Móvel. É o cérebro de toda a operação e será aqui que é feita a gestão dos clientes, negócio e de toda a operação.

**Base de dados e API** - Feita a partir de Firebase, contém toda a informação da plataforma CityParking, ou seja, aqui está armazenada toda a informação relativa aos clientes da plataforma assim como toda a informação enviada pelos fornecedores de estacionamento relativa aos lugares de estacionamento da cidade/município em questão.

**Website** - Feito a partir de HTML, CSS e Bootstrap, este é apenas um website demonstrativo das funcionalidades da Plataforma CityParking. Aqui o cliente poderá visualizar o que poderá fazer utilizando a aplicação CityParking assim como registar-se na plataforma.

**App Móvel** - Feita através de Android Studio e JavaScript, é aqui que o utilizador poderá efetuar todas as operações relativas à plataforma CityParking, tais como, criação de conta ou reserva de lugares. Os dados criados ou utilizados estarão seguros na base de dados da plataforma.

**Gestão de clientes** - Aliada à base de dados, aqui está os dados dos clientes e a informação relativa aos lugares reservados pelos mesmos. Aqui será preciso ter mecanismos de segurança para salvaguardar a confidencialidade dos dados pessoais de cada cliente.

**Gestão de negócio** - Aqui será feita a gestão do número de lugares reservados, número e clientes, gestão de recursos humanos que possam ser necessários assim como a gestão de todos os lucros e gastos.

**Gestão operacional** - Aliada à base de dados aqui é feita a gestão dos lugares de estacionamento disponíveis aos utilizadores da plataforma CityParking.

### 3.2 Arquitetura operacional

Visual Paradigm Standard (Valente (Universidade de Aveiro))

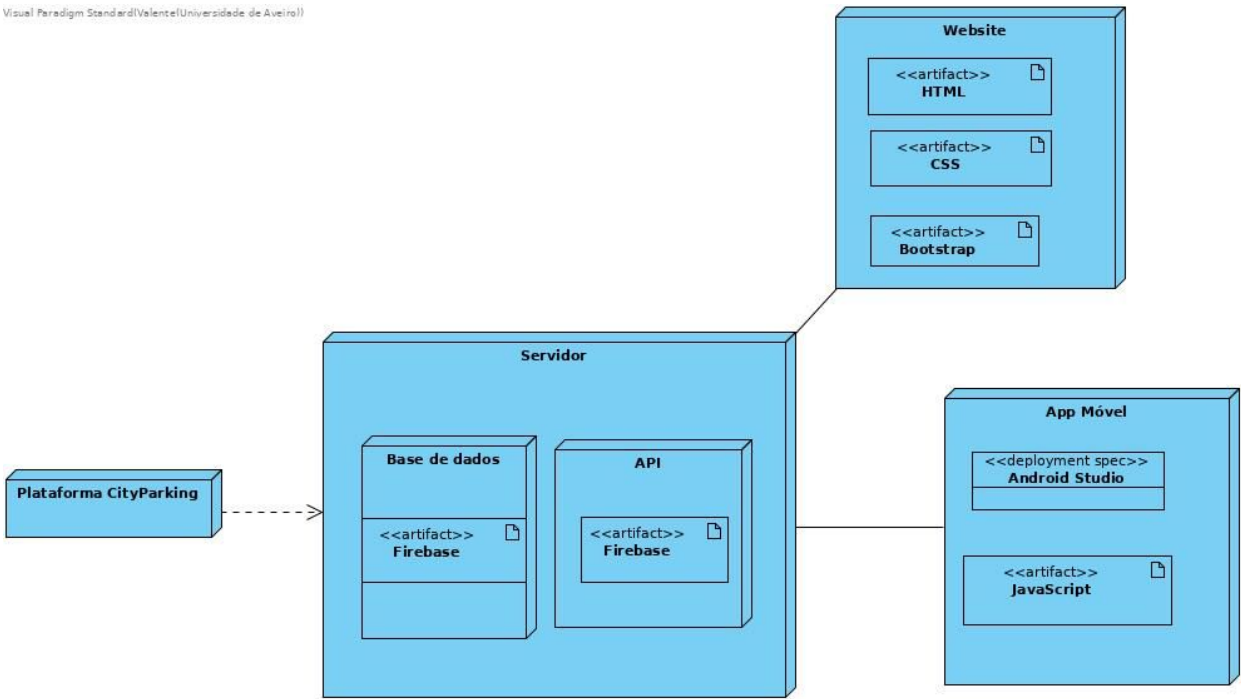


Diagrama 2: Arquitetura operacional

## 4 Modelo de comportamento

### 4.1 Interação entre módulos

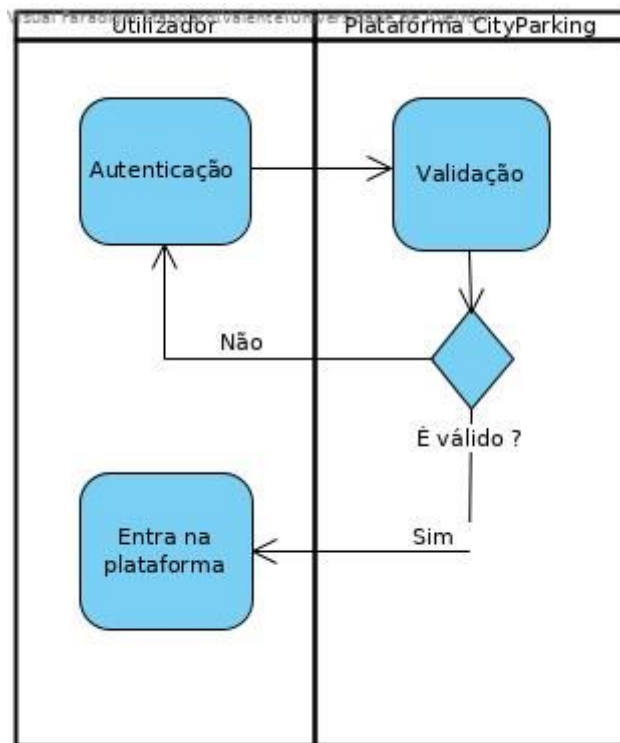


Diagrama 3: Autenticação

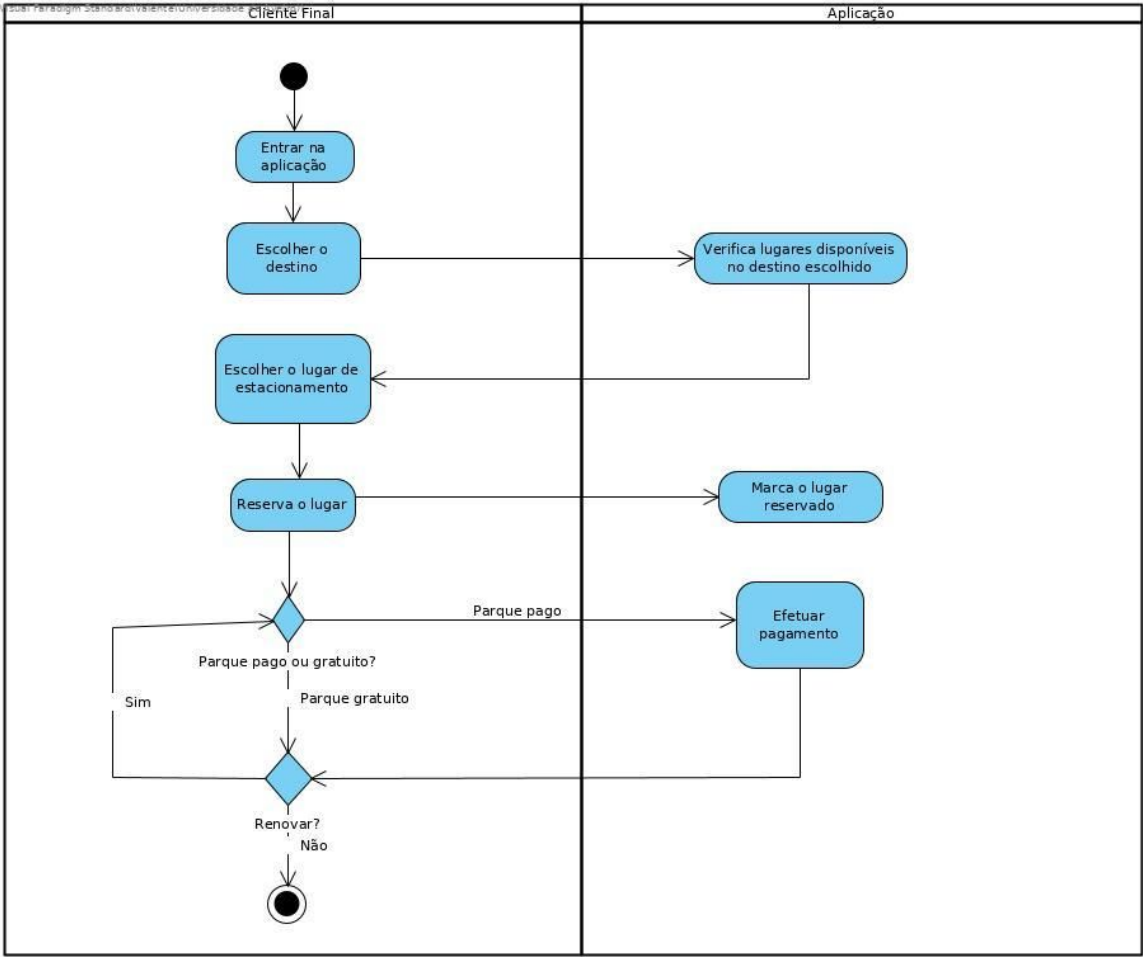


Diagrama 4: Reserva do lugar de estacionamento

**Nota:** A entrada na aplicação explicitada no Diagrama 4 (acima) utiliza as ações do Diagrama 3 - Autenticação.

4.2 Operações de sistema

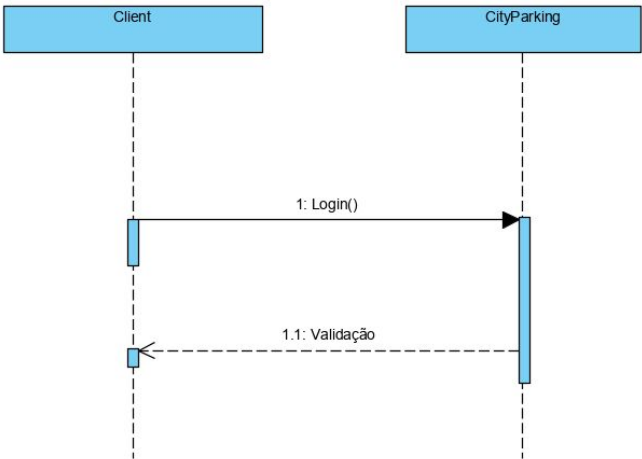


Diagrama 5: Autenticação na aplicação

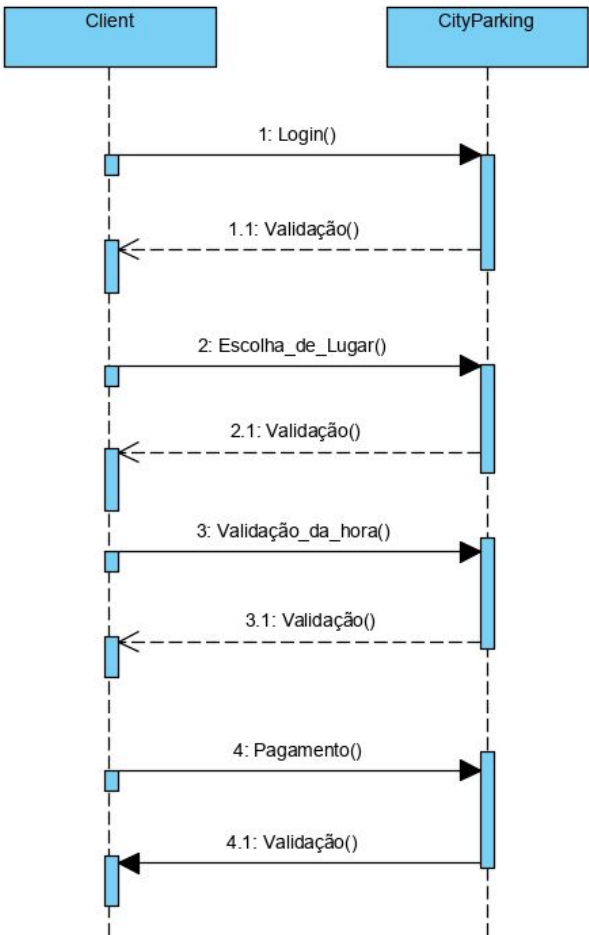


Diagrama 6: Esquematização do processo para a reserva e pagamento do lugar de estacionamento

## 4.3 Integrações com sistemas externos

Os sistemas externos utilizados pela aplicação CityParking são:

- Empresa de pagamento: MbWay
- Empresas de serviços: Firebase e Google

A empresa Firebase providencia a base de dados para a aplicação CityParking assim como uma API.

Já a Google, através da sua funcionalidade Google Maps permite à aplicação CityParking usufruir de um mapa completo e conhecido pelos utilizadores.

### 1 - Empresas de serviços

#### 1.1 - Google Maps

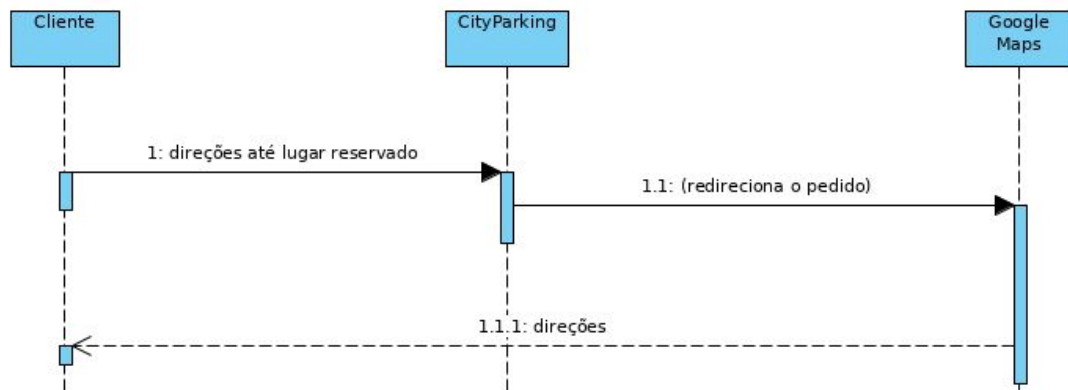


Diagrama 7: Comunicação com Google Maps

1.2 - Firebase

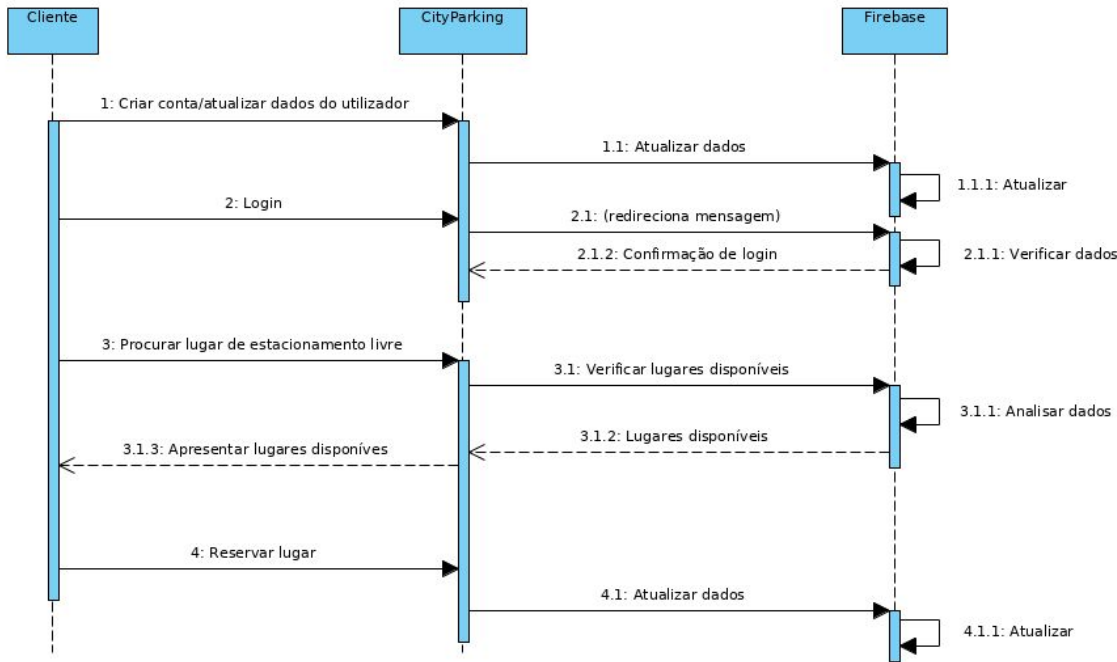


Diagrama 8: Comunicação com Firebase

2 - Empresas de pagamentos

2.1 - MbWay

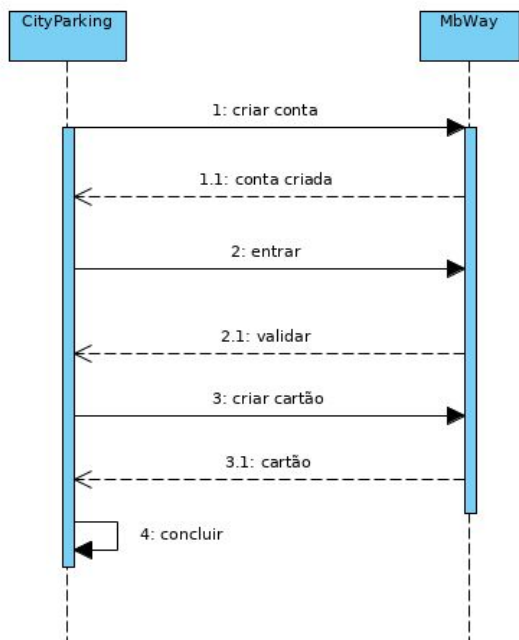


Diagrama 9: Comunicação da aplicação CityParking com MbWay

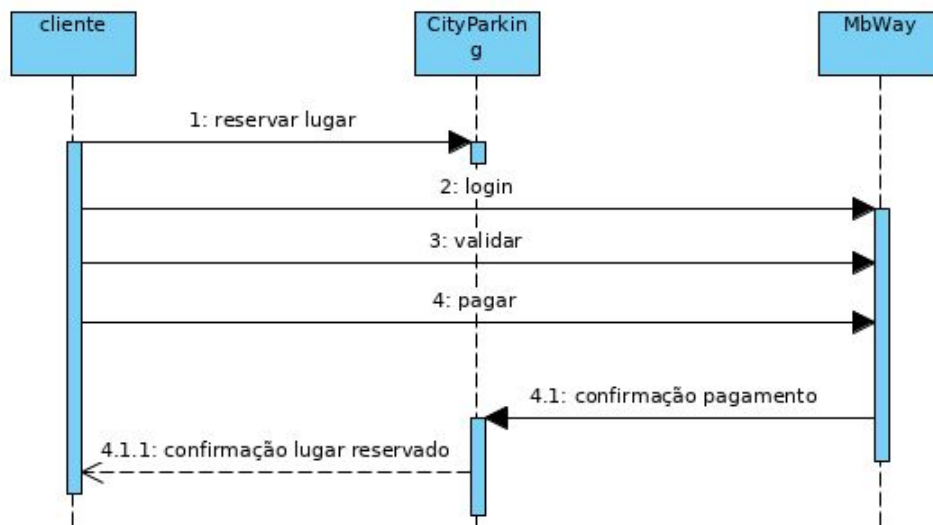


Diagrama 10: Comunicação com MbWay



## 5 Tecnologias e implementação

Para o desenvolvimento da aplicação móvel está a ser utilizado Android Studio, juntando Java e algumas funções de JavaScript. Sendo que o protótipo inicial não será lançado para IOS, Android Studio é a melhor aplicação para desenvolver a aplicação em mente.

Android Studio tira partido do facto de ser baseado em IntelliJ para se tornar num poderoso editor de código e compilador de aplicações de Android, com uma interface apelativa e várias ferramentas esta aplicação torna-se a melhor escolha para o desenvolvimento da aplicação móvel.

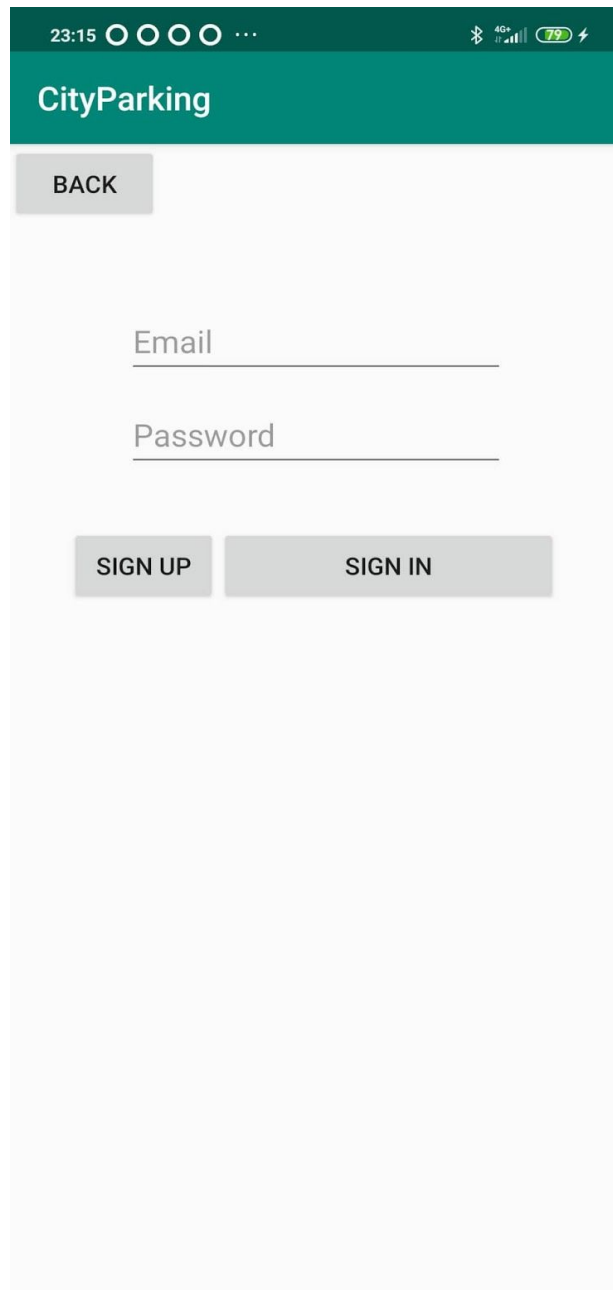
Nesta primeira abordagem está a ser utilizado Firebase como plataforma da aplicação. Tendo como propósito ser uma base de dados e API, a Firebase tira partido de ser uma infraestrutura da Google para dar garantia de qualidade. Visto ter funcionalidades como autenticação, hosting, realtime database a escolha com o objetivo de rapidez e facilidade de trabalho ficou óbvia.

Ambas as tecnologias apresentadas anteriormente funcionam bem juntas pelo que a implementação está segura.

Por fim para construir o website, sendo este de exposição, apenas foi necessário usar as tecnologias básicas de web development, tais como HTML5, CSS e Bootstrap para um template inicial.

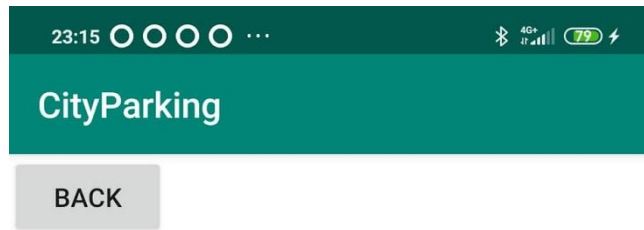
Todo o código desenvolvido pelo grupo assim como o website está alocado na plataforma GitHub no seguinte repositório [https://github.com/FJ-RATO/AMS\\_Project.github.io](https://github.com/FJ-RATO/AMS_Project.github.io).

## 6 Protótipos dos produtos



The image shows a mobile application interface for 'CityParking'. At the top, there is a dark green header bar with the text 'CityParking' in white. Below the header, there is a light gray background. At the top left of this background, there is a 'BACK' button. In the center, there are two input fields: 'Email' and 'Password', each with a horizontal line for text entry. Below these fields, there are two buttons: 'SIGN UP' and 'SIGN IN', both in a light gray box. The top of the screen shows a status bar with the time '23:15', four circles, and icons for Bluetooth, 4G+, signal strength, and battery level (79%).

Figura 1: Login screen

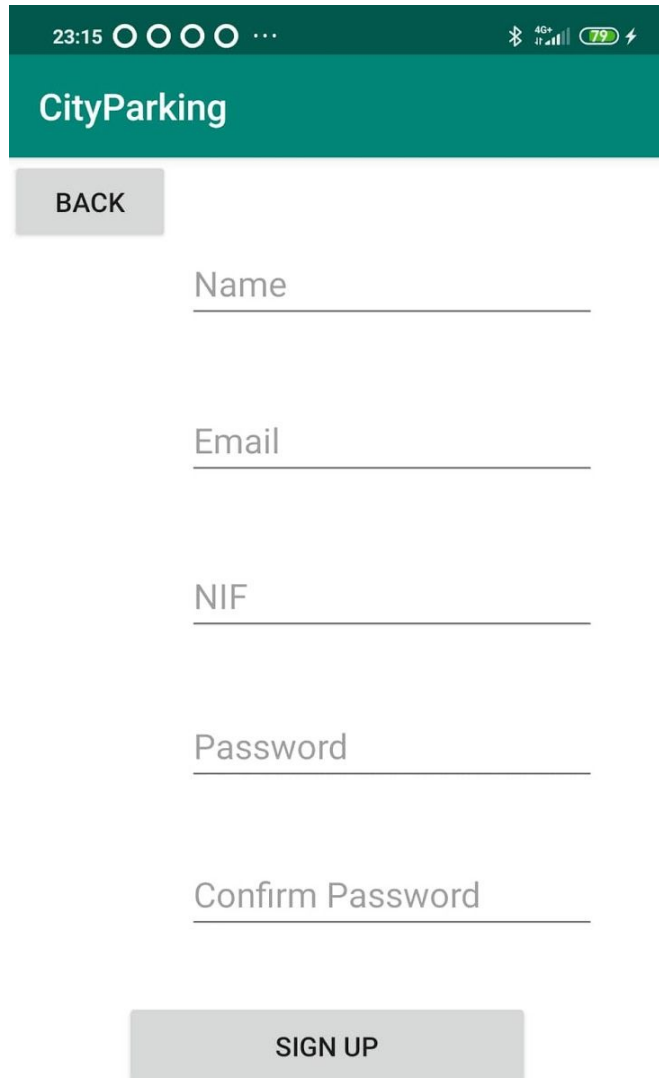


Name: Not logged in

NIF: Not logged in

Email: Not logged in

Figura 2: User not logged In



The image is a mobile app mockup for a sign-up screen. At the top, there is a dark green header bar with the text "CityParking" in white. Below the header, there is a light gray button labeled "BACK". The main area of the screen contains five text input fields, each with a label above it: "Name", "Email", "NIF", "Password", and "Confirm Password". At the bottom of the form, there is a large light gray button labeled "SIGN UP". The top of the screen shows a status bar with the time "23:15", four white circles, and icons for Bluetooth, 4G+ signal, and a battery level of 79%.

23:15 ○ ○ ○ ○ ... 4G+ 79%

**CityParking**

BACK

Name

Email

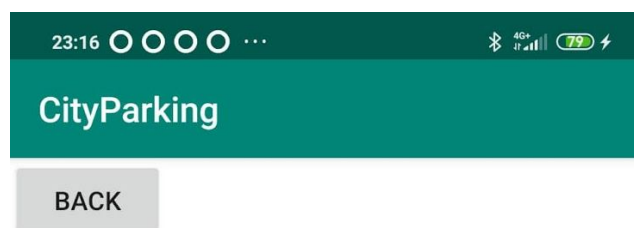
NIF

Password

Confirm Password

SIGN UP

Figura 3 : Sign Up screen



Name: Daniel Correia

NIF: 123456789

Email: daniel@ua.pt

Figura 4 : User logged in

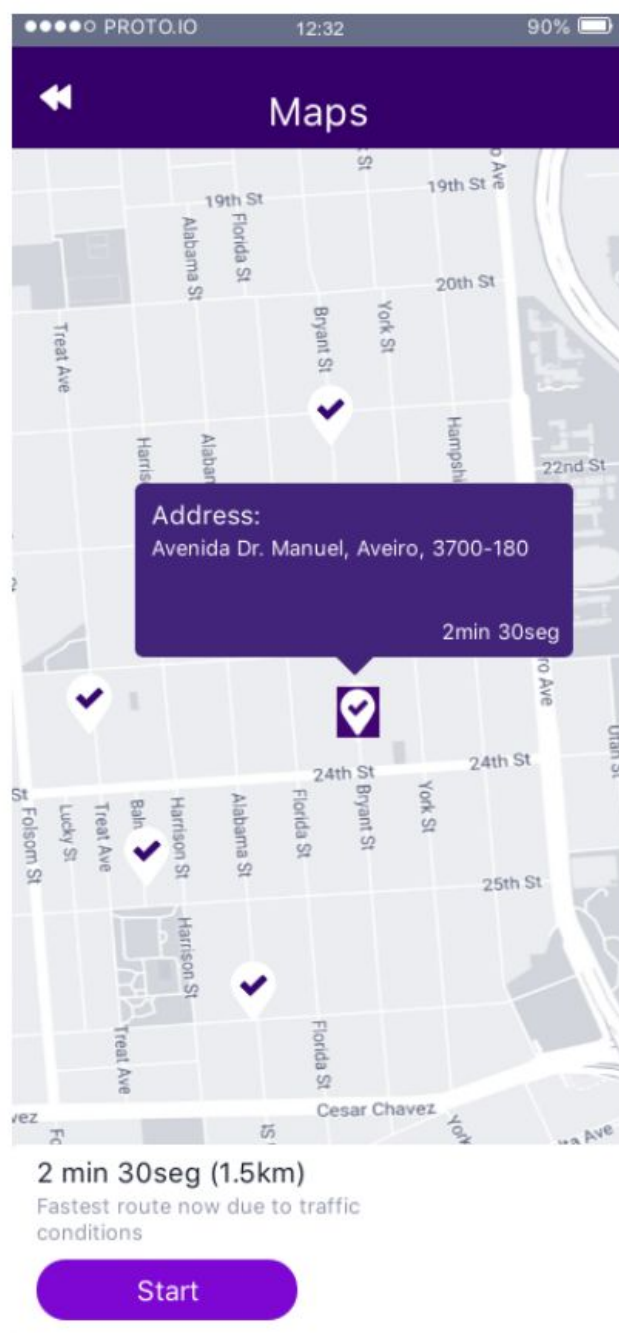


Figura 5 : Protótipo do funcionamento do mapa

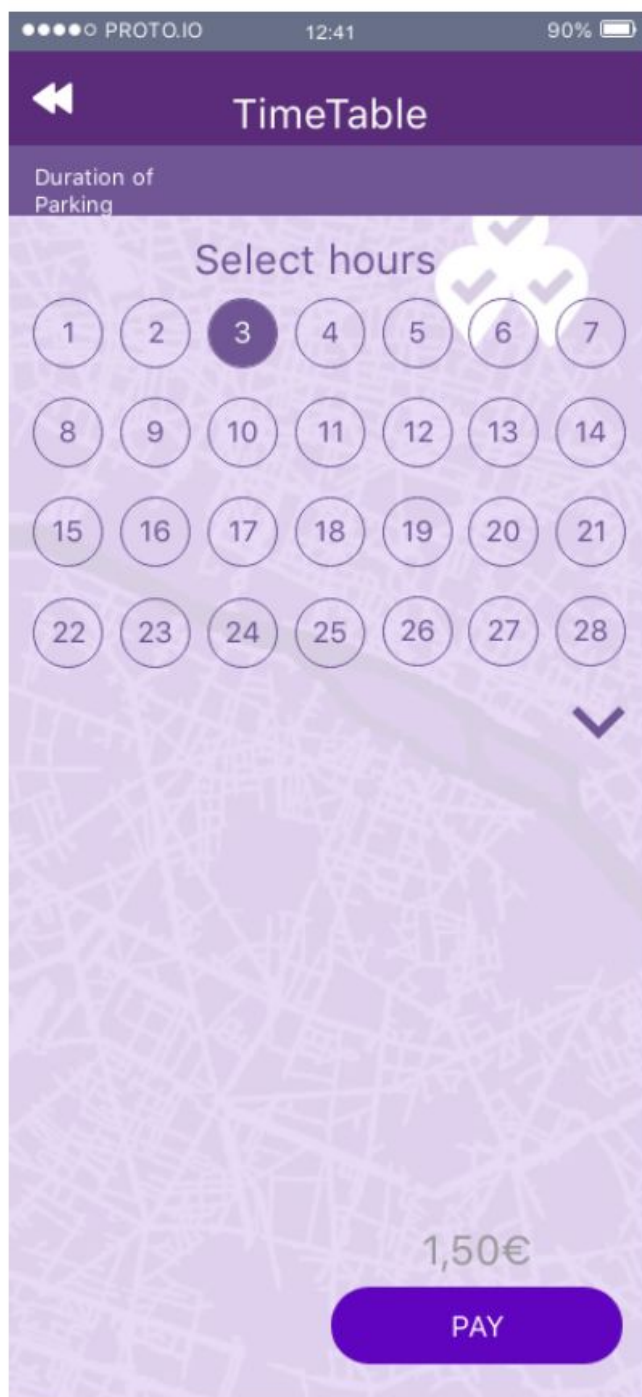


Figura 6 : Protótipo da escolha de horas de estacionamento

## 7 Anexos

Diagramas de Deployment - <http://www.dca.fee.unicamp.br/~gudwin/ftp/ea976/CompDepl.pdf>

Firebase-[https://firebase.google.com/?gclid=Cj0KCOiAq97uBRCwARIsADTziyYPW0voXTc2rAWG9SAZOgSZ6daLKr1D9OgkW-CnERUI2zq9rkBzuygaArheEALw\\_wcB](https://firebase.google.com/?gclid=Cj0KCOiAq97uBRCwARIsADTziyYPW0voXTc2rAWG9SAZOgSZ6daLKr1D9OgkW-CnERUI2zq9rkBzuygaArheEALw_wcB)

AndroidStudio-[https://developer.android.com/studio/?gclid=Cj0KCOiAq97uBRCwARIsADTziyYWaFkJtaWnlUOLHD36rsB\\_SBCbJNpMqKDhuTV4awjxG02x\\_d\\_PYTlaAugaEALw\\_wcB](https://developer.android.com/studio/?gclid=Cj0KCOiAq97uBRCwARIsADTziyYWaFkJtaWnlUOLHD36rsB_SBCbJNpMqKDhuTV4awjxG02x_d_PYTlaAugaEALw_wcB)