

CMU

**Chargify**

Software Design Document

Name (s): Micael Dias

Student No: 8200383

Date: 15/02/2023

## TABLE OF CONTENTS

<b>1. INTRODUCTION</b>	<b>2</b>
1.1 Purpose	2
1.2 Scope	2
1.3 Overview	2
<b>2. SYSTEM ARCHITECTURE</b>	<b>2</b>
2.1 Architectural Design	2
<b>3. DATA DESIGN</b>	<b>3</b>
3.1 Room Data Description	3
3.2 Firestore Data Description	3
<b>4. HUMAN INTERFACE DESIGN</b>	<b>3</b>
4.1 Overview of User Interface	3

## 1. INTRODUCTION

### 1.1 Purpose

O propósito deste documento de Design de Software (SDD) é descrever a arquitetura e o design do sistema do Chargify, que visa ajudar os utilizadores de veículos elétricos a encontrar e salvar o uso de postos de carregamento elétrico. O documento tem como público-alvo o regente da unidade curricular de Computação Model e Ubíqua.

### 1.2 Scope

O software descrito neste SDD tem como objetivo fornecer aos utilizadores de veículos elétricos uma ferramenta útil para localizar postos de carregamento elétrico próximos. A aplicação utiliza APIs abertas, como [openchargemap.org](http://openchargemap.org), para obter uma lista de postos de carregamento nas proximidades do utilizador. Além disso, a aplicação oferece outras funcionalidades úteis, como guardar pontos de carregamento favoritos, guardar o uso dos postos de carregamento numa linha de temporal e permitir a classificação dos postos de carregamento.

### 1.3 Overview

Este documento descreve a arquitetura e o design da aplicação, bem como as tecnologias utilizadas na sua implementação. O documento também descreve os requisitos funcionais e não funcionais, bem como os casos de uso e as interfaces do utilizador.

## 2. SYSTEM ARCHITECTURE

### 2.1 Architectural Design

O Chargify segue um padrão de arquitetura MVVM (Model-View-ViewModel), no qual os screens da aplicação (Jetpack Compose) são conectados aos seus respetivos ViewModels, que por sua vez se comunicam com os modelos de dados (Models) para recuperar e persistir dados. O design também utiliza componentes reutilizáveis, como aqueles encontrados na pasta `components` de modo a manter a consistência. Os dados são persistidos no Firestore tanto como no Room (Caching) para que possa haver um melhor desempenho e escalabilidade. Em resumo, o design inclui as seguintes camadas: UI, ViewModels, Entities, Services, e Repositories.

## 3. DATA DESIGN

### 3.1 Room Data Description

O sistema no Room será composto de duas entidades principais de dados, a Entidade de Estação (EvStationEntity) e a Entidade de Conexão (ConnectionEntity). A Entidade de Estação armazenará informações sobre as próprias estações, enquanto a Entidade de Conexão armazenará informações sobre os tipos de conexões disponíveis em cada estação.

Essas entidades serão relacionadas por meio de um relacionamento um-para-muitos, onde uma Estação pode ter várias Conexões, mas uma Conexão só pode pertencer a uma Estação. Para armazenar essas informações no banco de dados, usaremos a classe de dados "EvStationWithConnections", que armazena a entidade de estação e uma lista de conexões.

### 3.2 Firestore Data Description

O sistema do Firestore armazena informações sobre estações de carregamento de veículos elétricos e seu status, bem como as avaliações e classificações dos utilizadores e as informações da última localização conhecida. Os dados são armazenados em um banco de dados não-relacional do Firestore, do Google Cloud Platform. Existem três principais entidades no sistema:

- **BookmarkedEvStation:** informações sobre as estações de carregamento de veículos elétricos marcadas como favoritas, incluindo identificador de estação, informações do operador, informações de endereço, latitude e longitude.
- **EvStationRating:** informações sobre as avaliações e classificações das estações de carregamento de veículos elétricos, incluindo o identificador de estação, a média de classificação e o número de classificações.
- **LastKnownEvStation:** informações sobre a última localização conhecida das estações de carregamento de veículos elétricos, incluindo o identificador de estação, o horário da última atualização e informações de endereço.

## 4. HUMAN INTERFACE DESIGN

### 4.1 Overview of User Interface

A interface de utilizador do Chargify foi desenvolvida para ser intuitiva e fácil de utilizar, permitindo que os utilizadores localizem facilmente os postos de carregamento elétrico nas proximidades e avaliem a qualidade desses postos. A interface de utilizador foi projetada com base nas melhores práticas de design referidas pela Google na sua versão mais recente do Material 3. O Chargify foi desenvolvido para ser compatível com uma ampla variedade de

dispositivos Android, tablet ou smartphone, para que os utilizadores possam utilizar a aplicação em qualquer lugar e a qualquer momento.

A interface de utilizador é composta por vários ecrãs, cada uma com uma funcionalidade específica. Na tela principal, os utilizadores podem visualizar uma lista com os postos de carregamento elétrico nas proximidades, abaixo de cada posto de carregamento pode-se encontrar as seguintes informações: operacionalidade e avaliação do equipamento, número de carregadores que possui e respetivos tipos e a distancia da localização atual.

Ainda no ecrã principal o Chargify permite aos utilizadores classificar os postos de carregamento com base na sua experiência, navegar para a estação de carregamento, partilhar a estação através do ecrã de partilha nativo do Android, ligar para o operador responsável pelo posto de carregamento e guardar a estação.