

**UNIVERSIDADE FERNANDO PESSOA**



**LABORATÓRIO DE PROJETO INTEGRADO**

**IMOWOW**

## **Documento de Especificação de Requisitos de Software + Relatório de Projeto**

**Aluno:** Jorge Silva - 36448

**Docente:** Professor Paulo Rurato

**Docente Orientador:** Professor Ivo Pereira

**PORTO, JUNHO DE 2021**

# Índice

<b>1 Introdução.....</b>	<b>3</b>
1.1 Objetivo.....	3
1.1.1 Motivação.....	3
1.2 Perfil de Utilizador.....	3
1.3 Perspetiva do Produto.....	3
<b>2 Requisitos.....</b>	<b>4</b>
2.1 Requisitos Funcionais.....	4
2.2 Requisitos Não Funcionais.....	7
<b>3 Planeamento.....</b>	<b>9</b>
3.1 Tarefas e Diagrama de Gantt.....	9
3.1.1 Reuniões com Orientador.....	11
3.2 Recursos e Ferramentas.....	12
<b>4 Modelação.....</b>	<b>13</b>
4.1 Casos de Uso.....	13
4.2 Diagrama de Classes.....	15
4.3 Modelo ER.....	19
<b>5 Arquitetura Global.....</b>	<b>23</b>
<b>6 Conclusão.....</b>	<b>24</b>
<b>7 Referências.....</b>	<b>25</b>

## Índice de figuras

<u>Figura 1: Diagrama de Gantt .....</u>	<u>10</u>
<u>Figura 2: Diagrama de casos de uso.....</u>	<u>13</u>
<u>Figura 3: Diagrama de classes .....</u>	<u>15</u>
<u>Figura 4: Modelo ER.....</u>	<u>19</u>
<u>Figura 5: Modelo MVC .....</u>	<u>23</u>

## Índice de tabelas

<u>Tabela 1: Tarefas realizadas.....</u>	<u>10</u>
--	-----------

# 1. Introdução

## 1.1 Objetivo

Neste projeto pretende-se desenvolver um portal centralizador de preços médios de venda e de arrendamento para apoio à decisão de investimento no mercado imobiliário, ao qual lhe chamei de imowow.

### 1.1.1 Motivação

A razão pela qual escolhi este projeto foi pelo facto de me parecer o mais interessante em relação aos outros que me foram propostos, uma vez que tenho parentes que estão envolvidos nesta área de negócio e que, de uma forma indireta, me influenciaram a fazer esta escolha. O facto de não conhecer nenhuma aplicação que fizesse algo parecido a este tipo de serviço, foi outra razão que determinou a minha escolha. Com o passar do tempo e fruto de alguma pesquisa, observei que em diversos países este tipo de aplicações já existe e é bastante procurada, no entanto em Portugal apenas existe uma e o seu serviço é pago e são raras as pessoas que sabem da sua existência.

## 1.2 Perfil do Utilizador

A aplicação está desenhada para apenas 1 perfil de utilização.

O utilizador tem como função registar, através de um formulário, os dados de um possível negócio a realizar e consultar o preço médio por metro quadrado de cada região específica, assim como o histórico de preços praticados em meses anteriores nessa mesma região.

## 1.3 Perspetiva do produto

### 1.3.1 Interface do utilizador

A interface do utilizador fornece menus, barras de ferramentas, botões, painéis, e conteúdos que permitem um controlo mais facilitado a partir do teclado e do rato.

## 2. Requisitos

### 2.1 Requisitos Funcionais

#### [RF-01]

**Título:** A aplicação vai permitir que cada cliente crie uma conta de utilizador, com o seu email e uma palavra passe.

**Descrição:** Um cliente que queira utilizar os serviços efetuados pela aplicação terá de realizar a autenticação onde irá inserir o seu nome de utilizador, o seu email e uma palavra passe associada e essa mesma conta.

#### [RF-02]

**Título:** A aplicação deve permitir que o utilizador insira informação (preço, área, estado do imóvel, tipo de topologia, concelho, se tem piscina, garagem, jardim e elevador) num formulário sobre o possível negócio de venda de um imobiliário a analisar.

**Descrição:** O utilizador insere informações num formulário que contem “área”, “estado do imóvel”, “tipo de topologia”, “concelho”, se tem “piscina”, “garagem”, “jardim”, e “elevador” e, por último, o “preço”. Preço esse como sendo o preço pedido pelo referido imóvel, área como sendo os metros quadrados que o imóvel tem, e quanto ao estado, o utilizador pode escolher entre “Novo”, “Usado”, “Renovado” ou “Por Recuperar”. Quanto à topologia, é possível inserir “T1”, “T1+1”, “T2”, “T2+1”, “T3”, “T3+1”, “T4”. Posto isto, faz a escolha do concelho onde este imóvel se encontra e, de seguida, indica se “sim” ou “não” ao facto do imóvel ter piscina, garagem jardim e elevador. Por último, seleciona a opção de venda.

#### [RF-03]

**Título:** A aplicação deve conseguir calcular o preço do imóvel para negócio de venda com base no preço médio de venda por metro quadrado do concelho com o preço de metro quadrado do seu possível imóvel.

**Descrição:** O utilizador após preencher o formulário e dizer que se trata de um negócio de venda, a aplicação irá apresentar a sua própria avaliação do imóvel através de um cálculo feito com base no preço por metro quadrado daquele concelho e com base na

área apresentada. Esse cálculo tem também em conta os outros inseridos no formulário acrescentando uma percentagem diferente para cada parâmetro associado.

#### [RF-04]

**Título:** A aplicação deve conseguir avaliar o negócio de venda com base no preço efetuado pela nossa avaliação e com o preço pedido pelo imóvel.

**Descrição:** Após ter sido determinado pela aplicação o valor do imóvel, este consegue determinar se o negócio é “Muito Bom”, “Bom”, “Suficiente”, “Mau” e “Muito Mau”, através de um cálculo efetuado entre o preço de venda pedido do imóvel (dado tem de ser inserido no formulário) e preço efetuado pela aplicação.

#### [RF-05]

**Título:** A aplicação deve permitir ao utilizador inserir informação (preço do arrendamento, área, tipo de topologia, concelho, se tem piscina, garagem, jardim, entre outras características) num formulário sobre o possível negócio de arrendamento de um imobiliário a analisar.

**Descrição:** O utilizador insere informações num formulário que contem “área”, “estado do imóvel”, “tipo de topologia”, “concelho”, se tem “piscina”, “garagem”, “jardim”, e “elevador” e, por último, o “preço”. Preço esse como sendo o preço pedido pelo referido imóvel, área como sendo os metros quadrados que o imóvel tem, e quanto ao estado, o utilizador pode escolher entre “Novo”, “Usado”, “Renovado” ou “Por Recuperar”. Quanto à topologia, é possível inserir “T1”, “T1+1”, “T2”, “T2+1”, “T3”, “T3+1”, “T4”. Posto isto, faz a escolha do concelho onde este imóvel se encontra e, de seguida, indica se “sim” ou “não” ao facto do imóvel ter piscina, garagem jardim e elevador. Por último, seleciona a opção de arrendamento.

#### [RF-06]

**Título:** A aplicação deve conseguir calcular o preço do imóvel para negócio de arrendamento com base no preço médio de arrendamento por metro quadrado do concelho com o preço de metro quadrado do seu possível imóvel.

**Descrição:** O utilizador após preencher o formulário e dizer que se trata de um negócio de arrendamento, a aplicação irá apresentar a sua própria avaliação do imóvel através de um cálculo feito com base no preço por metro quadrado daquele concelho e com base

na área apresentada, também esse cálculo tem em conta os outros inseridos no formulário acrescentando uma percentagem diferente para cada parâmetro associado.

#### [RF-07]

**Título:** A aplicação deve conseguir avaliar o negócio de arrendamento com base no preço efetuado pela nossa avaliação e com o preço pedido pelo imóvel.

**Descrição:** Após ter sido determinado pela aplicação o valor do imóvel, este consegue determinar se o negócio é “Muito Bom”, “Bom”, “Suficiente”, “Mau” e “Muito Mau”, através de um cálculo efetuado entre o preço de arrendamento pedido do imóvel (dado tem de ser inserido no formulário) e preço efetuado pela aplicação.

#### [RF-08]

**Título:** A aplicação deve mostrar o custo médio por metro quadrado para o negócio de venda de cada região em específico.

**Descrição:** O utilizador será capaz de consultar o preço médio por metro quadrado do negócio de venda de um concelho em específico.

#### [RF-09]

**Título:** A aplicação deve mostrar o custo médio por metro quadrado para o negócio de arrendamento de cada região em específico.

**Descrição:** O utilizador será capaz de consultar o preço médio por metro quadrado do negócio de arrendamento de um concelho em específico.

#### [RF-10]

**Título:** A aplicação deve permitir ao utilizador inserir as suas limitações (zonas preferidas e quais são os seus limites a nível financeiro).

**Descrição:** O utilizador insere os concelhos que prefere e que pretende saber os preços atuais e anteriores praticados nessa zona, assim como o seu limite orçamental para o seu negócio de venda ou arrendamento.

#### [RF-11]

**Título:** A aplicação deve notificar o utilizador sobre quais os melhores investimentos a fazer (preferências).

**Descrição:** O utilizador ao inserir os concelhos que prefere, a aplicação irá ser capaz de dizer qual o concelho em que o preço por metro quadrado tem desvalorizado.

#### [RF-12]

**Título:** A aplicação deve guardar o histórico do preço do metro quadrado de cada concelho.

**Descrição:** A aplicação irá mostrar quais os preços do metro quadrado de cada concelho, praticados em meses anteriores. Estes dados irão ser atualizados mensalmente.

## 2.2 Requisitos Não Funcionais

#### [RNF-01]

**Título:** A aplicação vai apenas prestar todos os seus serviços se o utilizador tiver conta registada e sessão iniciada na mesma.

**Descrição:** Para o utilizador consultar a avaliação do seu imóvel ou consultar os preços por metro quadrado atuais ou histórico de preços de cada concelho, este terá de iniciar sessão na sua conta de utilizador ou, se este ainda não estiver registado, terá de efetuar o registo.

#### [RNF-02]

**Título:** A aplicação deverá poder ser configurada para trabalhar com o tipo de negócio de venda.

**Descrição:** O utilizador ao querer efetuar a avaliação do seu imóvel, a aplicação terá de estar preparada para fazer essa avaliação para o tipo de negócio de venda.

#### [RNF-03]

**Título:** A aplicação deverá poder ser configurada para trabalhar com o tipo de negócio de arrendamento.

**Descrição:** O utilizador ao querer efetuar a avaliação do seu imóvel, a aplicação terá de estar preparada para fazer essa avaliação para o tipo de negócio de arrendamento.

#### [RNF-04]

**Título:** A aplicação deverá fazer uma utilização racional do espaço da base de dados.

**Descrição:** A aplicação terá os seus dados guardados numa base de dados e todos esses dados estão em tabelas que garantem o uso de chaves primárias, sendo que em algumas tabelas, existe o uso de chaves primárias e, para facilitar a sua gestão, também usam chaves estrangeiras.

#### [RNF-05]

**Título:** A aplicação irá apresentar uma interface simples e limpa de modo a que cada utilizador consiga utilizar a mesma sem qualquer tipo de problema.

**Descrição:** A aplicação irá apresentar as suas diferentes funcionalidades, de um modo simples e direto, de tal forma que qualquer utilizador seja capaz de compreender o que cada uma é capaz de fazer, sem qualquer dificuldade.

#### [RNF-06]

**Título:** A aplicação irá assegurar a privacidade acerca de todos os dados fornecidos pelo utilizador.

**Descrição:** Todos os dados que o utilizador irá disponibilizar para fazer uso da aplicação serão intransmissíveis e pessoais. Os mesmos não serão expostos para outros utilizadores.

#### [RNF-07]

**Título:** A aplicação estará sempre disponível a ser utilizada.

**Descrição:** A aplicação irá estar sempre disponível para efetuar todas as suas funcionalidades a qualquer hora e a qualquer dia (exceto por motivos alheios à aplicação).

#### [RNF-08]

**Título:** A aplicação irá assegurar um desempenho rápido e fluído.

**Descrição:** Quando o utilizador efetuar qualquer tipo de operação na aplicação, essa operação irá ser efetuada e apresentada de forma imediata.



### 3. Planeamento

#### 3.1 Tarefas e Diagrama de Gantt

Nome da tarefa	Início	Fim
<b>- Comunicação</b>	08/02/2021	15/02/2021
Introdução ao projeto	08/02/2021	10/02/2021
Elicitação de Requisitos	10/02/2021	13/02/2021
<b>- Planeamento</b>	15/02/2021	23/03/2021
Previsão do programa	15/02/2021	17/02/2021
Elaboração de requisitos	18/02/2021	27/02/2021
Elaboração de casos de Uso	01/03/2021	06/03/2021
Seleção de tarefas	08/03/2021	13/03/2021
Elaboração do diagrama de classes	15/03/2021	23/03/2021
<b>- Modelação</b>	24/03/2021	10/04/2021
Análise do projeto	23/03/2021	27/04/2021
Elaboração do modelo ER	29/03/2021	03/04/2021
Desenho	05/04/2021	10/04/2021
<b>- Desenvolvimento</b>	12/04/2021	05/06/2021
<b>Codificação</b>	12/04/2021	05/06/2021
Desenvolvimento de código	19/04/2021	05/06/2021
Testes	10/05/2021	01/06/2021
<b>Implementação</b>	01/06/2021	05/06/2021
Check-Up Final	04/06/2021	06/06/2021
<b>- Apresentação</b>	07/06/2021	07/06/2021

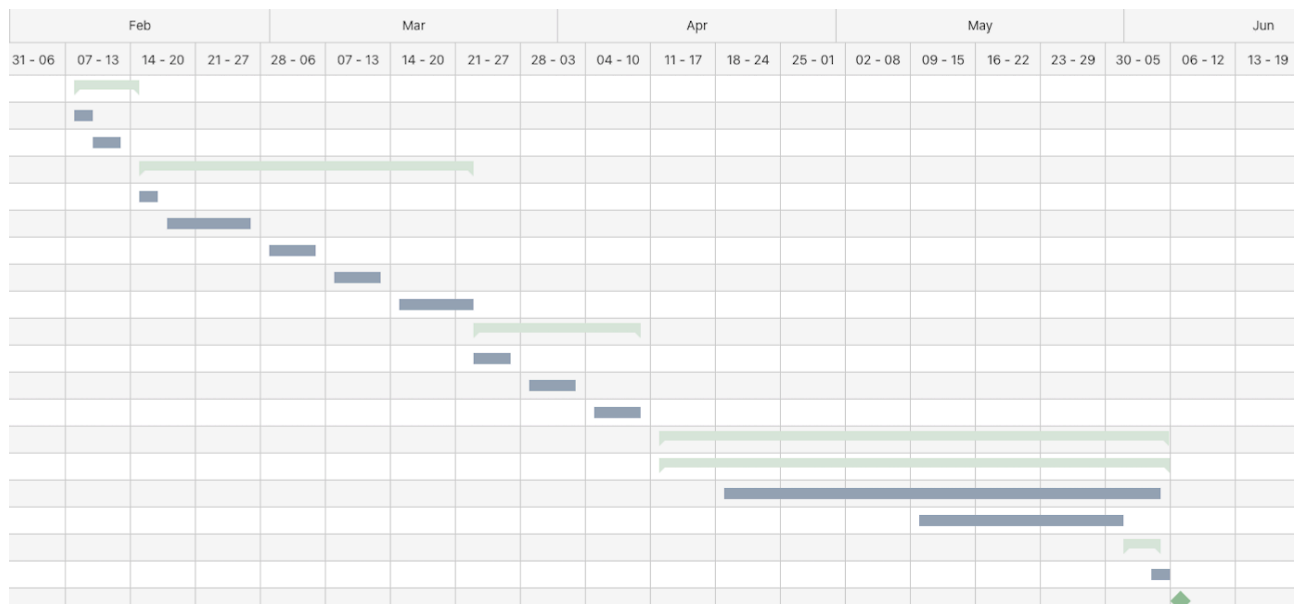


FIGURA 1 - DIAGRAMA DE GANTT

As tarefas referidas anteriormente são as principais tarefas planeadas, com uma tentativa de previsão da sua duração e enquadramento nos prazos existentes. É certo que poderiam ser apresentadas em algumas sub tarefas, no entanto, outras tarefas regulares, programadas ou não, como reuniões com o professor orientador Ivo Pereira ou com o professor Paulo Rurato, não são apresentadas, sendo encaradas como sub tarefas de várias tarefas mencionadas na tabela acima.

É importante referir que, não obstante a linearização deste planeamento e clara definição de tarefas predecessoras, haverá sempre flexibilidade para retrocessos a tarefas anteriores, quando se justifiquem alterações nos requisitos, planeamento ou implementação.

### 3.1.1 Reuniões com Orientador (Professor Ivo Pereira)

**Reunião do dia 10/02/2021** - Nesta reunião foi efetuada uma introdução ao projeto.

**Reunião do dia 18/02/2021** - Nesta reunião foi para falar sobre os possíveis requisitos funcionais e não funcionais, e seus casos de uso a apresentar. Ficou planeado entregar requisitos funcionais e não funcionais e casos de uso na próxima reunião.

**Reunião do dia 05/03/2021** - Nesta reunião foi feita uma análise aos casos de uso elaborados e iniciámos também a discussão sobre a linguagem de programação a usar neste projeto.

**Reunião do dia 18/03/2021** - Nesta reunião foi tomada a decisão da linguagem de programação a utilizar e foi também discutido o diagrama de classes nesta fase de planeamento.

**Reunião do dia 16/04/2021** - Nesta reunião ficou decidido qual base de dados a utilizar neste projeto e a constituição da mesma.

**Reunião do dia 23/04/2021** - Nesta reunião serviu como auxílio da montagem da base de dados em MySQL e mesma verificação.

**Reunião do dia 28/04/2021** - Nesta reunião fui alertado sobre as minhas dificuldades com a linguagem Python e o tempo que me restava para a elaboração do projeto e, por isso, foi tomada a decisão de mudar de linguagem. A nova linguagem escolhida foi o Java.

**Reunião do dia 07/05/2021** - Nesta reunião serviu apenas para esclarecimento de algumas dúvidas em relação ao mapeamento das minhas classes e dos seus atributos.

## 3.2 Recursos e Ferramentas

Backend:

- **Python**
  - *Framework*: Django
  - *Base de dados*: MySQL
- **JAVA**
  - *Framework*: Spring
  - *Base de dados*: H2 Database Engine

Frontend:

- **JavaScript**
- **HTML + CSS**

No momento em que fiz a escolha deste projeto, foi acordado com o professor Ivo Pereira que tinha de realizar a aplicação numa linguagem à minha escolha, desde que fosse uma linguagem orientada ao objeto. Desta forma, eu optei por escolher o Python, linguagem que desconhecia por completo e que me obrigava a aprender por mim mesmo uma linguagem nova. Usei como IDE o PyCharm da JetBrains. Para a construção e manipulação dos meus dados usei a base de dados denominada MySQL.

Com o passar do tempo o professor Ivo alertou-me para o facto de estar a atrasado, devido ao meu desconhecimento sobre a nova linguagem que estava a demorar a aprender, aconselhando-me a mudar para uma linguagem que me sentisse mais confortável a utilizar e que tivesse mais experiência. Optei então por mudar para a linguagem JAVA, usando o IDE IntelliJ IDE CE da JetBrains, onde já tinha realizado alguns projetos noutras unidades curriculares lecionadas. Escolhi a framework chamada Spring e a base de dados denominada por H2 Database Engine.

No desenvolvimento do meu projeto, na parte do frontend, optei por utilizar uma linguagem chamada JavaScript. Importa referir que não era, de todo, uma linguagem confortável para mim, mas com o passar do tempo adaptei-me bastante bem. No que concerne à parte visual do meu projeto, utilizei o HTML e CSS para a gestão das imagens apresentadas.

## 4. Modelação

### 4.1 Casos de Uso

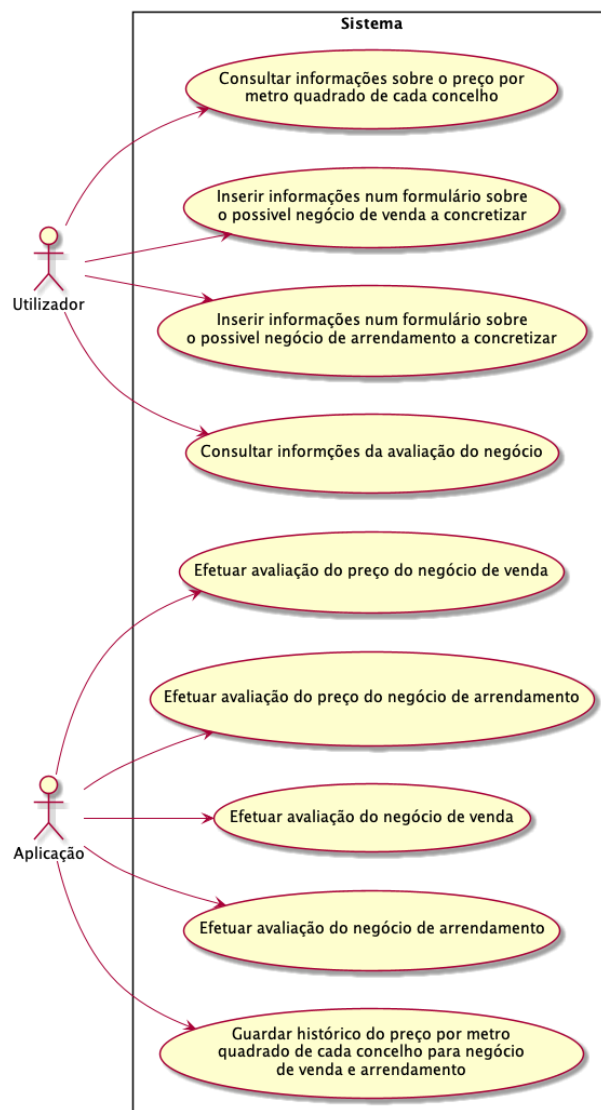


FIGURA 2 - DIAGRAMA DE CASOS DE USO

Através da observação da figura 2, representada acima, é possível observar os diferentes casos de uso. Estão representados dois atores: o utilizador e a aplicação.

O utilizador será capaz de consultar informações sobre o preço por metro quadrado de cada concelho, quer para o negócio de venda, quer para arrendamento. Este insere informação sobre o seu imóvel num formulário, que contem informações como: o concelho onde o imóvel se localiza, metros quadrados do imóvel, o seu estado (“Novo”, “Usado”, “Renovado”, “Por Recuperar”), o ano de construção, a sua topologia (“T1”, “T1+1”, “T2”,

“T2+1”, “T3”, “T3+1”, “T4”) e o possível negócio de venda ou arrendamento a concretizar. Após inserir os dados sobre o seu imóvel e de seleccionar a opção em que diz se o negócio se trata de uma venda ou de um arrendamento, o utilizador pode consultar o preço (em euros) da avaliação feita pela aplicação e a avaliação, relação entre preço pedido pelo imóvel e pela avaliação em euros determinada pela aplicação.

A aplicação, por sua vez, vai efetuar a avaliação do preço do imóvel para os dois tipos de negócio, venda ou arrendamento, através de umas percentagens que estão associadas aos tipos de dados inseridos pelo utilizador no formulário. Também irá fazer uma avaliação em relação ao preço pedido pelo imóvel e pelo preço calculado pela aplicação, essa avaliação pode ser: “Muito Mau”, “Mau”, “Insuficiente”, “Bom”, “Muito Bom”. Guarda também histórico de preços por metro quadrado de cada concelho. Esse histórico deverá ser atualizado mês a mês, assim como o preço atual do metro quadrado de cada concelho.

## 4.2 Diagrama de Classes

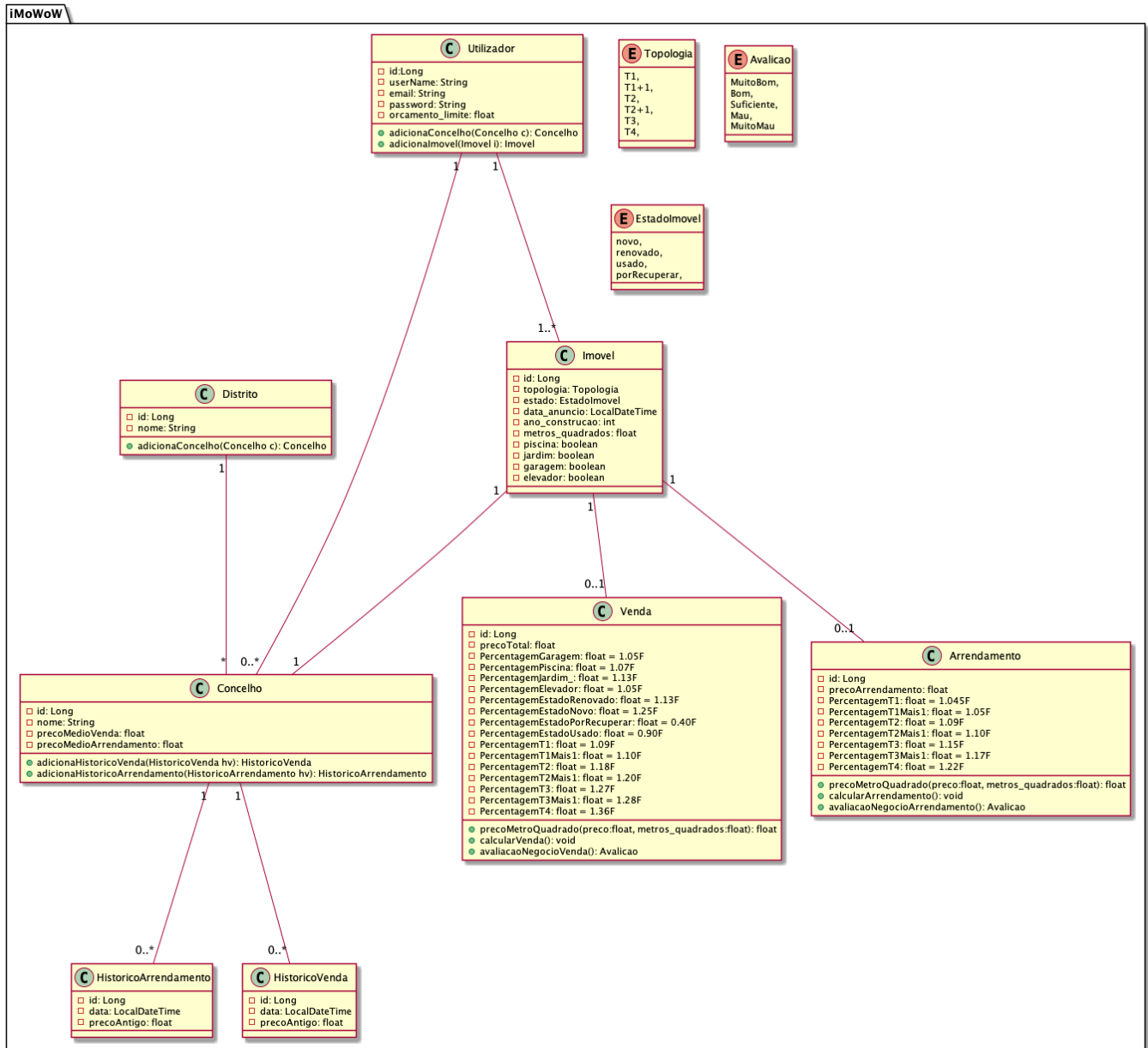


FIGURA 3 - DIAGRAMA DE CLASSES

Na figura 3, representada acima, podemos ver o diagrama de classes utilizado para o desenvolvimento deste projeto. É possível verificar que existem três classes que são do tipo enumerado: a classe Topologia, que representa os diferentes tipos de topologia (“T1”, “T1+1”, “T2”, “T2+1”, “T3”, “T3+1”, “T4”) presentes num imóvel (não foram consideradas mais topologias pois estas são as mais comuns num imóvel); A classe EstadoImovel, que representa os diferentes tipos de estados em que um imóvel se pode encontrar (“Novo”, “Usado”, “Renovado” ou “Por Recuperar”); E a classe Avaliação, que é responsável por retornar para o utilizador o tipo de avaliação feito pela aplicação. Essa avaliação pode ser “Muito Bom”, “Bom”, “Suficiente”, “Mau” e “Muito Mau”.

A classe Utilizador tem como atributos um id do tipo Long, que é um identificador único de cada utilizador; um userName que é do tipo String e que corresponde ao nome do utilizador; um email que é do tipo String que corresponde ao email da conta criada pelo utilizador; uma password que também é do tipo String e que irá corresponder à palavra passe associada à conta do utilizador, que deverá ser secreta e nunca vista por outros utilizadores. O utilizador tem uma relação com Imovel, onde um utilizador tem um ou mais imóveis e um imóvel tem um utilizador. O utilizador também tem uma relação com Concelho, onde apenas um utilizador tem vários concelhos, concelhos esses que são para fazer referência aos concelhos preferidos de cada utilizador onde este pode acompanhar através de uma notificação os preços atuais, por metro quadrado, desses mesmos concelhos. Tem como funções o adicionaConcelho, que é responsável por adicionar um concelho ao ArrayList de concelhos presentes na sua classe e a função adicionalmovel, que será capaz de adicionar um imóvel ao seu ArrayList de imóveis presentes na sua classe.

A classe Concelho tem como atributos um id do tipo Long, que é um identificador único de cada concelho; um nome do tipo String, que corresponde ao nome desse concelho; um precoMedioVenda do tipo float, que corresponde ao preço por metro quadrado atual desse concelho para o negócio de venda; um precoMedioArrendamento do tipo float que, corresponde ao preço por metro quadrado atual desse concelho para o negócio de arrendamento. Tem relação com o Distrito, onde um distrito tem mais que um concelho e um concelho apenas tem um distrito. A classe distrito também tem um id do tipo Long, que é um identificador único de cada distrito; um nome do tipo String, que corresponde ao nome desse distrito; e uma função que é o adicionaConcelho, que é responsável por adicionar um concelho ao ArrayList de concelhos presentes na sua classe. Também tem relação com os dois tipos de Históricos (histórico de venda e histórico de arrendamento) onde um concelho tem mais do que um histórico e um histórico tem um concelho, daí existirem duas funções (adicionaHistoricoVenda e adicionaHistoricoArrendamento) que adicionam histórico ao arrayList de históricos presentes na classe Concelho.

As classes de HistoricoVenda e HistoricoArrendamento têm como atributos um id do tipo Long, que é um identificador único de cada histórico; uma data do tipo LocalDateTime, que corresponde à data em que esse histórico foi criado; e um precoAntigo do tipo float, que corresponde ao preço praticado anteriormente por esse concelho.



A classe Imovel tem como atributos um id do tipo Long, que é um identificador único de cada imóvel; uma topologia do tipo enumerado, que corresponde ao tipo de topologia presente nesse imóvel; um estado do tipo enumerado EstadoImovel, que corresponde ao tipo de estado em que o imóvel se encontra; data\_anuncio do tipo LocalDateTime, que corresponde à data em que esse imóvel foi introduzido na aplicação; ano\_construcao do tipo integer que corresponde ao ano em que esse imóvel foi construído; metros\_quadradados do tipo float que, corresponde aos metros quadrados que o imóvel tem; e por último, atributos do tipo boolean, que corresponde ao facto de o imóvel ter piscina, jardim, garagem ou elevador. Tem relação com concelho, onde um imóvel tem um concelho. Tem também relação com venda e arrendamento, onde o negócio imóvel é do tipo venda ou arrendamento.

A classe Venda tem como atributo um id do tipo Long, que é um identificador único de cada venda; e um precoTotal do tipo float, que é o preço inserido pelo utilizador que corresponde ao valor pedido pelo imóvel para o negócio de venda. Tem como funções precoPorMetroQuadrado, que através dos metros quadrados do imóvel inseridos pelo utilizador e do preço por metro quadrado atual do concelho do negócio de venda, calcula um preço base desse imóvel. Existe ainda a função calcularVenda, que é responsável por realizar a avaliação do preço de venda desse imóvel. Se o imóvel tiver garagem aumenta 5% o seu valor, se tiver piscina valoriza 7%, se tiver jardim aumenta 13% e se tiver elevador valoriza mais 5%. Em relação ao estado do imóvel, se este for novo tem uma valorização de 25%, se for renovado valoriza 13%, se estiver por recuperar desvaloriza 60% e se for usado desvaloriza 10%. Quanto à topologia, se o imóvel for T1 vai valorizar 9%, se for T1+1 valoriza 10%, se for T2 vai valorizar 18%, se for T2+1 valoriza mais 20%, se for T3 vai valorizar 27%, se for T3+1 valoriza 28% e se for um T4 este vai valorizar 36%. Por último, a função avaliaNegocioVenda efetua a avaliação do negócio de venda, comparando o preço pedido pelo imóvel que foi inserido pelo utilizador, com o preço que foi efetuado pela função de calcularVenda.

A classe arrendamento tem como atributo um id do tipo Long, que é um identificador único de cada arrendamento e um precoArrendamento do tipo float, que é o preço inserido pelo utilizador que corresponde ao valor pedido pelo imóvel para o negócio de arrendamento. Tem como funções precoPorMetroQuadrado, que através dos metros quadrados do imóvel inseridos pelo utilizador e do preço por metro quadrado atual do concelho do negócio de arrendamento calcula um preço base desse imóvel. Tem ainda a função calcularArrendamento, que é responsável por realizar a avaliação do preço de ar-

rendamento desse imóvel. Se o imóvel for T1 vai valorizar 4,5%, se for T1+1 valoriza 5%, se for T2 vai valorizar 9%, se for T2+1 valoriza mais 10%, se for T3 vai valorizar 15%, se for T3+1 valoriza 17% e se for um T4 este vai valorizar 22%. Por último, tem a função `avaliaNegocioArrendamento` que efetua a avaliação do negócio de arrendamento, comparando o preço pedido pelo imóvel que foi inserido pelo utilizador com o preço que foi efetuado pela função de `calcularVenda`.

## 4.3 Modelo ER

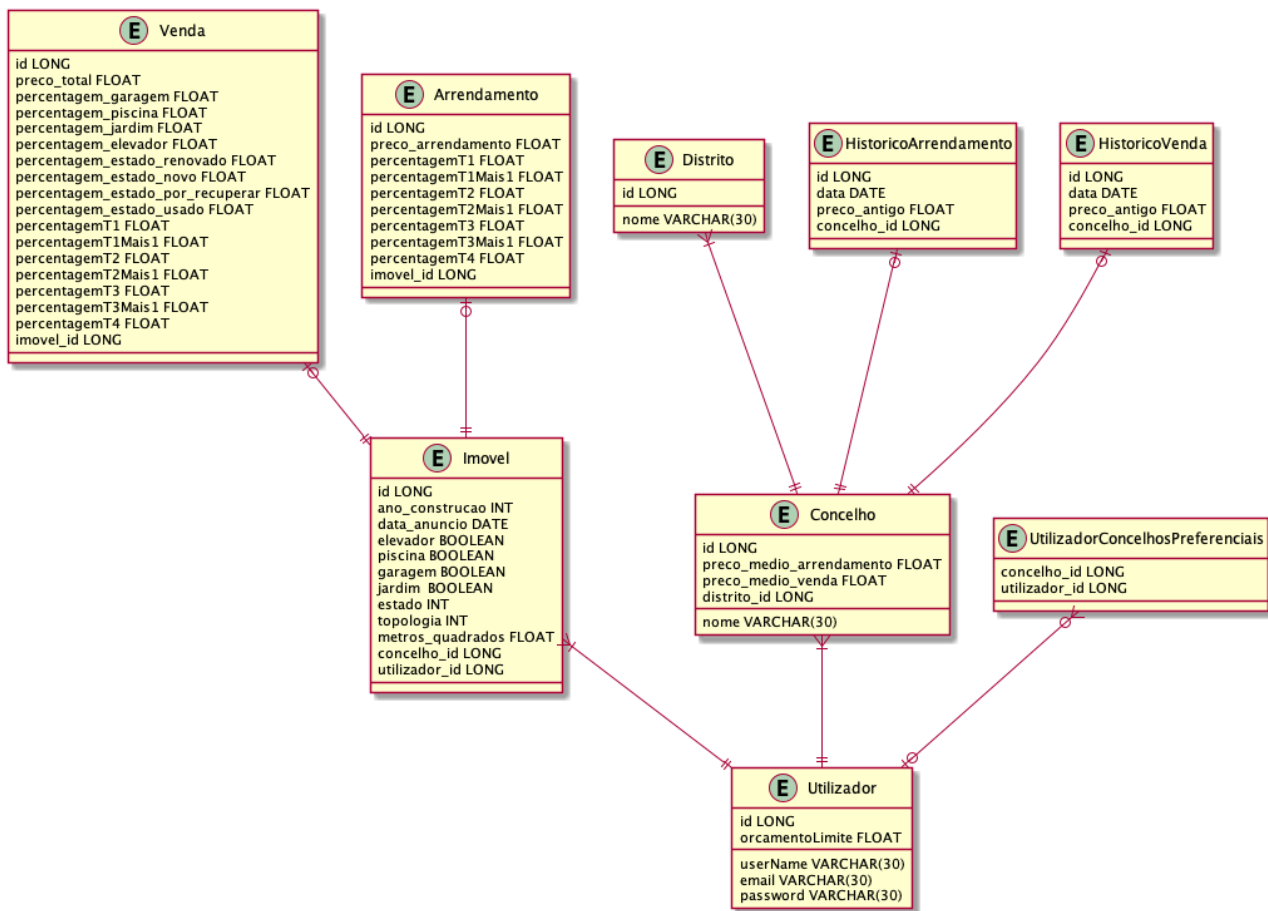


FIGURA 4 - MODELO ER

Segue-se uma pequena explicação das tabelas e suas relações, agrupadas por funcionalidade. Os tipos de dados não são especificados, devendo, para o efeito, ser consultado o esquema.

“**Venda**” - referente a elementos de configuração do tipo de negócio do imóvel

- id: chave primária incremental
- precoTotal: preço pedido pelo imóvel
- percentagem\_garagem: percentagem associado ao facto de o imóvel ter garagem
- percentagem\_piscina: percentagem associado ao facto de o imóvel ter piscina
- percentagem\_jardim: percentagem associado ao facto de o imóvel ter jardim
- percentagem\_elevador: percentagem associado ao facto de o imóvel ter elevador
- percentagem\_estado\_novo: percentagem associado ao facto de o imóvel estar novo

- `percentagem_estado_usado`: percentagem associado ao facto de o imóvel estar usado
- `percentagem_estado_por_recuperar`: percentagem associado ao facto de o imóvel estar por recuperar
- `percentagem_estado_renovado`: percentagem associado ao facto de o imóvel estar renovado
- `percentagem_T1`: percentagem associado ao facto de o imóvel ser T1
- `percentagem_T1Mais1`: percentagem associado ao facto de o imóvel ser T1+1
- `percentagem_T2`: percentagem associado ao facto de o imóvel ser T2
- `percentagem_T2Mais1`: percentagem associado ao facto de o imóvel ser T2+1
- `percentagem_T3`: percentagem associado ao facto de o imóvel ser T3
- `percentagem_T3Mais1`: percentagem associado ao facto de o imóvel ser T3+1
- `percentagem_T4`: percentagem associado ao facto de o imóvel ser T4
- `imove_id`: chave estrangeira com identificação do imóvel a que pertence esta venda

**“Arrendamento”** - referente a elementos de configuração do tipo de negócio do imóvel

- `id`: chave primária incremental
- `precoArrendamento`: preço pedido pelo imóvel
- `percentagem_T1`: percentagem associado ao facto de o imóvel ser T1
- `percentagem_T1Mais1`: percentagem associado ao facto de o imóvel ser T1+1
- `percentagem_T2`: percentagem associado ao facto de o imóvel ser T2
- `percentagem_T2Mais1`: percentagem associado ao facto de o imóvel ser T2+1
- `percentagem_T3`: percentagem associado ao facto de o imóvel ser T3
- `percentagem_T3Mais1`: percentagem associado ao facto de o imóvel ser T3+1
- `percentagem_T4`: percentagem associado ao facto de o imóvel ser T4
- `imove_id`: chave estrangeira com identificação do imóvel a que pertence este arrendamento

**“Imovel”** - armazena dados sobre o imóvel

- id: chave primária incremental
- ano\_construcao: ano em que o imóvel foi construído
- data\_anuncio: data do momento em que o imóvel foi criado
- elevador: verifica se o imóvel tem elevador
- piscina: verifica se o imóvel tem piscina
- garagem: verifica se o imóvel tem garagem
- jardim: verifica se o imóvel tem jardim
- estado: estado em que o imóvel se encontra
- topologia: topologia referente ao imóvel
- metros\_quadrados: área total do imóvel
- concelho\_id: chave estrangeira com identificação do concelho a que o imóvel pertence
- utilizador\_id: chave estrangeira com identificação do utilizador a que o imóvel pertence

**“Utilizador”** - armazena dados dos utilizadores

- id: chave primária incremental
- orcamentoLimite: orçamento limite que o utilizador tem disponível para dar por um imóvel
- userName: nome do utilizador utilizado para login
- email: email do utilizador utilizado para login
- password: password do utilizador para login

**“Concelho”** - armazena dados sobre concelhos

- id: chave primária incremental
- nome: nome próprio do concelho

- preco\_medio\_arrendamento: preço médio por metro quadrado do concelho para o negócio de arrendamento
- preco\_medio\_venda: preço médio por metro quadrado do concelho para o negócio de venda
- distrito\_id: chave estrangeira com identificação do distrito a que pertence este concelho

“**Distrito**” - armazena dados sobre distritos

- id: chave primária incremental
- nome: nome próprio do distrito

“**UtilizadorConcelhosPreferenciais**” - armazena dados sobre os concelhos preferenciais de um utilizador

- concelho\_id: chave estrangeira com identificação do concelho preferencial
- utilizador\_id: chave estrangeira com identificação do utilizador

“**HistoricoArrendamento**” - armazena dados relativos ao histórico de preços praticados anteriormente para o negócio de arrendamento

- id: chave primária incremental
- data: data associada ao histórico
- preco\_antigo: preço praticado anteriormente do negócio de arrendamento
- concelho\_id: chave estrangeira com identificação do concelho

“**HistoricoVenda**” - armazena dados relativos ao histórico de preços praticados anteriormente para o negócio de venda

- id: chave primária incremental
- data: data associada ao histórico
- preco\_antigo: preço praticado anteriormente do negócio de venda
- concelho\_id: chave estrangeira com identificação do concelho

## 5. Arquitetura MVC

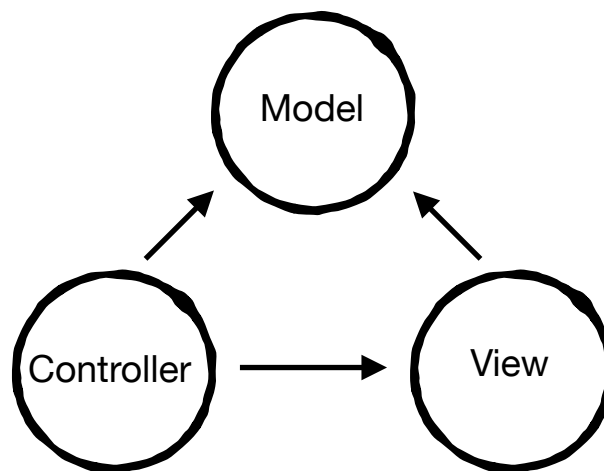


FIGURA 5 - MODELO MVC

Como arquitetura global, para este projeto optei pelo MVC (Model View Controller) pois este é um padrão de arquitetura de software responsável por contribuir na otimização da velocidade entre os requisitos feitos pelo utilizador e facilita a troca de informações entre interface e a base de dados, o que faz com que as respostas sejam mais rápidas e dinâmicas.

Como é que funciona? O browser é responsável por enviar os requisitos via HTTP para o servidor. O controlador irá procurar qual é a classe responsável para tratar esse requisito. Posteriormente, o controlador passa os dados para o modelo, que por sua vez executa todas as regras de negócio, como cálculos, validações e acesso à base de dados. O resultado dessas mesmas operações realizadas pelo modelo é retornado ao controlador. O controlador retorna o nome da vista que processa os dados, transformando os resultados em HTML. Por último, o HTML é retornado ao browser do utilizador.

## 6. Conclusão

Neste projeto que me comprometi a fazer fiquei agradado com o resultado final, sinto que se tivesse começado a desenvolver a parte do código desde início em JAVA conseguia realizar todos os requisitos que me tinha proposto, no entanto, acho que realizei todos os requisitos que tinham mais impacto no projeto, não fiz o requisito de autenticação que é o [RF-01] visto que a autenticação não tem um grande impacto na utilização da aplicação pois o projeto foi desenvolvido para apenas um tipo de utilizador, também não fiz o [RF-10] e o [RF-11] que tem haver com a escolha de concelhos preferenciais do utilizador e visto que não tenho a autenticação realizada e os referidos requisitos dependem disso. Contudo, a aplicação está totalmente preparada para implementar esse requisitos que apenas não foram apresentados por questões de tempo. Foi um grande desafio para mim aprender uma linguagem de programação que desconhecia por completo a sua implementação (Python) e trabalhar com uma linguagem que não era de toda a minha zona de conforto (JavaScript), contudo sinto que superei os meus obstáculos de forma eficaz e estou bastante feliz com o trabalho apresentado. Também aprendi a desenvolver um trabalho sozinho, onde os resultados apenas e só dependiam de mim, pois nunca tinha feito um trabalho desta dimensão sozinho.

Em relação ao design do meu trabalho também fiquei bastante contente para comigo mesmo, sinto que apresentei uma interface simples e direta onde até uma criança que saiba ler consiga realizar todas funcionalidades da minha aplicação, o único aspeto que podia alterar e que seria fácil de o fazer, seria a apresentação do histórico de preços praticados num concelho, onde os organizava numa tabela onde colocava nas linhas a data a que esse histórico está relacionado e nas colunas o preço a que corresponde a essa data.

De salientar também que o valor apresentado da avaliação do imóvel efectuado pela aplicação não representa o valor real do imóvel, pois o negócio do imobiliário não é uma ciência exata e maior parte dos fatores estão relacionados com o aspeto visual, e o valor do preço por metro quadrado de uma região é alterado dia após dia. Também podia ter explorado mais fatores como por exemplo se o imóvel tem vista para o mar, se a piscina é pública ou privada, entre outros, mas o prazo de entrega para este projeto não permitia que todos esses fatores fossem explorados.



## 7. Referências

- Pereira, I., & Moreira, A. (2020). Apontamentos da Disciplina de Engenharia de Software. Porto, Portugal: Universidade Fernando Pessoa.
- Gouveia, F. (2020). Apontamentos da Disciplina de Base de Dados. Porto, Portugal: Universidade Fernando Pessoa.
- Moreira, R. (2019). Apontamentos da Disciplina de Linguagem de Programação II. Porto, Portugal: Universidade Fernando Pessoa.
- (2021). Django - Everything you need to know about Django: <https://djangostars.com>
- (2021). Spring - What's New, Upgrade Notes, Supported Versions, and other topics, independent of release cadence: <https://spring.io>
- (2021). PlantUML - Drawing UML with PlantUML: [http://pdf.plantuml.net/PlantUML-Language\\_Reference\\_Guide\\_en.pdf](http://pdf.plantuml.net/PlantUML-Language_Reference_Guide_en.pdf)
- (2021). Bootstrap - Getting Started with Bootstrap · The world's most popular mobile-first and responsive front-end framework: <http://getbootstrap.com/>
- Bootstrap. (s.d.). Bootstrap - the world's most popular mobile-first and responsive front-end framework. Obtido de Bootstrap.