

**INSTITUTO POLITÉCNICO DE BEJA**

**ESTIG**

**ENGENHARIA INFORMÁTICA**

O Projeto U-bike Portugal

Discentes:

Hugo Alexandre Silva - Nº 18544

Jorge Colaço - Nº *16524*

Docente:

Isabel Sofia Sousa Brito

Beja, Abril 2020

Índice

[1 – Introdução 2](#_Toc516511299)

[2 – Análise 3](#_Toc516511300)

[2.1 – Recolha de informação 3](#_Toc516511301)

[2.1.1 – A marca 3](#_Toc516511302)

[2.1.2 – Entrevistas a utilizadores 4](#_Toc516511303)

[2.1.3 – Utilização da Aplicação 5](#_Toc516511304)

[2.2 – Especificação de requisitos 8](#_Toc516511305)

[2.2.1 – Requisitos funcionais 8](#_Toc516511306)

[2.2.2 – Requisitos não funcionais 10](#_Toc516511307)

[2.3 – Diagramas de casos de uso 10](#_Toc516511308)

[3 – Desenho 11](#_Toc516511309)

[3.1 – Diagramas de sequência 11](#_Toc516511310)

[3.2 – Diagrama de classes 11](#_Toc516511311)

[4 – Mecanismos de validação e controlo de versões 12](#_Toc516511312)

[5 – Conclusão 13](#_Toc516511313)

Índice de Figuras

[Figura 1- 8](#_Toc516770284)

[Figura 2– 8](#_Toc516770285)

[Figura 3– 9](#_Toc516770286)

[Figura 4– 9](#_Toc516770287)

[Figura 5– 10](#_Toc516770288)

[Figura 6– 10](#_Toc516770289)

[Figura 7– 13](file:///C:\Users\watna\Documents\2%20ano%202%20semestre\Eng.%20Software\Projeto1718_Rui_Watna\Relatorio%20Versao%20IV.docx#_Toc516770290)

[Figura 8– 14](file:///C:\Users\watna\Documents\2%20ano%202%20semestre\Eng.%20Software\Projeto1718_Rui_Watna\Relatorio%20Versao%20IV.docx#_Toc516770291)

# 

# 1 – **Introdução**

Neste trabalho prático vamos promover a mobilidade suave e prevê a realização de ações concretas de incentivo à adoção de hábitos de mobilidade mais sustentáveis nas comunidades académicas do ensino superior, baseado na cadeira de Engenharia de Software foi proposto desenvolver o projeto de software da empresa U-bike Portugal. Não se torna difícil encontrar softwares similares ao projeto proposto, por isso visando a realização da análise do sistema que compõe a empresa, conhecendo os seus objetivos e metas de negócio, podemos determinar melhor as funcionalidades, utilizadores e casos de uso.

Programa Operacional Sustentabilidade e Eficiência no Uso de Recursos (PO SEUR), propõem criar uma alternativa de transporte amiga do ambiente, mais económica, introduzindo hábitos de vida saudável e diminuindo o tempo de deslocação, sobretudo dos estudantes, entre o local de residência e o campus do IPBeja. Visando essa alternativa vamos basear o relatório em pesquisa, análise, desenho e controle de gestão. A pesquisa terá a obtenção de informação de utilizadores e visão do fornecedor do serviço, assim determinaremos os quesitos do sistema de negócio, as suas funcionalidades, além do melhor desenho. Essa análise de recolha de informação obtida juntamento com o obejtivo nós ajudará a identificar todos os requisitos que a aplicação necessita, além da elaboração de um diagrama, caso de uso interligado à utilização da aplicação.

Objetivo deste trabalho é desenvolver a plataforma informática que serve de base ao projeto, tendo em vista a monitorização da utilização das bicicletas, bem como a verificação de, pelo menos, dois indicadores, a saber: a medição da eficiência energética e a introdução de hábitos de vida saudável. Na elaboração do desenho vamos demostrar através de diagramas de sequência UML e de classes esse objetivo, utilizando o programa Visual Paradigm.

Na fase da controle de gestão iremos demonstrar como foi implementado o mecanismo de controlo de versão usado durante a realização deste projeto.

# 2 – Análise

## 2.1 – Recolha de informação

Ao recolher a informação necessaria para desenvolvermos um execelente trabalho, analisamos os serviços e modalidades que dispõe de outros semelhantes e o que oferecem, ou seja, dentro dos requisitos e dados necessários para a elaboração de um execelente sistema que atenda as demandas dos utilizadores e os objetivos do fornecerdor. Realizamos entrevistas através de um formulário online com diversos possiveis utilizadores do serviço da U-bike Protugal e analisamos as diversas funcionalidades da aplicação concorrente e da pataforma atual do serviço da U-bike Portugal.

É essencial conhecer a história e sobretudo os objetivos, metas a que se atem a empresa de modo a ser possível analisar de uma perspetiva comercial e desenvolver um projeto de software adequado aos serviços esperados. Também é importante analisar o impacto desejado na sociedade, em relação a criar uma consciência e hábitos de uma vida mais saudável. Assim conhecer as funcionalidades que podem ser mais utilizadas, bem como a forma como são introduzidas no cotidiano para que na proxima etapa do projeto possamos descrever alguns requisitos funcionais e não funcionais acertivos.

### 2.1.1 – A U-bike Portugal

O Projeto U-bike Portugal visa promover a mobilidade suave em particular com uso da bicicleta, em comunidades académicas de 15 instituições de ensino superior (IES). O projeto enquadra-se no [Portugal 2020](https://poseur.portugal2020.pt/pt/portugal-2020/) concretamente no [POSEUR](https://poseur.portugal2020.pt/pt/candidaturas/avisos/poseur-07-2015-31-aviso-projeto-u-bike-portugal/) - Programa Operacional Sustentabilidade e Eficiência no Uso de Recursos.

São € 6.059.027,90 de investimento total do projeto, onde € 5.616.089,28 correspondem a investimento elegível, sendo € 4.773.675,88 comparticipados pelo Fundo de Coesão. O Instituto da Mobilidade e dos Transportes, I.P., é a entidade da Administração Pública que coordena o Projeto, assegurando a existência de regras comuns e a posterior monitorização das operações locais, com vista a harmonizar a aplicação e o impacto do Projeto a nível nacional.

Serão atribuídas 3.234 bicicletas (2.096 elétricas e 1.138 convencionais), com base em normas definidas por cada instituição e em cumprimento do [regulamento geral do projeto U-bike Portugal](http://www.imt-ip.pt/sites/IMTT/Portugues/Noticias/Documents/2016/Regulamento_Rev19022016.pdf) para uma utilização de longa duração (ex.: aluguer durante um semestre ou um ano letivo) que origine a criação de hábitos regulares de utilização deste meio de transporte.

O Projeto teve início em setembro de 2016, com prazo máximo de execução de 2 anos. Prevê-se que sejam percorridos anualmente 2.412.141 Km em bicicletas, somando 26 municípios ao todo, o que corresponde a uma poupança de 166,34 toneladas equivalentes de petróleo e à consequente redução de 505 toneladas equivalentes de CO2. Os pagamentos podem ser feitos através da aplicação onde o utilizador poderá cadastra uma forma de pagamento para o serviço, sendo assim preservados suas credenciais, informações e trazendo segurança para o utilizador e fiabilidade para escolher a melhor forma possível para usufruir do serviço.

As opções de pagamento podem incluir:

* Cartões de crédito e cartões de crédito pré-pagos Visa, MasterCard
* MBway
* Cartões de débito que podem ser processados na função crédito
* Paypal
* Criar uma conta na aplicação para deposito através de entidade e referência, onde o utilizador pode depositar a quantia que pretende e esse valor fica atralado a sua conta de utilizador e será descontado conforme uso.

### 2.1.2 – Entrevistas com utilizadores

**Utilizadores**

Os utilizadores estão atribuidos ao conjunto de atores, neste caso são elementos externos ao sistema mas que estão integrados nesse grupo de utilizadores que irão fazer uso do sistema para executar tarefas, funções primárias e secundárias do sistema, havendo cada sessão dentro do mesmo.

Podemos usar atores no sistema que desempenham mais do que uma ação, podendo obter recuros para proceder com os casos de uso e que executam as ações como suporte. Diversos tipos de utilizadores exigem casos de uso específicos em diversos sistemas, na elaboração da entrevista utilizamos um pouco dos conhecimento adquiridos nas matérias de TWAM, Interação Pessoa Computador e PDF do curso de “Questionarios, o que é”, além de aplicar nossos conhecimentos pessoais e do conteúdo da matéria Engenharia de Software.

--------entrará imagens e dados do formulario online---

<https://forms.gle/d67kXXgLY41HwJw6A>

### 2.1.3 – Utilização de Aplicações

Enquanto utilizador fizemos uma pesquisa de três sistema diferentes, a saber: uBike, MobiCascais, U-bikeLeiria, foi criada uma simulação nesses aplicativos de uma forma possível avaliando a utilização dos serviços, ações, funcionalidade e objetivos, as funcionalidades descritas e analisadas são compativeis com o que procuramos e podemos tirar como proveito certas funcionalidades, dados e recursos para a elaboração do projeto em questão. As aplicação encontram-se disponível para as seguintes plataformas como iOS, Android e Windows, através das playStore de cada um e instalando-a num dispositivo móvel.

Apos a instalação, como em qualquer aplicativo é necessário a introdução de dados pessoais que serão guardados no perfil do utilizador. Informações como nome completo, e-mail, palavra-passe e data de nascimento, entre outros, são dados obrigatórios de introdução. Com essas ações formulamos nossas funcionalidades e desenvolvimento de sistema em desenho de alto nivel que poderemos demonstrar a visualização das imagens a seguir (figura 1, 2, 3, 4, 5 e 6).

Figura 1-PÁGINA INICIAL

Figura 2–PÁGINA DE DADOS PESSOAIS

Figura 3–(EMAIL)

Figura 4–(PASSWORD)

Figura 5–DADOS PESSOAIS

Figura 6–ACEITE DA POLÍTICA

## 2.2 – Especificação de requisitos

Através da recolha de informação realizada anteriormente, os sistemas são definidos durante as fases iniciais do processo do desenho, com essa ação chegamos a possibilidade conhecer e identificar os objetivos do sistema, suas pricipais funcionalidades e determinar a ação do negócio.

Compreendendo quais são os objetivos a que a U-bike Portugal espera atingir, nos propomos a analisar, identificar e descrever essas características e funcionalidades da aplicação dentro da forma de requisitos e ampliando o uso se assim for o caso. Definindo os requisitos do sistema é preciso ter em consideração alguns pontos, que são as bases dos requisitos:

* Objetivos do sistema;
* Dados do sistema;
* Modelos de clientes e utilizadores;
* Padrão de uso e designer;
* Análise de sistemas semelhantes;

Os requisitos compõem o conjunto de necessidades estabelecido pelo cliente/usuário que define a estrutura e o comportamento do software que esta sendo desenvolvido, através de uma definição clara de requisitos e uma especificação dos mesmos, podemos citar: os processos, os dados que se espera serem gerados, as restrições operacionais, as pessoas que irão se utilizar do software e todas essas questões relacionadas entre si. Para garantir que os requisitos definidos e implementados são satisfatórios, coerentes e plenamente funcionais foram definidas as premissas que o software que esta a ser desenvolvido possa:

* Permitir que o fornecedor resolva os problemas relacionados ao negócio da sua empresa.
* Atender as necessidades ou restrições da organização ou dos outros módulos do sistema.
* Permitir a flexibilidade, facilidade e uso do sistema para qualquer utilizador.

### 2.2.1 – Requisitos funcionais

Os requisitos diretos são chamados de requisitos funcionais e se refere as diferentes funcionalidades que o cliente/utilizador deseja que o sistema faça, especifica uma função que o sistema realiza tendo em consideração as funcionalidades a implementar seja por comando do utilizador ou acionados automaticamente por eventos internos ou externos do sistema. A descrição do requisito direto deve descrever detalhadamente o que se espera que o sistema faça, sem a preocupação de como ele será construído, esses requisitos capturam as funcionalidades sob o ponto de vista do utilizador.

**Cliente:**

* Efetua cadastro;
* Efetuar login;
* Seleciona serviço;
* Realiza o pagamento pelo tempo que deseja usar;
* Efetua a solicitação da bicicleta;
* Recebe codigo de liberação do cadeado;
* Verifica área de uso e condução da bicicleta;
* Entrega a bicicleta no local de recolha;
* Efetua a liberação do codigo no aplicativo;
* Excede o tempo anteriormente solicitado a bicicleta bloqueia;
* Realiza o pedido de nova liberação;
* Verifica saldo financeiro no aplicativo para uso do serviço;
* Entrega realizada da bicicleta;
* Partilhar serviço com amigos;

### 2.2.2 – Requisitos não funcionais

Os requisitos não funcionais ou indiretos procuram garantir um funcionamento correto e seguro do sistema, os requisitos indiretos referem-se as qualidades globais do software, como sua facilidade de manutenção, segurança, facilidade de utilização e acessibilidade, desempenho ao imporem certas restrições ao mesmo, limitando as possibilidades de implementação. A descrição destes requisitos está associada a definição de usabilidade propriamente dita, baseando no descrito, a forma desse funcionamento visa criar um ambiente seguro, eficiente e estável, melhorando a experiência do utilizador.

Assim, é possível demarcar os seguintes requisitos:

* Confidencialidade de dados dos utilizadores;
* Percepção comum entre desenvolvedores, utilizadores sobre o sistema;
* Interação entre pessoa computador;
* Utilização em varios dispositivos;
* Uma fonte confiável para estimativa de custos, pessoal e prazos.
* Comunicação com o Google Maps e localização;
* Utilização de uma base de dados;

## 2.3 – Diagramas de casos de uso

O diagrama de caso de uso descreve/auxilia no levantamento dos requisitos funcionais do sistema, um novo sistema que será desenhado, é uma excelente ferramenta para o levantamento dos requisitos funcionais do sistema. UML é um acrônimo para a expressão Unified Modeling Language. Pela definição de seu nome, vemos que a UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos, então no diagrama de casos de uso, descrevendo um conjunto de funcionalidades do sistema e suas interações com elementos externos e entre si.

* Cenário: compreende uma sequência de passos que descreve uma interação entre um utilizador e o sistema.
* Ator: utilizador do sistema, seja cliente ou fornecedor do serviço.
* Use Case: tarefa ou uma funcionalidade realizada pelo ator.
* Comunicação: é o que liga um ator com um caso de uso.

Possíveis casos de uso para o sistema foram consideradas com as seguintes possibilidades:

* Qual função um ator vai obter do sistema?
* Definição de Requisitos - Novos casos de usos geralmente geram novos requisitos conforme o sistema vai sendo analisado e modelado;
* O sistema armazena informação em qual banco de dados?
* Geração de Casos de Teste - A junção de todos os cenários para um caso de uso pode sugerir uma bateria de testes para cada cenário.
* Haveria possibilidade de notificação com o ator sobre mudanças no sistema?
* Comunicação com os Clientes - Pela sua simplicidade, sua compreensão não exige conhecimentos técnicos, portanto o cliente pode entender muito bem esse diagrama, que auxilia o pessoal técnico na comunicação com clientes;

Visando os pontos citados acima podemos pensar no seguinte:

* Quem é o publico alvo do sistema?
* Quem fornece informação ao sistema?
* Quem é responsável pela administração do sistema?
* O sistemas comunica com algum outro tipo de sistema?

Nesta etapa identificamos os seguintes atores do sistema:

Figura 7– AUTORES DO SISTEMA

Figura 8– DIAGRAMA DE CASO DE USO

# 3 – Desenho

## 3.1 – Diagramas de sequência

O diagrama consiste em mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização de uma operação, sequência mostrada numa interação, são mensagens ou comunicação se assim podemos dizer entre vários objetos num certo contexto, analisando os casos de uso e as operações. Esta sequências tratam de interações em um determinado caso de uso e é apresentada de forma cronológica, assim conseguimos averiguar as tarefas de cada objeto no sistema.

Para cada caso de uso desenvolvemos os respetivos diagramas de sequência:

---------- imagens diagramas de sequência ------

## 

## 3.2 – Diagrama de classes

Diagrama de classes é um diagrama estático, em programação esse diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos.O diagrama de classes não é usado apenas para visualizar, descrever e documentar diferentes aspetos de um sistema, é composto por atributos e operações, descrevendo as informações, sob a forma de atributos mas também para construir um código executável do aplicativo de software. O diagrama de classes descreve os atributos e tarefas de uma classe ou parte espécifica e também as restrições impostas ao sistema.

Visando a criação do sistema U-bike, para o IPBeja por exemplo, criamos o seguinte diagrama de classes:

---------- imagens diagramas de classe ------

# 4 – Mecanismos de validação e controlo de versões

Um sistema de controle de versão busca gerenciar diferentes versões de um projeto que implique trabalhos de grupo ou até mesmo individual. Passa a ser essencial atribuir determinadas tarefas a cada colaborador, com isso buscamos resultados como: eficiência, segurança, rapidez de execução. Nesta colaboração se torna cada vez mais indispensável esse mecanismos de controlo de versões. Este tipo de ferramentas possibilita a atualização de ficheiros, codigos e até mesmo verificação de utilizadores, além da consulta de versões anteriores dentro do ficheiro.

Cada utilizador poderá consultar os ficheiros no repositório, tendo claro um acesso associado ao repositório, oferecendo uma maneira muito mais segura e eficaz de organizar os projeto, foi fundamental utilizarmos esse mecanismo de controlo de versões no nosso projeto, assim dividindo ou corrigindo as tarefas, podemos trabalhar de forma pratica e agil sobre o projeto.

Optamos por utilizar a ferramenta GitHub, sendo um ferramenta de repositório online, da empresa Microsoft atualmente, gratuito e público, apesar de incluir opções de privacidade caso necessite, é também uma ferramenta que contribui para a segurança no desenvolvimento do projeto, podendo armazenar várias versões dentro do mesmo trabalho.

<https://github.com/haspimenta/EngenhariaSoftware>

---------- imagens do gitHub dos commits ------

# 5 – Conclusão

A conclusão deste trabalho prático, permitiu compreender melhor todos os processos associados à implementação e gestão eficiente juntamente conceitos obtidos o ciclo da materia de engenharia de software, colocando em prática conceitos relacionados com o planejamento de casos de uso, construção dos diagramas de classes, diagramas de sequência e a metodologia UML em si.

Os resultados obtidos podem continuar sendo melhorados e adequados dentro das conformidades dos códigos de uma maneira mais integra e enxuta assim como ampliar as ações e execução do projeto. Compreendemos a importância destes elementos no desenvolvimento de um software, os conceitos estudados nesse projeto podem ser utilizados em outras aplicações e projetos similares uma vez que concede um planejamento eficaz do projeto, procurando encobrir possíveis falhas ou erros no sistema, bem como o aumento da eficiência de trabalho.

No que diz respeito a configuração, funcionalidades, desempenho, segurança, a realização do projeto demonstrou que as dificuldades encontradas estão relacionadas à identificação de casos de uso e obtenção de informações concisas, pois precisamos de dados coerentes, evitando repetições ou redundâncias. De uma forma simples e com todos os resultados obtidos, buscamos demonstrar em geral todos os passos executados para a elaboração do projeto, assim como as funcionalidades e demais especificações atraves de impressões em tela, permitindo assim sua reprodução de valores, especificações e atendimento quanto aos quesitos.

Concluímos o presente trabalho com o sentimento de ter alcançado resultados satisfatórios, proposto pelo projeto U-bike Portugal, através da docente Isabel Brito e que não existe limite para a aplicação neste modelo de funcionamento e melhoria de qualquer parte do projeto, além de despertar a consciencia das praticas necessarias no desenvolvimento de um projeto de software durante todo o percurso, desde o levanto de dados e requisitos a entrega do mesmo.

**6 – Bibliografia**

<https://pt.wikipedia.org/wiki/Diagrama_de_classes>

<https://pt.wikipedia.org/wiki/Diagrama_de_caso_de_uso>

<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/interacao/sequencia.htm>

<https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5>

<http://www.linhadecodigo.com.br/artigo/3224/analise-e-especificacoes-de-requisitos.aspx>

<http://www.macoratti.net/07/12/net_fer.htm>

<https://edisciplinas.usp.br/pluginfile.php/3720765/course/section/857581/Aula02_CasosDeUso.pdf>