

**Curso Profissional**  
**Técnico de Gestão e Programação de Sistemas Informáticos**  
[Edição 2019-22]

# **Relatório do Projeto de Aptidão Profissional(PAP)**

Filipe Aguilar Apolinário Correia -12ºC

Técnico de Gestão e Programação de  
Sistemas Informáticos | 2019-2022



# Índice

Índice	2
1. Resumo	3
2. Introdução	4
3. Enquadramento teórico	5
4. Finalidades	6
5. Metodologia	7
6. Cronograma	8
7. Recursos	9
8. Desenvolvimento	10
Desenvolvimento da base de dados	10
Desenvolvimento do website	13
Desenvolvimento da aplicação para telefone	16
Problemas e soluções encontrados	24
9. Desenvolvimento futuro	25
10. Avaliação	26
11. Reflexão final / Conclusão	27
12. Bibliografia	28
13. Anexos	29

# 1. Resumo

Ao longo deste projeto, procurei desenvolver uma aplicação mobile e um website para a área do turismo. No fundo, a pretensão seria criar um posto turístico que tivesse aberto 24 horas por dia, ou seja, que estivesse sempre disponível, para que a qualquer momento pudéssemos ter acesso à informação que desejássemos.

Tenho como principal objetivo, com o presente trabalho, que as pessoas, quando se deslocam em lazer, conheçam o local onde se encontram. Assim, quando lá chegarem, uma API (Application Programming Interface) vai revelar a sua localização geográfica para, dessa forma, ser possível ter acesso aos principais pontos turísticos do meio onde se encontram.

De forma a ir corrigindo problemas e bugs detetados pelos utilizadores ou inserir locais novos ou aspetos considerados interessantes para novas experiências, terei comigo administradores. Podemos, desta forma, dizer que esta aplicação evoluirá de forma constante, nunca estagnando, sendo sempre possível melhorá-la.

## 2. Introdução

O presente relatório pretende dar cumprimento ao disposto na alínea c) do ponto 1 do artigo 20.º da Portaria 550-C/2004 de 21 de Maio, que refere a necessidade de elaboração da auto-avaliação e elaboração do relatório final relativo à Prova de Aptidão Profissional. Foi ainda tido em consideração o Regulamento Interno do Agrupamento de Escolas Dr. Serafim Leite, nomeadamente o que é referido no artigo 9.º, do Anexo IV – Regulamento da Prova de Aptidão Profissional. Tendo em consideração todas as opções relativamente à Prova de Aptidão Profissional, implementei um software de gestão turística portuguesa.

A ideia para este projeto surgiu-me do facto de me deslocar diariamente para a escola de bicicleta. Este meio de transporte permite-me, por um lado, uma grande independência de mobilidade e, por outro, contribuir para a preservação do ambiente assim como para o meu bem-estar físico e mental. Como pretendo também adquirir uma e-bike e aventurar-me em viagens mais longas, achei interessante desenvolver uma ferramenta tecnológica que possibilite uma planificação da viagem, mais concretamente no que diz respeito ao abastecimento elétrico.

Com o desenvolvimento da aplicação mobile e do website, procuro satisfazer uma necessidade que me parece importante já que muitas vezes as pessoas deslocam-se a locais turísticos, mas depois têm dificuldade em organizar a sua visita. Havendo diversos locais a visitar, torna-se muito mais fácil, no meu entender, recorrer a uma ferramenta digital e utilizá-la para potenciar o tempo que estamos nesse local e organizar um roteiro da visita, a partir dos interesses de cada um. No fundo, esta aplicação seria um posto de turismo sempre disponível.

Utilizei o MySQL como base de dados, uma base de dados relacional que usa o SQL para fazer ligações e alterações. Recorri à linguagem PHP para back-end e HTML, javascript (JS) e CSS para fazer o front-end. Já na aplicação mobile utilizei Kivy, KivyMD e Python. Importei as bibliotecas e API's que achei necessárias para a criação da mesma.

### 3. Enquadramento teórico

O meu projeto foi maioritariamente desenvolvido em PHP, JS, HTML, CSS, Python e Kivy.

O PHP foi uma linguagem de programação que foi abordada ao longo do meu percurso formativo. Mesmo assim, tive necessidade de realizar um trabalho de pesquisa para alguns pormenores não abordados na escola, o que, por sua vez, me permitiu enriquecer os meus conhecimentos.

Segundo o livro ‘Linguagens web’ de Alexandre Pereira e Carlos Poupá, o PHP “É a linguagem de programação para internet mais amplamente utilizada. Pode ser executada em quaisquer plataformas: Unix, Linux, Windows, Solaris, HP-UX. Pode ser executada a partir de diversos servidores de Internet, entre eles, o Apache e o IIS.” (p. 257). Devido à sua grande abrangência, pareceu-me a linguagem adequada a este projeto.

No desenvolvimento do website, utilizei PHP, para back-end, ou seja, criar uma ligação com o servidor. Recorri ao HTML, JS, CSS Bootstrap para fazer o front-end, isto é, a interface visível para o utilizador. Por sua vez, o Bootstrap foi utilizado para tornar o website mais responsivo e bonito em toda a sua constituição.

Relativamente à aplicação mobile, recorri ao Kivy, KivyMD para a estrutura e ao Python para a elaboração do código fonte da aplicação.

Para guardar os dados, usei o MySQL, uma plataforma gratuita de Base de Dados que me ajudou bastante para poder guardar os dados online e ter acesso a partir do telemóvel e do computador, podendo dessa forma fazer todas as consultas necessárias para gerir o sistema informático.

Para a realização deste projeto, fiz também uma pesquisa sobre softwares concorrentes no mercado das aplicações móveis para averiguar qual a melhor opção a utilizar. Depois de uma análise, podemos ver que existem imensas opções que permitem ao utilizador gerir um sistema informático de uma forma autónoma, ficando tudo guardado na base de dados à qual pode ter sempre acesso, facilitando desta forma a sua consulta em qualquer altura.

## 4. Finalidades

Para o desenvolvimento desta aplicação, foi usado o software Visual Studio Code para programação Web e Pycharm Community Edition para a programação para dispositivos móveis, ambos programas gratuitos e com grandes capacidades.

O presente projeto pretende servir de reflexo do longo trabalho de aprendizagem realizado ao longo de três anos de formação no Curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos.

A realização, apresentação e defesa da PAP (Prova de Aptidão Profissional), perante um júri, será, por um lado, um grande desafio, pois colocará os meus conhecimentos à prova e, por outro lado, uma etapa importante do meu processo de aprendizagem, na medida em que me defrontarei certamente com novos obstáculos, que irão ser constantes ao longo do meu percurso profissional.

## 5. Metodologia

Concretizei este projeto com a ambição de conseguir a maior aprendizagem possível, quer pelas tecnologias que investiguei e aprendi sozinho, para conseguir fazer certas funcionalidades mais complexas, quer pelas que me foram dadas a conhecer pelos diferentes docentes das várias disciplinas técnicas que fazem parte do curso. Consegui enquadrar neste projeto os conhecimentos que foram adquiridos ao longo destes 3 anos. Tentei que o resultado final do presente trabalho fosse o mais intuitivo possível, perguntando a opinião (especialmente ao nível de questões estéticas) a colegas e professores que me deram ideias novas e críticas construtivas quer para o website, quer para a aplicação móvel. Investi uma grande parte do meu tempo neste projeto e considero que foi uma aprendizagem constante, já que me apercebi que mesmo os professores podem aprender connosco, tal como nós aprendemos com eles.

Comecei pela realização da base de dados, depois passei ao desenvolvimento do website, onde a parte do css apresentou-se-me como um obstáculo inesperado e, de certa forma, difícil de resolver. Porém, com a ajuda de uma ferramenta criada pela Microsoft à qual tive acesso, chamada Github Copilot tornou-se mais fácil e consegui superar este desafio. Posteriormente, passei para a aplicação mobile, vendo diversos frameworks e formas de colocar em prática a referida aplicação.

Assim, de forma a escolher a melhor linguagem de programação que se adequasse ao presente trabalho, e já estando eu familiarizado com a interface da IDE e da linguagem que me permitiu desenvolver os meus conhecimentos, optei pelo Python.

## 6. Cronograma

	2021				2022						
Atividades	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul
Desenvolver base de dados e website	x	x	x	x	x						
Desenvolvimento da aplicação						x	x	x	x	x	
Testes finais	...	...									x



## 7. Recursos

### Hardware

- o Computador

### Software

- Visual Studio Code

- Extensão do Visual Studio "Kivy-Snippet", "Bracket Pair Colorizer", "IntelliCode", "KLang", "Tab Nine", "GitHub Copilot", "Tabnine AI" e "PHP Intelephense"

- Python 3.8

- Pycharm

- kivy do pip/PPA do python

- kivyMD

- Kivy-Garden

- Pillow

- Pygments

- certify

- cffi

- charset-normalizer

- click

- cryptography

- decorator

- docutils

- future

- geocoder

- idna

- listview

- mysql-connector.python

- pip

- protobuf

- pyc parser

- pyfiglet

- ratelim

- requests

- setuptools

- six

- urllib3

- wheel

- Extensões Pycharm:

- Tabnine

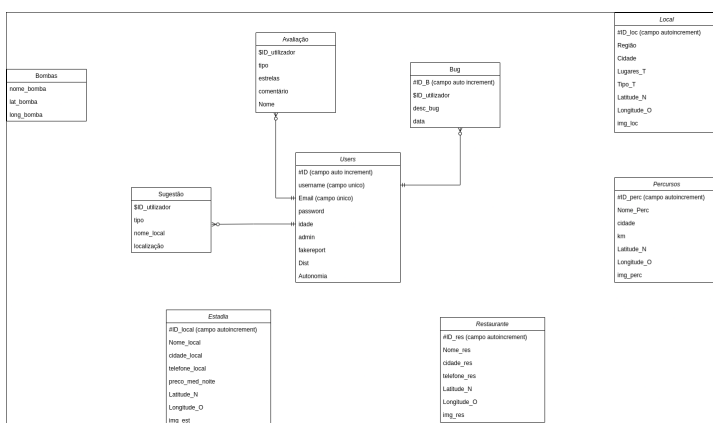
- Github Copilot

Xampp com PHP 8.0

## 8. Desenvolvimento

Desenvolvimento da Base de Dados:

Diagrama de Entidades e Relacionamentos (DEA):







Esta base de dados engloba 9 tabelas:

A tabela users contém a informação dos utilizadores e as suas permissões.


#	Nome	Tipo	Agrupamento (Collation)
<input type="checkbox"/> 1	id	int(11)	
<input type="checkbox"/> 2	username	varchar(255)	utf8mb4_general_ci
<input type="checkbox"/> 3	email	varchar(255)	utf8mb4_general_ci
<input type="checkbox"/> 4	password	varchar(255)	utf8mb4_general_ci
<input type="checkbox"/> 5	idade	int(3)	
<input type="checkbox"/> 6	admin	int(11)	
<input type="checkbox"/> 7	fakereport	int(11)	
<input type="checkbox"/> 8	dist	float	
<input type="checkbox"/> 9	autonomia	float	

As tabelas restaurantes, local, percursos e estadia contêm as informações dos diferentes sítios, assim como as coordenadas dos mesmos; o nome; imagens; etc.

#	Nome	Tipo
<input type="checkbox"/> 1	ID_Perc 	int(11)
<input type="checkbox"/> 2	Nome_Perc 	varchar(4)
<input type="checkbox"/> 3	cidade	varchar(255)
<input type="checkbox"/> 4	km	double
<input type="checkbox"/> 5	Latitude_N 	double
<input type="checkbox"/> 6	Longitude_O	double
<input type="checkbox"/> 7	img_perc	varchar(255)

#	Nome	Tipo
1	Nome_local	varchar(255)
2	cidade_local	varchar(255)
3	telefone_local	varchar(20)
4	preco_med_noite	double
5	ID_local 	int(11)
6	Latitude_N	double
7	Longitude_O	double
8	img_est	varchar(255)

#	Nome	Tipo
<input type="checkbox"/> 1	Regiao	enum('Norte', 'Centro', 'Sul')
<input type="checkbox"/> 2	Cidade	varchar(500)
<input type="checkbox"/> 3	Lugares_T	varchar(500)
<input type="checkbox"/> 4	ID_Loc 	int(11)
<input type="checkbox"/> 5	Tipo_T	enum('T_S', 'T_SP', 'T_A', 'T_D', 'T_R', 'T_E', 'T_...')
<input type="checkbox"/> 6	Latitude_N	double
<input type="checkbox"/> 7	Longitude_O	double
<input type="checkbox"/> 8	img_loc	varchar(100)

#	Nome	Tipo	Agrupamento (Collation)
1	Nome_res	varchar(255)	utf8mb4_general_ci
2	cidade_res	int(11)	
3	telefone_res	varchar(20)	utf8mb4_general_ci
4	preço_med_refeição	double	
5	ID_res 	int(11)	
6	Latitude_N	double	
7	Longitude_O	double	
8	img_res	varchar(11)	utf8mb4_general_ci

A tabela bombas contém o nome e coordenadas das 2916 bombas de Portugal Continental. Estas bombas provieram de um ficheiro JSON, e eu corri um script para as pôr na base de dados. Estas bombas, prevejo que possam ser postos de abastecimento de bicicletas elétricas.

#	Nome	Tipo
1	nome_bomba	varchar(255)
2	lat_bomba	varchar(255)
3	long_bomba	varchar(255)

A tabela sugestão contém a sugestão de lugares, que futuramente podem ser adicionados na aplicação.

#	Nome	Tipo
1	id 	int(11)
2	tipo	varchar(255)
3	nome_local	varchar(255)
4	localização	varchar(255)

A tabela bug apresenta as informações de possíveis bugs relatados pelo utilizador.

	#	Nome	Tipo
<input type="checkbox"/>	1	id 	int(11)
<input type="checkbox"/>	2	ID_B 	int(11)
<input type="checkbox"/>	3	desc_bug	varchar(255)
<input type="checkbox"/>	4	data	varchar(10)

A tabela avaliação permite aos utilizadores que visitam os locais, atribuírem uma avaliação para os administradores perceberem a satisfação dos utilizadores quanto aos locais disponíveis.

#	Nome	Tipo
1	id 	int(11)
2	tipo	varchar(255)
3	estrelas	int(11)
4	comentário	varchar(255)
5	nome	varchar(255)

Desenvolvimento do Website:

## Register

Nome de utilizador
Email
idade
Password
Confirmar Password
<b>Regista-te</b>

(/comum/register)

## Login

Email
Password
<b>Login</b>

(/comum/login)

Para desenvolver o website utilizei HTML, JS, CSS, PHP e Bootstrap.

Criei inicialmente 2 formulários, 1 para login e 1 para registo.

Quando o utilizador entra, e este tem permissão de administrador, são apresentadas 2 tabelas para ele ter a visão geral dos acontecimentos até ao momento.

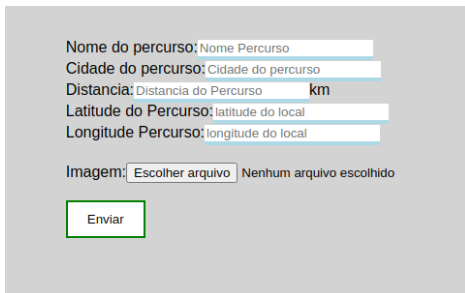
Numero de Bugs por Corrigir	Numero de Utilizadores registrados nesta plataforma	Numero Avaliações Positivas (+ de 3 estrelas)
3	18	3

Numero de Percursos existentes	Numero de Locais existentes	Numero de Locais para estadiar existentes	Numero de restaurantes existentes
5	5	1	1

(bug.php)

Também têm um formulário, para conseguir inserir locais.



(/admin/percursos.php)

É utilizado neste caso o PHP, para guardar as informações em variáveis e realizar o insert na base de dados.

```
$lat = $_POST['lat'];
$lon = $_POST['lon'];
$nome_reg = $_POST['Reg_loc'];
$tipo_T = $_POST['Reg'];
$cidade = $_POST['Cidade_loc'];
$nome = $_POST['nome_loc'];
//acabar o insert
$sql = "INSERT INTO local (Regiao, Cidade, Lugares_T, ID_Loc, Tipo_T, Latitude_N, Longitude_O, img_loc) VALUES ('$nome_reg', '$cidade', '$nome', NULL, '$tipo_T', '$lat', '$lon', '$fileName');";
if (mysqli_query($link, $sql)) {
    window.alert("O local foi adicionado com sucesso");
} else {
    window.alert("Erro ao adicionar local");
}
```

(bug.php)

Também é possível ver os bugs reportados pelos utilizadores, por meio de uma tabela onde o utilizador pode ver detalhes como a descrição e a data do bug. Pode reportar o utilizador por indicar um bug inexistente, e pode marcar o bug como corrigido, eliminando-o da tabela SQL.

```
<table border="2">
  <tr>
    <th>Descrição Bug</th>
    <th>Detalhes</th>
    <th>Fake Report</th>
    <th>Corrigido</th>
  </tr>
  <tr>
    <td><?php echo $data['desc_bug']; ?></td>
    <td><a class="a" href="detalhes.php?id=<?php echo $data['ID_B']; ?>">Detalhes</a></td>
    <td><a class="a" href="fakereport.php?id=<?php echo $data['ID_B']; ?>">Report</a></td>
    <td><a class="a" href="delete.php?id=<?php echo $data['ID_B']; ?>">Já Corrigido</a></td>
  </tr>
</table>
<?php
while ($data = mysqli_fetch_array($records)) {
  ?>
}
```

(bug.php)

O administrador ainda pode criar contas para outros utilizadores, com ou sem permissão de administrador

Username:   
 Email:   
 Password:   
 Repetir Password:   
 Idade:   
 Permissão de administrador: Escolha uma opção ▼

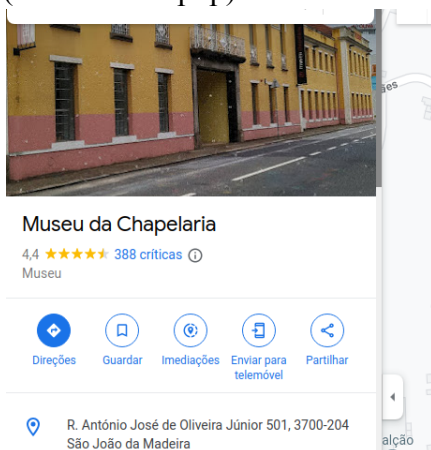
(/admin/criar\_admin.php)

Já quando o login é efetuado, mas o utilizador não tem permissão de administrador, este consegue ver os locais inseridos pelos administradores, e se clicar neles, é direcionado para a página do google maps que corresponde a aquele local.

Utilizei a API do google maps, por manipulação de string, neste caso, o URL.

<a href="#">museu da chapelaria</a> 	<a href="#">gergeg</a> 	<a href="#">31</a> 
<a href="#">345</a> 	<a href="#">Capela dos Ossos</a> 	

(/user/Locais.php)

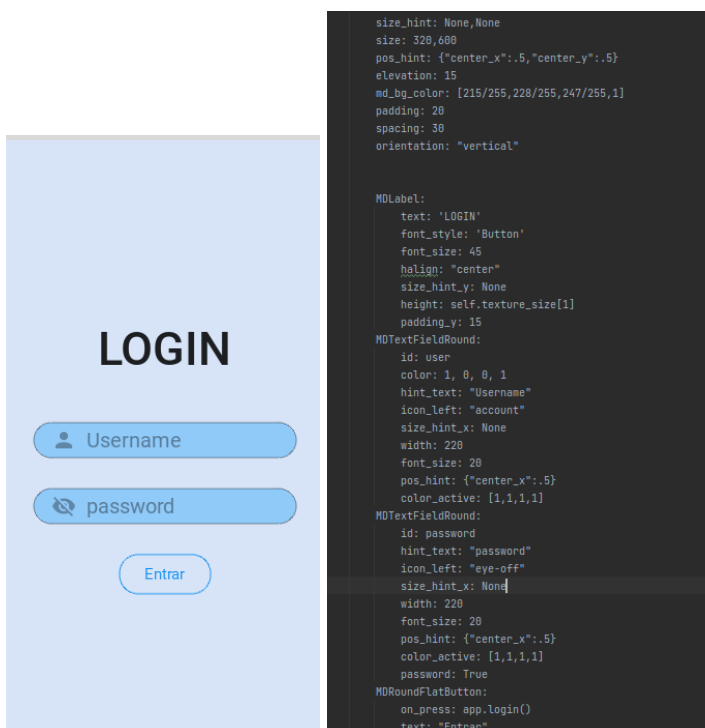


Esta tabela é feita com CSS Grid, para se auto ajustar à entrada de novos itens na grelha.

Desenvolvimento da Aplicação para telefone:

Para desenvolver a Aplicação para telefone, utilizei Python, e uma Framework do Python Kivy, KivyMD.

Primeiramente desenvolvi a interface do login em Kivy, ficheiro com extensão .kv



(login/login.kv)



```

class Login(MDApp):
    def build(self):
        return Builder.load_file("login.kv")

    def login(self):

        username = self.root.ids.user.text
        password = self.root.ids.password.text
        mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            passwd="",
            database="pap"
        )
        cursor = mydb.cursor()

        passe_enc = hashlib.md5(password.encode()).hexdigest()

        query2 = f"SELECT * FROM users where username = '{username}' and password = '{passe_enc}';"
        cursor.execute(query2)

        dados = cursor.fetchall()

```

(login/login.py)

Aqui podemos ver um exemplo de como tudo funciona.

Aqui carregamos o ficheiro login.kv (ficheiro da interface em Kivy) no python, posteriormente, se o utilizador clicar no botão Login, é executada a função login, onde está a ser obtida o username e a password. Depois, é criada uma ligação a base de dados e um Select, para ver se o utilizador existe.

Se o utilizador existir, é escrito num ficheiro TXT o username do mesmo, afim de saber qual o utilizador com login naquela hora.

Seguindo o mesmo raciocínio desenvolvi o registo, o formulário quando o utilizador se esquecer da Password e o formulário de verificação de utilizador.

The image shows three mobile app screens with a light blue background and rounded rectangular input fields.

- REGISTER:** Contains input fields for Username, email, password, repetir password, and idade. A 'Registrar' button is at the bottom.
- VERIFICAÇÃO:** Contains input fields for Email and Código. It has 'Enviar Código' and 'Verificar Conta' buttons.
- RECUPERAR PASSWORD:** Contains input fields for Username, nova\_password, repetir\_nova\_passw, and Código. It has 'Enviar Email' and 'Mudar a Password' buttons.

(register/register.kv) (verificação/verificação.kv) (esquecer\_password/esquecer.kv)

No que toca a utilizar o e-mail tive um contratempo, já que a google decidiu criar e ativar o OAuth 2.0. Tive que perceber como solucionar o problema, assim, a resolução passou por não entrar com a password original, mas sim com uma chave gerada apenas para este propósito, de conseguir que a aplicação tenha acesso ao email.

Na criação das funcionalidades da aplicação, comecei com os mais simples tais como consultas e alteração de dados da base de dados, como é o caso das Definições, Sugestão de Locais, Identificação de Bugs e Avaliação.

The image shows three mobile app screens with a blue header bar containing a hamburger menu icon and the text 'Conhece Portugal'.

- Settings Screen:** Contains text labels 'Distancia maxima a que o local possa estar', 'Autonomia da sua bicicleta (0 de n for eletr)', and a 'Guardar Configurações' button.
- Bug Reporting Screen:** Contains input fields for 'Tipo de local (restaurante, percurso, etc.)', 'Nome do local', and 'Localização do local'. It has an 'Enviar bug' button.
- Evaluation Screen:** Contains input fields for 'Tipo de local (restaurante, percurso, etc.)', 'Quantidade de estrelas que dás a este local', 'Comentario construtivo sobre o local', and 'Nome do local a avaliar'. It has an 'enviar avaliação' button.

(vdd\_APP/APP.kv)

O que me custou mais a fazer, sem dúvida, foi a funcionalidade de ir a algum lado de bicicleta e saber onde abastecer. Nessa funcionalidade eu peço o nome do local que o utilizador deseja ir e o programa faz o resto sozinho. Ou seja, informa sobre um percurso onde já estão presentes os locais de abastecimento.

```

local = self.root.ids.local_a_ir.text
if (local != ""):
    with open("../auxiliares/users.txt", "r") as myfile:
        data = myfile.readlines()
        user = listToString(data)
        mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            passwd="",
            database="pap"
        )
        cursor = mydb.cursor()
        query = "SELECT autonomia from users where username = '" + user + "';"
        cursor.execute(query)
        dados = cursor.fetchall()
        for aut in dados:
            aut_user = aut[0]
            print(str(aut_user)+"<-autonomia")

```

(vdd\_APP/APP.py)

Nesta imagem é visível, onde o programa obtém na variável local, o sítio onde o utilizador deseja ir. Posteriormente vai ler o ficheiro txt, onde está guardado o username do utilizador que está atualmente com o login efetuado. Depois, é criada uma ligação à base de dados, onde com um select conseguimos ver a autonomia da bicicleta desse utilizador.

```

if (aut_user == 0.0):
    local.replace(" ", "+")
    query = "https://www.google.com/maps/dir/?api=1&destination=" + local + "+WA&travelmode=bicycling"
    webbrowser.open(query)

```

(vdd\_APP/APP.py)

Aqui estamos a verificar se a autonomia do utilizador é 0, neste caso a bicicleta não é elétrica logo, ele não precisa de passar em nenhum posto de combustível, já que não precisa de abastecer a bicicleta, assim, vai diretamente ao destino.

```

elif (aut_user > 0.0):

    g = geocoder.api.ip("me")

    # coordenadas atuais
    localizacao_inicial = str(g.lat) + "," + str(g.lng)
    atual = {
        "latitude": g.lat,
        "longitude": g.lng
    }

    # converter autonomia de km pa coordenadas, 1 coordenada são aproximadamente 111 km.
    aut_coord = aut_user / 111

    # coordenadas do destino
    geolocator = Nominatim(user_agent="MyApp")
    destino = geolocator.geocode(f"{local}")
    destino = {
        "longitude": destino.longitude,
        "latitude": destino.latitude
    }

```

(vdd\_APP/APP.py)

Se a bicicleta for elétrica, ou seja, a autonomia for maior que 0, vamos ver a localização atual do utilizador através da utilização da biblioteca geocoder. Depois de obtermos a latitude e a longitude do utilizador, convertemos a autonomia em coordenadas e a partir deste momento vamos deixar de trabalhar com quilómetros (mas sim vamos trabalhar com coordenadas). Posteriormente, vamos obter as coordenadas do destino (latitude e longitude).

```

distancia_atual_destino_km = distanceInKmBetweenEarthCoordinates(float(atual["latitude"]), float(atual["longitude"]),
                                                                    float(destino["latitude"]), float(destino["longitude"]))

```

(vdd\_APP/APP.py)

Aqui estamos a calcular o geodésico entre as coordenadas do utilizador, e as coordenadas do destino. O cálculo de geodésico (geodésico – definido como conexão entre 2 pontos numa esfera). A Terra movimenta-se segundo um movimento geodésico que, no espaço curvo, conforme descrito na Teoria da Relatividade Geral de Einstein, consiste numa aparente trajetória retilínea, mas que na realidade se curva conforme as “dobras” do espaço-tempo.

```
def degreesToRadians(degrees):
    return degrees * math.pi / 180

def distanceInKmBetweenEarthCoordinates(lat1, lon1, lat2, lon2):
    earthRadiusKm = 6371

    dLat = degreesToRadians(lat2 - lat1)
    dLon = degreesToRadians(lon2 - lon1)

    lat1 = degreesToRadians(lat1)
    lat2 = degreesToRadians(lat2)

    a = math.sin(dLat / 2) * math.sin(dLat / 2) + math.sin(dLon / 2) * math.sin(dLon / 2) * math.cos(lat1) * math.cos(lat2)
    b = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))

    return earthRadiusKm * b
```

(vdd\_APP/APP.py)

O cálculo do geodésico é realizado da seguinte forma: a partir da latitude e longitude do local onde se encontra e da latitude e longitude do local para onde quer ir. Com base nesses dados é fornecida a distância, em km, desses dois pares ordenados de coordenadas geográficas.

```
distancia_atual_destino_coord = distancia_atual_destino_km / 111
if (distancia_atual_destino_coord > aut_coord):
    checkpoints = []
    while distancia_atual_destino_coord > aut_coord:
        # calcular área que a bike pode andar
        soma_catetos = 2 * aut_coord ** 2
        hipotenusa = math.sqrt(soma_catetos)
        meio_lado = hipotenusa / 2
        v1 = [float(atual["latitude"]) - float(meio_lado), float(atual["longitude"]) + float(meio_lado)]
        v4 = [float(atual["latitude"]) + float(meio_lado), float(atual["longitude"]) - float(meio_lado)]

        # obter bombas da área
        query_bombas = f"select * from bombas where lat_bomba > {v1[0]} and lat_bomba < {v4[0]} and long_bomba > {v1[1]} and long_bomba < {v4[1]}"
        cursor.execute(query_bombas)
        bombas = cursor.fetchall()
```

(vdd\_APP/APP.py)

Aqui, estamos a passar a distância entre o utilizador e o local para onde deseja ir, para coordenadas. Se a autonomia for menor que a distância, a bicicleta vai ter que passar por bombas, portanto, vamos calcular a área em que a bomba possa estar.

```

# calcular bomba mais perto do destino
distancia_bomba_mais_perto_do_destino = float("inf")
bomba_mais_perto_do_destino = None
for bomba in bombas:
    bomba = {
        "nome": bomba[0],
        "latitude": bomba[1],
        "longitude": bomba[2]
    }
    distBombaDestino_coord = distanceInKmBetweenEarthCoordinates(float(bomba["latitude"]),
                                                                    float(bomba["longitude"]),
                                                                    float(destino["latitude"]),
                                                                    float(destino["longitude"])) / 111

    if distBombaDestino_coord < distancia_bomba_mais_perto_do_destino:
        distancia_bomba_mais_perto_do_destino = distBombaDestino_coord
        distancia_atual_destino_coord = distBombaDestino_coord

        atual["latitude"] = bomba["latitude"]
        atual["longitude"] = bomba["longitude"]

        bomba_mais_perto_do_destino = bomba

# adicionar bomba mais perto do destino ao array de checkpoints

```

(vdd\_APP/APP.py)

Neste momento estamos a calcular o local da bomba mais perto do destino, já que ele não vai abastecer em todas as bombas que encontrar, por um lado não necessita e por outro, levaria a um percurso muito demorado e o algoritmo tornar-se-ia inviável.

```

# adicionar bomba mais perto do destino ao array de checkpoints
try:
    if checkpoints[-1]["latitude"] != bomba_mais_perto_do_destino["latitude"] or checkpoints[-1]["longitude"] != bomba_mais_perto_do_destino["longitude"]:
        checkpoints.append(bomba_mais_perto_do_destino)
    else:
        checkpoints = []
        break
except Exception as e:
    checkpoints.append(bomba_mais_perto_do_destino)

for c in checkpoints:
    print(c)
if len(checkpoints) == 0:
    print("Percurso impossível")
print(len(checkpoints))
if 0 < len(checkpoints) <= 7:
    print("")
    #como vai passar por varios locais temos de usar /
    inicio_url = "https://www.google.com/maps/dir"
    coord_inicio = str(localizacao_inicial)
    coord_checkpoint = ""
    aux = ""
    for checkpoint in checkpoints:
        coord_checkpoint = str(checkpoint["latitude"]) + "," + str(checkpoint["longitude"]) + "/"
        aux += coord_checkpoint
    coord_fim = str(destino["latitude"]) + "," + str(destino["longitude"])
    print(coord_inicio)
    print(aux)
    print(coord_fim)
    url = inicio_url + "/" + coord_inicio + "/" + aux + coord_fim + "/" + "data=!3m1!4b1!4m2!4m1!3e1"
    print(url)
    webbrowser.open(url)

```

(vdd\_APP/APP.py)

Nesta imagem estou a adicionar a bomba mais perto do destino, ao array checkpoints, que vão ser as bombas por onde ele tem que passar, e fazer com que o script construa o URL que vai ser usado no google maps, através de manipulação de string.

Ao adicionarmos parâmetros temos que usar o caractere “/” para as separar, e usamos no final do URL “data=!3m1!4b1!4m2!4m1!3e1” para dizer que queremos percorrer de bicicleta.

A API do google maps e a sua documentação foram elementos cruciais para a elaboração desta funcionalidade.

Entender como funcionam os URLs do google maps, no que toca á procura de locais, direções e mapas. Bem como entender sobre coordenadas geográficas e sobre a localização dos meridianos no globo foram importantíssimos para conseguir fazer a funcionalidade mais importante, que é a base de todo este projeto.

## Problemas encontrados e as suas resoluções:

Com este projeto tive vários problemas, quer na parte web, quer na aplicação mobile.

Na parte web, tive dificuldades em utilizar o bootstrap, já que nunca o tinha utilizado. Encontrei um curso chamado "CS50's Web Programming with Python and JavaScript 2020" que vi, para perceber como funcionava o bootstrap, a fim de poder-lo utilizar no meu projeto.

Outra das dificuldades que tive foi com que a aplicação mobile tivesse acesso às imagens que seriam inseridas pelos administradores na página web, e para isso teria de guardar as imagens na base de dados. Para guardar as imagens na base de dados, teria de converter a imagem para um ficheiro binário, que implicava uma perda de qualidade da imagem. Após muita procura percebi que os servidores têm armazenamento que pode ser utilizado para guardar as imagens.

Uma das dificuldades mais significativas foi conseguir por a imagem no servidor local, e para isso teria que pôr a imagem no diretório do servidor local (xampp). Para isso recorri a vários fóruns, vídeos e cursos online para resolver esse problema.

Já na aplicação mobile tive várias complicações na instalação do Kivy, pois o windows estava a dar muitas incompatibilidades com o PIP e o PYinstaller, logo mudei de sistema operativo no meu computador e pus uma distro do linux, chamada linux mint, já que poderia utilizar o PPA para instalar o Kivy.

Uma das maiores dificuldades foi aprender a sintaxe do Kivy e KivyMD. Superei essa dificuldade graças à documentação do Kivy e do KivyMD; "Curso de Python: Apps para Android, iOS, Linux, Win, e Mac" e curso da codemy.com de "Python Kivy GUI".

Outra das grandes dificuldades que tive foi para saber a distância entre 2 pares de coordenadas geográficas. Para resolver esse problema resolvi perguntar a um professor de geografia como fazê-lo, mas este apenas soube responder utilizando um mapa, no entanto, eu queria analiticamente. Após muita investigação cheguei ao cálculo do geodésico, isto é, o cálculo da conexão entre 2 pontos numa esfera, sendo a esfera o planeta terra. Isto porque a terra movimenta-se segundo um movimento geodésico que, no espaço curvo, conforme descrito na Teoria da Relatividade de Einstein, consiste numa aparente trajetória retilínea, mas que na realidade se curva conforme as "dobras" do espaço-tempo.

Uma grande dificuldade que tive foi conseguir realizar a funcionalidade de passar por vários pontos para chegar a um local de bicicleta, mas após muito esforço e persistência consegui resolver esse problema



## 9. Desenvolvimento futuro

Desde o início, este projeto foi pensado de forma a ser o mais simples e intuitivo possível. Porém, chegando agora ao fim vejo que houve coisas em que me precipitei um pouco, não dedicando tempo para a procura de softwares que desconhecia, que eram melhores para a realização deste projeto. Em relação a base de dados usei MySQL, uma base de dados relacional; mas, contudo, a minha base de dados poderia ser não relacional, visto que o campo do ID é chave primária da tabela "mãe" e está presente em quase todas as tabelas deste projeto. Para além disso, a base de dados que uso, está num servidor local, podendo estar num servidor, para toda a gente poder aceder e usufruir, sem ter que pôr a base de dados deste projeto no seu dispositivo. Daria também a liberdade se na aplicação mobile, houvesse uma funcionalidade para ver os locais (locais, restaurantes, locais para pernoitar e percursos) mais próximos do utilizador. Seria ainda interessante ser enviada uma notificação ao utilizador com sugestões de locais para ele visitar. Para além disso, como não consegui entender o mecanismo que o Kivy usa para mudar de ecrãs, não o fiz, mas pedia a quem tiver interesse, que o faça.

## 10. Avaliação

Considero que este projeto me influenciou de forma positiva, possibilitando-me ver a quantidade de softwares e ferramentas que tinha à minha disposição, o que me fez perceber ainda mais o porquê da informática ser tão dinâmica e estar em constante evolução. Aprender uma ferramenta nova para mim (Kivy, KivyMD), já que nunca a tinha usado foi um desafio muito interessante, que me fez perceber que quanto mais me aprofundar nesta matéria (informática) mais posso contribuir por exemplo uma empresa a escolher a melhor ferramenta a usar numa certa situação.

Considero que este projeto foi inovador tanto para mim, criador como para os postos de turismo, que pretendo que estejam em constante evolução.

## 11. Reflexão final / Conclusão

No final deste projeto posso concluir que na generalidade foi muito bom para a minha aprendizagem, quer na questão de aquisição de novos conceitos, quer na questão de autonomia, e de me obrigar a pesquisar e aprofundar os meus conhecimentos sobre um certo tema. Percebi que o mundo tecnológico está em permanente desenvolvimento e que esta circunstância vai fazer com que eu esteja atento às novidades, ou seja, em permanente atualização ao longo do meu percurso profissional.

O objetivo final, seria ter um website e uma aplicação mobile, onde o utilizador tivesse a percepção de onde estava e o que tinha por perto. Este teria acesso a lugares turísticos, restaurantes, hotéis, e percursos, para os que gostam de atividade física, que podem ser realizados a pé ou de bicicleta.

A principal ideia da aplicação é, com a ajuda da base de dados, não necessitar de constantes atualizações, já que se auto-atualiza com novos locais sempre que estes forem inseridos pelos administradores.

Apreendi bastante, pois tive de investigar várias linguagens e plataformas que não constam no currículo escolar, como o Python, onde tive de passar imensas horas em pesquisas exaustivas, na Framework Kivy e KivyMD, onde tive de me adaptar à sua sintaxe e a sua forma de fazer as coisas, mas no final de tudo valeu a pena.

Esta aplicação irá certamente contribuir para o meu futuro, tanto académico como profissional, como será a primeira de muitas.

## 12. Bibliografia

Abreu, L. (2012). HTML 5. 3. ed. Lisboa. Editora de informática.

Pereira, A. & Poupá, C. (2013). Linguagens Web. 5. ed. Lisboa. Edições Sílabo.

<https://developers.google.com/maps/documentation/urls/get-started>

<https://www.edx.org/course/cs50s-web-programming-with-python-and-javascript>

<https://www.youtube.com/watch?v=k4QCoS-hj-s&list=RDCMUFCB0dxMudkws1q8w5NJEAmw&index=1>

<https://kivy.org/doc/stable/>

<https://kivymd.readthedocs.io/en/latest/>

<https://github.com/kivymd/KivyMD>

<https://www.php.net/>

<https://stackoverflow.com/questions/5772769/how-to-copy-a-file-from-one-directory-to-another-using-php>

[https://www.w3schools.com/php/php\\_file\\_upload.asp](https://www.w3schools.com/php/php_file_upload.asp)

<https://www.w3schools.blog/php-file-upload>

<https://www.geeksforgeeks.org/write-a-code-to-upload-a-file-in-php/>

## 13. Anexos

