



Instituto Politécnico de Tomar
Escola Superior de Tecnologia de Tomar



Bike Sharing

Projeto apresentado ao Instituto Politécnico de Tomar
para cumprimento dos requisitos necessários à obtenção do
grau de Licenciado em Engenharia Informática

Alunos: João Francisco Fidalgo Valido 21870
Duarte Faria da Mota Gonçalves Valente 22180

Orientador: Prof. Luís Miguel Oliveira

2021/2022

Setembro

Índice

1.	Introdução	7
1.1.	Motivação	7
1.2.	Objetivos	8
1.3.	Desenvolvimentos	8
1.4.	Organização Relatório	9
2.	Estado de Arte	10
3.	Bicicletas Comunitárias	20
3.1.	Escolha de componentes	21
3.1.1.	Servidor	21
3.1.2.	Conectividade entre a bicicleta e o servidor	22
3.1.3.	Hardware para bicicleta	22
3.1.4.	Base de Dados	23
3.1.5.	Aplicação Mobile	25
3.1.6.	Website	26
3.2.	Perspetivas de uso	26
3.2.1.	Utilizador	26
3.2.2.	Gestor	27
3.3.	Conectividade entre a bicicleta e o servidor	29
3.3.1.	Localização e autorização	31
	Conectividade LoraWAN	32
3.3.2.	Autorização e estado de bicicleta	34
3.4.	Base de Dados	36
3.4.1.	Estrutura da Base de dados	36
3.5.	Servidor	38
3.5.1.	Conexão com a Base de dados	39
3.5.2.	Conexão com a TTN	40
3.5.3.	Interface entre as aplicações e website com a Base de Dados	42
3.6.	Aplicação Mobile	46
3.6.1.	Registar	46
3.6.2.	Login	49
3.6.3.	Welcome	49
3.6.4.	Mapa	51
3.6.5.	Aluguer	52

3.6.6.	Publicação nas Stores	55
3.7.	Website de Gestão	56
3.7.1.	Página Inicial	56
3.7.2.	Início de sessão	57
3.7.3.	Página de Entrada	58
3.7.4.	Adicionar Gestor	58
3.7.5.	Lista de Utilizadores	59
3.7.6.	Lista de Bicicletas	60
3.7.7.	Lista de Alugueres	61
3.7.8.	Mapa	62
3.7.9.	Estatísticas.....	63
4.	Conclusão e Trabalho Futuro	64
	Bibliografia.....	66

Índice de Figuras

Figura 1 - Rastreador GPS	16
Figura 2 - Exemplo cadeado mecânico	18
Figura 3 - Placa de Arduino MKR Wan 1300	23
Figura 4 - Arquitetura geral de funcionamento do projeto.....	28
Figura 5 - Arquitetura LoraWAN	29
Figura 6 - Módulo Ublox Neo-6m GPS	Erro! Marcador não definido.
Figura 7 - Esquema geral da placa de arduino	34
Figura 8 - Variáveis de ligação.....	32
Figura 9 - Função de conectividade à rede LoraWAN	33
Figura 10 - Variável para enviar através da TTN com as coordenadas	Erro! Marcador não definido.
Figura 11 - Código ativação/desativação LED	35
Figura 12 - Tabelas de Dados mongoDB.....	37
Figura 13 - Gráficos Criados no MongoDB	38
Figura 14 - Key de conexão onde inserimos a nossa palavra-passe fornecida pelo MongoDB	39
Figura 15 - Exemplo de conexão com a Base de Dados no lado do Servidor	39
Figura 16 - Conexão do servidor com a TTN	40
Figura 17 - Servidor assim que recebe uma mensagem da TTN.....	41
Figura 18 - Use case "Alugar Bicicleta"	43
Figura 19 - Use case 'Gestão'.....	44
Figura 20 - Ecrã "Registar"	47
Figura 21 - Ecrã "Verificação de E-mail"	47
Figura 22 - E-mail Verificado	48
Figura 23 - E-mail de Verificação	48
Figura 24 - E-mail não verificado.....	48
Figura 25 - Ecrã Login.....	50
Figura 26 - Ecrã Welcome	50
Figura 27 - Mapa	51
Figura 28 - Ecrã de Aluguer	52
Figura 29 - Ecrã Terminar aluguer	53
Figura 30 - Ecrã de Bicicleta em uso	53
Figura 31 - Ecrã de Resumo	54
Figura 32 - Cálculo do preço	55
Figura 33 - Página Inicial	57
Figura 34 - Página de Início de Sessão	57
Figura 35 - Página de Entrada.....	58
Figura 36 - Página de Criação de conta.....	59
Figura 37 - Lista de Utilizadores	60
Figura 38 - Lista de bicicletas	61
Figura 39 - Lista de Alugueres	61
Figura 40 - Mapa	62

Glossário

The Things Network (TTN) – é um servidor de rede LoRaWAN, sendo o componente crítico para qualquer rede Lora.

LoRa – Long Range (“Longo alcance”), é um protocolo de comunicação de baixo consumo de energia e longo alcance para dispositivos de radiofrequência. A sua modulação é excelente para comunicações P2P (“Point to Point”, Ponto a Ponto).

LoRaWAN - É o protocolo que estabelece a comunicação através do meio físico LoRa, nele são definidos todos os agentes que participam da rede, bem como parâmetros dos pacotes.

App – aplicação para ambientes móveis.

BD - Base de Dados.

IPT - Instituto Politécnico de Tomar.

Payload – Os dispositivos Lora enviam suas mensagens como bytes em vez de objetos inteiros JSON, para economizar largura de banda. Mais especificamente no nosso caso: Mensagem enviada através do Arduíno com destino ao Servidor TTN, e por sua vez encaminhado para o nosso Servidor.

Downlink - Mensagem enviada através do nosso Servidor com destino à TTN, e por sua vez encaminhado para a Placa de Arduíno.

1. Introdução

1.1. Motivação

A mobilidade continua a ser um problema das cidades. A disponibilização de vias adequadas à circulação de bicicletas, associado a disponibilização de bicicletas de uso partilhado, é vista como um contributo valioso para a resolução deste problema.

São várias as soluções de disponibilização de bicicletas de uso partilhado em funcionamento um pouco por todo o lado, tendo sido identificados dois tipos de problemas associados a estes serviços.

O primeiro tipo de problema está relacionado com a utilização responsável. O segundo, com a gestão das bicicletas, por exemplo assegurar que existem bicicletas disponíveis onde são necessárias.

Em termos de resoluções, tendo em conta o primeiro problema mencionado no parágrafo anterior, através da aplicação mobile (explicada mais à frente), o utilizador ao efetuar um alugar, o seu e-mail fica registado na base de dados, juntamente com o identificador da bicicleta, o que em caso de danos, torna possível a identificação do responsável. Para o segundo problema, será desenvolvido um site de gestão para o gestor, (explicado mais à frente), com estatísticas que permitam ao gestor analisar as mesmas e efetuar as mudanças que achar necessárias para uma melhor utilização por parte dos utilizadores.

1.2. Objetivos

Este projeto tem como objetivo o desenvolvimento de uma aplicação e um website. Sendo a primeira uma aplicação de autenticação destinada aos utilizadores, que após criação de conta e a sua respetiva autenticação, permita visualizar e alugar bicicletas na cidade de Tomar.

O website tem como propósito disponibilizar ao gestor do serviço de aluguer de bicicletas, informações necessárias para gerir o serviço, tais como estatísticas acerca do uso e da localização das bicicletas.

1.3. Desenvolvimentos

Tanto para a gestão de bicicletas, por parte do gestor, como para a autorização e criação de utilizadores, para visualizar e alugar as mesmas, foi necessário o desenvolvimento de um Website e uma aplicação Mobile, um servidor e um hardware que desbloqueia e recolhe informação acerca da localização das bicicletas.

1.4. Organização Relatório

O relatório está dividido em cinco capítulos.

- 1.** A Introdução apresenta a motivação, os problemas e as soluções desenvolvidas.
- 2.** No capítulo Estado da Arte são apresentados projetos de gestão e aluguer de bicicletas, e as tecnologias mais utilizadas
- 3.** O capítulo Bicicletas Comunitárias descreve a arquitetura do projeto, a sua concretização e os resultados obtidos.
- 4.** O capítulo quatro apresenta as conclusões e identifica aspetos que podem ser explorados no futuro para melhorar as soluções apresentadas.

2. Estado de Arte

As preocupações ambientais, nos últimos anos, têm vindo a provocar o crescimento no setor dos transportes elétricos assim como uma maior adesão a transportes não motorizados. Assim, o aluguer destes veículos tem feito um grande sucesso dentro das grandes cidades, onde também os próprios municípios têm apostado em melhorar as condições de acessibilidade com a construção de mais passeios, ciclovias, zonas de lazer e restauração.

Para uma melhor perceção do mercado, foi realizada uma pesquisa sobre empresas que fornecem este tipo de serviços para analisar a maneira como os prestam e as soluções que utilizam para resolver os problemas associados tanto ao aluguer de bicicletas como à gestão e manutenção destas. Dentro desta pesquisa é possível destacar algumas das mais conhecidas em Portugal, como por exemplo:

- **GiRA**

Partilha bicicletas na cidade de Lisboa e é operada pela Empresa Municipal de Mobilidade e Estacionamento de Lisboa. Foi lançada em setembro de 2017, após uma fase piloto na freguesia do Parque das Nações sob o nome Lisboa Bike Sharing. Possui um horário de funcionamento de vinte horas diárias, estando disponíveis para utilização das 06:00 às 02:00.

Para poderem utilizar as bicicletas os utilizadores devem instalar a aplicação para dispositivos móveis GiRA e criar a sua conta de utilizador, conta esta na qual devem subscrever um dos passes disponíveis (diário, mensal e anual) para estarem assim aptos a utilizar estas bicicletas.

De forma a garantir a segurança das bicicletas, todas estas encontram-se bloqueadas em docas espalhadas pela cidade, que apenas as desbloqueiam caso os utilizadores as desbloqueiem na aplicação. Isto garante que os utilizadores sejam responsáveis com as bicicletas alugadas, uma vez que ao estarem

associadas às suas contas obriga que as devolvam nas docas em bom estado e em segurança. Para além disto as bicicletas contam também com um sensor GPS, permitindo assim a sua localização em tempo real.

• Lisbon Bike Rentals

Partilha também bicicletas na cidade de Lisboa e possui a maior frota de bicicletas de Portugal. No entanto o intuito desta empresa passa pelo aluguer de bicicletas de forma individual.

Para efetuar estes alugueres os utilizadores devem aceder ao website da empresa e criar conta onde podem escolher o modelo da bicicleta que pretendem alugar assim como a data de aluguer. Ainda é dada a possibilidade de a bicicleta ser entregue ao domicílio ou levantada em loja física. Por fim, o pagamento do aluguer pode ser feito também no website ou na loja assim que a bicicleta for levantada. O aluguer para além da bicicleta, dispõe ainda de um mecânico profissional pronto para efetuar manutenção nas bicicletas.

Em relação à segurança das bicicletas, esta é pouca ou nenhuma uma vez que não contam com nenhum dispositivo de rastreamento ou sistema antifurto. Os seus clientes necessitam apenas de assinar uma autorização que ficam encarregues da segurança da bicicleta.

A adição de um aplicativo móvel como complemento ao seu serviço pode ser algo a implementar no futuro para a empresa uma vez que iria facilitar a marcação dos alugues assim como a chamada do profissional disponibilizado.

• Rent Riders

Assim como a Lisbon Bike Rentals o intuito desta empresa é o aluguer de bicicletas “comuns” aos clientes e não a partilha de bicicletas. Rent Riders

permite o aluguer de modelos citadinos, de estrada ou de montanha e os alugueres começam nos 22,73€ por dia.

Para efetuar os alugueres os seus clientes entram em contacto com a empresa pelo site da mesma ao criarem uma conta de utilizador. Assim, podem efetuar o aluguer e levantar a bicicleta na loja física da Rent Riders.

Assim como a Lisbon Bike Rentals as bicicletas não contão com nenhum sistema de segurança que as proteja enquanto estão a ser alugadas.

• Bluedragon

Localiza-se no Porto e possui uma loja física onde os utilizadores se devem deslocar para efetuarem o aluguer em um de dois modelos disponíveis. A reserva das bicicletas é possível ser feita através do site da empresa, mas o processo de aluguer pode apenas ser efetuado na loja. O aluguer inclui ainda para além da bicicleta, um capacete, um cadeado, assistência caso seja necessária e um mapa.

Assim como as duas empresas apresentadas anteriormente, a Bluedragon também não apresenta nenhuma solução relativamente á segurança das bicicletas, uma vez que estas ficam totalmente aos cuidados do utilizador sem a possibilidade de saber onde se encontram ou o se não apresentam nenhum problema.

A opção de colocar um sensor GPS nas bicicletas poderia ser uma boa adição a estas bicicletas uma vez que iria aumentar o fator segurança e permitir a análise de alguns dados estatísticos.

• OFO

Ofo[5], foi uma empresa de partilha de bicicletas com sede em Pequim fundada em 2014. Em 2017, contavam com cerca de 10 milhões de bicicletas em 250 cidades e 20 países, foi avaliada em até US\$ 2 bilhões e chegou a ter mais de 62,7 milhões de

utilizadores ativos mensais. Em 2018, a Ofo anunciou uma redução maciça nas operações. Já em 2020, enfrentando uma grande dívida, a empresa deixou de operar com o aluguel de bicicletas.

Para alugar uma bicicleta o utilizador tinha de instalar o aplicativo móvel da empresa e após a criação da sua conta, era possível então procurar uma bicicleta nas proximidades, uma vez que esta contava com um recetor GPS, e alugá-la.

Esta empresa teve um grande sucesso pois implementou um sistema em que as bicicletas não necessitavam de ser entregues numa estação, uma vez que o seu bloqueio era feito na própria bicicleta por um dispositivo de bloqueio mecânico. Outro fator que impulsionou este grande alcance foi também a possibilidade de recolher informações sobre as bicicletas e otimizar os seus serviços.

• Mobike

Mobike foi fundada pela Beijing Mobike Technology Co. É uma empresa de *Bike Sharing* que possui sede em Pequim, China e é, pelo número de bicicletas, a maior operadora de bicicletas partilhadas (para aluguel) do mundo, tornando Xangai a cidade com mais partilha de bicicletas do mundo em dezembro de 2016.

À semelhança da OFO, a Mobike também não possui a necessidade de colocar as suas bicicletas em estações de bloqueio, o que facilita o acesso às bicicletas uma vez que estas podem-se encontrar e ser deixadas em qualquer lado.

Os seus utilizadores necessitam de ter a aplicação móvel da Mobike instalada e uma conta de pagamento associada á aplicação para poderem procurar e escolher uma bicicleta para alugar. Esta por sua vez assim que autorizada por parte da Mobike será desbloqueada para o utilizador. Após o uso, é também feito através da aplicação o término do aluguer.

• LimeBike

A Neutron Holdings, atua sob o nome de Lime e é uma empresa de transporte com sede em São Francisco, EUA. Ele opera scooters elétricos, bicicletas elétricas e ciclomotores elétricos em várias cidades ao redor do mundo.

O sistema oferece veículos sem estação que os usuários encontram e desbloqueiam por meio de um aplicativo de smartphone que sabe a localização dos veículos disponíveis via GPS.

A recolha dos dados destas bicicletas permite também uma enorme fonte de informação para os gestores da aplicação, uma vez que tem acesso a informações como: rotas feitas pelas bicicletas, localizações de maior concentração de bicicletas, tempos de aluguer ou até críticas feitas pelos utilizadores.

• Jump

Jump compartilha scooters e bicicletas elétricas sem estação, opera nos Estados Unidos, Nova Zelândia, Canadá, França, Alemanha, Portugal, Reino Unido, México e Austrália.

Os seus utilizadores conseguem procurar e desbloquear as bicicletas ao instalar a aplicação Uber e o pagamento do aluguer é feito pela conta da dos utilizadores na Uber. Cada bicicleta Jump tem um motor elétrico de 250 watts que aciona a roda dianteira, motor este que por consumir alguma energia necessita de que funcionários da Jump trocam as suas baterias a cada três dias.

Para realizar o seu bloqueio, as bicicletas devem ser travadas em um bicicletário na calçada.

Resumo da pesquisa de empresas com serviços semelhantes

É possível assim concluir que apesar de existirem várias opções no mercado português, a envolvente tecnológica não é explorada por muitas destas empresas que podiam agilizar muito mais o processo de aluguer das suas bicicletas assim como facilitar a gestão e manutenção das mesmas. A pouca segurança das bicicletas também é um fator a ter em conta, uma vez que as bicicletas de pelo menos em três das quatro empresas nacionais analisadas não conta com qualquer dispositivo que rastreamento ou de bloqueio.

Confirma-se então, que por outro lado, a componente tecnológica é um dos principais focos no negócio do Bike Sharing Services por parte das grandes empresas estrangeiras, permitindo estas alcançar grandes números de utilizadores por todo o mundo. Sendo que este grande abrangimento do negócio seja também resultado da análise dos dados recolhidos por parte dos seus gestores que otimizam os seus serviços com intenção de aumentarem a aderência ao serviço. Assim, podendo assim também, analisar melhor quais as tecnologias mais utilizadas e mais eficientes nestes sistemas de alugues de bicicletas com o objetivo de encontrar o que se pode adequar mais ao projeto Bike Sharing.

2.1. Resumo geral e Tecnologias utilizadas

A localização das bicicletas é um dos serviços mais procurados pelos utilizadores e pelos gestores. Sendo por isso comum equipar as bicicletas com rastreadores GPS [Figura 1].



Figura 1 - Rastreador GPS

Para transmissão destes dados não é comum serem utilizados sistemas sofisticados de ligação internet ou de telecomunicações por satélite. Um simples sinal de rede móvel através de um cartão SIM [6], que é fácil de conseguir quase em todo o lugar, mesmo dentro de edifícios, serve para o envio destas informações. A localização pode ser enviada em apenas alguns bytes assim como um identificador para o número do remetente/bicicleta.

A utilização responsável e a taxação do serviço envolvem a autenticação e autorização [2] dos utilizadores pois para terem acesso a certos recursos são necessárias algumas etapas de segurança adicionais. A autenticação é necessária para validar com segurança a identidade do sujeito e é um precursor crucial para a autorização. As políticas de autorização são iniciadas após a conclusão do processo de autenticação. O processo de autorização determina então se os utilizadores estão aptos para utilizarem os serviços disponibilizados.

Em termos de autenticação para segurança do próprio utilizador, existem várias ferramentas para melhorar processo de autenticação mantendo assim os seus dados seguros. A autenticação [8] MFA (Multi-Factor Authentication) utiliza diferentes meios de autenticação. Durante o login com nome de utilizador e senha, o utilizador deve fornecer um código de acesso único que a aplicação envia para o seu email ou telemóvel, fornecendo assim um nível mais alto de garantia durante a etapa de autenticação para melhorar a segurança. Outra opção é a autenticação biométrica que é um processo de segurança que se baseia nas características biológicas únicas dos indivíduos para verificar se eles são quem dizem ser. Os sistemas de autenticação biométrica comparam características físicas ou comportamentais com dados armazenados, confirmados e autênticos numa base de dados para serem assim comparados no momento da autenticação.

Em relação à segurança das bicicletas existem várias abordagens para prevenir o seu roubo, contando todas elas com uma série de equipamentos como:

- MCUs de 32 bits para coordenar e fazer a gestão do processamento de todos os dados captados pelos sensores da bicicleta, permitindo assim organizar e coordenar todo o sistema necessário ao funcionamento da bicicleta;
- Antena Bluetooth [3], que permite o desbloqueio seguro da bicicleta de forma eficaz. Este desbloqueio é feito através da partilha de uma chave criptográfica com o utilizador por meio da aplicação movel, que o possibilita ligar o Bluetooth no seu telemóvel e desbloquear a bicicleta na aplicação. Esta chave expirará automaticamente após o check-out, protegendo assim a privacidade do utilizador e a bicicleta. Uma vantagem desta comunicação é ser também uma comunicação de baixo consumo energético, o que por sua vez reduz o tamanho da bateria necessária.
- NFC (Near Field Communication) Tags para identificação da bicicleta na aplicação uma vez que se pode encontrar um grande aglomerado de bicicletas, tornando difícil identificar qual a bicicleta que se pretende alugar.
- MEMS (MicroElectro-Mechanical Sistemas) que são pequenos sensores que servem para detetar adulterações na bicicleta, permitindo assim os seus

gestores monitorar as bicicletas e em caso de necessidade enviar um técnico ao local onde se encontra a bicicleta para efetuar reparações.

- Cadeados mecânicos ativados por um sinal elétrico para bloquear as bicicletas de não ser necessário colocar a bicicleta em locais pré-definidos [Figura 2].



Figura 2 - Exemplo cadeado mecânico

Existem ainda outros componentes que podem estar presentes nas bicicletas como: baterias para fornecer energia a outros componentes da bicicleta ou para alimentar um possível motor elétrico caso a bicicleta seja elétrica, painéis solares para carregar estas baterias e ainda acionadores de motor para permitir o bloqueio das bicicletas elétricas.

Para efetuar comunicação entre a bicicleta e a rede, em alternativa à utilização da rede móvel uma vez que esta pode ser dispendiosa, podem ser utilizadas tecnologias tais como LoRaWAN [1] que permite o envio o envio de mensagens sem custos a partir de dispositivos com capacidade de processamento e de armazenamento de energia baixos.

No entanto, o desempenho das redes LoRa pode ser afetado pelo número de dispositivos conectados à mesma gateway, devido ao aumento do atraso e da taxa de pacotes perdidos.

Relativamente á segurança, LoRaWAN utiliza duas chaves criptográficas de 128 bits para proteger os dados enviados dos dispositivos. Uma chave criptográfica faz a autenticação do dispositivo na rede e a outra na aplicação de back-end.

Em relação ao seu alcance, consegue efetuar conexões a longa distância caso se verifique uma boa linha de visão, logo o posicionamento do gateway é crítico e deve ser estudado cuidadosamente para assegurar uma boa cobertura num ambiente urbano.

3. Bicicletas Comunitárias

Para conseguir resolver os problemas descritos nos objetivos mencionados na Introdução, foi necessário desenvolver uma aplicação e um website, sendo o website apenas para o gestor ter acesso a toda a informação acerca, tanto dos utilizadores, como das bicicletas e alugueres das mesmas. E a aplicação, destinada a todos os utilizadores com o propósito de alugarem bicicletas disponíveis e distribuídas pela cidade de Tomar.

Tendo em conta o propósito dos desenvolvimentos referidos em cima, foi decidido que a aplicação para o gestor seria desenvolvida em formato Website, para o mesmo conseguir aceder a todos os dados e estatísticas que desejar consultar. No caso dos utilizadores, o que faria mais sentido seria desenvolver uma aplicação Mobile, na qual os utilizadores teriam de criar conta, e por sua vez, teriam a opção de escolher uma bicicleta do seu agrado, (i.e. uma bicicleta que estivesse próxima), para alugar e utilizar a mesma. A aplicação em formato mobile seria o mais adequado pois os utilizadores, ao ver uma das nossas bicicletas bloqueadas na rua, tudo o que teriam de fazer para as alugar seria: efetuar o download da aplicação, criar uma conta, identificar a bicicleta que estão a visualizar, na aplicação, e por sua vez alugá-la.

Para concretizar as ideias descritas no parágrafo anterior, foi necessário a utilização de mais 4 componentes:

- Uma Base de Dados, para guardarmos todo o tipo de dados necessários dos utilizadores, bicicletas e alugueres;
- Hardware para desbloquear e localizar a bicicleta. Este hardware deve ser de baixo custo e baixo consumo.
- Um Servidor TTN para realizar o envio e receção de mensagens da placa arduíno;
- E por fim um Servidor com o propósito de fazer a interligação de todos os componentes descritos anteriormente.

De forma geral, tanto a aplicação, como o website, necessitam de executar algumas funções, como por exemplo o registo de utilizadores, o aluguer de bicicletas, etc. O que só se torna possível graças ao back-end, tanto do site como da aplicação mobile, desenvolvido no servidor, o qual, por sua vez necessita de consultar ou atualizar dados já existentes das tabelas de registos, os quais se encontram armazenados numa base de dados. O próprio hardware para as bicicletas, necessita de estar conectado ao servidor, para enviar as coordenadas e por sua vez receber mensagens que autorizem o uso da bicicleta (i.e. bloqueio e desbloqueio). A comunicação entre Servidor e o Hardware, é suportada pela TTN [17].

Que através das mensagens enviadas, as coordenadas são recebidas no servidor, e o estado do LED é recebido no hardware através de downlinks (mensagem enviadas do servidor com destino ao hardware/end-device).

3.1. Escolha de componentes

Aqui serão descritas e explicadas as razões da escolha dos componentes integrados na topologia geral do projeto [Figura 4].

3.1.1. Servidor

Para permitir o funcionamento de todas as componentes do projeto é necessário um servidor que interligue todas estas partes. Para tal foi decidido que o ambiente de programação a utilizar seria Node.js [25][26] pois utiliza maioritariamente JavaScript como linguagem de programação. Esta decisão deve-se também á utilização de React Native [15] para o desenvolvimento das aplicações moveis, pois React Native também utiliza JavaScript como principal linguagem de programação e é bastante popular, possuindo assim uma grande comunidade de suporte.

Para alojar este servidor foi utilizado o Heroku [29], um ecossistema de serviços na Cloud e de alta qualidade que possibilitou o desenvolvimento do projeto, mais rápido, pois não seria necessário aprender sobre configuração de servidor ou de

gerenciamento de rede e ao mesmo tempo utilizar serviços confiáveis que fornecem as funcionalidades necessárias.

3.1.2. Conectividade entre a bicicleta e o servidor

A primeira escolha foi em relação à rede que seria utilizada para se tornar possível estabelecer a comunicação entre as bicicletas e o resto do sistema. Foram encontradas as seguintes hipóteses de redes: 4G, SigFox, e a LoraWAN. Não foi necessária uma busca intensiva para escolher uma das 3 opções, pois após algum tempo de pesquisa e comparação, concluiu-se que para utilizar tanto a rede 4G como a SigFox seria necessário associar alguns custos mensais ao projeto, ou até mesmo associar a uma operadora para que fosse disponibilizado o acesso à rede em questão, já no caso da rede LoraWAN [17], existe cobertura na cidade de Tomar onde estão instalados dois Gateways no campus do IPT. Logo não existiriam custos adicionais nesse aspeto, o que facilitou bastante a escolha.

3.1.3. Hardware para bicicleta

Tendo, a rede LoraWAN, sido a nossa rede de comunicação escolhida, agora o próximo passo, a escolha do dispositivo final (“End device”). Para esta etapa foi necessário pensar num tipo de dispositivo com a capacidade de ser programável, com comunicação à rede escolhida, e que fosse possível obter as coordenadas geográficas do mesmo.

Depois de um pequeno “brainstorming” com o professor orientador do projeto, ficou decidido que a melhor escolha de dispositivo final, para implementar no projeto, seria a Placa de Arduíno MKR Wan 1300 [19][Figura 3], (contava também com a ligação e antena para efetuar a comunicação), devido ao facto de na mesma já estar integrado a conectividade com a rede LoraWAN, o que facilita a programação do arduíno no aspeto de enviar e receber dados utilizando a rede escolhida. A placa de arduíno foi fornecido pelo professor.

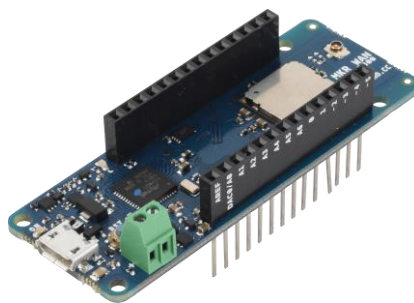


Figura 3 - Placa de Arduino MKR Wan 1300

3.1.4. Base de Dados

Para concretizar o objetivo do projeto foi fundamental a escolha de uma base de dados para permitir o funcionamento da aplicação mobile, do website de gestão da aplicação e até mesmo do Arduino, pois estas necessitam de vários dados para executarem o seu propósito e servirem tanto os clientes como os seus gestores. Para tomar esta escolha foram analisadas as diferentes opções existentes e qual poderia se adequar mais ao projeto.

A opção mais comum para este tipo de exigências são Base de Dados relacionais, pois representam os dados em tabelas de forma direta e intuitiva. Estas Bases de Dados relacionais permitem assim relacionar estas tabelas os seus dados, tornando o acesso à informação mais complexo e interligado. Uma desvantagem deste tipo de Base de Dados é também o crescimento da sua complexidade conforme o número de dados aumenta.

Uma hipótese de utilização de uma Base de Dados deste tipo seria o MySQL que utiliza a linguagem SQL como interface e é atualmente um dos sistemas de gestão de base de dados mais populares, contando com mais de 10 milhões de instalações pelo mundo. O facto de já termos trabalhado com este sistema no passado também era um ponto forte para a sua escolha.

Analizados também outros sistemas de gestão de base de dados como PostgreSQL, Microsoft SQL Server, Oracle, entre outros, mas nenhum parecia trazer

vantagens em relação ao MySQL uma vez que não tínhamos experiência com estas plataformas.

Foi também posta a opção de utilizar uma Base de Dados não relacional, uma vez que os dados são organizados em coleções e armazenados em estruturas similares a JSON, o que facilita na programação, uma vez que JSON é das estruturas de dados mais comuns, especialmente em linguagens como JavaScript, linguagem essa que foi utilizada para desenvolver a aplicação movel(react-native) o website(react) e o servidor(node.js), sendo este o principal fator que levou á sua escolha, assim como não existir a obrigação um utilizar um modelo relacional já que as verificações de todas as operações feitas aos dados são possíveis de realizar por parte do servidor.

Surgiu então, nesta fase, a opção da utilização do MongoDB, um software open source e multiplataforma de gestão de base de dados, escrito em C++. MongoDB ao ser uma Base de Dados não relacional e disponibilizar a utilização do serviço de cloud gratuito, aliado á simplicidade de armazenamento e gestão dos dados e permitir também a criação de gráficos estatísticos, necessários ao desenvolvimento do website de gestão, passou então a ser a escolha para desempenhar o papel tão importante de base de dados no projeto.

3.1.5. Aplicação Mobile

Esta aplicação tinha como principal objetivo a autenticação dos utilizadores, para que os mesmos tenham a oportunidade de alugar bicicletas espalhadas pela cidade de Tomar. Mas antes de qualquer desenvolvimento, era necessário escolher qual ambiente de desenvolvimento seria utilizado para construir a aplicação. Escolha essa que, após alguma pesquisa, se resumiu às seguintes hipóteses: Android Studio, Apache Cordova e React Native.

A primeira hipótese, Android Studio, foi a primeira ideia, devido ao facto de já termos utilizado este ambiente em outra unidade curricular, o que era uma grande vantagem, pois não seria preciso fazer uma introdução do zero à aplicação, contudo tinha uma grande desvantagem, apenas desenvolve para smartphone com o sistema operativo Android. Em prol do sucesso da nossa aplicação, tendo como objetivo desenvolver a aplicação, tanto para Android como para iOS, o que, como o nome indica, este ambiente permite apenas o desenvolvimento de aplicações para o sistema Android, o que obrigou a procurar por outras soluções que abrangessem ambos os sistemas.

Após alguma pesquisa, foram conseguidos dois resultados: Apache Cordova e React-Native, já tínhamos ouvido falar dos dois, mas apesar de ambos serem uma excelente opção para cumprir os requisitos, acabou por se escolher utilizar a biblioteca de JavaScript, React-Native, por 2 motivos: primeiro porque através da mesma linguagem era possível desenvolver o website para gestor (explicado mais à frente), e como segundo motivo, pelo simples facto de também já termos utilizado a mesma noutra unidade curricular, o que nos deixou um pouco mais confortáveis para desenvolver uma aplicação do zero.

3.1.6. Website

Para garantir o bom funcionamento da aplicação mobile é essencial uma plataforma de gestão onde os responsáveis pela aplicação possam ter acesso a todo o tipo de dados e informações presentes na Base de Dados. Assim, como nas outras componentes, foi necessária a escolha de uma framework para o seu desenvolvimento. Esta uma escolha fácil, pois tendo em conta as escolhas anteriores explicadas, o ambiente de programação que se adequava mais era o React.

React utiliza maioritariamente JavaScript como linguagem de programação e foi desenvolvido pelo Facebook, e permite criar rapidamente e facilmente interfaces para websites e aplicações.

Para aceder aos conteúdos deste website é necessário ter conta de gestor, contas essas que apenas podem ser criadas por outro gestor já registado e com permissões para tal.

Para alojar o servidor utilizamos novamente o Heroku pois como é o mesmo utilizado para alojar o nosso servidor e a experiência com a plataforma tinha sido bastante positiva ficou decidido que esta seria também a melhor opção para o website de gestão.

3.2. Perspetivas de uso

Aqui será apresentado de uma forma resumida a perspetiva de uso tanto do utilizador como do gestor.

3.2.1. Utilizador

Da perspetiva do utilizador, tudo começa na aplicação mobile, na qual o utilizador cria a sua conta, ao registar um e-mail, será enviado para o mesmo um e-mail de verificação, de maneira a identificar e responsabilizar o utilizador por ações futuras. As informações todas do utilizador (no caso da password é diferente, pois a

mesma é guardada em formato hash, para que apenas o utilizador tenha conhecimento da mesma), são encaminhadas para o servidor, que por sua vez as guarda na base de dados. Tendo o utilizador verificado o seu e-mail, passa agora a constar dos registos de utilizadores, o que lhe dá acesso a efetuar o login na aplicação. O utilizador já registado e com login efetuado tem acesso ao ecrã “Welcome” onde tem a hipótese de prosseguir para o mapa de bicicletas, que sempre que é carregado, é atualizado com a informação mais recente acerca dos atributos das bicicletas, principalmente da sua localização. Localização essa fornecida através do hardware que consta de cada bicicleta, que para além de enviar as coordenadas para o servidor, consegue também simular o bloqueio da bicicleta através de um LED. Caso a bicicleta se encontre disponível para usar, o utilizador é registado juntamente com a bicicleta escolhida e formam assim um novo registo na tabela de alugueres. Assim que o utilizador terminar o aluguer, o mesmo será atualizado e apresentará as informações de custo do aluguer, (foi nos instruído para não nos preocuparmos com o método de pagamento, por essa mesma razão não oferecemos nenhuma maneira de o utilizador cobrar esse custo).

3.2.2. Gestor

Em termos da aplicação Web destinada para o gestor, o mesmo tem a possibilidade de efetuar login no site, ganhando dessa maneira acesso a todo o conteúdo disponível do site, desde as tabelas dos utilizadores, bicicletas e alugueres das mesmas, criação de outras contas de gestor, adicionar bicicletas, até aos separadores de estatísticas no qual o gestor tem a possibilidade de analisar qual será o melhor local e hora para colocar as bicicletas, com o propósito de chegar ao máximo de clientes possível.

Muito resumidamente é isto que é pretendido desenvolver, e mais à frente serão apresentados melhores e mais aprofundados esclarecimentos de cada componente individualmente.

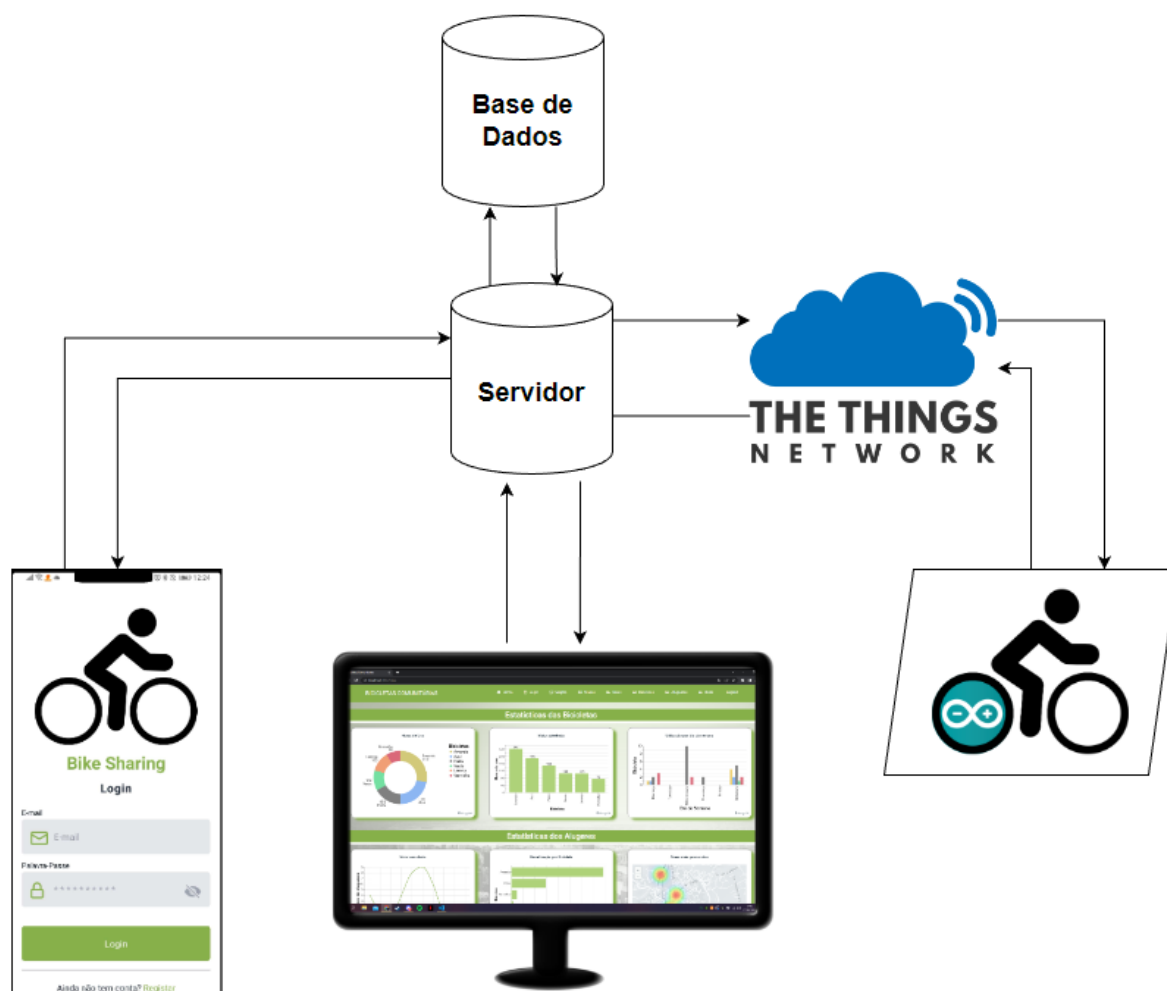


Figura 4 - Arquitetura geral de funcionamento do projeto.

3.3. Conectividade entre a bicicleta e o servidor

Para além do custo zero, no caso deste projeto, a rede LoRaWAN tem outras vantagens, como por exemplo o baixo custo devido ao facto de apenas um Gateway conseguir suportar vários dispositivos finais ou “End Devices”, estes são dispositivos LoRaWAN remotos que enviam e recebem informação dos Gateways que representam basicamente as antenas de conexão, as informações trocadas com os dispositivos são enviadas pela internet para o “Network Server” que tem como função tratar do encaminhamento de um certo dispositivo para a sua respetiva aplicação e vice-versa. Por fim, as informações chegam ao “Application Server” [Figura 6], que representa a plataforma onde são finalmente exibidas. Uma rede LoRaWAN pode ter mais de um “Application Server” uma vez que os seus dados podem ter várias finalidades. Como vantagens temos o seu baixo consumo o que permite aos “End Devices” serem usados com bateria por vários anos, tendo ainda como vantagem o seu alcance, que consegue atingir até 10km em áreas rurais. Passando agora para as desvantagens, a rede LoRaWAN apresenta um tamanho de pacote baixo, isto é, consegue apenas transmitir alguns bytes por mensagem, tem ainda como desvantagem uma alta latência, isto é, a comunicação não é instantânea, tendo ainda como desvantagem o limite de utilização responsável (i.e. por dia apenas é possível enviar um número limitado de mensagens).

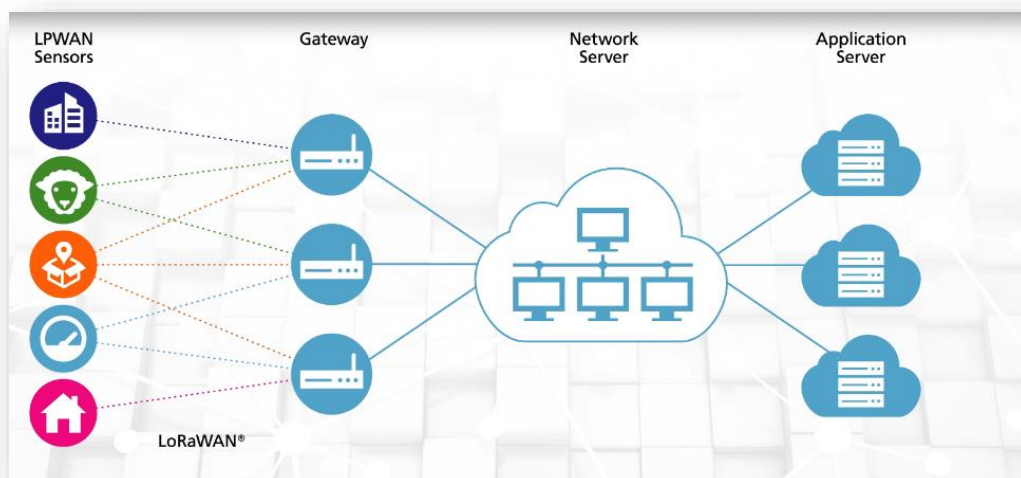


Figura 5 - Arquitetura LoRaWAN

O “Application Server” da TTN é o principal ponto de configuração de toda a rede uma vez que é por lá que os “End Devices” são adicionados á rede através do preenchimento de um formulário com alguns dados do equipamento como indicação do fabricante, modelo, versão e outras características, permitindo assim que comuniquem com o “Application Server”.

Após esta adição é possível fazer a manutenção destes mesmo equipamentos e até alterar as suas configurações assim como visualizar as mensagens enviadas pelos “End Devices” como as mensagens de envidadas para eles. Para além destas funções é também possível a simulação de mensagens para realizar alguns testes com os equipamentos.

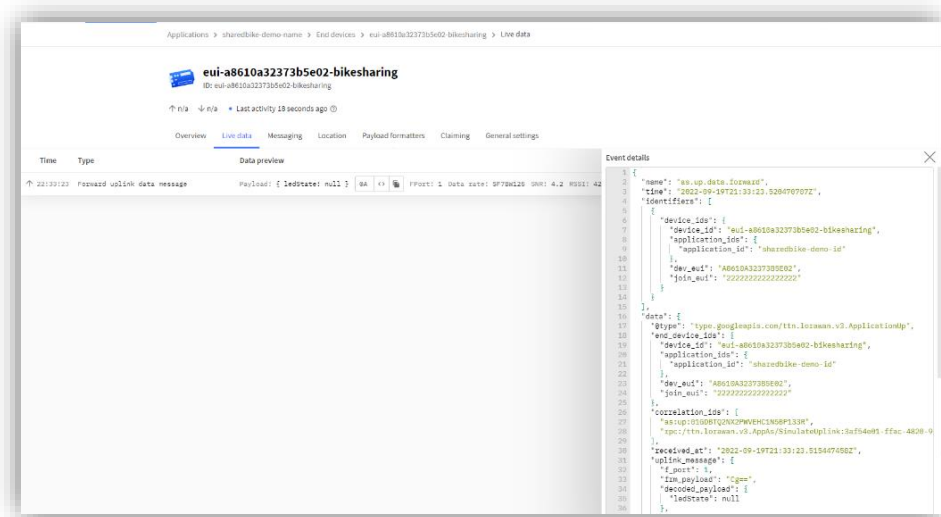


Figura 6 - TTN Website

Tendo em conta todas as informações descritas no parágrafo anterior foi escolhida a rede LoraWAN [Figura 5], para estabelecer comunicação entre as bicicletas e todo o resto do projeto.

3.3.1. Localização e autorização

Estando a questão de conectividade à rede LoraWAN resolvida, passou-se a tratar da maneira de obter as coordenadas geográficas, o que não demorou muito, pois, após alguma pesquisa, foi encontrado o Módulo Ublox Neo-6m GPS [39][40][Figura 7], sendo este módulo compatível com a placa de Arduino, a escolha ficou bastante fácil.



Figura 5 - Módulo Ublox Neo-6m GPS

Na parte do GPS ocorreram alguns problemas no começo, pois o GPS e o respetivo módulo vinham descarregados, isto é, sem energia nenhuma, o que só foi a descoberto mais tarde. Descobrimos o porquê do mesmo não se ligar, e após algumas horas de espera, o led azul, integrado no módulo, começou a piscar. Por volta desta altura o GPS estava carregado, mas ainda não enviava coordenadas nenhuma. Após alguma pesquisa, e de uma reunião com o professor Luís Oliveira, concluiu-se que, cada vez que o GPS é ligado, é necessário esperar que o mesmo encontre alguns satélites para conseguir guardar as suas coordenadas, contudo, sendo a primeira vez, foi necessário aguardar algumas horas para que o mesmo conseguisse detetar o máximo de satélites possíveis, conseguindo assim construir um mapa de satélites, e por fim ser o mais preciso possível.

Passados estes primeiros obstáculos todos, iniciou-se o desenvolvimento do código, tendo em conta que já era possível receber a localização exata do GPS. A partir daí, foi necessário apenas colocar as coordenadas recebidas em 2 variáveis diferentes, a longitude na variável “cur_lng” e a latitude na “cur_lat”. De seguida, os dados recebidos são convertidos em formato String e guardados novamente em

outras 2 variáveis “vlat” e “vlng”, onde por fim colocadas numa outra variável, de nome “msg”, como mostra a [Figura 8], na qual separámos a longitude e a latitude por uma vírgula (‘,’), para tornar a receção da mesma mais simples.

```
//formação da msg que se vai enviar para a ttn  
String msg = vlat + "," + vlng ;
```

Figura 6 - Variável para enviar através da TTN com as coordenadas

As coordenadas mencionadas no parágrafo anterior foram criadas na função “loop ()”, na qual se a localização do GPS for válida e a qualidade de serviço do GPS for menos que 10, as variáveis são criadas, é criado também um “delay ()”, que de 10 em 10 segundos, envia a mensagem, em forma de payload para a TTN. Caso a localização do GPS seja inválida e a qualidade de serviço do GPS seja maior que 10, a mensagem “No GPS data!” é escrita na consola, assim como o número de satélites que o GPS encontrou, e a qualidade de serviço atual do GPS.

Conectividade LoraWAN

Para o Arduíno se ligar à rede LoraWAN, é necessário introduzir os seguintes dados: appEui, e a appKey, [Figura 9]. Ambas as variáveis são geradas no Servidor TTN. A appEui é um identificador usado para identificar o server de conexão durante a ativação da ligação “End device” – “Application Server”.

```
// AppEui  
static String appEui = "1[REDACTED]1";  
// AppKey  
static String appKey = "B01[REDACTED]5F";
```

Figura 7 - Variáveis de ligação

O código de conectividade à rede LoraWAN passa pela função “connectr()”, [Figura 10], na qual o Arduino aguarda até uma conexão ser encontrada. Ao encontrar, faz login com as credenciais, caso as credenciais sejam inválidas aparece a seguinte mensagem na consola: ‘conn: FAIL’, caso contrário aparece a mensagem ‘conn: OK’ e o código prossegue.

```
//conexão com a ttn
void connectr() {
  //mensagem de conexão
  Serial.println("conn: connecting...");
  connected = 0;
  //procura a conexão com a TTN
  if (!modem.begin(EU868)) {
    while (1) {}
  };
  //assim que encontra a conexão faz login com as respetivas credenciais
  connected = modem.joinOTAA(appEui, appKey);
  //se não forem válidas
  if (!connected) {
    //mensagem de erro
    Serial.println("conn: FAIL");
    //se não
  } else
  {
    //mensagem de ligação estabelecida
    Serial.println("conn: OK");
  }
}
```

Figura 8 - Função de conectividade à rede LoraWAN

Após estabelecida a conexão entre ao Arduino e o Servidor TTN, o Arduino manda uma mensagem à TTN para confirmar a comunicação, se não existir conexão volta a tentar conectar-se, caso exista conexão, é criado um pacote de envio e por sua vez enviado uma mensagem, depois a resposta da TTN é guardada na variável “err”. Se for recebida alguma resposta, aparece na consola a seguinte mensagem: “data: OK” e as coordenadas lidas pelo GPS são enviadas (explicado mais à frente), caso contrário aparece a mensagem: “data: ERROR”.

3.3.2. Autorização e estado de bicicleta

Em termos da Placa de Arduíno, falta apenas implementar uma maneira de mostrar ao utilizador que a bicicleta se encontra disponível ou já em uso, tendo em conta a instrução de não ser necessário a preocupação com um sistema literal e físico, foi implementado um LED no esquema de Arduíno, [Figura 7], que cada vez que a bicicleta se encontra disponível o LED liga, caso contrário apaga, para proteger e prevenir que o LED recebesse mais energia do que aguenta, foi necessário implementar também um conjunto de três resistências em série, de 330Ω , o que equivale a aproximadamente $1k\Omega$.

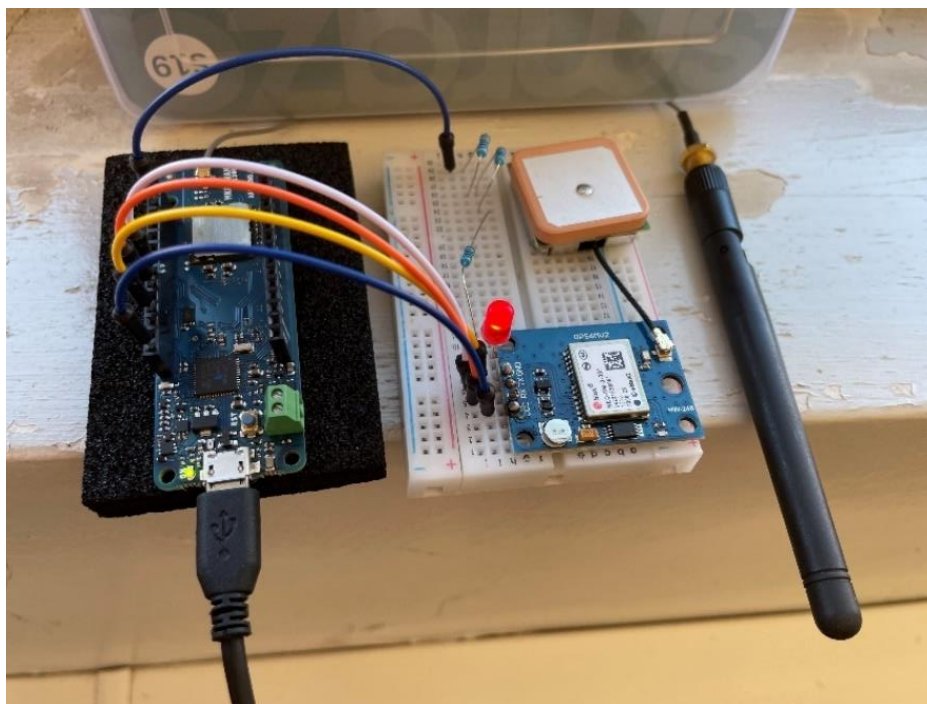


Figura 9 - Esquema geral da placa de arduíno

Por fim, o acionamento do led simula o desbloqueio da bicicleta, quando o Arduíno recebe um downlink (as mensagens de downlink são mensagens enviadas do servidor, para a TTN que por sua vez, encaminha a mensagem para o hardware da

bicicleta) enviado do Servidor, (explicado mais à frente), o led desliga-se, caso o Arduíno não receba downlink nenhum, o led permanece acesso, [Figura 11].

```
if (!modem.available()) {  
    Serial.println("No downlink message received at this time.");  
    digitalWrite(LED, LOW);  
    last_lat = cur_lat;  
    last_lng = cur_lng;  
    return;  
}  
char rcv[64];  
int i = 0;  
while (modem.available()) {  
    rcv[i++] = (char)modem.read();  
}  
digitalWrite(LED, HIGH);  
last_lat = cur_lat;  
last_lng = cur_lng;
```

Figura 10 - Código ativação/desativação LED

O mais difícil na questão do led foi enviar uma mensagem a partir do Servidor, com destino ao hardware instalado na bicicleta, do valor da variável “available” da Bicicleta, isto é, se o valor do atributo “available” da Bicicleta estiver a “true”, não é enviado nenhum downlink, mas assim que o valor do atributo for alterado para “false”, é enviado um downlink, através da TTN, com destino ao Arduíno, para avisar o mesmo que deve ativar o LED (Este processo não é instantâneo devido à velocidade da rede LoraWAN).

3.4. Base de Dados

3.4.1. Estrutura da Base de dados

Em relação à estrutura da base de dados ficou decidido a divisão em duas partes:

- Utilizadores
- Bicicletas

Os Utilizadores estão divididos em três diferentes coleções e as Bicicleta em duas [Figura 12].

Utilizadores:

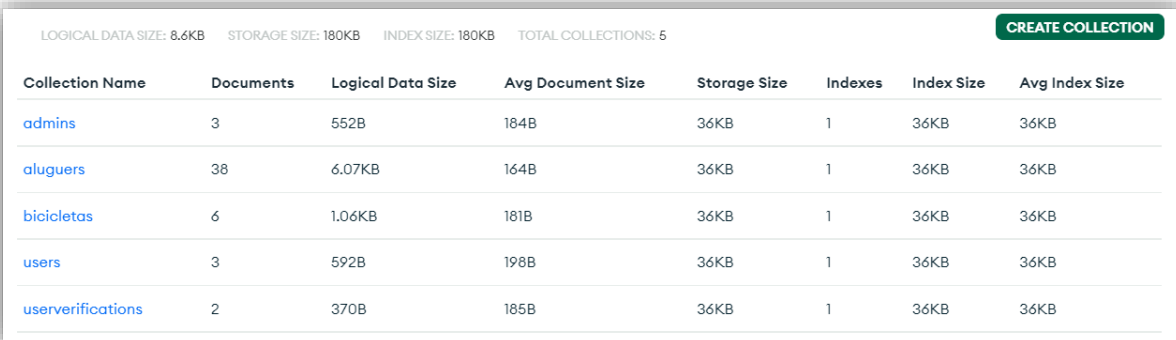
- **Users:** Coleção onde são armazenadas todas as informações (nome, email, data de nascimento, palavra-passe encriptada) dos utilizadores.
- **UserVerifications:** Coleção para onde os dados dos utilizadores vão após a criação de conta na nossa aplicação. Assim que os utilizadores confirmam a criação de conta estes dados são transferidos para a tabela Users.
- **Admins:** Coleção semelhante à tabela de utilizadores, mas que contem os dados dos gestores que tem acesso ao website de gestão da aplicação.

Bicicletas:

- **Bicicletas:** Coleção dos dados das bicicletas e armazenam informações como: nome da bicicleta, preço do aluguer por minuto, latitude, longitude, horas de utilização, se esta

bloqueada ou desbloqueada e ainda uma imagem representativa da bicicleta.

- **Alugueres:** Coleção relativa aos alugues efetuados contendo informações como a bicicleta que foi alugada, o utilizador que a alugou, o preço do aluguer, a hora de começo do aluguer, a hora a que terminou, e o tempo que durou.



Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
admins	3	552B	184B	36KB	1	36KB	36KB
aluguers	38	6.07KB	164B	36KB	1	36KB	36KB
bicicletas	6	1.06KB	181B	36KB	1	36KB	36KB
users	3	592B	198B	36KB	1	36KB	36KB
userverifications	2	370B	185B	36KB	1	36KB	36KB

Figura 11 - Tabelas de Dados mongoDB

Uma ferramenta importante, que MongoDB [38] fornece é a possibilidade de criar gráficos para a análise os dados [Figura 14]. Ajudando assim na componente do website de gestão, permitindo aos gestores perceberem o que se vai passando tanto com as bicicletas tanto com os utilizadores.

Devido ao facto da interface do website de MongoDB para a criação destes gráficos ser simples e intuitiva, conseguindo extrair facilmente alguns dos dados da nossa base de dados para uma componente visual e personalizá-la de modo que sejam perceptíveis os resultados desejados a apresentar.

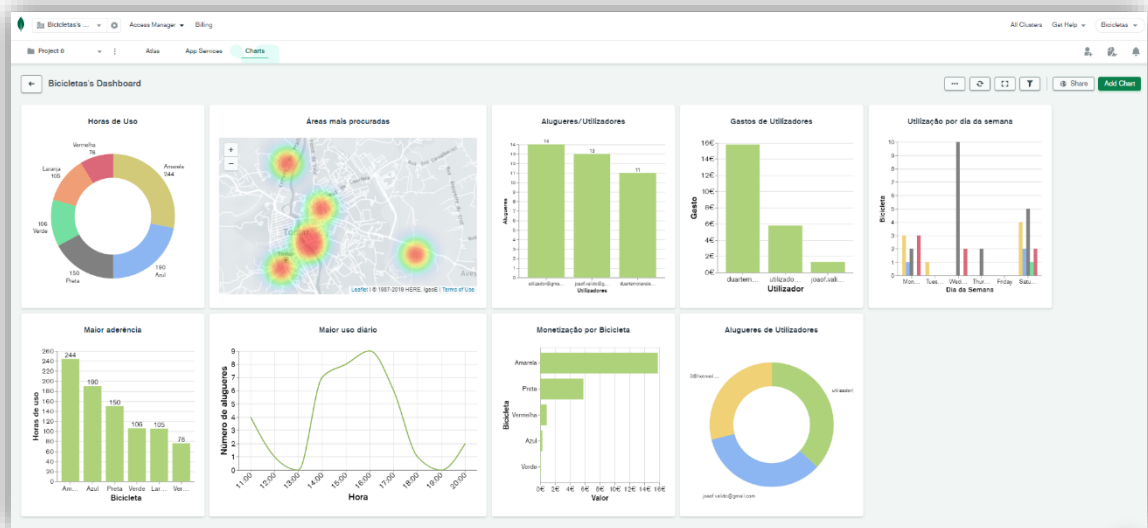


Figura 12 - Gráficos Criados no MongoDB

Concluindo assim, é possível analisar o papel importante que a base de dados desempenha. Passando então a descrever a próxima e mais importante componente do projeto que é o servidor. Este que realiza todas as operações sobre esta base de dados.

3.5. Servidor

Para compreender melhor o funcionamento do servidor, este é dividido nas seguintes partes:

- Conexão com a Base de Dados;
- Conexão com a TTN;
- Interface entre as aplicações e website com a Base de Dados (API);

3.5.1. Conexão com a Base de dados

Para poder disponibilizar os dados às diferentes aplicações o servidor é responsável por estabelecer conexão com a Base de Dados tanto para ler, inserir, editar ou eliminar a informação necessária. Assim, para tal é utilizada a biblioteca ‘mongoose’ [37] que fornece uma solução direta, baseada em esquemas permitindo modelar os dados presentes na Base de Dados. A conexão é feita a partir de uma simples linha de código. [Figura 16], onde apenas é necessário indicar a chave de conexão, [Figura 15], fornecida pelo MongoDB juntamente com a nossa palavra-passe do MongoDB e conseguimos assim estar em sincronia com a Base de Dados.

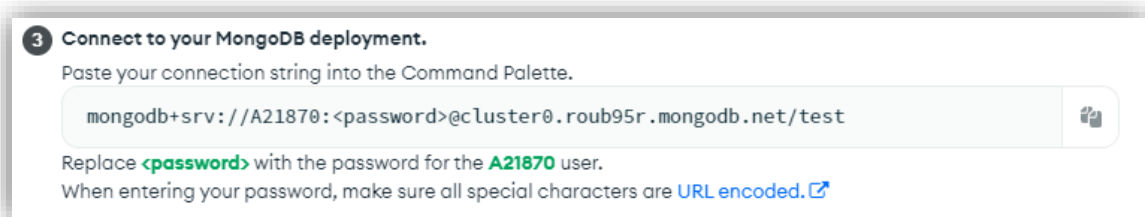


Figura 13 - Key de conexão onde inserimos a nossa palavra-passe fornecida pelo MongoDB

```
const mongoose = require('mongoose');
mongoose.connect('mongodb+srv://A21870:<password>@cluster0.roub95r.mongodb.net/test');
```

Figura 14 - Exemplo de conexão com a Base de Dados no lado do Servidor

Assim que estabelecida a conexão é possível então utilizar os modelos presentes na Base de Dados tanto na interface entre as aplicações e website como na conexão entre o servidor e a TTN pois a partir do modelo ‘mongoose’ a troca de informações com a Base de Dados torna-se fácil, assim, com simples linhas de código é possível aceder às tabelas desejadas.

3.5.2. Conexão com a TTN

Uma função fundamental necessária é estabelecer ligação com as bicicletas logo para tal, uma vez que as bicicletas se encontram ligadas à The Things Network [17], o servidor deve, por sua vez conectar-se à TTN.

Para realizar tal conexão é utilizado MQTT (MQ Telemetry Transport), um protocolo de conexão máquina a máquina (M2M), projetado para realizar transporte de mensagens de publish/subscribe [18]. A Things Network usa MQTT para notificar ativações de dispositivos e mensagens na rede, mas também permite publicar uma mensagem de downlink para um dispositivo específico como resposta a uma mensagem recebida.

Para realizar esta ligação, [Figura 17], no lado do servidor utilizamos a biblioteca indicada no próprio site da TTN, ‘mqtt’ e indicamos o servidor ao qual nos queremos conectar, que no nosso caso é o servidor europeu ‘http://eu1.cloud.thethings.network/’, as respetivas credenciais do End Device e outras opções de configuração da conexão.

```
var client = mqtt.connect('http://eu1.cloud.thethings.network',options);

// MQTT setup
var mqtt = require('mqtt');
var options = {
  port: 1883,
  clientId: 'mqtts_' + Math.random().toString(16).substr(2, 8),
  username: 'ard-lora-test-bici-ipt@ttn',
  password: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  keepalive: 60,
  reconnectPeriod: 1000,
  protocolId: 'MQIsdp',
  protocolVersion: 3,
  clean: true,
  encoding: 'utf8'
};
```

Figura 15 - Conexão do servidor com a TTN

Assim estabelecida a ligação, o servidor já é capaz de receber todas as mensagens enviadas entre o Arduíno e a TTN assim como publicar downlinks para serem recebidos mais tarde pelo Arduino.

Para melhor compreensão do funcionamento do servidor assim que é recebida uma mensagem podemos observar o esquema da Figura 16.

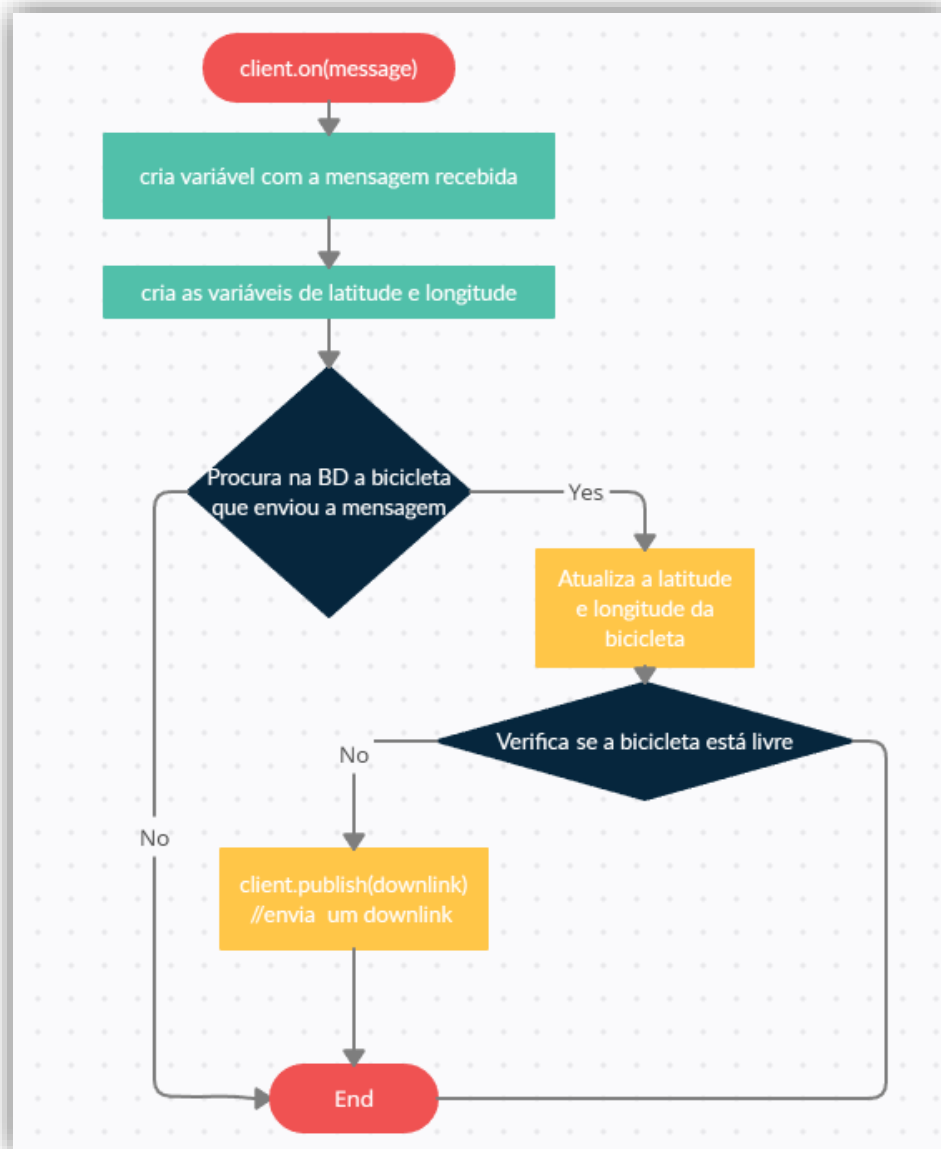


Figura 16 - Servidor assim que recebe uma mensagem da TTN

Assim, torna-se possível verificar que o servidor responde à TTN apenas se encontrar a bicicleta que enviou a mensagem, atualiza as coordenadas dessa mesma bicicleta com os valores recebidos na mensagem que veio da TTN e ainda verifica se a bicicleta se encontra livre e caso não se encontre envia um downlink à TTN para esta por sua vez enviar ao módulo Arduino.

3.5.3. Interface entre as aplicações e website com a Base de Dados

Tanto a aplicação mobile como o website de gestão fornecem e necessitam de todo o tipo de informações presentes na Base de Dados. Logo é importante que a criação de uma interface que consiga fornecer ao front-end do projeto estes dados de forma simples e útil para um funcionamento rápido e otimizado.

Existem quatro “requests” diferentes que esta interface oferece para realizar estas trocas de informação.

- User;
- Admin;
- Bicicleta;
- Aluguer;

Cada uma destes requests tem funções com propósitos diferentes, contando posts (pedidos com resposta em espera) ou com gets (apenas pedidos de informação).

User

Por sua vez esta request divide-se em cinco sub-requests:

1- SignUp:

Responsável por permitir a aplicação criar contas de utilizador ao fornecer à interface o email, o nome, a palavra-passe e a data de nascimento criando assim um request à API para validar estas credenciais e esperar a resposta da API para informar a aplicação do resultado da criação da conta.

2- VerifyEmail:

Serve para enviar um email de confirmação de conta assim que o utilizador cria conta. E notifica a aplicação que o email foi enviado para por sua vez esta apresentar a mensagem ao utilizador.

3- ResendEmail:

Se o utilizador não receber o email pode pedir um reenvio à aplicação que utiliza esta rota para o realizar.

4- SignIn:

Request que verifica os dados do utilizador enviados pela aplicação e retorna uma mensagem consoante o resultado dos dados introduzidos.

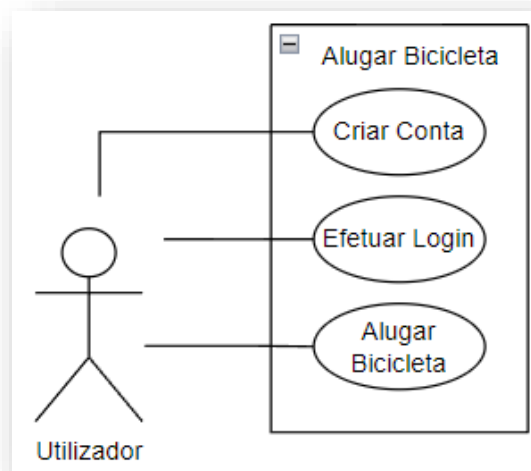


Figura 17 - Use case "Alugar Bicicleta"

Admin

Muito similar à request User, mas esta apenas possui duas sub- requests e destina-se apenas ao website de gestão:

1- SignUp:

Permite criar contas de admin com dados fornecidos no website de gestão e enviados para esta rota que por sua vez faz várias validações e caso passe em todas cria uma conta de admin e responde de volta ao request com a mensagem de resultado da criação de conta. Esta rota apenas é acessível no website de gestão a alguém com permissões suficientes para criar estas contas de admim como vamos explicar mais à frente.

2- SignIn:

Valida e verifica os dados introduzidos no website de gestão e devolve uma mensagem consoante os dados introduzidos.

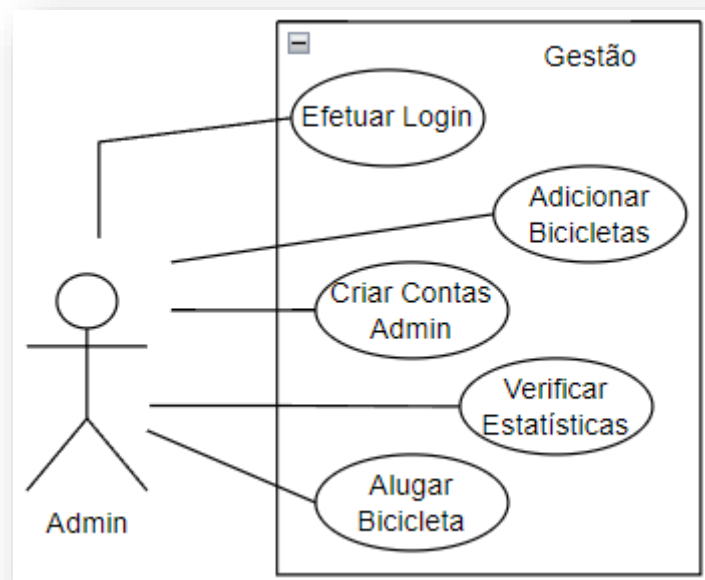


Figura 18 - Use case 'Gestão'

Bicicleta

Rota dividida em duas sub-rotas:

1- AddBike:

Request utilizado no website de gestão e que cria uma bicicleta na Base de Dados ao fornecer o nome, o preço ao minuto e o link da imagem da bicicleta. Este post à interface devolve uma mensagem ao website consoante os dados recebidos e as verificações feitas.

2- AllBikes:

Request utilizado para devolver todas as bicicletas e os seus dados que se encontram na Base de Dados.

Aluguer

Requests relativa aos alugueres das bicicletas e que se divide em quatro sub-requests:

1- AddAluguer:

Utilizada na aplicação, esta rota necessita de uma bicicleta e de um utilizador como parâmetros recebidos. Verifica se a bicicleta escolhida já se encontra a ser utilizada, ou se este aluguer já foi terminado, caso passe nas verificações cria um aluguer e devolve à aplicação a mensagem resultante deste *post*.

2- VerificaAluguer:

Request importante para o bom funcionamento da aplicação uma vez que serve para validar o estado dos alugueres das bicicletas. Assim que os utilizadores seleccionam uma bicicleta na aplicação é feito um *post* à API com a bicicleta que foi seleccionada e o utilizador que a seleccionou. Por sua vez a API devolve uma mensagem à aplicação para esta sabe o que deve apresentar ao utilizador.

3- CloseAluguer:

Possibilita o utilizador de terminar um aluguer e atualiza as informações tanto do aluguer, ao calcular o tempo e preço do aluguer, assim como da bicicleta ao adicionar o número de horas do aluguer ao número de horas da bicicleta e mudar o estado da bicicleta de indisponível para disponível.

4- AllAlugueres:

Request utilizada para devolver todas os alugueres presentes na Base de Dados.

3.6. Aplicação Mobile

3.6.1. Registrar

Estando a escolha realizada, começou o processo de desenvolvimento. Numa primeira fase, foi necessário tratar da criação dos utilizadores, de maneira a que os mesmo conseguissem autenticar-se através do seu e-mail, para evitar a criação de contas a partir de e-mails inexistentes e/ou que utilizadores registassem e-mails de outras pessoas sem o seu consentimento. Após o utilizador registar-se, [Figura 21], isto é, assim que o utilizador carregar no botão “Registrar”, a aplicação envia os dados da conta para a BD, mas com o atributo “verified” a “false”, pois o utilizador ainda não se verificou, é redirecionado para o ecrã de “Verificação de E-mail”, [Figura 22], no qual é enviado ao utilizador, para o e-mail introduzido no registo, [Figura 24], no qual o utilizador tem de aceder e carregar no link disponibilizado para conseguir autenticar a sua conta. No caso de o utilizador não ter recebido e-mail nenhum, no ecrã de “Verificação de E-mail”, tem a hipótese de reenviar o e-mail de verificação, pressionando a palavra/botão “Resend”. O e-mail é enviado através do método nodemailer (), tendo como remetente o e-mail criado para este projeto “bicicletascomunitarias@hotmail.com”. O link disponibilizado no e-mail de confirmação, tem uma duração máxima de 6 horas, isto para prevenir que sejam criados utilizadores com e-mail aos quais os mesmos não têm acesso. Ao carregar no link/palavra “here” do e-mail, o utilizador é enviado para uma das 2 páginas de html. Caso o utilizador tenha se registado à menos de 6 horas, será encaminhado para a página HTML, [Figura 23], e a sua conta será verificada, logo o atributo “verified” da tabela de utilizador, também passa a “true”, o que por sua vez permite ao utilizador efetuar o Login com os seus dados. No caso de o utilizador já ter a sua conta verificada ou o período de 6 horas já ter passado, ser-lhe-á apresentada a página HTML, [Figura 25].

Nome

E-mail

Data de Nascimento

Palavra-Passe

Confirmar Palavra-Passe

Registrar

Already have an account? [Login](#)

Figura 19 - Ecrã "Registrar"

Bem-Vindo!

Por favor verifique o seu email através do link enviado parautilizador123@hotmail.com

Continuar →

Não recebeu o email? [Resend](#) em 27 segundo(s)

Figura 20 - Ecrã "Verificação de E-mail"

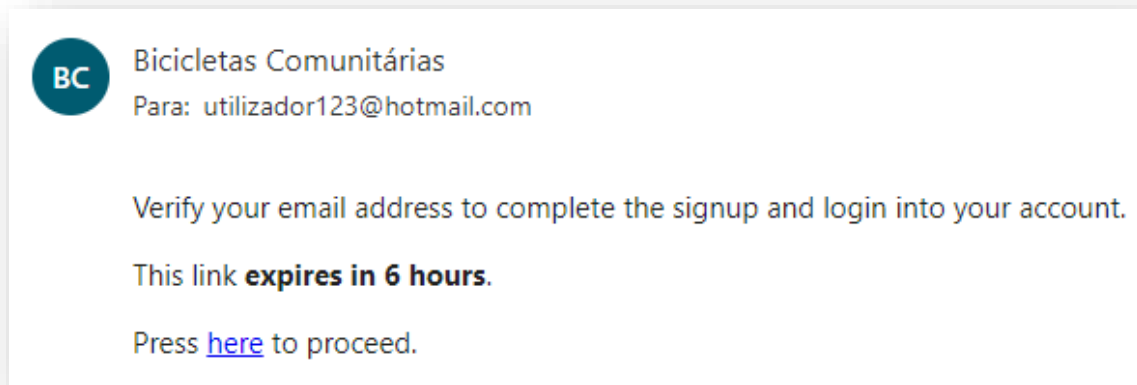


Figura 22 - E-mail de Verificação

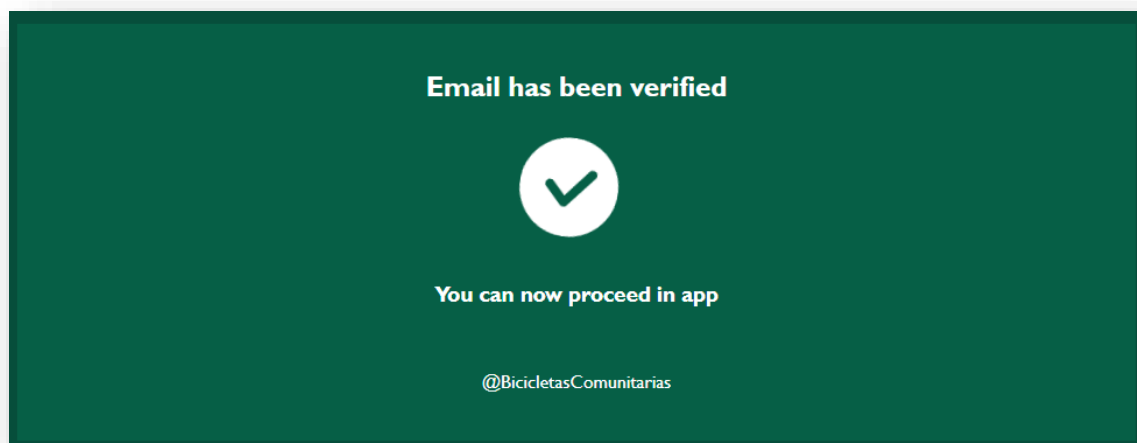


Figura 21 - E-mail Verificado

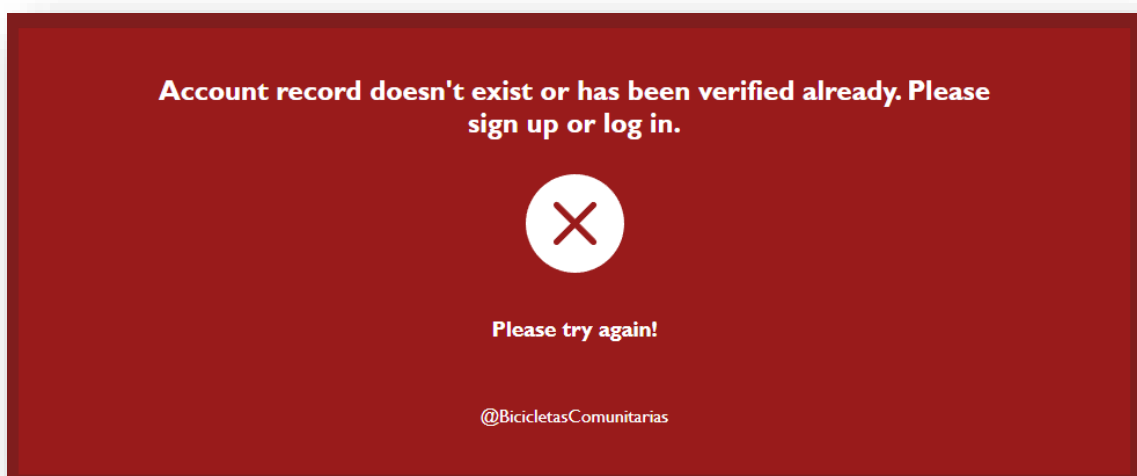


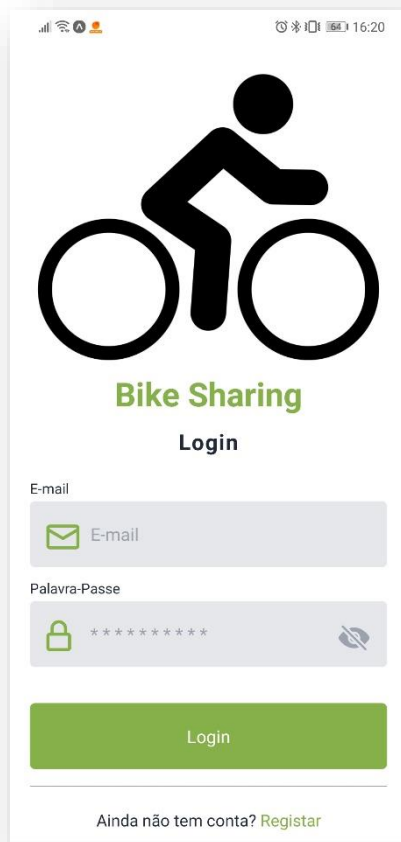
Figura 23 - E-mail não verificado

3.6.2. Login

Após a verificação de e-mail, o utilizador passa agora a conseguir efetuar login na aplicação através da página ‘Login’, [Figura 27], a partir do qual, após colocar a password e carregar no botão “Login”, a aplicação compara os dados inseridos nos campos com os respetivos dados do utilizador guardados na BD, no caso da password, a mesma é guardada em formato hash, para que ninguém senão o próprio utilizador saiba a sua password. Caso os dados não correspondam por algum motivo, aparecerá esse mesmo motivo escrito no ecrã para que o utilizador perceba o que o está a impedir de efetuar o Login.

3.6.3. Welcome

Caso o utilizador introduza os dados corretamente, ao carregar no botão “Login”, será encaminhado para a página “Welcome”, [Figura 26], na qual terá a hipótese de dar Logout e retornar à página de Login, ou então, ao carregar no botão “Procurar Bicicletas”, será encaminhado para a página onde é possível visualizar o mapa com as bicicletas disponíveis.



The login screen features a large black silhouette of a person riding a bicycle at the top. Below it, the text "Bike Sharing" is displayed in green, followed by "Login" in black. There are two input fields: "E-mail" with an envelope icon and "Palavra-Passe" (Password) with a lock icon and a toggle for visibility. A green "Login" button is positioned below the fields. At the bottom, a link "Ainda não tem conta? Registrar" is shown.

16:20

Bike Sharing

Login

E-mail

E-mail

Palavra-Passe

Login

Ainda não tem conta? [Registrar](#)

Figura 24 - Ecrã Login



The welcome screen has a light gray background. At the top is an icon showing a bicycle above three stylized people. Below this, the text "Bem-Vindo!" is in green, followed by "Utilizador" and the email "utilizador@gmail.com". A circular profile picture placeholder is shown. At the bottom are two green buttons: "Logout" and "Procurar Bicicletas".

16:23

Bem-Vindo!

Utilizador
utilizador@gmail.com

Logout

Procurar Bicicletas

Figura 25 - Ecrã Welcome

3.6.4. Mapa

Ao carregar no botão “Procurar Bicicletas”, o utilizador é direcionado para a página do Mapa, [Figura 28], na qual o mesmo tem a possibilidade de visualizar onde todas as bicicletas estão situadas. O mapa utilizado é o mapa disponibilizado pelo próprio React-Native [52], que no caso do iOS, para correr na aplicação, vai buscar o mapa integrado no telemóvel, o mapa da própria Apple. No caso do Android, o telemóvel vai buscar o mapa da Google, o que da primeira vez que submetemos a aplicação na Play Store, ao carregar no botão “Procurar Bicicletas”, a aplicação crashava, isto é, fechava automaticamente. Conseguimos aceder aos logs da aplicação, o que nos permitiu perceber que a aplicação não tinha permissão para utilizar os mapas do Google, por isso tivemos de associar ao código uma API do Google Maps [61], de maneira a ganharmos a tal permissão para utilizar os mapas na nossa aplicação.



Figura 26 - Mapa

3.6.5. Aluguer

Ao carregar numa bicicleta, o utilizador pode ser enviado para dois sítios diferentes. Se a bicicleta se encontrar disponível, o mesmo é enviado para o ecrã da bicicleta, [Figura 29], no qual, os dados, tanto da bicicleta, como os do utilizador, são inseridos automaticamente, as bicicletas são distinguidas através de um código de cores, é também indicado o custo do aluguer por minuto. Os dados de cada bicicleta são adicionados pelo administrador (explicado mais à frente).

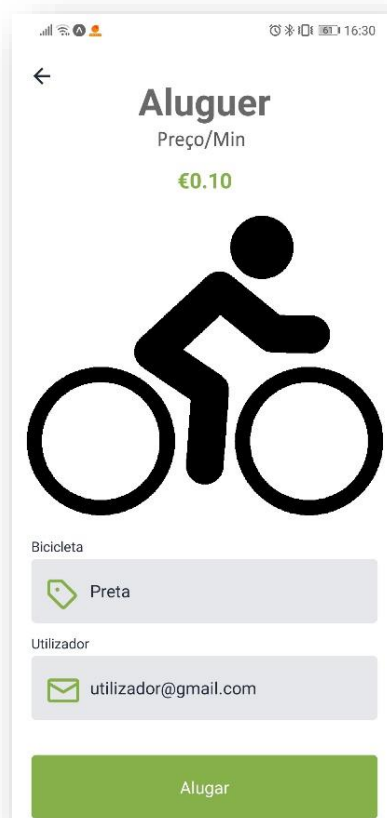


Figura 27 - Ecrã de Aluguer

Caso a bicicleta se encontre já em uso (a aplicação vai à BD verificar se a bicicleta em questão tem o atributo “available” igual a false, caso tenha, a bicicleta não está disponível, mas sim em uso), o utilizador é enviado para o ecrã da “Bicicleta em uso”, [Figura 31], no qual, tem a hipótese de voltar atrás através do botão “Procurar outra bicicleta”, ou carregando no “mini-mapa” disponível.

O utilizador ao pressionar o botão “Alugar” do ecrã de aluguer [Figura 29], é registado na BD a hora de início para mais tarde, ao terminar aluguer, fazer o cálculo do custo de aluguer, é também direcionado para o ecrã de Terminar o aluguer, [Figura 31], onde tem o preço por minuto da bicicleta que alugou, a bicicleta representada pela sua cor identificadora, e um botão a vermelho para finalizar o seu aluguer.

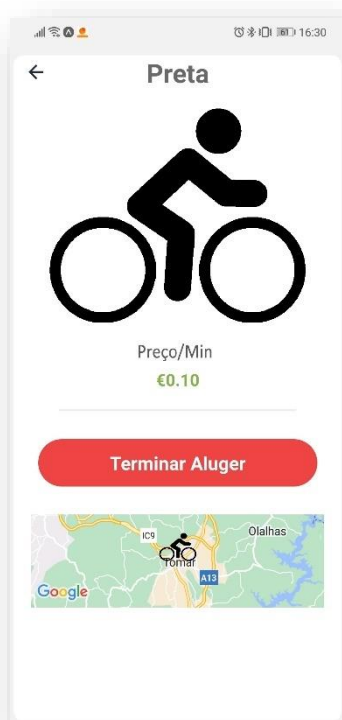


Figura 28 - Ecrã Terminar aluguer



Figura 29 - Ecrã de Bicicleta em uso

Ao carregar no botão “Terminar Aluguer”, é guardado a hora exata do momento em que o utilizador pressiona o botão, o valor do atributo “available” passa para true, os minutos de uso também são adicionados e atualizados através do cálculo, [Figura 32]. O utilizador é enviado para a página de Resumo, que tem como propósito mostrar ao utilizador o custo do seu aluguer, assim como o tempo que o mesmo esteve a aluguer a bicicleta. O custo do aluguer é calculado através da expressão demonstrada na [Figura 33].



Figura 30 - Ecrã de Resumo

```
const startHour = data[0].startHour;  
const hoursUsed = Math.round(Math.abs(Date.now() - startHour) / 60000)  
const precoTotal = (priceHour * hoursUsed)  
Aluguer.updateOne(  
  { bike, available: "false" },  
  { "available": true, endHour: Date.now(),  
    hoursUsed: hoursUsed,  
    price: precoTotal })
```

Figura 31 - Cálculo do preço

3.6.6. Publicação nas Stores

Em termos de publicação nas loja, já foi possível publicar a aplicação na Play Store (Android) [22], na primeira tentativa a aplicação crashava na altura de aceder ao mapa com as bicicletas, pelo facto de não ter integrado a API do Google Maps no nosso código, assim que corrigido esse pormenor, foi possível submeter a aplicação novamente e desta vez ficou tudo operacional, embora ainda só esteja em testes abertos, e disponível apenas para as pessoas com autorização, a aplicação já funciona na sua totalidade.

No caso da App Store (iOS) [24], já foi submetida a aplicação e a mesma esteve em estado de revisão, passado uns dias recebemos uma resposta a informar que deveríamos implementar a funcionalidade de eliminar as contas de utilizadores, para que os utilizadores consigam apagar as contas assim que o desejarem. Futuramente, iremos implementar essa funcionalidade na nossa aplicação e aí tentaremos submeter novamente a aplicação na App Store.

Queremos deixar aqui um agradecimento especial ao professor José Casimiro Pereira, pois sem a sua ajuda não teria sido possível submeter a nossa aplicação em ambas as stores.

3.7. Website de Gestão

Para garantir o bom funcionamento da aplicação mobile é essencial uma plataforma de gestão onde os responsáveis pela aplicação possam ter acesso a todo o tipo de dados e informações presentes na Base de Dados. Assim, como nas outras componentes, foi necessária a escolha de uma framework para o seu desenvolvimento. Esta uma escolha fácil, pois tendo em conta as escolhas anteriores explicadas, o ambiente de programação que se adequava mais era o React.

React utiliza maioritariamente JavaScript como linguagem de programação e foi desenvolvido pelo Facebook, e permite criar rapidamente e facilmente interfaces para websites e aplicações.

Para aceder aos conteúdos deste website é necessário ter conta de gestor, contas essas que apenas podem ser criadas por outro gestor já registado e com permissões para tal.

Para alojar o servidor utilizamos novamente o Heroku pois como é o mesmo utilizado para alojar o nosso servidor e a experiência com a plataforma tinha sido bastante positiva ficou decidido que esta seria também a melhor opção para o website de gestão.

3.7.1. Página Inicial

Ao entrar no website é nos apresentada a página boas-vindas [Figura 34], na qual o utilizador pode escolher entrar na sua conta.



Figura 32 - Página Inicial

3.7.2. Início de sessão

Página onde o gestor introduz as suas credenciais [Figura 35], que posteriormente são enviadas ao servidor e devolvida uma resposta que por sua vez é verificada pelo website para permitir ou não o início de sessão do gestor.

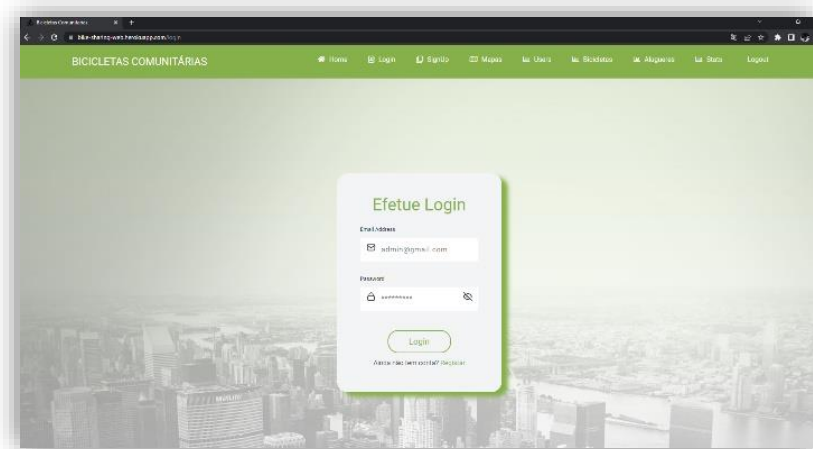


Figura 33 - Página de Início de Sessão

3.7.3. Página de Entrada

Assim que o gestor consegue entrar no website é lhe mostrada uma mensagem de boas-vindas [Figura 36], assim como a opção de sair da sua conta.

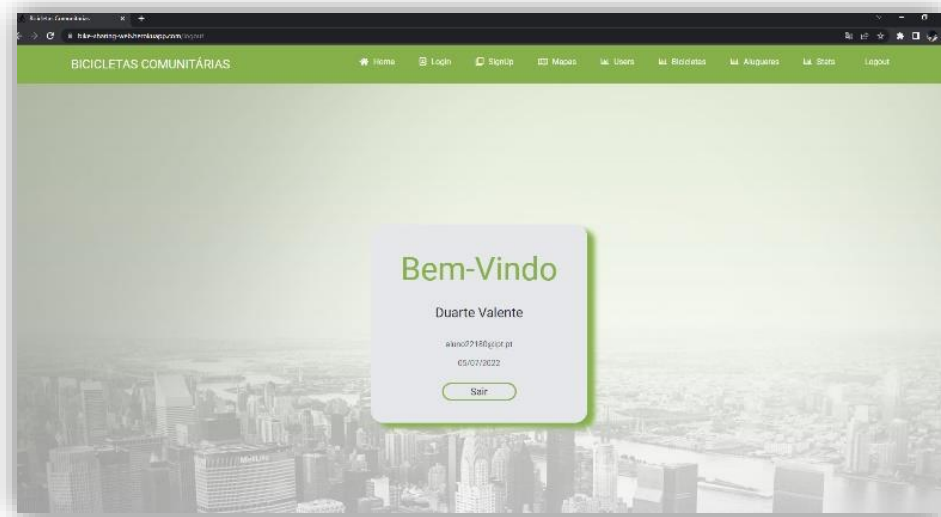


Figura 34 - Página de Entrada

3.7.4. Adicionar Gestor

Como já referido anteriormente, apenas um gestor autorizado consegue criar uma nova conta de gestor o que permite o controlo de quem tem acesso ao website de gestão. Para tal o gestor deve preencher um formulário [Figura 37], com informações como: nome, email, data de nascimento e palavra-passe e submeter estes dados para aprovação por parte do servidor.

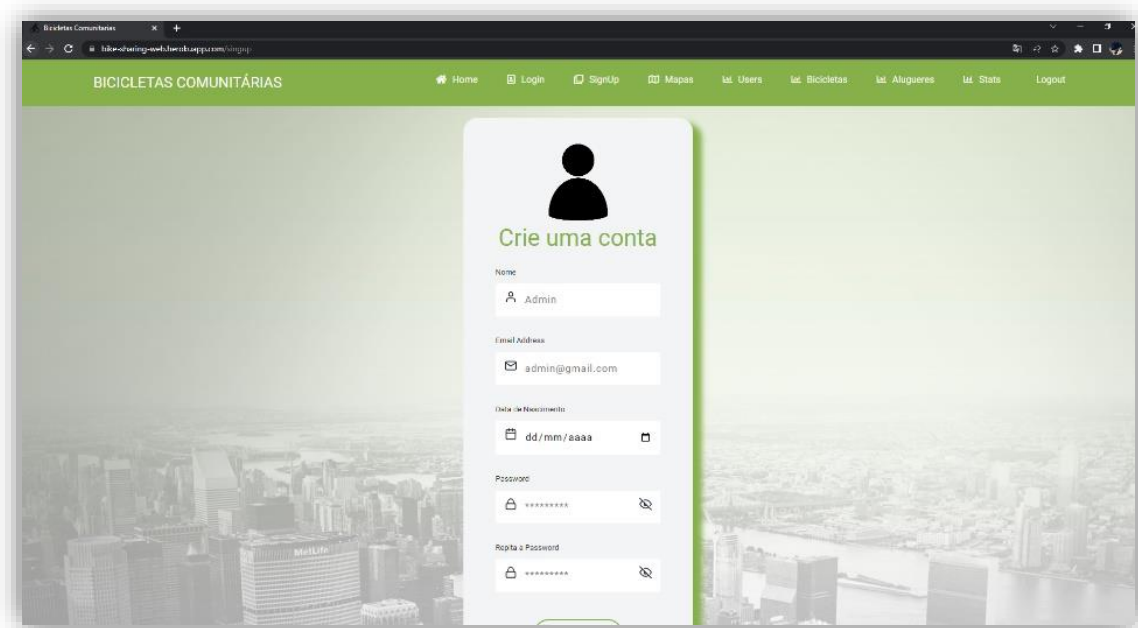


Figura 35 - Página de Criação de conta

3.7.5. Lista de Utilizadores

Vista que apresenta aos gestores a lista de utilizadores que estão registados na aplicação mobile [Figura 38].



Figura 36 - Lista de Utilizadores

3.7.6. Lista de Bicicletas

Esta página é responsável por apresentar a lista de todas as bicicletas [Figura 39], assim como algumas informações sobre as mesmas como fotografia, minutos totais de uso e disponibilidade.

Para além destas informações permite o gestor adicionar uma nova bicicleta. Para a sua adição o gestor deve preencher um formulário com nome, preço por minuto e imagem da nova bicicleta.

Bicicleta	Preço/Minuto(€)	Latitude	Longitude	Uso Total(min)	Disponibilidade (T/F)	Link de Imagem
Verde	0.10	39.998	-8.414	196	true	
Laranja	0.10	39.601	-8.409	105	true	
Vermelha	0.10	39.612	-8.413	76	false	
Preta	0.10	39.606	-8.407	190	true	
Amarela	0.10	39.600	-8.391	244	true	
Azul	0.15	39.629	-8.400	190	false	

Voltar Adicionar Bicicleta

Figura 37 - Lista de bicicletas

3.7.7. Lista de Alugueres

Permite analisar todos os alugueres já realizados pelos utilizadores [Figura 40]. É possível ver informações como a data e hora do aluguer, a bicicleta que foi alugada, o utilizador que a alugou, se já terminou, a data e hora a que foi terminado, o tempo de aluguer e o preço do aluguer.

Bicicleta	Utilizador	Preço (€)	Hora Inicio	Hora fim	Minutos de Uso	Finalizado (T/F)
Preta	utilizador@gmail.com	0.20	03/09/2022, 16:27:28	03/09/2022, 16:29:46	2	true
Preta	utilizador@gmail.com	1.90	03/09/2022, 16:30:31	05/09/2022, 17:09:37	2919	true
Amarela	joao.valido@gmail.com	0.40	05/09/2022, 12:08:58	05/09/2022, 12:12:40	4	true
Preta	joao.valido@gmail.com	0.00	05/09/2022, 17:29:32	05/09/2022, 17:29:59	0	true
Vermelha	joao.valido@gmail.com	0.20	05/09/2022, 17:51:47	05/09/2022, 17:54:16	2	true
Vermelha	joao.valido@gmail.com	0.00	05/09/2022, 17:56:17	05/09/2022, 17:56:29	0	true
Vermelha	joao.valido@gmail.com	0.20	05/09/2022, 17:56:38	05/09/2022, 17:58:36	2	true
Azul	joao.valido@gmail.com	0.30	05/09/2022, 17:56:44	05/09/2022, 17:58:43	2	true
Preta	utilizador@gmail.com	0.00	05/09/2022, 17:57:51	05/09/2022, 17:58:06	0	true
Preta	utilizador@gmail.com	0.40	07/09/2022, 12:48:25	07/09/2022, 12:52:27	4	true
Vermelha	joao.valido@gmail.com	0.00	07/09/2022, 12:51:02	07/09/2022, 12:51:05	0	true
Preta	utilizador@gmail.com	0.70	07/09/2022, 15:48:57	07/09/2022, 15:56:19	7	true
Preta	utilizador@gmail.com	1.80	07/09/2022, 15:58:41	07/09/2022, 16:16:57	18	true
Vermelha	joao.valido@gmail.com	0.00	07/09/2022, 16:10:48	07/09/2022, 16:10:53	0	true
Preta	utilizador@gmail.com	0.10	07/09/2022, 16:25:23	07/09/2022, 16:26:46	1	true
Preta	utilizador@gmail.com	0.10	07/09/2022, 16:44:35	07/09/2022, 16:46:00	1	true
Preta	utilizador@gmail.com	0.00	07/09/2022, 16:48:32	07/09/2022, 16:48:51	0	true

Figura 38 - Lista de Alugueres

3.7.8. Mapa

Vista que apresenta um mapa [48] [Figura 41], com todas as bicicletas da aplicação e que permite o gestor seleccionar uma bicicleta para ver o seu nome, a sua disponibilidade e os minutos de uso da mesma.

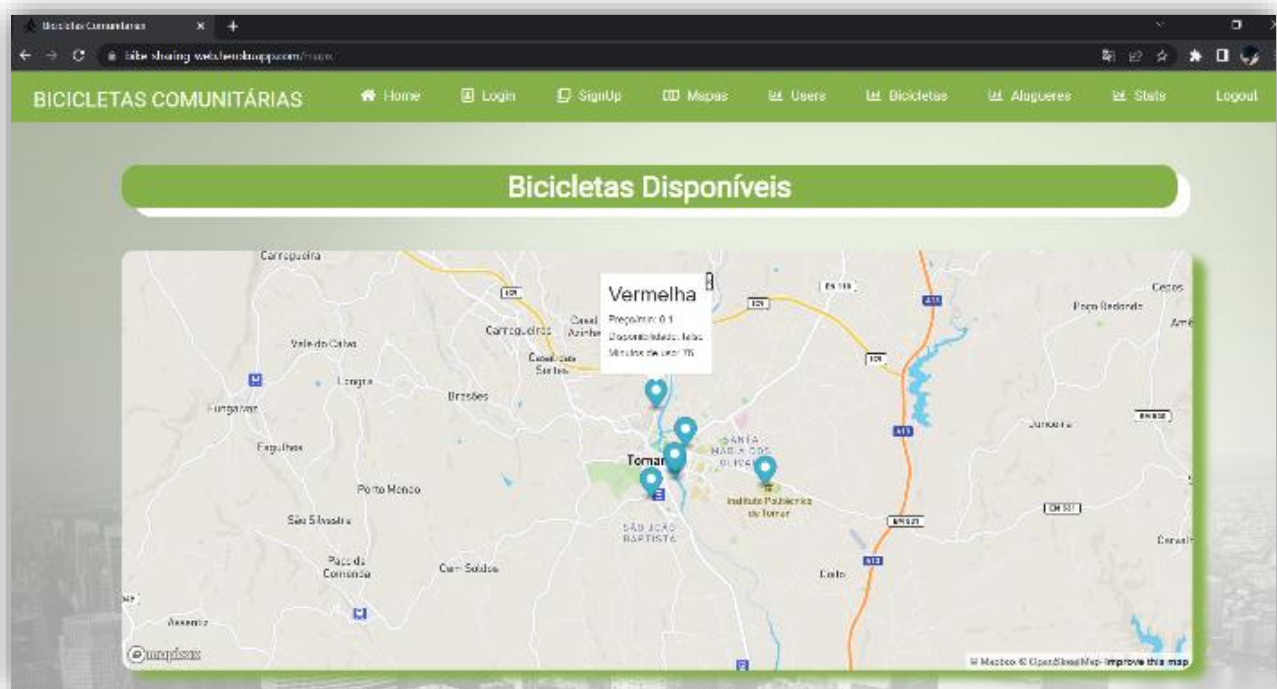


Figura 39 - Mapa

3.7.9. Estatísticas

Por fim, os gestores têm acesso á página de estatísticas das bicicletas, dos utilizadores e dos alugueres que permitem tirar algumas conclusões sobre o funcionamento geral da aplicação de forma que consigam otimizar a mesma.

Esta página apresenta estatísticas como:

- Bicicletas:
 - Horas de Uso;
 - Bicicleta com maior aderência;
 - Dias da semana com maior aderência;
- Alugueres:
 - Hora do dia de maior uso;
 - Bicicleta que fatura mais;
 - Área de maior aderência;
- Utilizadores:
 - Utilizador com mais alugueres;
 - Utilizador com mais gastos;

4. Conclusão e Trabalho Futuro

Neste projeto foram desenvolvidas duas aplicações em React, com o propósito de fazer uma aplicação mobile, na qual os utilizadores conseguem criar conta e por sua vez alugar bicicletas. Um website no qual o gestor consegue controlar o estado e localização das suas bicicletas e analisar algumas estatísticas relacionadas com os utilizadores, as bicicletas e os respetivos alugueres das mesmas. Foi criado ainda um Servidor, também em Node.js, para servir de elo de ligação entre as aplicações e o resto do esquema do projeto (BD, TTN, Arduino).

O produto final Bicicletas Comunitárias, pode ser utilizado pelos utilizadores que desejarem assim que a mesma estiver visível para o mundo inteiro, e não só para os desenvolvedores.

Para trabalhos futuros propõem-se:

- Aplicação:
 - Adição do método de pagamento na hora de terminar o aluguer, de maneira que possa ser cobrado o tempo de aluguer aos utilizadores;
 - Implementar a funcionalidade de eliminar utilizadores na aplicação de maneira que cumpra os requisitos para ser submetida na App Store;
- Website:
 - Adicionar a opção de editar as tabelas de dados das bicicletas por parte do gestor, no website;
 - Alojamento do site, a base de dados, e o nosso servidor, num serviço de maior qualidade, pois estão alojados num servidor de cloud, o que por vezes faz com que as conexões aos mesmos sejam um pouco demoradas;

- Bicicletas:
 - Alterar a maneira de identificar as bicicletas, pois, o método usado é limitado.
 - Evoluir sistema na lógica de multi-tender (i.e. o projeto, devido à rede escolhida (LoraWAN) torna-se bastante versátil, pois a mesma tem cobertura espalhada pelo mundo);

Queremos também deixar aqui um obrigado de ambas as partes ao orientador, professor Luís Oliveira, por todo o auxílio e orientação.

Bibliografia

- [1] 11 setembro 2022. [Online]. Available: <https://www.eseye.com/resources/iot-explained/lora-vs-cellular-for-iot/>.
- [2] 4 agosto 2022. [Online]. Available: <https://www.infoguardsecurity.com/why-you-need-both-authorization-and-authentication/>.
- [3] 11 setembro 2022. [Online]. Available: <https://www.bluetooth.com/blog/bluetooth-low-energy-a-key-partner-for-bike-sharing/>.
- [4] 10 setembro 2022. [Online]. Available: <https://www.linkedin.com/pulse/why-does-make-sense-combine-lorawan-cellular-iot-rohit-gupta>.
- [5] 5 setembro 2022. [Online]. Available: [https://en.wikipedia.org/wiki/Ofo_\(company\)](https://en.wikipedia.org/wiki/Ofo_(company)).
- [6] 16 junho 2022. [Online]. Available: <https://cyclinghikes.com/are-bird-scooters-gps/>.
- [7] 26 junho 2022. [Online]. Available: <https://cyclinghikes.com/are-bird-scooters-gps/>.
- [8] 15 julho 2022. [Online]. Available: <https://www.biometricupdate.com/202101/bird-introduces-id-verification-tool-with-biometrics-support-powered-by-au10tix>.
- [9] 20 junho 2022. [Online]. Available: <https://www.soracom.io/blog/how-hard-is-it-to-build-an-electric-scooter-fleet-like-bird-or-lime/>.
- [10] 5 julho 2022. [Online]. Available: <https://felgo.com/doc/felgo-appdemos-maps-example/>.
- [11] 2 junho 2022. [Online]. Available: <https://www.mobindustry.net/blog/how-to-successfully-integrate-apple-pay-into-your-ios-app/>.
- [12] 25 julho 2022. [Online]. Available: <https://expo.dev/>.
- [13] 5 junho 2022. [Online]. Available: <https://forum.arduino.cc/t/ublox-neo-6m-gps-module-not-working-not-powering-on/879760>.
- [14] 5 julho 2022. [Online]. Available: <https://docs.expo.dev/build-reference/apk/>.
- [15] 3 julho 2022. [Online]. Available: <https://reactnative.dev/>.
- [16] 15 julho 2022. [Online]. Available: <https://reactnative.dev/docs/0.63/geolocation>.
- [17] 1 julho 2022. [Online]. Available: <https://www.thethingsnetwork.org/>.
- [18] 12 julho 2022. [Online]. Available: <https://www.thethingsindustries.com/docs/integrations/pubsub/>.
- [19] 20 julho 2022. [Online]. Available: <https://docs.arduino.cc/hardware/mkr-wan-1300>.

- [20] 13 julho 2022. [Online]. Available: <https://github.com/react-native-maps/react-native-maps>.
- [21] 22 julho 2022. [Online]. Available: <https://github.com/crazycodeboy/react-native-splash-screen>.
- [22] 9 setembro 2022. [Online]. Available: <https://play.google.com/console/about/>.
- [23] 25 agosto 2022. [Online]. Available: <https://cloud.google.com/cloud-console>.
- [24] 6 setembro 2022. [Online]. Available: <https://appstoreconnect.apple.com/login>.
- [25] 15 junho 2022. [Online]. Available: <https://nodejs.org/en/>.
- [26] 17 junho 2022. [Online]. Available: <https://www.w3schools.com/nodejs/>.
- [27] 15 junho 2022. [Online]. Available: <https://www.w3schools.com/js/>.
- [28] 25 agosto 2022. [Online]. Available: <https://reactnavigation.org/>.
- [29] 25 junho 2022. [Online]. Available: <https://heroku.com>.
- [30] 24 junho 2022. [Online]. Available: <https://www.mongodb.com/>.
- [31] 6 setembro 2022. [Online]. Available: <https://grafana.com/docs/grafana/latest/setup-grafana/configure-grafana/>.
- [32] 23 agosto 2022. [Online]. Available: <https://developers.google.com/maps>.
- [33] 3 setembro 2022. [Online]. Available: <https://chafikgharbi.com/expo-android-manifest/>.
- [34] 1 setembro 2022. [Online]. Available: <https://www.botnroll.com/pt/arduino-controladores/4032-arduino-mkr-wan-1300-sem-antena.html>.
- [35] 4 setembro 2022. [Online]. Available: <https://www.thethingsindustries.com/docs/devices/downlink-queue-ops/>.
- [36] 6 setembro 2022. [Online]. Available: https://pipedream.com/@avbentem/ttn-forum-schedule-downlink-after-uplink-p_6lCYmd/readme.
- [37] 20 agosto 2022. [Online]. Available: <https://www.mongodb.com/docs/drivers/node/current/>.
- [38] 26 julho 2022. [Online]. Available: <https://mongoosejs.com/>.
- [39] 25 julho 2022. [Online]. Available: <https://forum.arduino.cc/t/ublox-neo-6m-gps-module-not-working-not-powering-on/879760/20>.
- [40] 15 julho 2022. [Online]. Available: <https://arduino.stackexchange.com/questions/8273/ublox-neo-6m-gps-module-not-responding>.

- [41] 4 setembro 2022. [Online]. Available: <https://portal.ublox.com/s/question/0D52p00008HKCsJCAX/ublox-neo6m-not-getting-fix-on-arduino-power-works-otherwise>.
- [42] 5 julho 2022. [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/message-types/>.
- [43] 15 junho 2022. [Online]. Available: <https://reactnative.dev/docs/environment-setup>.
- [44] 11 junho 2022. [Online]. Available: <https://www.asapdevelopers.com/build-a-react-native-login-app-with-node-js-backend/>.
- [45] 23 agosto 2022. [Online]. Available: <https://v5.reactrouter.com/native/guides/quick-start>.
- [46] 22 agosto 2022. [Online]. Available: <https://redux.js.org/introduction/getting-started>.
- [47] 22 agosto 2022. [Online]. Available: <https://blog.bitsrc.io/top-5-map-libraries-for-react-in-2021-20a37ff5234>.
- [48] 2 setembro 2022. [Online]. Available: <https://www.mapbox.com/>.
- [49] 11 setembro 2022. [Online]. Available: <https://docs.microsoft.com/en-us/javascript/api/overview/powerbi/powerbi-client-react>.
- [50] 22 agosto 2022. [Online]. Available: <https://www.digitalocean.com/community/tutorials/how-to-add-login-authentication-to-react-applications>.
- [51] 7 setembro 2022. [Online]. Available: <https://www.mongodb.com/docs/charts/>.
- [52] 26 agosto 2022. [Online]. Available: <https://www.npmjs.com/package/react-native-navigation>.
- [53] 22 julho 2022. [Online]. Available: <https://stackoverflow.com/questions/47558468/how-to-get-current-location-using-react-native-maps>.
- [54] 18 julho 2022. [Online]. Available: <https://blog.logrocket.com/react-native-maps-introduction/>.
- [55] 12 agosto 2022. [Online]. Available: <https://www.youtube.com/watch?v=1fAvRKCjdc0>.
- [56] 13 agosto 2022. [Online]. Available: <https://www.quora.com/What-backend-language-should-I-use-with-React-Native>.
- [57] 13 junho 2022. [Online]. Available: <https://www.qulix.com/about/blog/the-best-server-side-language/>.
- [58] 2 agosto 2022. [Online]. Available: <https://www.youtube.com/watch?v=LE4Mgkrf7Sk>.

- [59] 17 agosto 2022. [Online]. Available: <https://github.com/react-native-maps/react-native-maps/issues/140>.
- [60] 2 setembro 2022. [Online]. Available: <https://stackoverflow.com/questions/57181839/error-while-updating-property-region-of-a-view-managed-by-airmap-null-longit>.
- [61] 9 setembro 2022. [Online]. Available: <https://cloud.google.com/>.
- [62] 21 agosto 2022. [Online]. Available: <https://docs.expo.dev/versions/latest/sdk/map-view/>.
- [63] 24 agosto 2022. [Online]. Available: <https://velog.io/@youjinee98/error-Error-while-updating-property-coordinate-of-a-view-managed-by-AIRMapMarker>.
- [64] 11 setembro 2022. [Online]. Available: <https://enterpriseiotinsights.com/20170612/internet-of-things/what-lowrawan-main-benefits-technology-tag23>.

Referências

Figura 1 – Rastreador GPS

https://www.alibaba.com/product-detail/365GPS-ZX302-mini-GPS-tracker-made_62357561521.html

Figura 2 – Exemplo cadeado mecânico

https://www.alibaba.com/product-detail/Free-parking-system-Smart-Sharing-Bicycle_60830340535.html

Figura 5 – Arquitetura LoraWAN

<https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/lora/arquitetura.html>

Figura 6 – Placa de Arduino MKR Wan 1300

<https://store.arduino.cc/products/arduino-mkr-wan-1300-lora-connectivity>

Figura 7 – Módulo Ublox Neo-6m GPS

https://www.amazon.com/dp/B01D1D0F5M/ref=mp_s_a_1_1_ssapa?crid=1TYNWZZTOJETE&keywords=Ublox+NEO-6M&qid=1658781144&sprefix=ublox+neo-6m%2Caps%2C493&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEExQzIPSkhJTTIBRUcyJmVuY3J5cHRIZElkPUEwMzY2MzQySDhRNvG3RUFUTzBMJmVuY3J5cHRIZEFkSWQ9QTA1MjkyOTNCNEs3TFJVVWlhVUEUmd2lkZ2V0TmFtZT1zcF9waG9uZV9zZWZyY2hfYXRmJmFidGlvbj1jbGlja1JIZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==