

x



ANDRÉ
MANUEL
FERREIRA
SANGUINETTI

ESTSchedule

Relatório de projeto de Mestrado em
Engenharia de Software

ORIENTADOR

Mestre Luís Miguel Lopes de Oliveira
Esteves, ESTSetúbal/IPS

Dezembro de 2020

ANDRÉ
MANUEL
FERREIRA
SANGUINETTI

ESTSchedule

JÚRI

Presidente: Doutor Cláudio Miguel Garcia
Loureiro dos Santos Sapateiro, ESTSetúbal/IPS

Orientador: Mestre Luís Miguel Lopes de
Oliveira Esteves, ESTSetúbal/IPS

Vogal: Especialista José Sena Pereira,
ESTSetúbal/IPS

Dezembro de 2020

Resumo

O projeto *ESTSchedule* é uma ferramenta desenvolvida no sentido de suportar a criação de previsões do número de estudantes para as unidades curriculares dos vários cursos da ESTSetúbal/IPS.

A escolha do tema *ESTSchedule* e conseqüente desenvolvimento desta solução, deriva da vontade do estudante aplicar os conhecimentos adquiridos ao longo do curso, aliada à colmatação de eventuais necessidades da Instituição.

As previsões são geradas tendo por base o histórico de inscrições nos anos letivos anteriores. Assim sendo, o projeto *ESTSchedule* tem uma componente que permite fazer gestão a nível de cursos/ramos e unidades curriculares, e – aliado às estatísticas existentes por curso/ramo e unidade curricular - podem-se gerar previsões de número de alunos para um determinado ano letivo.

Desenvolveu-se um projeto de raiz (*full-stack*), modelando uma base de dados e criando as camadas de serviço e de visualização. Estes três módulos são independentes uns dos outros, permitindo adaptar a novas tecnologias de uma maneira eficaz. A aplicação é acessível com qualquer dispositivo com um navegador de *Internet* e uma conexão à mesma.

Nas várias secções deste documento estão explicados em detalhe os diversos conceitos necessários à implementação, desde a parte técnica, à parte do negócio em si, com exemplos reais da aplicação em funcionamento.

Durante o desenvolvimento do projeto, as maiores dificuldades encontradas foram na determinação e implementação das regras de negócio. A parte técnica, como ficou bem planeada antes da implementação, não precisou de sofrer grandes alterações.

Palavras-chave: serviço, aplicação *web*, previsões, estatísticas, gestão

Abstract

The ESTSchedule project is a tool developed on the sense of allowing the creation of predictions of the number of students for the course units of the ESTSetúbal/IPS courses.

The theme ESTSchedule and respective development of the solution derives from the willness of applying all the knowledge acquired throughout the course by the student, hence bridging with eventual necessities of the Institution.

The predictions are done having in account the historical data of student assignments to the previous years. Having that, the ESTSchedule project has a component that allows the management at the courses/branches and unit courses level and – allied to the existing statistics per course/branch and unit course – it is allowed to generate predictions of the number of students for a given course year.

A full-stack project was developed from scratch, modelling a database and creating the service and visualization layers. These three modules are independent from each other, allowing to embrace new technologies in an effective manner. The web application is accessible from any device with an Internet browser and connection to it.

In the various sections of this document are explained in detail all the many concepts required for the implementation, starting with the architecture to the business itself, with real examples of the productionized version of the application.

During the development of the project, the biggest upsets were during the requirement and implementation phase of the business rules. The technical part, since it was well planned before the implementation, didn't require any major changes.

Keywords: Service, web application, predictions, statistics, management

Índice

Resumo	i
Abstract	ii
Índice	1
Lista de Figuras	3
Lista de Siglas e Acrónimos.....	4
1. Introdução	5
2. Arquitetura	6
2.1. Desenho de alto nível (<i>HLD</i>).....	6
2.1.1. Camada de interface de utilizador (<i>UI, front-end</i>).....	6
2.1.2. Camada de negócio (<i>business layer</i>)	9
2.2. Infraestrutura	11
3. Segurança	12
3.1. Autenticação	12
3.2. Rede (<i>Networking</i>).....	13
3.3. Negócio e serviço.....	14
3.3.1. Validações	14
3.3.2. Encriptação de passwords.....	16
4. Funcionalidades	17
4.1. Cursos/Ramos e Unidades Curriculares.....	17
4.1.1. Detalhes de um Curso	18
4.1.2. Associação de Unidades Curriculares.....	19
4.1.3. Unidades Curriculares	20
4.2. Previsões.....	21
4.2.1. Consulta de Previsões.....	21
4.2.2. Gerar Previsão.....	23
4.2.3. Histórico de alunos	26
4.3. Administração e perfis.....	27
5. Conclusões	31
6. Bibliografia.....	33
Anexo I - <i>Setup</i> inicial	34
Instalação.....	34
Configurações	34
Anexo II - <i>MySQL</i>	36
Configurações da instância <i>MySQL</i>	36
<i>Dump</i> exemplo.....	36
Carregamento de ficheiro <i>.sql</i> para a base de dados	37

DDL (Data Definition Language)	37
Anexo III - Definições de rede	40
Ativar port forwarding no router	40
Ativação do serviço Dynamic DNS	40
Anexo IV - Definições de SSL	41
Anexo V - Definições de servidor nginx	42
Anexo VI - Deployment do projeto	45
Deployment (camada de serviço)	45
Deployment (web-app)	48

Lista de Figuras

Figura 2.1.1 - Desenho de alto nível da arquitetura do projeto <i>ESTSchedule</i>	6
Figura 2.1.2 - <i>Layout</i> da aplicação a correr em modo <i>PWA (mobile)</i>	8
Figura 2.1.3 - Formulário de <i>login</i>	9
Figura 2.1.4 - Modelo relacional da base de dados	10
Figura 2.2.1 - Raspberry Pi 3B+ (imagem ilustrativa)	11
Figura 3.1.1 - Exemplo de um <i>JSON Web Token</i> codificado (esquerda) e da respetiva descodificação (direita)	12
Figura 3.3.1 - <i>Middleware</i> de validação de <i>JSON Web Token</i>	15
Figura 3.3.2 - <i>Middleware</i> de validação de perfil de utilizador	16
Figura 4 .1 - Aspeto do menu expandido (vista de administrador).....	17
Figura 4.1.1 - Ecrã de listagem de cursos de topologia “Licenciatura”	18
Figura 4.1.2 - Ecrã de detalhe de um curso	19
Figura 4.1.3 - Ecrã de associação de unidades curriculares a um curso/ramo.....	20
Figura 4.1.4 - Ecrã de listagem de unidades curriculares	21
Figura 4.2.1 - Ecrã de listagem de previsões	22
Figura 4.2.2 - Ecrã de detalhe de previsões.....	23
Figura 4.2.3 - Ecrã inicial de geração de previsões	24
Figura 4.2.4 - Ecrã auxiliar de estatísticas	25
Figura 4.2.5 - Formulário de modelos de previsão a considerar	25
Figura 4.2.6 - Resultado de uma previsão, por persistir.....	26
Figura 4.2.7 - Ecrã de histórico de alunos.....	27
Figura 4.3.1 - Ecrã de administração	28
Figura 4.3.2 - Detalhes de um utilizador (a partir do ecrã de administração).....	29
Figura 4.3.3 - Informações de perfil.....	29
Figura 4.3.4 - Formulário de alteração de detalhes básicos de utilizador	30

Lista de Siglas e Acrónimos

CSV	<i>Comma-Separated Values</i>
HLD	<i>High Level Design</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Tokens</i>
PWA	<i>Progressive Web Application</i>
REST	<i>Representational state transfer</i>
SSL	<i>Secure Sockets Layer</i>
TLS	<i>Transport Layer Security</i>
UI	<i>User Interface</i>
URL	<i>Uniform Resource Locator</i>
UX	<i>User Experience</i>

1. Introdução

O projeto *ESTSchedule* surge da vontade do aluno pretender desenvolver – de raiz – uma ferramenta que possa vir a ser útil à instituição onde estuda, conjugada com o desejo da real da ESTSetúbal/IPS, ao pretender obter um sistema centralizado que simule – ainda que parcialmente - o Sistema de Informação atual da Escola Superior de Tecnologia de Setúbal (no que a gestão de cursos/ramos e unidades curriculares diz respeito), assim como gerar previsões com base nos dados históricos disponíveis, utilizando diversos modelos de previsão. Com este projeto pretendeu-se aplicar todo o conhecimento adquirido ao longo do curso nas diversas áreas da Engenharia de *Software*, desde a segurança informática, qualidade de *software*, programação para a *web*, visualização de informação, entre outras.

Além disso, o sistema está criado de modo a poder ser evoluído e/ou substituir tecnologias parcialmente, sem nunca quebrar as restantes camadas do projeto. O projeto é uma prova de conceito pronta para ser instalada num ambiente produtivo, e ser gerido como tal.

Na próxima secção será apresentada a arquitetura de alto nível do projeto, seguida da descrição em detalhe da mesma, passando pela descrição das diversas camadas de segurança e explicação das demais funcionalidades deste. No final constam os anexos relevantes ao projeto. O código-fonte do projeto *ESTSchedule* está disponível num repositório privado do *GitHub* ^{[1], [2]}

2. Arquitetura

Nesta secção apresenta-se, de uma forma genérica, a arquitetura do projeto *ESTSchedule*. Utiliza-se uma arquitetura modular, que permite - a longo prazo - separar cada uma destas camadas. Tendo isto em consideração e, se for necessário, poder-se-á atualizar a tecnologia inerente a um dado módulo, sem afetar diretamente os restantes módulos.

2.1. Desenho de alto nível (HLD)

A arquitetura de alto nível do sistema, representada na figura 2.1.1, divide-se em dois grandes blocos distintos. A comunicação entre as duas camadas é feita com recurso a *Web Services* para a troca de informação. A comunicação é bi-lateral e o corpo dos pedidos pode ser efetuada nos formatos de *JSON* ou *CSV*.

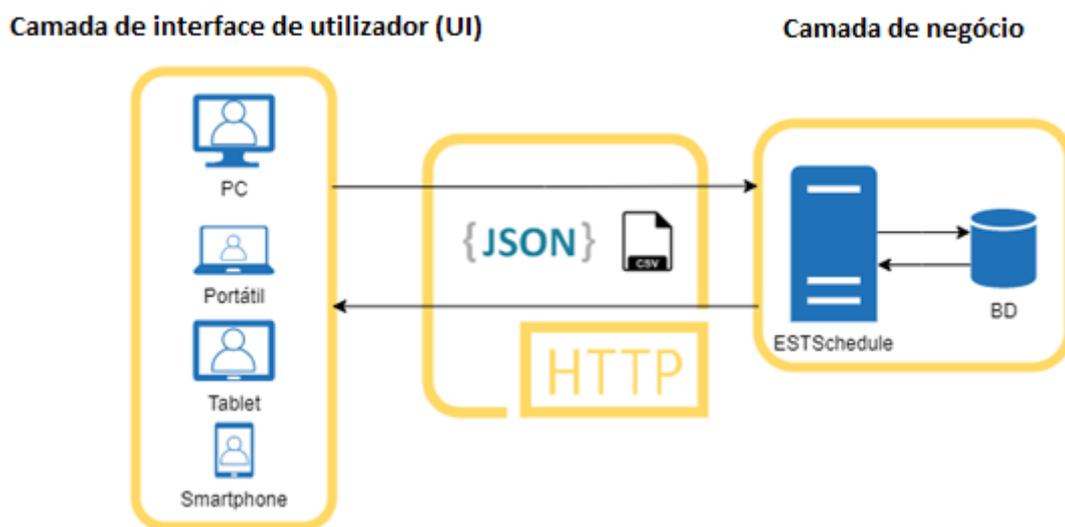


Figura 2.1.1 - Desenho de alto nível da arquitetura do projeto *ESTSchedule*

2.1.1. Camada de interface de utilizador (UI, front-end)

A camada de interface de utilizador é o local onde este último – a partir do seu computador pessoal, portátil, *tablet* ou *smartphone* – acede a uma aplicação *web* devidamente ajustada ao tamanho do ecrã do dispositivo, e efetua todas as operações que lhe são disponibilizadas. Esta camada está desenvolvida sobre uma combinação de duas *frameworks*: *VueJS* [3] e *Quasar* [4]. A escolha desta combinação de *frameworks* foi ponderada, sendo que a *VueJS* é uma *framework* de desenvolvimento em *JavaScript* para aplicações *web* com uma curva de desenvolvimento

baixa, é extremamente versátil e com uma *performance* exemplar. Por estes motivos é que faz parte das *frameworks* mais utilizadas por grandes corporações, à data deste documento. A escolha da *framework Quasar* serve para simplificar o processo de criação de *interfaces* para a *VueJS*, pois já tem todos os componentes de *User Interface* criados, poupando tempo de desenvolvimento e tendo a garantia que estes componentes irão proporcionar uma boa experiência ao utilizador. Ambas as *frameworks* primam pela sua completa documentação e contribuição da comunidade, no panorama atual.

Experiência de utilizador (*User Experience – UX*) é um tema relevante importante de mencionar. No seu conceito mais lato, pode ser definido como a experiência que um utilizador tem ao utilizar ou interagir com algo. Foca-se em tentar determinar e ajustar de acordo com as necessidades, habilidades, e mesmo limitações do próprio utilizador, tendo em conta o objetivo do negócio em questão. Na indústria das tecnologias de informação, o conceito engloba toda a experiência que o utilizador tem ao manusear a aplicação, desde por exemplo o *design*, ao conteúdo e à usabilidade. Boas práticas de *User Experience* promovem a melhoria da qualidade da interação do utilizador em todos estes aspetos. Mesmo sendo um conceito algo genérico, existem algumas métricas às quais uma equipa pode recorrer, a fim de garantir uma boa experiência de utilizador, sendo algumas delas, aplicando ao contexto das tecnologias de informação:

- Utilidade – A aplicação deve ser original e corresponder à necessidade do utilizador;
- Utilizável – A aplicação deverá ser simples e intuitiva de utilizar, com boa *performance*;
- Desejável – A aplicação deverá conter elementos visuais (por exemplo logotipo, identidade, temas) atrativos e que despoletem emoções positivas ao utilizador;
- Acessível – A aplicação deverá servir para pessoas com deficiências (por exemplo visuais, auditivas);

Nos dias que correm, é uma grande mais-valia ter uma aplicação com uma ótima experiência de utilizador, principalmente se formos donos de um negócio e queiramos expandir este para a *web*. É determinante ter uma aplicação que satisfaça as necessidades do utilizador/cliente em questão, uma vez que uma aplicação que não seja minimamente responsiva não irá – de modo algum – ter qualquer tipo de usabilidade a longo prazo. A fim de obter sucesso nos aspetos determinantes de uma boa experiência de utilizador, existem processos e metodologias que se podem aplicar para otimizar e obter os melhores resultados, destacando por exemplo:

- *Design* da Interface de utilizador, cujo principal foco passa por antecipar as ações que os utilizadores irão necessitar de executar, garantindo que essas mesmas ações são destacadas e de fácil acesso;
- Arquitetura da informação, cujo principal foco passa por moldar a maneira como a informação está organizada, estruturada e apresentada aos utilizadores
- Design visual, cujo principal foco passa por garantir que, esteticamente, a aplicação

é atrativa ao utilizador

A camada de *UI* (figura 2.1.2) foi desenvolvida de modo a permitir a utilização do sistema em diferentes tipos de dispositivos (*PC*, *Portátil*, *Tablet*, *Smartphone*). No que concerne aos dispositivos móveis, esta aplicação é disponibilizada na forma de *Progressive Web App (PWA)*, trazendo ao utilizador uma experiência semelhante a uma aplicação nativa do seu dispositivo. Ao aceder ao *URL* da aplicação pela primeira vez, é perguntado ao utilizador se quer instalar a aplicação no seu dispositivo. Todas as funcionalidades da aplicação *web* estão disponíveis tanto para *desktop* como para *mobile*, mudando apenas a disposição do conteúdo em alguns cenários.



Figura 2.1.2 - *Layout* da aplicação a correr em modo *PWA (mobile)*

Durante a experiência de utilização da interface, o utilizador poderá efetuar consultas e – eventualmente – manipulação de dados. Cada vez que uma operação deste género é efetuada, um *web service* é invocado, acionando os respetivos mecanismos de validação do pedido. Estas validações são feitas tanto do lado da interface (como por exemplo preenchimento de campos obrigatórios), como do lado do servidor (integridade da informação). O utilizador é notificado de toda e qualquer ação em que esteja a manipular informação.

A interface da aplicação é ajustada consoante o perfil do utilizador. Existem três perfis de utilizadores distintos: visualizador, editor e administrador. Detalhes sobre estes podem ser consultados na Secção 4 – *Administração e Perfis*.

Ao aceder pela primeira vez à aplicação, é pedido ao utilizador para inserir as suas credenciais no formulário de *login* (Ver figura abaixo). A sessão é válida durante uma hora. Após

esse tempo é necessário inserir novamente as credenciais.

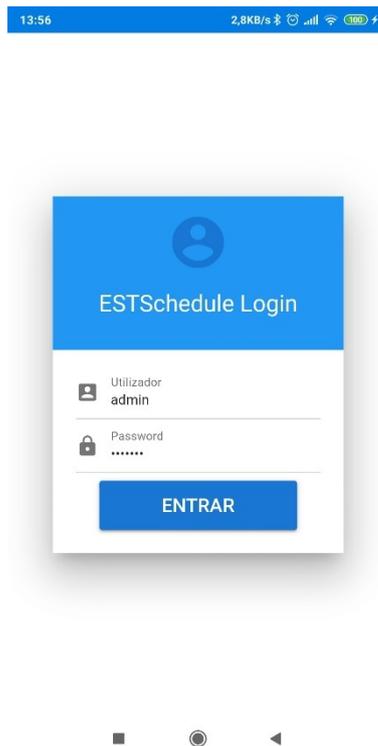


Figura 2.1.3 - Formulário de *login*

2.1.2. Camada de negócio (*business layer*)

A camada de negócio garante que as regras de negócio são cumpridas, permitindo separar a camada de *UI* da base de dados. Por regras de validação podem-se entender regras de validação de autenticação do utilizador (para garantir que ainda tem uma sessão válida, por exemplo), ou validar se um pedido está bem construído.

Esta camada está desenvolvida também em código *JavaScript*, com recurso à combinação de *frameworks NodeJS* [5] e *Express* [6]. A escolha destas *frameworks* também foi relativamente simples, uma vez que são amplamente utilizadas para este tipo de camada de negócio, no universo da programação no geral. A sua documentação é bastante rica e permite desenvolver aplicações leves e com uma elevada *performance*.

Os pedidos que provêm via *web services* são atendidos, validados e, conseqüentemente, realizados. Não há qualquer ligação direta entre a camada de experiência de utilizador e a base de dados.

A ligação à base de dados (ver modelo na figura abaixo) é efetuada através da camada de serviço, e é esta que – atendendo ao número de pedidos – utiliza uma *connection pool* pré-instanciada para efetuar as operações (*CRUD Create, Read, Update, Delete*).

A camada de serviço está desenvolvida sobre a *framework Express*, baseada em *NodeJS*. A base de dados utilizada é *MySQL*.

A base de dados está assente sobre três conceitos distintos:

- Gestão de cursos/ramos e unidades curriculares: Como que se de um Sistema de Informação se tratasse;
- Estatísticas e previsões: Onde se podem consultar estatísticas, dados históricos e gerar novas previsões sobre os dados já existentes;
- Administração: Gestão de utilizadores e permissões.

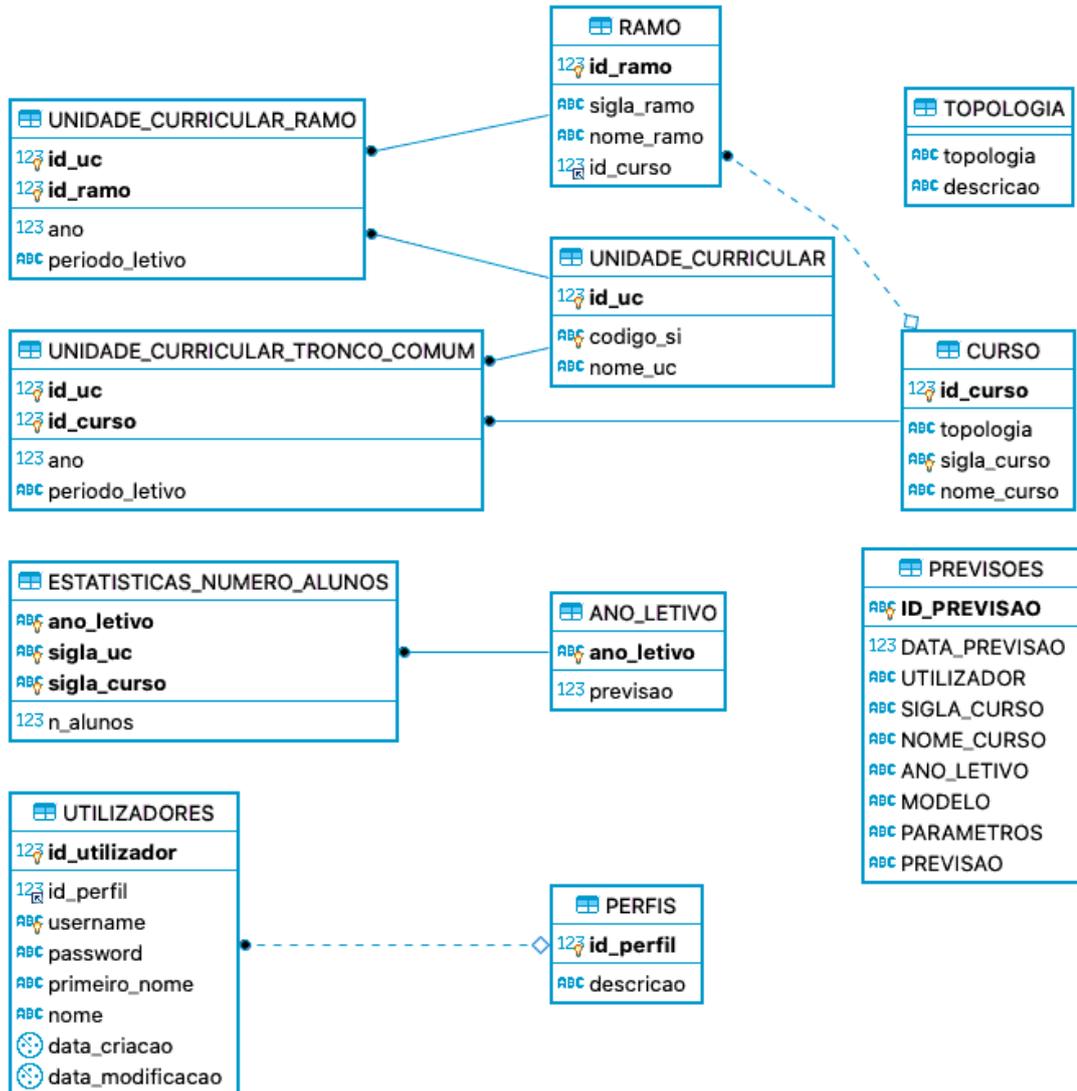


Figura 2.1.4 - Modelo relacional da base de dados

2.2. Infraestrutura

Todas as camadas do projeto (camada de experiência do utilizador e camada de negócio), estão instaladas no *Raspberry Pi 3B+* (ver figura abaixo), utilizando a rede caseira para *rotear* todo o tráfego da *Internet*. Todo o processo de instalação/manutenção e configurações estão disponíveis nos respetivos anexos deste documento.



Figura 2.2.1 - Raspberry Pi 3B+ (imagem ilustrativa)

Este *token* está desenhado para expirar a cada hora que passa após *login*, e é invalidado sempre que alguma alteração é feita ao nível do utilizador (como por exemplo a alteração de perfil), seja ela feita pelo próprio utilizador, ou não.

3.2. Rede (Networking)

Na vertente de *networking* do projeto, as medidas implementadas foram as seguintes:

TLS – TRANSPORT LAYER SECURITY [7]

O *Transport Layer Security (TLS)*, assim como o seu antecessor *Secure Sockets Layer (SSL)*, é um protocolo de segurança projetado para fornecer segurança nas comunicações sobre uma rede de computadores. Várias versões do protocolo encontram amplo uso em aplicações como navegação na *web*, *email*, mensagens instantâneas e voz sobre IP (*VoIP*). Os sites podem usar *TLS* para proteger todas as comunicações entre os seus servidores e navegadores *web*.

O protocolo *TLS* visa principalmente fornecer privacidade e integridade de dados entre duas ou mais aplicações que comuniquem entre elas. Quando protegidos por *TLS*, conexões entre um cliente (por exemplo, um navegador da *web*) e um servidor (por exemplo, *amfs.ddns.net*) devem ter uma ou mais das seguintes propriedades:

- A conexão é privada (ou segura) porque a criptografia simétrica é usada para encriptar os dados transmitidos. As chaves para essa encriptação simétrica são geradas exclusivamente para cada conexão e são baseadas num segredo compartilhado, negociado aquando o início da sessão. O servidor e o cliente negociam os detalhes de qual algoritmo de criptografia e chaves criptográficas utilizar antes do primeiro byte de dados ser sequer transmitido.
- A troca de informação é segura e confiável (nenhum invasor pode modificar as comunicações durante a negociação sem ser detectado).
- A identidade de ambas as partes da comunicação pode ser autenticada usando criptografia de chave pública. Essa autenticação pode ser opcional, mas geralmente é necessária para pelo menos uma das partes (geralmente o servidor).
- A conexão é confiável porque cada mensagem transmitida inclui uma verificação de integridade de mensagem usando um código de autenticação de mensagem para evitar perda não detectada ou alteração dos dados durante a transmissão.

HTTPS – HYPER TEXT TRANSFER PROTOCOL SECURE [8]

No cenário deste projeto em particular, todo o tráfego *HTTP* é redirecionado para *HTTPS*. *HTTPS (Hyper Text Transfer Protocol Secure - protocolo de transferência de hipertexto seguro)* é uma implementação do protocolo *HTTP* sobre uma camada adicional de segurança que utiliza

o protocolo *SSL/TLS*. Essa camada adicional permite que os dados sejam transmitidos através de uma conexão encriptada e que se verifique a autenticidade do servidor e do cliente por meio de certificados digitais. A porta *TCP* usada por norma para o protocolo *HTTPS* é a 443.

O protocolo *HTTPS* é utilizado, em regra, quando se deseja evitar que a informação transmitida entre o cliente e o servidor seja visualizada por terceiros, como por exemplo no caso de compras *online*. A existência na barra de endereços de um cadeado (que pode ficar do lado esquerdo ou direito, dependendo do navegador utilizado) demonstra a certificação de página segura (*SSL/TLS*). A existência desse certificado indica o uso do protocolo *HTTPS* e que a comunicação entre o browser e o servidor se dará de forma segura.

3.3. Negócio e serviço

Por segurança ao nível de negócio e serviço, entendem-se as validações que permitem garantir que a persistência dos dados está de acordo com o modelo definido. Além disso, as validações de *tokens*, assim como a encriptação de *passwords* explicadas a seguir fazem parte deste ponto da segurança.

3.3.1. Validações

A cada operação, o respetivo *token JWT* é validado, sendo que a operação em questão não prosseguirá se o mesmo for inválido (por exemplo se tiver sido manipulado, ou se tiver expirado), tal como se pode verificar no excerto de código-fonte na figura abaixo. Esta operação é efetuada para cada pedido ao respetivo *web service*, com recurso a um *middleware*.

```

const jwt = require("jsonwebtoken");
const config = require("../config/config");
const utilizadoresCtrl = require("../controller/UtilizadoresController");

/**
 * Função de validação de JSON Web Token
 *
 * @param {*} req
 * @param {*} res
 * @param {*} next
 */
function validateJwt(req, res, next) {
  var token = req.headers["authorization"];

  if(!token) {
    token = req.body.token
  }

  if (token) {
    token = token.replace("Bearer ", "");
    jwt.verify(token, config.jwt.secret, function (err, decoded) {
      if (err) {
        console.log('error', err)
        res.status(401).json({ message: "Sessão expirada" });
      } else {
        // Validar se utilizador existe
        utilizadoresCtrl.getUtilizador(decoded.username).then((user) => {
          if(user.length === 0) {
            res.status(403).json({ message: "Utilizador inexistente" });
          }

          // Ver se houve alterações no utilizador. Se sim, invalidar sessão
          if (decoded.data_modificacao !== null) {
            if (new Date(decoded.data_modificacao).getTime() !== user[0].data_modificacao.getTime()) {
              res.status(403).json({ message: "Tem de iniciar sessão novamente" });
            }
          }
        }).catch(err => {
          res.status(403).json({ message: "Erro ao validar utilizador" });
        })
        // if everything is good, save to request for use in the following routes
        req.decoded = decoded;
        next();
      }
    });
  } else {
    res.status(403).json({ message: "É necessário estar autenticado" });
  }
}

```

Figura 3.3.1 - *Middleware* de validação de *JSON Web Token*

No cenário em que o respetivo *token* é válido, é avaliado se o utilizador tem permissões para aceder a determinada operação, também com recurso a um *middleware* do *Express* (ver figura abaixo). Existem três perfis de utilizador, a fim de validar se este tem acesso ao recurso pretendido:

1. Visualizador (*viewer*): Pode executar a maior parte das operações de leitura;
2. Editor: Pode executar a maior parte das operações de leitura e de escrita;
3. Administrador: Pode executar todo o tipo de operações, sem restrição, em qualquer *web service*.

```

/**
 * Valida se o utilizador tem o role necessário
 * @param {*} role
 */
function hasRole(role) {
  return function (req, res, next) {
    if (req.decoded) {
      if (role > req.decoded.id_perfil) {
        res.status(403).json({ message: "Não tem autorização para fazer esta operação" });
      } else {
        next()
      }
    } else {
      res.status(403).json({ message: "É necessário estar autenticado" });
    }
  }
}

module.exports = {
  isViewer: hasRole(1),
  isEditor: hasRole(2),
  isAdmin: hasRole(3)
}

```

Figura 3.3.2 - *Middleware* de validação de perfil de utilizador

3.3.2. *Encriptação de passwords*

As *passwords* de cada utilizador ficam registadas na base de dados, utilizando a técnica sal (*salt*). Em criptografia, sal é uma porção de dados aleatória, utilizada como *input* adicional numa função que manipula uma palavra-chave (*hash*). É utilizada para salvaguardar *passwords*.

De um modo genérico, estas regras de segurança implementadas não deterioram significativamente a *performance* da aplicação em ambiente de produção, não acarretam custos acrescidos (zero licenças), e são de fácil implementação/manutenção.

4. Funcionalidades

O projeto *ESTSchedule* tem diversas funcionalidades implementadas (ver figura abaixo). Estas funcionalidades diferem para cada perfil de utilizador, as quais são explicadas durante esta secção. No *menu* (ver figura abaixo) existem diversas secções. A secção de “Cursos” está separada por topologias (CTeSP, Licenciaturas, Mestrados) e serve para fazer gestão a estes. A secção “Unidades Curriculares” permite fazer gestão às UCs. O separador “Previsões” permite obter uma listagem de previsões, gerar estas, assim como consultar informação relativa ao histórico de número alunos por unidade curricular.

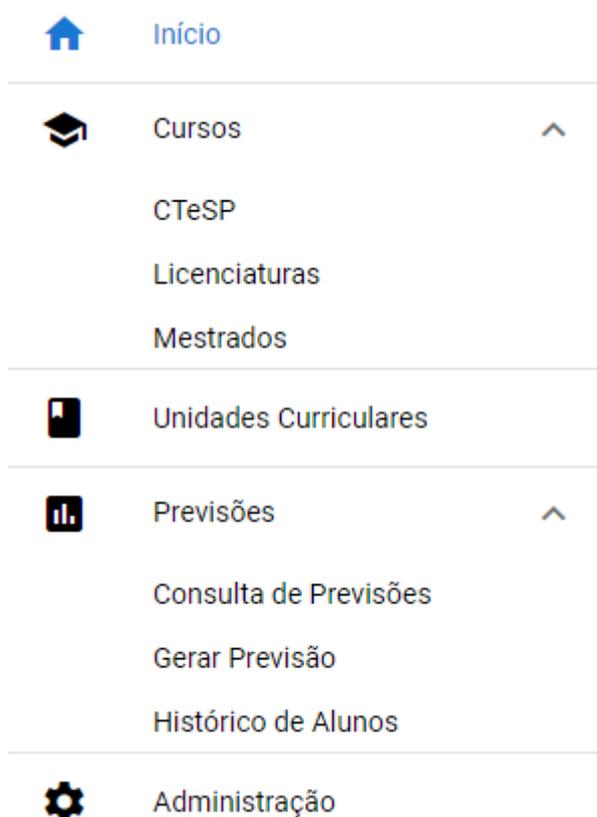


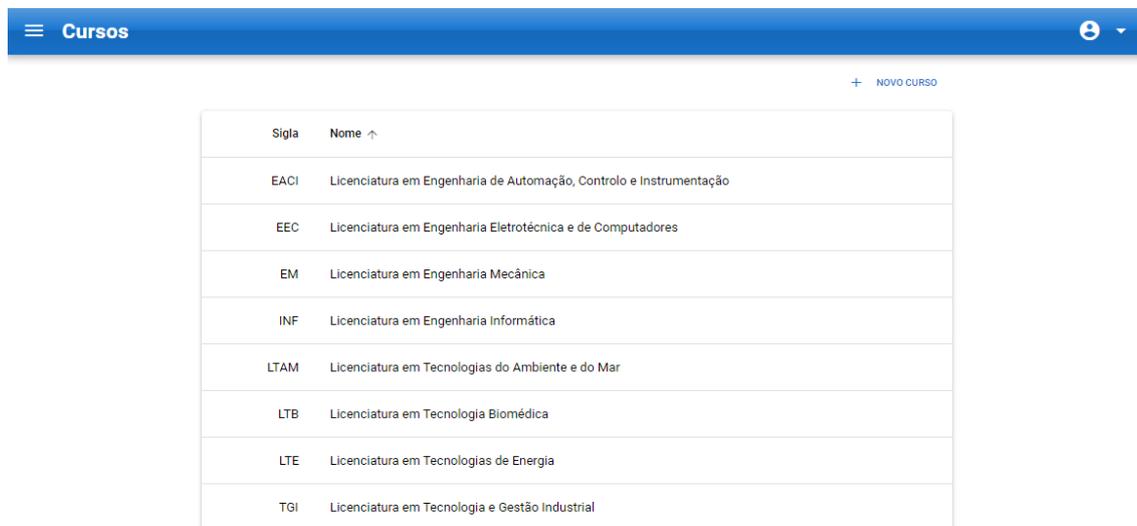
Figura 3.3 .1 - Aspeto do menu expandido (vista de administrador)

4.1. Cursos/Ramos e Unidades Curriculares

Relativamente aos cursos podemos:

- Adicionar um curso
- Consultar um curso
- Modificar um curso
- Eliminar um curso

É no separador “Cursos” que podemos obter informações sobre os mesmos, separados por tipologia, tal como se pode verificar na figura abaixo.



The screenshot shows a web interface with a blue header bar containing a hamburger menu icon, the text 'Cursos', and a user profile icon. Below the header, there is a '+ NOVO CURSO' link. The main content is a table with two columns: 'Sigla' and 'Nome'. The table lists ten different engineering and technology courses.

Sigla	Nome ↑
EACI	Licenciatura em Engenharia de Automação, Controlo e Instrumentação
EEC	Licenciatura em Engenharia Eletrotécnica e de Computadores
EM	Licenciatura em Engenharia Mecânica
INF	Licenciatura em Engenharia Informática
LTAM	Licenciatura em Tecnologias do Ambiente e do Mar
LTB	Licenciatura em Tecnologia Biomédica
LTE	Licenciatura em Tecnologias de Energia
TGI	Licenciatura em Tecnologia e Gestão Industrial

© Setúbal 2020

Figura 4.1.1 - Ecrã de listagem de cursos de tipologia “Licenciatura”

4.1.1. Detalhes de um Curso

No ecrã de detalhes de curso (figura abaixo), é possível obter quais são as Unidades Curriculares que estão associadas ao mesmo, assim como eventuais ramos que existam neste. É possível alterar o nome visível do curso. É também possível eliminar este curso ou só um ramo do curso, sendo que as associações às unidades curriculares também desaparecerão.

A criação de ramos permite adicionar, ao curso em questão um novo ramo. Após esse cenário é possível começar a associar unidades curriculares a este.

Validações:

- Não é possível criar cursos com siglas iguais
- Não é possível criar ramos com siglas iguais

☰ Detalhes Curso 👤

NOVO RAMO ELIMINAR CURSO VOLTAR

Licenciatura em Engenharia Informática ✓

Plano de Estudos

Tronco Comum GERIR UCS

Código	Nome	Ano	Período
INF32144	Matemática I	1	1S
INF32145	Economia e Gestão	1	1S
INF32146	Introdução à Programação Orientada por Objetos	1	1S
INF32147	Métodos de Estudo e Comunicação	1	1S
INF32148	Lógica Computacional	1	1S
INF32149	Sociologia das Organizações e Inovação	1	2S
INF32150	Programação Orientada por Objetos	1	2S
INF32151	Análise Numérica	1	2S
INF32152	Bases de Dados	1	2S
INF32153	Algoritmos e Tipos Abstratos de Dados	1	2S

Ramo de Engenharia de Software GERIR UCS

Código	Nome	Ano	Período
INF32154	Sistemas Operativos	2	1S
INF32155	Matemática Discreta	2	1S
INF32156	Interação Pessoa-Máquina	2	1S
INF32157	Programação Avançada	2	1S
INF32158	Complementos de Bases de Dados	2	1S
INF32169	Programação para a Internet	2	2S
INF32170	Probabilidades e Estatística	2	2S
INF32171	Modelação de Sistemas de Informação	2	2S
INF32172	Computação Distribuída	2	2S
INF32173	Aplicações Multimédia	2	2S

© Setúbal 2020

Figura 4.1.2 - Ecrã de detalhe de um curso

4.1.2. Associação de Unidades Curriculares

Ainda dentro do ecrã de detalhes de curso, ao clicar no botão “Gerir UCs”, é possível associar/eliminar uma ou mais unidades curriculares ao tronco comum de um curso (ver figura abaixo), ou a um dos seus ramos. Não é possível um curso/ramo ter a mesma UC associada no mesmo tronco comum/ramo. A fim de facilitar o utilizador, está disponibilizada uma caixa de pesquisa, que filtra tanto por sigla como por nome da unidade curricular.

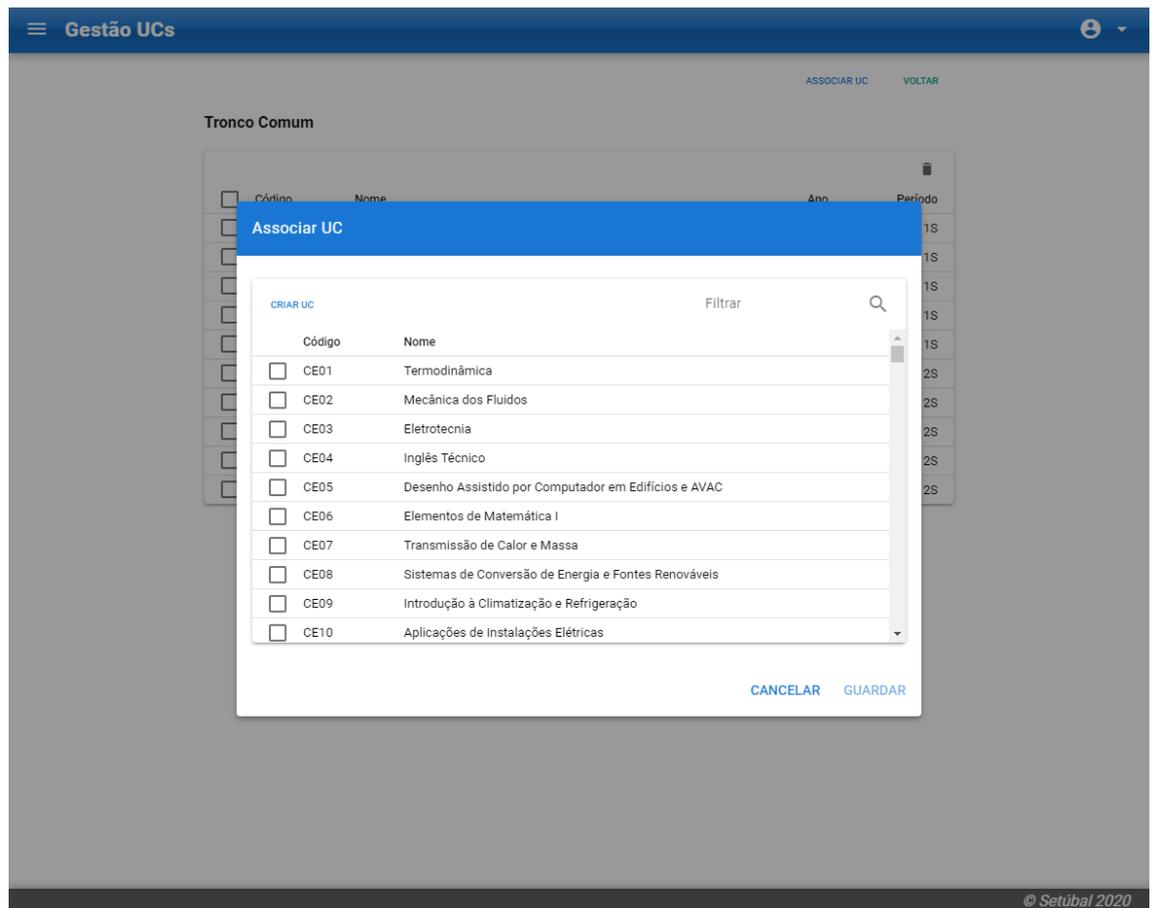


Figura 4.1.3 - Ecrã de associação de unidades curriculares a um curso/ramo

4.1.3. Unidades Curriculares

Neste ecrã (ver figura abaixo) é possível obter uma listagem de todas as unidades curriculares disponíveis para consulta. É possível criar unidades curriculares novas, sendo que a sigla desta é única. Além disso, também é possível eliminar uma ou mais unidades curriculares, ou pesquisar por estas. A aplicação permite também editar o nome visível de uma unidade curricular.

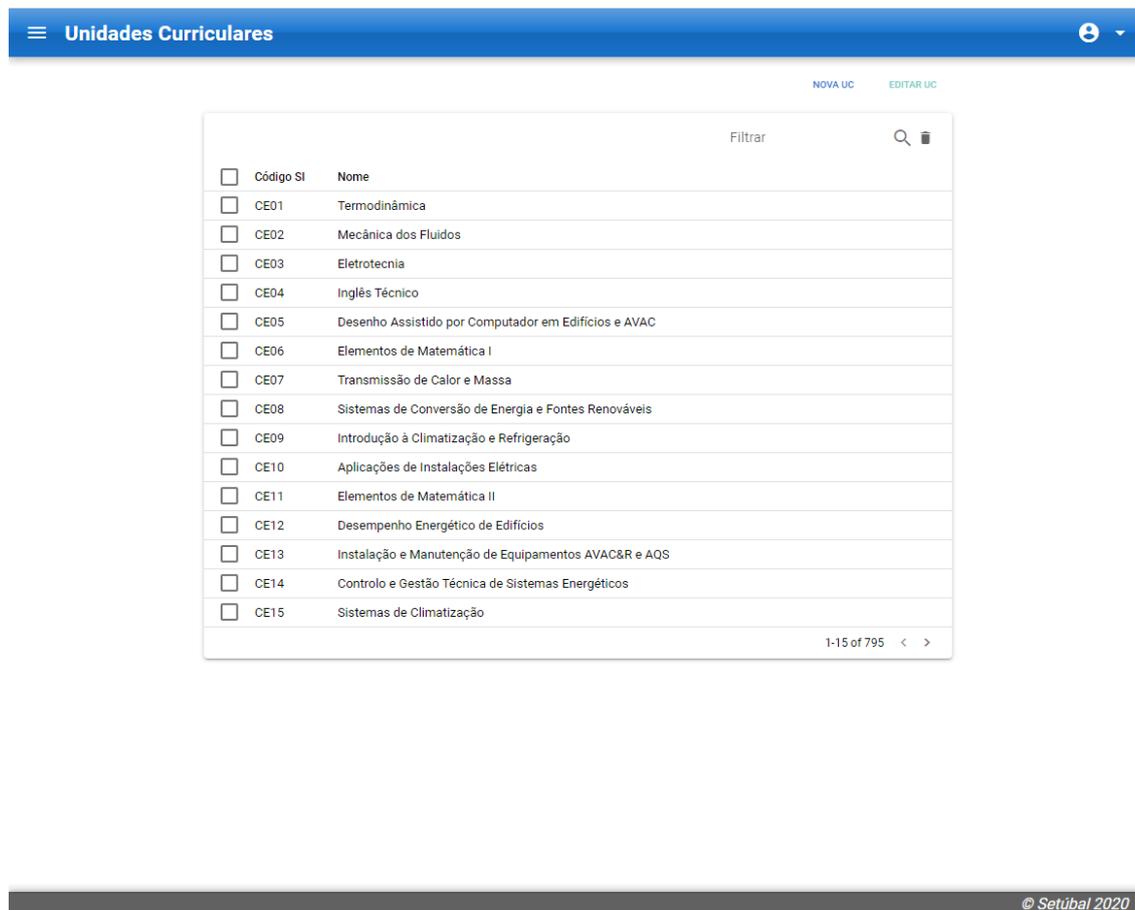


Figura 4.1.4 - Ecrã de listagem de unidades curriculares

4.2. Previsões

Este é o separador em que é possível consultar e gerir previsões (criar, eliminar), e obter informação estatística sobre o número de alunos, por ano, por unidade curricular.

4.2.1. Consulta de Previsões

No ecrã de consulta de previsões (ver figura abaixo) surge um modelo tabular de todas as previsões que foram guardadas na base de dados. É possível filtrar previsões através da caixa de texto respetiva, assim como eliminar uma ou mais previsões geradas. Neste ecrã está indicado qual foi o user que gerou a previsão, a que data, para que ano letivo, curso e qual o modelo aplicado para a previsão.

☰ Consulta de Previsões 👤

Filtrar 🔍

<input type="checkbox"/>	Data	Utilizador	Ano Letivo	Sigla Curso	Modelo Aplicado
<input type="checkbox"/>	01/07/2020 23:38:45	admin	2018/2019	EEC	Modelo 3 - Média móvel ponderada
<input type="checkbox"/>	01/07/2020 23:38:45	admin	2018/2019	EEC	Modelo 1 - Média últimos anos
<input type="checkbox"/>	01/07/2020 23:38:45	admin	2018/2019	EEC	Modelo 2 - Média dos declives
<input type="checkbox"/>	01/07/2020 23:38:45	admin	2018/2019	EACI	Modelo 1 - Média últimos anos
<input type="checkbox"/>	01/07/2020 23:38:45	admin	2018/2019	EACI	Modelo 2 - Média dos declives
<input type="checkbox"/>	01/07/2020 23:38:45	admin	2018/2019	EACI	Modelo 3 - Média móvel ponderada

1-6 of 6

© Setúbal 2020

Figura 4.2.1 - Ecrã de listagem de previsões

Ao clicar sobre um registo, podemos obter informação detalhada sobre o mesmo (ver figura abaixo). Neste ecrã, além de aparecer a informação que consta na listagem anterior, é possível saber quais foram os parâmetros aplicados, e saber os resultados em detalhe. As colunas **diferença** e **(%)** referem-se sempre ao ano letivo anterior, em questão para esta previsão.

É também possível exportar os resultados para CSV, através do botão respetivo.

O botão **voltar** permite regressar à listagem.

Data de geração:
01/07/2020 23:38:45

Gerado por:
admin

Previsão para:
2018/2019

Curso:
Licenciatura em Engenharia Eletrotécnica e de Computadores

Modelo:
Modelo 3 - Média móvel ponderada

Parâmetros utilizados:
{ "anos_anteriores": ["2015/2016", "2016/2017", "2017/2018"], "pesos": ["1", "2", "5"] }

Resultados					
Sigla	Nome	Alunos Ano Anterior	Previsão	Diferença	(%)
LEEC11101	Matemática I	74	83	+9	+12.16%
LEEC11102	Introdução à Engenharia Eletrotécnica	37	35	-2	-5.41%
LEEC11103	Introdução à Programação	48	50	+2	+4.17%
LEEC11104	Eletrotecnia I	72	75	+3	+4.17%
LEEC11105	Mecânica	67	69	+2	+2.99%
LEEC12106	Matemática II	76	77	+1	+1.32%
LEEC12108	Eletrotecnia II	108	114	+6	+5.56%
LEEC12110	Sistemas Digitais I	50	52	+2	+4.00%
LEEC21111	Matemática Aplicada	80	80	0	0.00%
LEEC21113	Eletrónica I	30	32	+2	+6.67%
LEEC22116	Controlo	61	64	+3	+4.92%
LEEC22117	Probabilidades e Estatística	76	74	-2	-2.63%
LEEC31121	Economia e Gestão	42	40	-2	-4.76%
LEEC12107	Equipamentos e Esquemas Eléctricos	12	16	+4	+33.33%
LEEC12109	Termodinâmica	14	17	+3	+21.43%

Figura 4.2.2 - Ecrã de detalhe de previsões

4.2.2. Gerar Previsão

Neste ecrã (ver figura abaixo) podemos seleccionar um ou mais cursos para gerar previsões, com base no ano letivo configurado pelo grupo de administração. É possível fazer um pré-filtro de cursos a seleccionar, utilizando a caixa de pesquisa indicada para tal. Ao clicar sobre qualquer curso, um ecrã auxiliar aparecerá, mostrando todas as estatísticas desse curso (figura 4.2.4), ao longo dos anos letivos.

Os modelos disponíveis para efetuar previsões são os seguintes:

1. Média de últimos anos (média aritmética simples) [9]. É obtida dividindo-se a soma das observações pelo número delas. Se tivermos uma série de n valores de uma variável x, a média aritmética simples será determinada pela seguinte expressão:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i$$

Equação 1 – Fórmula matemática da média aritmética simples

2. Média dos declives. Neste modelo calcula-se em todas as subsequências de anos letivos o declive em relação ao ano n – 1. Obtendo todos esses valores, determinamos a média dos declives

3. Média móvel ponderada [10]. Resulta na sequência das médias ponderadas por pesos diferentes, cuja fórmula é dada pela seguinte maneira:

$$\bar{p}_t = \frac{p_i + 1\omega_1 + \dots + p_i + n\omega_n}{\sum_{j=1}^n \omega_j}$$

Equação 2 – Fórmula matemática da média móvel ponderada

Gerar Previsão

Previsão para o ano letivo
2018/2019

Cursos Filtrar 🔍

<input type="checkbox"/>	Sigla	Nome
<input type="checkbox"/>	APIEF	Curso Técnico Superior Profissional em Climatização e Energia - APIEF
<input type="checkbox"/>	ARCIL	Curso Técnico Superior Profissional em Automação, Robótica e Controlo Industrial - ETLA (Sines)
<input type="checkbox"/>	CINEL	Curso Técnico Superior Profissional em Eletromedicina - CINEL (Lisboa)
<input type="checkbox"/>	DVAM	Curso Técnico Superior Profissional em Desenvolvimento de Videojogos e Aplicações Multimédia
<input type="checkbox"/>	GAIR	Curso Técnico Superior Profissional em Produção Aeronáutica - Centro Aeronáutico de Ponte de Sor
<input type="checkbox"/>	IPE	Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação - IPE (Lisboa)
<input type="checkbox"/>	IPESEC	Curso Técnico Superior Profissional em Sistemas Eletrónicos e Computadores - IPE (Lisboa)
<input type="checkbox"/>	QAA	Curso Técnico Superior Profissional em Qualidade Ambiental e Alimentar
<input type="checkbox"/>	TPSIB	Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação (Turma do Programa BrightS)
<input type="checkbox"/>	TSPARC	Curso Técnico Superior Profissional em Automação, Robótica e Controlo Industrial
<input type="checkbox"/>	TSPCDA	Curso Técnico Superior Profissional em Programação Web, Dispositivos e Aplicações Móveis

Modelo(s) de previsão:

1 - Média últimos anos

2 - Média dos declives

3 - Média móvel ponderada

[LIMPAR](#) [GERAR](#)

© Setúbal 2020

Figura 4.2.3 - Ecrã inicial de geração de previsões

Estatísticas (EEC)						
Sigla	Nome	2014/2015	2015/2016	2016/2017	2017/2018	201
LEEC11101	Matemática I	131	135	80	74	
LEEC11102	Introdução à Engenharia Eletrotécnica	40	41	26	37	
LEEC11103	Introdução à Programação	54	79	41	48	
LEEC11104	Eletrotecnia I	99	91	75	72	
LEEC11105	Mecânica	69	98	58	67	
LEEC12106	Matemática II	83	100	67	76	
LEEC12108	Eletrotecnia II	179	140	114	108	
LEEC12110	Sistemas Digitais I	81	65	51	50	
LEEC21111	Matemática Aplicada	85	81	78	80	
LEEC21113	Eletrónica I	51	43	30	30	

Figura 4.2.4 - Ecrã auxiliar de estatísticas

À medida que o utilizador seleciona cada modelo de previsão, as respetivas parametrizações destes aparecem e têm de ser preenchidas, de acordo com as regras de validação (ver figura abaixo).

Modelo(s) de previsão:

1 - Média últimos anos

Nº de últimos anos a considerar (min. 2)
3

2 - Média dos declives

Nº de últimos anos a considerar (min. 3)
3

3 - Média móvel ponderada

Nº de últimos anos a considerar (min. 2)
3

Pesos:

2015/2016
1

2016/2017
2

2017/2018
3

Figura 4.2.5 - Formulário de modelos de previsão a considerar

Uma vez tendo o formulário válido, é possível submetê-lo para gerar a previsão. Esta previsão é meramente descritiva até que o utilizador analise os resultados e os pretenda persistir, a partir do botão de **Gravar** (ver figura abaixo).

Sigla	Nome	Alunos Ano Anterior	Previsão	Diferença	(%)
LACI11001	Matemática I	51	62	+11	+21.57%
LACI11002	Eletrotecnia	18	25	+7	+38.89%
LACI11003	Química	56	49	-7	-12.50%
LACI11004	Introdução EACI	18	19	+1	+5.56%
LACI11005	Introdução à Programação	47	47	0	0.00%

Records per page: 5 1-5 of 30 < >

FECHAR

Figura 4.2.6 - Resultado de uma previsão, por persistir.

4.2.3. Histórico de alunos

No ecrã de histórico de alunos (ver figura abaixo), podemos consultar e filtrar estatísticas de todas as unidades curriculares de cada curso, ao longo dos anos letivos, quantos alunos estas tiveram.

É possível adicionar registos um a um, ou então importar um ficheiro CSV com estes valores. Exemplo de um CSV com dois registos:

```
ano_letivo,sigla_uc,sigla_curso,n_alunos
2014/2015,EM11200,EM,141
2014/2015,EM11201,EM,49
```

Para importar, o ano letivo tem de ser válido e respeitar a máscara **XXXX/XXXX**, e o curso tem de ter a respetiva unidade curricular associada no momento. O número de alunos deve ser um valor inteiro. Qualquer registo que não esteja de acordo com estas normas é ignorado. Também é possível seleccionar uma gama de resultados e exportá-los para CSV.

Histórico de Alunos ADICIONAR REGISTO IMPORTAR CSV

eecl 🔍 🗑️ 📄

<input type="checkbox"/> Ano Letivo ↓	Sigla UC	Sigla Curso	Nº de Alunos
<input type="checkbox"/> 2017/2018	LEEC11101	EEC	74
<input type="checkbox"/> 2017/2018	LEEC11102	EEC	37
<input type="checkbox"/> 2017/2018	LEEC11103	EEC	48
<input type="checkbox"/> 2017/2018	LEEC11104	EEC	72
<input type="checkbox"/> 2017/2018	LEEC11105	EEC	67
<input type="checkbox"/> 2017/2018	LEEC12106	EEC	76
<input type="checkbox"/> 2017/2018	LEEC12107	EEC	12
<input type="checkbox"/> 2017/2018	LEEC12108	EEC	108
<input type="checkbox"/> 2017/2018	LEEC12109	EEC	14
<input type="checkbox"/> 2017/2018	LEEC12110	EEC	50
<input type="checkbox"/> 2017/2018	LEEC12130	EEC	37
<input type="checkbox"/> 2017/2018	LEEC12131	EEC	45
<input type="checkbox"/> 2017/2018	LEEC21111	EEC	80
<input type="checkbox"/> 2017/2018	LEEC21112	EEC	21
<input type="checkbox"/> 2017/2018	LEEC21113	EEC	30
<input type="checkbox"/> 2017/2018	LEEC21114	EEC	12
<input type="checkbox"/> 2017/2018	LEEC21115	EEC	13
<input type="checkbox"/> 2017/2018	LEEC21134	EEC	37
<input type="checkbox"/> 2017/2018	LEEC21135	EEC	26
<input type="checkbox"/> 2017/2018	LEEC21136	EEC	25
<input type="checkbox"/> 2017/2018	LEEC21152	EEC	1
<input type="checkbox"/> 2017/2018	LEEC21153	EEC	2
<input type="checkbox"/> 2017/2018	LEEC22116	EEC	61
<input type="checkbox"/> 2017/2018	LEEC22117	EEC	76

© Setúbal 2020

Figura 4.2.7 - Ecrã de histórico de alunos

4.3. Administração e perfis

No ecrã de administração (ver figura abaixo) é possível efetuar algumas parametrizações globais do projeto. Também é possível criar utilizadores, eliminar ou alterar o perfil destes.

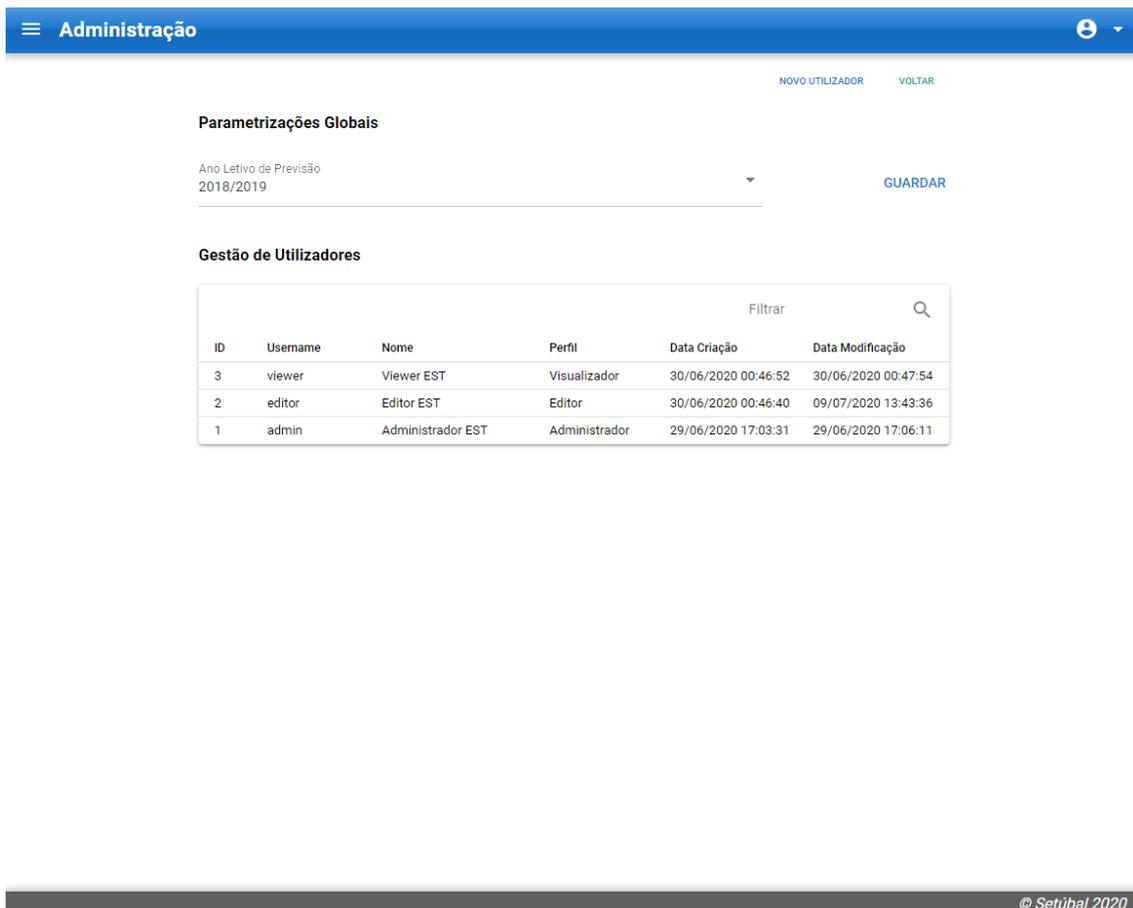


Figura 4.3.1 - Ecrã de administração

Os três perfis de utilizadores existentes são (ver figura abaixo):

- Visualizador: Permite efetuar todas as operações de consulta – à exceção das operações de administração.
- Editor: Além de visualizar, permite efetuar todas as operações de escrita de conteúdo – à exceção das operações de administração
- Administrador: Permite efetuar todo o tipo de operações na aplicação.

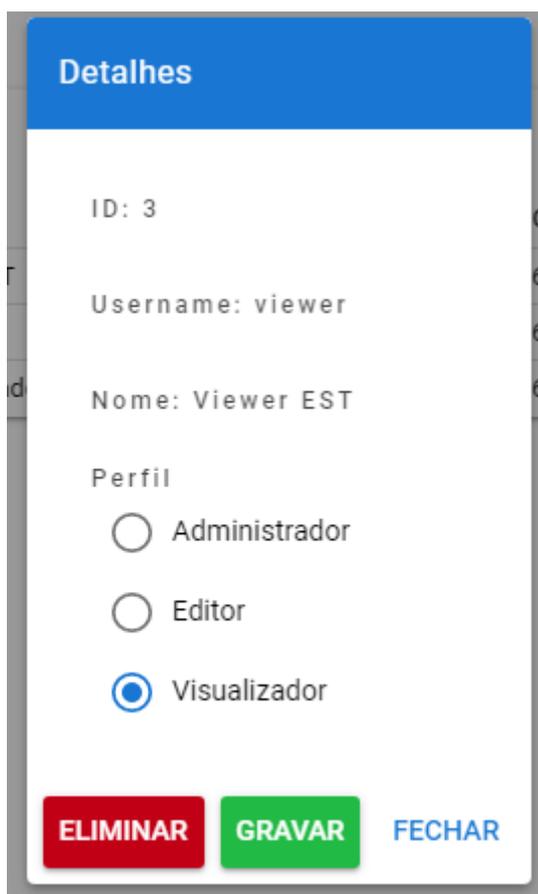


Figura 4.3.2 - Detalhes de um utilizador (a partir do ecrã de administração)

No topo direito do ecrã, além de estar disponível o botão de **Sair**, também é possível alterar a senha de acesso ao portal, do utilizador, assim como modificar informações básicas sobre este (nomes), tal como descrito na figura abaixo. Qualquer alteração nestes campos irão determinar a sessão ser expirada automaticamente, forçando o utilizador a fazer *login* de novo.



Figura 4.3.3 - Informações de perfil

Definições do utilizador admin

Primeiro e último nome
Administrador EST

Nome (Visível) (máx 20 caracteres)
Admin

CANCELAR **GUARDAR**

Figura 4.3.4 - Formulário de alteração de detalhes básicos de utilizador

5. Conclusões

Neste trabalho abordámos o assunto de criar uma aplicação *full-stack* de raiz aplicada a uma necessidade real (geração de previsões de números de alunos por unidade curricular) e pode-se concluir que é possível criar um sistema que efetue o tratamento de algum volume de dados, a baixo custo (*Raspberry Pi*), com resiliência, segurança e *performance*, aliando os conhecimentos técnicos adquiridos ao longo do Mestrado.

Num sentido lato, a pragmática de gerar previsões de número de alunos por unidade curricular foi cumprida, de acordo com o objetivo do projeto. Numa fase inicial de levantamento de requisitos, não foi contemplada uma possível análise dos resultados obtidos, contudo ficam aqui descritos alguns comentários aos mesmos, assim como conhecidas limitações destes:

- A qualidade dos resultados obtidos não está materializada nem é possível de quantificar, de uma maneira determinística. Isto é, o utilizador tem a possibilidade de gerar previsões, contudo serve somente a sua intuição de perceber se estes resultados são considerados “bons” ou “maus” e guardar os que realmente pretende.
- Os resultados apresentados carecem de eventual demonstração gráfica. Apesar de ser possível exportar os resultados das previsões e – por conseguinte – importar para outras ferramentas (*Excel*, *Tableau*, entre outras), a criação de *dashboards* com estes traria outro nível de utilidade à aplicação.
- É possível realimentar o circuito com dados gerados nas previsões, contudo é um processo considerado semiautomático, sendo que este processo poderia ser mais amigável, do ponto de vista do utilizador.

Em virtude do que foi mencionado nestes pontos e, sabendo que estas limitações são conhecidas, a não-implementação destas funcionalidades deve-se – em parte – ao facto de boa parte do tempo investido ao longo do projeto foi em garantir que todo o núcleo deste garante a respetiva integridade e resiliência ao nível técnico, que uma infraestrutura deste género necessita.

Como trabalho futuro considero que se pode divergir para dois ramos:

- Um ramo de manutenção, em que se podem adicionar novas funcionalidades aos ecrãs já existentes, como por exemplo adicionar novos modelos de previsão, com outro tipo de algoritmos;
- Um ramo evolutivo, em que se implementam novos conceitos de raiz. Uma temática interessante seria adicionar algoritmos que, alimentando e cruzando com outras fontes de dados, poder-se-ia – por exemplo – gerar os horários das diversas

turmas.

De referir que este trabalho foi muito importante para o aprofundamento deste tema, uma vez que me permitiu compreender melhor como desenvolver um projeto de raíz, além de me ter permitido aperfeiçoar competências técnicas e de investigação.

6. Bibliografia

[1] (Março 2020) <https://quasar.dev/>

[2] (Março 2020) <https://vuejs.org/>

[3] (Março 2020) <https://nodejs.org/en/>

[4] (Março 2020) <https://expressjs.com/>

[5] (Maio 2020) <https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/>

[6] (Maio 2020) <https://www.cloudflare.com/learning/ssl/what-is-https/>

[7] (Julho 2020) <https://www.todamateria.com.br/media/>

[8] (Julho 2020) <https://dicionariodoinvestidor.com.br/content/o-que-e-media-movel-aritmetica-simples-mma-mms/>

[9] (Novembro 2020) <https://www.usability.gov/what-and-why/user-experience.html>

[10] (Novembro 2020) <https://usabilitygeek.com/big-differences-ux-ui-design/>

Anexo I - Setup inicial

Neste anexo está indicado todo o processo de instalação da máquina servidor.

Imagem *unix* utilizada: **2020-02-13-raspbian-buster-lite.img**

Todo o código está disponível nos repositórios *GitHub* seguintes:

<https://github.com/andresanguinetti/estschedule-service/>

<https://github.com/andresanguinetti/estschedule-webapp>

Instalação

1. Transferir a imagem do *Raspbian Lite* através da [página oficial de download do Raspbian](#)
2. Com o cartão *micro SD* inserido no computador, determinar a localização deste dentro do *file system*, desmontá-lo e executar o seguinte comando:

```
sudo dd if=2020-02-13-raspbian-buster-lite.img of=/dev/rdisk2 bs=1m
```

Onde *rdisk2* é o *rdiskX* determinado. Aguardar pelo término do processo, pois este não é verboso. Pode demorar cerca de 10 minutos.

3. Em seguida, inserir este mesmo cartão *micro SD* no *Raspberry Pi*, ligá-lo e deixar a instalação decorrer automaticamente.

Configurações

1. SSH

```
$ sudo raspi-config
```

Ativar acesso *SSH*

2. Alterar a *password* do utilizador *pi* através do comando `passwd`
3. Nova entrada na rede *Wi-Fi*

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=PT

network={
ssid="Cabovisao-4149"
```

```
psk="XXXXXXXXXX"
}
```

4. Atribuição de um endereço *IP* estático

```
interface wlan0
static ip_address=192.168.1.100
static routers=192.168.1.1
static domain_name_servers=8.8.8.8
```

5. Instalação de todas as dependências necessárias

```
# Update the system libraries
sudo apt-get update
sudo apt-get upgrade

# Vi improved (text editor)
sudo apt-get install vim

# Nginx
sudo apt-get install nginx

# Certbot
sudo apt-get install certbot python-certbot-nginx

# NodeJS & npm
curl -sL https://deb.nodesource.com/setup_10.x | sudo bash -
sudo apt-get install -y nodejs

# MySQL server
sudo apt-get install mariadb-server
sudo mysql_secure_installation

# PM2
sudo npm install -g pm2
```

6. Proceder com as restantes configurações

Anexo II - MySQL

Neste anexo consta a criação do utilizador na base de dados, assim como a definição das tabelas.

Configurações da instância MySQL

```
pi@amfsrpi:~ $ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.3.22-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE est_schedule;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'estschedule'@'%' IDENTIFIED BY 'estschedule.x';
Query OK, 0 rows affected (0.007 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON est_schedule.* TO 'estschedule'@'%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye
```

Aceder ao ficheiro de configuração por omissão através de:

```
$ sudo vi /etc/mysql/my.cnf
```

Adicionar a entrada `bind-address = 0.0.0.0` a este ficheiro. Isto permite acesso remoto à instância MySQL, se necessário.

Dump exemplo

```
$ mysqldump -u estschedule -p est_schedule --opt > est_schedule_2020-05-01.sql
```

Carregamento de ficheiro .sql para a base de dados

```
$ mysql -h amfsrpi -u estschedule -p est_schedule < est_schedule_2020-05-01.sql
```

DDL (Data Definition Language)

```
CREATE TABLE TOPOLOGIA (  
    topologia VARCHAR(20) NOT NULL,  
    descricao VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE UNIDADE_CURRICULAR (  
    id_uc INT AUTO_INCREMENT,  
    codigo_si VARCHAR(20) NOT NULL UNIQUE,  
    nome_uc VARCHAR(255),  
    PRIMARY KEY (id_uc)  
);  
  
CREATE TABLE CURSO (  
    id_curso INT AUTO_INCREMENT,  
    topologia VARCHAR(20) NOT NULL,  
    sigla_curso VARCHAR(20) NOT NULL UNIQUE,  
    nome_curso VARCHAR(255),  
    PRIMARY KEY (id_curso)  
);  
  
CREATE TABLE RAMO (  
    id_amo INT AUTO_INCREMENT,  
    sigla_amo VARCHAR(20) NOT NULL,  
    nome_amo VARCHAR(255),  
    id_curso INT NOT NULL,  
    PRIMARY KEY (id_amo),  
    FOREIGN KEY (id_curso)  
        REFERENCES CURSO (id_curso)  
        ON DELETE CASCADE  
);  
  
CREATE TABLE UNIDADE_CURRICULAR_TRONCO_COMUM (  
    id_uc INT NOT NULL,  
    id_curso INT NOT NULL,  
    ano INT NOT NULL,  
    periodo_letivo VARCHAR(20) NOT NULL,  
    PRIMARY KEY (id_uc, id_curso),  
    FOREIGN KEY (id_uc)  
        REFERENCES UNIDADE_CURRICULAR (id_uc)
```

```

        ON DELETE CASCADE,
    FOREIGN KEY (id_curso)
        REFERENCES CURSO (id_curso)
    ON DELETE CASCADE
);

CREATE TABLE UNIDADE_CURRICULAR_RAMO (
    id_uc INT NOT NULL,
    id_amo INT NOT NULL,
    ano INT NOT NULL,
    periodo_letivo VARCHAR(20) NOT NULL,
    PRIMARY KEY (id_uc, id_amo),
    FOREIGN KEY (id_uc)
        REFERENCES UNIDADE_CURRICULAR (id_uc)
    ON DELETE CASCADE,
    FOREIGN KEY (id_amo)
        REFERENCES RAMO (id_amo)
    ON DELETE CASCADE
);

CREATE TABLE PERFIS (
    id_perfil INT(2) NOT NULL,
    descricao VARCHAR(255) NOT NULL,
    PRIMARY KEY(id_perfil)
);

CREATE TABLE UTILIZADORES (
    id_utilizador INT(5) NOT NULL AUTO_INCREMENT,
    id_perfil INT(2) NOT NULL,
    username VARCHAR(30) NOT NULL,
    password VARCHAR(255) NOT NULL,
    primeiro_nome VARCHAR(20) NOT NULL,
    nome VARCHAR(255) NOT NULL,
    data_criacao DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    data_modificacao DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (id_utilizador),
    UNIQUE KEY username (username),
    FOREIGN KEY (id_perfil)
        REFERENCES PERFIS (id_perfil)
);

CREATE TABLE ANO_LETIVO (
    ano_letivo VARCHAR(9) NOT NULL,
    previsao INT NOT NULL DEFAULT 0,
    PRIMARY KEY (ano_letivo)
);

CREATE TABLE ESTATISTICAS_NUMERO_ALUNOS (
    ano_letivo VARCHAR(9) NOT NULL,

```

```

    sigla_uc VARCHAR(30) NOT NULL,
    sigla_curso VARCHAR(30) NOT NULL,
    n_alunos INT(5) NOT NULL,
    PRIMARY KEY (ano_letivo, sigla_uc, sigla_curso),
    FOREIGN KEY (ano_letivo)
        REFERENCES ANO_LETIVO (ano_letivo)
);

CREATE TABLE PREVISAOES (
    ID_PREVISA0 VARCHAR(255) NOT NULL,
    DATA_PREVISA0 BIGINT NOT NULL,
    UTILIZADOR VARCHAR(30) NOT NULL,
    SIGLA_CURSO VARCHAR(20) NOT NULL,
    NOME_CURSO VARCHAR(255) NOT NULL,
    ANO_LETIVO VARCHAR(9) NOT NULL,
    MODELO VARCHAR(255) NOT NULL,
    PARAMETROS TEXT NOT NULL,
    PREVISA0 TEXT NOT NULL,
    PRIMARY KEY (ID_PREVISA0)
);

```

Anexo III - Definições de rede

Neste anexo constam todas as definições de rede que foram necessárias implementar para poder comunicar com o *Raspberry Pi*, a partir de uma rede caseira local, sem expôr informação de uma maneira insegura para a *Internet*

Ativar *port forwarding* no router

Os seguintes redirecionamentos foram criados:

- Porto 22 : *SSH*
- Porto 3306: Base de dados *MySQL* (só para acesso externo para efeitos de desenvolvimento)
- Porto 80: Servidor *HTTP* (por sua vez é redirecionado para 443)
- Porto 443: Servidor *HTTPS*

Local			External			Protocol	Description	Enabled	Remove All	
IP Address	Start Port	End Port	IP Address	Start Port	End Port				Edit	Remove
192.168.1.100	22	22	0.0.0.0	22	22	BOTH	RPi SSH	Yes	Edit	Remove
192.168.1.100	3306	3306	0.0.0.0	3306	3306	BOTH	RPi MySQL DB	Yes	Edit	Remove
192.168.1.100	80	80	0.0.0.0	80	80	BOTH	RPi HTTP	Yes	Edit	Remove
192.168.1.100	443	443	0.0.0.0	443	443	BOTH	RPi HTTPS	Yes	Edit	Remove

Ativação do serviço *Dynamic DNS*

Uma vez que é uma rede caseira e que o endereço *IP* não é estático, foi necessário registar um domínio (amfs.ddns.net)

DDNS Service:

User Name:

Password:

Host Name:

IP Address: **217.129.48.168**

Anexo IV - Definições de SSL

Para servir conteúdo via *HTTPS*, é necessário configurar o servidor *nginx* com certificados gerados automaticamente, executando o seguinte comando uma única vez, após instalação do *nginx* e configuração:

```
sudo certbot -nginx
```

Isto irá gerar os certificados necessários, e atualizar a configuração do servidor *nginx*.

Quando for perguntado ao utilizador o seguinte:

```
No names were found in your configuration files. Please enter in your domain name(s) (comma and/or space separated) (Enter 'c' to cancel):
```

Escrever: `amfs.ddns.net`, ou qualquer outro host que tenha selecionado.

Anexo V - Definições de servidor nginx

Efetuar uma cópia do ficheiro `/etc/nginx/sites-available/default`, se alguma vez for necessário restaurar

```
cp /etc/nginx/sites-available/default ~/nginx_sites-available_backup/
```

Editar o ficheiro `sites-available`:

```
pi@amfsrpi:~ $ sudo vi /etc/nginx/sites-available/default
root /var/www/html;

location = / {
    return 301 /estschedule;
}

location /estschedule {
    alias /var/www/estschedule/pwa/;
}

location /estschedule/api {
    proxy_pass http://localhost:3000/api;
}
```

Testar a configuração

```
sudo nginx -t
```

Se não houve falhas no teste, fazer *reload* da configuração:

```
sudo systemctl reload nginx
```

Ficheiro de configuração completo:

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
```

```

# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration

# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
#server {
#    listen 80;
#    listen [::]:80;
#
#    server_name example.com;
#
#    root /var/www/example.com;
#    index index.html;
#
#    location / {
#        try_files $uri $uri/ =404;
#    }
#}

server {

    # SSL configuration
    #
    listen 443 ssl ;
    listen [::]:443 ssl ;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
    server_name amfs.ddns.net; # managed by Certbot

    location /estschedule {
        alias /var/www/estschedule/dist/spa/;
    }

    location /estschedule/api {
        proxy_pass http://localhost:3000/api;
    }
}

```

```

}

# pass PHP scripts to FastCGI server
#
#location ~ /\.php$ {
#    include snippets/fastcgi-php.conf;
#
#    # With php-fpm (or other unix sockets):
#    fastcgi_pass unix:/run/php/php7.3-fpm.sock;
#    # With php-cgi (or other tcp sockets):
#    fastcgi_pass 127.0.0.1:9000;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}

    ssl_certificate /etc/letsencrypt/live/amfs.ddns.net/fullchain.pem; # managed
by Certbot
    ssl_certificate_key /etc/letsencrypt/live/amfs.ddns.net/privkey.pem; # managed
by Certbot
}

server {
    if ($host = amfs.ddns.net) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 ;
    listen [::]:80 ;
    server_name amfs.ddns.net;
    return 404; # managed by Certbot
}

```



```
$ pm2 monitor
```

```
Make pm2 auto-boot at server restart:
```

```
$ pm2 startup
```

```
To go further checkout:
```

```
http://pm2.io/
```

```
-----
```

```
[PM2] Init System found: systemd
```

```
Platform systemd
```

```
Template
```

```
[Unit]
```

```
Description=PM2 process manager
```

```
Documentation=https://pm2.keymetrics.io/
```

```
After=network.target
```

```
[Service]
```

```
Type=forking
```

```
User=pi
```

```
LimitNOFILE=infinity
```

```
LimitNPROC=infinity
```

```
LimitCORE=infinity
```

```
Environment=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/games:/usr/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
```

```
Environment=PM2_HOME=/home/pi/.pm2
```

```
PIDFile=/home/pi/.pm2/pm2.pid
```

```
Restart=on-failure
```

```
ExecStart=/usr/lib/node_modules/pm2/bin/pm2 resurrect
```

```
ExecReload=/usr/lib/node_modules/pm2/bin/pm2 reload all
```

```
ExecStop=/usr/lib/node_modules/pm2/bin/pm2 kill
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
Target path
```

```
/etc/systemd/system/pm2-pi.service
```

```
Command list
```

```
[ 'systemctl enable pm2-pi' ]
```

```
[PM2] Writing init configuration in /etc/systemd/system/pm2-pi.service
```

```
[PM2] Making script booting at startup...
```

```
[PM2] [-] Executing: systemctl enable pm2-pi...
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-pi.service →
```

```
/etc/systemd/system/pm2-pi.service.
```

```
[PM2] [v] Command successfully executed.
```

```
+-----+
```

```
[PM2] Freeze a process list on reboot via:
```

```
$ pm2 save
```

```
[PM2] Remove init script via:
```

```
$ pm2 unstartup systemd
```

```
pi@amfsrpi:/opt/estschedule-service $ pm2 save
```

```
[PM2] Saving current process list...
```

```
[PM2] Successfully saved in /home/pi/.pm2/dump.pm2
```

```
pi@amfsrpi:/opt/estschedule-service $
```

Deployment (*web-app*)

Fazer um *build* da *web-app* na máquina de desenvolvimento

```
amfs@Andres-MacBook-Air ~/workspace/faculdade/estschedule-webapp (master) $ quasar  
build -m pwa
```

Enviar os ficheiros para o *Raspberry Pi*

```
amfs@Andres-MacBook-Air ~/workspace/faculdade/estschedule-webapp (master) $ scp -r  
dist pi@amfs.ddns.net:tmp
```

```
pi@amfsrpi:/var/www $ sudo mkdir estschedule
```

Mover esta pasta para /var/www/estschedule

```
pi@amfsrpi:~ $ sudo mv tmp/dist/ /var/www/estschedule/
```

Após este passo, tudo estará instalado na máquina