

Universidade de Aveiro



Learning-electronics: *A new way of learning*

André Pragosa Clérigo

Carlos Rafael de Jesus Palma Costa

Cláudio António Felgueiras Asensio

Edgar Antunes Abreu de Sousa

Hugo Miguel Ventura Domingos

Tiago Afonso Marques

Departamento de Eletrónica, Telecomunicações e Informática

Licenciatura em Engenharia de Computadores e Informática

Regente: Professor José Moreira

12 de janeiro, 2021

Resumo

Este projeto tem como objetivo a criação de uma plataforma adaptável e lúdica para a área da eletrônica, tanto para docentes como alunos. Baseados na nossa própria experiência com métodos de ensino online e na dos colegas do nosso departamento, pretendemos contribuir para a transição da aprendizagem dos circuitos elétricos de algo teórico a algo interativo.

Para atingir este objetivo deveremos ter em mente três componentes principais no projeto: como adaptar um tema tão extenso e complexo e criar um sistema modular e adaptável, ou seja, como organizar a informação; como implementar um sistema logístico para organização que consiga corresponder às necessidades dos vários alunos e docentes; finalmente como implementar as componentes lúdicas do projeto que, em simultâneo permitam dar formar e avaliar os alunos. Em suma os três pilares deste projeto são Informação, Organização e Diversão.

Siglas e Acrónimos

API	Application Programming Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input/Output
JSON	JavaScript Object Notation
MNA	Modified Nodal Analysis
SIACUA	Sistema Interativo de Aprendizagem por Computador – Universidade de Aveiro
SPICE	Simulation Program with Integrated Circuit Emphasis
UA	Universidade de Aveiro
UC	Unidade Curricular
UI	User Interface
URL	Uniform Resource Locator
XML	Extensible Markup Language

Glossário

Back-end

Back-end é entendido como a camada de acesso aos dados da aplicação, normalmente a parte do servidor num sistema web é considerado *back-end*.

Client-side

Client-side é a parte do código ou *software* que é executado na parte do usuário, ou seja, no seu computador.

Cross-site Request Forgery

Cross-site Request Forgery é a exploração maliciosa de um website onde comandos não autorizados são enviados através de um utilizador em que o serviço/aplicação confia.

Cross-site Scripting

Cross-site scripting é um tipo de vulnerabilidade de sites por meio do qual um atacante é capaz de inserir *scripts* maliciosos em páginas e aplicativos que seriam confiáveis e usá-los para instalar *software* malicioso nos navegadores dos usuários.

Framework

Na área da programação, uma *framework* é uma abstração na qual o *software* fornece uma funcionalidade genérica, pode ser alterado seletivamente por código adicional escrito pelo utilizador, fornecendo assim *software* específico para a aplicação. Uma *framework* fornece uma maneira padrão de construir aplicações num ambiente de *software* reutilizável que fornece uma funcionalidade específica como parte de uma plataforma de *software* maior para facilitar o desenvolvimento de produtos e soluções.

Front-end

Na área do desenvolvimento *web*, *Front-end* é o desenvolvimento da interface gráfica exibida ao utilizador num *website* usando elementos *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e *JavaScript*, para que os utilizadores consigam ver e interagir com o *website*.

Function callback

Function callback é quando é passada uma função como parâmetro para outra função, para ser invocada dentro dela.

Open source

Open source é o código-fonte disponibilizado gratuitamente para possível modificação e redistribuição. Os produtos incluem permissão para usar o código-fonte, documentos de design ou o conteúdo do produto.

Package manager

Package manager é um conjunto de ferramentas que trata de automatizar o processo de instalação, atualização, configuração e remoção de programas do sistema de forma eficiente.

Runtime environment

Runtime environment é o ambiente em que um dado programa ou aplicação é executado, ou seja, é a infraestrutura de *hardware* e *software* que suporta a execução do código em tempo real.

SQL Injection

Injeção de SQL é um tipo de ataque cibernético em que o atacante insere um código da sua autoria em um site de forma a aproveitar falhas em sistemas que interagem com bases de dados através de comandos SQL para violar as medidas de segurança e aceder a dados protegidos.

Strong typing

Strong typing geralmente refere-se ao uso de tipos de linguagem de programação que impõe restrições estritas na mistura de valores com diferentes tipos de dados. Quando tais restrições são violadas e ocorre um erro (exceção).

Superconjunto

Na matemática, o conjunto A é um subconjunto de um conjunto B se todos os elementos de A também forem elementos de B. B é então um superconjunto de A. É possível que A e B sejam iguais.

TypesScript

TypeScript é uma linguagem de programação desenvolvida e mantida pela Microsoft. É um superconjunto estrito de JavaScript e adiciona *strong typing* opcional à linguagem. TypeScript é usado para o desenvolvimento de grandes aplicativos e compilar o seu código para o equivalente em JavaScript. Como o TypeScript é um superconjunto de JavaScript, os programas JavaScript existentes também são programas TypeScript válidos.

Índice

Capítulo 1	Introdução.....	1
1.1	Contexto.....	1
1.2	Motivação.....	2
1.3	Objetivos.....	2
1.4	Estrutura do documento.....	3
Capítulo 2	Trabalho Relacionado.....	5
2.1	CircuitSolver.....	5
2.2	Sistemas Web e Aplicações Móveis.....	7
Capítulo 3	Estado da Arte.....	11
3.1	Tecnologias.....	11
3.1.1	Angular.....	12
3.1.2	Django.....	13
3.1.3	REST API.....	14
3.1.4	Vue.js.....	15
3.1.5	React.....	16
3.1.6	Node.js.....	17
Capítulo 4	Requisitos do Sistema e Arquitetura.....	19
4.1	Requisitos do Sistema.....	19
4.1.1	Elicitação de Requisitos.....	19
4.1.2	Descrição do Contexto.....	20
4.1.3	Atores.....	21
4.1.4	Casos de Uso.....	22
4.1.5	Requisitos não funcionais.....	23
4.1.6	Suposições e Dependências.....	24
4.2	Arquitetura do Sistema.....	25
4.2.1	Modelo de Domínio.....	25
4.2.2	Modelo Tecnológico.....	26
Capítulo 5	Conclusão.....	29
Capítulo 6	Referências.....	31

Lista de Figuras

Figure 1 - Progresso na aprendizagem de uma língua estrangeira	8
Figure 2 - Página do curso Cálculo-I no SIACUA	8
Figure 3 - Jogo 1vs1 sobre categoria Ciência no Trivia Crack	9
Figure 4 - Resolução passo-a-passo do exercício matemático	10
Figure 5 - Two-way data binding model do Angular	12
Figure 6 - Arquitetura model-viewmodel-model	15
Figure 7 - Utilização do DOM Virtual para o aumento de desempenho	16
Figure 8 - Apresentação do model caso de usos para ambos os autores	22
Figure 9 - Modelo de domínio	25
Figure 10 - Modelo tecnológico entre user-servidor	26
Figure 11 - Modelo tecnológico dentro do servidor	27

Capítulo 1

Introdução

Atualmente a evolução dos métodos de ensino tende cada vez mais para o apelativo e lúdico, este género de mudança afeta todas as áreas ligadas ao ensino, mas será principalmente notável nas áreas das engenharias, onde grande parte dos alunos se encontram desmotivados e desinteressados com os conteúdos programados relativos à área das matemáticas e das físicas devido aos métodos de ensino pouco interativos e de carácter teórico elevado. Na nossa área, engenharia de computadores, o mesmo acaba por acontecer no domínio da eletrónica onde a resolução, observação e análise de circuitos elétricos pode desmotivar quem ainda não é familiar com a área. É com base nessa reestruturação deste desatualizado estilo de aprendizagem que o nosso projeto surge, de forma a proporcionar uma forma de aprendizagem mais interativa e imersiva fazendo uso das tecnologias atuais.

O nosso projeto consiste numa plataforma web que pode ser acedida através do web browser e onde os utilizadores poderão encontrar exercícios sobre a temática da área da eletrónica, exercícios estes que estarão disponíveis em vários formatos para que a aprendizagem consiga ser fácil e eficaz, para isto, recorreremos a vários modelos de ensino interativo.

1.1 Contexto

Este é um projeto inserido na unidade curricular (UC) de Projeto em Engenharia de Computadores e Informática, o projeto tem como objetivo a criação de uma plataforma *web* interativa e didática na aprendizagem de UCs na área dos circuitos eletrónicos, como por exemplo a plataforma SIACUA que a Universidade de Aveiro (UA) disponibiliza com o mesmo propósito do nosso projeto, mas aplicado na área da matemática. Algo que consiga complementar outras fontes de informação e conteúdo programático, como é o caso do eLearning que é usado na UA.

1.2 Motivação

Face às dificuldades sentidas pelos alunos na cadeira de Análise de Circuitos, foi idealizado por ex-alunos apoiados pelo docente da cadeira uma aplicação *web* que pretendemos finalizar e que tem como objetivos:

- **Tornar o ensino mais apelativo:** Usando métodos mais interativos, por exemplo com a apresentação de exercícios na forma de jogos ou *quizzes*, para tornar o tema mais fácil e atrativo de entender, para que seja fomentado o uso desta aplicação dentro e fora do contexto de sala de aula.
- **Ajudar tanto alunos como docentes:** Utilizando as tecnologias de que hoje em dia dispomos, propomo-nos a desenvolver um ambiente de trabalho mais dinâmico em que alunos e docentes podem com mais facilidade trocar informações, consultar/submeter exercícios sobre os vários temas da unidade curricular e realizarem-se testes modelo na plataforma.

1.3 Objetivos

O objetivo da criação desta plataforma *web* é ajudar todos os alunos que queiram aprender e aumentar os seus conhecimentos nas áreas da eletrónica e dos circuitos elétricos, conseguindo captar a sua atenção com a realização de testes, jogos e desafios para que o processo de aprendizagem seja mais fluido e divertido. Simultaneamente também temos como missão dar mais possibilidades e facilidade aos docentes destas UCs. Fornecendo diversas formas de inserção de exercícios na plataforma, facilitando assim a migração do papel para a *web*. Proporcionar ao docente uma visão geral da performance de um dos seus alunos em específico ou de uma das suas turmas através da implementação de um sistema de estatísticas.

1.4 Estrutura do documento

Para além da introdução este documento contém mais 5 capítulos. No Capítulo 2 são apresentados os trabalhos relacionados com este projeto, tanto em relação direta, como por exemplo trabalhos guiados pelo nosso coorientador. Como também *websites*/aplicações nas quais nos vamos inspirar para desenvolver esta plataforma. No Capítulo 3 é descrito o estado da arte, isto é, as tecnologias e ferramentas que vamos utilizar para poder construir a plataforma,

O Capítulo 4 descreve os requisitos e arquitetura do nosso sistema, são nesses pontos que descrevemos de modo mais detalhado quais as funcionalidades que o nosso sistema vai oferecer aos nossos atores. É também descrito e explicado a estrutura da nossa arquitetura num contexto do modelo tecnológico. apresentamos uma pequena conclusão e visão global do projeto em que estamos a trabalhar no Capítulo 5.

Finalmente vamos deixar todas as referências relativas a este documento no Capítulo 6.

Capítulo 2

Trabalho Relacionado

Este capítulo tem como objetivo apresentar o trabalho relacionado com o nosso projeto, apresentando as nossas fontes de inspiração e ideias retiradas.

Infelizmente existe pouco, ou quase nenhum, trabalho diretamente relacionado/comparado com o nosso projeto e é essa falha no mercado que incentivou a criação deste projeto. Nesta secção iremos focar-nos principalmente em duas áreas: no módulo de CircuitSolver, que é uma ferramenta de resolução de circuitos eletrónicos e em *websites* de aprendizagem lúdicos e em *websites* nos quais nos vamos inspirar e retirar certas características desejadas de cada componente.

2.1 CircuitSolver

Esta ferramenta disponibiliza funções capazes de, dado um certo circuito eletrónico gerar uma solução (resolver o circuito em si), uma explicação de como resolver o mesmo e gerando ainda três valores de respostas incorretos. O CircuitSolver usa *Simulation Program with Integrated Circuit Emphasis* (SPICE), que nos permite realizar simulações de circuitos relativamente a baixo custo.

Programas de simulação de circuitos dentro dos quais se encontra SPICE e os seus derivados, necessitam de uma lista que consiste, numa forma mais geral, dos componentes eletrónicos do circuito a ser testado e a que nós esses tais componentes se encontrariam conectados. A estrutura e complexidade desta listagem pode variar consideravelmente, mas tem como principal objetivo transmitir características dos componentes e informações de conectividade entre os componentes e nós. Ao ser passada a lista ao SPICE este traduz esse ficheiro num sistema de equações a ser resolvido. Este conjunto de equações é resolvido posteriormente fazendo uso da *Modified Nodal Analysis* (MNA). Na utilização da MNA geralmente obtemos um

sistema de equações maiores em comparação aos métodos tradicionais que pessoas usam para resolver problemas, no entanto este sistema que aparenta ser mais complexo para um ser humano é extremamente mais fácil de implementar em computador, o que nos oferece uma vantagem considerável na implementação de uma solução automatizada.

Para aplicação desta técnica é escrita uma equação para cada nó não conectado a uma fonte de tensão, igual à análise nodal, e uma equação para cada fonte de tensão. Este sistema de equações é analisado e resolvido, conseguindo conseqüentemente chegar à sua solução e explicação.

Ao usar esta ferramenta vemo-nos limitados a perguntas de escolha múltipla que tenham como resposta valores matemáticos, não sendo possível a resolução de perguntas de interpretação, como por exemplo a descrição da resposta temporal de um circuito de corrente alternada.

Inicialmente este módulo não era do nosso conhecimento e ficamos a saber da existência do mesmo a partir do nosso coordenador Prof. João Nuno Matos. Embora não esteja completo na sua totalidade e faltando algum refinamento, é utilizável, não atingindo o seu máximo potencial. A solução irá passar pela colaboração total ou parcial na finalização deste módulo.

Existem vários sites de resolução de equações matemáticas, que nos fornecem com resoluções passo a passo, no entanto, não existem grandes opções para uma ferramenta que realize o mesmo no domínio da eletrónica. O destaque desta tecnologia torna-se óbvia quando apresentados com esta realidade.

Outra razão para o destaque deste sistema é a facilidade por parte dos docentes de inserir exercícios, sendo apenas pedido ao professor a adição da descrição do circuito e a pergunta relativa ao mesmo, fazendo o módulo do CircuitSolver o resto do trabalho.

2.2 Sistemas Web e Aplicações Móveis

As fontes principais nas quais nos inspiramos são os seguintes:

- Google Classroom
- Duolingo
- Khan Academy
- SIACUA
- Trivia Crack
- Symbolab

O nosso projeto é baseado no aprimoramento de experiências de aprendizagem na área da eletrónica e na criação de um ambiente competitivo e divertido.

Para tal, do Google Classroom [4] destacamos os sistemas de organização com turmas/aulas, conceito de professor e de aluno sendo este *website* o mais comparável com a nossa idealização em termos de aprendizagem. Um conceito que por si é básico, ou seja, a criação de um grupo de utilizadores, sendo que neste caso há uma especificação nos utilizadores, tendo estes papéis diferentes. Como exemplo podemos dar o papel de um professor que será introduzir conteúdo na turma, sendo que o aluno poderá fazer perguntas sobre o mesmo, mas não irá obviamente disponibilizar exercícios para os outros resolverem.

Na famosa aplicação para a aprendizagem de línguas estrangeiras Duolingo [2], notamos que o sistema de análise da progressão individual presente na Figure 1, e os sistemas de aumento de dificuldade. Esta implementação é algo que o nosso sistema terá que implementar, e por isso, é bom candidato para inspiração futura.

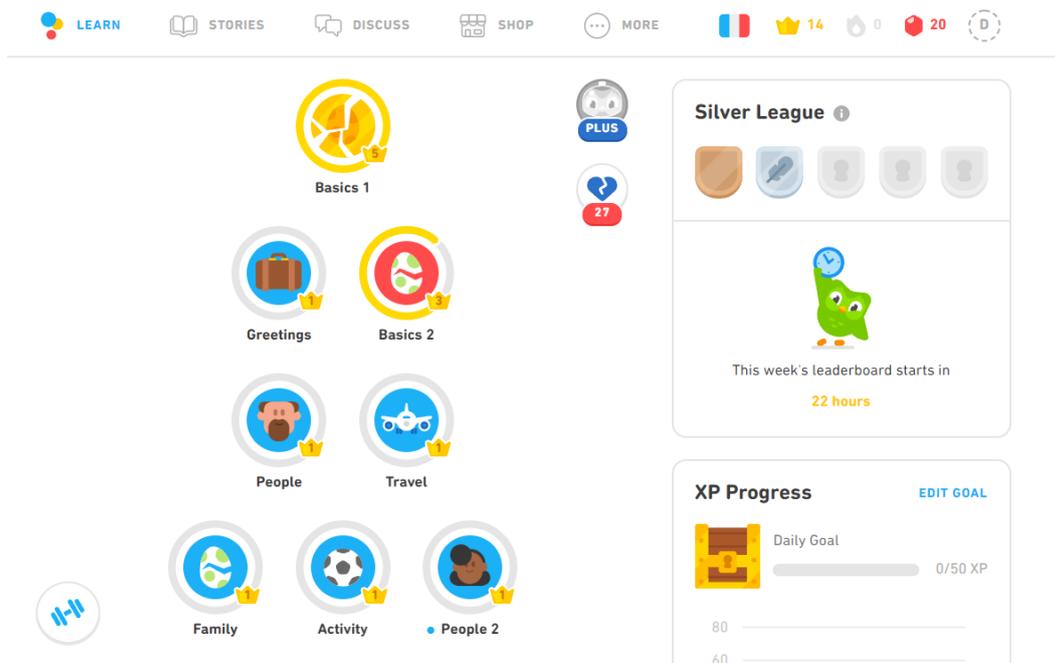


Figure 1 - Progresso na aprendizagem de uma língua estrangeira

Ao analisar o site SIACUA [7] Figure 2, disponibilizado pela UA, reparamos na existência de um módulo de parametrização de exercícios. Módulo este que pretendemos implementar através da utilização de trabalho desenvolvido anteriormente por antigos alunos chamado CircuitSolver, que na inserção de um exercício em formato SPICE, este é capaz de gerar variações desse exercício inserido. Outro aspeto interessante desse site é a disposição de um sistema de estatísticas do desempenho dos alunos ao realizar exercícios, algo que seria uma mais-valia na nossa plataforma e tornaria a vida dos nossos docentes mais facilitada.

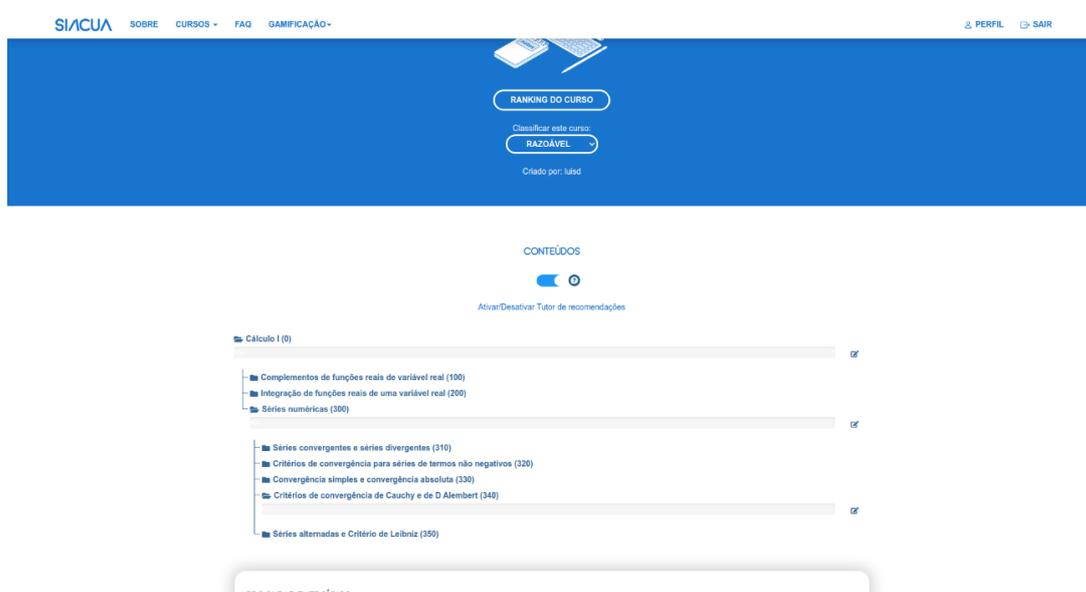


Figure 2 - Página do curso Cálculo-I no SIACUA

Do website Khan Academy [6] que é uma plataforma de aprendizagem académica que abrange tópicos como matemática e física, de um nível básico ao universitário, utilizando vídeos intercalados com quizzes e testes. Deste website notamos os aspetos de organização e desconstrução de tópicos complexos em segmentos digeríveis. Juntamente com alguns aspetos de seguimento da progressão no trajeto de aprendizagem.

O Trivia Crack (ou mais conhecido por Perguntados em Portugal), notamos os elementos lúdicos e competitivos Figure 3, funcionalidade esta que não é pioneira e também está disponível noutras aplicações móveis tal como o Kahoot. Este tipo de implementação lúdica na resposta de perguntas poderá, mas não será exclusivamente, uma das soluções que a nossa plataforma poderá ter.

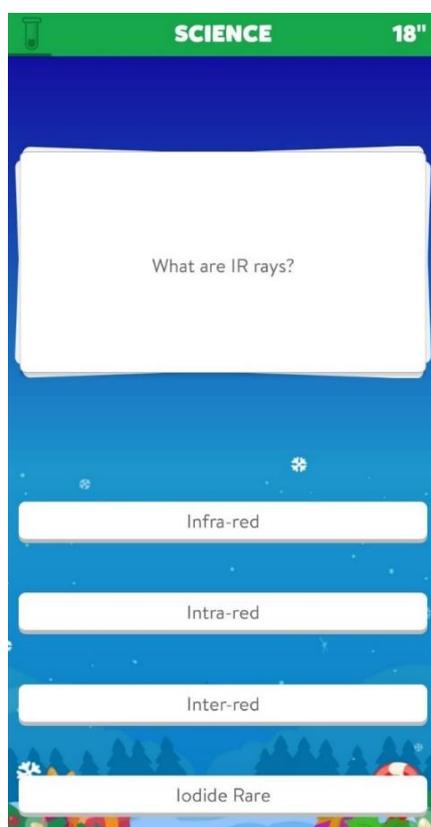


Figure 3 - Jogo 1vs1 sobre categoria Ciência no Trivia Crack

Finalmente, no que toca à resolução de exercícios olhamos obviamente para um website bastante conhecido, Symbolab [8], este sistema implementa a resolução de equações e outros tópicos relacionados com a área da matemática, um exemplo será, a introdução de uma equação ao qual nos permite visualizar o resultado, assim como visualizar uma resolução detalhada passo a passo do problema. A Figure 4 mostra um exemplo que segue o modelo do que pretendemos implementar para o nosso projeto, embora adaptado à área da eletrónica.

$x^2 - x - 6 = 0$ Go

Graph » Examples » Share Print

Solution Keep Practicing >

Solve with the quadratic formula Show Steps

$x^2 - x - 6 = 0$: $x = 3, x = -2$

Steps

$x^2 - x - 6 = 0$

Solve with the quadratic formula Hide Steps

Quadratic Equation Formula: Hide Definition

For a quadratic equation of the form $ax^2 + bx + c = 0$ the solutions are

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

For $a = 1, b = -1, c = -6$

$$x_{1,2} = \frac{-(-1) \pm \sqrt{(-1)^2 - 4 \cdot 1 \cdot (-6)}}{2 \cdot 1}$$

$$x_{1,2} = \frac{-(-1) \pm \sqrt{(-1)^2 - 4 \cdot 1 \cdot (-6)}}{2 \cdot 1}$$

Show Steps

$$\sqrt{(-1)^2 - 4 \cdot 1 \cdot (-6)} = 5$$

$$x_{1,2} = \frac{-(-1) \pm 5}{2 \cdot 1}$$

Separate the solutions

$$x_1 = \frac{-(-1) + 5}{2 \cdot 1}, x_2 = \frac{-(-1) - 5}{2 \cdot 1}$$

Show Steps

$$x = \frac{-(-1) + 5}{2 \cdot 1}; 3$$

Show Steps

$$x = \frac{-(-1) - 5}{2 \cdot 1}; -2$$

The solutions to the quadratic equation are:

$x = 3, x = -2$

Figure 4 - Resolução passo-a-passo do exercício matemático

Capítulo 3

Estado da Arte

O objetivo deste projeto inclui, mas não se restringe, à criação de uma plataforma web para a aprendizagem de circuitos eletrônicos, visto que o objetivo é construir algo robusto e com capacidade para aguentar uma grande capacidade de fluxo, de forma eficiente e segura.

Esta secção do relatório tem como intuito apresentar as tecnologias que nós entendemos serem as mais relevantes para este projeto, fazendo uma descrição do propósito das nossas escolhas e das principais características de cada uma das ferramentas referidas. Irão também ser enumeradas as principais vantagens e desvantagens que iremos ter em consideração durante o processo de seleção.

As seguintes tecnologias também foram destacadas porque com a combinação das mesmas poderá formar um *standard* empresarial [1] [5] usado no mercado de trabalho para criar uma aplicação robusta e funcional, em vez de ir por caminhos mais “fáceis” acabando por cair no domínio do *freelance* ou dos projetos pessoais, nomeadamente, *templates* pré feitos em HTML, CSS e JavaScript “puros”.

3.1 Tecnologias

Esta secção irá apresentar as tecnologias que são válidas para a realização deste projeto, fazendo uma breve descrição das suas características. Iremos descrever as vantagens e desvantagens que estas trazem consigo.

3.1.1 Angular

Angular é uma *framework* criada e desenvolvida pela Google [3] desde 2009, focada em *front-end* e baseada em TypeScript, rápida e leve, direcionada para o desenvolvimento de aplicações web, sendo uma das tecnologias mais utilizadas para o desenvolvimento de este tipo de aplicações.

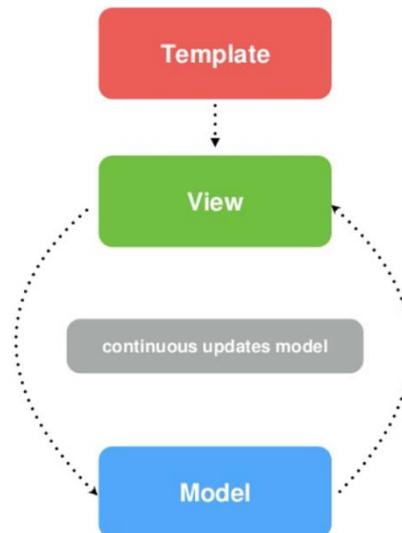


Figure 5 - Two-way data binding model do Angular

Escolhemos usar esta tecnologia porque nos oferece funcionalidades tais como a programação em componentes que nos permitem, por exemplo, usar pop-ups como componentes na parte interativa da nossa aplicação e *two-way data binding*, isto é, permite a automatização na circularização de dados uma vez que quando o valor de uma componente muda o *template* faz automaticamente uma atualização da página, não sendo preciso estarmos a escrever o mesmo código HTML e JavaScript sempre que se atualizam os valores das componentes Figure 5. Os componentes também são extremamente vantajosos para um grupo de 6 programadores porque dão-nos a capacidade de dividir tarefas e aumentar a escalabilidade da nossa aplicação. Deste modo, cada equipa ou indivíduo pode criar e desenvolver um componente de forma independente e no fim da criação do mesmo apenas temos que o integrar com a nossa aplicação. Já que este componente tem uma integração modular com o nosso sistema, este também nos permite a reutilização de código, tornando assim a nossa aplicação mais fácil de alterar já que o código referente a um componente utilizado várias vezes só é escrito uma vez. Esta modulação também nos permite uma grande escalabilidade, já que a nossa aplicação é efetivamente constituída por blocos e não por código corrido.

Outro ponto forte no uso de Angular é facto desta *framework* ser uma das mais populares na sua área, fazendo com que a mesma tenha mais fácil de aprender já que existe imenso conteúdo de qualidade disponibilizado online.

Por outro lado, esta tecnologia tem algumas desvantagens, como é o caso de não permitir grande otimização para os motores de busca, por ser muito verbosa e de tal modo robusta que pode acabar por se tornar menos eficaz para um trabalho específico como o nosso, existem outras tecnologias que são mais direccionadas para o *front-end*, tais como React ou Vue, que poderiam revelar-se mais eficazes no processo de aprendizagem.

3.1.2 Django

Django é uma *web framework open source* que pretende acelerar o processo de desenvolvimento e manutenção de uma aplicação web através do uso da linguagem Python. O código em Django é escrito usando princípios e padrões de design que encorajam a criação de código reusável e de fácil manutenção. Sendo Django usado há mais de uma década, este contém várias bibliotecas que permitem implementar as mais diversas funcionalidades como APIs, autenticação do utilizador entre outras sem a ocorrência de muitos problemas.

Esta *framework* é versátil, usada para construir quase todos os tipos de websites, desde *wikis* e sistemas de gestão até redes de comunicação social. Sendo capaz de comunicar com qualquer *client-side framework*, este está habilitado a fornecer conteúdo nos mais variados formatos (HTML, JSON (*JavaScript Object Notation*), XML (*Extensible Markup Language*), etc.).

Ao proporcionar tamanha versatilidade é completamente normal que este proporcione outros benefícios entre os quais a segurança de dados. Django ajuda os programadores a evitar erros de segurança mais comuns, como por exemplo, invés de guardar as palavras-passe em texto o Django por norma guarda as palavras-passe de forma encriptada, para além da prevenção destes erros por parte do programador também é possível a proteção contra várias vulnerabilidades conhecidas como *SQL Injection*, *cross-site scripting* e muitas outras.

Sendo o código em Django escrito em Python, que corre em várias plataformas, não nos é limitada a escolha de plataforma para o servidor. Além disso, Django é suportado por muitos fornecedores de serviços de hospedagem web, o que nos oferece grande portabilidade.

Outros benefícios que nos levaram a escolha desta tecnologia é o uso de Python para a criação e manutenção de base de dados sem necessitar de usar SQL, assim como a criação de páginas

dinâmicas utilizando modelos que facilitam assim a apresentação dos dados da página do utilizador, por exemplo.

Todavia sendo Python uma das linguagens que não é muito rápida em comparação com as outras disponíveis, um sacrifício é feito na rapidez da aplicação a ser desenvolvida, pelo que é necessário assegurar uma estrutura bem organizada e otimizada para evitar problemas de responsividade. Outro lado negativo de Django é o excessivo encapsulamento, isto é, tudo tem que ser definido explicitamente, o que leva a padrões de configuração que podem retardar o processo de desenvolvimento.

3.1.3 REST API

Uma Rest API é um estilo de arquitetura de *software* que define o que está em conformidade com um conjunto de restrições direcionadas para criação de serviços web e que permite a interação com servidores web.

Este tipo de serviço permite que os sistemas que solicitem informação ao servidor, neste caso a nossa aplicação em Django, possam aceder e manipular recursos em formato de texto, tais como HTML ou JSON. Isto usando um conjunto predefinido de operações *stateless*, ou seja, operações de comunicação baseadas em pares de requisição e resposta independentes uns dos outros, o que nos permite fazer várias requisições ao servidor sem este precisar de reter informação sobre cada interlocutor.

A nossa aplicação web irá possuir uma Rest API implementada com o propósito de fazer a comunicação entre o servidor Django e a aplicação em Angular via HTTP (*Hypertext Transfer Protocol*) o que permite que os módulos de Angular facilmente se alterem e façam *queries* à bases de dados no servidor ao mesmo tempo conseguimos um bom nível de desempenho e confiabilidade neste tipo de processos.

3.1.4 Vue.js

O Vue.js é uma *framework open source* baseada em TypeScript direcionada para o desenvolvimento *front-end*. Foi criada pelo Sr. Evan You [9] e ainda é mantida pelo mesmo e o resto da sua equipa.

Esta tecnologia é baseada na arquitetura *model-viewmodel-model*, que à semelhança de Angular, também oferece o desenvolvimento de páginas únicas, isto é, a criação de uma página única dinâmica que não depende do caminho dado na barra de URL (Uniform Resource Locator).

A arquitetura que este oferece e o separa das outras tecnologias existentes, consiste na separação do desenvolvimento da interface gráfica do usuário (o *view*) do desenvolvimento da lógica de *back-end* (o *model*) para que a visualização não dependa de nenhuma plataforma específica. O *model-viewmodel-model* é responsável por converter os objetos de dados do modelo de forma que os objetos sejam facilmente apresentados. Para além disto o Vue.js permite estender HTML com atributos HTML chamados *diretivas*. As *diretivas* oferecem funcionalidade para aplicações HTML e vêm como *diretivas embutidas* ou *definidas pelo usuário*.

No entanto, a biblioteca é maioritariamente focada na visualização de componentes gráficos, sendo que recursos avançados, tais como o roteamento, são tratados por bibliotecas e *packages* de suporte oficialmente suportados pela plataforma.

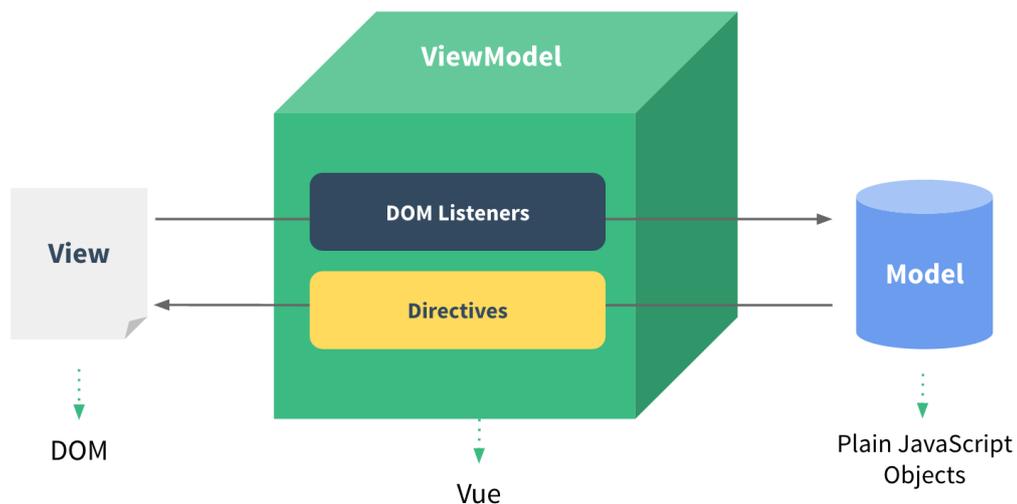


Figure 6 - Arquitetura *model-viewmodel-model*

3.1.5 React

O React é uma biblioteca JavaScript *open source* focada no desenvolvimento *front-end*, foi inicialmente criada por um empregado da Facebook e, atualmente, é mantida pela Meta (Ex conhecida por Facebook) e uma comunidade de *developers* e empresas.

Tendo algumas semelhanças com Angular, React é baseado na criação de componentes UI (*User Interface*), focando-se no modelo página única.

Uma das funcionalidades revolucionadoras que esta ferramenta oferece ao mercado é a disponibilização de um DOM (*Document Object Model*) virtual apresentado na Figure 7. O React cria uma cache da estrutura de dados (DOM) na memória, calcula as diferenças resultantes e, em seguida, atualiza o DOM exibido do navegador com eficiência. Este processo é chamado de reconciliação que permite o programador escrever código como se a página inteira fosse gerada em cada alteração, enquanto as bibliotecas React apenas geram subcomponentes que realmente mudam. Essa criação seletiva fornece um grande aumento de desempenho porque não há o esforço de recalculando o estilo CSS e o *layout* da página.

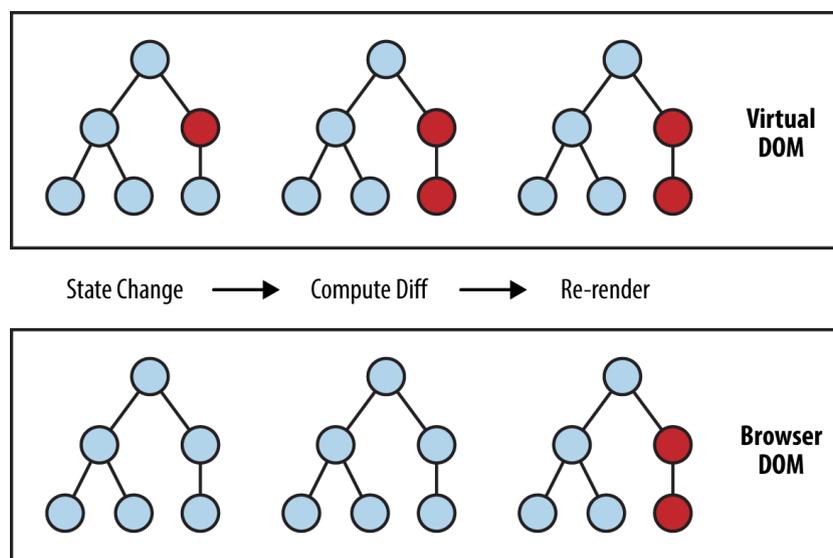


Figure 7 - Utilização do DOM Virtual para o aumento de desempenho

Por outro lado, esta não passa de uma biblioteca JavaScript que gere o estado de geração do DOM, logo é necessário o auxílio *software* adicional para a criação de um serviço *back-end* e até algumas funcionalidades mais específicas para o utilizador. Uma das alternativas, para diminuir a complexidade de mais uma linguagem nova seria o uso do Node.js para a complementaridade destas duas tecnologias.

3.1.6 Node.js

Node.js é um *software open source* e inicialmente desenvolvido por Ryan Dahl [10], esta ferramenta era gerida pela Node.js Foundation e atualmente faz parte da OpenJS Foundation [11], fundação que tem como objetivo promover o crescimento e a manutenção do JavaScript no ecossistema web.

Ao contrário das tecnologias baseadas em JavaScript apresentadas acima, o Node.js é uma ferramenta que cria um *runtime environment* para a criação de aplicações *back-end*, correndo código JavaScript que não provém de um web browser. Este sistema tem como intuito unificar o mundo web através do uso de JavaScript, isto significa que para criar um ambiente web, o que envolve *front-end* e *back-end*, só é necessário o uso de JavaScript, ao contrário do que é atualmente feito que é usar linguagem diferentes conforme estamos a trabalhar no *front-end* ou no *back-end*.

Este sistema é baseado em eventos assíncronos de I/O (*Input/Output*), correndo o código maioritariamente em *callbacks*. Esta abordagem permite que o servidor fique não bloqueante, ficando disponível a outros processos, isto oferece ao utilizador uma melhor usabilidade e capacidade de resposta. Para além disto o Node.js faz uso de um *package manager* bastante popular chamado npm, uma ferramenta que cada vez é mais popular e versátil.

Por outro lado, criar serviços de Node.js também tem desvantagens, devido à natureza do mesmo, escalabilidade não é o seu forte, sendo que em aplicações de baixa escala (tipicamente que apenas necessitam de um processador). A plataforma não oferece também capacidade de expansão tirando proveito de múltiplos *cores* que cada vez estão mais presentes no *hardware* direcionado a servidores.

Capítulo 4

Requisitos do Sistema e Arquitetura

4.1 Requisitos do Sistema

Esta secção irá apresentar os requisitos específicos do nosso sistema. Nos tópicos seguintes iremos descrever a Elicitação de requisitos, descrição do contexto, especificação dos atores, diagramas de casos de uso, requisitos não funcionais e as suposições e dependências do nosso sistema.

4.1.1 Elicitação de Requisitos

Para este segmento de relatório procuramos a assistência do nosso professor coorientador Nuno Matos, que para além de estar consciente do desenvolvimento do nosso trabalho também leciona várias cadeiras na área da eletrónica na UA. Juntamente com alguns dos nossos colegas na área de eletrónica inquirindo sobre as capacidade e sistemas desejados numa plataforma de aprendizagem. As funcionalidades desejadas para este sistema são as seguintes:

- Um sistema de organização dos exercícios por tema e dificuldade para podermos identificar dificuldades dos utilizadores e criar um crescimento gradual na dificuldade dos exercícios. Para algumas questões também seria disponibilizada uma resolução passo-a-passo para assistir no processo de aprendizagem daquele exercício-tipo.
- Um sistema que gera exercícios automaticamente para serem resolvidos pelo aluno com bases em ficheiros SPICE usando o módulo CircuitSolver. Juntamente com um sistema de upload através dos docentes para o aumento do banco de questões.
- Um jogo lúdico semelhante ao jogo da glória em que alunos competissem entre si para completar uma rota através de um sistema de posicionamento onde jogadores avançam

mais rapidamente se responderem acertadamente as questões geradas pelo sistema anteriormente mencionado.

- Um sistema de testes normal como pode ser disponibilizado no eLearning, onde os alunos têm acesso a uma perguntas de escolha múltipla com enunciado, imagem do circuito a acompanhar e 4 possíveis respostas.
- Um sistema de gestão e manuseamento de utilizadores neste caso alunos e docentes, para este efeito usando entidades como turmas. Juntamente com este o docente irá ter acesso a uma página de *overview* da turma onde poderá identificar dificuldades e o progresso geral da turma. Os alunos teriam apenas acesso á sua página de aluno onde poderiam aceder á sua informação pessoal e progresso
- E por fim o sistema de progressão dos alunos mencionado anteriormente terá de ser implementado juntamente com sistemas de recompensas (medalhas, high-scores etc..).

4.1.2 Descrição do Contexto

Esta secção pretende descrever a maneira como a nossa plataforma deve ser utilizada pelos vários utilizadores que nesta se inserem. É esperado que a nossa plataforma seja criada já com uma pequena base de dados de exercícios que sejam capazes de ajudar os nossos utilizadores, com isto dito, é esperado que no nosso sistema haja um docente que encoraje os seus alunos a criar uma conta na plataforma, de seguida, este irá poder criar um grupo ou turma que contenha os seus respetivos alunos.

Desta forma os utilizadores poderão tirar o proveito máximo desta plataforma podendo aceder a uma biblioteca de exercícios, realizar testes e jogos didáticos, mas também ter acesso a testes específicos disponibilizados pelo docente da sua turma. Assim, tanto da parte do aluno como do docente, haverá uma maior confiança e sintonia no tipo de conhecimento e raciocínio que é esperado do aluno para uma boa apreciação e conhecimento na matéria.

4.1.3 Atores

Os utilizadores alvo da nossa plataforma serão alunos e docentes que terão um conhecimento na área da tecnologia razoavelmente alto ou acima da média. No entanto a interface apresentada aos utilizadores da nossa plataforma será feita de modo que consiga ser utilizada por utilizadores com mínima experiência tecnológica.

Os conhecimentos necessários para usar esta plataforma serão relativamente baixos, até porque, esta plataforma será feita num ambiente web através de um website, algo que é bastante familiar para uma audiência num ambiente escolar e académico.

Os principais utilizadores da nossa plataforma irão dividir-se maioritariamente em dois grupos:

- **Aluno:** Representa todas os utilizadores que têm o objetivo de aprender a análise de circuitos eletrónicos não dependendo do facto de estes utilizadores serem especificamente estudantes. Este utilizador terá a seu dispor uma biblioteca de exercícios e problemas da área mesmo que o mesmo não esteja autenticado ou tenha uma conta na nossa plataforma. Ao ser um utilizador registado este terá mais funcionalidades ao seu dispor como: a realização de testes, jogos ou a visualização de estatísticas.
- **Docente:** O docente irá representar os utilizadores que são efetivamente professores ou professores auxiliares, utilizadores estes que beneficiam de uma plataforma que lhes facilite o ensino e aprendizagem dos alunos. Este docente também terá acesso à biblioteca de exercícios sem autenticação, mas terá funcionalidades mais interessantes quando o mesmo está registado e aprovado como docente. Essas funcionalidades englobam: adicionar exercícios à biblioteca, criar e gerir turmas de alunos, disponibilização de resoluções passo-a-passo e criação de testes específicos.

4.1.4 Casos de Uso

O diagrama presente na Figure 8 demonstra os nossos casos de uso para os dois atores definidos para a nossa plataforma.

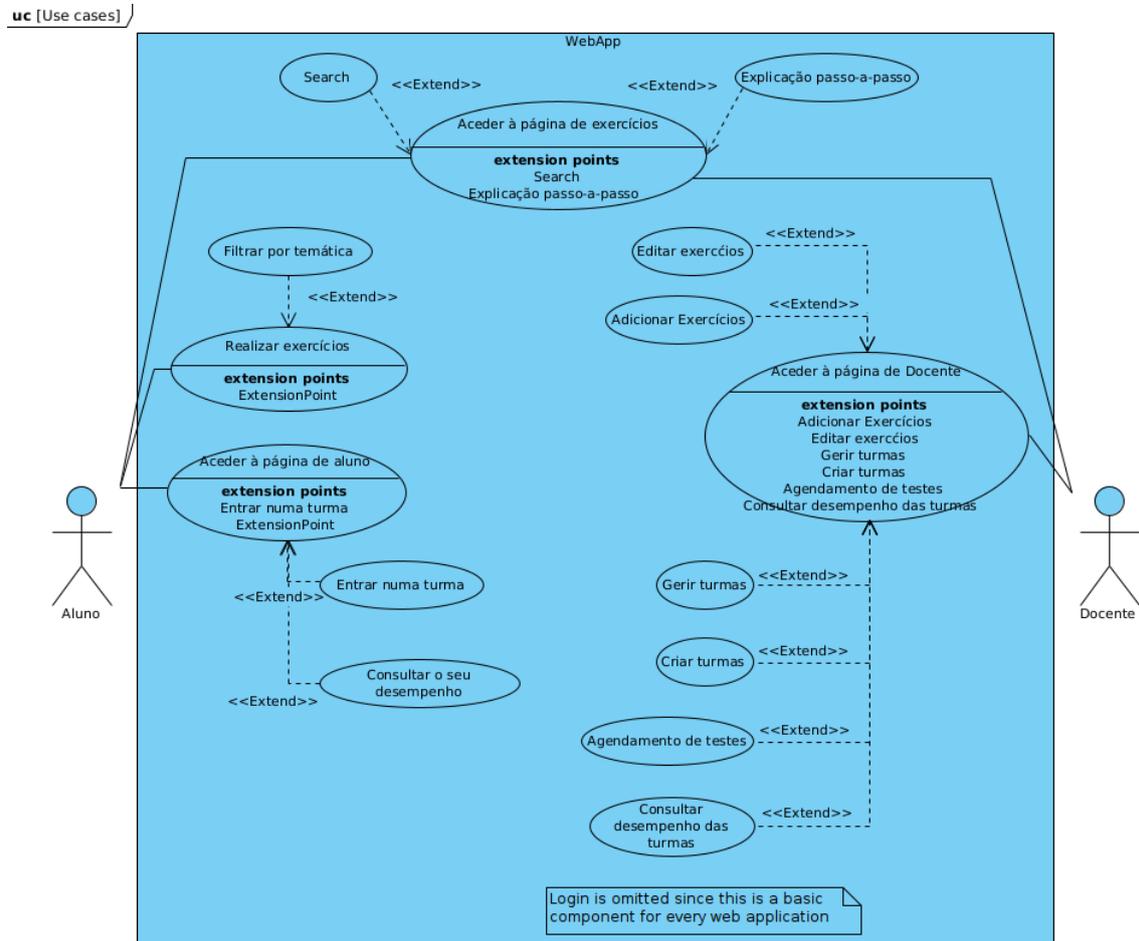


Figure 8 - Apresentação do modelo caso de usos para ambos os autores

Como pode ser visto, há uma componente que ambos partilham que é “Aceder à página de exercícios”, nesta página o utilizador pode procurar por um exercício em específico e ver a sua explicação passo-a-passo, algo que nem sempre irá estar disponível para todos os exercícios da plataforma. A procura por exercícios será feita por tema, isto é, o utilizador escolhe o tema em que a questão se enquadra e depois tem acesso a todas as questões desse tema e à pergunta que deseja consultar.

Falando sobre a realização de exercícios, na nossa plataforma os alunos podem realizar exercícios sobre análise de circuitos, podendo escolher entre duas maneiras para o fazer: Jogos didáticos ou Testes.

Os jogos didáticos consistem em jogos que, para avançar entre níveis ou aproximar-se da meta final o utilizador terá de responder corretamente á pergunta que lhe é feita. Os testes consistem em perguntas de escolha múltipla com 4 opções em que uma está correta. No final do teste é informado ao utilizador as perguntas que respondeu corretamente e as que falhou. Todas as perguntas efetuadas encontram-se na base de dados e podem ser modificados pelos docentes. Estas duas maneiras distintas de testar os conhecimentos do utilizador têm a opção de filtragem por tema, isto é, o utilizador pode escolher uma determinada matéria e todas a perguntar que surgirão enquadram-se dentro desse tema.

O aluno terá também obviamente acesso à sua página de utilizador, a página do aluno funciona como um perfil de cada aluno que tem e as várias informações sobre o mesmo. Dentro da página do aluno é possível escolher a turma a que ele pertence e é possível consultar os resultados da mesma (media de desempenho de todos os alunos da turma) e a nível individual (desempenho geral do aluno e também por cada tema do programa).

O docente por outro lado acederá à sua, a página do docente funciona também como um perfil mas desta vez do docente e conta com diferentes funcionalidades em relação a pagina do aluno. Na sua página um docente consegue fazer a manutenção de turmas que estejam a seu cargo podendo adicionar/remover alunos de uma determinada turma, criar uma nova turma, agendar testes para as suas turmas, enviando uma notificação com a data e a matéria para todos os alunos de X turmas. Poderá também consultar o desempenho da turma sendo possível discriminar o desempenho de um determinado aluno, percentagem de repostas corretas de uma questão proposta pelo docente e desempenho dos alunos numa determinada matéria do programa. Nesta página o docente pode também fazer a manutenção de exercícios que se encontram na base de dados podendo adicionar novos exercícios á base de dados ou editar exercícios que lá se encontram. Tal como já foi referido várias vezes esta adição de exercícios pode ser feita de maneira tradicional dando todos os elementos necessários para criar a pergunta ou pode ser inserida 1 pergunta genérica denotando quantas variações da mesma pergunta pertence fazer.

4.1.5 Requisitos não funcionais

A seguinte lista de requisitos irá apresentar a descrição dos vários elementos que fazem parte dos requisitos não funcionais do nosso sistema. Tendo em mente que este projeto consiste numa plataforma web os requisitos serão os seguintes:

- **Usabilidade:** Dadas as características do nosso sistema, para além do facto que esta plataforma em especial é de aprendizagem, o nosso sistema deverá ser intuitivo e de fácil aprendizagem para que a retenção de alunos e docentes seja maximizada. No caso de algum tipo de erro ou falha do sistema, este deverá ser informativo e de fácil compreensão para que os atores do nosso sistema não abandonem o sistema.
- **Segurança:** A segurança é um requisito extremamente importante, algo que ainda aumenta quando se fala de um sistema que trata de dados e problemas académicos e tendo informação pessoal de utilizadores potencialmente menores. Para isto iremos dar uso das ferramentas que o Django disponibiliza contra ataques de *cross-site scripting*, *SQL Injection*, *Cross-site Request Forgery*, entre outros. A informação privada de utilizadores e turmas terá que ser privada e apenas acessível às entidades autorizadas, disto isto, terá que ser implementado um modelo de autenticação para a nossa plataforma. Por fim, o uso do protocolo HTTPS (*Hypertext Transfer Protocol Secure*) é fundamental para que o nosso sistema consiga encriptar comunicações, e com isso, garantir uma conexão segura.
- **Capacidade:** Dada a natureza do nosso projeto é obvio que o nosso sistema tenha que ter uma grande capacidade de utilizadores, ou seja, o sistema terá que ser capaz de aguentar com a *load* de muitos utilizadores, garantido os recursos que normalmente são esperados numa *load* baixa, isto é, servir conteúdo, capacidade de reposta, etc.
- **Responsivo:** Falando na capacidade de resposta, esta terá que ser garantida em momentos de grande *flow* de utilizadores, mas outro dos requisitos (dado as tecnologias escolhidas) será ter uma resposta reativa extremamente baixa.
- **Portabilidade:** Apesar do nosso sistema ser baseado numa plataforma web, iremos ter que garantir que o nosso sistema consegue ser acedido por vários dispositivos focando-nos maioritariamente em computadores fixos e portáteis.

4.1.6 Suposições e Dependências

Dadas as características do nosso sistema, podemos dizer com confiança que as 3 grandes dependências do nosso projeto passam por depender da capacidade do servidor da plataforma

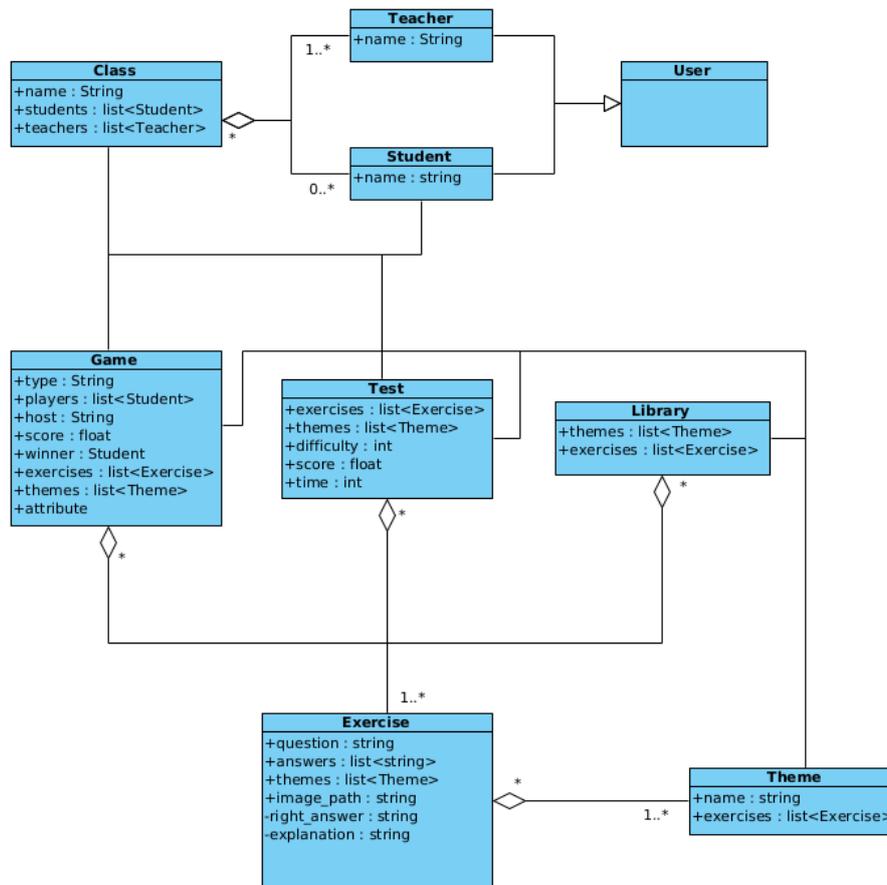
aguentar com um elevado número de pedidos, este passo irá coincidir com as especificações da máquina em questão (capacidade de processamento, conexão à rede, etc). Iremos depender também das empresas e comunidades *open source* que mantêm o *software* e *frameworks* usadas, nomeadamente a manutenção no que toca à segurança de dados. Finalmente, iremos depender da implementação já feita no CircuitSolver para a resolução e descrição de exercícios.

4.2 Arquitetura do Sistema

Esta secção irá apresentar os principais componentes do nosso sistema, apresentando o modelo de domínio e o modelo tecnológico.

4.2.1 Modelo de Domínio

O modelo de domínio apresentado na Figure 9 descreve as várias entidades e relações que existem no nosso sistema.



Library can be accessed by everyone even if the user is not authenticated and does not fall into Student or Teacher

Figure 9 - Modelo de domínio

Para começar a entidade de utilizador (user) pode ser dividida em dois atores, o docente (teacher) e o aluno (student) ambos têm acesso ligações às mesmas entidades, sendo que realizam operações diferentes. Apesar da ligação entre docente e realização de exercícios existir esta não é estritamente necessária, já que o grande objetivo é o aluno realizar os exercícios para adquirir e melhorar conhecimentos.

Outra entidade base para o uso do nosso sistema é uma turma (class), uma turma terá que ter 1 ou mais docentes, sendo que pode ter um número variável de alunos, no entanto tanto os docentes como alunos podem pertencer a várias turmas.

Quanto ao resto das entidades, podemos ver que o jogos e testes são entidades que podem ser usadas pelas turmas e alunos. Essas entidades não são nada mais nada menos do que um conjunto de exercícios (exercise), a entidade de exercício em si possui os atributos que seriam de esperar como pergunta, respostas, resposta correta, explicação passo-a-passo e imagem associada à mesma. A grande diferença que separa a entidade jogo de teste é que um teste é feito em modo singular por um aluno, sendo que um jogo já envolve um conjunto de dois ou mais alunos. Tal como o que separa a biblioteca das outras entidades relacionadas com o exercício ser o modo como a lista de exercícios é servida ao utilizador.

4.2.2 Modelo Tecnológico

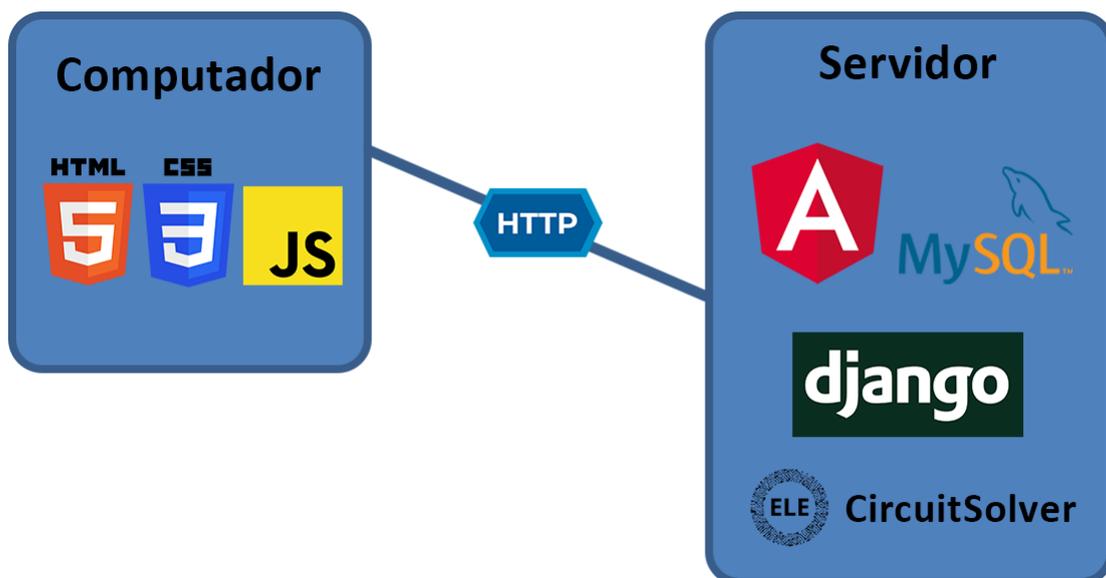


Figure 10 - Modelo tecnológico entre user-servidor

O nosso modelo tecnológico, que pode ser visto na Figure 10, é descrito por um utilizador a aceder ao nosso sistema através de um web browser suportado, que vai interpretar o conteúdo que lhe é dado através de HTML, CSS e Javascript. O nosso servidor pode ser dividido por 2 serviços, o servidor que mostra a página ao utilizador que é gerado em Angular e o servidor Django que serve de intermediário para as bases de dados. A comunicação por parte do utilizador para o nosso sistema é feita através do protocolo HTTP.

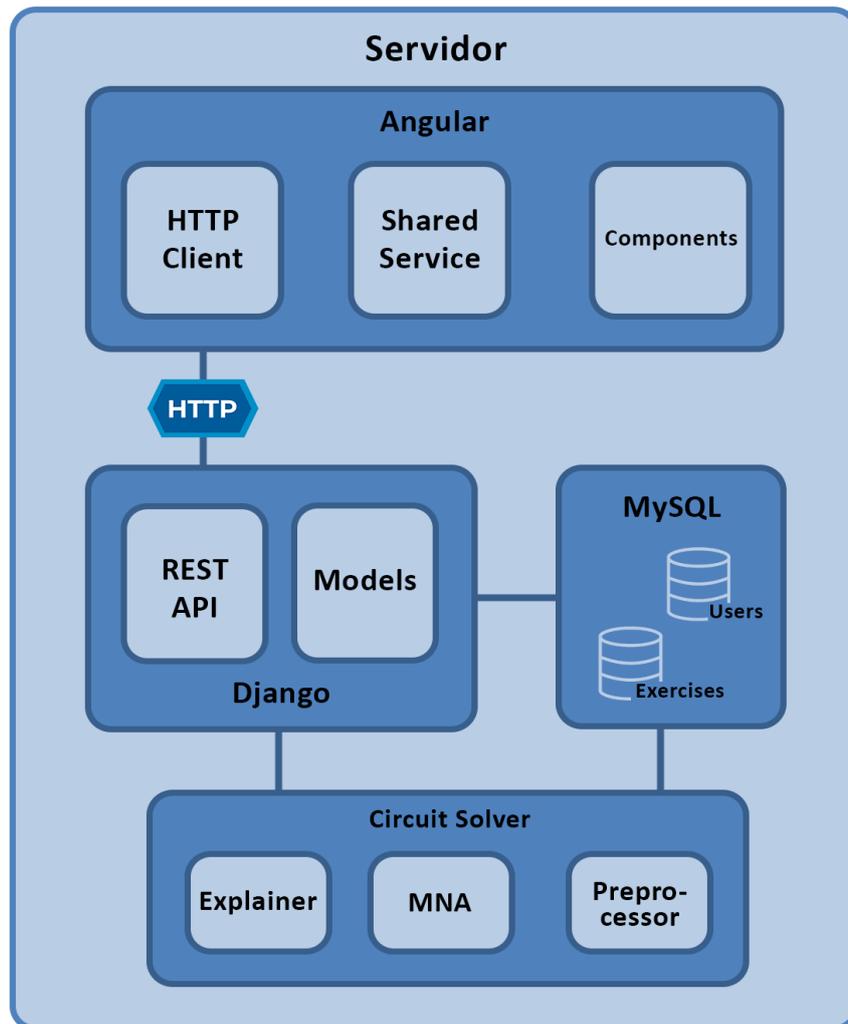


Figure 11 - Modelo tecnológico dentro do servidor

Por outro lado, é possível observar na Figure 11 as conexões e de que maneira usamos as tecnologias no lado do servidor. Tal como dito anteriormente o serviço de angular cria um servidor para servir conteúdo ao utilizador, este serviço faz pedidos à REST API Framework implementada no Django usando o protocolo HTTP. Por sua vez o serviço Django está ligado diretamente às nossas bases de dados e ao módulo de CircuitSolver, isto faz com que o nosso sistema seja seguro e de fácil escalabilidade.

Capítulo 5

Conclusão

Com base no que foi apresentado, este projeto tem em mente a modernização do ensino no domínio da eletrônica através do uso das tecnologias atuais, nomeadamente o desenvolvimento de uma plataforma web. Esta plataforma tem como objetivos tornar o ensino mais apelativo, através de métodos interativos. Pretende-se facilitar a compreensão de análise de circuitos por parte dos alunos e disponibilizar ferramentas necessárias para a criação de um ambiente de trabalho mais dinâmico e comunicativo. Para a criação desta plataforma web são utilizadas ferramentas *open source* e integrar módulos já desenvolvidos.

Capítulo 6

Referências

- [1] Arun Ravindran. 2018. Django Design Patterns and Best Practices - Second Edition: Industry-standard Web Development Techniques and Solutions Using Python (2nd. ed.). Slovakia
- [2] Duolingo. Retirado em 11 de janeiro, 2022 de <https://www.duolingo.com/>
- [3] Google. 2016. Retirado em 6 de dezembro, 2021 de <https://angular.io/>
- [4] Google Classroom. Retirado em 11 de janeiro, 2022 de <https://classroom.google.com/>
- [5] Hiren Dhaduk. 2020. Industry-Standard Web Development Techniques for Angular. (July 2020). Retirado em 6 de dezembro, 2021 de <https://dzone.com/articles/industry-standard-angular-web-development-techniqu>
- [6] Khan Academy. Retirado em 11 de janeiro, 2022 de <https://www.khanacademy.org/>
- [7] SIACUA. Retirado em 11 de janeiro, 2022 de <https://siacua.web.ua.pt/>
- [8] Symbolab. Retirado em 11 de janeiro, 2022 de <https://www.symbolab.com/>
- [9] Vue.js. Retirado em 11 de janeiro, 2022 de <https://vuejs.org/v2/guide/team.html>
- [10] Wikipedia. Retirado em 11 de janeiro, 2022 de https://en.wikipedia.org/wiki/Ryan_Dahl
- [11] Wikipedia. Retirado em 11 de janeiro, 2022 de https://en.wikipedia.org/wiki/OpenJS_Foundation