

Final

Academic-Manager
Sistema para a gestão escolar
RELATÓRIO DE ESTÁGIO DE MESTRADO

Rosa Elena do Nascimento Pascoal Garcia
MESTRADO EM ENGENHARIA INFORMÁTICA



novembro | 2016



Academic-Manager – Sistema para a gestão escolar

Rosa Elena do Nascimento Pascoal Garcia

(Licenciada)

*Tese Submetida à Universidade da Madeira para a
Obtenção do Grau de Mestre em Engenharia Informática*

Funchal – Portugal

Novembro - 2016

Orientador Interno:

Eduardo Leopoldo Fermé, PhD

Professor Associado com Agregação da facultade de Ciências Exatas e da Engenharia da Universidade da Madeira

ABSTRACT

This report was made with the intent of describing and presenting methodologies, tools and technologies used to develop my project, in curricular internship context and to finish the Master's degree in Computer Science and Engineering. With this document I also intend to expose my experience during the internship term.

The main goal of achieving a curricular internship arose from the need of putting into practice the knowledge acquired during the bachelor's degree and master's degree, to live a great experience outside the classroom, contact and meet the world of work and develop my areas of expertise.

Throughout this report, I will describe the entire process of developing a school management system tailored to Colégio Guadalupe's needs, as well as explain the techniques, methodologies and tools used to accomplish the main goal and, finally, I will present the results achieved.

In this project there were addressed areas such as Software Engineering and techniques for Managing Software projects.

KEYWORDS

Software Engineering

Web Application

Project Management

Agile Methods

RESUMO

Este relatório foi feito com a finalidade de descrever e apresentar metodologias, ferramentas e tecnologias utilizadas no desenvolvimento do projeto, no âmbito de estágio curricular para a conclusão do curso do mestrado em Engenharia Informática. Neste documento também pretendo expor a minha experiência vivida durante a duração do mesmo.

O objetivo de realizar um estágio curricular surgiu da necessidade de pôr em prática os conhecimentos adquiridos durante o percurso curricular da licenciatura e do mestrado, viver uma grande experiência fora da sala de aula, contatar e conhecer o âmbito laboral e desenvolver as minhas áreas de conhecimento.

Ao longo deste relatório, irei descrever todo o processo de desenvolvimento de um sistema para a gestão escolar desenvolvido à medida das necessidades do Colégio Guadalupe, assim como também explicarei as técnicas, metodologias e ferramentas utilizadas para cumprir com o objetivo principal e, finalmente, apresentarei os resultados obtidos.

Neste projeto foram abordadas áreas como a Engenharia de Software e técnicas para a gestão projetos de Software.

PALAVRAS-CHAVE

Engenharia de Software

Aplicação Web

Gestão de Projeto

Métodos Ágeis

AGRADECIMENTOS

O sucesso deste projeto deveu-se à contribuição de um amplo grupo de pessoas que, direta ou indiretamente, influenciaram positivamente o meu trabalho. Expresso, assim, os meus sinceros agradecimentos a todos.

Em primeiro lugar, quero que agradecer a Deus, aos meus pais e ao meu marido, pelo apoio incondicional, assim como a paciência e encorajamento que me transmitiram em todo momento, contribuindo, assim, para a realização dos meus objetivos.

Um especial agradecimento ao meu orientador Professor Doutor Eduardo Fermé, pelo seu total apoio e disponibilidade, por acreditar na mais valia da realização do estágio curricular, pela sua orientação, notáveis e imprescindíveis para a conclusão do trabalho e deste relatório.

À Universidade da Madeira por permitir realizar o estágio curricular, assim como a todos os professores que, através das aulas, me transmitiram conhecimentos necessários para a realização deste projeto.

De igual modo, quero agradecer a entidade proprietária do Colégio Guadalupe, o grupo CELFF - Centro de Estudos Línguas e Formação do Fogueteiro S.A, por me permitir realizar o estágio nas suas instalações, assim como, a total disponibilidade e ajuda da Direção e Administração do Colégio, sem a vossa colaboração não conseguiria alcançar o meu propósito.

A todo os colaboradores do Colégio, pois com eles aprendi muito, em especial aos professores que lá trabalham, que sempre estiveram dispostos a contribuir e colaborar comigo.

Por último, e não menos importante, quero agradecer à minha família, amigos e novos amigos que ganhei com esta experiência, eles garantiram o tão necessário equilíbrio emocional durante este processo.

ÍNDICE

| | |
|---|-----------|
| I. Introdução | 4 |
| I.1. Apresentação | 5 |
| I.2. Contexto do Estágio | 6 |
| I.3. Contexto Institucional | 7 |
| I.3.1. Descrição do Colégio | 7 |
| I.3.2. Localização | 8 |
| I.3.3. Organigrama | 8 |
| I.4. Tarefas a desempenhar no estágio | 10 |
| I.5. Problemática contextual: Sistema de inscrições e lançamento de notas | 12 |
| I.5.1. Matrícula/Inscrição | 12 |
| I.5.2. Lançamento de notas | 12 |
| I.5.3. Academic Manager | 14 |
| II. Estado da arte | 16 |
| II.1. Metodologia de trabalho e ferramentas de apoio | 17 |
| II.1.1. Metodologias Ágeis | 17 |
| II.1.2. As 5 dimensões de um sistema de software | 19 |
| II.1.3. Ferramentas utilizadas no desenvolvimento do software | 21 |
| II.1.4. Projetos semelhantes | 26 |
| II.2. Conclusão..... | 29 |
| III. Desenvolvimento do sistema | 30 |
| III.1. Ciclo de vida de um sistema de software..... | 31 |
| III.1.1. Modelo espiral..... | 31 |
| III.2. Requisitos..... | 33 |
| III.2.1. Requisitos funcionais | 33 |
| III.2.2. Requisitos não funcionais | 34 |
| III.3. Casos de utilização | 36 |
| III.3.1. Caso de utilização do módulo de inscrição | 36 |
| III.3.2. Caso de utilização do módulo de atividades extracurriculares | 37 |
| III.3.3. Caso de utilização do módulo dos serviços | 37 |
| III.3.4. Caso de utilização do módulo de lançamento de notas | 38 |
| III.3.5. Caso de utilização do módulo de currículo académico | 39 |
| III.4. Arquitetura | 40 |
| III.4.1. Estilo arquitetural – Cliente-Servidor | 40 |
| III.4.2. Padrão arquitetural..... | 41 |
| III.5. Diagrama de Entidade-Relação | 44 |
| III.5.1. Módulo de inscrição: | 45 |
| III.5.2. Módulo de atividades extracurriculares..... | 46 |
| III.5.3. Módulo de serviços | 47 |
| III.5.4. Módulo de lançamento de notas | 47 |
| III.5.5. Módulo de currículos académicos..... | 49 |
| III.5.6. Outras entidades | 49 |

| | | |
|----------|--|----|
| III.6. | Interfaces do utilizador | 51 |
| III.6.1. | Interfaces de configuração do sistema | 52 |
| III.6.2. | Interfaces do módulo principal – Inscrições | 55 |
| III.6.3. | Interface do módulo de atividades extracurriculares | 58 |
| III.6.4. | Interface do módulo de serviços | 59 |
| III.6.5. | Interface do módulo de currículos académicos | 60 |
| III.7. | Implementação | 62 |
| III.7.1. | Tecnologias utilizadas | 62 |
| III.8. | Funcionalidades implementadas | 73 |
| III.8.1. | Funcionalidades implementadas | 73 |
| III.8.2. | Funcionalidades não implementadas | 74 |
| III.9. | Software Testing | 75 |
| III.10. | Conclusão | 76 |
| IV. | Conclusão e trabalho futuro | 77 |
| IV.1. | Visão crítica do estágio | 78 |
| IV.2. | Perspetivas futuras | 80 |
| V. | Referências bibliográficas | 81 |
| VI. | Anexos | 85 |
| VI.1. | Anexo 1 – Repositório no Bitbucket | 86 |
| VI.2. | Anexo 2 – Organização de tarefas no trello | 87 |
| VI.3. | Anexo 3 – Base de dados em phpMyadmin | 88 |
| VI.4. | Anexo 4 – Desenho da base de dados em MySQL Workbench | 89 |
| VI.5. | Anexo 5 – Terminal de comando Git Bash | 90 |
| VI.6. | Anexo 6 – Exemplo de um currículo académico gerado pelo sistema – Página 1/2..... | 91 |
| VI.7. | Anexo 7 – Exemplo de um currículo académico gerado pelo sistema – Página 2/2..... | 92 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 - Logotipo do Colégio Guadalupe..... | 8 |
| Figura 2 - Logotipo do grupo Celf..... | 8 |
| Figura 3 - Organograma do Colégio Guadalupe | 9 |
| Figura 4 - As 5 dimensões de um projeto de software..... | 19 |
| Figura 5 - Exemplo de diagrama de flexibilidade | 21 |
| Figura 6- Diagrama de Flexibilidade de Academic-Manager..... | 21 |
| Figura 7 - Logotipo de phpMyAdmin..... | 22 |
| Figura 8 - Logotipo de MySQL Workbench | 23 |
| Figura 9 - Logotipo de PhpStorm | 24 |
| Figura 10 - Logotipo de Trello..... | 25 |
| Figura 11 - Logotipo de Git..... | 25 |
| Figura 12 - Logotipo de Bitbucket | 25 |
| Figura 13 - Modelo espiral do ciclo de vida de um sistema | 31 |
| Figura 14 - Diagrama de casos de utilização do módulo de inscrição | 36 |
| Figura 15 - Diagrama de casos de utilização do módulo de atividades extracurriculares | 37 |
| Figura 16 - Diagrama de caso de utilização do módulo de serviços | 38 |
| Figura 17 - Diagrama de casos de utilização do módulo de lançamento de notas | 38 |
| Figura 18 - Diagrama de casos de utilização do módulo de currículos académicos | 39 |
| Figura 19 - Arquitetura Cliente-Servidor..... | 41 |
| Figura 20 - Representação gráfica do padrão arquitetural MVC | 43 |
| Figura 21 - Diagrama Entidade-Relação | 44 |
| Figura 22 - Diagrama de entidade relação do módulo de inscrição | 46 |
| Figura 23 - Diagrama de entidade relação do módulo atividades extracurriculares | 47 |
| Figura 24 - Diagrama de entidade relação do módulo de serviços | 47 |
| Figura 25 - Diagrama de entidade relação do módulo de lançamento de notas | 48 |
| Figura 26 - Diagrama de entidade relação do módulo de currículos académicos | 49 |
| Figura 27 - Diagrama de entidade relação de entidades criadas pelo framework | 50 |
| Figura 28 - Vista para criar um novo ano letivo | 52 |
| Figura 29 - Vista para criar um novo ano escolar | 53 |
| Figura 30 - Vista para criar um novo curso | 53 |
| Figura 31 - Vista para criar uma nova disciplina..... | 54 |
| Figura 32 - Vista para criar uma nova turma | 55 |
| Figura 33 - Vista para criar um novo aluno..... | 56 |
| Figura 34 - Vista para criar um familiar | 57 |
| Figura 35 - Vista para realizar a inscrição de um aluno | 58 |
| Figura 36 - Vista de inscrição de alunos numa turma..... | 58 |
| Figura 37 - Vista para criar uma atividade extracurricular | 58 |
| Figura 38 - Vista de inscrição de um aluno numa atividade extracurricular..... | 59 |
| Figura 39 - Vista de criação de um serviço | 59 |
| Figura 40 - Vista de adquisição de uma serviço por parte de um aluno | 60 |
| Figura 41 - Vista de criação de currículo académico de um aluno..... | 60 |
| Figura 42 - Vista da listagem dos alunos com currículo académico criado | 61 |
| Figura 43 - Logo do framework Laravel | 64 |
| Figura 44 - Padrão MVC implementado por Laravel | 64 |
| Figura 45 - Logo de Vagrant..... | 66 |
| Figura 46 - Logo de VirtualBox | 66 |
| Figura 47 - Logo de Composer | 68 |
| Figura 48 - Logo de Packagist | 68 |
| Figura 49 - Logo de Laravel Collective | 69 |
| Figura 50 - Logo de Bootstrap | 71 |
| Figura 51 - Logo de Font Awesome..... | 72 |

ACRÓNIMOS/SIGLAS

MVC - Model View Controller.

HTTP - Hypertext Transfer Protocol

URL - Uniform Resource Locator

HTML - HyperText Markup Language

HTML5 - HyperText Markup Language 5

CSS - Cascading Style Sheets

PHP - Hypertext Preprocessor

PDF - Portable Document Format

XP - Extreme Programming

SQL - Structured Query Language

CSV - Comma-Separated Values

IDE - Integrated Drive Electronics

VCS - Version Control System

XML - eXtensible Markup Language

ISSO - International Standards Organization

QBE - Query by Example

I. INTRODUÇÃO

"Success is the result of perfection, hard work, learning from failure, loyalty, and persistence"

- Colin Powell

I.1. APRESENTAÇÃO

O presente relatório não é só apresentado com a finalidade de concluir o 2º ciclo de Engenheira Informática, como também tem o propósito de descrever toda a experiência e o grande leque de conhecimento adquirido fora do contexto de sala de aula no âmbito da realização de um estágio curricular na área educacional, no Colégio Guadalupe.

Durante o tempo do estágio, tive a oportunidade de pôr em prática todo o conhecimento adquirido durante o 1º e 2º ciclos do curso de Engenheira Informática da Universidade da Madeira, da mesma forma tive a verdadeira noção de como pode ser o mercado de trabalho na área da informática.

O Colégio Guadalupe deu-me a oportunidade de fazer um levantamento de informação das necessidades informáticas que possuem, encontrar as melhores soluções para os problemas encontrados, desenhar um sistema que pretende ser uma solução para os problemas diagnosticados e, finalmente, implementar alguns módulos do referido sistema.

I.2. CONTEXTO DO ESTÁGIO

Depois do término de todas as unidades curriculares do 2º Ciclo de Engenharia Informática e, para obter o grau de mestre na Universidade da Madeira, é necessário optar entre uma dissertação, trabalho de projeto ou estágio curricular.

A minha seleção baseou-se na necessidade de pôr em prática os conhecimentos adquiridos durante o percurso curricular da licenciatura e do mestrado, contactar e conhecer o âmbito laboral, viver uma grande experiência fora da sala de aula, desenvolver as minhas áreas de conhecimento, de forma a torná-las ferramentas úteis para o âmbito profissional. Por estes motivos, decidi fazer um estágio curricular.

Logo após a tomada de decisão, iniciou-se a árdua tarefa de encontrar alguma empresa/instituição que respondesse ao meu pedido. Tendo sido o Colégio Guadalupe, na Verdizela, Freguesia de Corroios, do Grupo Celf, que concordou com a realização deste estágio curricular.

O primeiro contato com o grupo CELF foi na E.P.H.T Madeira, onde me foi feita uma entrevista em que me foi dada uma introdução do que era o grupo CELF. Também me foi pedido um protocolo de estágio, onde deveria explicar em que consistia e demonstrar quais os objetivos do projeto a desenvolver (e o que pretendia fazer). Despois de passar esta fase, dirigi- -me ao Colégio Guadalupe para ser entrevistada novamente. Nesta entrevista, fui informada sobre as necessidades informáticas que o Colégio tinha.

O objetivo principal do estágio era desenhar e desenvolver uma aplicação web que se encarregasse do processo de matrícula/inscrição no colégio, lançamento de notas e gerador automático de currículos académicos. A este sistema dei-lhe o nome de *Academic-Manager*.

I.3. CONTEXTO INSTITUCIONAL

I.3.1. Descrição do Colégio

A entidade proprietária do Colégio Guadalupe, o CELFF - Centro de Estudos Línguas e Formação do Fogueteiro S.A., é responsável pela definição e aplicação dos princípios que orientam o ensino que o Colégio ministra, e garante a coesão de todos os que colaboraram na vida da Escola.

São órgãos do Colégio o Conselho de Administração, a Direção Pedagógica, o Conselho Pedagógico e o Conselho de Turma.

O Conselho de Administração é um órgão que pertence a CELFF S.A. e que a representa junto do Colégio Guadalupe.

O Colégio Guadalupe é um estabelecimento de ensino privado, situado em pleno pinhal da Aroeira. O seu nome foi inspirado na sua fundadora, Maria de Guadalupe que, há cerca de cinquenta anos, criou o Externato Infante D. Henrique e, mais tarde, o Externato Novo Dia.

Em condições ambientais e arquitetónicas de excelência, o Colégio tem-se diferenciado pela sua linha inovadora e de investigação levada a cabo pelos alunos, incentivados pelos professores.

As suas instalações reúnem todos os avanços pedagógicos e tecnológicos, fatores essenciais ao desenvolvimento de um ensino de qualidade.

Destaca-se pela sua integração na paisagem natural em que se encontra inserido, criando um ambiente de estudo que favorece a tranquilidade e sossego dos alunos, fator importantíssimo para o equilíbrio emocional dos mesmos.

Com espaços bem definidos para o Pré-escolar, o Ensino Básico e o Ensino Secundário, todo o edifício foi pensado no incentivo de partilha entre grandes e pequenos.

Princípios Orientadores

O Colégio Guadalupe elegeu os seguintes princípios na sua atividade educativa:

Promover no Aluno um ensino de qualidade, com aprendizagens bem-sucedidas, e um reconhecimento equilibrado de si próprio e um relacionamento positivo com os outros;

Contribuir para a estabilidade e segurança afetiva com respeito pelas características individuais de cada aluno;

Incentivar o respeito pela família, pela sua identidade cultural, e pelos valores nacionais;

Garantir a aquisição e estruturação de conhecimentos sobre a natureza, a sociedade e a cultura, promovendo o respeito pelos direitos do Homem e liberdades fundamentais, e pelos princípios consagrados na Constituição da Carta das Nações Unidas, Convenção Europeia dos Direitos do Homem e na Convenção sobre os Direitos da Criança;

Proporcionar o desenvolvimento físico, estimulando o desporto, adaptando a sua prática, às capacidades próprias de cada idade;

Estimular o desenvolvimento de aptidões técnicas no sentido da solução de problemas práticos e da iniciação ao conhecimento tecnológico;

Contribuir para o desenvolvimento do sentido crítico e da sensibilidade estética.

Todos estes princípios e linhas orientadoras ajudam a concretizar os fins a que se propõem.

I.3.2. Localização

O Colégio Guadalupe encontra-se localizado na Rua Parque Natural do Alvão, 26, Corroios, Seixal, 2855-620, Portugal.



Figura 1 - Logotipo do Colégio Guadalupe



Figura 2 - Logotipo do grupo Celff

I.3.3.Organograma

A entidade proprietária do Colégio Guadalupe, CELFF – Centro de Estudos Línguas e Formação do Fogueteiro S.A., é responsável pela definição e aplicação dos princípios

que orientam o ensino que o Colégio ministra e garante a coesão de todos os que colaboram na vida da Escola.

São órgãos do Colégio o Conselho de Administração, a Direção Pedagógica, o Conselho Pedagógico e os Conselhos de Turma.

O Conselho de Administração é um órgão que pertence à CELFF S.A. e que a representa junto do Colégio Guadalupe.

A estrutura interna do Colégio Guadalupe está constituída pela seguinte hierarquia:

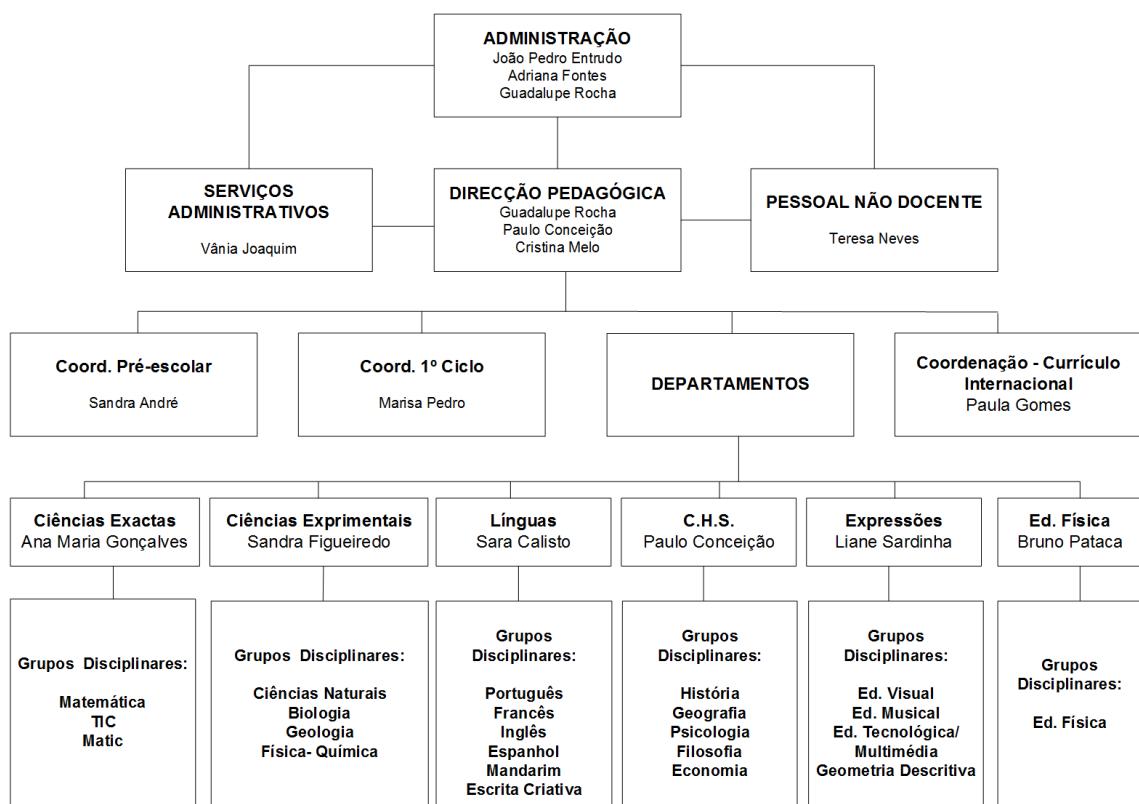


Figura 3 - Organogramma do Colégio Guadalupe

I.4. TAREFAS A DESEMPENHAR NO ESTÁGIO

Durante o tempo que estive no estágio, a direção do Colégio disponibilizou-me um espaço onde podia trabalhar as horas que considerasse necessárias, assim como também sempre esteve à minha disponibilidade, na medida do possível, pessoal docente e não docente para esclarecer a minhas dúvidas, explicar o funcionamento do sistema de matrículas e lançamento de notas, da mesma forma que me foram proporcionados documentos e acessos ao sistema informático.

Ao longo do estágio, as minhas tarefas foram divididas em três fases.

A primeira fase, foi o levamento de informação, dito levamento foi feito através de entrevistas às pessoas responsáveis pelas diferentes áreas, tive contacto com o pessoal da secretaria, os quais me explicaram como realizavam o processo de inscrições/matrícula dos alunos no Colégio, o processo de inscrições nas diferentes atividades extracurriculares e dos diferentes serviços que o Colégio oferece.

Depois tive a oportunidade de falar com a Direção Pedagógica e alguns professores, que me explicaram o processo de lançamento de notas, como eram os diferentes documentos de avaliação que os docentes utilizavam para avaliar os alunos, quais eram os diferentes passos do processo de lançamento de notas e, finalmente, como chegava esta informação aos encarregados de educação e aos alunos.

Ao logo da conversa, a direção manifestou a vontade de ter um sistema gerador automático de currículos académicos. Pretendiam que o sistema registasse um resumo, por ano letivo, das atividades onde os alunos participaram e, de certa forma, se se destacaram ao longo do percurso académico, atividades estas, como por exemplo, participação em competições desportivas, regionais, estatais, nacionais e/ou internacionais, participação em diferentes olimpíadas de matemática, biologia, filosofia, participação em projetos nacionais e internacionais, entre outras de relevo. Este sistema devia gerar um documento PDF com toda esta informação, assim como a informação pessoal e fotografia do aluno.

Ainda no levantamento de informação, tive a oportunidade de aceder e explorar o sistema informático utilizado pelos funcionários para o lançamento de notas e perceber melhor o problema manifestado neste processo. Fui convidada a presenciar um conselho de turma de avaliação intercalar e de avaliação periódica do primeiro período de cada um dos ciclos (1º, 2º, 3º ciclos e secundário) e tive acesso a documentos internos de avaliação. Também tive a oportunidade de presenciar a inscrição de um aluno por parte dos encarregados de educação no colégio e participar de forma não ativa nos processos a seguir.

Depois do processo do levantamento de informação, passei à fase de desenho do sistema. Após muita pesquisa, investigação e análise da informação recolhida, decidi que o sistema iria ser dividido em módulos. Um módulo principal de inscrição e quatro módulos secundários, o módulo de avaliações, módulo de serviços, módulo de atividades extracurriculares e, finalmente, o módulo de currículos académicos. Os módulos secundários não dependem uns dos outros, mas sim, do sistema principal, o que permite uma maior flexibilidade na hora de manutenção e evolução do sistema.

Posteriormente passei à fase de desenvolvimento. Nesta etapa, estive a estudar e analisar as diferentes tecnologias que existem atualmente no mercado, para compreender quais delas me ajudariam no desenvolvimento do sistema. Após a escolha, passei ao desenvolvimento, seguidamente, à implementação e, finalmente, aos testes.

I.5. PROBLEMÁTICA CONTEXTUAL: SISTEMA DE INSCRIÇÕES E LANÇAMENTO DE NOTAS

I.5.1. Matrícula/Inscrição

O processo de inscrição/matrícula é feito em documentos impressos, onde os encarregados de educação preenchem as fichas com as informações necessárias para a inscrição, tais como: dados pessoais, dados dos pais/encarregados de educação, ano de escolaridade onde pretende inscrever o educando, se pretende receber aulas de Educação Moral Religiosa e Católica, língua estrangeira II (no caso de ser aluno do 3º Ciclo), se deseja encomendar os manuais escolares, se necessita do serviço de transporte, de alimentação, atividades extracurriculares e se autoriza a divulgação de imagem. No caso do ensino secundário, escolhe-se o curso e as disciplinas específicas.

Com as informações fornecidas, o departamento administrativo cria grelhas em folhas de cálculo Excel, obtendo, assim, a informação necessária para gerir as turmas, alunos que desejam o serviço de transportes, alimentação, atividades extracurriculares, entre outras.

Com base na informação fornecida pelos encarregados de educação no ato da inscrição, que posteriormente é compilada pelo departamento administrativo, a informação é inserida no chamado sistema de “alunos”, criando uma atualização na ficha de inscrição com os dados recentemente fornecido. No caso dos alunos novos, os dados são introduzidos de raiz no sistema “alunos” e criando um novo perfil.

I.5.2. Lançamento de notas

O processo de notas é ainda muito mais complicado, pois existem dois tipos de avaliações, as avaliações intercalares e as de final de período. Nas avaliações intercalares são atribuídas notas qualitativas, onde os encarregados de educação têm uma noção como está a decorrer o período antes do término do mesmo, estas ocorrem a meio do 1º e 2º períodos. As notas finais de período são quantitativas (0-20) e são atribuídas pelos professores responsáveis por cada disciplina no final de cada período (1º, 2º e 3º períodos).

Em cada uma das avaliações, os encarregados de educação recebem um registo de avaliação.

Para produzir os registos de avaliação intercalar, os professores de cada disciplina entram na *drive* do Google, onde se encontram livros Excel com cada uma das turmas. Em cada livro existem várias páginas, em que a primeira recebe o nome de “pauta”. Na pauta encontram-se cada um

dos alunos que integram a turma e cada uma das disciplinas que eles frequentam com os parâmetros a ser avaliados. Cada professor deve preencher os espaços correspondentes à disciplina nos parâmetros de avaliação de cada aluno. As folhas seguintes são os registos de avaliação de cada um dos alunos com interligação à primeira folha (pauta), estas ligações permitem que se preencham os registos automaticamente.

Nas avaliações de final de período, o procedimento é ligeiramente diferente. Cada professor tem uma grelha de avaliação por cada turma, nesta grelha encontram-se os diferentes âmbitos de avaliação, temos as competências científicas e 0sa competências atitudinais, cada uma com a sua respetiva percentagem, dependendo da disciplina e do ciclo de ensino, dentro de cada competência existem várias avaliações. Nas competências científicas são avaliados os testes e trabalhos, os conhecimentos cognitivos do aluno nesse período. Nas competências atitudinais é avaliada a participação, o empenho e o respeito pelas normas na sala de aula. Nas células finais das grelhas encontra-se a classificação final de cada aluno nesse período referente a essa disciplina com uma escala de 0-20 valores.

Ao mesmo tempo, no *Google drive*, é criado, por parte dos professores encarregues dos registos de avaliação, um novo livro com as avaliações de final de período por cada turma. Neste livro, voltamos a encontrar várias folhas, a primeira folha encontra-se uma nova “pauta”, mantendo a mesma estrutura da pauta intercalar, podemos ver os nomes dos alunos que integram a turma, as disciplinas e quadros para preencher com a nota final, falta totais e faltas injustificadas de cada disciplina, nesta pauta cada professor completa a(s) célula(s) referente(s) a(s) sua(s) disciplina(s) com a classificação final obtida da grelha de avaliação, nas folhas seguintes podemos encontrar os registos de avaliação de cada um dos alunos que são preenchidos automaticamente por ligações feitas da “pauta”, os registos de avaliação constam de duas páginas, na primeira temos as disciplinas, a classificação final de período, aulas previstas, dadas e assistidas, faltas totais e injustificadas. No fim da página, existem dois quadros, o primeiro quadro recebe o nome de “Apreciação Global/Síntese Descritiva”, onde o conselho de turma escreve um comentário referente ao desempenho do aluno esse período, o segundo quadro é chamado “Observações”, este quadro só é preenchido quando existe alguma observação referente a algumas das disciplinas do registo de avaliação.

Na segunda página, existe um quadro chamado a “apreciação por disciplina”, uma avaliação qualitativa referente a cada disciplina; cada professor deve preencher com uma cruz a avaliação correspondentes de cada aluno (**I** - Insuficiente; **SP** - Suficiente Pouco; **S** - Suficiente; **B** - Bom; **MB** - Muito Bom), as quais devem coincidir com a nota final. Esta apreciação é feita aluno a aluno, tornando o processo muito repetitivo e com uma grande probabilidade de cometer algum erro.

A marcação de faltas é feita através de livros de pontos. Durante o período cada professor sinaliza no livro de ponto os alunos ausentes nas aulas, no fim do período o diretor de turma passa todas as faltas para o sistema alunos. Este sistema informático contabiliza as faltas referentes a cada aluno e a cada disciplina, estes totais são introduzidos na pauta no *Google drive* e são usadas, novamente, as ligações para introduzir as faltas automaticamente em cada um dos registos.

As aulas previstas e dadas são inseridas por cada professor no registo de avaliação do aluno N°1, isto quer dizer, na primeira folha a seguir à da pauta, os registos a seguir têm uma ligação a esta página e, assim, só é preenchido uma vez por cada turma, as aulas assistidas e a subtração entre as aulas dadas e as faltas totais.

Antes de serem publicadas as pautas e entregues os registos de avaliação aos encarregados de educação, é convocado um conselho de turma para discutir as notas e falar sobre a turma. No fim da reunião, algumas das notas poderão sofrer alterações ou encontrar algum erro em algumas das notas; são emendados os erros e, finalmente, é preenchido por cada professor um documento físico chamado “propostas de nível”, onde os professores fazem as conversões das notas de 0-20 para 1-5 para serem introduzidas no sistema alunos.

Após todo este processo, são impressas todas as pautas (de 0-20 e de 1-5) e registos de avaliação para fazer o “pente fino”. Nesse dia, após muitas horas de verificação de material, são publicadas as pautas de 0-20 no placard de informações do colégio e os diretores de turma colocam na plataforma “Moodle” a pauta da sua direção de turma para poderem ser consultadas pelos encarregados de educação. As pautas de 1-5 geradas pelo sistema “Alunos” são impressas e arquivadas.

Depois deste processo, é entregue, em mão, aos encarregados de educação uma cópia do registo de avaliação do seu educando.

I.5.3. Academic Manager

A Direção do Colégio desejava um sistema que simplificasse e facilitasse todos estes processos. Academic-Manager, foi o nome dado ao programa informático que iria ser desenhado e desenvolvido ao longo do estágio.

Pretendia-se que as matrículas fossem feitas no sistema e que este gerasse todos os relatórios que eram feitos em documentos excels, assim como também gerir as turmas, as inscrições nas diferentes disciplinas, gerir o módulo das atividades extracurriculares e dos serviços. Referente ao módulo de lançamento de notas, deseja-se que no sistema fosse possível fazer as grelhas de

cada uma das disciplinas e professores, e que a nota gerada automaticamente preenchesse as pautas (de 0-20 e de 1-5) e os registos de avaliação de cada um dos alunos.

Aquando do estágio, não existia nenhuma formatação para elaborar o currículo académico do aluno, por este motivo, foi-me pedido pela Direção do Colégio para elaborar um módulo dentro do sistema que produzisse esse documento. Neste módulo, pretende-se que o sistema gere um currículo académico com a descrição de destaque do aluno durante o seu percurso escolar, quer seja tanto no âmbito académico, distinções académicas, projetos, projetos internacionais, estágios/cursos, certificações nacionais e internacionais, quer méritos desportivos, partições em clubes, entre outros.

II. ESTADO DA ARTE

II.1. METODOLOGIA DE TRABALHO E FERRAMENTAS DE APOIO

O objetivo principal deste projeto é o desenho e desenvolvimento de um software para gestão académico. De modo a alcançar este propósito, recorri a alguma pesquisa e ao conhecimento adquirido durante o curso e, desta forma, perceber qual seria a metodologia e as ferramentas que se adequavam ao meu projeto em particular.

II.1.1. Metodologias Ágeis

Os métodos ágeis surgem a partir da necessidade de procurar flexibilizar o desenvolvimento de software, diminuir a burocracia, tornar o desenvolvimento mais rápido e permitir respostas rápidas às mudanças.

Manifesto Ágil (2001) - Filosofia de qualquer método Ágil: "*Estamos evidenciando maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através desse trabalho passamos a entender que:*

Indivíduos e interações são mais importantes que processos e ferramentas.

Software funcionando é mais importante do que documentação completa e detalhada.

Colaboração com o cliente é mais importante do que negociação de contratos.

Adaptação a mudanças é mais importante do que seguir o plano inicial.

Ou seja, mesmo tendo valor os itens à direita, valorizamos mais os itens à esquerda." [1]

Ágil é uma nova forma de gestão e desenvolvimento de Software que usa uma abordagem de planeamento e execução interativa e incremental voltada para processos empíricos (complexos, caóticos ou com muita incerteza, tem mudança ao longo do processo, não são repetitivos e são imprevisíveis) que divide o problema em produtos menores e que visa entregar software funcionando regularmente, visa a aproximação e maior colaboração da equipa de desenvolvimento com os experts de negócios, comunicação cara a cara, redução dos riscos associados às incertezas dos projetos, abraçar e responder às mudanças de forma mais rápida e natural e, é claro, a satisfação final dos clientes por meio da adoção de práticas de gestão e de engenharia de software com foco nos valores e princípios do *Lean* e do Ágil. O seu principal objetivo é entregar o produto que o cliente realmente deseja e que será útil e com qualidade. [2]

Os 12 princípios ágeis adaptam-se às características deste sistema em particular:

1. A maior prioridade é satisfazer o cliente através da entrega contínua e atempada de software com valor agregado.
2. Mudanças nos requisitos são bem-vindas, mesmo quando o desenvolvimento já se encontra em andamento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.

3. Entregar frequentemente o software a funcionar, de poucas semanas a poucos meses, com preferência da menor escala de tempo.
4. As equipas de negócios e dos desenvolvedores devem trabalhar diariamente em conjunto ao longo de todo o projeto.
5. Construção de projetos em torno de indivíduos motivados. Eles deverão encontrar-se num ambiente propício ao desenvolvimento criativo e ter o apoio necessário, de modo a que chefias demonstrem confiança para efetuar o seu trabalho.
6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipa de desenvolvimento é através de conversa presencial.
7. Software a funcionar é a medida crucial para o progresso.
8. Os processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e utilizadores devem ser capazes de manter um ritmo constante.
9. Contínua atenção à excelência técnica e bom design aumenta a agilidade.
10. Simplicidade (a arte de maximizar a quantidade de trabalho não realizado) é essencial.
11. As melhores arquiteturas, requisitos e designs emergem de equipas auto-organizáveis.
12. Em intervalos regulares, a equipa reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento. [2]

II.1.1.1. Metodologias XP

O XP usa uma abordagem orientada a objetos como seu paradigma de desenho. O processo é composto por quatro atividades: Planeamento, Projeto, Codificação e Teste, que são repetidas iteração a iteração.

Planeamento: É criado pelo cliente um conjunto de histórias que descrevem características e funcionalidades necessárias para o software ser construído. Cada história dá entrada no sistema de controlo da metodologia e é indexada. O cliente atribui-lhe um valor de prioridade. Os membros da equipa analisam esta lista e atribuem-lhe custos. Se a história precisar de mais tempo, pede-se ao cliente que a divida. Novas histórias podem ser adicionadas a qualquer momento. O passo seguinte é a equipa, em colaboração com o cliente, decidir que histórias vão ficar prontas na iteração seguinte e definir uma data.

Projeto: A filosofia inerente é KIS (keep it simple), é desencorajado o desenvolvimento de uma funcionalidade extra, porque o programador developer acha que mais tarde deve ser precisa. Frequentemente, geram-se protótipos, operacional de partes do projeto ou da totalidade. O XP encoraja a reconstrução, uma técnica de construção/projeto (é o processo de alterar e aperfeiçoar o sistema de software interno, sem que se altere o comportamento externo.)

Codificação: Antes do código, recomenda o processo, que se crie uma bateria de testes unitários para que a história fique satisfatória. Então, o foco dos programadores é a satisfação destes testes unitários. Para a codificação, o XP recomenda que esta seja feita em pares (duas cabeças trabalham melhor do que uma), isto garante outros aspectos como qualidade e rapidez (existe algum trabalho científico que comprava que o trabalhar em pares não prejudica o rendimento, pelo contrário, habitualmente consegue-se mais produtividade).

Teste: Os testes unitários são mantidos ao longo das várias iterações e passam a fazer parte de uma bateria de testes de regressão, que não são mais do que todos os testes unitários agrupados para serem testados periodicamente de uma vez em períodos curtos, pode ser de horas, ao final do dia ou no final da semana. A ideia é confirmar que nada deixou de funcionar. [5]

II.1.2. As 5 dimensões de um sistema de software

Há cinco dimensões que devem ser gerenciados num projeto de software: funcionalidades, qualidade, custo, tempo e equipa. Estas dimensões não são todas independentes. Por exemplo, aumentamos a equipa, o tempo pode ser reduzido (embora não necessariamente) e o custo pode aumentar. O equilíbrio mais comum é encurtar o tempo ou adicionar funcionalidades e sacrificar a qualidade. O equilíbrio entre essas cinco dimensões não é simples ou linear. Para cada projeto, precisamos decidir quais as dimensões que são críticas e como equilibrar as restantes, para que possamos atingir os objectivos-chave do projeto.

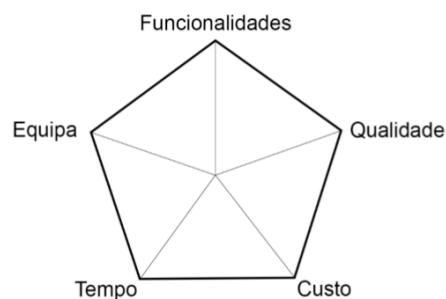


Figura 4 - As 5 dimensões de um projeto de software

Cada uma destas cinco dimensões pode ter um dos seguintes papéis num determinado projeto: restrição (constraint), controlador (driver), ou grau de liberdade (degree of freedom).

Uma restrição é um fator limitante que não está ao alcance do controlo do líder do projeto. Se uma equipa de tamanho fixo é atribuída a um projeto, esta torna-se uma restrição. O custo é uma restrição num projeto no âmbito de um contrato de preço fixo, enquanto a qualidade será um constrangimento para um projeto de desenvolvimento de software de uma peça de equipamento médico ou num sistema de controlo de voo de um avião.

Às vezes, é possível considerar o custo um constrangimento ou um condutor, isto porque poderia ser considerado um objetivo principal e um fator limitante. Da mesma forma, um conjunto de recursos específicos pode ser o principal condutor do projeto, mas também podem ser vistos como uma restrição se o conjunto de recursos não é negociável.

Qualquer dimensão de projeto pode não ser nem um condutor, nem uma restrição, neste caso torna-se um grau de liberdade. Estes são fatores de objetivos de um projeto. Por exemplo, em alguns projetos de sistemas de informação internos, os condutores são características e qualidade, e a equipa é a restrição, de modo que os graus de liberdade se tornam o tempo e o custo.

Um aspeto importante deste modelo não é saber quais as dimensões que serão restrições, controladores ou graus de liberdade, mas sim a negociação prévia das prioridades relativas de cada dimensão. Nem todas as cinco dimensões podem ser controladores ou restrições, este processo de negociação ajuda então a definir as regras e limites do projeto.

As dimensões são classificadas de acordo com a flexibilidade que o líder de projeto tem sobre estas. Uma restrição não brinda qualquer flexibilidade, um controlador oferece pouca flexibilidade, mas um grau de liberdade fornece flexibilidade suficiente para equilibrar com as excedentes dimensões. Esta classificação é feita numa escala de 0 a 10, em que 10 significa totalmente flexível e 0 completamente restringido. O resultado é representado graficamente num diagrama de “*Kiviat*”, onde cada eixo representa o nível de flexibilidade que o líder de projeto tem sobre a dimensão a que corresponde e, por este último motivo, são chamados diagramas de flexibilidade. [48]

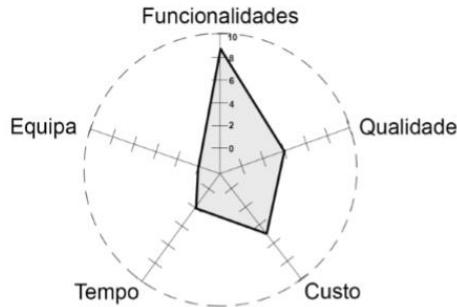


Figura 5 - Exemplo de diagrama de flexibilidade

A forma do polígono resultante apresenta a indicação visual dos aspetos essenciais de cada projeto, tal como exemplificado na figura 5. Ao redefinir uma dimensão, alterando o seu valor no eixo, obriga a um eventual ajuste compensatório das restantes dimensões. [48]

Deste modo, assumindo o papel de líder de projeto, exponho na figura 6, o diagrama de flexibilidade do sistema Academic-Manager. É possível observar que a equipa era a principal restrição, o custo era pouco flexível, porque utilizaria sempre ferramentas e tecnologias com poucos custos adicionais, o tempo de entrega era outra dimensão pouco flexível. Quanto às funcionalidades, tinha liberdade de selecionar as que pretendia desenvolver.

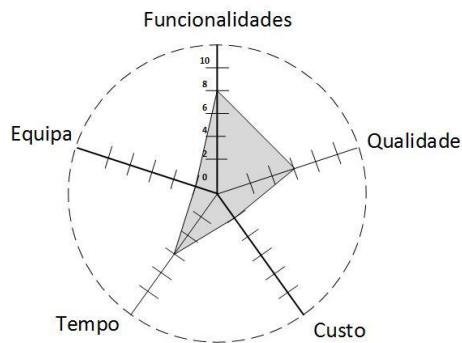


Figura 6- Diagrama de Flexibilidade de Academic-Manager

II.1.3. Ferramentas utilizadas no desenvolvimento do software

Para conseguir desenhar e desenvolver o sistema com sucesso e maior facilidade, recorri ao uso de algumas ferramentas que ajudariam significativamente o cumprimento dos meus objetivos.

II.1.3.1.phpMyAdmin

phpMyAdmin é uma ferramenta de software livre escrito em PHP, destinado a lidar com a administração do MySQL pela Web. phpMyAdmin suporta uma ampla gama de operações em MySQL e MariaDB. Operações (gestão de bases de dados, tabelas, colunas, relações, índices, usuários, permissões, entre outros) podem ser realizada através da interface do utilizador, enquanto ainda se tem a capacidade de executar diretamente qualquer instrução SQL.

Características:

- Interface web intuitiva;
- Suporte para a maioria dos recursos do MySQL;
- Importa dados de CSV e SQL;
- Exportar dados para vários formatos: CSV, SQL, XML, PDF, ISO / IEC 26300 - Texto OpenDocument e Spreadsheet, Palavra, LATEX e outros;
- Administra vários servidores;
- Cria gráficos do layout do seu banco de dados em vários formatos;
- Cria consultas complexas usando consulta por exemplo (QBE);
- Pesquisa globalmente numa base de dados ou um subconjunto dela;
- Transforma dados armazenados em qualquer formato, utilizando um conjunto de funções pré-definidas, como exibir BLOB-dados como uma imagem ou fazer o download-link [10]



Figura 7 - Logotipo de phpMyAdmin

II.1.3.2.MySQL Workbench

MySQL Workbench é uma ferramenta visual unificada para desenho de base de dados, desenvolvedores e DBAs. MySQL Workbench fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para configuração do servidor, administração de usuários, backup e muito mais. Esta ferramenta está disponível no Windows, Linux e Mac OS X. [11]



Figura 8 - Logotipo de MySQL Workbench

II.1.3.3.PHPStorm

Assistência de codificação inteligente: PhpStorm é um IDE PHP que realmente “recebe” o código. Suporta PHP 5,3/5,4/5,5/5,6/7,0, oferece prevenção on-the-fly de erro, auto-completa e refatoriza de código, depuração de configuração zero, e um HTML estendido, CSS e JavaScript editor.

Editor de código PHP inteligente: O IDE fornece conclusão de código inteligente, destaque de sintaxe, configuração de formatação código estendido, on-the-fly verificação de erros, dobramento de código, suporta misturas de idiomas e muito mais. Refatorações automatizadas que tratam o seu código com cuidado, ajudando a tornar as configurações globais do projeto facilmente e com segurança.

Análise de Qualidade de Código: Centenas de inspeções de código verificam o seu código à medida que escreve e inspecionam todo o projeto procurando possíveis erros. Soluções rápidas e eficientes para a maioria das inspeções, torna mais fácil a correção ou alteração do código de forma instantânea.

Fácil Código Navegação e Pesquisa: PhpStorm ajuda a obter em torno de seu código mais eficiente e poupar tempo quando se trabalha com grandes projetos.

Depuração, testes e Profiling. PhpStorm fornece poderosas ferramentas embutidas para depuração, testes e perfis das suas aplicações.

Depuração: depuração Zero-configuração faz com que seja muito fácil de depurar suas aplicações PHP. Além disso, PhpStorm oferece inúmeras opções para depurar seu código PHP com o Visual Debugger, para que possa: inspecionar variáveis e relógios definidos pelo usuário, definir pontos de interrupção e avaliar uma expressão em tempo de execução, aplicações remotas de depuração, depurar uma página em várias sessões simultaneamente.

Testes: é possível desenvolver PHPUnit testes no PhpStorm e executá-los instantaneamente a partir de um diretório, arquivo ou classe. Cobertura de código de PHPUnit mostra o quanto do código é submetido a testes.

Profiling: é possível obter um perfil das aplicações com Xdebug ou Zend Debugger e verificar relatórios agregados em PhpStorm [12]



Figura 9 - Logotipo de PhpStorm

II.1.3.4.Trello

Trello é uma aplicação web para fazer listagens dentro de listagens. O que a torna único é a sua versatilidade: cada item em uma lista pode ser adicionado a todas as outras listas, imagens, vídeos, documentos, etc. Além disso, é extremamente poderosa para uso colaborativo. Pode ser adicionado qualquer número de utilizadores, designar tarefas, colocar datas limite, tem o seu próprio calendário que pode sincronizar com o Google.

Trello baseia-se no método Kanban para gestão de projetos, com cartões que viajam por diferentes tarefas em função do seu estado: assim, costumamos ter uma lista de coisas por fazer (*to do*, ou pendentes), coisas que se estão a ser feitas (*doing*, ou em processo), ou que já foram feitas (*done*, ou terminadas) [13]

Através deste programa consegui organizar as tarefas que tinha por fazer, impor metas semanais e ter uma visão geral e detalhada da evolução do projeto.

Uma das vantagens desta ferramenta é que não é necessário apagar nada, quando finalizava uma tarefa, arquivava esse cartão e desaparecia da listagem, mas em qualquer momento era possível aceder aos cartões arquivados.



Figura 10 - Logotipo de Trello

II.1.3.5. Git

Git é um sistema de controlo de versões cuja diferença dos restantes, é armazenar e pensar sobre a informação de uma forma global. A maior distinção entre Git e qualquer outro VCS (Subversion e similares inclusos) está na forma como o Git trata os dados. Conceitualmente, a maior parte dos outros sistemas armazena informação como uma lista de mudanças por arquivo. Esses sistemas (CVS, Subversion, Perforce, Bazaar, etc.) tratam a informação que mantém como um conjunto de arquivos e as mudanças feitas a cada arquivo ao longo do tempo. Git considera que os dados são como um conjunto de snapshots (captura de algo num determinado instante, como numa foto) de um minissistema de arquivos. Cada vez que se salva ou consolida (commit) o estado do projeto no Git, é como se ele tirasse uma foto de todos os seus arquivos naquele momento e armazenasse uma referência para essa captura. Para ser eficiente, se nenhum arquivo foi alterado, a informação não é armazenada novamente, apenas um link para o arquivo idêntico anterior que já foi armazenado. [14]

Com a utilização de Git, consegui criar um controlo das versões do Academic-Manager e cópias de segurança.



Figura 11 - Logotipo de Git

II.1.3.6. Bitbucket:

Bitbucket é um serviço de alojamento baseado na web para os projetos que utilizam o sistema de controlo de versões Mercurial e Git. Bitbucket tem um serviço grátis e um comercial. Com esta ferramenta é exequível criar repositórios privados. É possível criar um número ilimitado de repositórios privados, são grátis enquanto tiver cinco ou menos usuários. [31]



Figura 12 - Logotipo de Bitbucket

II.1.4. Projetos semelhantes

No mercado, existem alguns sistemas utilizados para a gestão escolar, entre os quais podemos encontrar.

II.1.4.1. Alunos

É um sistema de gestão escolar que contempla o ensino regular, recorrente e profissional para Básico e Secundário, incluindo o 1º Ciclo. Entre as suas funcionalidades encontramos as avaliações, assiduidade, matrículas, estatísticas de sucesso/insucesso escolar, comunicações aos encarregados de educação, exames do básico e ligação ao ENES e ENEB.

Este sistema permite a impressão de mapas e listagens, das quais destacamos: Pautas de Avaliação, Fichas Informativas, Termos (9º ano), Registos Biográficos, Extratos de Assiduidade, Certificados, Gráficos de Sucesso/ Insucesso Escolar, Listas de Turma, Impressos de Matrícula, Listagens diversas. Software certificado pelo Gabinete Coordenador do Sistema de Informação do Ministério da Educação (MISI). [49]

II.1.4.2. eSchooling

O E-Schooling Server é um sistema de informação escolar desenvolvido para o ensino público e privado com capacidades para gerir todas as tipologias de ensino.

Desenvolvido através das mais modernas tecnologias, utilizando motores de Bases de Dados robustos e fiáveis, está completamente adaptado à nova legislação, permitindo abordar e gerir através do mesmo sistema o Ensino Profissional, Cursos CEF e Ensino Regular, entre outros.

Permite de uma forma simples e através de uma interface agradável toda a gestão escolar de uma escola profissional ou regular, abordando entre outros os seguintes temas: Livro de Ponto digital, Gestão de Cursos, Gestão de Alunos, Gestão de Faltas, Gestão de Docentes e Funcionários, Gestão de Horários e Exceções de Horário. [50]

II.1.4.3. praxis

praxis – Sistema de Gestão Escolar desenhado para os ensinos básico, secundário e superior. Totalmente web e composto por diferentes portais, permitindo uma utilização por diferentes entidades – Escola, Docente, Encarregado de Educação e Aluno – para uma melhor Gestão Administrativa, Financeira e de Recursos.

Através deste sistema é possível fazer matrículas, organizar os recursos humanos, otimizar as infraestruturas, disponibilizar informação, divulgar eventos, comunicar à distância, lançar notas e sumários, consultar horários e a assiduidades dos alunos e submeter pedidos pessoais à administração.

Este sistema conta com um módulo para os encarregados de educação, nos quais é possível seguir o percurso do aluno, contatar docentes e serviços administrativos, fazer pagamentos, consultar atividades ou apresentar justificações. [51]

II.1.4.4. Inovar alunos

O inovar alunos é uma aplicação adaptada a cada nível de ensino: Creche, Pré-Escolar, Básico, Secundário, CEF, Vocacional, EFA, Profissional, Recorrente, DREF – Profijj, DREF – Unecas, DREF – Oportunidade.

Esta aplicação, entre muitas coisas, permite realizar sumários, coordenação de Turma, gestão de cursos, avaliações, entre outros. Entre as características que distinguem inovar alunos de outros sistemas gestores académicos podemos encontrar as seguintes:

- Divide-se em três grandes áreas: Administrativa, Docente e Gestão;
- Os docentes podem consultar informação atualizada sobre toda a atividade das turmas;
- Possibilidade de registo de sumários de aula, atividades não letivas e substituições;
- Qualquer professor pode fazer o lançamento das suas propostas de avaliação, de forma contínua;
- Acompanhamento e elaboração de relatórios para crianças com necessidades educativas especiais;
- Redação, envio e/ou impressão de atas de reuniões, assim como, de relatórios;

- Envio de notificações, emails, SMS para docentes, alunos e encarregados de educação;
- Tratamento, análise e filtragem de dados sobre os alunos, em tempo real, com possibilidade de exportação para diferentes formatos. [52]

Na seguinte tabela, é possível observar as funcionalidades pedidas pelo cliente Vs sistemas no mercado.

| Requisitos do cliente/Sistemas no mercado | Alunos | eSchooling | praXis | Inovar alunos |
|--|--------|------------|--------|---------------|
| Adicionar, editar, gerir e/ou remover anos letivos. | X | X | X | X |
| Adicionar, editar e/ou remover utilizadores. | X | X | X | X |
| Adicionar, editar e/ou remover anos escolares. | X | X | X | X |
| Adicionar, editar e/ou remover cursos. | X | X | X | X |
| Adicionar, editar e/ou remover disciplinas. | X | X | X | X |
| Adicionar, editar, gerir e/ou remover turmas. | X | X | X | X |
| Adicionar, editar e/ou remover serviços. | -- | -- | -- | -- |
| Adicionar, editar e/ou remover atividades extracurriculares. | -- | -- | -- | -- |
| Adicionar, editar e/ou remover alunos. | X | X | X | X |
| Fazer a inscrição no colégio através do sistema. | X | X | X | X |
| Fazer a renovação de matrícula no colégio. | X | X | X | X |
| Inscrição nas diferentes atividades extracurriculares. | -- | -- | -- | -- |
| Aquisição serviços através do sistema. | -- | -- | -- | -- |
| Criar listagens dos alunos por atividade extracurricular. | -- | -- | -- | -- |
| Criar listagens dos alunos por serviço. | -- | -- | -- | -- |
| Criar listagens dos alunos por turma. | X | X | X | X |
| Criar listagens dos alunos por disciplinas. | X | X | X | X |
| Criar listagens dos alunos por curso. | X | X | X | X |
| Criar listagens dos alunos por ano escolar. | X | X | X | X |
| Criar listagens dos alunos por ciclo de ensino. | -- | X | X | X |
| Visualizar, preencher, adicionar, modificar e/ou eliminar os parâmetros de todos os tipos de avaliações. | -- | -- | -- | -- |
| Gerar automaticamente os registo de avaliação de cada um dos alunos em PDF. | X | X | X | X |
| Gerar automaticamente todas as pautas necessárias. | X | X | X | X |
| Gerar automaticamente os alunos destacados nos quadros de honra e de excelência. | -- | X | X | X |
| Gerar automaticamente o currículo académico dos alunos em PDF. | -- | -- | -- | -- |

Tabela 1 - Requisitos Vs Sistemas no mercado

II.2. CONCLUSÃO

Em suma, neste capítulo, foi possível observar a importância das diferentes metodologias de trabalho e ferramentas de apoio utilizadas na realização deste projeto, assim como os projetos semelhantes que há no mercado.

A metodologia Ágil permitiu flexibilizar o desenvolvimento do software, diminuir a burocracia, tornar o desenvolvimento mais rápido e possibilitar respostas céleres às mudanças. Por sua vez, a metodologia XP facilitou a criação de um sistema de melhor qualidade, em menos tempo e de forma mais económica que o habitual. Estes objetivos são alcançados através de um pequeno conjunto de valores, princípios e práticas que diferem, essencialmente, da forma tradicional de se desenvolver software.

As ferramentas utilizadas no desenvolvimento do projeto, como o phpMyAdmin, MySQL Workbench, PHPStorm, Trello, Git e Bitbucket, permitiram lidar com a administração da base de dados, desenhar a base de dados, editar o código fonte, organizar as tarefas por fazer, controlar as versões do projeto e criar repositórios privados, respetivamente. Sem dúvida alguma, estas ferramentas facilitaram o cumprimento dos objetivos, assim como permitiram uma maior organização.

No fim deste capítulo, foi feita uma análise dos diferentes projetos semelhantes ao projeto por mim desenvolvido presentes no mercado. Estes sistemas não cumprem inteiramente com as principais funcionalidades pedidas pelo cliente.

Ao criar um sistema de raiz, é possível desenvolver soluções personalizadas, à medida das verdadeiras necessidades encontradas, adaptado aos requisitos específicos, software desenhado que vai ao encontro do requerido, e que solucione os problemas existentes. Da mesma forma, possibilita desenvolver novas funcionalidades à medida que estas se tornam necessárias, sem contar que, neste caso, o meu projeto, tendo em conta as suas especificidades, era uma solução mais económica, tendo em conta as presentes no mercado, o que para a empresa foi uma mais valia. O facto de eu estar presente no local, onde o projeto foi implementado, também foi um aspeto bastante positivo, pois facilitava a comunicação entre os diversos utilizadores, fazendo constante diagnose da evolução da sua aplicação, conseguindo fornecer uma resposta mais pronta aquando do surgimento do problema.

III. DESENVOLVIMENTO DO SISTEMA

III.1. CICLO DE VIDA DE UM SISTEMA DE SOFTWARE

O processo de desenvolvimento de software é o conjunto de atividades que constituem o progresso de um sistema. As atividades são agrupadas em etapas como: a definição de requisitos, análise, projeto, desenvolvimento, teste e implementação.

A sequência e a presença de cada uma destas atividades depende da escolha de um modelo de ciclo de vida entre o cliente e a equipa de desenvolvimento.

Não existe um modelo ideal. O perfil e complexidade do negócio do cliente, o tempo disponível, o custo, a equipa, o ambiente operacional são fatores que influenciarão diretamente na escolha do ciclo de vida de software a ser adotado.

Pelas características do projeto, foi implementado o modelo em espiral que foi escolhido, entre outros modelos, sendo este o mais adequado.

III.1.1. Modelo espiral

O modelo em espiral combina as iterações do modelo de prototipagem com o controle e sistematização do modelo em cascata. Assim sendo, cada ciclo do modelo em espiral possui quatro etapas principais:

- Planeamento: onde se determinam os objetivos, alternativas e restrições ao software.
- Análise de riscos: onde as alternativas são analisadas e os riscos são identificados /resolvidos.
- Engenharia: onde acontece o desenvolvimento do produto ao nível seguinte.
- Atualização: onde se planeia o ciclo seguinte.

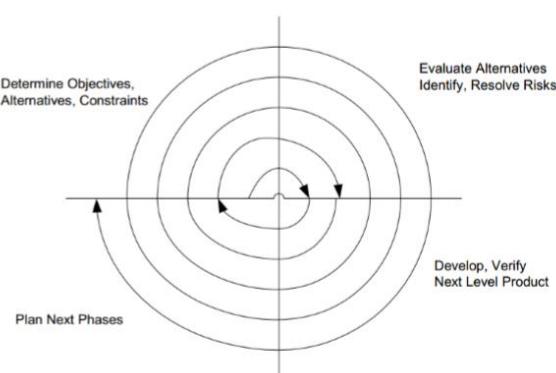


Figura 13 - Modelo espiral do ciclo de vida de um sistema

A cada ciclo, o modelo espiral gera um protótipo ligeiramente diferente do anterior sendo uma versão mais sofisticada do software, na medida em que não há etapas fixas no modelo espiral, isto é, cada fase ocorre mais do que uma vez. O raio da espiral representa o custo do processo e o ângulo da espiral representaria o progresso realizado na construção do software.

O processo espiral não termina após a entrega do software, como nos outros modelos, podendo ser adaptado para uma aplicação posterior, como um projeto de aperfeiçoamento do produto entregue. [53] e [54].

Neste capítulo, será apresentado o processo de desenvolvimento do sistema, isto é, os requisitos funcionais e não funcionais do sistema, casos de utilização, arquitetura implementada, diagramas de entidade e relação, interfaces do utilizador, a implementação, funcionalidades implementadas, software *testing* e, finalmente, a conclusão.

III.2. REQUISITOS

Nesta secção, serão apresentados os requisitos funcionais e não funcionais do sistema *Academic-Manager*.

Estes requisitos foram obtidos através do levantamento de informação, realizado mediante entrevistas às pessoas responsáveis pelas diferentes áreas; acesso e exploração do sistema informática utilizado pelos funcionários para o lançamento de notas. Estive, também, presente num conselho de turma de avaliação intercalar e de avaliação periódica do primeiro período de cada um dos ciclos (1º, 2º, 3º ciclos e secundário) e tive acesso a documentos internos de avaliação. Tive, ainda, a oportunidade de presenciar a inscrição de um aluno pelo seu encarregado de educação no colégio e participar, de forma não ativa, nos procedimentos seguintes.

Depois da análise da informação recolhida, foram listados os requisitos funcionais e não funcionais do sistema a ser desenvolvido.

III.2.1.Requisitos funcionais

RF1. Os utilizadores devem ser capazes de adicionar, editar, gerir e/ou remover anos letivos.

RF2. Os utilizadores devem ser capazes de adicionar, editar e/ou remover utilizadores.

RF3. Os utilizadores devem ser capazes de adicionar, editar e/ou remover anos escolares.

RF4. Os utilizadores devem ser capazes de adicionar, editar e/ou remover cursos

RF5. Os utilizadores devem ser capazes de adicionar, editar e/ou remover disciplinas.

RF6. Os utilizadores devem ser capazes de adicionar, editar, gerir e/ou remover turmas.

RF7. Os utilizadores devem ser capazes de adicionar, editar e/ou remover serviços.

RF8. Os utilizadores devem ser capazes de adicionar, editar e/ou remover atividades extracurriculares

RF9. Os utilizadores devem ser capazes de adicionar, editar e/ou remover alunos.

RF10. Os utilizadores devem ser capazes de fazer a inscrição no colégio através do sistema

RF11. Os utilizadores devem ser capazes de fazer a renovação de matrícula no colégio.

RF12. O sistema deve permitir que os utilizadores realizem a inscrição nas diferentes atividades extracurriculares

RF13. O sistema deve permitir que os utilizadores adquiriram serviços através do sistema

RF14. Os utilizadores devem ser capazes de criar listagens dos alunos por atividade extracurricular

RF15. Os utilizadores devem ser capazes de criar listagens dos alunos por serviço.

RF16. Os utilizadores devem ser capazes de criar listagens dos alunos por turma

RF17. Os utilizadores devem ser capazes de criar listagens dos alunos por disciplinas

RF18. Os utilizadores devem ser capazes de criar listagens dos alunos por curso.

RF19. Os utilizadores devem ser capazes de criar listagens dos alunos por ano escolar.

RF20. Os utilizadores devem ser capazes de criar listagens dos alunos por ciclo de ensino.

RF21. A aplicação deverá permitir aos utilizadores visualizar, preencher, adicionar, modificar e/ou eliminar os parâmetros de todos os tipos de avaliações

RF22. O sistema deve gerar automaticamente os registos de avaliação de cada um dos alunos em PDF

RF23. O sistema deve gerar automaticamente todas as pautas necessárias.

RF24. O sistema deve gerar automaticamente os alunos destacados nos quadros de honra e de excelência.

RF25. O sistema deve gerar automaticamente o currículo académico dos alunos em PDF

III.2.2.Requisitos não funcionais.

Requisitos de Produto

Requisitos de usabilidade

RU1. A interface gráfica deverá ser fácil de utilizar.

RU2. A interface gráfica deverá ser simples.

RU3. A interface gráfica deverá ser intuitiva.

Requisitos de fiabilidade

RF1. O sistema não deverá crashar por razões inerentes a este

Requisitos de desempenho

RDE1. Qualquer página da aplicação não deve demorar em média mais do que 5seg. a ser carregada

Requisitos de robustez

RR1. A aplicação recupera de uma forma controlada da ocorrência de erros.

Requisitos Organizacionais

Requisitos de Desenvolvimento

RD1. A aplicação deverá ser implementada usando: Apache, HTML, PHP, MySQL, CSS, JavaScript e Jquery.

RD2. A aplicação deverá ser implementada seguindo a arquitetura cliente-servidor.

RD3. A aplicação deverá ser implementada seguindo o padrão de desenho MVC.

Requisitos Externos

Requisitos de Segurança

RS1. A aplicação não deverá permitir o acesso de nenhum usuário sem primeiro realizar o login.

III.3. CASOS DE UTILIZAÇÃO

Os diagramas de caso de utilização documentam o que o sistema faz do ponto de vista do usuário. Descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo. Além disto, permite capturar os requisitos de sistemas sob desenho ou consideração, descrever a funcionalidade fornecida por esses sistemas e determinar os requisitos dos sistemas colocam em seu ambiente. [19] e [20] (descrever a funcionalidade fornecida por esses sistemas e determinar os seus requisitos inseridos no seu ambiente)

Academic-Manager é constituído por um módulo principal (módulo de inscrição) e quatro submódulos (atividades extracurriculares, serviços, lançamento de notas e currículo académico). Por este motivo, e para melhor entendimento, apresento os diagramas de caso de utilização por módulos.

III.3.1. Caso de utilização do módulo de inscrição

No modulo (módulo) de inscrição os atores que interagem com o sistema são os seguintes:

Encarregado de educação: um encarregado de educação que pretenda fazer a matrícula no Colégio.

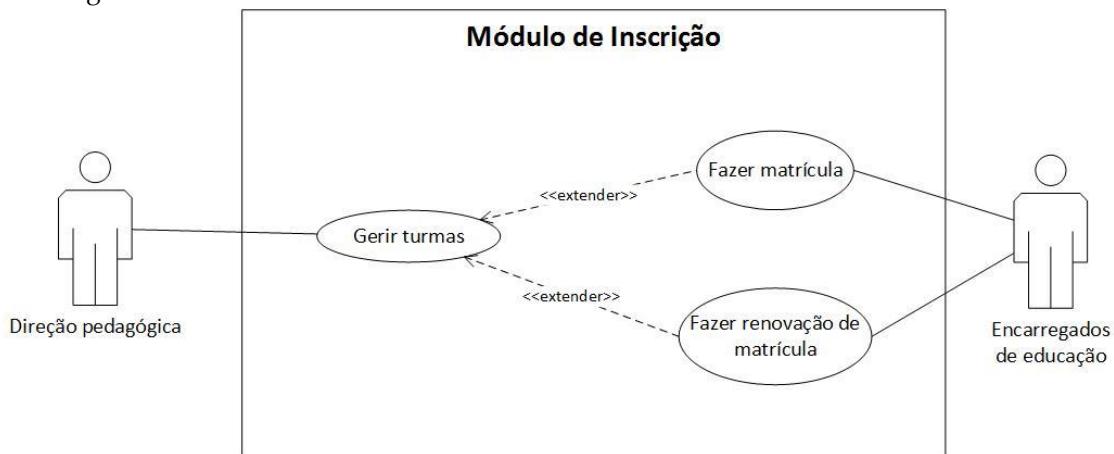


Figura 14 - Diagrama de casos de utilização do módulo de inscrição

Direção pedagógica: a Direção do Colégio que vai ser a encarregue de gerir a inscrição do aluno no sistema.

III.3.2.Caso de utilização do módulo de atividades extracurriculares

No módulo de atividades extracurriculares, os atores que interagem com o sistema são os seguintes:

Direção: um colaborador do Colégio com privilégios de direção vai ser o encarregue de gerir as atividades extracurriculares que o colégio dispõe.

Encarregado de educação: um encarregado de educação que pretenda que o seu educando frequente alguma atividade extracurricular.

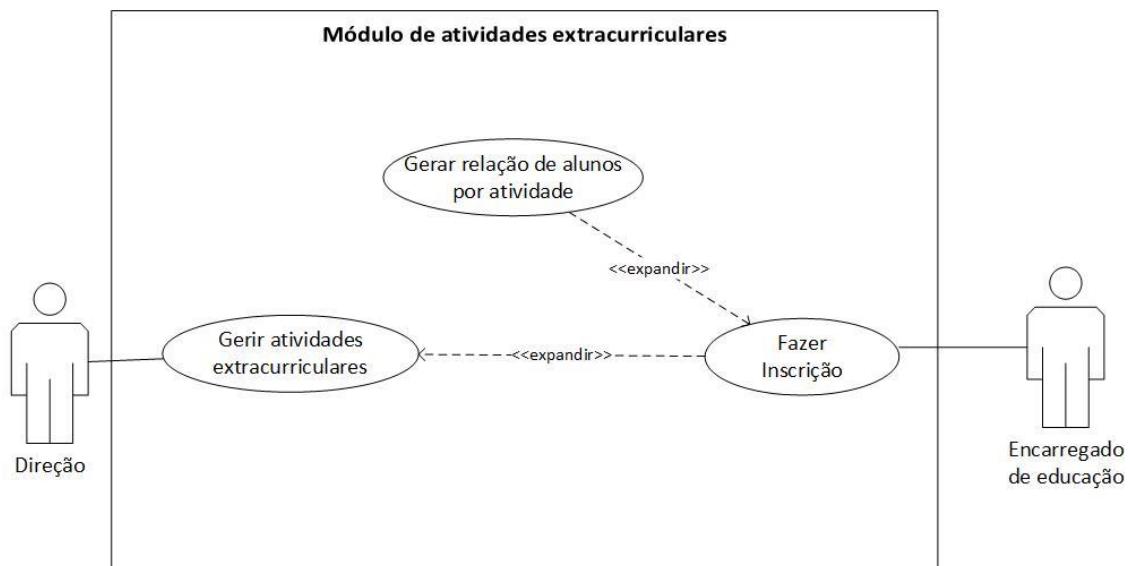


Figura 15 - Diagrama de casos de utilização do módulo de atividades extracurriculares

III.3.3.Caso de utilização do módulo dos serviços

No módulo dos serviços, os atores que interagem com o sistema são os seguintes:

Direção: um colaborador do colégio com privilégios de direção é que vai gerir os serviços no sistema

Encarregado de educação: um encarregado de educação que pretenda que o seu educando usufrua de algum serviço que o colégio oferece

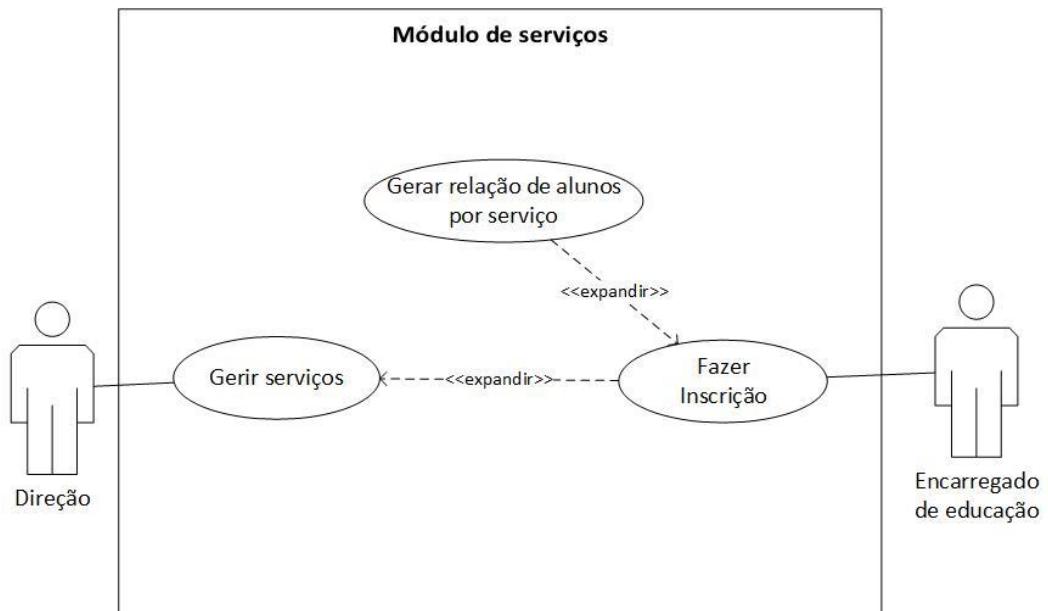


Figura 16 - Diagrama de caso de utilização do módulo de serviços

III.3.4.Caso de utilização do módulo de lançamento de notas

No módulo de lançamento de notas, os atores que interagem com o sistema são o professor titular ou diretor de turma e os professores de cada uma das disciplinas, após o preenchimento dos dados, o sistema gera os documentos necessários para finalizar este processo.

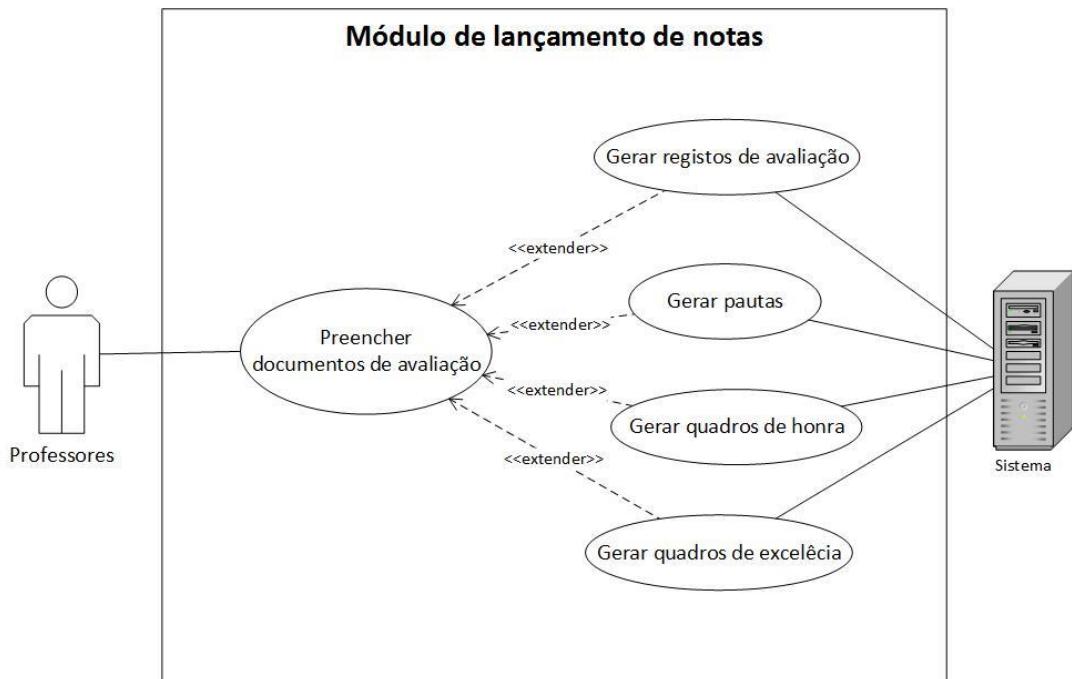


Figura 17 - Diagrama de casos de utilização do módulo de lançamento de notas

III.3.5.Caso de utilização do módulo de currículo académico

No módulo de currículo académico, o ator que interage com o sistema é o professor titular ou diretor de turma preenchendo os dados necessários para o sistema gerar automaticamente os currículos académicos.

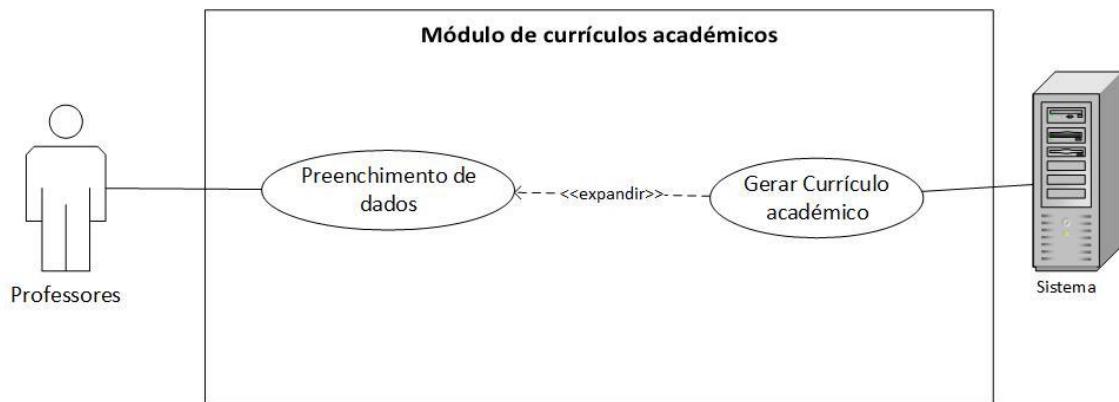


Figura 18 - Diagrama de casos de utilização do módulo de currículos académicos

III.4. ARQUITETURA

“A arquitetura de *software* de um programa ou sistema computacional é a estrutura ou estruturas do sistema, as quais incluem os elementos de *software*, as propriedades externamente visíveis desses elementos e as relações entre eles. Exatamente que estruturas considerar e as formas de representar variam de acordo com os objetivos de engenharia.” [21]

A arquitetura não é o *software* operacional, mas sim, uma representação que permite analisar a efetividade do projeto no atendimento dos requisitos, considerar alternativas de arquitetura na hora de realizar mudanças no projeto e minimizar os riscos associados à construção de um software.

Existem três razões chave pelas quais a arquitetura de *software* é importante:

- As representações da arquitetura facilitam a comunicação entre as partes interessadas no desenvolvimento do sistema;
- Evidencia decisões de projeto iniciais que terão profundo impacto em todo o trabalho de engenharia de *software* que se segue depois e, tão importante quanto, no sucesso final do sistema como uma entidade operacional;
- A arquitetura constitui um modelo relativamente pequeno e intelectualmente compressível de como o sistema é estruturado e como os seus componentes trabalham em conjunto. [22]

III.4.1. Estilo arquitetural – Cliente-Servidor

Numa arquitetura cliente-servidor, a funcionalidade do sistema está organizado em serviços, com cada serviço entregue a partir de um servidor separado. Os clientes são os utilizadores desses serviços e servidores de acesso para fazer uso deles.

Um sistema que segue o padrão cliente-servidor é organizado como um conjunto de serviços e servidores associados e os clientes que acedem e usam os serviços. Os componentes principais deste modelo são:

1. Um conjunto de servidores que oferecem serviços a outros componentes. Exemplos de servidores incluem servidores de impressão que oferecem serviços de impressão, servidores de arquivos que oferecem serviços de gestão de arquivos e um servidor de compilação, que oferece serviços de compilação de linguagens de programação.

2. Um conjunto de clientes que recorre aos serviços oferecidos pelos servidores. Não será normalmente várias instâncias de um programa de cliente executando simultaneamente em diferentes computadores.
3. Uma rede que permite que os clientes accedem a esses serviços. A maioria dos sistemas cliente-servidor são implementados como sistemas distribuídos, conectados usando protocolos de Internet. [24]

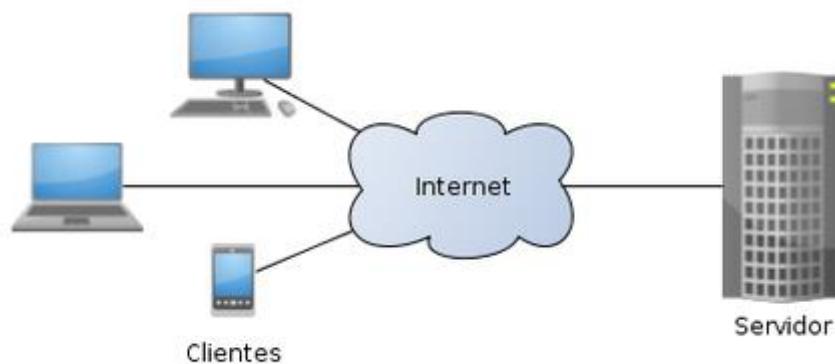


Figura 19 - Arquitetura Cliente-Servidor

III.4.2. Padrão arquitetural

Os padrões arquiteturais são formulários que solucionam problemas arquiteturais recorrentes. Uma estrutura arquitetural é um conjunto de componentes nos quais é possível construir um determinado tipo de arquitetura. Muitas das maiores dificuldades arquiteturais devem ser resolvidas na estrutura geralmente direcionadas a um domínio específico.

Os padrões arquiteturais podem ser agrupados por categorias de acordo com as características do sistema

- De acordo com a estrutura: os padrões de esta categoria ajudam a evitar um “mar” de componente ou objetos. Suportam uma descomposição controlada de uma tarefa geral do sistema em subtarefas cooperantes. Esta categoria inclui o padrão de camadas, pipes e filtros e o padrão *Blackboard*;
- Sistemas distribuídos: esta categoria inclui o padrão Broker, este padrão fornece uma infraestrutura completa para arquiteturas distribuídas;
- Sistemas interativos: esta categoria comprehende dos padrões, MVC (*Model View Controller*) e *Presentation-Abstraction-Control*. Ambos os padrões apoiam a estrutura dos sistemas de software que oferece iteração humano-computador;

- Sistemas adaptáveis: padrão *Reflection* e o padrão *Microkernel* apoiam fortemente a extensão de aplicações e a sua adaptação à evolução das tecnologias e mudanças dos requisitos funcionais. [23]

Academic Manager é um sistema que deseja permitir um alto grau de iteração com o utilizador através das interfaces gráficas com o objetivo de facilitar o uso da aplicação, mas, ao mesmo tempo, manter o núcleo funcional independente da interface do utilizador. Isto exige escolher uma arquitetura que suporte a adaptação das partes da interface do usuário, sem causar grandes efeitos para a funcionalidade específica do sistema ou o modelo de dados subjacente ao software. Por este motivo, o padrão escolhido foi o MVC.

III.4.2.1.MVC (*Model View Controller*)

O padrão de arquitetura MVC (*Model View Controller*) divide um sistema em três componentes lógicos que interagem uns com os outros. O modelo gera os dados do sistema e as operações associadas sobre esses dados. O componente Vista define e gera o modo como os dados são apresentados ao usuário. O componente controlador gera interação do utilizador [24]

Utilizar o padrão MVC num sistema traz benefícios:

- Múltiplas vistas com o mesmo modelo: MVC separa estritamente o modelo dos componentes da interface do utilizador, portanto, muitas vistas podem usar o mesmo modelo;
- Vistas sincronizadas: o mecanismo de propagação de mudanças do modelo garante que todos os utilizadores indexados são notificados das alterações nos dados da aplicação no momento correto. Isto sincroniza dependentes todas as vistas e controladores;
- Vistas e controladores conectáveis: a separação conceptual de MVC permite trocar os objetos de um modelo de vista e controlador. Objetos da interface do utilizador podem ser substituídos inclusive no tempo de execução.
- Permutabilidade de "aparência": porque o modelo é independente de todos os códigos de interface de utilizador, um porto de uma aplicação MVC para uma nova plataforma não afeta o núcleo funcional da aplicação.

- Framework potencial: é possível basear uma estrutura de *framework* sobre este padrão. [23]

A figura 20 explica o funcionamento do padrão MVC passo a passo:

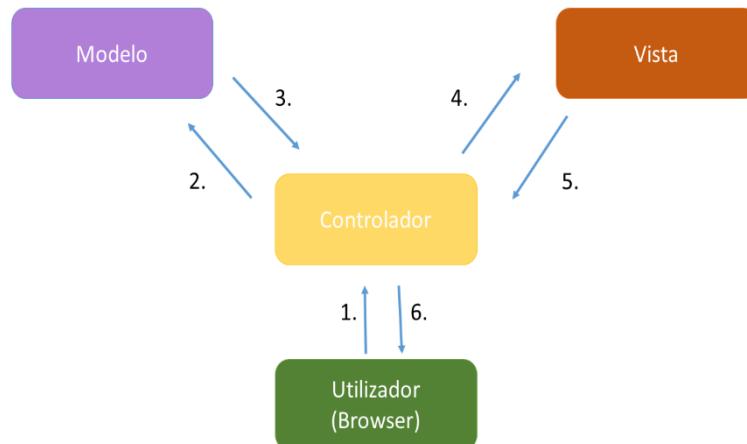


Figura 20 - Representação gráfica do padrão arquitetural MVC

1. O utilizador envia um pedido ao controlador pelo URL de um browser;
2. O controlador solicita ao modelo os dados;
3. O modelo envia os dados;
4. O controlador seleciona uma vista para mostrar os dados;
5. É devolvido a vista selecionada ao controlador;
6. O controlador devolve uma vista que carrega os dados do modelo selecionado.

III.5. DIAGRAMA DE ENTIDADE-RELAÇÃO

Um diagrama de entidade-relação é uma técnica de modelagem de dados que cria uma representação das entidades de um sistema de informação e as relações entre as entidades. Há três ingredientes num diagrama de entidade-relação.

- As entidades: representam pessoas, lugares, elementos ou conceitos;
- Os atributos: representam propriedades as qualidades descritivas de uma entidade, também são conhecidos por elementos de dados;
- Relações: representa as ligações entre as entidades. [39]

Através deste diagrama obtemos o desenho da base de dados do sistema Academic-Manager representado pela figura 21.

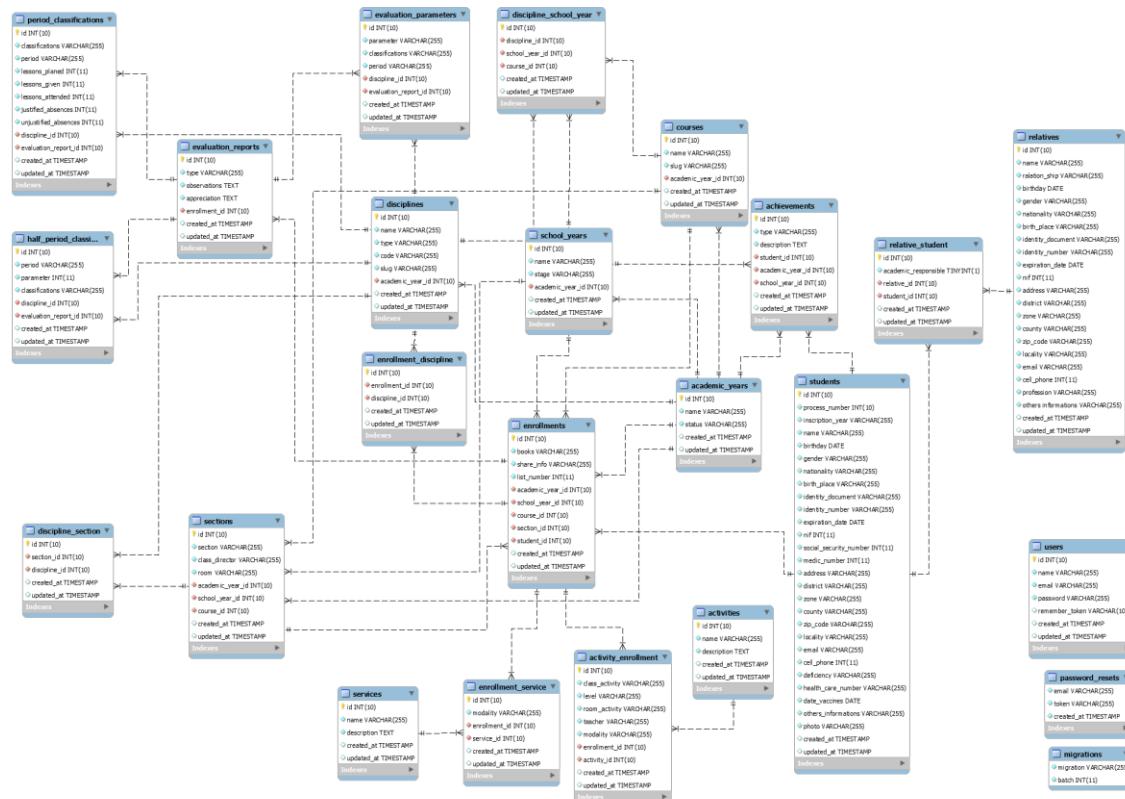


Figura 21 - Diagrama Entidade-Relação

Como o sistema foi dividido em módulos e para melhor compreensão, passo a explicar o diagrama de entidade e relação de cada módulo.

III.5.1. Módulo de inscrição:

Módulo principal do sistema. No módulo principal do sistema encontramos as seguintes entidades representadas na figura 22:

- Academic years (anos letivos): contém informação dos anos letivos. Esta entidade servirá de referência (referência) para armazenar informação relativas aos anos letivos.
- School years (anos escolares): contém informação dos anos escolares que estão disponíveis para fazer a matrícula.
- Courses (Cursos): entidade que armazena informações sobre o curso que o estabelecimento de ensino dispõe.
- Courses- School years (Cursos-anos escolares): Relação de muito-para-muitos entre a entidades “Courses” (Cursos) e “School years (anos escolares)” .
- Disciplines (Disciplinas): contém informação referente as disciplinas que estão disponíveis no estabelecimento de ensino.
- Sections (turmas): contém os dados de uma turma, como por exemplo, o diretor de turma, a sala a que ano letivo pertence, ano escolar e curso.
- Students (alunos): contém os dados dos alunos necessários para realizar a inscrição no sistema.
- Relatives (parentes): contém os dados dos parentes necessários para realizar a inscrição no sistema.
- Relatives-students (parentes-alunos): relação de muito-para-muitos entre a entidades “Relatives (parentes)” e “Students (alunos)”, além de identificar qual dos parentes é o encarregado de educação do aluno em causa.
- Enrollments (Matrículas): contém toda a informação referente a inscrição de um aluno no sistema.
- Enrollments-Disciplines (Matrículas-Disciplinas): relação de muito-para-muitos entre a entidades “Enrollments (Matrículas)” e “Disciplines (Disciplinas)” .
- Sections-Disciplines (turmas-Disciplinas): relação de muito-para-muitos entre a entidades “Sections (turmas)” e “Disciplines (Disciplinas)” .

- Disciplines-School years (Disciplinas-anos escolares): entidade que relaciona as entidades “Disciplines” e “Courses- School years”.

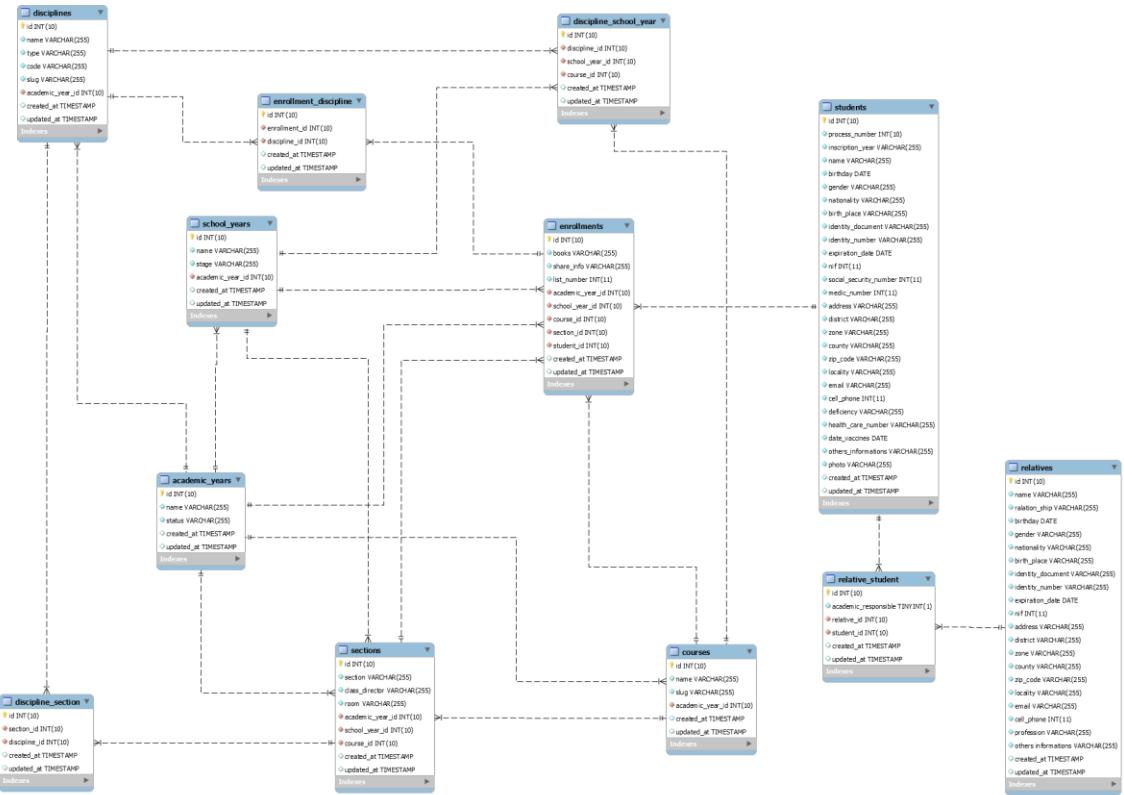


Figura 22 - Diagrama de entidade relação do módulo de inscrição

III.5.2. Módulo de atividades extracurriculares

Módulo que gera a atividades extracurriculares que a estabelecimento de ensino dispõe, como se apresenta na figura 23

- Activities (Atividades): entidade que gera as atividades extracurriculares disponíveis na instituição.
- Activities-Enrollments (Atividades- Matrículas): relação de muito-para-muitos entre a entidades “Activities (Atividades)” e “Enrollments (Matrículas)”. Esta entidade é a que serve de ligação entre o módulo de inscrição e o módulo de atividades extracurriculares, através da entidade matrícula.

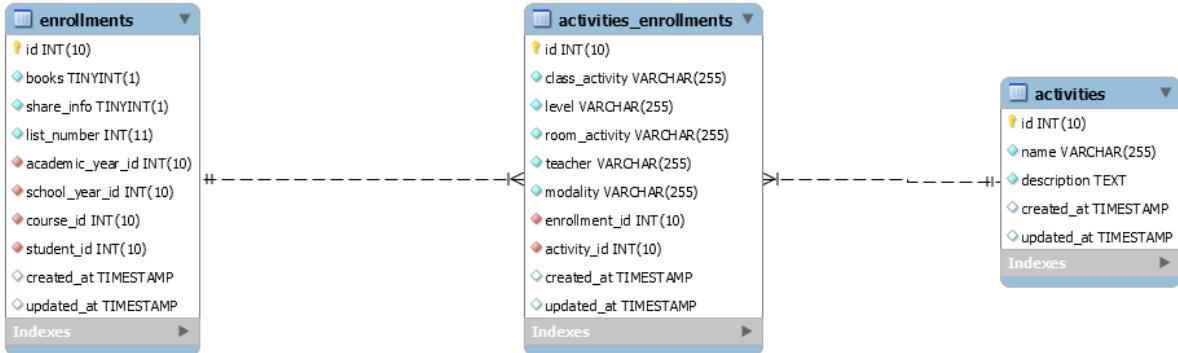


Figura 23 - Diagrama de entidade relação do módulo atividades extracurriculares

III.5.3. Módulo de serviços

Módulo do sistema que trata dos serviços, como mostra la figura 24

- Services (Serviços): entidade que gere os serviços disponíveis na instituição.
- Enrollments-Services (Matrículas-Serviços): Relação de muito-para-muitos entre a entidades “Enrollments (Matrículas)” e “Services (Serviços)”. Esta entidade é a que serve de ligação entre o módulo de inscrição e o módulo de serviços através da entidade matrículas.

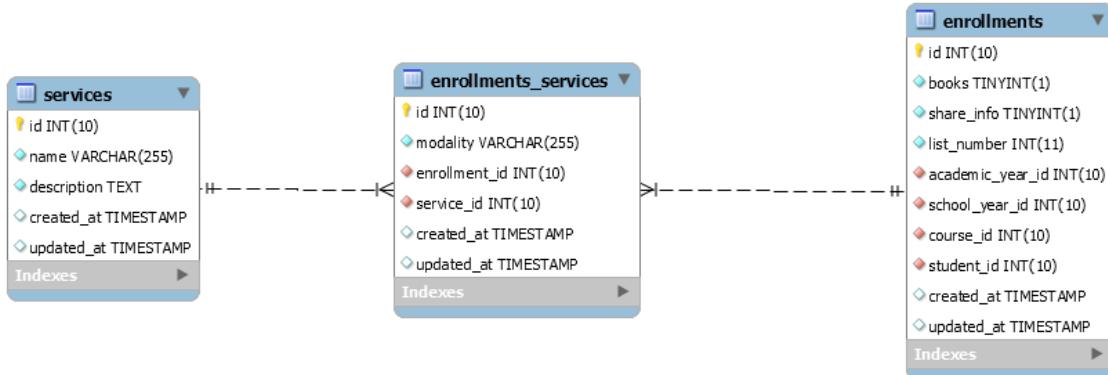


Figura 24 - Diagrama de entidade relação do módulo de serviços

III.5.4. Módulo de lançamento de notas

Este módulo é o encarregue do processo de lançamento de notas, nele participam as seguintes entidades, como mostra a figura 25

- Evaluation reports (relatórios de avaliações): armazena a informação necessária para fazer os relatórios de avaliações. Esta entidade serve de ligação entre o módulo de inscrição e o módulo de lançamento de notas.
- Evaluation parameters (parâmetros de avaliação): entidade que contém os parâmetros de avaliação e as notas relativas a esses parâmetros. Esta entidade tem uma relação com a tabela disciplinas do módulo de inscrição
- Half period classifications (classificações de metade de período): contém as notas obtida pelos alunos na avaliação intercalar, do 1º e 2º período. Esta entidade tem uma relação com a tabela disciplinas do módulo de inscrição.
- Period classifications (classificações de período): contém as notas obtidas pelos alunos no final do 1º, 2º e 3º período. Esta entidade tem uma relação com a tabela disciplinas do módulo de inscrição.

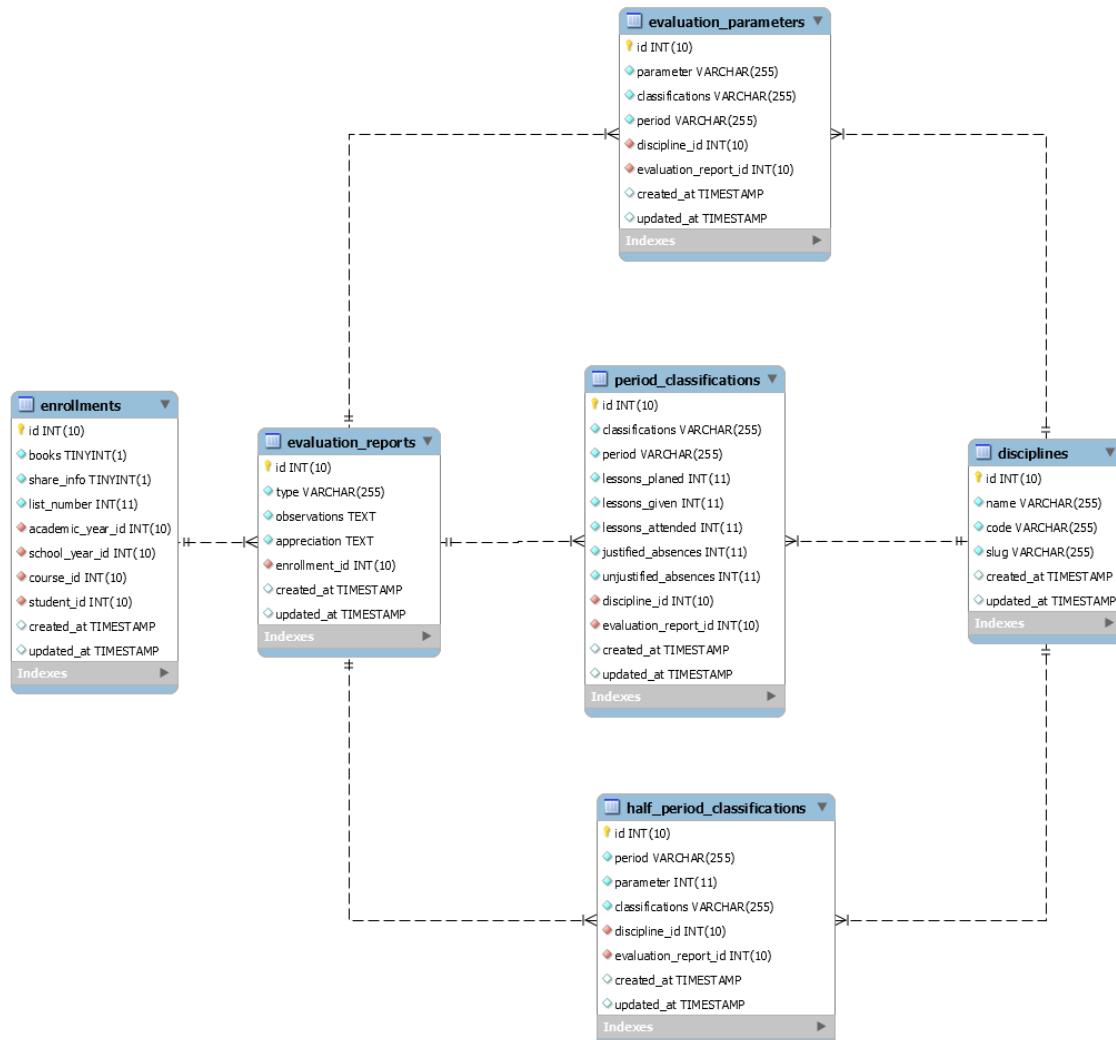


Figura 25 - Diagrama de entidade relação do módulo de lançamento de notas

III.5.5. Módulo de currículos académicos

O módulo encarregue pela criação do currículo académico contém uma única entidade, como é possível ver na figura 26

- Achievements (conquistas): responsável por gerar o currículo académico, contém dados como o âmbito e descrição das conquistas. Esta entidade está relacionada com os alunos, uma entidade do módulo de inscrição.

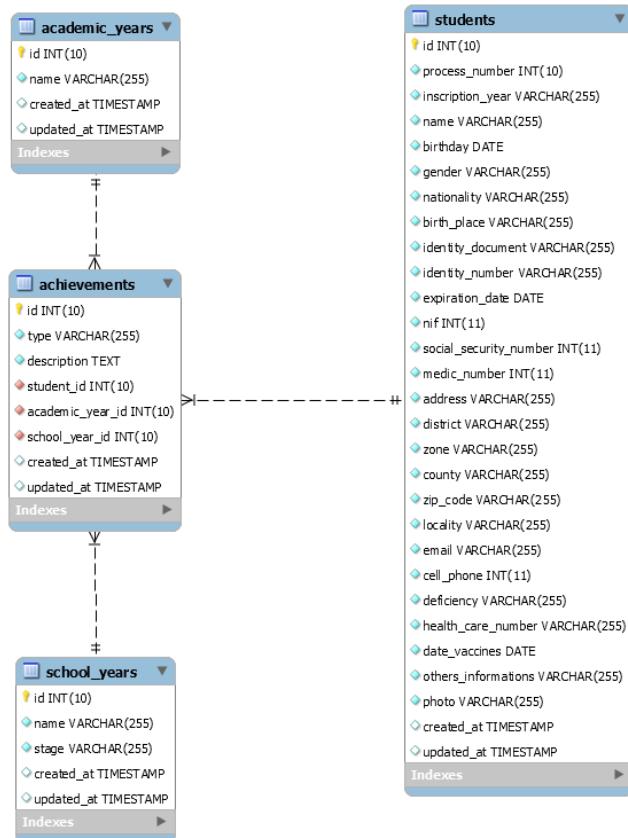


Figura 26 - Diagrama de entidade relação do módulo de currículos académicos

III.5.6. Outras entidades

Laravel para funcionar corretamente cria três entidades representadas na figura 27

- Migrations (Migrações): esta tabela guarda as migrações feitas na base de dados.
- Users (Utilizadores): laravel cria esta tabela por definição, para guardar os dados dos utilizadores que irão utilizar o sistema

- Passwords_resets: o mesmo que que a tabela *users*, é um componente do sistema de login de utilizadores de Laravel.

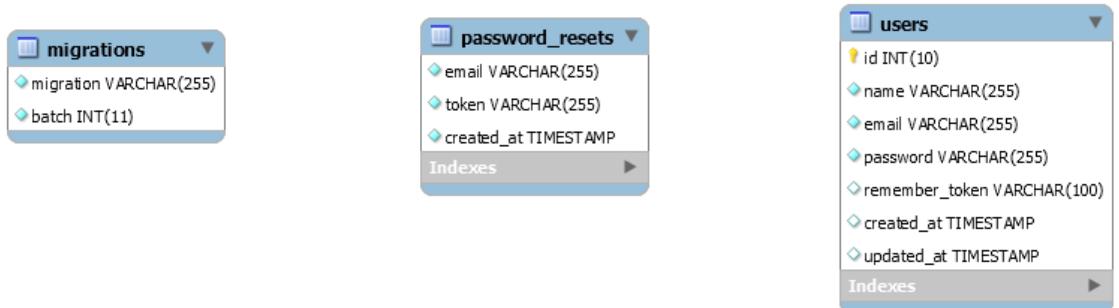


Figura 27 - Diagrama de entidade relação de entidades criadas pelo framework

III.6. INTERFACES DO UTILIZADOR

As interfaces do utilizador criam um meio de comunicação efetivo entre o humano e o computador. Elas são muito importantes, porque se um *software* for difícil de utilizar, se compele a incorrer erros ou frusta os esforços de atingir os objetivos, o utilizador não vai gostar do sistema, independentemente do poder computacional apresentado, do conteúdo fornecido ou das funcionalidades oferecidas. As interfaces devem ser corretas, pois moldam a percepção do software pelo utilizador.

Existem três regras de ouro que orientam o desenvolvimento de interfaces do utilizador eficazes:

1. Deixar o utilizador no comando;
2. Reduzir a carga de memória do utilizador;
3. Tornar a interface consistente.

O desenvolvimento de uma interface do utilizador começa com uma série de tarefas de análise. A análise dos usuários define os perfis de vários usuários finais e é reunida com base em uma série de fontes técnicas e comerciais. A análise de tarefas define as tarefas e ações de utilizadores, usando uma abordagem de refinamento ou orientada para objetos, aplicando casos de uso, elaboração de tarefa e objetos, análise de fluxo de dados e representações hierárquicas de tarefas para entender completamente a interação homem-computador. A análise do ambiente identifica as estruturas físicas e sociais em que a interface deve operar.

Uma vez identificadas as tarefas, são criados e analisados cenários de utilizadores para definir um conjunto de ações e objetos de interface. Isto fornece uma base para a criação de um layout de ecrã que represente a design gráfico e o posicionamento de ícones, a definição de textos descritivos no ecrã, a representação e a colocação de títulos nas janelas, bem como a especificação de itens de menu principais e secundários. Uma grande variedade de ferramentas de implementação é usada para construir um protótipo para avaliação das interfaces.

A interface de utilizador é a janela para o *software*. Em muitos casos, muda a percepção do utilizador quanto à qualidade de um sistema. Se esta “janela” for confusa, o utilizador poderá rejeitar um sistema que de uma forma seria considerado poderoso.

[22]

III.6.1. Interfaces de configuração do sistema

Com o objetivo de configurar o sistema para o seu funcionamento, foram criadas as seguintes interfaces

III.6.1.1. Interface - anos letivos

Como mostra a imagem 28, esta interface permite criar o ano letivo sobre o qual o sistema vai funcionar, assim como mudar o estado do ano letivo de “ativo” ou “desativo”. O sistema não permite ter mais do que um ano letivo ativo. Através desta interface, também conseguimos observar os anos escolares que já foram criados, assim como o estado dos mesmos.

| Ano letivo | Estado | |
|------------|----------|---|
| 2016/2017 | Ativo | Editar Apagar |
| 2015/2016 | Desativo | Editar Apagar |

Figura 28 - Vista para criar um novo ano letivo

III.6.1.2. Interface - anos escolares

A interface representada pela figura 29 permite a criação de anos escolares. Para criar o ano letivo é obrigatório escolher o ciclo ao qual pertence. Através desta interface, também podemos ver os anos escolares criados relativamente ao ano letivo ativo.

Figura 29 - Vista para criar um novo ano escolar

III.6.1.3. Interface - cursos

A figura 30 apresenta a interface desenvolvida para a criação de cursos do ano letivo ativo, nesta interface também podemos ver os cursos criados anteriormente.

Figura 30 - Vista para criar um novo curso

III.6.1.4. Interface – Disciplinas

Através desta interface, é possível criar as disciplinas alusivas ao ano letivo ativo, assim como associar as disciplinas aos cursos e aos anos escolares. Mediante esta interface representada pela figura 31, também podemos ver as disciplinas já criadas.

The screenshot shows a web-based application interface for managing disciplines. At the top, there's a header with a user icon and the text 'Utilizador'. Below the header, the title 'Nova Disciplina' is displayed. The main area is divided into two sections: 'Nova disciplina' (New discipline) and 'Disciplinas' (Disciplines).

Nova disciplina: This section contains input fields for creating a new discipline. The 'Disciplina' field is filled with 'Matemática'. The 'Tipo' field has a dropdown menu showing 'Geral'. The 'Código' field contains '1234', and the 'Abreviatura' field contains 'Mat'. The 'Curso' field is set to 'Ensino Básico', and the 'Ano escolar' field is set to '1º Ano'. A blue 'Salvar' (Save) button is located at the bottom of this section.

Disciplinas: This section displays a table of existing disciplines. The table has columns: Disciplina, Tipo, Código, and Abreviatura. Two rows are visible: one for 'Português' (Type Geral, Código Port1, Abreviatura Port) and another for 'Matemática' (Type Geral, Código empty, Abreviatura Mat). Each row has 'Editar' and 'Apagar' (Edit and Delete) buttons. Above the table, there are filters: 'Mostrar 10 registos' (Show 10 records) and a search bar 'Procurar:'.

Figura 31 - Vista para criar uma nova disciplina

III.6.1.5. Interface – Turmas

Na figura 32, conseguimos ver o processo de criação de turmas relativas ao ano letivo ativo. Nesta interface, criamos a turma, assim como, associamos as turmas a um ano escolar a um curso e adicionamos as disciplinas que serão lecionadas a determinada turma. Esta interface também nos permite ver as turmas já criadas, assim como alguns dos seus atributos.

Figura 32 - Vista para criar uma nova turma

III.6.2. Interfaces do módulo principal – Inscrições

Após o acabamento do processo de configuração, podemos prosseguir com as interfaces dos módulos implementados.

O processo de inscrição é realizado em quatro etapas. Uma primeira etapa, representada pela figura 33, onde é registado o aluno, nesta interface são preenchidos os dados do aluno que vai fazer a matrícula.

Os primeiros dados requeridos é o tipo de identificação e o seu respetivo número, isto porque o sistema faz uma pesquisa para garantir que não vamos criar um aluno já criado no sistema.

Nesta interface existem alguns dados obrigatórios, como o tipo de identificação, o número, o nome e o sexo.

O número de processo é um número interno gerado automaticamente pelo sistema.

Novo aluno

Novo aluno

| | |
|------------------------------|--|
| Documento de identificação | Nº de documento de identificação |
| | 11111111 1ZZ1 |
| Data de Validade | Nº de identificação fiscal |
| dd/mm/aaaa | 11111111 |
| Nº de Processo | Ano de admissão no Colégio |
| 1010 | |
| Nome do aluno | Data de Nascimento |
| nome | dd/mm/aaaa |
| Sexo | Nacionalidade |
| | Portugal |
| Naturalidade | Número segurança social |
| Portugal | 11111111 |
| Número de utente | Morada |
| 11111111 | |
| Distrito | Freguesia |
| | |
| Concelho | Código Postal |
| | 0000-000 |
| Localidade | email |
| | |
| Telemóvel | O aluno parece de alguma doença ou alergia? Sim, qual? |
| 99999999 | |
| Número de assistência médica | Data de validade de vacinas |
| | dd/mm/aaaa |
| Outras informações | Foto |
| | <input type="button" value="Escolher ficheiro"/> Nenhum fiche... selecionado |

Salvar

Figura 33 - Vista para criar um novo aluno

A segunda etapa é o registo de familiares. É possível associar vários familiares a um aluno e escolher um deles para ser o encarregado de educação, o primeiro dado a introduzir é o nif (número de identificação fiscal), desta forma o sistema verifica se existe algum familiar já registado no sistema com este número, se existir, o sistema preenche automaticamente os restantes dados permitindo também a edição, no caso de ser necessário. No caso de não estar registado, o utilizador tem que preencher os dados requeridos. A figura 34 representa esta interface.

The screenshot shows a user interface for creating a new family member ('Novo Familiar'). The form is divided into several sections:

- Identificação:** Nº de identificação fiscal (11111111), Parentesco (dropdown), Nome (nome), Data de Nascimento (dd/mm/aaaa), Sexo (dropdown), Nacionalidade (Portugal), Naturalidade (Portugal), Documento de identificação (dropdown), Data de Validade (dd/mm/aaaa).
- Endereço:** Morada (dropdown), Freguesia (dropdown), Distrito (dropdown), Concelho (dropdown), Código Postal (0000-000), Localidade (dropdown).
- Contato:** email (dropdown), Telemóvel (99999999), Profissão (dropdown), Outras informações (dropdown).
- Guardião:** Encarregado de educação? (checkbox).

At the bottom are two buttons: 'Continuar' (Continue) and 'Adicionar outro familiar' (Add another family member).

Figura 34 - Vista para criar um familiar

A figura 35 mostra a terceira etapa do processo, na qual o utilizador, para acabar o processo de inscrição, deve escolher o curso, o ano letivo, dependendo do ano tem que escolher as disciplinas específicas. Nesta interface, neste caso, tem que autorizar ou não a divulgação de imagem do educando nas redes sociais do colégio, assim como também encomendar ou não os manuais escolares pelo colégio.

Figura 35 - Vista para realizar a inscrição de um aluno

A quarta, e última, etapa deste processo consiste na inscrição do aluno numa das turmas, assim como a assinatura do número de listagem. Esta etapa está representada pela figura 36.

Figura 36 - Vista de inscrição de alunos numa turma

III.6.3. Interface do módulo de atividades extracurriculares

O módulo de atividades extracurriculares foi desenvolvido com dois interfaces.

Uma primeira interface, representada pela figura 37, na qual é possível criar a atividade extracurricular, assim como também adicionar uma descrição. Nesta interface também é possível ver as atividades já existentes no sistema.

Figura 37 - Vista para criar uma atividade extracurricular

Uma segunda interface, na qual o aluno se escreve na atividade, tal como é possível observar na figura 38.

The screenshot shows a registration form for a student activity. At the top, there's a header with the title 'Inscrição' and a breadcrumb navigation 'Home > Inscrição'. The form has several sections:

- Ano escolar:** A dropdown menu showing '1 [E-B]'
- Encomenda os livros escolares pelo Colégio?**: A dropdown menu showing 'Sim'
- Autoriza a divulgação de imagem nas redes sociais e site do Colégio?**: A dropdown menu showing 'Sim'
- Actividades:** A list box containing 'Natação'
- Serviços:** A list box containing 'Transporte'

At the bottom left is a blue 'Salvar' (Save) button.

Figura 38 - Vista de inscrição de um aluno numa atividade extracurricular

A inscrição de um aluno em atividades extracurricular, é concretizado no momento da inscrição no colégio.

III.6.4. Interface do módulo de serviços

No módulo de serviços podemos encontrar dois interfaces.

A figura 39, representa a primeira interface deste módulo. Nesta interface é possível criar os serviços, adicionar uma descrição e visualizar os serviços criados.

The screenshot shows a service creation interface. At the top, there's a header with the title 'Novo serviço' and a breadcrumb navigation 'Home > Novo serviço'. The form has two main sections:

- Novo serviço**: Contains fields for 'Nome' (Nome) and 'Descrição' (Description), both currently empty. Below these is a blue 'Salvar' (Save) button.
- Serviços**: A table listing existing services. The columns are 'Nome' (Name) and 'Descrição' (Description). One row is visible: 'Transporte' and 'Transporte'. To the right of the table are buttons for 'Procurar:' (Search) and 'Editar' (Edit). Navigation buttons at the bottom include 'Anterior' (Previous), '1' (Current page), and 'Seguinte' (Next).

Figura 39 - Vista de criação de um serviço

A segunda interface, a qual é possível observar através da figura 40, o aluno seleciona os serviços que quere adquirir.

The screenshot shows a registration form with the following fields:

- Ano escolar:** 1º [E-B]
- Encomenda os livros escolares pelo Colégio?**: Sim
- Autoriza a divulgação de imagem nas redes sociais e site do Colégio?**: Sim
- Actividades:** Natação
- Serviços:** Transporte

A blue "Salvar" (Save) button is at the bottom left.

Figura 40 - Vista de adquisição de uma serviço por parte de um aluno

A adquisição de um serviço, por parte de um aluno, é executado no momento da inscrição no colégio

III.6.5. Interface do módulo de currículos académicos

O módulo de currículos académicos, está representada por duas interfaces. Uma primeira interface, onde podemos criar os currículos dos alunos, isto é, adicionar as atividades nas quais o aluno em causa obteve algum destaque. Devido à existência de atividades onde vários alunos se destacaram ao mesmo tempo, nesta interface, representada pela figura 41, podemos selecionar mais do que um aluno ao mesmo tempo para acrescentar este aspeto ao seu currículo académico

The screenshot shows a form for creating a new curriculum with the following fields:

- Ano letivo:** 2016/2017
- Ano escolar:** 1º Ano
- Alunos:** (empty dropdown)
- Âmbito:** (empty dropdown)
- Descrição:** (text area)

A blue "Salvar" (Save) button is at the bottom left.

Figura 41 - Vista de criação de currículo académico de um aluno

Na segunda interface relacionada com este módulo é possível ver a listagem de alunos com currículo académico criado, com as opções de aditar e imprimir o respetivo currículo em formato pdf. Na figura 42 observamos a interface acima descrita.

| Curriculos | | | |
|---|--------------------|--------------------|---|
| Mostrar 10 ▾ registos | | Procurar: | |
| Foto | ALUNO | Numero de Processo | |
|  | Afonso [REDACTED] | 1168 |  |
|  | António [REDACTED] | 364 |  |
|  | António [REDACTED] | 365 |  |
|  | Beatriz [REDACTED] | 341 |  |
|  | Carla [REDACTED] | 343 |  |

Figura 42 - Vista da listagem dos alunos com currículo académico criado

III.7. IMPLEMENTAÇÃO

III.7.1. Tecnologias utilizadas

Escolher a tecnologia a utilizar no projeto é fundamental no processo de desenvolvimento, esta irá definir como o sistema vai ser desenvolvido e como os utilizadores irão interagir com ele.

Manutenção, sustentabilidade e escalabilidade são algumas das variáveis a ter em conta no momento de enfrentar esse desafio, por este motivo, é essencial fazer um estúdio e pesquisa das tecnologias que existem no mercado, para perceber quais delas é a melhor escolha.

Após a pesquisa e análise das ferramentas mais utilizadas atualmente, decidi, pelas suas características e porque já conhecia esta ferramenta, utilizar PHP como linguagem de desenvolvimento principal, juntamente com MySQL como gestor de base de dados e outras tecnologias, as quais passo a explicar com mais detalhe.

III.7.1.1. PHP

PHP, que significa "PHP: Hypertext Preprocessor", é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a web e pode ser misturada dentro do código HTML. A sintaxe da linguagem lembra C, Java e Perl, e é fácil de aprender. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente e rapidamente, mas pode-se fazer muito mais do que isso com PHP.

O que distingue o PHP de algo como o *Javascript* no lado do cliente é que o código é executado no servidor, gerando o HTML que é então enviado para o navegador. O navegador recebe os resultados da execução desse script, mas não sabe qual é o código fonte. É possível, inclusive, configurar o servidor web para processar todos os seus arquivos HTML com o PHP, assim, não haverá realmente nenhum modo dos utilizadores descobrirem se o programador está a usar essa linguagem ou não.

A melhor coisa em usar o PHP é que é extremamente simples para um iniciante, mas oferece muitos recursos para um programador profissional. [25]

Numa primeira fase, comecei a desenvolver em PHP, mas rapidamente me apercebi que precisava de outro instrumento que me ajudasse a desenvolver mais depressa, com

mais segurança, com práticas de desenvolvimento web modernas, etc. Nesse momento, comecei a estudar a possibilidade de utilizar um *framework* para auxiliar estas minhas necessidades.

III.7.1.2. Frameworks PHP

Construir aplicações de *software* pode ser um processo complexo, demorado, porém utilizando um *framework* pode ajudar a desenvolver projetos mais rápidos (através da reutilização de componentes e módulos genéricos), e trabalhar melhor (construir (retirar) uma base estrutural unificada). Usando um *framework* também facilita a escalabilidade e manutenção a longo prazo através do cumprimento de padrões de desenvolvimento, mantendo o código organizado e permitindo sua aplicação para evoluir e crescer ao longo do tempo. [15] O uso de um *framework* traz inúmeráveis vantagens:

- Tornar possível o desenvolvimento rápido;
- Fornecer o código bem organizado, reutilizável e de fácil manutenção;
- Permitir que o código cresça a tempo, como aplicações web que utilizam *framework* são escaláveis;
- Poupar as preocupações de segurança de baixo nível de um site;
- Seguir o padrão MVC (*Model-View-Controller*), que garanta a separação de apresentação e lógica;
- Promover práticas de desenvolvimento web modernas, tais como ferramentas de programação orientada a objeto; [16]
- Fornece grande apoio comunitário;
- Ter utilitários e bibliotecas;
- Abstração de banco de dados; [17]

Teoricamente, os *frameworks* foram criados para ajudar e promover soluções flexíveis para problemas comuns e bases de código úteis. Depois de analisar o mercado de *frameworks*, decidi escolher Laravel para desenvolver o sistema gestor académico.

Laravel:

Embora Laravel seja um *framework* de PHP relativamente novo (foi lançado em 2011), de acordo com recente pesquisa *online* da Sitepoint é o *framework* mais popular entre os desenvolvedores. Laravel tem um enorme ecossistema com um instante de hospedagem e plataformas de implementação, no site oficial encontram-se muitos tutoriais *screencast* chamados *Laracasts*.

Laravel tem muitas características que tornam o desenvolvimento rápido de aplicações possível. Laravel tem o seu próprio motor de templates chamado "Blade", sintaxe elegante que facilita as tarefas mais frequentemente, como autenticação, sessões, filas, caching e roteamento RESTful. Laravel também inclui um ambiente de desenvolvimento local chamado de Homestead, que é uma caixa de Vagrant [16].



Figura 43 - Logo do framework Laravel

Na figura 46, podemos observar como Laravel implementa o padrão MVC.

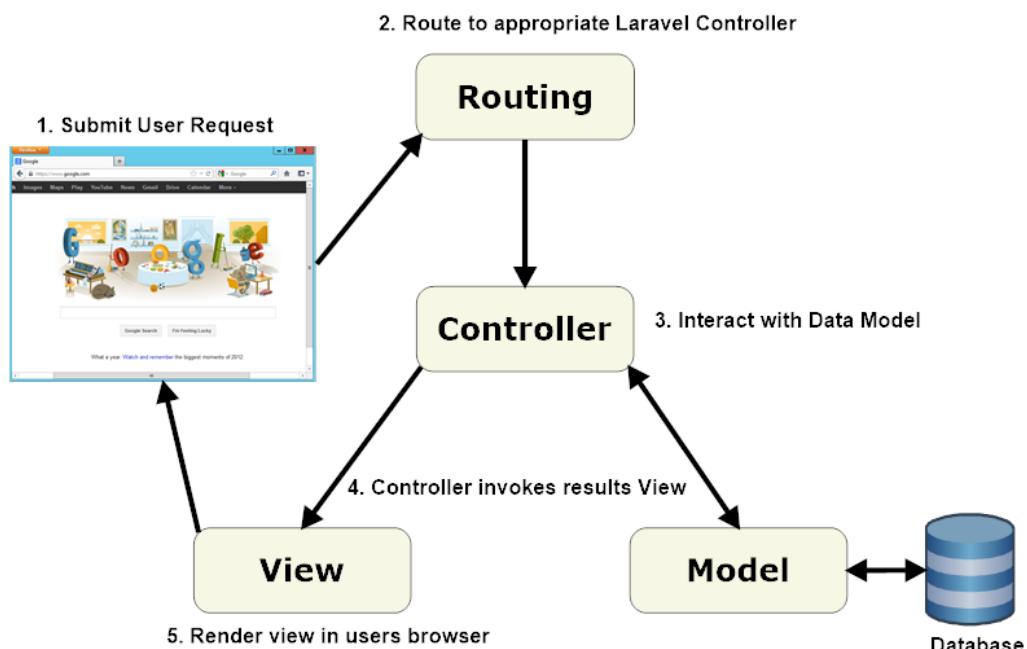


Figura 44 - Padrão MVC implementado por Laravel

1. O utilizador enviar um pedido
2. A Ruta envia ao controlador apropriado.
3. O controlador interage com o Modelo.
4. O controlador invoca a vista resultante.
5. É fornecida a vista ao navegador do usuário.

Laravel para ser instalado, pede os seguintes requisitos:

- PHP >= 5.5.9;
- Extensão PHP OpenSSL;
- Extensão PHP PDO;
- Extensão PHP Mbstring;
- Extensão PHP Tokenizer.

Laravel esforçar-se por fazer todo o desenvolvimento em PHP uma experiência simples, por este motivo, existe várias ferramentas que ajudam e facilitam a instalação de Laravel e a sua utilização.

Vagrant:

Vagrant é uma ferramenta de desenvolvimento que facilita a criação de ambientes virtuais para o desenvolvimento. Em Vagrant podemos instalar e configurar o *software* em uma máquina virtual, para que possamos fingir que estamos no servidor no qual o aplicativo Web vai ficar. A principal vantagem é que, se queremos desenvolver um aplicativo da Web, não tem que configurar nada no computador e preenchê-lo com programas como o Apache, Nginx, PHP, MySQL, etc. Vagrant tem "caixas" com sistemas operativos para desenvolver diretamente sobre eles. Se, por exemplo, queremos uma máquina virtual com LAMP ou Node.js, podemos baixá-lo, configurá-lo rapidamente e, logo, começar a programação.

O facto de não ter que configurar nada no computador, faz com que, se por equívoco desconfiguramos alguma ferramenta seja na máquina virtual e não diretamente no computador. [36] e [37].



Figura 45 - Logo de Vagrant

Para conseguir instalar e utilizar Vagrant, precisamos de instalar no nosso computador um *software* de virtualização, como por exemplo Virtual Box

VirtualBox:

É um *software* de virtualização para x86 que foi originalmente desenvolvido pela empresa alemã *Innotek GmbH*, mas que passou a ser de propriedade da Sun Microsystems em fevereiro de 2008, quando comprou *Innotek*. Através desta aplicação é possível instalar sistemas operativos adicionais, conhecidos como "sistemas convidados" dentro de outro sistema operativo "hospedeiro", cada um com seu próprio ambiente virtual. [38]



Figura 46 - Logo de VirtualBox

Após a instalação de um *software* de virtualização, neste caso VirtualBox e seguidamente de Vagrant, o passo seguinte para instalar Laravel no nosso computador é instalar Laravel Homestead.

Laravel Homestead:

Laravel Homestead é o pacote oficial de Vagrant "box" que fornece um ambiente de desenvolvimento maravilhoso sem instalar PHP, HHVM, servidor web ou qualquer outro *software* de servidor no computador local. As caixas Vagrant são totalmente descartáveis. Se alguma coisa correr mal, é possível destruir e recriar a caixa em questão de minutos.

Homestead é executado em qualquer sistema Windows, Mac ou Linux e inclui Ubuntu 14.04, PHP 5.6, HHVM, Nginx, MySQL, Postgres, Node (Con Bower, Grunt y Gulp), Redis, Memcached, Beanstalkd, Laravel Envoy, Extensiones Fabric + HipChat e Blackfire Profiler.

Uma vez instalados e configurados VirtualBox e Vagrant, devemos incluir a box de laravel/homestead na instalação de Vagrant, utilizando o seguinte comando de consola. *vagrant box add laravel/homestead.* [18]

Para criar um projeto em laravel, existem duas formas, a primeira é descarregando o ficheiro master o repositório de GitHub e a outra é usando *Composer* desde a consola, neste caso decidi utilizar *Composer*.

Composer:

Composer é uma ferramenta para gerenciamento de dependências em PHP. Permite declarar as bibliotecas dependentes que o projeto precisa, instala-as e atualiza-as. [26]

Utilizando Composer, basta simplesmente especificar quais os pacotes (códigos reutilizáveis) que o projeto precisa (podendo estes pacotes também ter dependências) e vai automaticamente fazer *dowloand* e incluir nos locais apropriados do projeto. Caso seja preciso acrescentar, remover ou atualizar algum pacote, o gerenciador também consegue fazer isso.

Composer funciona, basicamente, através de duas vertentes: um repositório para os pacotes (Packagist) e instruções via linha de comando para gerenciamento dos pacotes (para procurar, instalar, atualizar, remover).

A instalação dos pacotes é feita por projeto e por *default*, nada é instalado globalmente. Por isso o Composer é considerado mais um Gerenciador de Dependências do que um Gerenciador de Pacotes.

Depois de configurar corretamente o arquivo “*composer.json*” para instalar as dependências informadas, é feita uma verificação para ver se há algum erro de sintaxe, acontece a busca no repositório pelo pacote informado, o *download* é realizado e a “instalação” feita no diretório apropriado.

O primeiro passo para entrar no mundo de gerenciamento de pacotes PHP é instalar o Composer. [27]



Figura 47 - Logo de Composer

Packagist:

Packagist é o repositório principal de Composer. Um repositório de Composer é basicamente o lugar onde se obtêm os pacotes. O objetivo de Packagist é converter-se no repositório central que utilizem todos os utilizadores de Composer. Portanto, é possível incluir no “*require*” do ficheiro “*composer.json*” qualquer pacote disponível em Packagist. No site oficial de packagist, é possível procurar, navegar e encontrar milhares de pacotes disponíveis. [28] e [29]



Figura 48 - Logo de Packagist

Para realizar o sistema, recorri a vários pacotes em packagist:

- laravelcollective/bus: este pacote fornece um método conveniente de encapsular as tarefas que a aplicação precisa para executar de forma simples e fácil de entender "comandos".
- laravelcollective/html: é um pacote que permite a criação de formulários html; para conseguir utilizar este pacote é necessário instalar primeiro laravelcollective/bus.
- yajra/laravel-datables-oracle: este pacote foi criado para lidar com o trabalho do lado do servidor de DataTables jQuery Plugin via AJAX usando Eloquente ORM, Fluente Query Builder ou Collection.

Laravel Collective:

Com a evolução de laravel, ter-se-ão perdido alguns componentes do núcleo principal. Com o objetivo de manter estes elementos foi criado Laravel Collective. [30]



Figura 49 - Logo de Laravel Collective

Estas foram as tecnologias utilizadas no back-end. No front-end foram utilizadas as seguintes tecnologias:

JavaScrip:

Javascript é uma linguagem que surgiu com o objetivo inicial de programar certos comportamentos sobre as páginas web, respondendo à interação do utilizador e à realização de automatismos simples. Neste contexto, podemos dizer que nasceu como uma linguagem de *scripting* do lado do cliente, embora, hoje *Javascript* seja muito mais. As necessidades das aplicações web modernas e o HTML 5, têm provocado que o uso de *javascript*, que encontramos hoje, tenha chegado a uns níveis de complexidade e prestações tão grandes como outras linguagens de primeiro nível. [42]

JQuery:

jQuery é a livraria de *Javascript* mais conhecida. Com Jquery é possível escrever código *Javascript* que pode ser capaz de se executar sem erros em qualquer navegador, incluído os antigos e implementa muitas funcionalidades.

jQuery permite, além de programar novas funcionalidades através de *plugins*, pode fazer coisas tão variadas como validação de formulários, sistemas de modelos, passe de diapositivos, interfaces de utilizadores avançados, etc. [42]

AJAX:

AJAX é uma técnica que permite a comunicação assíncrona entre um servidor e um browser em formato XML através de programas escritos em *Javascript*. O principal objetivo de AJAX é intercâmbio de informações entre o servidor e o cliente (*browsers*) sem ter a necessidade de recarregar o *site*. Desta forma, ganhamos em usabilidade, experiência e produtividade do usuário final. [43]

HTML 5:

HTML é a linguagem com a qual se escreve as páginas web. As páginas web podem ser vistas pelos utilizadores através dos browsers. Podemos dizer, então, que HTML é a linguagem utilizada pelos browsers para mostrar as páginas webs aos utilizadores.

Esta linguagem permite aglutinar textos, sons e imagens e combiná-los de acordo com os nossos gostos. Além disso, e é aqui onde reside a vantagem com respeito a livros e revistas, HTML permite a introdução de referências às outras páginas por meio de ligações de hipertexto. [44]

HTML5 é a mais nova versão de HTML. HTML5 incorpora características nas quais se encontra suporte à medida, de modo a melhorar o suporte à criação de aplicações web, que podem interagir com o usuário, os seus dados locais e servidores de maneira mais fácil e efetiva. [45]

CSS 3:

CSS é uma linguagem utilizado na apresentação de documentos HTML. Um documento HTML é uma página web, por isso podemos dizer que CSS serve para organizar a apresentação e o aspetto de uma página web. Esta linguagem é principalmente utilizada pelos browsers para eleger uma multiplicidade de opções de apresentação como cores, tipos e tamanhos de letras, etc.

A filosofia de CSS baseia-se em tentar separar o que é a estrutura do documento HTML da sua apresentação. Por exemplo, o *site* seria o que há por baixo (o conteúdo) e o CSS seria um cristal de cor que faz com que o conteúdo se veja de um a forma ou de outra. Usando esta filosofia, resulta muito facilmente alterar o aspetto a uma página web, basta alterar “o cristal” [46]

A especificação de CSS3 traz novidades interessantes que permitem fazer *sites* mais elaborados e mais dinâmicas, com maior separação entre estilos e conteúdos. CSS3 dá suporte a muitas necessidades dos *sites* atuais, sem ter que recorrer a truques de designs ou linguagens de programação. [47]

Bootstrap:

Bootstrap é uma *framework* criada originalmente pelo Twitter, que permite criar interfaces web com CSS e *JavaScript*, cuja particularidade é adaptar a interface do *site* para o tamanho do dispositivo que é exibido. Isto é, o *site* adapta-se automaticamente ao tamanho de um PC, um tablet ou outro dispositivo. Esta técnica de design e desenvolvimento é conhecido como "*responsive design*" ou desenho adaptativo.

Embora a oferta de todas as possibilidades que Bootstrap oferece ao criar interfaces web, o design criado com Bootstrap é simples, limpa e intuitiva, isso dá-lhe flexibilidade ao carregar e adaptar-se a outros dispositivos. O *framework* traz vários elementos com estilos predefinidos fácil de configurar: botões, menus suspensos, formulários, incluindo os seus elementos e integração jQuery para oferecer janelas e tooltips dinâmicos. [34] e [35]



Figura 50 - Logo de Bootstrap

Font Awesome:

Font Awesome é uma fonte que podemos utilizar para implementar ícones sem a necessidade de imagens e sem técnicas intrusivas.

As suas características são:

- Fácil implementação com CSS, não é necessário JavaScript e é possível modificar propriedades como a cor, tamanho, sombreado, etc.;
- SVG significa que cada ícone se vê impressionante em qualquer tamanho;
- Suporte para IE7 sem a necessidade de hacks;
- Suporte para retina display, o que significa que são magníficos em ecrans de alta resolução;
- Estão pensados para trabalhar corretamente com Twitter Bootstrap 2, mas também se podem utilizar sozinhos;
- É grátis. [32] e [33]

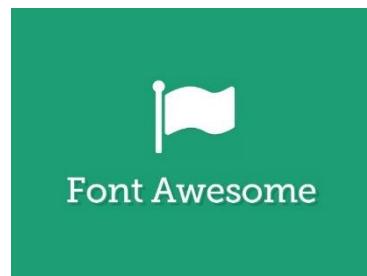


Figura 51 - Logo de Font Awesome

III.8. FUNCIONALIDADES IMPLEMENTADAS

À medida que o sistema ia sendo implementado, surgiu a necessidade de estabelecer quais das funcionalidades tinham prioridade para a implementação. Os motivos principais para tomar esta decisão foi o tempo e dificuldade em implementar alguns módulos. Por este motivo, alguns dos requisitos não foram implementados ou implementados de forma parcial, deste modo, as funcionalidades principais foram as que tiveram mais prioridade na hora da implementação.

III.8.1. Funcionalidades implementadas

No caso do sistema *Academic-Manager*, o módulo principal, inscrições, foi considerado o mais importante, pois através deste módulo é possível configurar o sistema para um novo ano letivo, assim como introduzir os alunos no mesmo. Por esse motivo, foi considerado uma prioridade.

Passo a destacar todas as funcionalidades implementadas na totalidade:

RF1. Os utilizadores devem ser capazes de adicionar, editar, gerir e/ou remover anos letivos.

RF2. Os utilizadores devem ser capazes de adicionar, editar e/ou remover utilizadores.

RF3. Os utilizadores devem ser capazes de adicionar, editar e/ou remover anos escolares.

RF4. Os utilizadores devem ser capazes de adicionar, editar e/ou remover cursos

RF5. Os utilizadores devem ser capazes de adicionar, editar e/ou remover disciplinas.

RF6. Os utilizadores devem ser capazes de adicionar, editar, gerir e/ou remover turmas.

RF7. Os utilizadores devem ser capazes de adicionar, editar e/ou remover serviços.

RF8. Os utilizadores devem ser capazes de adicionar, editar e/ou remover atividades extracurriculares

RF9. Os utilizadores devem ser capazes de adicionar, editar e/ou remover alunos.

RF10. Os utilizadores devem ser capazes de fazer a inscrição no colégio através do sistema

RF11. Os utilizadores devem ser capazes de fazer a renovação de matrícula no colégio.

RF12. O sistema deve permitir que os utilizadores realizem a inscrição nas diferentes atividades extracurriculares

RF13. O sistema deve permitir que os utilizadores adquiriram serviços através do sistema

RF14. Os utilizadores devem ser capazes de criar listagens dos alunos por atividade extracurricular

RF15. Os utilizadores devem ser capazes de criar listagens dos alunos por serviço.

RF16. Os utilizadores devem ser capazes de criar listagens dos alunos por turma

RF17. Os utilizadores devem ser capazes de criar listagens dos alunos por disciplinas

RF18. Os utilizadores devem ser capazes de criar listagens dos alunos por curso.

RF19. Os utilizadores devem ser capazes de criar listagens dos alunos por ano escolar.

RF20. Os utilizadores devem ser capazes de criar listagens dos alunos por ciclo de ensino.

RF25. O sistema deve gerar automaticamente o currículo académico dos alunos em PDF

III.8.2. Funcionalidades não implementadas

Os sobrantes requisitos que não foram referidos no subcapítulo anterior, são os que não foram implementados. Em suma, não foi implementado o módulo de avaliações, o qual se refletia nos seguintes requisitos:

RF21. A aplicação deverá permitir aos utilizadores visualizar, preencher, adicionar, modificar e/ou eliminar os parâmetros de todos os tipos de avaliações

RF22. O sistema deve gerar automaticamente os registos de avaliação de cada um dos alunos em PDF

RF23. O sistema deve gerar automaticamente todas as pautas necessárias.

RF24. O sistema deve gerar automaticamente os alunos destacados nos quadros de honra e de excelência.

III.9. SOFTWARE TESTING

Um *software* precisa de ser testado para se descobrirem os erros que foram cometidos durante o projeto. Os testes de *software* são um elemento indispensável para a garantia da qualidade de *software* e representa uma revisão final das especificações, do desenho e da codificação. [40] e [41]

Em cada fase do ciclo de vida de desenvolvimento do *software* é necessário realizar uma série de testes que permitem constatar que o sistema desenvolvido satisfaz as especificações dessa fase. Desta forma, na fase de análise do sistema, foram realizados os testes de sistema na base de dados, constatando que cada elemento encaixa de forma adequada e que alcança a funcionalidade e rendimento do sistema.

Durante a fase de análise de requisitos, foram realizados os testes de validação, que consiste em comprovar que o sistema desenvolvido satisfaz as expectativas do cliente, isto é, satisfazem-se os requisitos funcionais e não funcionais.

Durante a fase de desenho, foram aplicados os testes de integração, que consistem em comprovar que o sistema se haja construído corretamente. Os testes foram feitos com os módulos secundários do sistema, foram integrados ao módulo principal e foi comprovado que funciona corretamente.

No fim, foram feitos os testes unitários através dos testes de caixa negra e caixa branca.

Através das interfaces gráficas, foram feitos os testes de caixa negra, os quais pretendiam demonstrar que o sistema funciona adequadamente, isto é, que os dados introduzidos são aceites de forma e que o seu comportamento é o esperado.

Após a execução de estes testes e a correção de alguns erros, foram realizados os testes de caixa branca, estes testes comprovaram os caminhos lógicos do *software* através de minuciosas probas dos detalhes dos procedimentos.

Os testes unitários foram executados às interfaces referentes às funcionalidades implementadas.

III.10. CONCLUSÃO

A utilização de metodologias, ferramentas, estilos e padrões arquiteturais, juntamente com as tecnologias foram indispensáveis para o desenvolvimento e implementação do sistema *Academic-Manager*

Através da metodologia Ágil e XP foi possível flexibilizar o desenvolvimento de *software*, estimular a interação contínua com o cliente, para que as necessidades fossem visualizadas e compreendidas e esta forma obter o sucesso do projeto.

As ferramentas utilizadas como phpMyAdmin (para lidar com a administração da base de dados); PHPStorm (para editar o código fonte); Tello (para organizar as tarefas que tinha por fazer, impor metas semanais e ter uma visão geral e detalhada da evolução do projeto); MySQL Workbench (para desenhar de base de dados); Git (para o controlo de versões do projeto) e, finalmente, Bitbucket (para criar repositórios privados do projeto), permitiram desenhar e desenvolver o sistema com sucesso e maior facilidade.

Os estilos e padrões arquiteturais permitiram analisar a efetividade do projeto no atendimento dos requisitos, considerar alternativas de arquitetura na hora de realizar mudanças no projeto e minimizar os riscos associados à construção de um *software*.

Escolher a tecnologia a utilizar no projeto foi fundamental no processo de desenvolvimento. Inicialmente, comecei a desenvolver só em PHP, mas, rapidamente, me apercebi que precisava de outro instrumento, por esse motivo, decidi utilizar o *framework* Laravel para o desenvolvimento do sistema, juntamente com outras tecnologias, estes instrumentos me ajudaram e promoveram soluções flexíveis para problemas comuns e bases de código úteis.

Na hora da implementação, foi feito uma análise para determinar quais dos requisitos eram a prioridade na hora da implementação. Resumidamente, o módulo de inscrição, atividades extracurriculares, serviços e currículos académicos, foram implementados na totalidade, ao contrário do módulo de lançamento de notas que não foi implementado.

Durante todo o processo de desenvolvimento e implementação do sistema, e como parte da metodologia utilizada, foram realizados testes para certificar a qualidade do sistema, através dos testes foi possível encontrar e corrigir alguns erros, assim como obter um constante feedback dos utilizadores finais

IV. CONCLUSÃO E TRABALHO FUTURO

"The only way to do great work is to love what you do"

- Steve Jobs

IV.1. VISÃO CRÍTICA DO ESTÁGIO

A decisão de realizar um estágio curricular como dissertação foi, sem dúvida, favorável para minha carreira profissional, pois através do projeto desenvolvido, consegui adquirir competências, conhecimento e experiência que só conseguiria obter desta forma.

Neste capítulo final, pretendo efetuar uma reflexão analítica sobre o percurso do meu estágio curricular, de um ponto de vista geral.

Realizar o estágio numa empresa da área da educação, e não da área da informática, trouxe alguns desafios que consegui, com perseverança, ultrapassar:

- Visto que não existia no colégio uma pessoa com conhecimentos amplos na área da programação, a execução do trabalho recaiu essencialmente sobre mim, o que me trouxe alguns desafios acrescidos, contudo, todos os elementos que constituem a empresa, com as suas questões pertinentes e sugestões, me ajudaram a alcançar respostas e gerir a tomada de decisões, encontrando, assim, um rumo.
- O facto de não ter um orientador interno, nem uma pessoa com experiência na área de desenvolvimento que me pudesse ajudar e orientar, fez com que este estágio curricular fosse, ainda, mais desafiante. Devido a este facto, o processo, pontualmente, detinha-se em determinadas etapas, pois precisava de investigar e encontrar as respostas para os problemas que iam surgindo por mim própria.
- Como consequência desta falta de orientação, recorri muitas vezes a fóruns e pesquisa na internet, isto, por vezes, trouxe algumas questões, isto é, as informações que encontramos na internet nem sempre são fidedignas ou não se adequam aos nossos problemas. Esta situação levou-me a desenvolver uma visão mais crítica.

Com as dificuldades apresentadas, e como foi um processo individual, acabei por adquirir um amplo leque de conhecimentos, ajudou-me a desenvolver a capacidade de resolver problemas, tornando-me mais autónoma e independente, da mesma forma que consegui um crescimento na minha carreira profissional.

O facto de estar em constante contato com os utilizadores finais do sistema, ajudou-me a perceber, realmente, o que eles pretendiam que o sistema fizesse. Ter acesso a

documentos internos, presenciar os processos executados pelos utilizadores também facilitou a compreensão dos problemas que possuíam.

Ao fazer o estágio curricular, tive a oportunidade de escolher a tecnologia e ferramentas que queria utilizar. Através de pesquisa, descobri as ferramentas que estão a ser atualmente utilizadas no mercado laboral e, as quais, foram muito úteis e facilitaram o desenvolvimento do projeto. Da mesma forma, tive a liberdade de escolher a linguagem de programação com a qual me sentia mais confortável para trabalhar, assim como a arquitetura e metodologia a utilizar.

Nos primeiros tempos do curso, não se tem uma ideia clara do que se gosta e do que se quer fazer. Como o campo da informática é muito amplo, enveredar por um dos ramos mais específicos desta área e ter uma experiência laboral efetiva em determinada vertente de conhecimento, leva-nos a ter uma noção mais ampla sobre os nossos gostos pessoais e sobre as áreas em que pretendemos continuar a investir como profissionais, descobrindo, deste modo, o que realmente se pretende fazer no futuro.

Sem dúvida alguma, fazer o estágio curricular enriqueceu o meu *currículo vitae*, assim como me preparou para o campo laboral e me fez perceber o que pretendia fazer após esta experiência enriquecedora. Permitiu-me, deste modo, estar exposta a uma cultura e ambiente empresarial de diferentes. Tive a oportunidade de apreender as melhores práticas de comunicação e comportamento profissional, assim como, melhorar a comunicação escrita e oral para com os colegas diretos e os clientes. Também foi uma valia para refinar as competências sociais.

Toda esta aprendizagem fez com que esteja tanto profissionalmente, como pessoalmente, melhor preparada para um próximo emprego.

IV.2. PERSPECTIVAS FUTURAS

O facto de realizar o estágio num Colégio e não numa empresa de desenvolvimento informático fez com que o sistema desenvolvido fosse da minha total responsabilidade. Como todas as funcionalidades não foram implementadas, uma das perspetivas futuras é continuar a desenvolver o *software*, realizar melhorias no sistema, implementar novos requisitos, como criar acesso aos encarregados de educação, possibilitar a redação dos sumários através da ferramenta, permitir marcar faltas, faltas de material, atrasos em tempo real, desenvolver o módulo de lançamento de notas de tal forma que seja flexível e que se consiga adaptar às necessidades dos estabelecimentos de ensino, para depois entrar no campo comercial com este produto, isto é, como o projeto foi feito de raiz por mim, poderei, eventualmente, utilizá-lo para fins comerciais.

No ponto de vista pessoal, depois desta experiência produtiva, quero continuar a aprofundar os meus conhecimentos em tecnologias do desenvolvimento de *software*, procurar mais desafios e continuar a crescer do ponto de vista profissional. Deste modo, amadurecerei profissionalmente!!

V. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] “Engenharia de Software para Concursos - Parte 4 - Metodologias Ágeis” http://brunomarota.blogspot.pt/2012/04/engenharia-de-software-para-concursos_29.html, acedido em 19 de agosto, 2016,
- [2] “O que são essas tais de metodologias Ágeis ?”, https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas_o_que_s_c3_a3o_essas_tais_de_metodologias_c3_a1geis?lang=en, acedido em 19 de agosto, 2016, Juliana Berossa Steffen.
- [3] “Effective Practices for Software Solution Delivery”, <http://www.ambyssoft.com>, acedido em 19 de agosto, 2016, Copyright 1997-2016 Scott W. Ambler.
- [4] “Extreme Programming: A gentle introduction”, <http://www.extremeprogramming.org/>,
- [5] “Métodos ágeis: características, pontos fortes e fracos e possibilidades de aplicação” https://run.unl.pt/bitstream/10362/2003/1/WPSeries_09_2009Tomas.pdf, acedido em 19 de agosto, 2016,
- [6] “Metodologias ágeis alinhada às metas organizacionais”, <https://www.infoq.com.br/news/2015/08/agile-adoption-method>, acedido em 19 de agosto, 2016, por Savita Pahuja , traduzido por Lu Araujo
- [7] “The Agile Manifesto” <http://www.drdobbs.com/open-source/the-agile-manifesto/184414755?queryText=the+agile+manifesto>, acedido em 19 de agosto, 2016, Martin Fowler and Jim Highsmith
- [8] “Conceitos básicos sobre Metodologias Ágeis para Desenvolvimento de Software” <http://www.devmedia.com.br/conceitos-basicos-sobre-metodologias-ageis-para-desenvolvimento-de-software-metodologias-classicas-x-extreme-programming/10596>, acedido em 19 de agosto, 2016, Daniel Fonseca
- [9] “Modelos de Processos Ágeis: conceitos e princípios” <http://www.devmedia.com.br/modelos-de-processos-ageis-conceitos-e-principios/30059>, acedido em 19 de agosto, 2016, Higor Medeiros.

- [10] "Bringing MySQL to the web", <https://www.phpmyadmin.net/>, acedido em 21 de agosto, 2016,
- [11] "MySQL Workbench", <https://www.mysql.com/products/workbench/>, acedido em 21 de agosto, 2016
- [12] "PhpStorm", <https://www.jetbrains.com/phpstorm/features/>, acedido em 21 de agosto, 2016
- [13] "Trello: el administrador de proyectos definitivo", <http://dispersium.es/trello-el-administrador-de-proyectos-definitivo/>, acedido em 21 de agosto, Eduardo, 2016
- [14] "Primeiros passos - Noções Básicas de Git", <https://git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git>, acedido em 22 de agosto, 2016,
- [15] "15 Best Free PHP Frameworks", <http://beebom.com/best-free-php-frameworks/>, acedido em 22 de agosto, 2016,
- [16] "10 PHP Frameworks For Developers – Best Of", <http://www.hongkiat.com/blog/best-php-frameworks/>, acedido em 22 de agosto, 2016,
- [17] "Top 5 Best PHP Frameworks for 2016 To Become a Master Developer", <https://webandmobiletech.wordpress.com/2016/01/01/top-5-best-php-frameworks-for-2016-to-become-a-master-developer/>, acedido em 22 de agosto, 2016, Gregory Gillis
- [18] "Laravel", <https://laravel.com/>, acedido em 22 de agosto, 2016,
- [19] "UML Use Case", <http://www.uml-diagrams.org/use-case.html>, acedido em 26 de agosto, 2016,
- [20] "O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML", <http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>, acedido em 26 de agosto, 2016,
- [21] Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice, Second Edition" Addison Wesley, 2003
- [22] Roger S. Pressman, "Engenharia de Software, uma abordagem profissional, Sétima Edição"
- [23] Frank Buschmann, Regine Meunier, Hnas Rohnert, Peter Sommerlad, Michael Stak, "Pattern-oriented software architecture A sistem of patterns", JOHN WILEY & SONS, 2001

[24] Ian Sommerville, "Software Engineering ninth edition" pearson 2011

[25] PHP, <http://php.net/>, acedido em 5 de setembro, 2016

[26] Composer, Dependency Manager for PHP, <https://getcomposer.org/>, acedido em 7 de setembro, 2016.

[27] 6 truques de ECMAScript 6, <http://desenvolvimentoparaweb.com/>, acedido em 7 de setembro, 2016.

[28] Packagist The PHP Package Repository, <https://packagist.org>, acedido em 7 de setembro, 2016.

[29] Packagist, http://librosweb.es/libro/composer/capitulo_2/packagist.html, acedido em 7 de setembro, 2016.

[30] Laravel Collective, <https://laravelcollective.com>, acedido em 7 de setembro, 2016.

[31] Bitbucket, <https://www.atlassian.com/software/bitbucket>, acedido em 7 de setembro, 2016.

[32] "Usando iconos sin imágenes con Font Awesome"
<http://maquetando.com/recursos/fonts/usando-iconos-sin-imagenes-con-font-awesome/>,
acedido em 7 de setembro, 2016.

[33] Font Awesome, <http://fontawesome.io/>, acedido em 7 de setembro, 2016.

[34] Bootstrap, <http://getbootstrap.com/>, acedido em 7 de setembro, 2016.

[35] ¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB?
<http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>, acedido em 7 de setembro, 2016.

[36] Tutorial Vagrant 1: Qué es y cómo usarlo, <https://geekytheory.com/tutorial-vagrant-1-que-es-y-como-usarlo/>, acedido em 9 de setembro, 2016.

[37] Vagrant, <https://www.vagrantup.com/>, acedido em 9 de setembro, 2016.

[38] VirtualBox, <https://www.virtualbox.org/>, acedido em 9 de setembro, 2016.

[39] ¿Qué es DER (ERD)? <https://www.lucidchart.com/pages/es/qu%C3%A9-es-diagrama-entidad-relaci%C3%B3n>, acedido em 13 de setembro, 2016.

[40] José Carlos Cordeiro Martins [2007]. "Técnicas para gerenciamento de projetos de software"

[41] Fernando Alonso, Loic Martínes, Fco. Javier Segovia [2005] "Introducción a la ingeniería del software. Modelos de desarrollo de programas"

[42] Javascript a fondo, <http://www.desarrolloweb.com/javascript/#quees>, acedido em 18 de setembro, 2016.

[43] ¿Qué es AJAX? ¿Para qué sirve?, <http://www.ibrugor.com/blog/que-es-ajax-para-que-sirve/>, acedido em 18 de setembro, 2016.

[44] Introducción al HTML, <http://www.desarrolloweb.com/articulos/534.php>, acedido em 18 de setembro, 2016.

[45] Introdução ao HTML5, https://developer.mozilla.org/pt-PT/docs/Web/HTML/HTML5/Introdu%C3%A7%C3%A3o_ao_HTML5, acedido em 18 de setembro, 2016.

[46] Qué es y para qué sirve el lenguaje CSS (Cascading Style Sheets - Hojas de Estilo), http://www.aprenderaprogramar.com/index.php?option=com_content&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&Itemid=163, acedido em 18 de setembro, 2016.

[47] Manual de CSS 3 <http://www.desarrolloweb.com/manuales/css3.html>, acedido em 18 de setembro, 2016.

[48] Karl E. Wiegers, Creating a Software Engineering Culture, 2011.

[49] ALUNOS, Gestão de Alunos, <http://www.jpmabreu.com/cgi-bin/jpmcgi.jpm/pagina?id=software&num=1>, acedido em 19 de setembro, 2016.

[50] Soluções eSchooling <http://codevision.pt/solutions/>, acedido em 19 de setembro, 2016.

[51] praXis - Sistema de Gestão Escolar, <http://www.infinite.pt/solucoes/praxis.html>, acedido em 19 de setembro, 2016.

[52] inovar alunos, <http://www.inovar-mais.pt/produtos>, acedido em 19 de setembro, 2016.

[53] Introdução aos Processos de Software e o Modelo Incremental e Evolucionário, <http://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839>, acedido em 12 de novembro, 2016.

[54] Modelo espiral, <http://modeloesspiral.blogspot.pt>, acedido em 12 de novembro, 2016.

VI. ANEXOS

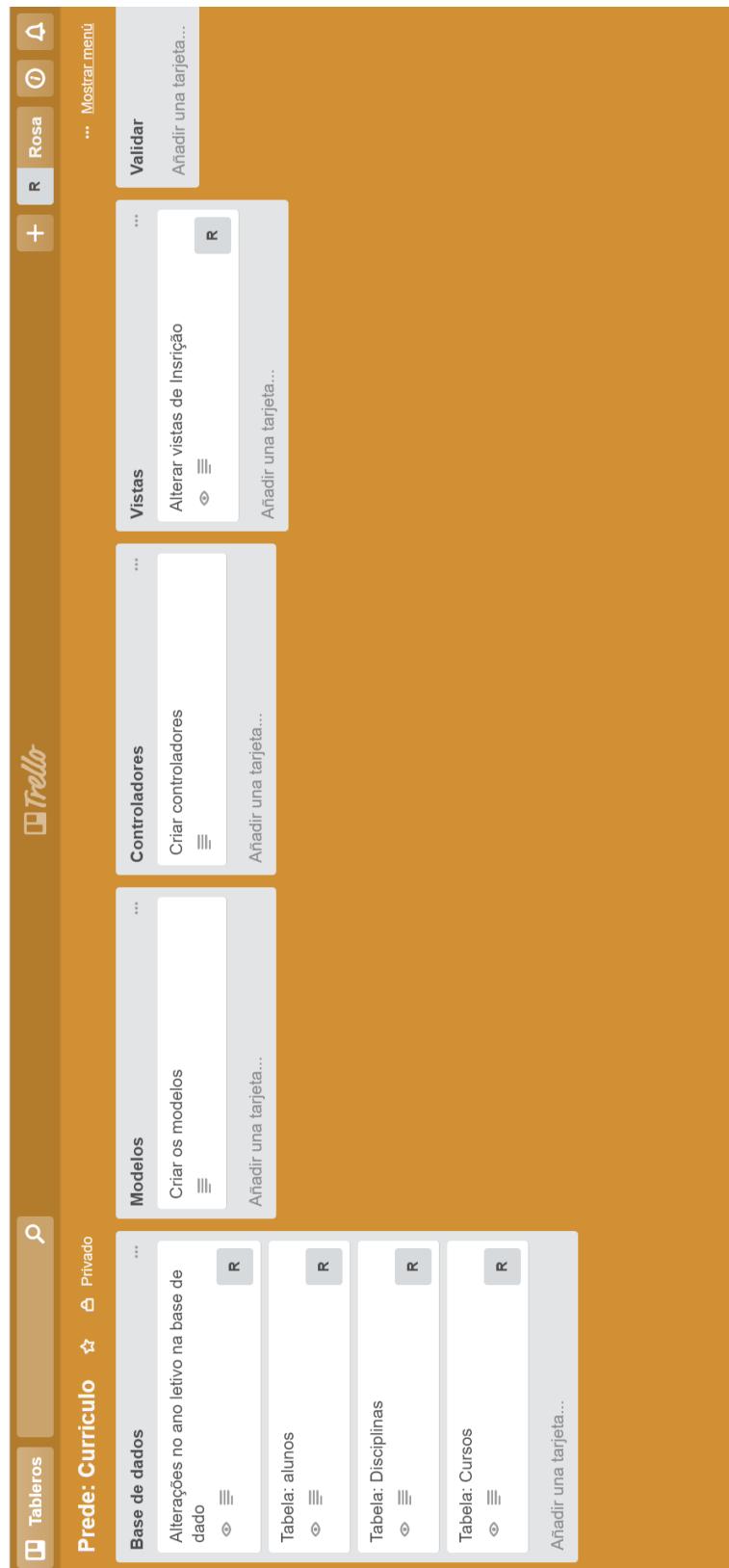
VI.1. ANEXO 1 – REPOSITÓRIO NO BITBUCKET

The screenshot shows a Bitbucket repository page for the project "academic-manager". The "Source" tab is selected. At the top right, there is a "New file" button. Below it, a dropdown menu shows "master" and a "Pull request" icon. The main area lists the repository's contents:

| File/Folder | Description | Last Commit |
|----------------|-----------------------|-------------|
| app | | 2016-08-27 |
| bootstrap | | 2016-08-27 |
| config | | 2016-08-27 |
| database | | 2016-08-28 |
| public | | 2016-08-27 |
| resources | | 2016-08-27 |
| storage | | 2016-08-27 |
| tests | | 2016-08-27 |
| .env.example | primer commit | 2016-08-27 |
| .gitattributes | primer commit | 2016-08-27 |
| .gitignore | añadir base de vistas | 2016-08-28 |
| artisan | primer commit | 2016-08-27 |
| composer.json | add datatables | 2016-09-13 |
| composer.lock | add datatables | 2016-09-13 |
| gulpfile.js | primer commit | 2016-08-27 |

Below the file list, there is a navigation bar with links: Source, Commits, Branches, Pull requests, Pipelines (NEW), Downloads, and a back arrow icon.

VI.2. ANEXO 2 – ORGANIZAÇÃO DE TAREFAS NO TRELLO



VI.3. ANEXO 3 – BASE DE DADOS EM PHPMYADMIN

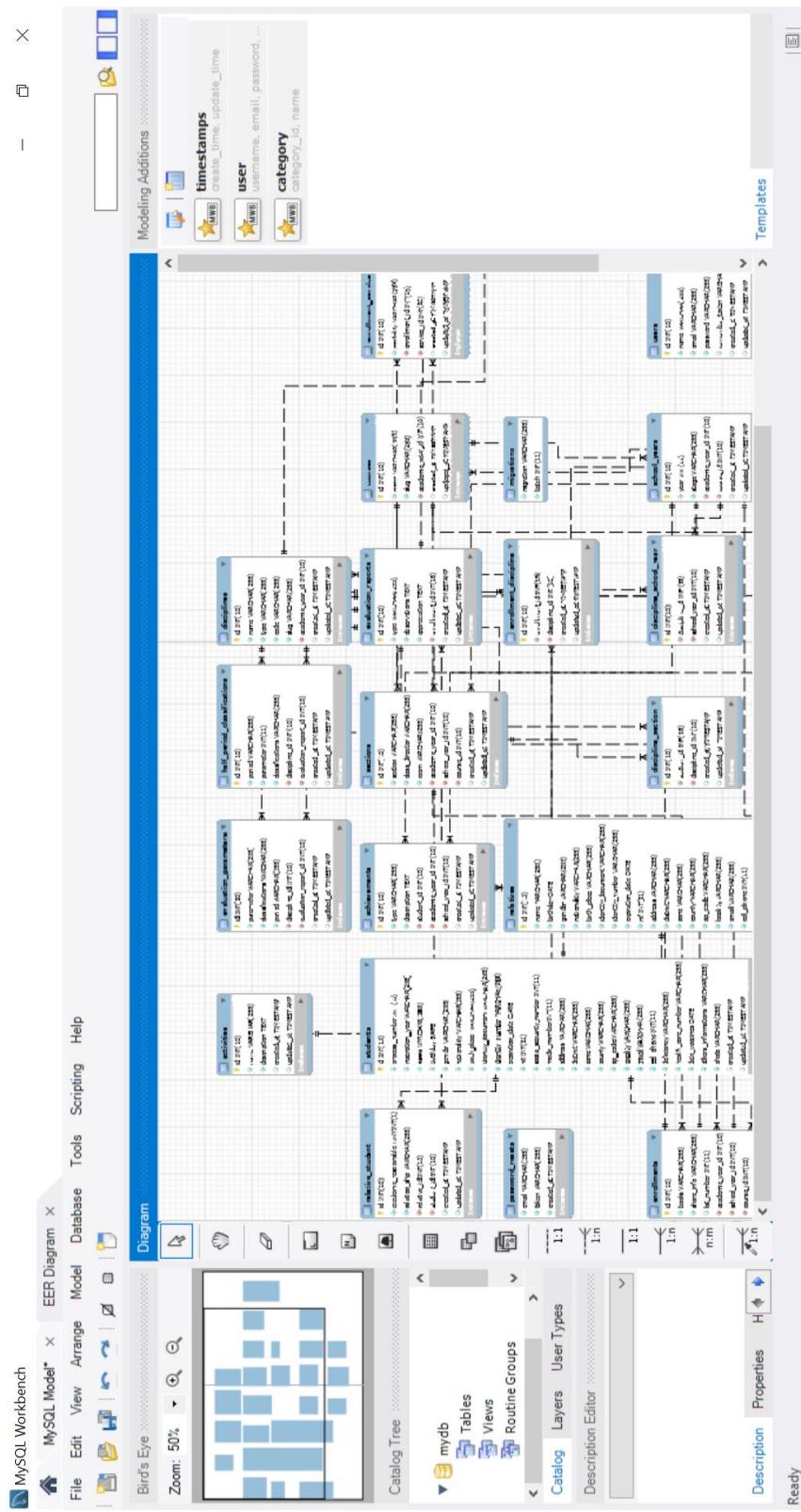
phpMyAdmin

Servidor: 192.168.10.10 » Base de Dados: academic-manager

Estrutura SQL Pesquisar Pesquisa por formulário Exportar Importar Operações Privilégios Mais

| Tabela | Ações | Registros | Tipo | Agrupamento (Collation) | Tamanho | Suspen |
|-----------------------------|--|-----------|--------|-------------------------|---------|--------|
| academic_years | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 4 | InnoDB | utf8_unicode_ci | 16 kB | |
| achievements | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 64 kB | |
| activities | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 16 kB | |
| activity_enrollment | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 48 kB | |
| courses | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 20 | InnoDB | utf8_unicode_ci | 32 kB | |
| disciplines | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 32 kB | |
| discipline_school_year | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 48 kB | |
| discipline_section | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 48 kB | |
| enrollments | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 96 kB | |
| enrollment_discipline | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 48 kB | |
| enrollment_service | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 48 kB | |
| evaluation_parameters | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 48 kB | |
| evaluation_reports | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 32 kB | |
| half_period_classifications | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 48 kB | |
| migrations | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 25 | InnoDB | utf8_unicode_ci | 16 kB | |
| password_resets | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 16 kB | |
| period_classifications | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 16 kB | |
| relatives | ★ Procurar Estrutura Pesquisar Inserir Limpa Elimina | 0 | InnoDB | utf8_unicode_ci | 16 kB | |
| Console word Resets | | | | | | |

VI.4. ANEXO 4 – DESENHO DA BASE DE DADOS EM MYSQL WORKBENCH



VI.5. ANEXO 5 – TERMINAL DE COMANDO *GIT BASH*



```
vagrant@homestead: ~/Code/academicic-cv
$ cd /C/laravel/workspace/Homestead/ && vagrant up && vagrant ssh
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'laravel/homestead' is up to date...
==> default: A newer version of the box 'laravel/homestead' is available! You cu
rrently
==> default: have version '0.4.4'. The latest is version '0.6.0'. Run
==> default: 'vagrant box update' to update.
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
==> default: Adapter 1: nat
==> default: Adapter 2: hostonly
==> default: Forwarding ports...
==> default: 80 (guest) => 8000 (host) (adapter 1)
==> default: 443 (guest) => 44300 (host) (adapter 1)
==> default: 3306 (guest) => 33060 (host) (adapter 1)
==> default: 5432 (guest) => 54320 (host) (adapter 1)
==> default: 50000 (guest) => 9000 (host) (adapter 1)
==> default: 50001 (guest) => 5001 (host) (adapter 1)
==> default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running pre-boot VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
==> default: SSH username: vagrant
==> default: SSH address: 127.0.0.1:2222
==> default: SSH auth method: private key
==> default: Warning: Remote connection disconnect. Retrying...
==> default: Warning: Remote connection disconnect. Retrying...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Setting hostname...
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
==> default: /vagrant => C:/laravel/workspace/Homestead
==> default: /home/vagrant/code => C:/laravel/workspace/php
==> default: Machine already provisioned, run 'vagrant provision' or use the '--provision'
==> default: flag to force provisioning. Provisioners marked to run always will still run.
We come to Ubuntu 14.04.4 LTS (GNU/Linux 3.19.0-25-generic x86_64)

* Documentation: https://help.ubuntu.com/
Last login: Sun Oct 27:49:35 2016 from 10.0.2.2
vagrant@homestead:~$ cd code
-bash: cd: code: No such file or directory
vagrant@homestead:~$ cd Code
vagrant@homestead:~/Code$ vagrant manager/
-bash: academic-manager/: is a directory
vagrant@homestead:~/Code$ cd academic-manager/
-bash: academic-manager/: is a directory
vagrant@homestead:~/Code$ cd academic-manager/
```

VI.6. ANEXO 6 – EXEMPLO DE UM CURRÍCULO ACADÉMICO GERADO PELO SISTEMA – PÁGINA 1/2



2014/2015

Âmbito: Projetos nacionais

Descrição: Participação no Projeto "A Empresa"

2015/2016

Âmbito: Concurso

Descrição: Participação nas "11 e 12 horas de gestão" do ISCTE

2015/2016

Âmbito: Distinções académicas

Descrição: Integração no quadro de honra no 1º período

2015/2016

Âmbito: Outros

Descrição: Participação do dia aberto da FCT

2015/2016

Âmbito: Outros

Descrição: Participação na semana da economia e gestão na UCP e visita à fundação Champalimaud

VI.7. ANEXO 7 – EXEMPLO DE UM CURRÍCULO ACADÉMICO GERADO PELO SISTEMA – PÁGINA 2/2



Notas:

| Disciplina | Nota |
|---------------------|-------------|
| Biologia | 10 |
| Biologia e Geologia | 10 |
| Economia C | 10 |
| Educação Física | 10 |
| Filosofia | 10 |
| Física e Química A | 10 |
| Inglês | 10 |
| Matemática A | 10 |
| Português | 10 |

Direção Pedagógica

Serviços Administrativos