

Imptel-Online

PROJETO DE DESENVOLVIMENTO DE SOFTWARE

Grupo 3

20221605 – Edja Tamara Monteiro Nicolau da Silva

20220631 – Joaquim Manuel Igreja Cláudio

20220481 – Laís da Silva Ferreira



SUMÁRIO

1. INTRODUÇÃO	3
2. PROBLEMÁTICA	4
• Acesso limitado às informações:	4
• Acesso Limitado ao Percorso Académico:	4
• Horários em Placards Físicos nas portas das salas:	4
• Pedidos de Documentação na Secretaria:	4
3. HIPÓTESE.....	6
• Centralização das informações académicas:	6
• Histórico académico acessível:	6
• Digitalização dos horários académicos:	6
• Otimização da solicitação de documentação académica:	7
4. OBJETIVOS.....	8
• Geral	8
• Específicos	8
5. PÚBLICO-ALVO	9
6. TAREFAS PRINCIPAIS	10
7. LEVANTAMENTO DE REQUISITOS	12
8. CASOS DE USO.....	14
9. ESPECIFICAÇÕES TÉCNICAS	16
• O Web site (Front-End)	16
▪ HTML 5:	16
▪ CSS 3:	16
▪ JavaScript:	16
• O servidor Web (Back-End)	16
• Base de Dados	16
10. ENGENHARIA DE SOFTWARE	17
11. SISTEMAS DISTRIBUÍDOS	20
12. INTELIGÊNCIA ARTIFICIAL	23
13. SEGURANÇA INFORMÁTICA.....	26
14. ANÁLISE COMPETITIVA.....	28

15. PLANO DE TRABALHO.....29

1. INTRODUÇÃO

A ideia do projeto é desenvolver uma aplicação web para uma **instituição do ensino secundário**, com o objetivo de marcar presença na internet e, através de uma abordagem tecnológica, propor uma solução integrada para os problemas típicos de uma escola desta categoria. A proposta deste portal é de concentrar os sistemas administrativo e pedagógico, bem como a comunicação e a gestão interna dos diversos elementos daquela instituição, nomeadamente, alunos, docentes e o secretariado.

Esta aplicação web será implementada no **Instituto Médio Privado de Tecnologias - IMPTEL**, uma escola angolana do ensino secundário técnico/profissional que conta com dois campi na província de Luanda, que atualmente faz recurso a sistemas e procedimentos obsoletos do ponto de vista tecnológico e administrativo.

Com este projeto, pretendemos centralizar e otimizar as formas de interação entre a instituição e os seus membros, promovendo uma gestão mais eficiente e uma comunicação mais eficaz, utilizando o portal de estudantes como a principal forma para centralizar todas as atividades e informações relevantes.

2. PROBLEMÁTICA

Através de um levantamento de requisitos realizado junto da instituição/cliente conseguimos identificar os seguintes problemas:

- **Acesso limitado às informações:** a ausência de um portal de estudante resulta na descentralização das informações, obrigando os alunos a recorrerem a diversos meios para obter as informações necessárias. Esta prática não aumenta apenas o esforço e o tempo necessário para localizar dados relevantes, como também expõe os utilizadores ao risco de encontrar informações imprecisas ou desatualizadas;
- **Acesso Limitado ao Percurso Académico:** a instituição disponibiliza as notas dos alunos de fixadas em vitrines, o que frequentemente pode resultar na perda de informação por parte dos estudantes. Este método limita a capacidade dos discentes de avaliar e monitorizar o seu percurso académico de forma eficaz. Sem acesso a um histórico consolidado de notas, os estudantes enfrentam dificuldades em realizar uma análise abrangente do seu desempenho. Esta lacuna compromete a capacidade dos discentes de identificar áreas de melhoria e de efetuar os ajustes necessários ao seu plano de estudos;
- **Horários em Placards Físicos nas portas das salas:** os horários das aulas são exibidos em placards físicos nas portas das salas, o que obriga discentes e docentes a fazer o registo das informações manualmente, seja através de fotografias ou anotações em papel. Quando ocorrem alterações de horários, salas de aula ou docentes, tais modificações não são imediatamente refletidas nos placards. Mesmo quando são atualizadas, essas mudanças podem não ser notadas por todos. Esta situação resulta frequentemente em informações desatualizadas, causando diversos problemas, tais como atrasos, falta de comparência às aulas, deslocamento para salas erradas e falta de preparação adequada para as aulas.
- **Pedidos de Documentação na Secretaria:** a solicitação de documentação, como declarações de estudante, é efetuada na

secretaria da instituição, onde os estudantes frequentemente enfrentam filas. Este processo pode resultar em atrasos na chegada às aulas ou na falta de comparência, uma vez que o tempo necessário para realizar tais procedimentos pode sobrepor-se ao horário das aulas. A espera na secretaria não só reduz o tempo disponível para atividades académicas e pessoais, como também pode gerar frustração.

3. HIPÓTESE

Para solucionar os diversos problemas anteriormente citados, apresentamos a aplicação Web '**Imptel - Online**' que servirá de ferramenta para aplicar as seguintes soluções:

- **Centralização das informações académicas:** O discente terá acesso a toda a informação necessária de forma unificada e atualizada, incluindo dados como preçário, contacto, localização, eventos e redes sociais, além de contar com o suporte de um chatbot para esclarecer dúvidas frequentes. Esta abordagem elimina a necessidade de procurar em vários locais, reduzindo o esforço e o tempo gasto na busca de informações. Além disso, minimiza o risco de encontrar informações imprecisas ou desatualizadas, proporcionando uma experiência mais eficiente para os discentes;
- **Histórico académico acessível:** com a implementação da aplicação, o discente poderá aceder ao seu histórico académico de forma prática, sem depender de vitrines ou ter que esperar pelo final trimestre para saber a sua avaliação final. Todas as avaliações serão disponibilizadas no portal pelos professores de cada disciplina, garantindo que os discentes tenham acesso às notas.
- **Digitalização dos horários académicos:** os horários estarão disponíveis em formato digital na aplicação, permitindo a visualização atualizada do horário contendo turma, número da sala e docentes. Qualquer alteração que seja feita ao longo do ano letivo será atualizada e refletida imediatamente na aplicação, reduzindo problemas com atrasos e deslocamentos indesejados.

- **Otimização da solicitação de documentação académica:** para melhorar o pedido de documentos, como declarações de estudante, a aplicação incluirá uma funcionalidade para a solicitação online. Esta abordagem eliminará a necessidade de enfrentar filas na secretaria e reduzirá o tempo de espera, permitindo que os discentes recebam a documentação necessária sem comprometer o horário das aulas.

4. OBJETIVOS

- **GERAL**

Implementar uma aplicação web denominada '**Imptel-Online**' com o objetivo de otimizar a gestão académica e administrativa(secretaria) da instituição de ensino. Esta aplicação proporcionará aos discentes, docentes e ao secretariado uma plataforma eficiente e acessível para a visualização e atualização de informações.

- **ESPECÍFICOS**

A proposta desta aplicação propõe-se a alcançar os seguintes objetivos:

1. Melhorar a interação entre os alunos, docentes e o secretariado;
2. Permitir que os alunos solicitem os documentos diretamente pelo portal de estudantes;
3. Permitir que os alunos tenham acesso as suas notas;
4. Disponibilizar os horários das aulas das turmas;
5. Facilitar o acesso a informações essenciais, como preçário, localização e horários de funcionamento;
6. Adicionar uma funcionalidade que permita gerar horários de aulas, para professores e alunos, com recurso a inteligência artificial.

5. PÚBLICO-ALVO

Idade dos Utilizadores: O nosso público-alvo compreende pessoas numa ampla faixa etária. Desde jovens adultos e adolescentes que façam parte da instituição, sendo eles estudantes, professores, secretários e a administração. Com isso, caracterizamos o nosso público-alvo como sendo pessoas com idades entre os 15 e os 60 anos de idade.

Motivação para Aceder ao Portal de Estudantes: Identificámos diversas motivações que levam a entidade escolar a aceder ao portal de estudantes, cada uma com questões diferentes.

Aluno: Os alunos são os que mais tiram vantagens do portal, nomeadamente, acesso ao horário, acesso a informações atualizadas, acesso a notas, pedido de documentos a partir da plataforma.

Docência: Inserção de notas, acesso a informações, acesso ao horário.

Secretariado: Lançamento de notas, atendimento ao aluno, inserção e atualização da informação.

6. TAREFAS PRINCIPAIS

Aluno

Efetuar login / Ter acesso ao que o portal tem para oferecer.

Realizar pedidos / Efetuar o pedido através do portal sem ter que esperar por longas filas.

Aceder ao horário / Ter acesso ao horário de forma digital e atualizada, evitando assim deslocações desnecessárias.

Ver notas / Ter acesso ao histórico de notas das disciplinas, e à sua média ponderada do curso.

Procurar informações / Ter informações atualizadas de eventos, preço, localização de forma mais fiável, sem precisar deslocar-se ou correr o risco de obter informações desatualizadas ou imprecisas.

Docência

Inserir notas / Lançar as notas no portal de estudantes de modo que os alunos as possam ver e evitar ter que transportar, sempre que requisitado, documentos onde constam as notas dos alunos.

Aceder ao horário / Ter acesso ao horário de forma digital e atualizada, evitando assim deslocações desnecessárias.

Procurar informações / Ter informações atualizadas de eventos, localização, de forma mais fiável, sem precisar deslocar-se ou correr o risco de obter informações desatualizadas ou imprecisas.

Secretariado

Lançar notas no portal de estudantes / Tornar as notas que os professores inserem visíveis, evitando solicitações constantes de notas.

Fornecer documentos requisitados pelos alunos / Fornecer documentos via email, evitando a aglomeração dos alunos na secretaria.

Atualizar informações no portal de estudantes / Manter as informações atualizadas, preservando a integridade da instituição.

Efetuar pagamentos/ Ter um histórico dos pagamentos efetuados e não efetuados, evitando informação inconsistente.

7. LEVANTAMENTO DE REQUISITOS

Através de uma entrevista realizada junto da instituição/cliente foi possível levantar os seguintes requisitos funcionais e não-funcionais:

FUNCIONAIS:

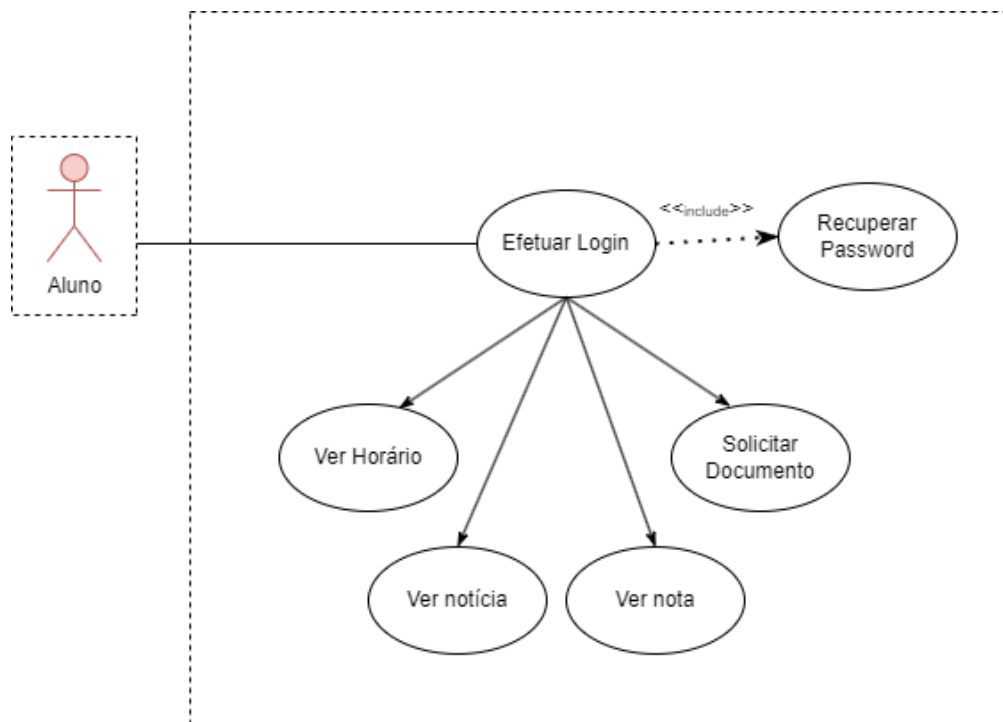
- Registrar utilizadores (*Aluno, Professor e Secretaria*).
- Criar pedidos de documentação (*Aluno*).
- Conceder documentação solicitada (*Secretaria*).
- Inserir, consultar e publicar notas (*Professor e Secretaria*).
- Criar, consultar e publicar horários de aulas.
- Registrar e consultar pagamentos de propinas e emolumentos (*Secretaria e Aluno**).
- Emitir fatura/recibo de pagamento (*Secretaria*).
- Criar, consultar e alterar turmas (*Secretaria e Aluno**).
- Associar/inscrever alunos em turmas (*Secretaria*).
- Obter análise de desempenho académico (*Aluno*).
- Publicar e obter o preçário do ano atual (*Secretaria e Aluno**).
- Apresentar dados estatísticos que auxiliem a tomada de decisão (gráficos e tabelas).
- Apresentar informações de contato e localização.

NÃO-FUNCIONAIS:

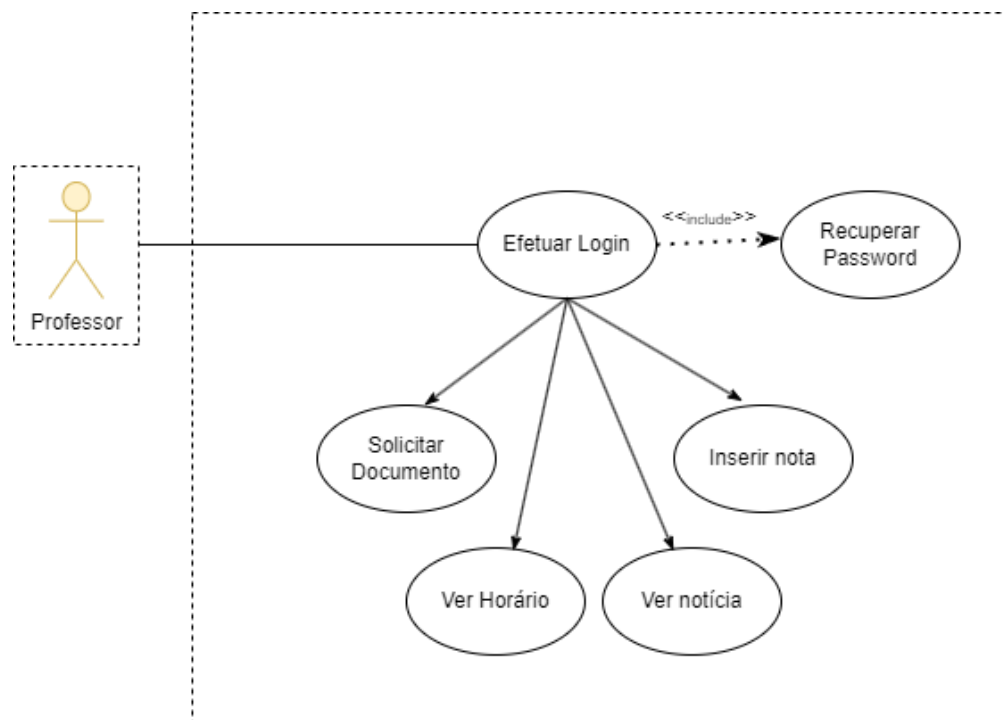
- Na UI, usar as cores que identificam a escola (azul-marinho e verde) e o logo académico.
- Interface simples e limpa, sem muitos elementos animados e coloridos para transmitir a seriedade e o rigor característicos da instituição.
- Criar o mínimo de ramificações/páginas possível para garantir uma navegação menos confusa e mais direta.
- Usar uma arquitetura de software extensível, para garantir uma App multiplataforma e a implementação futura de novas funcionalidades.

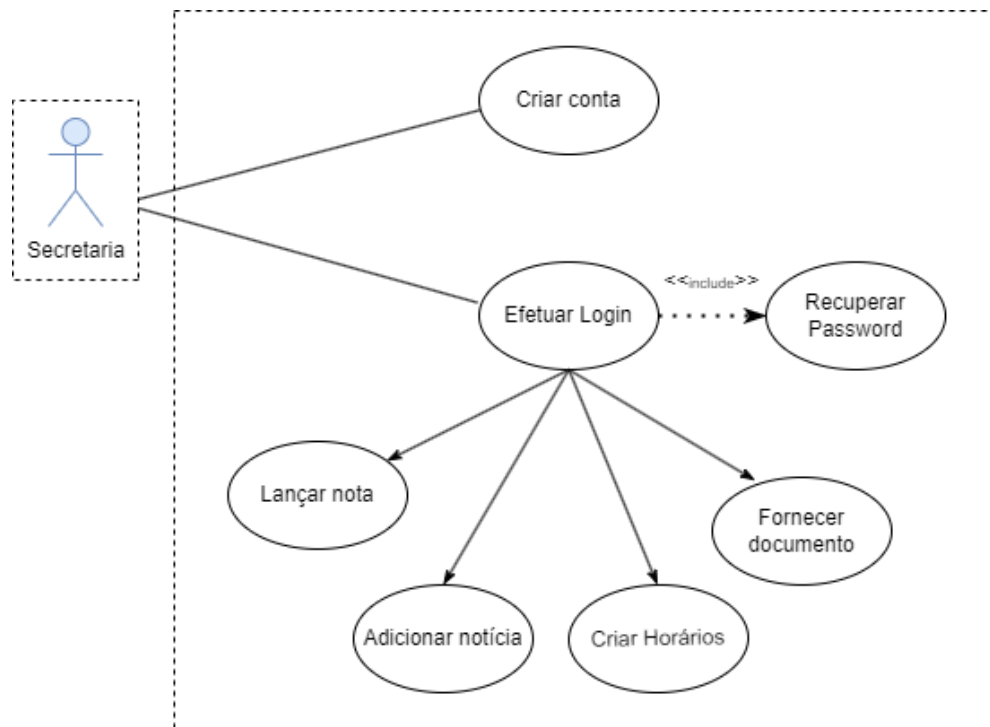
8. CASOS DE USO

Perfil: *Aluno*



Perfil: *Professor*



Perfil: Secretaria

9. ESPECIFICAÇÕES TÉCNICAS

Esta aplicação Web será desenvolvida de acordo com os requisitos e especificações apresentados pelos docentes das UCs envolvidas neste projeto académico.

Com isso, apresentam-se a seguir as tecnologias que darão suporte a esta solução:

- **O WEB SITE (FRONT-END)**

Toda a componente de visualização da Web App será desenvolvida utilizando as seguintes tecnologias:

- **HTML 5:** Estrutura (esqueleto) das páginas Web.
- **CSS 3:** Estilização e harmonização das páginas Web.
- **JavaScript:** Recursos interativos e de ligação com o Back-End.

Serão também utilizadas *Frameworks* e Bibliotecas das mesmas tecnologias, como o *React*.

- **O SERVIDOR WEB (BACK-END)**

O *server-side* será completamente desenvolvido recorrendo ao *framework .NET*.

Será desenvolvida uma **API RESTful** para lidar com as requisições HTTP provenientes dos web-clients (navegadores) e padronizar a integração entre o Front-End e o Back-End. A arquitetura do sistema será baseada em **microservices**.

- **BASE DE DADOS**

Para armazenar e garantir a persistência dos dados da nossa aplicação Web, será utilizado o **PostgreSQL** que é um sistema de gerenciamento de bases de dados relacional.

10. ENGENHARIA DE SOFTWARE

Modelo de desenvolvimento a ser utilizado para o projeto **Impstel-Online**, a melhor abordagem é a evolutiva, ou seja, o uso de metodologias ágeis (Scrum) combinadas com a arquitetura de microsserviços.

Característica	Modelo Tradicional	Metodologias Ágeis
Flexibilidade	Baixa, mudanças são difíceis após o início.	Alta, adapta-se facilmente a mudanças contínuas.
Tempo de Entrega	Longo; entrega somente ao final de todas as fases.	Curto, entregas parciais e incrementais constantes.
Documentação	Extensa e rigorosa.	Documentação leve, foco no cliente.
Controlo de Qualidade	Verificação ao final de cada fase.	Testes contínuos e ao longo de cada sprint.
Ajustes e Feedbacks	Pouca possibilidade de ajustes após as fases.	Feedback contínuo e ajustes a cada iteração.
Escalabilidade	Difícil de escalar com eficiência.	Fácil de escalar devido à modularidade dos microsserviços.

Vantagens das metodologias ágeis para o projeto

Flexibilidade: Microsserviços permitem que diferentes partes da aplicação sejam desenvolvidas e atualizadas independentemente, ideal para uma abordagem ágil;

Entrega Rápida: Entregar incrementos funcionais ao longo do tempo garante o cumprimento dos prazos estabelecidos pelo professor de projeto;

Adaptação às Mudanças: À medida que novos requisitos ou necessidades surgirem, será mais fácil ajustar a aplicação sem grandes impactos no cronograma ou no funcionamento do sistema como um todo.

Projeto X Produto

Produto: O produto é uma aplicação web chamada Imptel-Online, destinada a centralizar e otimizar a gestão académica e administrativa de uma instituição de ensino secundário. O objetivo principal é facilitar o acesso dos alunos e funcionários a informações importantes, como notas, horários e documentos, além de digitalizar processos administrativos.

Processo: O processo de desenvolvimento envolve a análise de requisitos, especificações técnicas, e a criação de uma aplicação web com componentes Front-End (HTML, CSS, JavaScript) e Back-End (.NET e API RESTful), utilizando um banco de dados relacional PostgreSQL para armazenar informações académicas e administrativas. O foco é na centralização e automação de tarefas administrativas, com melhorias contínuas baseadas em feedback dos utilizadores.

Modelo de dados

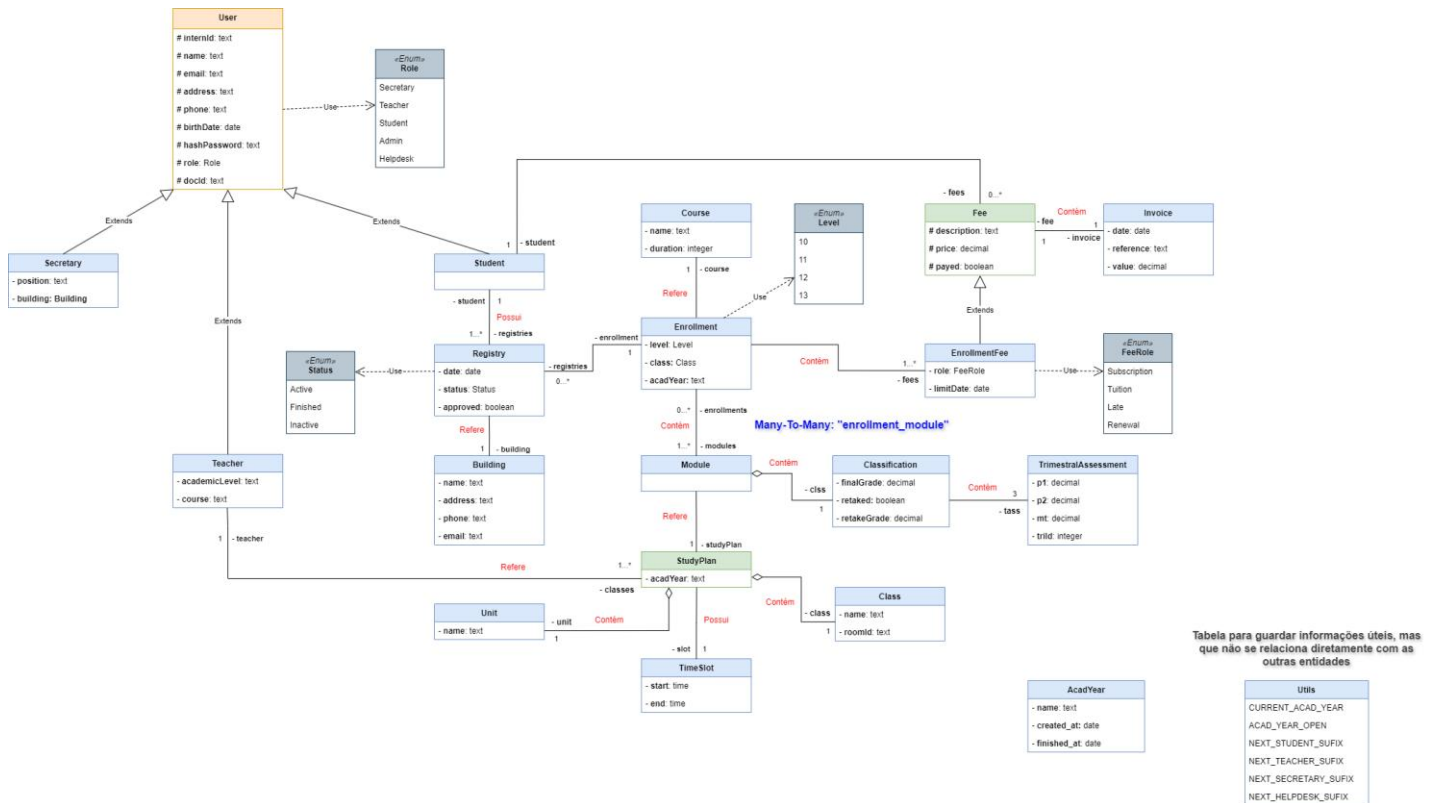


Fig.1 – Diagrama UML do modelo de dados.

Nota: poderá consultar o modelo de dados com melhor resolução através do link para o repositório no Github: [data-model.draw.png](#)

11. SISTEMAS DISTRIBUÍDOS

Este diagrama foi elaborado de acordo com os requisitos do Professor da UC de Sistemas Distribuídos. O diagrama ilustra uma arquitetura baseada em microsserviços numa infraestrutura em cloud, recorrendo a diversas tecnologias que serão abordadas na descrição do diagrama.

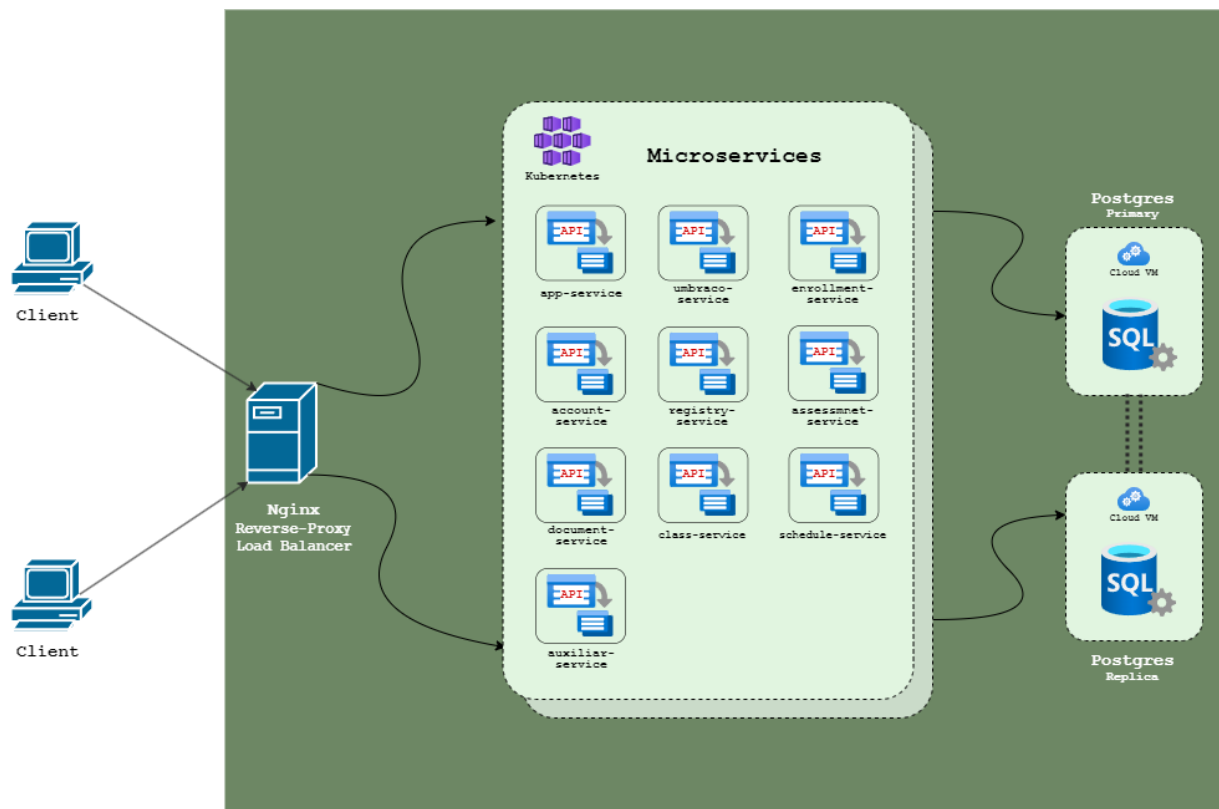


Fig.2 – Arquitetura do software baseada em sistemas distribuídos.

Nota: poderá consultar a arquitetura do software com melhor resolução através do link para o repositório no Github: [architecture-design.draw.png](https://github.com/your-repo-name/architecture-design.draw.png)

- **Cliente**

Os clientes representam os browsers, utilizados pelos utilizadores para aceder à aplicação (e.g. Chrome, Safari, Firefox, Edge).

- **Reverse Proxy**

O proxy reverso atua como intermediário entre os browsers (clientes) e os microserviços (API), desempenhando diversas funções importantes, tais como:

Redireciona as requisições HTTP para a API que detém o recurso solicitado, otimizando assim o uso dos recursos da aplicação. O proxy reverso funciona como gestor das requisições, encaminhando-as para o componente mais apto a lidar com elas no momento (**load balance**).

Armazena em cache os recursos de requisições recentes, permitindo que, em caso de repetição do pedido por parte de um cliente, o proxy possa respondê-lo diretamente sem realizar nova requisição a API (**cached**).

Adiciona uma camada de segurança, já que o endereço IP dos webserver nunca é revelado para o cliente, pois, cada requisição-resposta é filtrada pelo servidor proxy.

Nesta implementação utilizamos o Nginx como agente *reverse proxy* e *Load Balancer*.

- **Container**

Na arquitetura, os containers alojam os microserviços, expondo uma API bem definida que permite a utilização dos recursos de cada microserviço através de *endpoints*. Nesta implementação utilizados Docker e Kubernetes para orquestração de containers.

- **Máquina Virtual - Cloud**

As máquinas virtuais são utilizadas para hospedar os servidores de base de dados, garantindo um ambiente de execução consistente.

- **Microserviços**

Um microserviço corresponde à implementação das funcionalidades específicas de uma entidade. Na aplicação, existem vários microserviços e réplicas. Por exemplo:

As operações de registo, consulta e atualização dos dados da conta de um utilizador são realizadas por um microserviço específico.

A gestão de horários é assegurada por outro microserviço.

A componente aplicacional (Front/App) é também implementada como um microserviço.

Cada microserviço encontra-se isolado dentro de um container que comunica com o exterior através de uma API, e todos os microserviços são alojados numa máquina virtual na cloud. A gestão dos containers é realizada pelo Docker.

- **Sistema de Gestão de Bases de Dados (SGBD)**

A aplicação utiliza um Sistema de Gestão de Bases de Dados relacional para armazenar e persistir os dados, como os dados dos alunos e professores, por exemplo. Existem duas bases de dados: a primária e a réplica, sendo que esta última recebe uma cópia exata dos dados da base de dados principal em tempo real. Nesta implementação, utilizamos o SGDB PostgreSQL.

12. INTELIGÊNCIA ARTIFICIAL

Gestão de horários:

Na disciplina de Inteligência Artificial será aplicada a Gestão de Horários. A inteligência artificial será utilizada para gerar horários, recorrendo a mecanismos de busca e otimização. Será tido em conta que um professor pode lecionar em várias turmas, mas apenas numa de cada vez.

Na interface do aluno, este terá acesso ao horário da turma em que está inscrito, considerando feriados, férias e todas as eventuais interrupções, internas e externas, das aulas.

Na interface do professor, este terá acesso ao seu horário individual, indicando as disciplinas e as turmas em que dará aulas, considerando também feriados, férias e todas as eventuais interrupções, internas e externas, das aulas.

O resultado será uma tabela interativa apresentada na área individual de cada utilizador (*Professor e Aluno*).

Esta componente de IA justifica-se na capacidade de analisar e otimizar o uso dos recursos disponíveis, como salas de aula disponíveis e turmas, para maximizar a eficiência na gestão destes recursos e minimizar conflitos. Isso é especialmente importante quando um professor leciona em várias turmas, pois a IA deverá garantir que não exista sobreposição de aulas.

Resolução de Problemas com CSP

O código implementado é uma solução baseada em Constraint Satisfaction Problem (CSP) para o problema de gerenciamento do horário escolar. O objetivo é organizar horários para turmas, professores, salas e turnos, respeitando restrições específicas. Abaixo, são detalhadas as técnicas e heurísticas utilizadas no algoritmo.

Técnicas de IA para Otimização

Filter: Forward Checking

No código, a técnica de Forward Checking é implementada para reduzir os domínios das variáveis após cada atribuição de valor. Isso é feito pela função *forward_checking*, que atualiza os valores possíveis das variáveis ainda não atribuídas, removendo aqueles que violam as restrições.

Por exemplo, ao atribuir um horário a uma aula, os horários conflitantes são removidos dos domínios de outras aulas que compartilham o mesmo professor ou sala.

Search: Backtracking

O algoritmo principal utiliza backtracking para explorar o espaço de soluções. A função *backtrack* tenta atribuir valores às variáveis de forma recursiva. Caso uma atribuição viole alguma restrição, a busca retrocede e tenta uma alternativa.

Essa abordagem permite explorar todas as combinações possíveis até encontrar uma solução viável ou esgotar as possibilidades.

Cutoff

Um tempo limite para execução (cutoff) é configurado no algoritmo, interrompendo a busca se o tempo máximo permitido for atingido. Isso evita ciclos infinitos em cenários onde o problema não possui solução viável. Esse controle é implementado com base na função *time.perf_counter*.

Heurísticas para Seleção de Variáveis

MRV (Minimum Remaining Value)

A heurística de valores remanescentes mínimos é aplicada na função *select_unassigned_variable*, que prioriza a escolha de variáveis com o menor número de valores disponíveis em seu domínio. Isso reduz as chances de atribuições conflitantes nas etapas posteriores da busca.

Degree

Quando há empate entre variáveis com o mesmo número de valores no domínio, a heurística de grau é utilizada. Essa abordagem escolhe a variável com o maior número de restrições relacionadas a outras variáveis não atribuídas, maximizando o impacto positivo da escolha no restante do problema.

Heurística para Verificação de Consistência

AC-3 (Arc Consistency Algorithm 3)

A consistência entre variáveis é reforçada pelo algoritmo AC-3, implementado na função *ac3*. Este algoritmo garante que os valores nos domínios das variáveis permaneçam consistentes com as restrições, propagando mudanças sempre que uma revisão detecta inconsistências. A função *revise* é usada para ajustar os domínios durante esse processo.

Funcionamento do Algoritmo

Entrada de Dados: O problema é representado por variáveis (aulas), domínios (possíveis combinações de sala, professor, horário e dia) e restrições (horários dos professores, turmas, turnos, entre outros).

Restrições: São definidas funções específicas, como *no_conflict_class* e *no_conflict_teacher*, para garantir que não haja conflitos nos horários entre turmas ou professores.

Execução e Resultados: O algoritmo gera uma solução final organizada por turma e dia da semana. Caso uma solução não seja encontrada, mensagens indicam possíveis causas, como restrições conflitantes ou domínios excessivamente restritivos.

13. SEGURANÇA INFORMÁTICA

Para garantir a segurança do software Imptel-Online, os tópicos a seguir são baseados nos conteúdos da UC de Segurança Informática e contém as componentes de segurança presentes na aplicação.

Proteção contra SQL Injection

- **Verificação de Entradas:** Validação das entradas de dados fornecidas pelos utilizadores para evitar que comandos maliciosos sejam inseridos em consultas SQL.
- **Consultas Parametrizadas:** Querys parametrizadas garante que os dados do utilizador sejam tratados como parâmetros e não como código executável.

Criptografia

- **Proteção de Dados dos Utilizadores:** Cifra de dados sensíveis, como palavras-passe, para evitar a sua exposição em caso de violação de dados (hashing de palavras-passe). Armazenamento palavras-passe de forma segura, dificultando ataques de força bruta.

Reverse Proxy

- **Reduz a Superfície de Ataque:** Um reverse proxy atua como intermediário entre o cliente e o servidor interno, ocultando a infraestrutura do sistema.
- **Filtro de Tráfego de Rede:** Inspecciona e filtra o tráfego de rede, bloqueando acessos não autorizados ou maliciosos antes de atingirem o sistema principal.

Arquitetura Distribuída

- **Descentralização Generalizada:** Divisão de serviços em microserviços independentes reduzindo o impacto de falhas e melhor escalabilidade.
- **Redundância:** Garante a disponibilidade do sistema, mesmo em caso de falhas.

Princípios de Saltzer e Schroeder

- **Predefinições Seguras (Fail-Safe):** Falhar de forma segura garante que, em caso de falhas, o sistema não exponha informações sensíveis nem se torne vulnerável. Negar acesso por omissão.
- **Privilégios Mínimos:** Configurações de contas e processos com apenas as permissões estritamente necessárias para executar as suas funções.

Controlo de Acessos

- **Gestão de Acessos Baseada em Funções (RBAC):** Restringe o acesso aos recursos do sistema com base nas funções atribuídas a cada utilizador, garantindo que cada um aceda apenas ao necessário para a sua função.

14. ANÁLISE COMPETITIVA

Ferramentas	Portal de estudante Universidade Católica de Angola	Portal de estudante IADE	Portal de estudante Instituto Superior Politécnico de Tecnologias e Ciências
Conceito	Plataforma online de apoio aos estudantes	Plataforma online de apoio aos estudantes	Plataforma online de apoio aos estudantes
Experiência do utilizador	Média. Navegação não intuitiva. Software lento	Média. Navegação intuitiva, difícil acesso à informação	Média. Navegação não intuitiva. Software lento
Recursos oferecidos	Notas, solicitação de documentos/exames entres outros recursos	Horários, notas, informações sobre eventos, solicitação de documentos/exames entre outros recursos	Acesso ao conteúdo das aulas, pagamentos, acesso às notas
Qualidade do conteúdo	Pouca qualidade	Boa qualidade	Qualidade média

15. PLANO DE TRABALHO

Data	Tarefa	Concluído
13-09-2024	Brainstorming	✓
16-09-2024	Definição do problema e possíveis soluções	✓
18-09-2024	Levantamento de requisitos	✓
20-09-2024	Definição das tecnologias, especificações técnicas	✓
24-09-2024	Modelo de dados	✓
30-09-2024	Arquitetura do software	✓
05-10-2024	Proposta inicial do projeto	✓
12-10-2024	Cenários e personas	✓
15-10-2024	Guiões de teste	✓
20-10-2024	Mockups e interfaces	✓
20-11-2024	Desenvolvimento back-end	WIP
25-11-2024	Desenvolvimento front-end	WIP
30-12-2024	Relatório final	WIP