UNIVERSIDADE CATÓLICA DE SANTOS
Centro de Ciências Exatas, Arquitetura e Engenharia
Ciência da Computação e Sistemas de Informação

Augusto Emanuel Batista Novaes Santos
Gabriel dos Santos Bello Botelho
Guilherme Marques Tavares
Jean dos Santos Gomes da Silva
Pablo Luis dos Santos Alves
Rafaela Mendes Lomba Pinho

Desenvolvimento de Plataforma ALUMNI – Unisantos

Santos, SP
2023

Desenvolvimento de Plataforma ALUMNI – Unisantos

Pré-Projeto de extensão para o desenvolvimento de um sistema de informação para uma plataforma ALUMNI da comunidade UNISANTOS, apresentado como projeto de extensão dos cursos de Ciência da Computação e Sistemas de Informação.

Project Owner: Profª Me. Claudia Maria Sodero Salles

Santos, SP
2023

SUMÁRIO

1. HISTÓRICO DE REVISÕES

Data	Versão	Descrição	Autor
15/05/2023	1.0	Criação	Augusto Novaes, Guilherme Marques
18/05/2023	1.0	Revisão	Gabriel Botelho, Jean Gomes, Pablo Alves, Rafaela Mendes
22/05/2023	1.1	Criação	Pablo Alves
24/05/2023	1.1	Revisão	Guilherme Marques, Rafaela Mendes
02/06/2023	1.2	Adição Autenticação	Pablo Luis

2. INTRODUÇÃO

A Universidade Católica de Santos é uma instituição de ensino superior fundada em 1952, com o início do funcionamento de sua Faculdade de Direito. Após uma série de ampliações e novos cursos, surge em 1986 A Universidade Católica de Santos, a primeira universidade da região metropolitana da baixada santista.

A instituição tem como missão

> [...] formar cidadãos com base nos princípios da solidariedade, da justiça e do respeito aos direitos humanos, fortalecidos pela ética cristã e com competência profissional para atuar em uma realidade sócio-cultural heterogênea e sujeita a frequentes mutações. (UNISANTOS, 2022)

As atividades de ensino, pesquisa e extensão estão fortemente presentes em seus cursos, que trabalham as três essenciais vertentes do ensino superior de forma integrada e continuada.

Para a Universidade,

> [...] a socialização da produção acadêmica e de conhecimentos técnico-científicos acontece por meio de projetos e serviços oferecidos gratuitamente às comunidades interna e externa, oriundos dos cursos de graduação e programas de pós-graduação desenvolvidos por escritórios-modelos, agências experimentais e clínicas-escola. Mantém convênios com empresas públicas e privadas e organizações diversas para cooperação técnica e oferta de estágio aos discentes.(UNISANTOS, 2022).

É neste contexto que a Plataforma ALUMNI está situado. Uma Rede Alumni visa manter conectados aqueles que nunca deixaram de fazer parte da comunidade acadêmica, garantindo a colaboração contínua destes entre os próprios egressos e os alunos atuais. Em mais de vinte anos de existência, os cursos de Tecnologia da Informação (Ciência da Computação e Sistemas de Informação) da universidade já formaram mais de 500 profissionais, muitos deles extremamente bem colocados no mercado. A plataforma terá como objetivo reconectar todos estes egressos à instituição, permitindo uma constante crescente da rede de contatos, assim como manter atualizada a história dos cursos.

Permitirá também aos alunos matriculados a criação de uma rede de contatos baseados nas suas áreas de interesse, assim como poderá ser, no futuro, uma ferramenta para seleção de candidatos a vagas de estágio e emprego no setor produtivo da região.

Por fim, poderá permitir aos professores a criação de grupos de discussão temáticos, com engajamentos de alunos e ex-alunos que podem contribuir para a ampliação contínua do conhecimento.

3. OBJETIVO GERAL

Desenvolver uma plataforma computacional que funcione como uma rede comunitária para alunos e ex-alunos da Universidade Católica de Santos. Na primeira etapa, as funcionalidades de cadastro (alunos, ex-alunos, professores e gestores de cursos) serão o foco.

4. OBJETIVOS ESPECÍFICOS

- Levantamento de requisitos funcionais de cadastro para cada um dos atores da plataforma;
- Levantamento de requisitos não-funcionais de desenvolvimento e funcionamento da plataforma;
- Modelagem do banco de dados da funcionalidade de cadastro;
- Definição das informações de projeto para a implementação da plataforma;
- Desenvolvimento da plataforma;
- implantação da Plataforma.

5. ABORDAGEM METODOLÓGICA

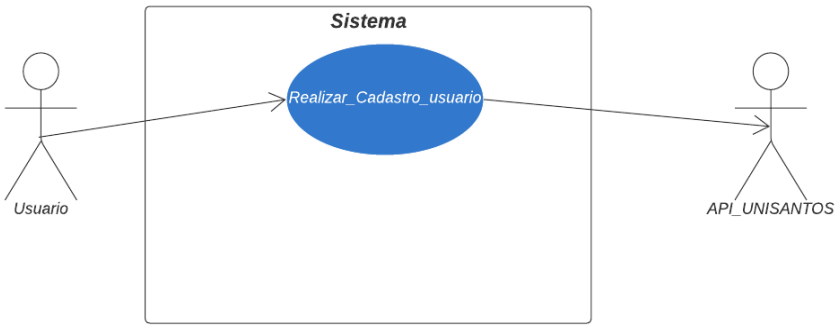
O estudo será conduzido com abordagem exploratória, considerando a natureza dos levantamentos de dados primários e secundários do processo de desenvolvimento de um sistema computacional e a necessidade de definição das tecnologias mais adequadas para o desenvolvimento da solução. Desta forma, o desenvolvimento tecnológico será conduzido em etapas, definidas da seguinte forma:

6. REQUISITOS E CASOS DE USO

RQ 1: Cadastro do usuário.

Descrição: O sistema deve ser capaz de armazenar o nome, senha, e-mail, data de nascimento, RM (registro de matrícula), cursos, e o tipo de usuário.

Nome	Descrição	Categoria	Desejável	Permanente
NF 1.1: Tempo de cadastro	O cadastro deve ser feito em até 3 segundos	Performance	(X)	



Descrição: O usuário deverá inserir seu RM. Com isso, o sistema irá efetuar uma busca para retornar e-mail institucional, nome, cursos e o telefone do usuário.

Ator: Usuário;

Pré-Condições: Possuir um vínculo com a universidade (Aluno, Ex-Aluno, Professor, Funcionário…);

Cenário Principal:

1. O usuário entrará com o RM;

2. O sistema irá retornar: e-mail institucional, nome do usuário, cursos do usuário e o telefone do usuário;

3. O usuário irá validar se as informações retornadas estão atualizadas;

4. O usuário irá informar a senha para login na plataforma;

5. O usuário clicará em cadastrar.

Cenário Alternativo:

1A. RM não localizado. O sistema deverá informar que o RM não foi localizado;

2A. As senhas não coincidem. O sistema informará que as senhas digitadas não coincidem;

3A. E-mail inválido. Caso o usuário altere o e-mail retornado ao informar o RM para um e-mail que não seja @unisantos, o sistema deverá informar que o e-mail é inválido;

4A. Algum campo em branco. O sistema não deverá cadastrar o usuário, informando que existem campos em branco e que os mesmos devem ser preenchidos;

5A. Alguma informação inválida. O sistema não deverá cadastrar o usuário, informando qual informação se encontra inválida;

6A. Termos de uso não aceitos. O sistema não deverá cadastrar o usuário, informando que os termos de uso não foram aceitos.

Figura: Processo 1, pesquisa de RM na API.

Figura: RM não localizado.

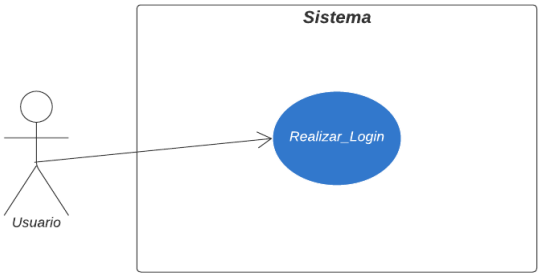
Figura: Processo 2, formulário de cadastro.

Figura: Processo 2, cadastro realizado com sucesso.

RQ 2: Login do usuário.

Descrição: O sistema deve possibilitar o login de usuários já cadastrados.

Nome	Descrição	Categoria	Desejável	Permanente
NF 2.1: Tela de login	A tela de login deve conter apenas dois campos	Usabilidade	(X)	
NF 2.1: Tempo de login	O login deve ocorrer em até 3 segundo	Desempenho	(X)	



Descrição: Após o usuário efetuar seu cadastro o sistema irá permitir que o mesmo faça seu login para poder utilizar a ferramenta;

Ator: Usuário

Pré-Condições: Possuir um cadastro

Cenário Principal:

1. O usuário irá informar seu e-mail institucional
2. O usuário irá digitar a sua senha
3. O usuário irá clicar no botão login

Cenário Alternativo:

3A. Dados inválidos. O sistema informará mostrará a seguinte mensagem: “E-mail e/ou senha inválidos”.

Figura: Tela de Login.

7 TECNOLOGIAS

Para o desenvolvimento do projeto de criação da plataforma ALUMNI da Universidade Católica de Santos, serão utilizadas várias tecnologias. A seguir, vou detalhar cada uma delas:

PHP (versão 8.0): O PHP é uma linguagem de programação amplamente utilizada para o desenvolvimento de aplicações web. A versão 8.0 é a mais recente, trazendo diversas melhorias e recursos. O PHP será a base do desenvolvimento da plataforma, permitindo a criação da lógica de negócio, manipulação de dados e interação com o banco de dados.

Composer: É uma ferramenta de gerenciamento de dependências para o PHP. Ele permite que você especifique as bibliotecas e pacotes necessários para o seu projeto e lida automaticamente com a instalação e atualização dessas dependências. Com o Composer, será possível versionar a aplicação e gerenciar as bibliotecas utilizadas no desenvolvimento da plataforma.

Doctrine: É uma biblioteca de persistência de dados para o PHP, que fornece um conjunto de ferramentas para mapear objetos PHP para estruturas de banco de dados relacionais. Com o Doctrine, será mais fácil e eficiente trabalhar com o banco de dados da aplicação, neste caso, o Postgres.

React: É uma biblioteca JavaScript amplamente utilizada para a criação de interfaces de usuário (UI). Ele permite o desenvolvimento de interfaces dinâmicas e interativas, facilitando a construção do front-end da aplicação ALUMNI. Além do React, também serão utilizados conceitos de HTML5 e CSS3 para estruturar e estilizar os elementos da interface. Além disso, o uso do TypeScript, uma linguagem superset do JavaScript, trará recursos adicionais de tipagem estática e aumento da produtividade no desenvolvimento.

Symfony (para o PHP 8): É um framework PHP que fornece um conjunto de componentes e ferramentas para desenvolvimento web. Ele facilita a criação de aplicações robustas e escaláveis, seguindo as melhores práticas de desenvolvimento. O Symfony será utilizado para aproveitar seus recursos e funcionalidades específicas para o PHP 8, que ainda estão em discussão.

PHPUnit: É um framework para realização de testes unitários no PHP. Ele fornece uma série de ferramentas e recursos para criar e executar testes

automatizados, garantindo a qualidade do código e facilitando a identificação de possíveis erros e problemas. Com o PHPUnit, será possível criar testes para verificar o correto funcionamento das diversas partes do código do projeto.

Selenium: É uma ferramenta para automação de testes em navegadores web. Ele permite criar testes automatizados que simulam a interação de um usuário com a aplicação em um navegador real. Com o Selenium, será possível criar testes que verifiquem o correto funcionamento da plataforma ALUMNI em diferentes navegadores, garantindo a funcionalidade e usabilidade em diferentes ambientes. Isso é especialmente importante considerando a variedade de navegadores utilizados pelos usuários.

O Selenium oferece recursos poderosos para interagir com elementos da página, preencher formulários, clicar em botões e validar resultados. Com essa ferramenta, será possível automatizar testes complexos, reduzindo a necessidade de testes manuais repetitivos e aumentando a eficiência do processo de desenvolvimento.

Em resumo, o Selenium será utilizado no projeto ALUMNI para elaborar testes automatizados que verifiquem a correta funcionalidade da plataforma em diferentes navegadores web. Isso permitirá identificar possíveis problemas e garantir uma experiência consistente e confiável para os usuários.

Dessa forma, com a utilização das tecnologias mencionadas, como PHP 8.0, Composer, Doctrine, React (com conceitos de HTML5, CSS3 e TypeScript), Symphony, PHPUnit, Selenium e Postgres, a plataforma ALUMNI da Universidade Católica de Santos será desenvolvida de maneira eficiente, robusta e com alta qualidade. Essas tecnologias oferecem recursos e ferramentas poderosas para atender aos requisitos do projeto, proporcionando uma experiência positiva tanto para os usuários atuais quanto para os ex-alunos e demais membros da comunidade acadêmica.

Para a base de dados, utilizaremos o Postgres, que é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto. Ele será utilizado como a base de dados da aplicação ALUMNI, armazenando todas as informações necessárias, como cadastros de usuários, informações dos cursos, entre outros dados relevantes. O Postgres é conhecido por sua confiabilidade, desempenho e recursos avançados, tornando-o uma escolha sólida para a plataforma.

Em resumo, a plataforma ALUMNI da Universidade Católica de Santos será desenvolvida utilizando PHP 8.0 como linguagem de programação, com o suporte do Composer para gerenciamento de dependências. A persistência de dados será feita com o auxílio do Doctrine, facilitando a interação com o banco de dados Postgres.

No lado do front-end, será utilizado o React para a criação da interface de usuário, aproveitando os conceitos de HTML5 e CSS3 para estruturação e estilização. Além disso, o TypeScript será utilizado para trazer recursos adicionais de tipagem estática.

Para garantir a qualidade do código, serão elaborados testes unitários com o PHPUnit, verificando o correto funcionamento de cada parte do código. Além disso, o Selenium será utilizado para a criação de testes automatizados, simulando a interação do usuário com a aplicação em diferentes navegadores.

Por fim, o Postgres será o banco de dados escolhido para armazenar todas as informações da plataforma, garantindo a confiabilidade e desempenho necessários.

Essas tecnologias foram selecionadas visando criar uma plataforma robusta, escalável e de alta qualidade, atendendo aos requisitos do projeto ALUMNI da Universidade Católica de Santos.

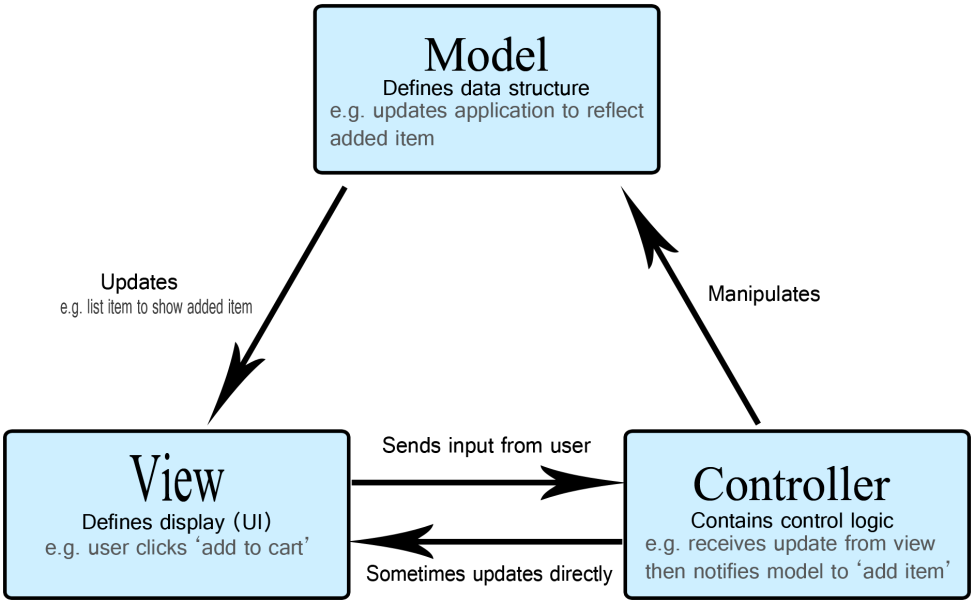
8 AUTENTICAÇÃO

Pensando nos aspectos de segurança da informação na aplicação, iremos utilizar alguns recursos para a elaboração da autenticação da aplicação, sendo eles:

- **Persistência de dados**: Será utilizado o PDO (PHP Data Object) para estabelecer uma conexão segura com o banco de dados. Além disso, iremos utilizar um ORM denominado Doctrine para efetuar a persistência de dados, de maneira com que possamos efetuar o controle e todo o tratamento de informações que serão direcionadas diretamente para nossa base de dados;
- **Criptografias**: Serão utilizados algoritmos de hash (como a própria criptografia MD5 do PHP) para criptografar os dados sensíveis do usuário, de maneira com que possamos fornecer uma proteção mais forte para a aplicação;
- **Json Web Token (JWT)**: E um padrão aberto (RFC 7519) para a criação de tokens de acesso baseados em JSON. É uma forma compacta e autônoma de transmitir informações entre partes de forma segura por meio de objetos JSON. Os tokens JWT são usados principalmente para autenticação e autorização em aplicativos web e serviços.

9 ARQUITETURA

O padrão MVC (Model-View-Controller): é amplamente utilizado em projetos PHP. Ele separa as preocupações do aplicativo em três componentes principais: Model (modelo), View (visão) e Controller (controlador). O modelo é responsável pela manipulação dos dados e regras de negócio, a visão é responsável pela apresentação dos dados ao usuário e o controlador gerencia a interação entre o modelo e a visão, tratando as requisições do usuário e coordenando as ações apropriadas.



10 CONCEITOS

Clean Code: refere-se a um estilo de escrita de código que é fácil de entender, manter e evoluir. Um código limpo segue princípios e boas práticas que visam tornar o código legível, conciso e de fácil compreensão. Isso inclui dar nomes significativos às variáveis, funções e classes, escrever funções curtas e bem estruturadas, evitar repetições desnecessárias, entre outros aspectos. O objetivo é produzir um código que seja de alta qualidade, fácil de dar manutenção e que possa ser compreendido por outros desenvolvedores. Como dito por Robert Cecil Martin “Tornar seu código legível é tão importante quanto torná-lo executável”.

Clean Architecture: É um estilo arquitetural que visa separar as preocupações e tornar o código fonte independente de frameworks, bancos de dados e tecnologias externas. A ideia central é definir camadas ou níveis de abstração dentro da aplicação, onde as camadas internas contêm a lógica de negócio e as camadas externas são responsáveis pela integração com o mundo externo, como interfaces de usuário, bancos de dados, APIs, entre outros. A Clean Architecture promove a manutenção do código, a testabilidade e a independência tecnológica.

Domain-Driven Design (DDD) :É uma abordagem de desenvolvimento de software que coloca o domínio da aplicação no centro do projeto. Ela enfatiza a colaboração próxima entre especialistas do domínio e desenvolvedores para criar um modelo de domínio rico e bem compreendido. O DDD propõe a criação de uma linguagem ubíqua compartilhada entre os especialistas do domínio e os desenvolvedores, e a utilização de padrões de design como Agregados, Entidades, Serviços de Domínio e Repositórios para representar e gerenciar o domínio de forma eficiente.

Test-Driven Development (TDD): É uma prática de desenvolvimento de software que enfatiza a escrita de testes automatizados antes da implementação do código. O ciclo do TDD consiste em escrever um teste que falhe, implementar a funcionalidade mínima para que o teste passe e, em seguida, refatorar o código para melhorar sua qualidade. Essa abordagem auxilia na criação de um código mais limpo, com maior cobertura de testes e menos bugs, além de promover um design de software mais modular e coeso.:

SOLID: É um conjunto de princípios de design de software que visam criar código de qualidade, flexível e de fácil manutenção. Cada letra do acrônimo SOLID representa um princípio específico:

- **Single Responsibility Principle** (Princípio da Responsabilidade Única): Uma classe deve ter apenas uma responsabilidade.
- **Open-Closed Principle** (Princípio Aberto-Fechado): Entidades de software devem ser abertas para extensão, mas fechadas para modificação.
- **Liskov Substitution Principle** (Princípio da Substituição de Liskov): Subtipos devem ser substituíveis por seus tipos base sem afetar a integridade do sistema.
- **Interface Segregation Principle** (Princípio da Segregação de Interfaces): Clientes não devem ser forçados a depender de interfaces que não utilizam.
- **Dependency Inversion Principle** (Princípio da Inversão de Dependência): Módulos de alto nível não devem depender de módulos de baixo nível. Ambos devem depender de abstrações.

Esses princípios SOLID fornecem diretrizes para escrever código limpo, modular e extensível. Eles promovem a separação de preocupações, o baixo acoplamento entre os componentes, a reutilização de código e a facilidade de manutenção.

Ao aplicar esses princípios, os desenvolvedores podem criar sistemas mais robustos, flexíveis e fáceis de dar manutenção. Eles ajudam a evitar a criação de código redundante, complexo e difícil de entender. Além disso, a adesão aos princípios SOLID facilita a realização de testes automatizados, pois o código se torna mais modular e isolado.

Em resumo, a adoção dos princípios SOLID, juntamente com as práticas de Clean Code, Clean Architecture, DDD e TDD, contribui para o desenvolvimento de software de alta qualidade, que atende aos requisitos de negócio, é facilmente mantido, testável e evoluível ao longo do tempo

REFERENCIAS

MARTIN, Robert Cecil: Clean Code, 2008.

MARTIN, Robert Cecil: The Clean Coder: A Code of Conduct for Professional Programmers, 2011.

MARTIN, Robert Cecil: Clean Architecture, 2017.

FOWLER, Martin, BECK, Kent: Refactoring Improving The Design Of Existing Code, 1999

EVANS, Eric: Domain-Driven Design: Tackling Complexity in the Heart of Software, 2003

REINALDO, Lucas: Conceitos isolados do Clean Code: Formatação Vertical e Horizontal do código. Disponível em: https://codejourney.com.br/conceitos-isolados-do-clean-code/ Acesso em 8 jun 2023.

JESUS, Daniel: Domain Driven Design. Disponível em: https://medium.com/beelabacademy/domain-driven-design-vs-arquitetura-em-camadas-d01455698ec5#:~:text=O%20conceito%20de%20DDD%20%C3%A9%20uma%20abordagem%20de%20modelagem%20de,neg%C3%B3cios%20que%20tratamos%20como%20dom%C3%ADnio. Acesso em 8 jun 2023.

WILLIANS, Wesley: O que é DDD – Domain Driven Design. Disponível em: https://fullcycle.com.br/domain-driven-design/ Acesso em 8 jun 2023

ZANETTE, Alyson: Clean Code: boas práticas para manter o seu código limpo!. Disponível em: https://becode.com.br/clean-code/ Acesso em 8 jun 2023

HIGOR: Introdução ao Padrão MVC. Disponível em: https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308 Acesso em 8 jun 2023

GUEDES, Marylene: O que é MVC?. Disponível em: https://www.treinaweb.com.br/blog/o-que-e-mvc#:~:text=O%20MVC%20sugere%20uma%20maneira,camada%20de%20controle%20(controller). Acesso em 8 jun 2023

LIMA, Clyson: O que é JWT?. Disponível em: https://www.treinaweb.com.br/blog/o-que-e-jwt . Acesso em 8 jun 2023

AYLAN: Como o JWT funciona. Disponível em: https://www.devmedia.com.br/como-o-jwt-funciona/40265 . Acesso em 8 jun 2023

ADRIANO, Thiago: JSON Web Token – Conhecendo o JWT na teoria e na prática. Disponível em: https://imasters.com.br/desenvolvimento/json-web-token-conhecendo-o-jwt-na-teoria-e-na-pratica . Acesso em 8 jun 2023