



Fundação Presidente Antônio Carlos de
Conselheiro Lafaiete
Engenharia de Computação



Relatório Técnico

Gerenciador de Minicursos

Everton Izabel da Silva

Conselheiro Lafaiete, 20 de outubro de 2021

Everton Izabel da Silva

Relatório Técnico

Gerenciador de Minicursos

Trabalho de conclusão de curso apresentado como parte das atividades para obtenção do título de Bacharel em Engenharia da Computação, da Fundação Presidente Antônio Carlos (FUPAC).

Conselheiro Lafaiete
20 de outubro de 2021

*“A mão queimada ensina melhor.
Depois disso o conselho sobre o fogo
chega ao coração.”– Gandalf - O Cinzento
(J.R.R. Tolkien)*

Resumo

A cada seis meses, a FUPAC de Conselheiro Lafaiete realiza uma mostra de robótica e automação. Na programação do evento existem diversas atividades onde a comunidade acadêmica pode participar e interagir. Uma destas atividades, e, tema deste trabalho se trata de uma oportunidade dada para alunos dos cursos de engenharia, onde eles podem ministrar minicursos, para os participantes do evento. Após a realização de diversas edições da mostra, ficou notório que o processo é limitado, por que o controle das inscrições é ocorre utilizando formulários do Google, com isso o acesso as informações relacionadas a adesão dos minicursos fica centralizada em apenas uma pessoa. Diante dos fatos apresentados está pesquisa tem por objetivo a prototipação de um sistema de gerenciamento de minicursos, para isso será necessário a criação da documentação relacionada a arquitetura do sistema, banco de dados e o desenvolvimento de um MVP.

Palavras-chaves: MVP, Laravel, React, Mysql, PHP, MVC.

Lista de ilustrações

Figura 1 – Mapeamento das três partes de uma aplicação para o MVC. [6]	5
Figura 2 – Objetos utilizados no MVC e suas interações. [6]	5
Figura 3 – Arquitetura proposta.	12
Figura 4 – Diagrama de entidade relacional do projeto	14
Figura 5 – Diagrama de casos de uso do sistema	14
Figura 6 – Janela de login do sistema	16
Figura 7 – Janela de falha no login do sistema	16
Figura 8 – Diagrama de sequencia do requisito de login	17
Figura 9 – Lista de minicursos e botão para criar minicurso	19
Figura 10 – Cadastro de minicurso etapa 1	20
Figura 11 – Cadastro de minicurso etapa 2	20
Figura 12 – Cadastro de minicurso erro de validação	21
Figura 13 – Diagrama de sequencia do requisito de cadastro de minicurso	22
Figura 14 – Lista de minicurso editar clique no item da lista.	24
Figura 15 – Cadastro de minicurso etapa 1	25
Figura 16 – Cadastro de minicurso etapa 2	25
Figura 17 – Cadastro de minicurso erro de validação	26
Figura 18 – Diagrama de sequencia do requisito de editar minicurso	27
Figura 19 – Lista de minicurso editar clique no item da lista.	29
Figura 20 – Diagrama de sequencia do requisito de deletar minicurso	30
Figura 21 – Lista de minicurso liberados para se inscrever.	32
Figura 22 – Interface de visualização e inscrição de minicurso.	32
Figura 23 – Diagrama de sequencia do requisito de visualizar minicurso	33
Figura 24 – Diagrama de sequencia do requisito de se inscrever em um minicurso	35
Figura 25 – Diagrama de sequencia do requisito de informar pagamento	37
Figura 26 – Diagrama de sequencia do requisito de cancelar inscrição	39
Figura 27 – Diagrama de sequencia do requisito de emitir certificado	41

Lista de abreviaturas e siglas

MVP	Mínimo viável possível;
JWT	<i>JSON Web Token</i> ;
APP	<i>Application</i> ;
GUI	<i>Graphical User Interface</i> ;
PHP	<i>Hypertext Préprocessor</i> ;
HTML	<i>HyperText Markup Language</i> ;
CGI	<i>Common Gateway Interface</i> ;
IIS	<i>Internet Information Services</i> ;
OOP	<i>Object-oriented programming</i> ;
XML	<i>Extensible Markup Language</i> ;
XHTML	<i>EXtensible HyperText Markup Language</i> ;
PDO	<i>PHP Data Objects</i> ;
ODBC	<i>Open Database Connectivity</i> ;
LDAP	<i>Lightweight Directory Access Protocol</i> ;
IMAP	<i>Internet Message Access Protocol</i> ;
SNMP	<i>Simple Network Management Protocol</i> ;
NNTP	<i>Network News Transfer Protoco</i> ;
POP3	<i>Post Office Protocol</i> ;
HTTP	<i>Hypertext Transfer Protocol</i> ;
WDDX	<i>Web Distributed Data eXchange</i> ;
MVC	<i>Model, View, Controller</i> ;
ORM	<i>Object Relational Mapper</i> ;
SQL	<i>Structured Query Language</i> ;

UI	<i>User Interface;</i>
DOM	<i>Document Object Model;</i>
API	<i>Application Programming Interface;</i>
SIGEV	Sistema Gerenciador de Eventos;

Sumário

1	INTRODUÇÃO	1
2	OBJETIVOS	2
2.1	Objetivo Geral	2
2.2	Objetivos Específicos	2
2.3	Produto final	2
3	METODOLOGIA	3
4	REVISÃO DE LITERATURA	4
4.1	Padrão MVC	4
4.2	PHP	6
4.2.1	O que php pode fazer?	6
4.3	Framework Laravel	8
4.3.1	Recursos	8
4.3.2	Vantagens	9
4.3.3	Desvantagem	9
4.4	React	9
4.4.1	Vantagem de react ao javascript puro	10
5	DESENVOLVIMENTO	12
5.1	Arquitetura	12
5.1.1	Arquitetura do sistema	12
5.1.2	API	12
5.1.3	Interfaces do Usuário	12
5.1.4	Levantamento de requisitos	13
5.1.4.1	Requisitos funcionais	13
5.1.4.2	Requisitos não funcionais	13
5.2	Diagrama de Entidade Relacional	14
5.3	Diagrama de casos de uso	14
5.3.1	Detalhamento das funcionalidades	14
5.3.1.1	UC0001: Logar no Sistema	15
5.3.1.1.1	Regra de negócio	15
5.3.1.1.2	Requisitos de tela	16
5.3.1.1.3	Diagramas	16
5.3.1.2	UC0002: Cadastrar minicursos	18

5.3.1.2.1	Regra de negócio	19
5.3.1.2.2	Requisitos de tela	19
5.3.1.2.3	Diagramas	19
5.3.1.3	UC0003: Editar minicursos	23
5.3.1.3.1	Regra de negócio	24
5.3.1.3.2	Requisitos de tela	24
5.3.1.3.3	Diagramas	24
5.3.1.4	UC0004: Deletar minicursos	28
5.3.1.4.1	Regra de negócio	28
5.3.1.4.2	Requisitos de tela	29
5.3.1.4.3	Diagramas	29
5.3.1.5	UC0005: Visualizar minicursos	31
5.3.1.5.1	Requisitos de tela	31
5.3.1.5.2	Diagramas	33
5.3.1.6	UC0006: Se inscrever	34
5.3.1.6.1	Diagramas	34
5.3.1.7	UC0007: Informar Pagamento	36
5.3.1.7.1	Requisitos de tela	36
5.3.1.7.2	Diagramas	36
5.3.1.8	UC0008: Remover Inscrição	38
5.3.1.8.1	Requisitos de tela	38
5.3.1.8.2	Diagramas	38
5.3.1.9	UC0009: Emitir Certificado	40
5.3.1.9.1	Requisitos de tela	40
5.3.1.9.2	Diagramas	40
6	CONSIDERAÇÕES FINAIS	42
6.1	Conclusão	42
	REFERÊNCIAS	43

1 Introdução

A cada seis meses, a FUPAC de Conselheiro Lafaiete realiza uma mostra de robótica e automação. Na programação do evento existem diversas atividades onde a comunidade acadêmica pode participar e interagir. Uma destas atividades, e, tema deste trabalho se trata de uma oportunidade dada para alunos dos cursos de engenharia, onde eles podem ministrar minicursos, para os participantes do evento. Após a realização de diversas edições da mostra, ficou notório que o processo é limitado, por que o controle das inscrições ocorre via *Google Forms* (<https://workspace.google.com/google/forms>), com isso o acesso as informações relacionadas a adesão dos minicursos fica centralizada em apenas uma pessoa.

Verificasse na literatura trabalhos como de Brandon Oliveira, onde é apresentado o SIGEV[1] que originou-se da necessidade de atender à demanda de soluções administrativas para a realização de encontros entre especialistas e acadêmicos. Trata-se de um sistema WEB que automatiza tarefas inerentes e repetitivas de um evento, como as inscrições, definição de atividades e emissão de certificados. Entretanto, alunos da UniEvangelica criaram o *EventPRO*[2] que originou-se ao suprir uma lacuna de gestão existente na promoção de Eventos Científicos por instituições de ensino Superior, o objetivo deste projeto foi desenvolver um sistema de software que auxilie a organização de todas as fases de um evento científico, desde a divulgação até a emissão de certificados. Embora as propostas tenham funcionalidades semelhantes, não foram encontrado documentação ou solução para replicação do projetos.

Entretanto, diante dos fatos apresentados está pesquisa tem por objetivo a prototipação de um sistema de gerenciamento de minicursos, onde professores e alunos poderão gerenciar os minicursos em tempo real. Portanto, será necessário elaborar toda documentação pertinente a arquitetura sistema para que após isso seja iniciada a criação da estrutura dos projetos *backend* e *frontend* do MVP.

Sendo assim, para atingir os objetivos propostos este relatório técnico foi organizado em cinco sessões: Objetivos, Metodologia, Revisão da Literatura, Desenvolvimento e Considerações Finais.

2 Objetivos

Nesta sessão é apresentado os objetivos gerais, específicos e final do projeto.

2.1 Objetivo Geral

Desenvolver um MVP que atenda aos requisitos funcionais e não funcionais de gerenciamento de minicursos na Universidade.

2.2 Objetivos Específicos

Nesta sessão está listada as etapas previstas no desenvolvimento do projeto.

- Modelar banco de dados da aplicação.
- Criar um `template` base para o *frontend*
- Criar estrutura e sistema `de login` no *backend*
- Desenvolver todas as funcionalidades do MVP.
- Publicar MVP para teste de bancada.

2.3 Produto final

O produto final deste projeto é um MVP onde alunos poderão se inscrever em minicursos oferecidos por outros alunos.

3 Metodologia

O procedimento metodológico adotado para nortear este estudo fundamenta-se em um método de pesquisa bibliográfica, em que foi adotado um protocolo de revisão sistemática para criar uma introdução geral das tecnologias que foram utilizadas no projeto.

A definição do procedimento metodológico é uma etapa fundamental que sucede a fase de definição dos objetivos gerais e específicos da pesquisa. Ele descreve o caminho para se atingir o objetivo de estudo proposto. Assim, deverá apresentá-lo de maneira coerente, clara e objetiva.

O estudo disponibilizado na plataforma scielo[3] descreve a pesquisa bibliográfica como um procedimento metodológico que fornece ao pesquisador possibilidades na busca de soluções para seu problema de pesquisa. Este tipo de pesquisa é apropriada, pois, uma vez que o pesquisador tem acesso a inúmeros repositórios de fontes científicas com baixo ou nenhum custo. Contudo, relatam ainda que o pesquisador deve ter distinção com a fidedignidade e validade científica das informações sob o risco de incorrer em possíveis incoerências e contradições causadas por publicações de baixa credibilidade. [4]. Já [5] *apud* [4] especifica que a pesquisa bibliográfica pode “fornecer instrumental analítico para qualquer outro tipo de pesquisa, mas também pode esgotar-se em si mesma. Essa afirmação indica que a pesquisa bibliográfica pode não somente acompanhar e complementar outros tipos de pesquisa, mas também operar como um método independente [4].

Visto que o projeto tem um foco estabelecido de desenvolvimento de um MVP e que as tecnologias escolhidas para a sua concepção foram escolhidas por possuir uma comunidade ampla e serem compatíveis com os ambientes da universidade, a revisão bibliográfica proposta para o estudo iniciou-se com a seleção dos principais repositórios científicos relacionados as tecnologias que são utilizadas no projeto. Portanto, por meio pesquisas na máquina de busca do *Google Scholar* (<https://scholar.google.com.br/>) identificou-se os principais acervos digitais relacionados a *Arquitetura de Software* e as tecnologias de *Desenvolvimento Web* e *Desenvolvimento Mobile*.

Após a leitura das publicações mais relevantes selecionadas nesta primeira etapa, o desenvolvimento do projeto inicia pela etapa de arquitetura, onde é estabelecido os requisitos funcionais e não funcionais do projeto. Partindo da definição dos requisitos é iniciado a criação da documentação do projeto, ou seja diagramas de casos de uso, sequencia, entidades, funcionalidades. Em seguida é iniciado o desenvolvimento das aplicações *backend* e *frontend*.

4 Revisão de Literatura

Neste capítulo será apresentada uma revisão bibliográfica nos principais repositórios científicos acerca das principais tecnologias utilizadas para o desenvolvimento do MVP do gerenciador de minicursos da Universidade.

4.1 Padrão MVC

O MVC é utilizado em muitos projetos devido a arquitetura que possui, o que possibilita a divisão do projeto em camadas muito bem definidas. Cada uma delas, o Model, o Controller e a View, executa o que lhe é definido e nada mais do que isso [6].

A utilização do padrão MVC traz como benefício o isolamento das regras de negócios da lógica de apresentação, que é a interface com o usuário. Isto possibilita a existência de várias interfaces com o usuário que podem ser modificadas sem a necessidade de alterar as regras de negócios, proporcionando muito mais flexibilidade e oportunidades de reuso das classes [6].

Uma das características de um padrão de projeto é poder aplicá-lo em sistemas distintos. O padrão MVC pode ser utilizado em vários tipos de projetos como, por exemplo, desktop, web e mobile [6].

A comunicação entre interfaces e regras de negócios é definida através de um controlador, que separa as camadas. Quando um evento é executado na interface gráfica, como um clique em um botão, a interface se comunicará com o controlador, que por sua vez se comunica com as regras de negócios [6].

Imagine uma aplicação financeira que realiza cálculos de diversos tipos, como os de juros. Você pode inserir valores para os cálculos e também escolher que tipo de cálculo será realizado. Isto tudo é feito pela interface gráfica, que para o modelo MVC é conhecida como View. No entanto, o sistema precisa saber que você está requisitando um cálculo, e para isso, terá um botão no sistema que quando clicado gera um evento [6].

Este evento pode ser uma requisição para um tipo de cálculo específico como o de juros simples ou juros compostos. Fazem parte da requisição os valores digitados no formulário e a seleção do tipo de cálculo que o usuário quer executar sobre o valor informado. O evento do botão é como um pedido a um intermediador (Controller) que prepara as informações para então enviá-las para o cálculo. O controlador é o único no sistema que conhece o responsável pela execução do cálculo, neste caso, a camada que contém as regras de negócios. Esta operação matemática será realizada pelo Model assim que ele receber um pedido do Controller [6].

O Model realiza a operação matemática e retorna o valor calculado para o Controller, que também é o único que possui conhecimento da existência da camada de visualização. Tendo o valor em mãos, o intermediador o repassa para a interface gráfica que exibirá para o usuário. Caso esta operação deva ser registrada em uma base de dados, o Model se encarrega também desta tarefa [6].

O MVC inicialmente foi desenvolvido no intuito de mapear o método tradicional de entrada, processamento, e saída que os diversos programas baseados em GUI utilizavam. No padrão MVC, teríamos então o mapeamento de cada uma dessas três partes para o padrão MVC conforme ilustra a Figura abaixo [6].

Input ► Processing ► Output
Controller ► Model ► View

Fig. 1. Mapeamento das três partes de uma aplicação para o MVC. [6]

A Figura abaixo demonstra que a entrada do usuário, a modelagem do mundo externo e o feedback visual para o usuário são separados e gerenciados pelos objetos Modelo (Model), Visão (View) e Controlador (Controller) [6].

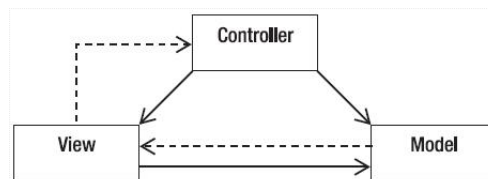


Fig. 2. Objetos utilizados no MVC e suas interações. [6]

Explicando cada um dos objetos do padrão MVC tem-se:

- Primeiramente o controlador (Controller), que interpreta as entradas do mouse ou do teclado enviadas pelo usuário e mapeia essas ações do usuário em comandos que são enviados para o modelo (Model) e/ou para a janela de visualização (View) para efetuar a alteração apropriada [6];
- Por sua vez, o modelo (Model) gerencia um ou mais elementos de dados, responde a perguntas sobre o seu estado e responde a instruções para mudar de estado. O modelo sabe o que o aplicativo quer fazer e é a principal estrutura computacional da arquitetura, pois é ele quem modela o problema a ser resolvido [6];
- Por fim, a visão (View) gerencia a área retangular do display e é responsável por apresentar as informações para o usuário através de uma combinação de gráficos e textos. A visão não sabe nada sobre o que a aplicação está atualmente fazendo, pois tudo que ela realmente faz é receber instruções do controle e informações do modelo e então exibi-las. A visão também se comunica de volta com o modelo e com o controlador para reportar o seu estado [6];

Portanto, a principal ideia do padrão arquitetural MVC é a separação dos conceitos - e do código. O MVC é como a clássica programação orientada a objetos, ou seja, criar objetos que escondem as suas informações e como elas são manipuladas e então apresentadas em uma interface simples para o mundo. Entre as diversas vantagens do padrão MVC estão a possibilidade de reescrita da GUI ou do Controller sem alterar o modelo, reutilização da GUI para diferentes aplicações com pouco esforço, facilidade na manutenção e adição de recursos, reaproveitamento de código, facilidade na manutenção do código sempre limpo [6].

4.2 PHP

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

Em vez de muitos comandos para mostrar HTML (como acontece com C ou Perl), as páginas PHP contém HTML em código mesclado que faz "alguma coisa"(neste caso, mostra "Olá, eu sou um script PHP!"). O código PHP é delimitado pelas instruções de processamento (tags) de início e fim `<?php` e `?>` que permitem que você entre e saia do "modo PHP"[7].

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 era o código fonte. Você pode inclusive configurar seu servidor web para processar todos os seus arquivos HTML com o PHP, e então não há como os usuários dizerem o que você tem na sua manga [7].

4.2.1 O que php pode fazer?

Qualquer coisa. O PHP é focado principalmente nos scripts do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer, como coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas o PHP pode fazer muito mais [8]. Existem três áreas principais onde os scripts PHP são usados:

- Scripts no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para isto funcionar: o interpretador do PHP (CGI ou módulo do servidor), um servidor web e um navegador web. Você precisa rodar o servidor web conectado a uma instalação do PHP. Você pode acessar os resultados de seu programa PHP com um navegador web, visualizando a página

PHP através do servidor web. Tudo isso pode rodar na sua máquina pessoal se você estiver apenas experimentando programar com o PHP [8].

- Scripts de linha de comando. Você pode fazer um script PHP para executá-lo sem um servidor ou navegador. A única coisa necessária é o interpretador PHP. Esse tipo de uso é ideal para script executados usando o cron (Unix, Linux) ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto simples [8].
- Escrever aplicações desktop. O PHP provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas se você conhece bem o PHP, e gostaria de usar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, você pode usar o PHP-GTK para escrever programas assim. Você também tem a possibilidade de escrever aplicações multi-plataformas desse jeito. O PHP-GTK é uma extensão do PHP, não disponibilizada na distribuição oficial [8].

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, macOS, RISC OS e provavelmente outros. O PHP também tem suporte à maioria dos servidores web atualmente. Isso inclui o Apache, o IIS e muitos outros. E isso inclui qualquer servidor web que possa utilizar o binário FastCGI do PHP, como o lighttpd e o nginx. O PHP trabalha tanto como módulo quanto como um processador CGI [8].

Com o PHP, portanto, você tem liberdade de escolha de sistema operacional e de servidor web. Além disso, você pode escolher entre utilizar programação estruturada ou programação orientada a objeto (OOP), ou ainda uma mistura das duas [8].

Com PHP você não está limitado a gerar somente HTML. As habilidades do PHP incluem geração de imagens, arquivos PDF e até animações Flash (utilizando libswf e Ming) criados dinamicamente, on the fly. Você pode facilmente criar qualquer texto, como XHTML e outros arquivos XML. O PHP pode gerar esses arquivos e salvá-los no sistema de arquivos, em vez de mostrá-los em tela, formando um cache no lado do servidor para seu conteúdo dinâmico [8].

Uma das características mais fortes e mais significativas do PHP é seu suporte a uma ampla variedade de banco de dados. Escrever uma página web consultando um banco de dados é incrivelmente simples usando uma das extensões específicas de banco de dados (por exemplo, mysql), ou usando uma camada de abstração como o PDO ou conectar a qualquer banco de dados que suporte o padrão "Open Database Connection" usando a extensão ODBC. Outros bancos de dados podem utilizar cURL ou sockets, como o CouchDB [8].

O PHP também tem suporte para comunicação com outros serviços utilizando

protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (no Windows) e incontáveis outros. Você também pode abrir sockets de rede e interagir diretamente usando qualquer outro protocolo. O PHP também suporta o intercâmbio de dados complexos WDDX, utilizado em virtualmente todas as linguagens de programação para web. Falando de comunicação, o PHP implementa a instanciação de objetos Java e os utiliza transparentemente como objetos PHP [8].

O PHP tem recursos úteis para processamento de texto, incluindo expressões regulares compatíveis com Perl (PCRE), e muitas outras extensões e ferramentas para analisar e acessar documentos XML. O PHP padroniza todas as extensões XML a partir da base sólida da libxml2, além de estender o conjunto de recursos adicionando suporte a SimpleXML, XMLReader e XMLWriter [8].

4.3 Framework Laravel

O Laravel é um framework, que facilita o desenvolvimento de uma aplicação e, sem dúvida, sua utilização poupa tempo e custos para quem o utiliza. É possível dizer que ele é um conjunto de bibliotecas que quando utilizadas em conjunto, criam uma base completa e segura para a criação de outras aplicações. Levando isso em consideração é possível dizer que com a abstração que ele oferece se gaste mais tempo nos problemas que precisam ser resolvidos do que em arquitetura e configurações [9].

4.3.1 Recursos

Dentre os diversos recursos do Laravel, podemos citar como principais:

- Padrão MVC
- Templates Blade
- Módulo de autenticação
- ORM

O Laravel trabalha com o padrão MVC e possui um sistema de template que facilita a criação da camada de visualização de dados. Com **está** camada, é possível criar páginas simples e intuitivas de forma rápida e eficaz. Dentre alguns dos recursos do Blade, se destacam: Herança de layouts, sistema de tags, seções e uso de código PHP nos templates [9].

Também possui um módulo de autenticação completo que provê todos os recursos para sua implementação, como: Autenticação de usuários, autorização de usuários, recuperação de senhas, logout, controle de sessão e cadastro de usuários [9].

E por fim o recurso mais poderoso o ORM que permite que não seja preciso criar códigos SQL para manipular ou criar tabelas no Banco de Dados. Todo processo é feito utilizando código PHP que, posteriormente, será convertido em instruções SQL. Implementa o padrão Active Record, onde cada model da aplicação representa uma tabela no banco de dados [9].

4.3.2 Vantagens

Como o laravel é um framework consolidado e com uma ampla comunidade ele agrega diversos benefícios a um projeto como pode ser visto abaixo.

- Redução de custos;
- Redução no tempo de desenvolvimento;
- Concentração nos problemas ao invés de na arquitetura;
- Redução de tempo e custo em manutenção:

Fatoração de aspectos comuns a várias aplicações;

Uso de herança permite corrigir todas as aplicações com a troca de uma classe mãe;

- Melhor consistência e compatibilidade entre aplicações;
- Alavancagem do conhecimento de especialistas;
- Documentação completa;
- Comunidade ativa;
- Amplamente utilizado;
- Gratuito.

4.3.3 Desvantagem

A principal desvantagem do laravel está na infraestrutura, não é possível hospedar um sistema desenvolvido em laravel em hospedagem compartilhada..

4.4 React

Como definido por seus criadores, React é “uma biblioteca JavaScript declarativa, eficiente e flexível para a criação de interfaces de usuário (UI)”. Essa biblioteca surgiu em

2011, no Facebook, e passou a ser utilizada na interface do mural de notícias da rede social [10].

No ano seguinte, passou a integrar também a área de tecnologia do Instagram e de várias outras ferramentas da empresa. Em 2013, o código foi aberto para a comunidade, o que colaborou para sua grande popularização [10].

Muitas pessoas confundem o react com um framework mas ele é uma biblioteca, uma coleção de funcionalidades relacionadas que podem ser chamadas pelo desenvolvedor para resolver problemas específicos como a criação de interfaces de usuário reaproveitáveis, no caso do React [10].

Já a principal característica de um framework é a Inversão de Controle, também conhecida como Hollywood Principle. Ou seja, quem dita o que (e como) algo deve ser feito é o framework, não o desenvolvedor como é ele quem chama o seu código, e não o contrário [10].

O importante disso tudo é que, em geral, bibliotecas são mais flexíveis e menos complexas do que frameworks, no caso do React, sua única e principal função é a criação de interfaces de usuário, que organiza o que será mostrado para o usuário final na tela sem se preocupar em saber sobre o resto [10].

O modo como o React trabalha para criar interfaces de usuário (ou User Interfaces, as UIs) é por meio da quebra de toda a estrutura da aplicação em componentes [10].

Se você tem experiência na criação de sites, deve estar acostumado a ter todo o código HTML de uma página em um único arquivo [10].

O que o React propõe é o contrário: separar todo o código em pequenas partes (em arquivos diferentes), que se comportam como componentes reutilizáveis [10].

4.4.1 Vantagem de react ao javascript puro

O React trabalha com o DOM (Document Object Model) e atualiza os componentes de acordo com seus estados [10].

Em uma aplicação JavaScript tradicional, o programador deve se preocupar em descobrir quais dados mudaram para poder alterar o DOM e os estados dos elementos criados. Isso é muito trabalhoso e pouco eficiente [10].

O que o React propõe é a criação do seu próprio DOM, mais eficiente, no qual os componentes vivem, o que é mais conhecido como Virtual DOM [10].

Assim, toda vez que um componente é renderizado, o React atualiza o Virtual DOM de cada componente já renderizado e busca as mudanças. E como o Virtual DOM é leve, esse processo é muito rápido [10].

O React então compara o Virtual DOM com uma imagem do DOM feita antes da atualização e descobre o que realmente mudou, atualizando somente os componentes que mudaram de estado, isso gera um enorme ganho de performance para a aplicação [10].

5 Desenvolvimento

Esta sessão tem por objetivo apresentar detalhes acerca da arquitetura e definição do escopo do projeto.

5.1 Arquitetura

Esta sessão do documento se destina aos detalhamentos das estratégias adotadas no desenvolvimento do sistema, como arquitetura, levantamento de requisitos, modelagem do banco de dados e detalhamento dos casos de uso.

5.1.1 Arquitetura do sistema

Inicialmente será abordado as estratégias adotadas na arquitetura do sistema e troca de mensagens. Como pode ser observado na figura 3 as aplicações de interface com o usuário enviam mensagens para os endpoints da API e recebem os dados de acordo com o tipo de solicitação e permissões.

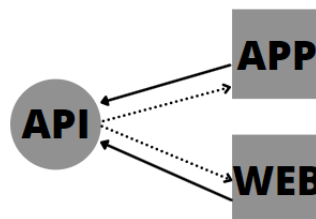


Fig. 3. Arquitetura proposta.

5.1.2 API

A API ou backend da aplicação é responsável por receber as requisições da interface do usuário e avaliar quais podem ser executadas e quais devem ser descartadas. A unidade de controle foi desenvolvida sob uma estrutura de funções que podem ser requisitadas pela interface onde cada função possui uma permissão diferente. Quando estas funções são requisitadas o sistema avalia se o usuário possui a permissão vinculada a função para que a mesma seja executada.

5.1.3 Interfaces do Usuário

As interfaces do usuário são compostas por dois projetos, o MVP do aplicativo foi concebido com o uso da biblioteca react native e está previsto que o MVP da pagina web seja

concebido com com react.js. De forma geral as interfaces com o usuário possuem relações diretas com as funcionalidades disponibilizadas pela API, portanto a responsabilidade de uma interface é validar os dados imputados pelo usuário, enviar e receber dados da API.

5.1.4 Levantamento de requisitos

Este levantamento de requisitos foi elaborado após a feira de robótica levando em consideração as falhas encontradas no processo de oferta de um minicurso.

5.1.4.1 Requisitos funcionais

- O sistema deve permitir que o usuário cadastrado realize login no sistema;
- O sistema deve ser modelado de acordo com o perfil vinculado ao usuário exibindo apenas funcionalidades que o mesmo possui permissão;
- O sistema deve permitir que o usuário ministrante cadastre um novo minicurso;
- O sistema deve permitir que o usuário ministrante edite seus minicursos;
- O sistema deve permitir que o usuário ministrante delete um minicurso;
- O sistema deve permitir que todos usuários vejam os minicursos cadastrados e liberados;
- O sistema deve possuir uma área para visualizar informações detalhadas do minicurso;
- O sistema deve permitir que o usuário ministrante visualize quem já se inscreveu no seu minicurso e quantos já pagaram a taxa de inscrição.
- O sistema deve permitir que o usuário professor libere um minicurso recém cadastrado;
- O sistema deve permitir que o professor informe pagamento de matrícula em um minicurso;
- O sistema deve permitir que o usuário aluno se inscreva em um minicurso;
- O sistema deve permitir que o usuário aluno remova sua inscrição;
- O sistema deve permitir que o usuário aluno colete seu certificado;

5.1.4.2 Requisitos não funcionais

- O sistema deve impedir acesso a usuários não cadastrados;
- O sistema deve manter permissões específicas para cada funcionalidade do sistema;
- O sistema deve impedir que um curso não liberado seja oferecido;

5.2 Diagrama de Entidade Relacional

Nesta sessão é apresentado o diagrama de entidade relacional do projeto.

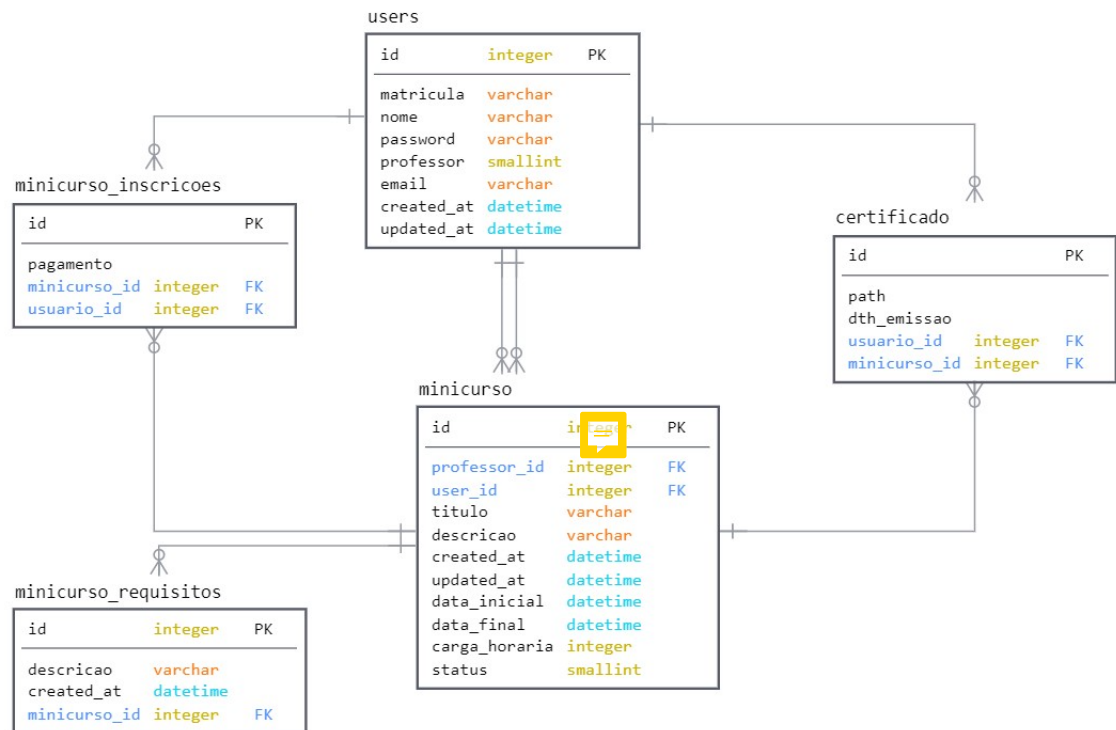


Fig. 4. Diagrama de entidade relacional do projeto

5.3 Diagrama de casos de uso

Nesta sessão é apresentado o diagrama de casos de uso do projeto.

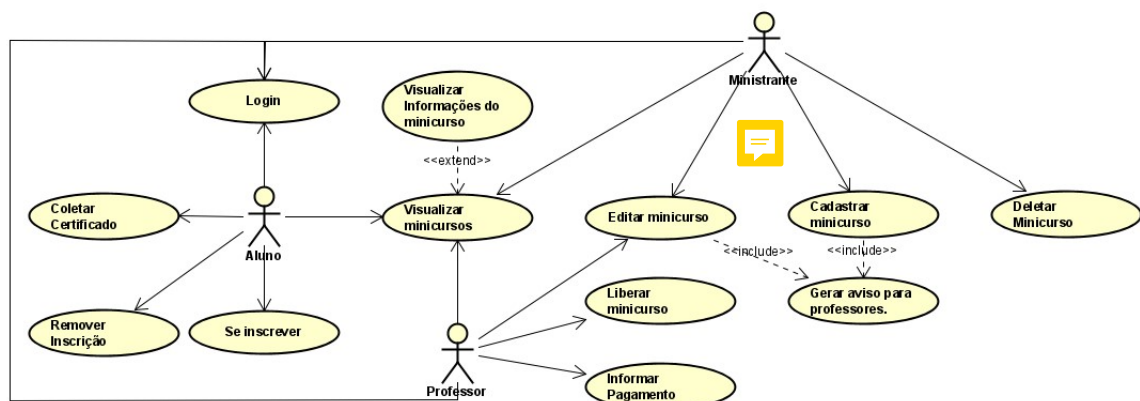


Fig. 5. Diagrama de casos de uso do sistema

5.3.1 Detalhamento das funcionalidades

Nesta sessão é apresentado os detalhamentos das funcionalidades do MVP.

5.3.1.1 UC0001: Logar no Sistema

- **REQUISITO:** O sistema deve permitir que aluno e professores realizem login no sistema utilizando usuário e senha do portal do aluno.
- **OBJETIVO:** Criar uma funcionalidade que controle o acesso no sistema através de login e senha;
- **DESCRIÇÃO:** O sistema deve apresentar as funcionalidades de acordo com seu perfil de usuário;
- **ATOR(ES):** Aluno e Administrador do Sistema;
- **PRE CONDIÇÃO:** Interligar sistema ao banco de dados da universidade.
- **PÓS CONDIÇÃO:** Carregar dashboard com as informações do usuário e ações possíveis.
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

O sistema apresenta a interface de login RN001;

O usuário preenche o formulário informado a matrícula e senha;

O sistema valida se as credenciais informadas são validas;

O sistema identifica que as credenciais são validas;

O sistema gera um token e retorna para o aplicativo autenticar suas próximas ações.

O sistema redireciona o usuário para a dashboard e disponibiliza o menu, dashboard, ministrar e minicursos;

O caso de uso é encerrado.

FLUXO ALTERNATIVO:

O sistema identifica que as credenciais informadas são invalidas;

O sistema aborta a operação e exibe a mensagem “Usuário e Senha informados não constam em nossa base de dados tente novamente”.

O caso de uso é encerrado;

5.3.1.1.1 Regra de negócio

Campos Obrigatórios	
RN001	Matricula Senha

5.3.1.1.2 Requisitos de tela



The login screen features a light gray background. At the top left is a circular institutional logo. To its right is the text 'unipac.br' in a large, bold, orange font, with 'Lafayette' in a smaller, gray font underneath. Below the header are two white input fields with rounded corners. The first field is labeled 'Matrícula' and the second is labeled 'Senha'. At the bottom center is a dark blue rectangular button with the word 'ENTRAR' in white capital letters.

Fig. 6. Janela de login do sistema



This screen is identical to the one in Figure 6, but it includes an error message. Below the 'unipac.br Lafayette' logo, the text 'Usuário não autorizado!' is displayed in a red font. The 'Matrícula', 'Senha', and 'ENTRAR' button remain in their original positions and colors.

Fig. 7. Janela de falha no login do sistema

5.3.1.1.3 Diagramas

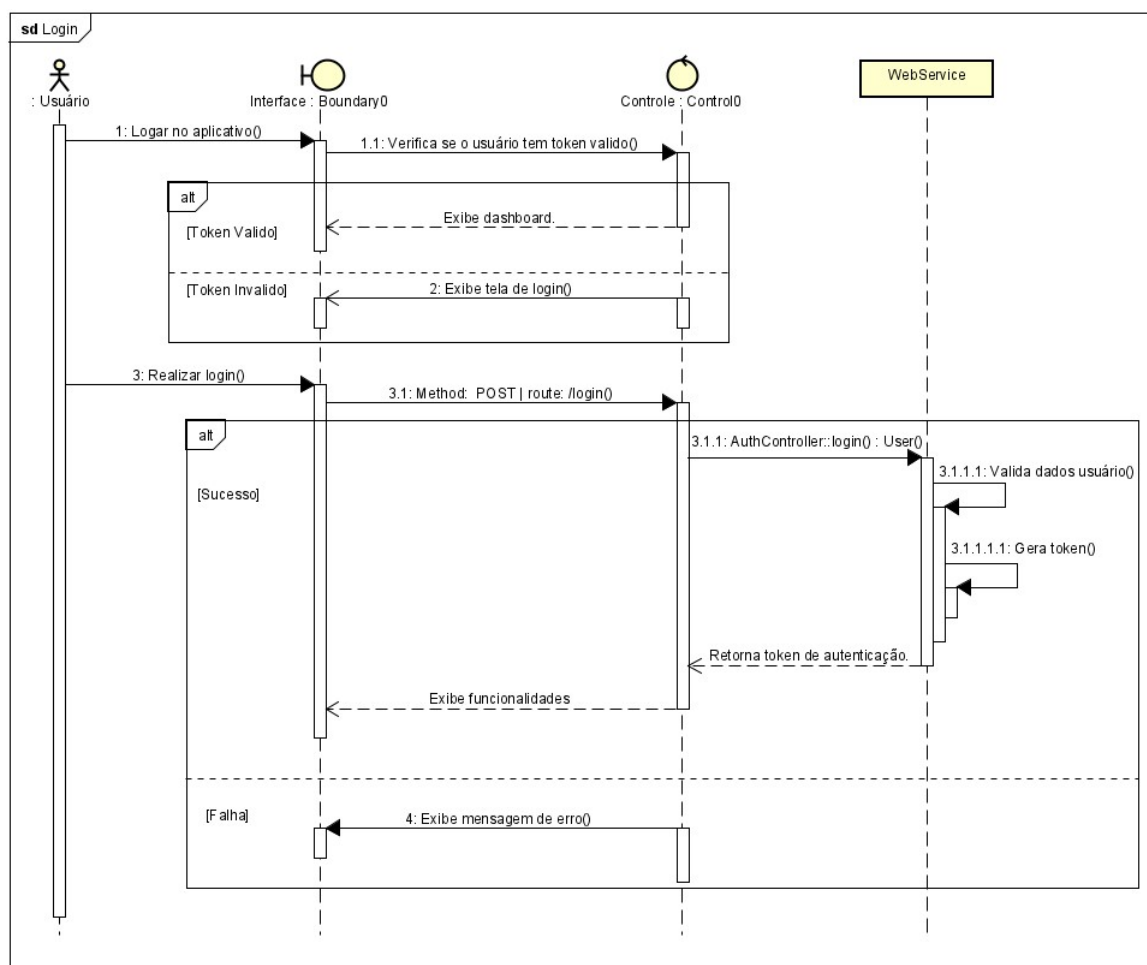


Fig. 8. Diagrama de sequencia do requisito de login

5.3.1.2 UC0002: Cadastrar minicursos

- **REQUISITO:** O sistema deve permitir que alunos cadastrem um minicurso.
- **OBJETIVO:** Criar interface visual e lógica para armazenar o minicurso no banco de dados.
- **DESCRIÇÃO:** O sistema deve permitir que o aluno cadastre um minicurso e que após esse cadastro esteja concluído ele seja disponibilizado para um professor realizar a liberação do mesmo.
- **ATOR(ES):** Aluno;
- **PRE CONDIÇÃO:** O aluno deve estar logado para realizar esta ação.
- **PÓS CONDIÇÃO:** Exibir a lista de solicitações do aluno.
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

O aluno utiliza o menu ministrar para acessar a interface e clica no botão criar minicurso, para iniciar o cadastro da RN002;

O usuário preenche o formulário informado título, datas, carga horaria, descrição e a lista de pré-requisitos;

O sistema valida se as credenciais informadas são validas

O sistema valida se todos os campos foram preenchidos

Verifica se os dados são validos, com base no tamanho e valor informado;

Salva o minicurso no banco de dados e envia notificação para os professores;

d. O sistema redireciona o usuário para a interface do menu ministrar;

O caso de uso é encerrado.

FLUXO ALTERNATIVO:

O sistema identifica que os dados informados são inválidos;

O sistema aborta a operação e apresenta o erro para que o usuário possa corrigir o problema.

O caso de uso é encerrado;

5.3.1.2.1 Regra de negócio

Campos Obrigatórios	
RN002	Título
	Descrição
	Pré-Requisitos[]
	Data inicial
	Data Final
	Carga horária

5.3.1.2.2 Requisitos de tela

Nova Solicitação

Atenção, os minicursos só serão liberados após a aprovação de um professor! Preencha todos os campos corretamente.

CRIAR MINICURSO

Minhas Solicitações

Titulo	Status	Data
Linux	Liberado	22/08/2021
Mysql	Liberado	22/08/2021
LGPD	Liberado	22/08/2021

Fig. 9. Lista de minicursos e botão para criar minicurso

5.3.1.2.3 Diagramas

1 - INFORMAÇÕES DO MINICURSO

Ex: Curso de introdução a programação

Início: 06/09/2021 Término: 06/09/2021

Inserir Data Inserir Data

Carga horária do curso. Ex: 8

Breve descrição, citar os tópicos que o curso irá abordar.
Por exemplo:

Curso de introdução a programação com C#

- Linguagem c#
- Estruturas condicionais
- Estruturas de repetição
- Estrutura de seleção
- Orientação a objetos

DASHBOARD MINISTRAR CURSOS

Fig. 10. Cadastro de minicurso etapa 1

2 - PRÉ-REQUISITOS DO MINICURSO

Titulo	Opções
-	Delete

Breve descrição do requisito.

Adicionar Requisito

VOLTAR CONCLUIR

DASHBOARD MINISTRAR CURSOS

Fig. 11. Cadastro de minicurso etapa 2

1 - INFORMAÇÕES DO MINICURSO

Preencha o título antes de avançar!

Ex: Curso de introdução a programação

Início: 06/09/2021 Término: 06/09/2021

Inserir Data Inserir Data

Carga horaria do curso. Ex: 8

Breve descrição, citar os topicos que o curso ira abordar.
Por exemplo:

Curso de introdução a programação com C#

- Linguagem c#
- Estruturas condicionais
- Estruturas de repetição
- Estrutura de seleção
- Orientação a objetos

VOLTAR AVANÇAR

DASHBOARD MINISTRAR CURSOS

Fig. 12. Cadastro de minicurso erro de validação

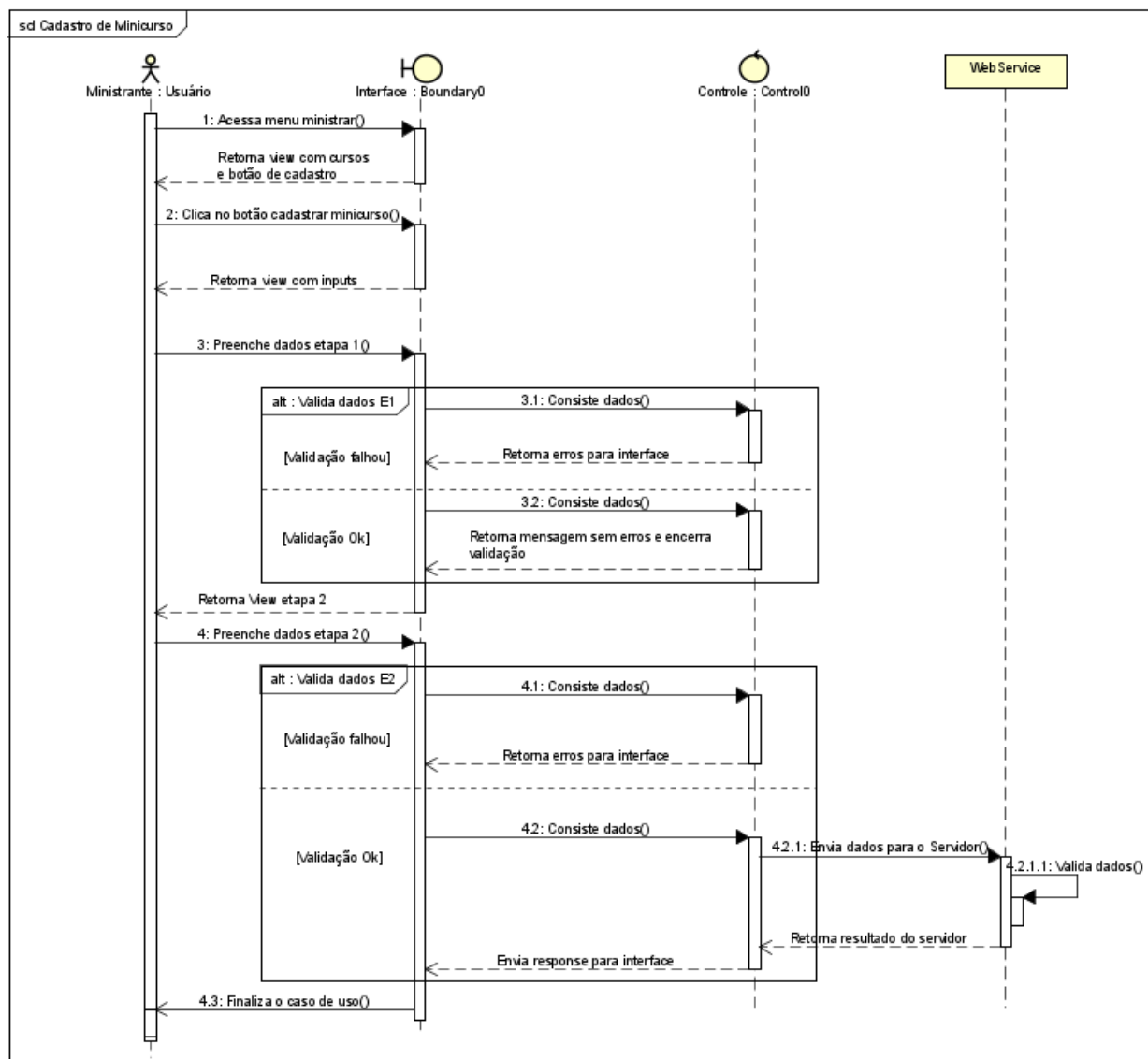


Fig. 13. Diagrama de sequencia do requisito de cadastro de minicurso

5.3.1.3 UC0003: Editar minicursos

- **REQUISITO:** O sistema deve permitir que alunos editem seus minicursos.
- **OBJETIVO:** Modificar interface de cadastro para permitir que o minicurso seja editado.
- **DESCRIÇÃO:** O sistema deve permitir que o aluno edite um minicurso e que após a edição ele seja disponibilizado para um professor avaliar e realizar a liberação do mesmo.
- **ATOR(ES):** Aluno;
- **PRE CONDIÇÃO:** O aluno deve estar logado para realizar esta ação.
- **PÓS CONDIÇÃO:** Exibir a lista de solicitações do aluno.
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

O aluno utiliza o menu ministrar para acessar a lista de solicitações.

O aluno clica em um minicurso da lista de solicitações

Carrega os dados do minicurso e preenche os inputs assim o usuário pode editar o que desejar, conforme: **RN003**;

O usuário edita a gosto informando título, datas, carga horaria, descrição e a lista de pré-requisitos

O sistema valida se as credenciais informadas são validas;

O sistema valida se todos os campos foram preenchidos.

Verifica se os dados são validos, com base no tamanho e valor informado.

Salva o minicurso no banco de dados e envia notificação para os professores.

O sistema redireciona o usuário para a interface do menu ministrar;

O caso de uso é encerrado

FLUXO ALTERNATIVO:

O sistema identifica que os dados informados são inválidos;

O sistema aborta a operação e apresenta o erro para que o usuário possa corrigir o problema.

O caso de uso é encerrado;

5.3.1.3.1 Regra de negócio

Campos Obrigatórios	
RN003	Título
	Descrição
	Pré-Requisitos[]
	Data inicial
	Data Final
	Carga horária

5.3.1.3.2 Requisitos de tela

Nova Solicitação

Atenção, os minicursos só serão liberados após a aprovação de um professor! Preencha todos os campos corretamente.

CRIAR MINICURSO

Minhas Solicitações

Titulo	Status	Data
Linux	Liberado	22/08/2021
Mysql	Liberado	22/08/2021
LGPD	Liberado	22/08/2021

Three arrows point to the 'Data' column of the table, indicating that clicking on the date allows editing the item.

Fig. 14. Lista de minicurso editar clique no item da lista.

5.3.1.3.3 Diagramas

1 - INFORMAÇÕES DO MINICURSO

Ex: Curso de introdução a programação

Início: 06/09/2021 Término: 06/09/2021

Inserir Data Inserir Data

Carga horaria do curso. Ex: 8

Breve descrição, citar os topicos que o curso ira abordar.
Por exemplo:

Curso de introdução a programação com C#

- Linguagem c#
- Estruturas condicionais
- Estruturas de repetição
- Estretutura de seleção
- Orientação a objetos

DASHBOARD MINISTRAR CURSOS

Fig. 15. Cadastro de minicurso etapa 1

2 - PRÉ-REQUISITOS DO MINICURSO

Titulo	Opções
-	Delete

Breve descrição do requisito.

Adicionar Requisito

VOLTAR CONCLUIR

DASHBOARD MINISTRAR CURSOS

Fig. 16. Cadastro de minicurso etapa 2

1 - INFORMAÇÕES DO MINICURSO

Preencha o título antes de avançar!

Ex: Curso de introdução a programação

Início: 06/09/2021 Término: 06/09/2021

Inserir Data Inserir Data

Carga horaria do curso. Ex: 8

Breve descrição, citar os topicos que o curso ira abordar.
Por exemplo:

Curso de introdução a programação com C#

- Linguagem c#
- Estruturas condicionais
- Estruturas de repetição
- Estrutura de seleção
- Orientação a objetos

VOLTAR AVANÇAR

DASHBOARD MINISTRAR CURSOS

Fig. 17. Cadastro de minicurso erro de validação

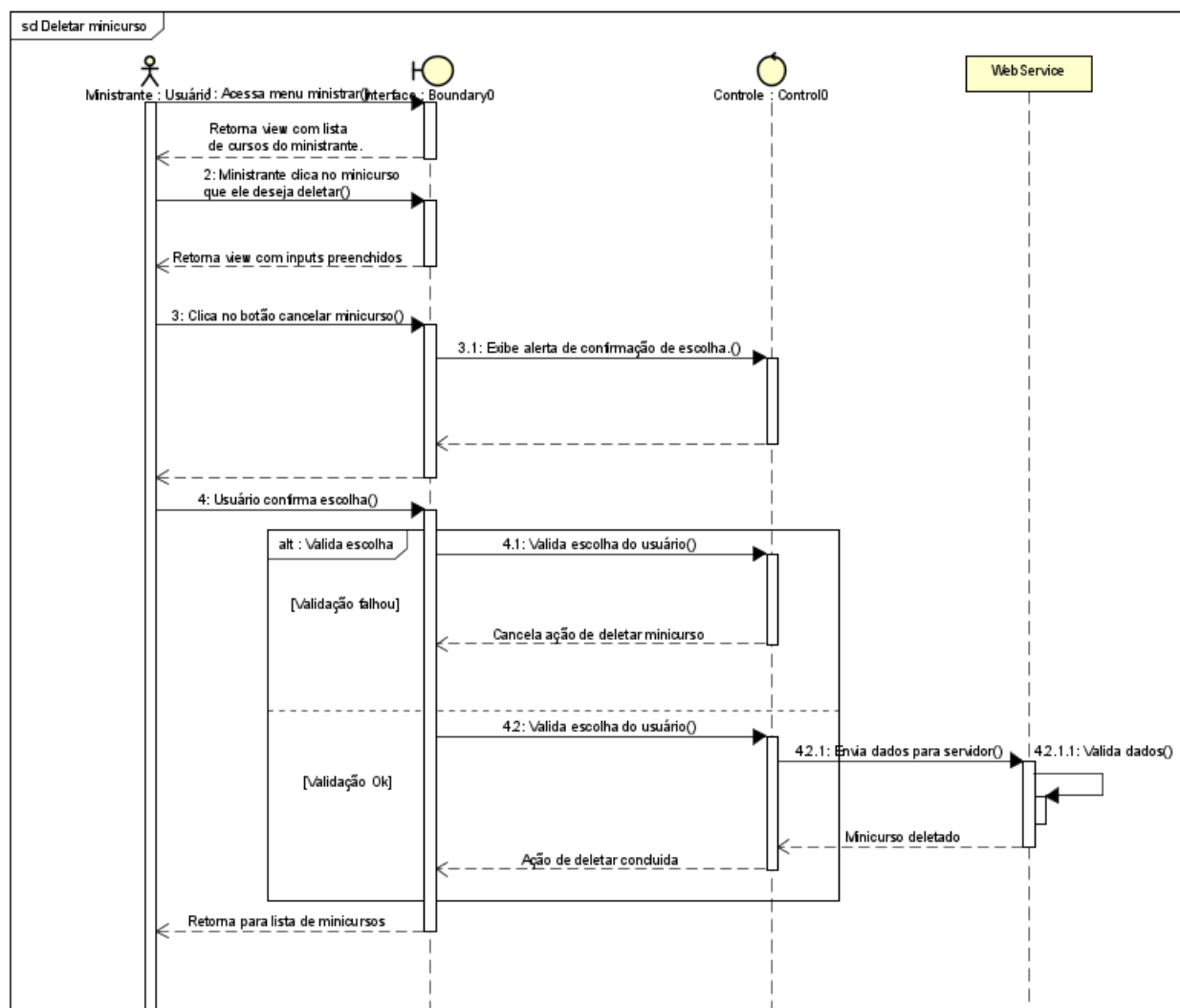


Fig. 18. Diagrama de sequencia do requisito de editar minicurso

5.3.1.4 UC0004: Deletar minicursos

- **REQUISITO:** O sistema deve permitir que o aluno e os professores deletem minicursos.
- **OBJETIVO:** Permitir que professores e alunos ministrantes deletem minicursos.
- **DESCRIÇÃO:** O usuário ministrante pode deletar apenas os seus minicursos, o professor pode deletar todos.
- **ATOR(ES):** Aluno e Professor;
- **PRE CONDIÇÃO:** Login, possibilitar visualização de todos os minicursos, cadastro de minicursos.
- **PÓS CONDIÇÃO:** Recarregar lista de minicursos cadastrados.
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

O sistema apresenta a interface solicitações de minicursos, que pode ser acessada pelo menu ministrar.

O usuário seleciona o minicurso que deseja deletar

Usuário clica no botão deletar minicurso RN004.

O sistema redireciona o usuário para a lista de minicursos, com a lista de solicitações atualizada.

O caso de uso é encerrado;

FLUXO ALTERNATIVO:

O sistema identifica que as credenciais informadas são inválidas.

O sistema aborta a operação e exibe a mensagem “Usuário e Senha informados não constam em nossa base de dados tente novamente”.

O sistema identifica que o minicurso selecionado possui informações inválidas.

Redireciona para o menu ministrar com mensagem de erro “Dados do minicurso inválidos “.

O caso de uso é encerrado;

5.3.1.4.1 Regra de negócio

Campos Obrigatórios	
RN004	ID Minicurso Token de Login

5.3.1.4.2 Requisitos de tela

Nova Solicitação

Atenção, os minicursos só serão liberados após a aprovação de um professor! Preencha todos os campos corretamente.

CRIAR MINICURSO

Minhas Solicitações

Titulo	Status	Data
Linux	Liberado	22/08/2021
Mysql	Liberado	22/08/2021
LGPD	Liberado	22/08/2021

Fig. 19. Lista de minicurso editar clique no item da lista.

5.3.1.4.3 Diagramas

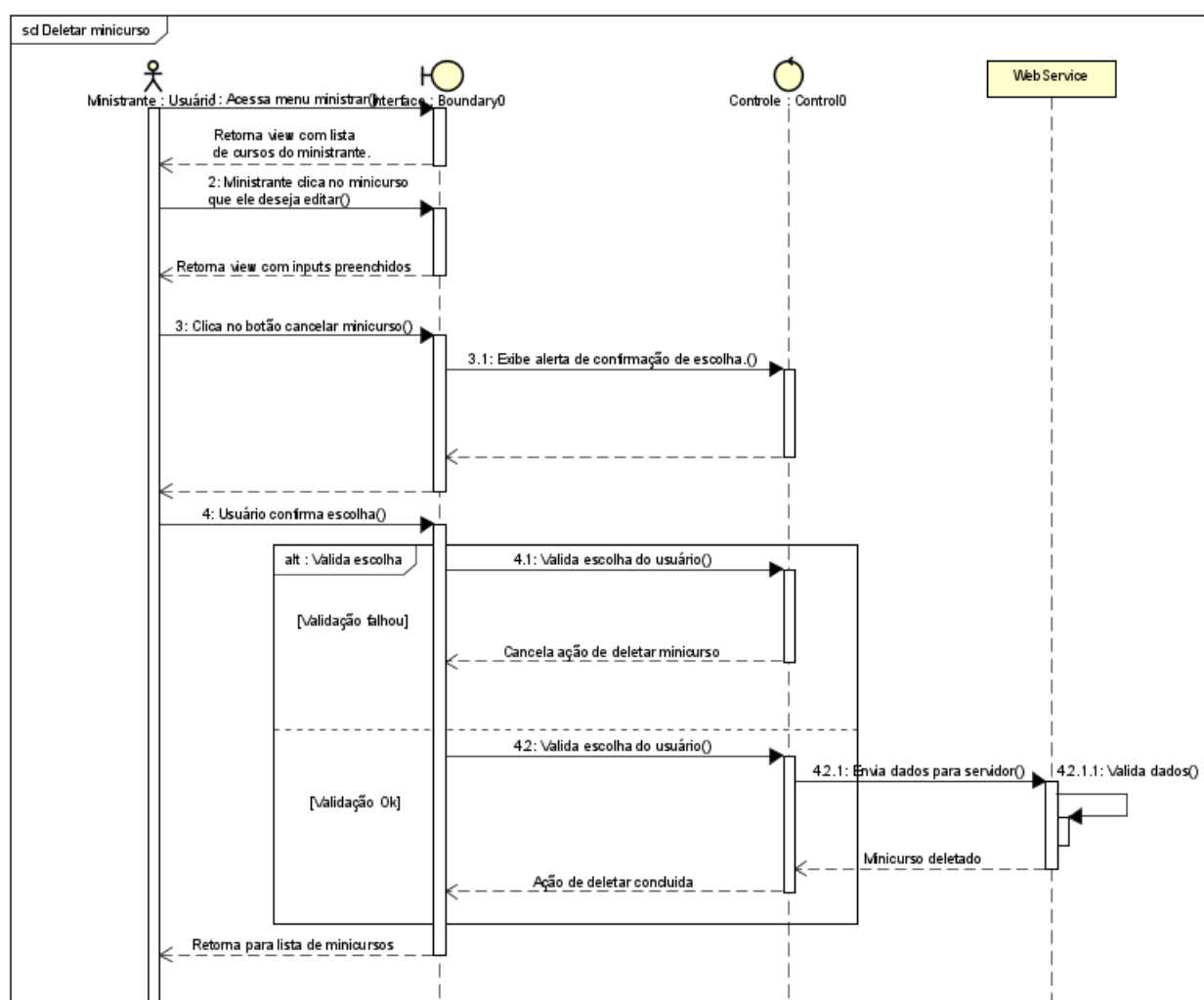


Fig. 20. Diagrama de sequencia do requisito de deletar minicurso

5.3.1.5 UC0005: Visualizar minicursos

- **REQUISITO:** O sistema deve possuir uma área para alunos visualizarem os minicursos já liberados pelos professores.
- **OBJETIVO:** Permitir que os alunos que desejam se inscrever em um minicurso tenha acesso a um repositório com as informações gerais do curso.
- **DESCRIÇÃO:** O aluno deve poder se inscrever e remover sua inscrição, desde que ele ainda não tenha efetuado o pagamento da matrícula.
- **ATOR(ES):** Aluno;
- **PRE CONDIÇÃO:** Login.
- **PÓS CONDIÇÃO:** Selecionar um minicurso.

- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

Aluno acessa o menu minicursos.

Sistema exibe todos minicursos já liberados por um professor para o aluno.

Informações gerais do curso

Título

Quantidade de inscritos

Descrição resumida

Data início/fim

Carga horária

Aluno seleciona minicurso e todas informações do curso são carregadas.

Opções

Se inscreve UC0006

Voltar para lista

O caso de uso é encerrado;

FLUXO ALTERNATIVO:

O sistema identifica que as credenciais informadas são inválidas.

O caso de uso é encerrado;

5.3.1.5.1 Requisitos de tela



Fig. 21. Lista de minicurso liberados para se inscrever.

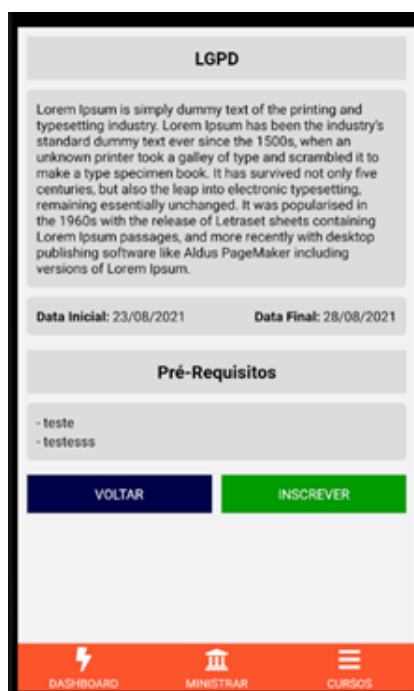
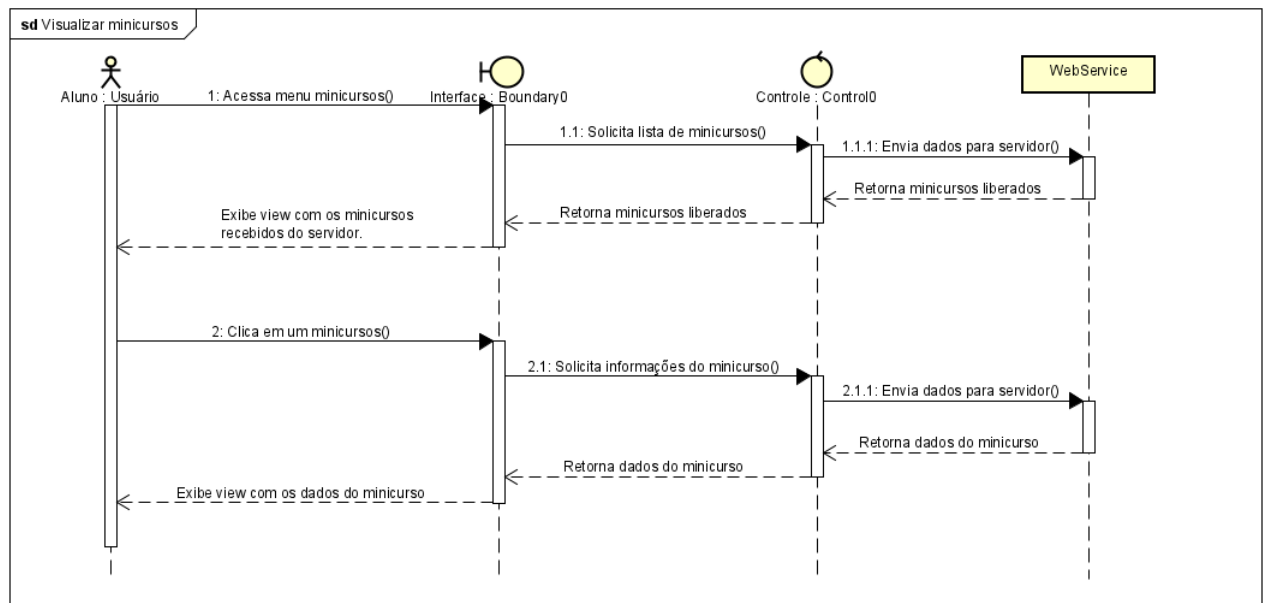


Fig. 22. Interface de visualização e inscrição de minicurso.

5.3.1.5.2 Diagramas

**Fig. 23.** Diagrama de sequencia do requisito de visualizar minicurso

5.3.1.6 UC0006: Se inscrever

- **REQUISITO:** Registrar inscrição do aluno e emitir alerta para aluno ministrante e professor.
- **OBJETIVO:** Registrar intenção do aluno e emitir alerta para os responsáveis pelo minicurso.
- **DESCRIÇÃO:** O sistema deve armazenar a inscrição no banco de dados e emitir alerta para o professor e aluno ministrante. O sistema também deve informar ao aluno inscrito que ele deve procurar o coordenador do curso para pagar a taxa de inscrição.
- **ATOR(ES):** Aluno, Professor, Aluno ministrante;
- **PRE CONDIÇÃO:** Login.
- **PÓS CONDIÇÃO:** Exibir alerta ou redirecionar para dashboard.
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

Partindo do fluxo UC0005 o aluno visualizou as informações e utilizou a função “Se inscrever”.

Sistema envia alertas para o professor e aluno.

O caso de uso é encerrado;

FLUXO ALTERNATIVO:

O sistema identifica que as credenciais informadas são inválidas.

O caso de uso é encerrado;

5.3.1.6.1 Diagramas

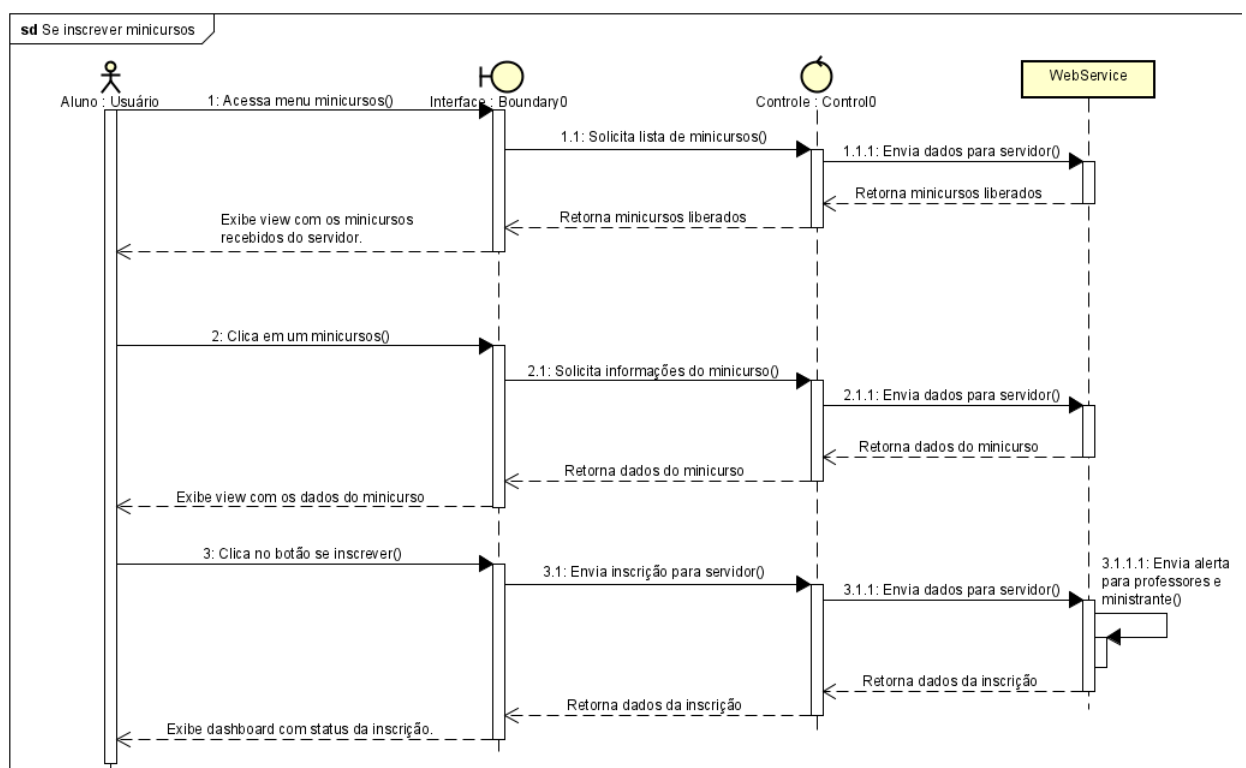


Fig. 24. Diagrama de sequencia do requisito de se inscrever em um minicurso

5.3.1.7 UC0007: Informar Pagamento

- **REQUISITO:** O sistema deve possuir uma área para o professor informar o pagamento de uma inscrição.
- **OBJETIVO:** Informar o pagamento da inscrição de um aluno.
- **DESCRIÇÃO:** O sistema deve exibir para o professor todos os alunos com pagamento pendente, para que ele filtre pelo nome na hora de informar o pagamento.
- **ATOR(ES):** Professor;
- **PRE CONDIÇÃO:** Login, Permissão de Administrador..
- **PÔS CONDIÇÃO:** Exibir alerta de sucesso e voltar para janela para informar novo pagamento..
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

Professor acessa a plataforma web visualiza a funcionalidade informar pagamento.

O sistema lista todos alunos com pagamento pendente.

Professor pressiona botão informar pagamento.

Sistema exibe alerta de sim ou não.

Se sim, altera registro no banco de dados e limpa os filtros.

Se não, volta para lista.

O caso de uso é encerrado;

FLUXO ALTERNATIVO:

O sistema identifica que as credenciais informadas são inválidas.

O caso de uso é encerrado;

5.3.1.7.1 Requisitos de tela

PENDENTE JANELA

5.3.1.7.2 Diagramas

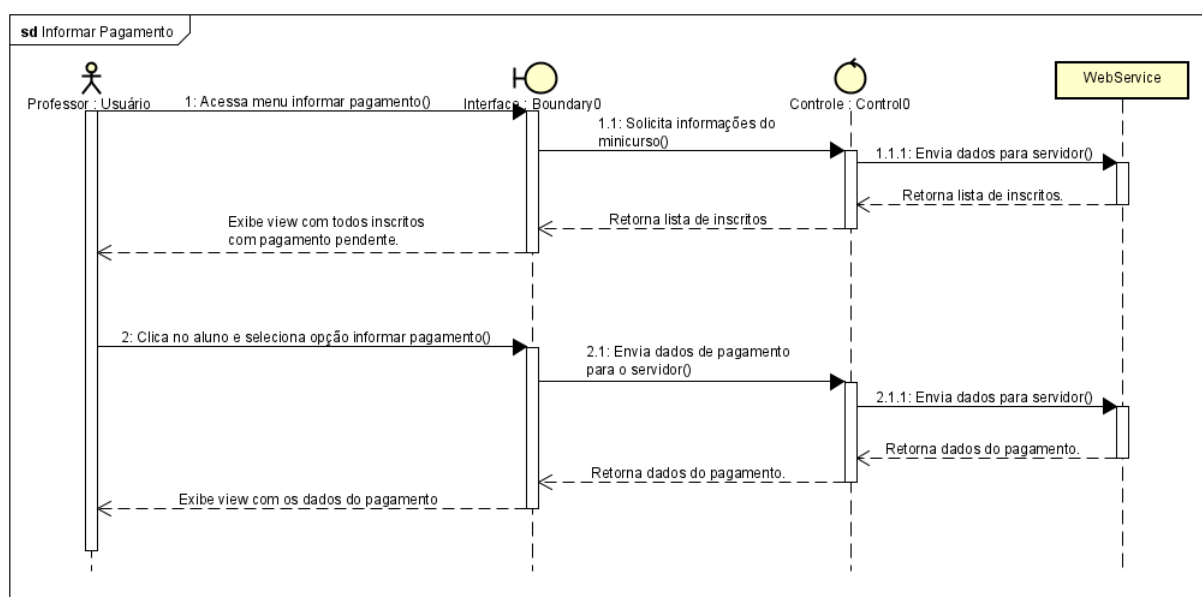


Fig. 25. Diagrama de sequencia do requisito de informar pagamento

5.3.1.8 UC0008: Remover Inscrição

- **REQUISITO:** Permitir que o aluno remova a inscrição do minicurso.
- **OBJETIVO:** Deixar livre para que o aluno remova a inscrição quando ele quiser.
- **DESCRIÇÃO:** O sistema deve permitir que o aluno mude de ideia e remova a inscrição em um minicurso, desde que ele ainda não tenha realizado o pagamento.
- **ATOR(ES):** Aluno;
- **PRE CONDIÇÃO:** Login.
- **PÔS CONDIÇÃO:** Exibir alerta e redirecionar para dashboard.
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

Aluno acessa minicurso através da dashboard.

Aluno clica no botão remover inscrição.

Sistema valida se a remoção pode ser realizada.

Se sim, redireciona o aluno para lista de minicursos.

Se não, informa que ele não pode mais remover a inscrição.

O caso de uso é encerrado;

FLUXO ALTERNATIVO:

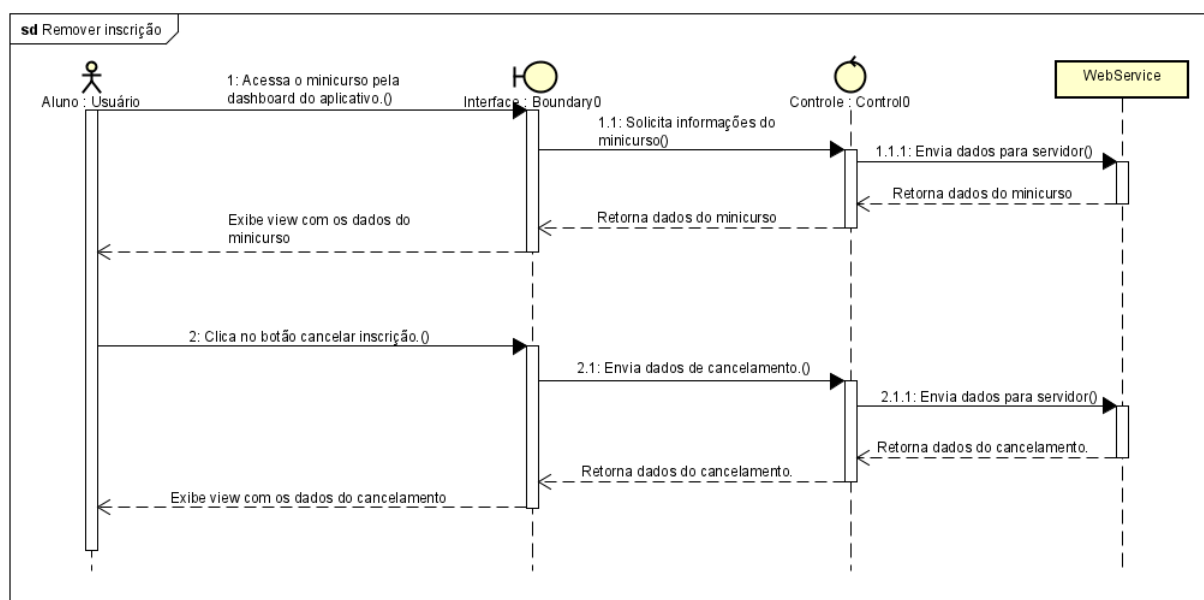
O sistema identifica que as credenciais informadas são inválidas.

O caso de uso é encerrado;

5.3.1.8.1 Requisitos de tela

PENDENTE JANELA

5.3.1.8.2 Diagramas

**Fig. 26.** Diagrama de sequencia do requisito de cancelar inscrição

5.3.1.9 UC0009: Emitir Certificado

- **REQUISITO:** O sistema deve possuir uma funcionalidade para o aluno emitir o certificado.
- **OBJETIVO:** Permitir que o aluno faça download do certificado.
- **DESCRIÇÃO:** O sistema deve gerar o certificado e enviar por email para os alunos.
- **ATOR(ES):** Aluno;
- **PRE CONDIÇÃO:** Login.
- **PÔS CONDIÇÃO:** Exibir alerta e redirecionar para dashboard.
- **FLUXO DE EVENTOS:**

FLUXO PRINCIPAL:

Aluno seleciona minicurso concluído através da dashboard.

Sistema carrega informações do minicurso.

Aluno clica no botão emitir certificado.

Sistema envia certificado por email.

O caso de uso é encerrado;

FLUXO ALTERNATIVO:

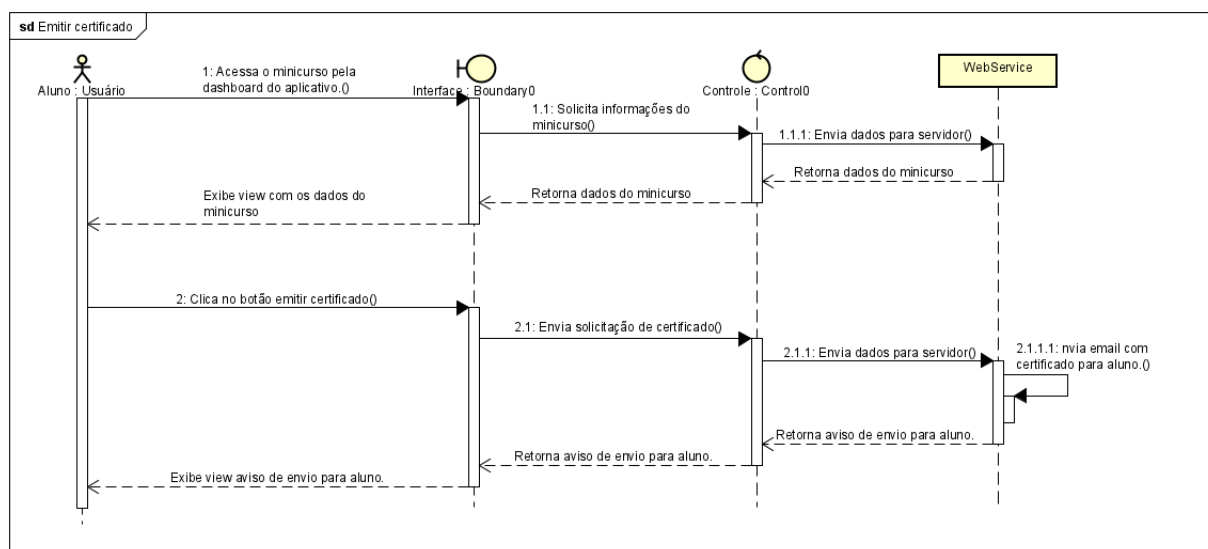
O sistema identifica que as credenciais informadas são inválidas.

O caso de uso é encerrado;

5.3.1.9.1 Requisitos de tela

PENDENTE TELA

5.3.1.9.2 Diagramas

**Fig. 27.** Diagrama de sequencia do requisito de emitir certificado

6 Considerações Finais

6.1 Conclusão

PENDENTE Podemos dizer que este é um sistema complexo, para realizar a desafiadora maratona que é compreender como cada parte funciona e gerenciar com maestria esses componentes é um grande desafio [9], [10]. Para isso os computadores modernos são equipados com um SO esse dispositivo de software tem a função de fornecer uma plataforma simples e limpa para o usuário de forma a ajuda-lo nas entradas e saídas de dados. Em uma visão simplista podemos ver na figura 5 onde o SO se encontra em relação entre hardware e o usuário [10] *apud* [9].

Referências

- [1] PROJETO SIGEV – sistema gerenciador de eventos, Disponível em: <http://ocs.ifsp.edu.br/index.php/conict/xiconict/paper/viewPaper/7029> [Acessado em 25 Outubro 2021].
- [2] EventPRO –Sistema de Gerenciamento de Eventos, Disponível em: <http://anais.unievangelica.edu.br/index.php/praticasdocentes/article/view/5508/3150> [Acessado em 25 Outubro 2021].
- [3] Metodologia Científica no Brasil: ensino e interdisciplinaridade, Disponível em: <https://doi.org/10.1590/2175-623654484> [Acessado em 25 Outubro 2021].
- [4] METODOLOGIA CIENTÍFICA TEORIA E APLICAÇÃO NA EDUCAÇÃO A DISTÂNCIA, Disponível em: <https://portais.univasf.edu.br/dacc/noticias/livro-univasf/metodologia-cientifica-teoria-e-aplicacao-na-educacao-a-distancia.pdf> [Acessado em 25 Outubro 2021].
- [5] ESCOLHAS METODOLÓGICAS EM ESTUDOS ORGANIZACIONAIS, Disponível em: <https://www.scielo.br/j/osoc/a/xyjfxnsm6jNH4mGWRyZhPHL/?format=pdf&lang=pt> [Acessado em 25 Outubro 2021].
- [6] Higor Medeiros, A., 2019. Introdução ao padrão MVC - DevMedia. [online] DevMedia. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-laravel/> [Acessado em 25 Abril 2021].
- [7] O que é o PHP. PHP, Disponível em: https://www.php.net/manual/pt_BR/intro-what-is.php [Acessado em 25 Abril 2021].
- [8] O que o PHP pode fazer. PHP, Disponível em: https://www.php.net/manual/pt_BR/intro-whatcando.php [Acessado em 25 Abril 2021].
- [9] Paula de Andrade, A., 2019. O que é Laravel?. [online] TreinaWeb. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-laravel/> [Acessado em 25 Abril 2021].
- [10] React: o que é e como funciona essa ferramenta? TABLELESS, Disponível em: <https://tableless.com.br/react-o-que-e-e-como-funciona-essa-ferramenta/> [Acessado em 25 Abril 2021].

-
- [11] Pesquisa Bibliográfica - Metodologia Científica, Disponível em: <https://www.metodologiacientifica.org/tipos-de-pesquisa/pesquisa-bibliografica/> [Acessado em 25 Abril 2021].
- [12] Projeto de Pesquisa - Metodologia Científica, Disponível em: https://blog.mettzer.com/projeto-de-pesquisa/#0_que_e_projeto_de_pesquisa [Acessado em 25 Abril 2021].
- [13] GITHUB do projeto, Disponível em: <https://bitbucket.org/evtnlife/minicursos/> [Acessado em 25 Outubro 2021].