

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
SISTEMAS DE INFORMAÇÃO

PEDRO HENRIQUE CAMPOS DA SILVA

**Desenvolvimento do Software LIFEDU -
Adaptação da Metodologia SCRUM para
Desenvolvimento**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para a obtenção do
grau de Bacharel em Sistemas de Informação

Prof. Luis Henrique Silva Rodrigues
Orientador

Diamantina, Julho de 2019

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

, Pedro Henrique Campos da Silva

Desenvolvimento do Software LIFEDU - Adaptação da Metodologia SCRUM para Desenvolvimento / Pedro Henrique Campos da Silva . – Diamantina: , 2019.

47 f.: il.

Trabalho de Conclusão de Curso – Universidade Federal dos Vales do Jequitinhonha e Mucuri. Sistemas de Informação, Diamantina, BR–MG, 2019. Orientador: Luis Henrique Silva Rodrigues.

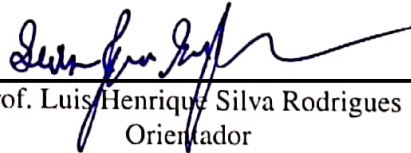
1. Sistema. 2. Software. 3. Desenvolvimento. 4. Engenharia de software. 5. Ferramentas de Software. I. Luis Henrique Silva Rodrigues, . II. Título.

Pedro Henrique Campos da Silva

Desenvolvimento do Software LIFEDU:

Adaptação da Metodologia SCRUM para Desenvolvimento

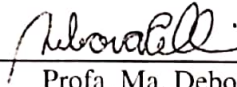
Monografia de projeto final de graduação , defendida por Pedro henrique Campos da Silva e aprovada em 18 de julho de 2019, em Diamantina, Minas Gerais.



Prof. Luis Henrique Silva Rodrigues
Orientador



Prof. Dr. Eduardo Pelli
Integrante da Banca Examinadora



Profa. Ma. Debora Pelli
Integrante da Banca Examinadora

Diamantina

2019

Todas as riquezas do mundo não valem um bom amigo.
Voltaire

AGRADECIMENTOS

Agradeço primordialmente à minha família por todo o apoio ao decorrer de todo o trajeto do Bacharelado, principalmente à minha mãe por todos os conselhos, dedicação e amor. À minha querida irmã Ana Paula Campos por todo apoio prestado e suporte nos momentos cruciais que me levaram à concretizar meu objetivo. Ao meu querido e ilustríssimo primo Cesar Campos por todo o apoio e paciência. Aos meus amigos que ao longo dos anos que passamos juntos se demonstraram presentes nos momentos de alegria e tristeza que compartilhamos. Aos Professores Luis Henrique Silva Rodrigues, Eduardo Pelli e Alessandro Vivas Andrade por todos os ensinamentos e conselhos que foram lecionados durante o período da minha graduação. Aos meu irmãos do clube do XVI por todo o apoio e amizade durante o período de conclusão do curso.

RESUMO

Com a expansão das universidades por todo o território nacional, o número de estudantes que ingressam nas universidades brasileiras cresce constantemente. Um dos problemas enfrentados é o desnivelamento entre os alunos, visto que uma quantidade significativa destes adentram a universidade com conhecimentos deficientes, assim afetando o seu desempenho no decorrer do seu respectivo curso. Este projeto tem como objetivo a construção de um software para que os alunos cadastrados possam assistir conteúdos construídos pela LIFE sobre as matérias ministradas em aula, de maneira a contribuir com o aprendizado e buscar uma melhora do desempenho do aluno perante os cursos de exatas, visando reduzir a taxa de retenção e evasão dos alunos em seus respectivos cursos.

Palavras-chave: Sistema. Software. Desenvolvimento. Engenharia de software. Ferramentas de Software.

Keywords

ABSTRACT

With the expansion of universities across the country, the number of students entering public universities in Brazil has increased, and one of the problems faced is a lack of students' level, since a significant number of them enter the university with deficient knowledge, affecting their performance in the course of their respective course. This project aims to significantly contribute to the student's learning and performance before the courses of exact, in order to reduce the retention rate of students in the course.

Palavras-chave: System. Software. Development. Software Engineering. Software Tools.

LISTA DE SIGLAS

LIFE	Laboratório Interdisciplinar de Formação de Educadores.
SI	Sistemas de Informação.
IDE	Integrated Development Environment (Ambiente Integral de Desenvolvimento).
HTML	HyperText Markup Language.
browser	navegador de rede ou navegador da internet.
WWW	World Wide Web.
W3C	World Wide Web Consortium.
CSS	Cascading Style Sheets.
JS	Java Script.
libs	Library.
frontend	interface frontal.
RF	Requisito Funcional.
RNF	Requisito Não Funcional.
Login	Acesso a uma conta de computador, celular ou outro serviço fornecido por um sistema informático.
Logoff	Refere-se a terminar o uso de um sistema computacional.
BD	Banco de dados.
VSCode	Visual Studio Code.

LISTA DE ABREVIATURAS

UFVJM Universidade Federal Dos Vales Jequitinhoha e Mucuri.

GAAL Geometria Analítica e Algebra Linear.

UML Unified Modeling Language.

ECMA European Computer Manufacturers Association.

Twitter Rede Social.

IEEE Institute of Electrical and Electronics Engineers.

CRUD Create, Read, Update and Delete.

NPM Node Package Manager.

LISTA DE FIGURAS

Figura 2.1:	Imagem ilustrativa Kanban	16
Figura 2.2:	Camadas Engenharia de Software	16
Figura 3.1:	Lista de funções	21
Figura 3.2:	Diagrama de caso de uso	25
Figura 3.3:	Tabela de permissões	26
Figura 3.4:	<i>Product Backlog</i>	27
Figura 3.5:	<i>Sprint Backlog 1</i>	27
Figura 3.6:	<i>Sprint Backlog 2</i>	28
Figura 3.7:	<i>Sprint Backlog 3</i>	30
Figura 3.8:	<i>Sprint Backlog 4</i>	31
Figura 3.9:	<i>Sprint Backlog 5</i>	32

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Contextualização	11
1.2	Problema	12
1.3	Objetivos	12
1.3.1	Objetivos Gerais	12
1.3.2	Objetivos Específicos	12
2	REFERENCIAL TEÓRICO	13
2.1	HTML 5	13
2.2	CSS 3	13
2.3	Java Script	14
2.4	Node.Js	14
2.5	Vue.Js	14
2.6	Bootstrap	14
2.7	MySQL e Workbench	15
2.8	Visual Studio Code	15
2.9	Kanban	15
2.10	Engenharia de Software	16
2.10.1	Processos de software	16
2.10.2	Métodos	16
2.10.3	Metodologia	17
3	DESENVOLVIMENTO DO TRABALHO PROPOSTO	20
3.1	Introdução ao Desenvolvimento do LIFEDU	20
3.2	Planejamento	20
3.2.1	Cronograma	20
3.3	Execução	21
3.3.1	Coleta de Dados	21
3.3.2	Desenvolvimento	26
4	ANÁLISE DE RESULTADOS	34
5	CONSIDERAÇÕES FINAIS	35
	REFERÊNCIAS	36
	APÊNDICE A ANEXOS	38

1 INTRODUÇÃO

1.1 Contextualização

Com a constante evolução das tecnologias da informação e comunicação adstrita com as transformações na sociedade, é válido que os educadores sintam a necessidade de criar medidas que consigam atender a esse desenvolvimento (COLL, MONEREO, 2010). Essas medidas podem se adequar aos diferentes níveis de ensino, de acordo com a lei de diretrizes e bases da educação nacional (LDB), de 1996, compreendidos em educação básica e educação superior (MENEZES, 2001).

A educação superior é compreendida, no cenário internacional, como um bem público (Unesco, 2009). As universidades são instrumentos de transformação social e desenvolvimento dos elementos que lá estão inseridos, assim tendo esse papel reconhecido por todos, a expansão da educação superior pública e gratuita é um bem necessário.

Programas como Sistema de Seleção Unificada (SISU), que é realizado através do aproveitamento do Exame Nacional do Ensino Médio (ENEM), possibilitam que os candidatos possam ingressar no 1º e 2º semestres do ano acadêmico. Há ainda o Sistema de Seleção Seriada (SASI), que corresponde a uma seleção seriada realizada em três etapas que avaliam o conhecimento dos estudantes, podendo ser aproveitado também o ENEM como equivalente a 3º etapa. Esses programas têm como objetivo ampliar o número de pessoas a ingressar nas universidades, facilitando assim a imersão destes cidadãos no ambiente acadêmico.

No Brasil, entre os períodos de 2003 à 2011, houve um processo de interiorização e criação de Universidades Federais. Universidades essas responsáveis pela capacitação de pessoas para que se tornem aptas ao mercado de trabalho. A UFVJM (Universidade Federal dos Vales do Jequitinhonha e Mucuri) foi uma das universidades criadas nesse processo de interiorização, se estabelecendo no município de Diamantina, além de possuir campus em Janaúba e Teófilo Otoni, no estado de Minas Gerais.

Presente no campus JK - Juscelino Kubitschek da UFVJM, o curso de Sistemas de Informação tem por objetivo a formação de profissionais capacitados a atuarem em diferentes organizações, realizando desenvolvimento, avaliação, gestão e utilização de modernas tecnologias de informação (Decom, 2019).

O desenvolvimento de *software* é “o ato de elaborar e implementar um sistema computacional, isto é, transformar a necessidade de um utilizador ou de um mercado em um produto e *software*” (Birrell, 1985). Algumas disciplinas lecionam. Corresponde, ainda, em uma realização com a finalidade de implementar um sistema que transforma a necessidade e/ou problema de um cliente em um produto de *software* que possa suprir essa lacuna de maneira usual e robusta. Através de *frameworks* (conjunto de técnicas, ferramentas ou conceitos utilizados para resolver um problema) e linguagens consolidadas no mercado e

na comunidade *open-source* (código aberto).

1.2 Problema

Um dos problemas a ser solucionado pela Universidade Federal Dos Vales Jequitinhonha e Mucuri é o elevado índice de retenção e evasão de discentes em determinadas disciplinas como Cálculo I, II e GAAL (Geometria Analítica e Álgebra Linear). Esse problema acontece porque é alta a quantidade de estudantes que ingressam na universidade com conhecimentos de base deficientes, oriundos de um sistema de ensino público escolar abaixo do ideal, de acordo com a pesquisa realizada pela Pearson International (The Learning Curve, 2012).

O índice de retenção e evasão dos alunos da UFVJM para determinadas disciplinas, como “Funções de uma Variável”, no curso de Ciência e Tecnologia, oscilou entre 40,54 e 91,67 (por cento) entre 2016/01 e 2017/02 (Memorando nº 355, Prograd, 2018, p. 03). A disciplina Cálculo I, possuindo 769 discentes matriculados contou com uma taxa de aprovação de apenas 20,16 (155 alunos), contrapostos a um índice de reprovação de 48,63 (274 alunos), exemplo fático da situação apresentada (Memorando nº 155, Prograd, 2018, p. 03).

Muitas ferramentas de auxílio a estudantes, como por exemplo “MeSalva”, “Passeidireto”, proporcionam conteúdo gratuito de maneira limitada, no entanto, para obter acesso ao restante do conteúdo é preciso investir uma quantia financeira, muitas vezes acima da capacidade econômica dos estudantes.

1.3 Objetivos

1.3.1 Objetivos Gerais

O objetivo do desenvolvimento do sistema LIFEDU, para viabilizar e disponibilizar vídeos produzidos por professores, tornando-os acessíveis de uma maneira rápida, eficiente e gratuita aos alunos. Além disso, proporcionar gerenciamento de materiais e reuniões através do Laboratório Interdisciplinar de Formação de Educadores (LIFE).

1.3.2 Objetivos Específicos

O presente *software* vê como objetivos específicos:

- Agilizar a necessidade de gerenciamento de materiais do supracitado LIFE, visando automatizar o gerenciamento de reuniões, materiais e empréstimos de equipamento de forma eficiente e eficaz.
- Disponibilizar conteúdo enriquecedor de conhecimento para alunos da UFVJM.
- Facilitar o acesso dos alunos a conteúdos específicos.
- Contribuir para a diminuição do número de retenção de alunos em disciplinas específicas.

2 REFERENCIAL TEÓRICO

Na construção deste projeto foram utilizadas as seguintes linguagens de programação: Html 5, CSS3, JavaScript e NodeJs. Os seguintes *frameworks* foram utilizados: Vue.Js, Bootstrap. Em relação à engenharia de *frameworks*, foi concebido no presente trabalho a metodologia *Scrum*, bem como a utilização da ferramenta Kanban. Para a construção do banco de dados a linguagem escolhida foi o MySQL, gerenciado pelo programa Workbench. As linguagens foram ministradas pela IDE Visual Studio Code.

2.1 HTML 5

O HTML ou HyperText Markup Language 5, sua versão mais recente, é uma linguagem de marcação simples que é bastante utilizada na WWW (World Wide Web), sendo umas das tecnologias base na construção de sistemas web, onde qualquer página da web requer um mínimo de Html. Baseada na sua simplicidade, uma vasta gama de browsers o interpretam de maneira fluída e usual, contemplado no manual "HTML 5 TUTORIAL" elaborado pela W3C (2017c).

A linguagem foi criada por Tim Berners-Lee, Robert Cailliau e outros a partir de 1989 (W3C, 2017c). Com o passar dos anos, o Html sofreu atualizações e correções em seu conteúdo, ampliando suas funções e agregando mais valor a sua gama de possibilidades. Na versão atual (5), novos recursos continuam a ser introduzidos para ajudar os desenvolvedores de aplicações web, além de novos elementos continuarem a ser introduzidos com base em pesquisas (W3C, 2017c).

2.2 CSS 3

CSS ou Cascading Style Sheets, é um mecanismo adotado por desenvolvedores para adicionar estilo às páginas da web, incluindo o design, layout e variações de exibição para inúmeros tamanhos de telas e diversos dispositivos (W3Schools, 2019).

A proposta de Hakon Wium Lie para resolver o problema de formatar as informações mudou o conceito de todos os desenvolvedores web em 1994, quando propôs a criação do CSS, e em 1995, juntamente com Bert Bos, o projeto foi adotado pela W3C (W3C, 2017). Com o avanço dos estudos na área ao passar dos anos, o CSS foi se aperfeiçoando e hoje se encontra na versão 3.0. Visto de uma maneira geral, o CSS é o responsável por dar cores, melhorar o visual das páginas web, agregando bastante valor ao visual da página.

2.3 Java Script

Java Script é linguagem de programação considerada de alto nível. Ela é dinâmica, fracamente tipada e multiparadigma. Essa linguagem permite implementar funcionalidades que em seu âmbito são mais complexas, mostram em tempo real conteúdos atualizados, animações gráficas, dentre vários outras funcionalidades (MDN, 2019).

O JS foi criado por Brendan Eich em 1995 e teve seu nome iniciado como LiveScript e posteriormente atualizado para Java Script. A linguagem é mantida pela ECMA - Associação Europeia de Fabricantes de Computadores (GitHub, 2019). O JS passou por um longo processo de evolução e hoje se encontra na versão ECMAScript 2018.

2.4 NodeJs

O NodeJs é um interpretador de código JavaScript orientado a eventos de modo assíncrono, com o objetivo de migrar a programação do JS para o lado dos servidores, sendo capaz de criar milhares de conexões/requisições simultâneas em tempo real, sendo baseado no interpretador V8 JavaScript Engine (NodeJsOrg, 2019).

Em 2009, Ryan Dahl criou o NodeJS baseado na percepção de que o navegador não possuía a habilidade de diagnosticar a porcentagem em que se encontrava o carregamento de um arquivo no browser, sem consulta prévia a um servidor (NodeJsOrg, 2019). Desde então o nodejs vem evoluindo e atualmente se encontra na versão 10.16.0.

2.5 Vue.Js

Vue (pronúncia "view") é uma estrutura progressiva para construir interfaces com o usuário (VuejsOrg, 2019). Foi criado por Evan You, que enquanto desfrutava seu trabalho no Google, após adquirir conhecimentos acerca do AngularJs (Framework de Java Script), decidiu adequar os conceitos que mais apreciava do Angular e criar uma versão mais leve. Foi oficialmente lançada em 2014 (VueJsOrg, 2019).

A utilização deste framework é extremamente simples, podendo plugar diversas libs e várias outras possibilidades, fazendo com que o desenvolvedor trabalhe da maneira como preferir e facilitando seu trabalho. O VueJs encontra-se atualmente em sua versão estável 2.6.10, porém no presente trabalho foi utilizado uma versão inferior a atual.

2.6 Bootstrap

De acordo com W3C (2017a), o Bootstrap é um framework web com código-fonte aberto para ser usufruído durante o desenvolvimento de componentes relacionados à interface e front-end, assim deixando que a aparência se torne mais elegante, rápida e prática.

Esse framework foi criado por Mark Otto e Jacob "Fat"Thornton, em meados de 2010, durante uma Hack Week (semana de hackers), promovida pela Twitter. Apesar de ter sido criado há quase 10 anos, o projeto teve sua maior visualização em 2019 (Bootstrap Introduction, 2019). O Bootstrap encontra-se na versão 4.3, porém na realização do presente projeto foi utilizada uma versão inferior.

2.7 MySQL e Workbench

Em nução com Oracle, MySQL é a linguagem para banco de dados de código aberto mais famosa do mundo, sendo eficiente, confiável e fácil de usar, agregando um bom valor ao desenvolvimento do presente software. O MySQL foi criado por Michael Widenius em meados de 1979. No entanto, somente em 1995 a primeira versão do MySQL foi definitivamente lançada (Oracle Tutorial, 2019).

O MySQL Workbench é um software que permite a um desenvolvedor modelar, projetar e gerenciar visualmente um banco de dados. Esta ferramenta inclui tudo o que um desenvolvedor de dados precisa para criar modelos complexos de ER, engenharia direta e reversa, oferecendo também os principais recursos para executar tarefas difíceis em relação a gerenciamento de alterações e documentação (MySQL Workbench, 2019).

2.8 Visual Studio Code

O Visual Studio Code ou VSCode é um editor de código-fonte, assim como diversos aplicativos presentes em um sistema operacional (Windows, Mac e Linux) pode ser executado normalmente na área de trabalho. O VSCode é muito poderoso, possuindo um rico ecossistema de extensões para várias linguagens de programação (como C, Java Script, Node.js, CSS, etc) (Visual Studio Docs, 2019).

A Microsoft anunciou o Visual Studio Code em 2015 e também no mesmo ano obteve a licença permissiva do MIT - Massachusetts Institute of technology, porém, apenas em 2016 foi lançada para a web (Visual Studio Docs, 2019).

2.9 Kanban

Kanban é um termo originalmente japonês, que significa “cartão” ou “sinalização”. Essa técnica foi criada pelo engenheiro Taiichi Ohno, que havia recebido uma tarefa de aumentar a produtividade da empresa Toyota. Em 1953, Ohno viajou para o Estados Unidos, e durante essa viagem o engenheiro notou como os clientes de um supermercado pegavam os produtos das prateleiras. Observando o comportamento dos clientes na seguinte ordem:

- Pegando somente os produtos que eram necessários;
- Pegando os produtos no momento necessário;
- Pegando os produtos na quantidade necessária.

Ao seu retorno ao Japão, Ohno desenvolveu o Kanban, uma técnica que usufrui de cartões que contém informações para controlar a produção de acordo com a necessidade (TOYOTA PRODUCTION SYSTEM, 2019). A seguinte imagem pode ser ilustrada acerca da técnica kanban:



Figura 2.1: Imagem ilustrativa Kanban

No presente trabalho foi utilizado uma versão digital da técnica Kanban, o software Trello.

2.10 Engenharia de Software

O desenvolvimento de um software demanda um conjunto de atividades que devem ser realizadas para que esse objetivo possa ser alcançado. “A engenharia de software é uma tecnologia em camadas” (PRESSMAN 2006, p.39), ilustrado na Figura 2.2.



Figura 2.2: Camadas Engenharia de Software

Com essa constatação, é plausível dizer que o software deve passar pelos processos da engenharia para que o objetivo seja concluído com eficiência e eficaz.

2.10.1 Processos de software

Segundo Pressman, "a base para a engenharia de software é a camada de processos"(2006, p.39). Os processos são conjuntos ações que são realizadas na criação de algum software. Cada processo, por si só, possui um objetivo relativamente pequeno, porém bem definidos, que são de grande valor para concluir o propósito. Também sendo responsável por definir uma metodologia que deve ser estabelecida para a entrega efetiva de tecnologia de engenharia de software (PRESSMAN, 2006).

2.10.2 Métodos

Usualmente os engenheiros de software precisam desenvolver sistemas com um tamanho grande e uma complexidade alta, para que a qualidade seja priorizada durante todo o processo de construção do sistema. O escopo e a complexidade do software são modelados objetivando melhor compreensão e utilização (PRESSMAN, 2006). "Os métodos da engenharia de software baseiam-se em um conjunto de princípios básicos que governam cada área da tecnologia e inclui atividades de modelagem e outras técnicas descritti-

vas"(PRESSMAN 2006, p.40). Os métodos ágeis são uma alternativa para se desenvolver sistemas. Assim, no presente trabalho foi escolhida a metodologia *Scrum*.

2.10.3 Metodologia

De acordo com Pressman, "uma metodologia (framework) estabelece o alicerce para um processo de engenharia de software completo, por meio da identificação de um pequeno número de atividades estruturais aplicáveis a todos os projetos de software, independentemente de tamanho ou complexidade"(PRESSMAN 2006, p.40).

O *framework* escolhido para o desenvolvimento do presente trabalho foi o *Scrum*, por meio do qual é possível estabelecer os alicerces dos processos da engenharia de software por completo.

2.10.3.1 *Scrum*

Scrum é um framework no qual as pessoas podem resolver problemas complexos e adaptativos, enquanto entregam produtos de forma produtiva e criativa com o mais alto valor (Schwaber, Sutherland, 2013). De tal forma, encaixando-se perfeitamente no trabalho proposto.

O empirismo, por sua vez, é uma teoria filosófica que tem como princípio a noção de que o conhecimento é proveniente, principalmente, das experiências sensoriais. Isso implica dizer que são provenientes das sensações, sejam elas visuais, auditivas, táteis, etc. O *Scrum* é fundamentado nas teorias empíricas de controle dos processos, utilizando abordagens iterativas e incrementais, possuindo três pilares que apoiam esse controle:

- **Transparência:** "Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados"(Schwaber, Sutherland, 2013, p. 04).
- **Inspeção:** "Os usuários *Scrum* devem, frequentemente, inspecionar os artefatos *Scrum* e o progresso em direção a detectar variações"(Idem, 2013).
- **Adaptação:** Quando é constatado que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, implicando que ajustes devem ser realizados de forma mais breve possível (Idem, 2013).

Os papéis fundamentais para que o funcionamento do *Scrum* seja realizado com sucesso são:

- **Product Owner** é o responsável por ampliar o trabalho do time desenvolvimento e o valor do produto (Schwaber, Sutherland, 2013).
- **Time de Desenvolvimento** é composto por profissionais que executam seus trabalhos entregando uma versão utilizável do produto que irá se agregar ao produto final (Schwaber, Sutherland, 2013).
- **Scrum Master** é o responsável pelo entendimento e aplicação do *Scrum* (Schwaber, Sutherland, 2013).

2.10.3.2 Funcionamento do Scrum

O ponto de partida do *Scrum* é a visão do produto que é realizada pelo *Product Owner*. Esse panorama pode ser definido como: o que deve ser criado, o objetivo. Esta visão é transcrita em uma lista de funções que sofrem um processo chamado de grooming, este processo é uma refinação da lista de funções, ou seja, a lista é ordenada de acordo com a prioridade das funções, resultando no *Product Backlog*, “O Backlog do Produto é uma lista ordenada de tudo que deve ser necessário no produto, e é uma origem única dos requisitos para qualquer mudança a ser feita no produto “ (Schwaber, Sutherland, 2013). O PB (*Product Backlog*) é construído pelo *Product Owner* e elenca todas as funções, que devem ser feitas no produto, estes possuem os atributos de descrição, ordem, estimativa e valor. Os itens que estão presentes no topo da lista são os que possuem maior ordem.

Com o *Product Backlog* finalizado, se iniciam as *Sprints*, “O coração do *Scrum* é a *Sprint*” (Schwaber, Sutherland, 2013, p.08). A *Sprint* é composta por: um planejamento das *Sprints*, reuniões diárias, o trabalho de desenvolvimento, uma revisão *Sprint* e a retrospectiva da *Sprint* (Schwaber, Sutherland, 2013).

Para executar o planejamento das *Sprints* (*Sprint Planning*) é feita uma reunião. Nesta são traçadas quais as funções que irão ser construídas ao decorrer da *Sprint*. É construído, então, o *Sprint Backlog*, que consiste em "um conjunto de itens do Backlog do Produto selecionados para a *Sprint*" (Schwaber, Sutherland, 2013, p.15).

Durante o processo de execução da *Sprint*, ocorrem as reuniões diárias que é um evento time-boxed de 15 minutos com o objetivo de sincronizar as atividades e criar um plano para as próximas 24 horas. Durante a reunião os membros do time de desenvolvimento realizam três perguntas deixando esclarecidos os seguintes pontos:

- O que eu fiz ontem que ajudou a cumprir o objetivo da *Sprint*?
- O que eu farei hoje para ajudar a cumprir o objetivo da *Sprint*?
- Existe algum obstáculo que impeça o objetivo da *Sprint*?

Assim essas reuniões de tempo relativamente curto acontecem todos os dias o que acarreta em um aumento da probabilidade do time de desenvolvimento atingir o objetivo (Schwaber, Sutherland, 2013).

Todos os dias são construídas as funções ou partes delas utilizando as ferramentas descritas anteriormente. O time de desenvolvimento é o responsável por esta parte, onde o desenvolvimento ocorre diariamente. Para a auxiliar na construção das funções foi utilizado o Kanban capítulo 2.9.

Chegando ao final, da *Sprint*, antes de seu encerramento, são realizadas duas atividades que são:

- Revisão *Sprint* ou *Sprint Review*: nesta parte é revisado o que foi realizado, com o objetivo de verificar se a execução dos processos está de acordo com o objetivo. O *Product Owner* esclarece quais funções do Backlog do produto estão finalizados ou não. Também é discutido o que foi bem durante a *Sprint*, quais os problemas que foram encontrados, como estes problemas foram resolvidos e se o *Product Backlog* precisar passar por mudanças.
- Retrospectiva *Sprint*: este trecho é uma oportunidade para inspecionar a si próprio e criar um plano para melhorias, analisando os pontos negativos e positivos que foram encontrados durante todo o processo da *Sprint*. A retrospectiva ocorre depois da Revisão *Sprint* e assim se encerra a *Sprint*.

Com encerramento de uma *Sprint*, parte do produto é finalizada, podendo ser entregue ao cliente para utilização. Uma nova *Sprint* se inicia imediatamente após a conclusão da anterior retornando para etapa de planejamento da *Sprint* e assim esse ciclo se repete até que o produto esteja totalmente finalizado (Schwaber, Sutherland, 2013).

3 DESENVOLVIMENTO DO TRABALHO PROPOSTO

3.1 Introdução ao Desenvolvimento do LIFEDU

Como visto anteriormente, várias ferramentas foram apresentadas e a utilização em conjuntos destas é de grande importância. Neste capítulo será reportado a utilização de todas as ferramentas e metodologias para que os objetivos do trabalho possam ser alcançados.

3.2 Planejamento

Para a execução do presente trabalho foi-se construído um planejamento, para que todas as atividades fossem executadas de acordo com o prazo e esforço, seguindo a seguinte ordem:

- Leitura de artigos e livros para enriquecimento do conhecimento;
- Estudo e capacitação acerca das linguagens escolhidas;
- Início da escrita do TCC - Trabalho de Conclusão de Curso;
- Desenvolvimento do sistema;
- Realização de testes e coleta de resultados.

3.2.1 Cronograma

As atividades foram desenvolvidas conforme o cronograma mostrado na Tabela 3.1.

Tabela 3.1: Cronograma de atividades

	2019						
	Jan	Fev	Mar	Abr	Mai	Jun	Jul
Leitura Artigos	X	X	X	X	X		
Estudos Linguagens		X	X	X	X		
Escrita TCC				X	X	X	X
Desenvolvimento						X	X
Teste e Resultados						X	X
Defesa							X

3.3 Execução

3.3.1 Coleta de Dados

3.3.1.1 Levantamento de Requisitos

O levantamento de requisitos é uma das etapas no desenvolvimento de sistemas de informação. Essa etapa é responsável por diagnosticar, identificar e modelar as necessidades a serem atendidas pelo sistema de informação. Logo, essa atividade é cada vez mais imprescindível e relevante em cenários dinâmicos. A engenharia de requisitos é formada por um conjunto de técnicas utilizadas de forma a levantar, detalhar, documentar e validar os requisitos de um produto de software (FILHO, 2000). O levantamento de requisitos se iniciou com o entendimento do funcionamento da LIFE e, para tanto, foram realizadas atividades de entrevistas, observação das tarefas cotidianas e também a identificação dos atores.

A partir das informações coletadas, dentre todas as funções vistas e revisadas, apenas as seguintes funções exercidas pelos entrevistados serão incorporadas pelo sistema, podendo ser melhor visualizada na figura 3.1:

Funções
1. Gerenciamento de Matérias
2. Gerenciamento de reuniões
3. Gerenciamento de conteúdos(vídeos)
4. Gerenciamento de usuários

Figura 3.1: Lista de funções

Em sua continuidade, foram feitas no projeto as devidas construções das funcionalidades, que foram classificadas da seguinte maneira:

- **Requisitos Funcionais:** Os requisitos funcionais descrevem a funcionalidade ou os serviços que se espera que o sistema realize em benefício dos usuários (FILHO, 2000).
- **Requisitos Não Funcionais:** Os requisitos não funcionais são aqueles que não dizem respeito diretamente às funcionalidades fornecidas pelo sistema. Podem estar relacionados a propriedades de sistemas emergentes, como confiabilidade, tempo de resposta, espaço em disco, desempenho e outros atributos de qualidade do produto (FILHO, 2000).

3.3.1.2 Requisitos Funcionais

- **Cadastro de Alunos (RF1):** O sistema deve disponibilizar uma área para que visitantes do sistema web possam se cadastrar, e que as devidas funcionalidades aluno sejam empregadas a este cadastro.
- **Cadastro de Professor (RF2):** O sistema deve disponibilizar uma área para que os administradores possam cadastrar os professores que irão lecionar suas devidas matérias, assim agregando suas devidas funções e acessos.
- **Cadastro de Administrador (RF3):** O sistema deve disponibilizar uma área para que possam ser cadastrados os administradores que possuem um maior nível de acesso.

- Autenticação (RF4): O sistema deve disponibilizar uma tela de login na qual o usuário irá entrar com suas credenciais e terá acesso às funcionalidades de acordo com seu papel(aluno, professor ou administrador).
- Cadastro de Materiais (RF5): O sistema deve disponibilizar uma tela de cadastro de materiais que serão de cunho do departamento específico.
- Cadastro de reuniões (RF6): O sistema deve disponibilizar uma tela de cadastro de reuniões que será realizada pelo administradores e professores.
- Cadastro de conteúdo - vídeos (RF7): O sistema deve disponibilizar uma tela de cadastro dos vídeos que foram gravados para que sua inserção no sistema seja possível.
- Edição de aluno (RF8): O sistema deve disponibilizar uma tela de para que os usuários do tipo "aluno"possam editar e atualizar suas informações de cadastro como senha, nome e demais informações.
- Edição de professor (RF9): O sistema deve disponibilizar uma tela de edição e atualização para que os usuários do tipo professor possam editar suas informações de registro.
- Edição de administrador (RF10): O sistema deve disponibilizar uma tela de edição para que os administradores possam editar e atualizar suas informações pessoais
- Edição de materiais (RF11): O sistema deve disponibilizar uma tela de edição para que os materiais de cunho do departamento que foram registrados possam ser editados e atualizados pelos administradores do sistema.
- Edição de reuniões (RF12): O sistema deve disponibilizar uma tela de edição para que os administradores possam editar informações referentes às reuniões cadastradas.
- Edição de conteúdo - vídeos (RF13): O sistema deve disponibilizar uma tela de edição e atualização do conteúdo referente aos vídeos que estão contidos no sistema.
- Remoção de aluno (RF14): O sistema deve disponibilizar uma tela para que os administradores possam remover alunos e uma outra tela para que os alunos que assim desejarem possam excluir suas contas.
- Remoção de professor (RF15): O sistema deve disponibilizar uma tela para que assim os professores que desejem ou que não ocupem mais seus cargos possam excluir suas contas do sistema, assim como os administradores também possuem esta função.
- Remoção de administrador (RF16): O sistema deve disponibilizar uma tela para que os administradores possam excluir outros administradores removendo os que não fazem mais parte da equipe.
- Remoção de materiais (RF17): O sistema deve disponibilizar uma tela para que os administradores possam excluir os materiais que não mais utilizam.

- Remoção de reuniões (RF18): O sistema deve disponibilizar uma tela de remoção de reuniões que não mais serão executadas ou que desejam excluir por demais motivos.
- Remoção de conteúdo - vídeos (RF19): O sistema deve disponibilizar uma tela de remoção de vídeos para que tanto professores quanto administradores do sistema possam retirar o conteúdo que não mais os convém.
- Geração de atas - reuniões (RF20): O sistema deve disponibilizar uma tela para os administradores criarem suas referentes atas das reuniões que foram cadastradas.
- Edição de atas - reuniões (RF21): O sistema deve disponibilizar uma tela de edição e atualização para que os administradores possam manipular o conteúdo referente às atas das reuniões.
- Remoção de atas - reuniões (RF22): O sistema deve disponibilizar uma tela para que os administradores possam remover as atas das reuniões.
- Geração de empréstimos - materiais (RF23): O sistema deve disponibilizar uma tela para que os administradores possam gerir os empréstimos dos materiais cadastrados no sistema.
- Edição de empréstimos - materiais (RF24): O sistema deve disponibilizar uma tela de edição e atualização dos empréstimos que foram registrados no sistema.
- Remoção de empréstimos - materiais (RF25): O sistema deve disponibilizar uma tela para remover os materiais que foram emprestados e que devidamente foram devolvidos.

3.3.1.3 *Requisitos Não Funcionais*

- Compatibilidade com dispositivos móveis (RFN1): O sistema deve ser compatível, podendo ser acessado por dispositivos móveis.
- Acessos Simultâneos (RFN2): O sistema deve ser acessado simultaneamente por vários dispositivos.

3.3.1.4 *Atores*

Os atores em si não são funções do sistema, eles representam algo ou alguém que deve interagir com o sistema. Atores podem ser definidos ou encontrados, realizando a definição do problema, conversas com clientes ou por observação das tarefas realizadas no ambiente em que o sistema irá funcionar. Um ator pode:

- Somente prover informações para o sistema.
- Prover e aderir informações para o sistema.
- Somente aderir informações para o sistema.

Os presentes atores foram definidos:

- Aluno: o aluno é aquele que irá usufruir do conteúdo das aulas que o sistema irá proporcionar. Este tem permissões para visualizar, categorizar os conteúdos (vídeos) e também possuindo permissão para atualização e edição de dados pessoais, devendo ser devidamente registrado com seu email e senha.

- Professor: o professor é aquele que irá ministrar seus conteúdos (vídeos), este possuindo permissões de atualização, edição e exclusão tanto de dados pessoais quanto de conteúdo.
- Administrador: O administrador é aquele que irá usufruir das demais funções que o sistema irá oferecer, possuindo todas as permissões exceto a de lançar comentários sobre vídeos e categorizá-los, tendo controle do sistema.

3.3.1.5 Modelagem

"Um modelo é uma simplificação da realidade "(Rumbaugh et al. 1999, p.06). A modelagem é uma parte essencial para todos os tamanhos de projetos de software (pequeno, médio ou grande). No desenvolvimento do presente software é necessário uma visão panorâmica do sistema. A UML é uma linguagem de modelagem de propósito geral que todos os modeladores podem usar (Rumbaugh et al. 1999).

3.3.1.6 Diagrama de caso de uso

O comportamento do software em questão é documentado em um modelo de caso de uso. Na UML, um ator é representado como um stickman. Um caso de uso, por sua vez, é representado como uma figura oval. Considerando os requisitos que foram levantados e descritos anteriormente, os três atores do sistema podem ser demonstrado no seguinte caso de uso do software (figura 3.2):

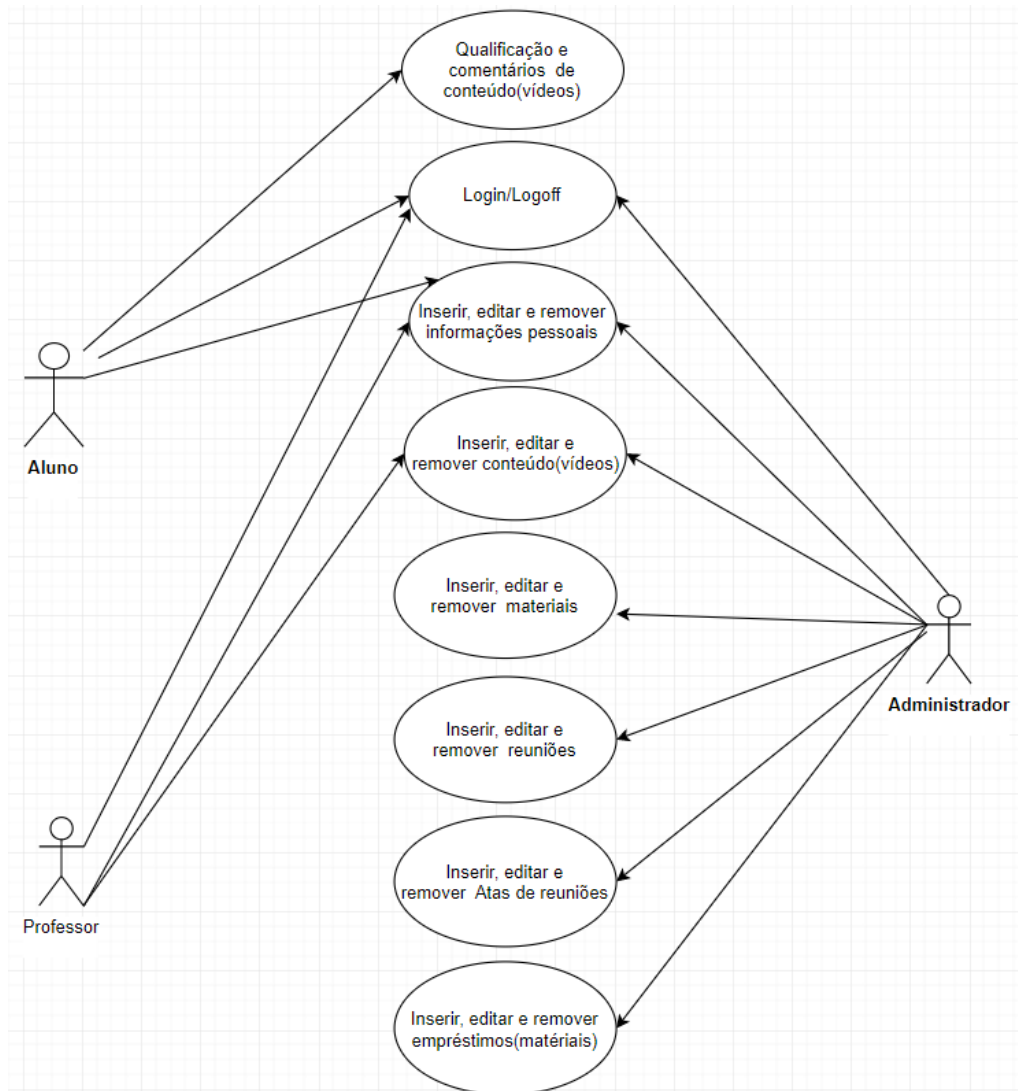


Figura 3.2: Diagrama de caso de uso

3.3.1.7 Permissões

Para exemplificar as relações de permissão que cada tipo usuário pode ter, foi construído uma tabela para uma melhor compreensão, conforme a figura 3.3:

	Administrador	Professor	Aluno
Fazer Login/Logoff	Sim	Sim	Sim
Inserir, editar e remover dados pessoais	Sim	Sim	Sim
Inserir, editar e remover conteúdo(vídeos)	Sim	Sim	Não
Inserir, editar e remover materiais	Sim	Não	Não
Inserir, editar e remover atas(Reuniões)	Sim	Não	Não
Inserir, editar e remover empréstimos(matérias)	Sim	Não	Não

Figura 3.3: Tabela de permissões

- **Nível de administrador:** possui totalidade de funções, abrangendo o maior nível de acesso;
- **Nível de professor:** possui um menor número de permissões em relação ao administrador. Logo, seus acessos ao sistema são limitados;
- **Nível de aluno:** possui um pequeno número de permissões, o que lhe concede menor número de acesso a funções quando comparado aos outros dois níveis.

3.3.1.8 Análise de Viabilidade do Projeto

Para que o projeto se torne viável, conhecimentos na área de linguagens de programação, estrutura e análises de algoritmos devem ser de total compreensão para a produção do software.

3.3.1.9 Viabilidade Técnica

- Conhecimento das linguagens, ferramentas e frameworks estipulados;
- Conhecimento sobre bancos de dados relacionais;
- Funções do sistema com complexidade baixa.

3.3.1.10 Viabilidade Econômica

- Nenhum software, ferramentas utilizadas apresentaram custo no período de produção do sistema.
- Devido a custos relacionados a domínio e hospedagem, se torna inviável a disponibilização do produto na web momentaneamente.

3.3.2 Desenvolvimento

3.3.2.1 Product Backlog

Como visto nos capítulos anteriores, a visão do produto (página 18) pode ser concluída com o auxílio do capítulo 4.3.1 (coleta de dados). Assim foi criado e apresentado o *Product Backlog*, conforme explicitado na Figura 3.4:

Identificação	Item	Priorização	Estimativa	Descrição
1	Desenvolvimento do BD	Alta	5 dias	Desenvolver o banco de dados de maneira relacional, para que toda a aplicação à utilize.
2	Desenvolvimento do módulo de Administrador	Alta	10 dias	Módulo responsável pela administração do sistema, podendo inserir, deletar e editar em nível 3.
3	Desenvolvimento do módulo de Professor	média	5 dias	Módulo responsável pelos usuários do tipo professor, podendo inserir, deletar e editar em nível 2.
4	Desenvolvimento do módulo de Aluno	baixa	3 dias	Módulo responsável pelos usuários do tipo aluno, podendo inserir, deletar e editar em nível 1.
5	Incorporação dos módulos	baixa	2 dias	Integrar todos os módulos ao sistema

Figura 3.4: *Product Backlog*

3.3.2.2 *Sprints Backlog*

De acordo com o capítulo 2.11.3.2, as reuniões foram executadas, e em conciliação com o *Product Backlog*, foram criadas as cinco *Sprints Backlog*, cujo conteúdo pode ser visualizado a partir da figura 4.5 até a figura 4.9. A utilização do Kanban capítulo 2.9 está presente no desenvolvimento das *Sprints* a ferramenta foi de grande auxílio para organização e controle das funções.

3.3.2.3 *Sprint 1*

identificação	item	Priorização	Estimativa	Descrição
1	Criação do Diagrama de Entidade e Relacionamento	Alta	2 dias	Elaborar e planejar o diagrama de entidade e relacionamento
2	Criar Tabelas	média	1 dia	Criação de tabelas
3	Devida colocação de chaves Estrangeiras	média	2 dias	Alocação e criação das chaves estrangeiras

Figura 3.5: *Sprint Backlog 1*

Através das funções que foram planejadas e categorizadas com o *Product Backlog*, dá-se início à primeira *Sprint*, tal como demonstrado na Figura 4.5. Essa *Sprint* se remete à criação do banco de dados do projeto, começando com a sua modelagem, onde foi criado o diagrama de entidade e relacionamento, chaves estrangeiras e os relacionamentos em acordo com os requisitos que foram coletados.

Ao final da execução do desenvolvimento da primeira *Sprint* foi constatado o sucesso na execução das tarefas que seguem a seguinte ordem:

1. Criação do diagrama de entidade e relacionamento.
2. Criação das tabelas.
3. Devida colocação das chaves estrangeiras.

Como declarado nos capítulos anteriores, as reuniões diárias foram executadas eventualmente ao decorrer da execução das sprints e as três perguntas básicas foram realizadas agregando valor no processo e foi realizada a primeira *Sprint Review*, concluindo que:

1. Todas as funções da *Sprint* estão finalizadas.
2. O que está sendo feito está de acordo com o esperado.
3. A *Sprint* foi executada dentro do prazo.
4. Não há necessidade de mudanças ou atualizações no *Product Backlog*.

5. Não houve problemas.

Em acordo com os capítulos anteriores, executa-se a primeira Retrospectiva, cujos resultados demonstraram:

1. Pontos Positivos:

- Tarefas executadas dentro do prazo.
- Agilidade de produção.

2. Pontos Negativos:

- Nenhum.

3. O que deve ter continuidade:

- Agilidade de produção.

4. O que deve ter término:

- Nenhuma ação.

Assim que todas as tarefas foram executadas com êxito, a *Sprint* é encerrada, dando início a próxima *Sprint*. Este ciclo se repete até que o produto final esteja finalizado.

3.3.2.4 *Sprint 2*

Com o encerramento da *Sprint 1*, foram selecionadas, de acordo com o *Product Backlog*, as tarefas referentes ao desenvolvimento da *Sprint 2*.

identificação	item	Priorização	Estimativa	Descrição
1	Criações das telas de login, cadastro, edição e remoção de materiais, reuniões, atas, empréstimos, aulas e administrador	Alta	2 dias	Criação de todas as telas para as devidas funções.
2	Inserção, remoção e edição referente à reuniões e atas	Alta	2 dias	Construções das devidas funções para manipulação no banco de dados para reuniões e atas
3	Inserção, remoção e edição referente materiais e empréstimos	Alta	2 dias	Construções das devidas funções para manipulação no banco de dados para materiais e empréstimos
4	Inserção, remoção e edição referente à conteúdo(vídeos)	Alta	2 dias	Construções das devidas funções para manipulação no banco de dados para conteúdos(vídeos)
5	Inserção, remoção e edição referente à dados pessoais dos administradores	média	1 dia	Construções das devidas funções para manipulação no banco de dados para dados pessoais
6	Criação de logoff	baixa	1 dia	Criação de logoff

Figura 3.6: *Sprint Backlog 2*

Essa *Sprint* inicia-se com criação das telas do sistema, sendo efetuadas para uma eficiente adequação a utilização do usuário. Dando prosseguimento, foram produzidas operações de CRUD(Criação, consulta, atualização e destruição) em relação às reuniões e as suas respectivas atas. Posteriormente, as mesmas operações foram realizadas para materiais e empréstimos, conteúdo (vídeos) e dados pessoais, finalizando com a tarefa de logoff. Com término do desenvolvimento da segunda *Sprint* fora constatado o sucesso na execução das tarefas, que possuem a ordem:

1. Criação das telas do sistema para todos os níveis;
2. Execução do CRUD referente às reunião e atas;
3. Execução do CRUD referente aos materiais e empréstimos;
4. Execução do CRUD referente ao conteúdo (vídeos);
5. Execução do CRUD referente aos dados pessoais;
6. Execução do logoff.

Prosseguindo com os trabalhos foi-se iniciada a segunda *Sprint Review*, ficando evidente que:

1. Todas as funções da *Sprint* estão finalizadas.
2. O que está sendo feito está de acordo com o esperado;
3. A *Sprint* foi executada dentro do prazo;
4. Não há necessidade de mudanças ou atualizações no *Product Backlog*.

Ao término desta etapa, a segunda Retrospectiva foi executada e foi concluído que:

1. Pontos Positivos:
 - Tarefas executadas dentro do prazo;
 - Reutilização de código.
2. Pontos Negativos:
 - Atividades cansativas
3. O que deve ter continuidade:
 - Reutilização de código.
 - Técnicas de relaxamento.
4. O que deve ter término:
 - *Sprint* longa, assim evitando estresse.

Com todas as tarefas realizadas com êxito, a segunda *Sprint* é encerrada.

3.3.2.5 *Sprint 3*

Se iniciando a terceira *Sprint*, foram selecionadas as tarefas em concordância com o *Product Backlog*.

identificação	item	Priorização	Estimativa	Descrição
1	Controle de acesso no nível de Professor	Alta	2 dias	Liberar acesso para professor, com as devidas funções
2	Liberar Acesso a visualização de reuniões	média	1 dia	Liberar acesso para professor, com as devidas funções
3	Liberar Acesso à conteúdo(vídeo) inserção, remoção e edição	média	1 dia	Liberar acesso para professor, para manipulação de conteúdo
4	Liberar Acesso à dados pessoais inserção, remoção e edição	média	1 dia	Liberar acesso para professor, para manipulação de dados pessoais

Figura 3.7: *Sprint Backlog 3*

A terceira *Sprint* dá-se início com o controle de acesso, ou seja, quais as funções já elaboradas o professor poderá utilizar. Caso seja necessário o acesso a alguma função que não foi estipulada no *Product Backlog*, as devidas mudanças devem ser implementadas. Entretanto, nenhuma mudança foi constatada para esta tarefa. Com o controle devidamente feito para o nível professor, o desenvolvimento da terceira *Sprint* é encerrado.

1. Controle de acesso no nível;
2. Liberar visualização às reuniões;
3. Liberar manipulação de conteúdo (vídeos);
4. Liberar manipulação aos dados pessoais.

Ademais, foi-se iniciada a terceira *Sprint Review*, ficando evidente que:

1. Todas as funções e liberações da *Sprint* estão finalizadas.
2. O que está sendo feito está de acordo com o esperado;
3. A *Sprint* foi executada dentro do prazo;
4. Não há necessidade de mudanças ou atualizações no *Product Backlog*.

Com o encerramento desta etapa, a terceira Retrospectiva foi executada e ficou constatado que:

1. Pontos Positivos:
 - Tarefas executadas dentro do prazo;
 - Reutilização de código.
2. Pontos Negativos:
 - Nenhum.
3. O que deve ter continuidade:
 - Reutilização de código.
4. O que deve ter término:
 - Nenhuma ação.

Assim ao final da *Sprint 3* realizada com êxito, a terceira *Sprint* chega ao seu fim.

3.3.2.6 *Sprint 4*

Prosseguindo com a execução da quarta *Sprint*, de acordo com o *Product Backlog*, as seguintes tarefas foram executadas:

identificação	item	Priorização	Estimativa	Descrição
1	Controle de acesso no nível de Aluno	Alta	1 dia	Liberar acesso para aluno com as devidas funções.
2	Liberar visualização à conteúdo(vídeos)	média	1 dia	Liberar acesso para aluno para visualização de conteúdo
3	Liberar Acesso à dados pessoais inserção, remoção e edição	média	1 dia	Liberar acesso para aluno para manipulação de dados pessoais

Figura 3.8: *Sprint Backlog 4*

A seguir, se inicia a terceira *Sprint Review* ficando evidente que:

1. Todas as funções e liberações da *Sprint* estão finalizadas.
2. O que está sendo feito está de acordo com o esperado;
3. A *Sprint* foi executada dentro do prazo;
4. Não há necessidade de mudanças ou atualizações no *Product Backlog*.

Com o encerramento desta etapa, a terceira Retrospectiva foi executada e ficou constatado que:

1. Pontos positivos:
 - Tarefas executadas dentro do prazo;
 - Reutilização de código.
2. Pontos Negativos:
 - Nenhum.
3. O que deve ter continuidade:
 - Reutilização de código.
4. O que deve ter término:
 - Nenhuma ação.

Assim, com o êxito apresentado pela *Sprint 4*, ela se encerra, dando início à próxima.

3.3.2.7 *Sprint 5*

Adiante, inicia-se a quinta e última *Sprint*, que em concordância com o *Product Backlog*, exerce as seguintes tarefas:

Na quinta *Sprint* foi realizada a utilização de todos os módulos do sistema, a fim de testar todas as funcionalidades que o sistema apresenta, concluiu-se que sistema está pronto para utilização. Ao se completar todas as tarefas em relação a *Sprint 5*, pôde ser notada a seguinte ordem de ação:

identificação	item	Priorização	Estimativa	Descrição
1	utilização do sistema como Administrador, Professor e Aluno	Alta	2 dias	Utilizar o sistema nos três níveis

Figura 3.9: *Sprint Backlog 5*

1. Todas as funções da *Sprint* foram testadas com êxito.
2. O que está sendo feito está de acordo com o esperado;
3. A *Sprint* foi executada dentro do prazo;
4. Não há necessidade de mudanças ou atualizações no *Product Backlog*;
5. Houve um erro na inserção de uma matéria;
6. A devida função foi consertada.

Iniciando a última *Sprint Review* ficou claro que:

1. Todas as funções da *Sprint* foram testadas com êxito.
2. O que está sendo feito está de acordo com o esperado;
3. A *Sprint* foi executada dentro do prazo;
4. Não há necessidade de mudanças ou atualizações no *Product Backlog*;
5. Correção de erros lógicos;

A quinta Retrospectiva foi executada e ficou constatado que:

1. Pontos positivos:
 - Tarefas executadas dentro do prazo;
 - Todas as verificações obtiveram êxito.
2. Pontos Negativos:
 - Ocorreu um erro inesperado;
 - Frustração.
3. O que deve ter continuidade:
 - Reutilização de código.
4. O que deve ter término:
 - Frustração.

3.3.2.8 *Estrutura Geral*

A estrutura geral que fora construída no desenvolvimento será disponibilizada e anexada a este documento.

3.3.2.9 Justificativa da Escolha dos Métodos/Ferramentas

Dentre todos os benefícios que a tecnologia pode nos agregar, uma delas é tornar as tarefas e suas execuções mais proveitosas e rápidas, pois a inovação não é sobre a invenção, mas sim gerar valor para o usuário. Partindo-se desse espírito de sempre querer inovar e buscar conhecimento, foram escolhidas as ferramentas utilizadas neste presente projeto, no intuito de aprender novas tecnologias que o mercado disponibiliza para a solução dos problemas.

3.3.2.10 Recursos Utilizados para Projeto/Desenvolvimento

Para a codificação do presente sistema foi utilizada a IDE Visual Studio Code, na sua versão 1.36. No que se refere às linguagens de programação, todas foram utilizadas de forma a atingir o objetivo. A administração dos arquivos foi realizada pelo VSCode. No que tange os assuntos associados ao banco de dados, foi utilizado a ferramenta MySQL Workbench, na sua versão 8.0 e todas as operações e modelagens acerca do BD foram executadas pela mesma. Os frameworks utilizados na construção do sistema foram devidamente instalados via NPM, que é o gerenciador de pacotes do Node, uma ferramenta indispensável e extremamente importante e relevante.

4 ANÁLISE DE RESULTADOS

Inicialmente, as linguagens e frameworks propostos exigem um nível de conhecimento regular, visto que o sistema web não é de extrema complexidade e a conjunção desta é viável e bastante eficiente para construção de softwares de diferentes âmbitos e tamanhos.

A disponibilização de vídeos com conteúdos disciplinares é de grande ajuda para auxiliar o desenvolvimento dos alunos em sua carreira acadêmica. Um sistema web devidamente construído pode auxiliar esse desenvolvimento. Foi constatada a criação de um sistema web que atende essa visão e que, com sua devida utilização, pode agregar valor para diversos alunos.

Os processos de gerenciamento de materiais, empréstimos e reuniões podem ser automatizados com um sistema, ficando evidente que o software desenvolvido no presente trabalho possui essa capacidade.

Em relação à metodologia, esta foi adaptada para o número de elementos presentes no time, o seu entendimento é de complexidade baixa, porém, a sua utilização é de extrema importância e relevância, constatando que a sua utilização é efetiva em diferentes âmbitos e tamanhos de softwares.

5 CONSIDERAÇÕES FINAIS

Quando se iniciou o desenvolvimento do software LIFEDU, constatou-se que existia um grande número de alunos que sofrem com o desnivelamento proporcionado pelo sistema de educação brasileira, o que acarreta um índice de retenção em disciplinas como Cálculo I, II e GAAL. O desenvolvimento do sistema partiu da necessidade de uma disponibilização de conteúdo (vídeos) educacionais para determinadas disciplinas, objetivando solucionar esse problema. Durante a execução do trabalho, verificou-se que é viável a construção do sistema nos âmbitos técnicos como linguagens de programação, utilização de frameworks e banco de dados, onde a integralização das ferramentas utilizadas foram bastante eficientes cumprindo com o esperado.

No decorrer do desenvolvimento, o framework scrum foi de extrema importância pois o *Product Backlog* deixa bem evidente as funções que foram construídas. Além disso, a execução das sprints conciliadas com o kanban agregaram muito valor ao gerenciamento de funções e tempo. Para concluir, a utilização do scrum foi de grande ajuda para a finalização de todas as tarefas em seus respectivos períodos, ficando evidente a sua eficiência no desenvolvimento do presente trabalho.

Para futuras implementações do sistema é viável a disponibilização do LIFEDU para os estudantes da UFVJM, através de um sistema online, podendo assim, de fato, contribuir com a diminuição de retenção dos alunos, uma vez que momentaneamente se torna inviável tornar a aplicação acessível devido às viabilidades econômicas de domínio e servidor.

REFERÊNCIAS

COLL, MONEREO. César Coll e Carles Monereo . Psicologia da educação virtual: aprender e ensinar com as tecnologias da informação e da comunicação. Porto Alegre: Artmed, 2010. Acesso em: 01 de Jul.2019. Citado na página 11.

MENEZES, Ebenezer Takuno de; SANTOS, Thais Helena dos. Verbete níveis de ensino. Dicionário Interativo da Educação Brasileira - Educabrazil. São Paulo: Midiamix, 2001. Disponível em: <<https://www.educabrazil.com.br/niveis-de-ensino/>>. Acesso em: 01 de jul. 2019. Citado na página 11.

Unesco. Disponível em: <<http://www.unesco.org/new/pt/brasil>>. Acesso em: 20 Nov.2018. Citado na página 11.

Memorando nº 355/2018, 09 de agosto de 2018. Disponível em: <http://www.ufvjm.edu.br/formularios/cat_view/430-479-480-536-assuntos-2018.html> . Acesso em : 01 Jan.2019. Citado na página 12.

Memorando nº 115/2018/Prograd, 16 de março de 2018. Disponível em: <<http://www.ufvjm.edu.br/formularios/>>. Acesso em: 03 Jan.2019. Citado na página 12. Birrell, N.D. (1985). A Practical Handbook for Software Development. [S.l.]: Cambridge University Press. citado na página 11.

DECOM. Cursos, Sistemas de Informação. 2019. Disponível em: <<http://decom.ufvjm.edu.br/site/index.php/mn-cursos/mn-sistemas-informacao>>. Acesso em: 20/06/2019. Citado na página 11.

Pearson International, the learning curve. 2012. Disponível em: <<https://www.pearson.com/corporate/news/media/news-announcements/2012/11/pearson-launches-the-learning-curve.html>>. Acesso em: 20/06/2019. Citado na página 12.

TOYOTA, Sistema Toyota de Produção<TOYOTA PRODUCTION SYSTEM> , Disponível em:<<https://www.toyota.com.br/mundo-toyota/toyota-production-system/>>. Acesso em: 20/06/2019. Citado na página 12.

PRESSMAN, R. S. Engenharia de Software. Rio de Janeiro: McGraw-Hill, 2006. Citado nas páginas 16 e 17.

SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. 2013. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v1/scrum-guideus.pdf>>. Acesso em: 02/06/2019. Citado na página 16.

W3C. HTML 5 TUTORIAL. 2017. Disponível em: <<https://www.w3.org/>> Acesso em: 05/06/2019. Citado na página 13.

W3Schools. W3Schools TUTORIAL. 2019. Disponível em:<<https://www.w3schools.com/>> Acesso em: 06/06/2019. Citado na página 13.

Github. Github Getting Started. 2019. Disponível em: <<https://tableless.github.io/iniciantes/manual/js/>> Acesso em: 06/06/2019. Citado na página 14.

VuejsOrg. VueJs TUTORIAL. 2019. Disponível em: <<https://vuejs.org/>> Acesso em: 07/06/2019. Citado na página 14.

Bootstrap. Bootstrap Introduction. 2019. Disponível em: <<https://getbootstrap.com/docs/4.3/getting-started/introduction/>> Acesso em: 08/06/2019. Citado na página 14.

Oracle. Database SQL reference. Disponível em: <https://docs.oracle.com/cd/B19306_01/server.102/b14200/toc.htm> Acesso em: 09/06/2019. citado nas páginas 14 e 15.

Visual Studio. Visual Studio Docs. Disponível em: <<https://code.visualstudio.com/docs>> Acesso em: 15/06/2019. citado na página 15.

FILHO. PAULA FILHO. Wilson de Pádua. Engenharia de Software: fundamentos, métodos e padrões. São Paulo: LTC Editora, 2000. Citado na página 22.

Rumbaugh et al. Grady Booch. James Rumbaugh. Ivar Jacobson. 1999. UML Guia do Usuário. Citado na página 25.

APÊNDICE A ANEXOS

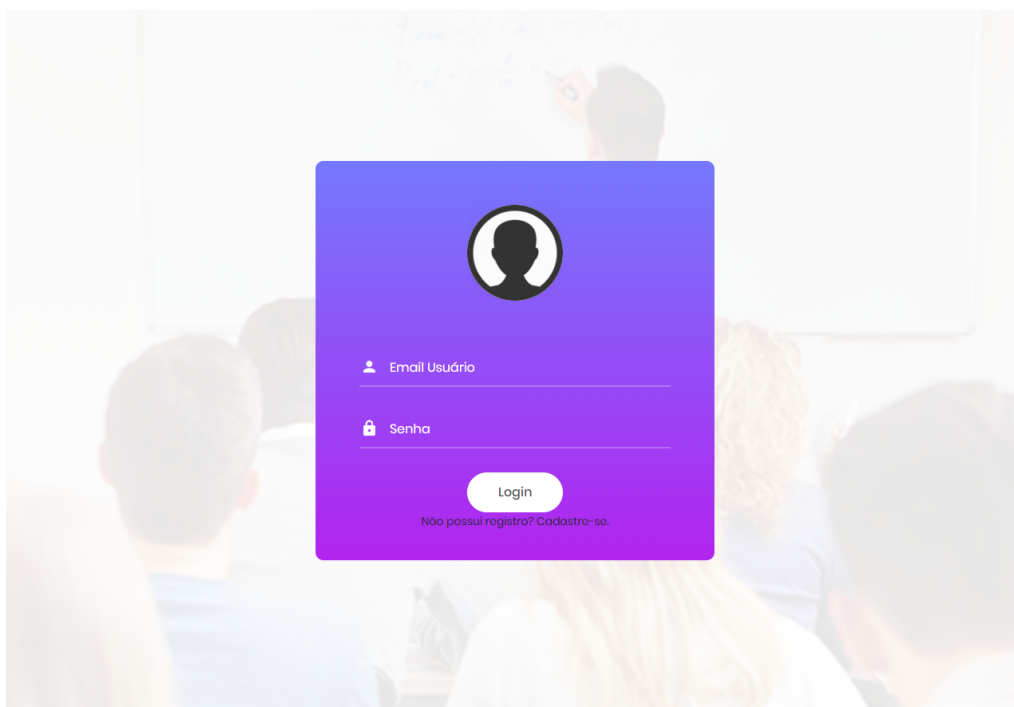
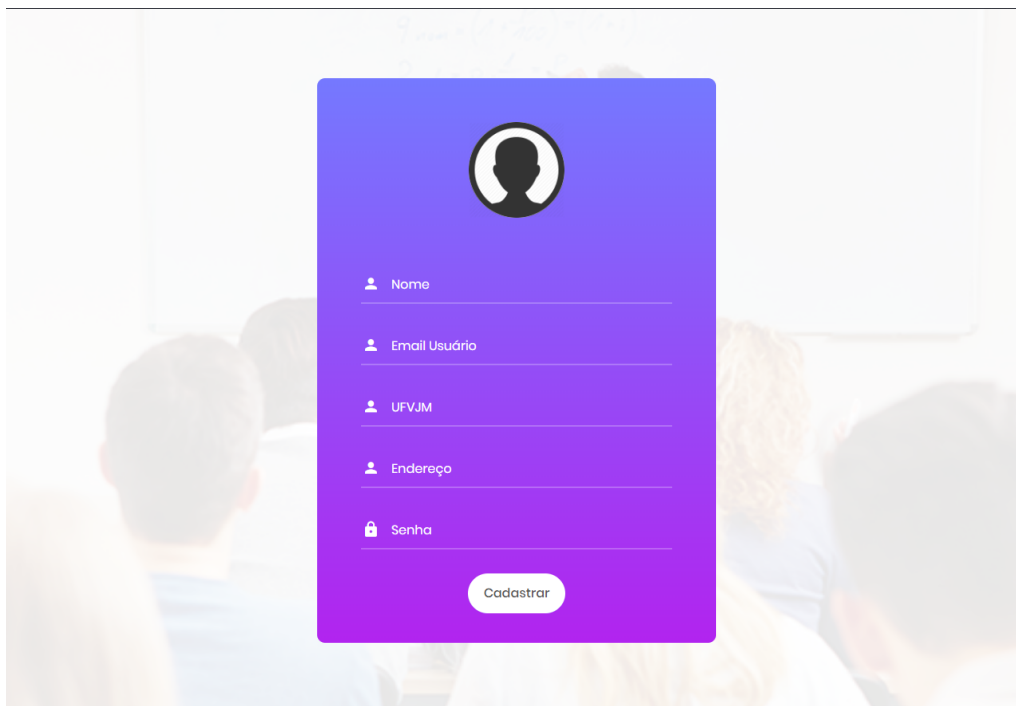
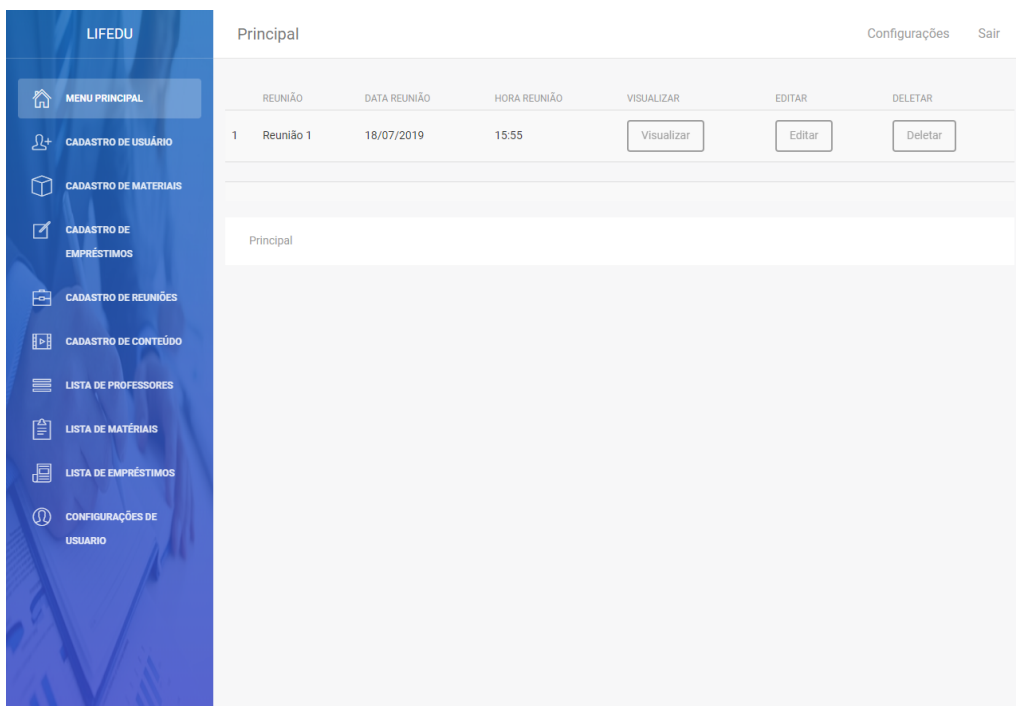


Figura A.1: Tela de Login



A screenshot of a student registration form. The form is a purple-to-blue gradient overlay on a blurred background of a classroom. It features a circular profile icon at the top. Below it are five input fields, each with a user icon: 'Nome', 'Email Usuário', 'UFVJM', 'Endereço', and 'Senha'. A 'Cadastrar' button is positioned at the bottom center of the form.

Figura A.2: Tela de cadastro de alunos



A screenshot of the administrator's main interface. On the left is a blue sidebar menu with the title 'LIFEDU' and various options like 'MENU PRINCIPAL', 'CADASTRO DE USUÁRIO', 'CADASTRO DE MATERIAIS', etc. The main content area is titled 'Principal' and shows a table of meetings. At the top right of the main area are links for 'Configurações' and 'Sair'.

REUNIÃO	DATA REUNIÃO	HORA REUNIÃO	VISUALIZAR	EDITAR	DELETAR	
1	Reunião 1	18/07/2019	15:55	<input type="button" value="Visualizar"/>	<input type="button" value="Editar"/>	<input type="button" value="Deletar"/>

Figura A.3: Tela Principal de Administrador

LIFEDU

Cadastrar Usuário

Configurações Sair

Cadastro de Usuário

UNIVERSIDADE	EMAIL	SENHA
UFVJM	phtixa@gmail.com	pedro123
NOME	SOBRENOME	TIPO DE USUÁRIO
Pedro Henrique	Campos	Professor
RUA	BAIRRO	N°
Company		pedro123
CIDADE	ESTADO	CEP
Diamantina	MG	39100-000

Cadastrar

Principal

Figura A.4: Tela de cadastro de usuário do tipo administrador e professor(Administrador)

LIFEDU

Cadastrar material

Configurações Sair

Cadastro de Material

SETOR	NOME DO MATERIAL	MARCA DO MATERIAL
LIFE		
UNIDADES		

Cadastrar

Principal

Figura A.5: Tela de cadastro de materiais(Administrador)

LIFEDU Cadastrar Empréstimo Configurações Sair

Cadastro de Empréstimos de Materiais

SETOR RESPONSÁVEL PELO EMPRÉSTIMO: LIFE

MATERIAL PARA EMPRÉSTIMO: Data Show

RESPONSÁVEL PELO EMPRÉSTIMO: Pedro Henrique

DATA DO EMPRÉSTIMO: dd/mm/aaaa

SETOR QUE IRÁ ADQUIRIR O MATERIAL EMPRESTADO: Company

RESPONSÁVEL PELA SOLICITAÇÃO DO EMPRÉSTIMO:

Cadastrar

Principal

Figura A.6: Tela de cadastro de empréstimos(Administrador)

LIFEDU Cadastrar Reunião Configurações Sair

Cadastro de Reuniões

NOME REUNIÃO:

SETOR DA REUNIÃO:

DATA DA REUNIÃO: dd/mm/aaaa

HORA DA REUNIÃO: :-

RESUMO DA REUNIÃO (PRINCIPAIS TÓPICOS):

Cadastrar

Principal

Figura A.7: Tela de cadastro de reuniões(Administrador)

Figura A.8: Tela de cadastro de vídeos(Administrador)

	NOME	DISCIPLINA	UNIVERSIDADE	VISUALIZAR
1	Pedro henrique	Cálculo I	UFVJM	Visualizar

Figura A.9: Tela de listagem de professores cadastrados(Administrador)

Lista de Materiais

Configurações Sair

Lista de Materiais Cadastrados no Sistema

	NOME	MARCA	SETOR	VISUALIZAR	EDITAR	DELETAR
1	Data Show	EPSON	LIFE	Visualizar	Editar	Deletar

Principal

Figura A.10: Tela de listagem de materiais cadastrados(Administrador)

Lista de Empréstimos

Configurações Sair

Lista de empréstimos

	MATERIAL EMPRESTADO	RESPONSAVEL PELO EMPRESTIMO	SETOR	VISUALIZAR	EDITAR	DELETAR
1	Data Show	Pedro henrique Campos	LIFE	Visualizar	Editar	Deletar

Principal

Figura A.11: Tela de listagem de empréstimos cadastrados(Administrador)

The screenshot displays the 'Editar Perfil' (Edit Profile) page in the LIFEDU system. On the left is a blue sidebar menu with the following items: MENU PRINCIPAL, CADASTRO DE USUÁRIO, CADASTRO DE MATERIAIS, CADASTRO DE EMPRÉSTIMOS, CADASTRO DE REUNIÕES, CADASTRO DE CONTEÚDO, LISTA DE PROFESSORES, LISTA DE MATERIAIS, LISTA DE EMPRÉSTIMOS, and CONFIGURAÇÕES DE USUÁRIO. The main content area is titled 'Editar Perfil' and contains the following form fields: UNIVERSIDADE (UFVJM), EMAIL (phtixa@gmail.com), NOME (Pedro Henrique), SOBRENOME (Campos), ENDEREÇO (Rua Vereador Gustavo botelho), Nº (empty), CIDADE (Diamantina), ESTADO (MG), and CEP (39100-000). A blue 'Atualizar Perfil' button is located at the bottom right of the form. The top right corner of the page has links for 'Configurações' and 'Sair'. The bottom left corner of the page has a 'Principal' link.

Figura A.12: Tela de configurações de usuário(Administrador)

This screenshot is identical to the one above, showing the 'Editar Perfil' page in the LIFEDU system. It features the same sidebar menu, form fields (UNIVERSIDADE: UFVJM, EMAIL: phtixa@gmail.com, NOME: Pedro Henrique, SOBRENOME: Campos, ENDEREÇO: Rua Vereador Gustavo botelho, Nº, CIDADE: Diamantina, ESTADO: MG, CEP: 39100-000), and the 'Atualizar Perfil' button. The page layout, including the 'Configurações' and 'Sair' links at the top right and the 'Principal' link at the bottom left, is consistent with the previous image.

Figura A.13: Tela de configurações de usuário(Administrador)

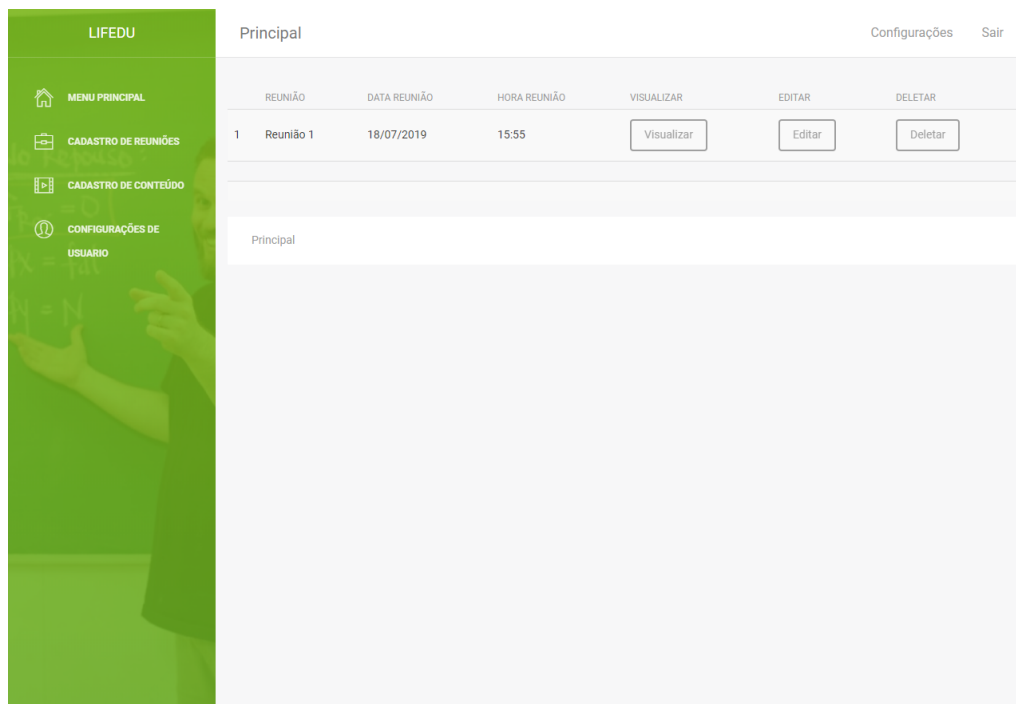


Figura A.14: Tela de principal(Professor)

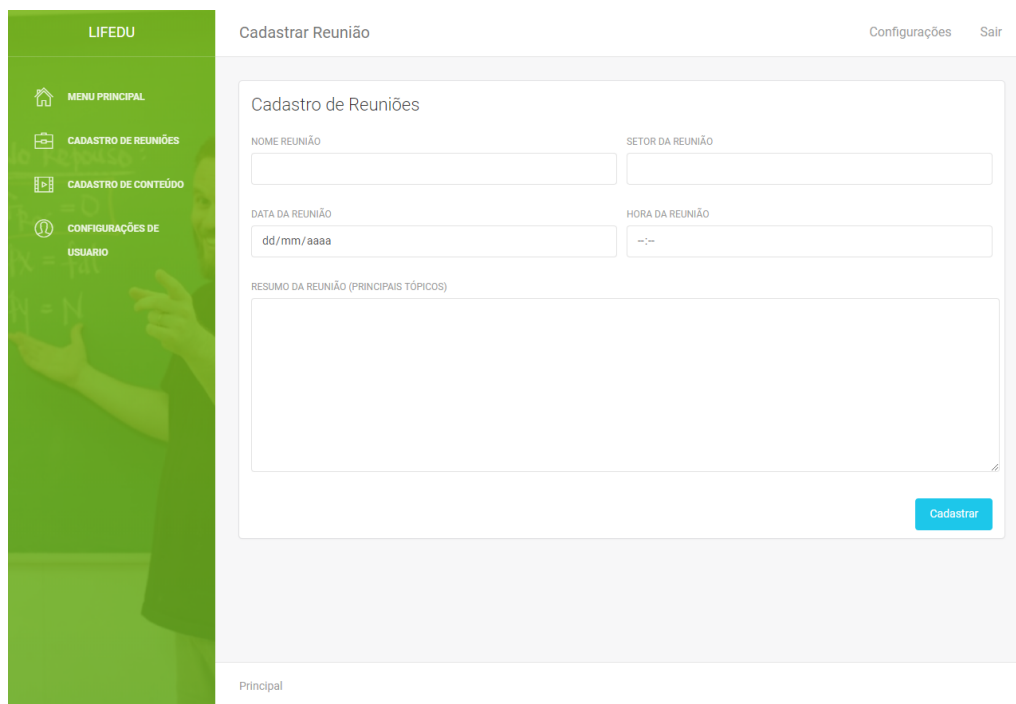


Figura A.15: Tela de cadastro reuniões (Professor)

LIFEDU Cadastrar Vídeos Configurações Sair

Cadastro de Vídeos

TÍTULO DO VÍDEO

DATA DA POSTAGEM

PROFESSOR RESPONSÁVEL

DESCRIÇÃO

ARQUIVO Nenhum arquivo selecionado

Principal

Figura A.16: Tela de cadastro de conteúdo (Professor)

LIFEDU Cadastrar Reunião Configurações Sair

Cadastro de Reuniões

NOME REUNIÃO

SETOR DA REUNIÃO

DATA DA REUNIÃO

HORA DA REUNIÃO

RESUMO DA REUNIÃO (PRINCIPAIS TÓPICOS)

Principal

Figura A.17: Tela de configurações (Professor)

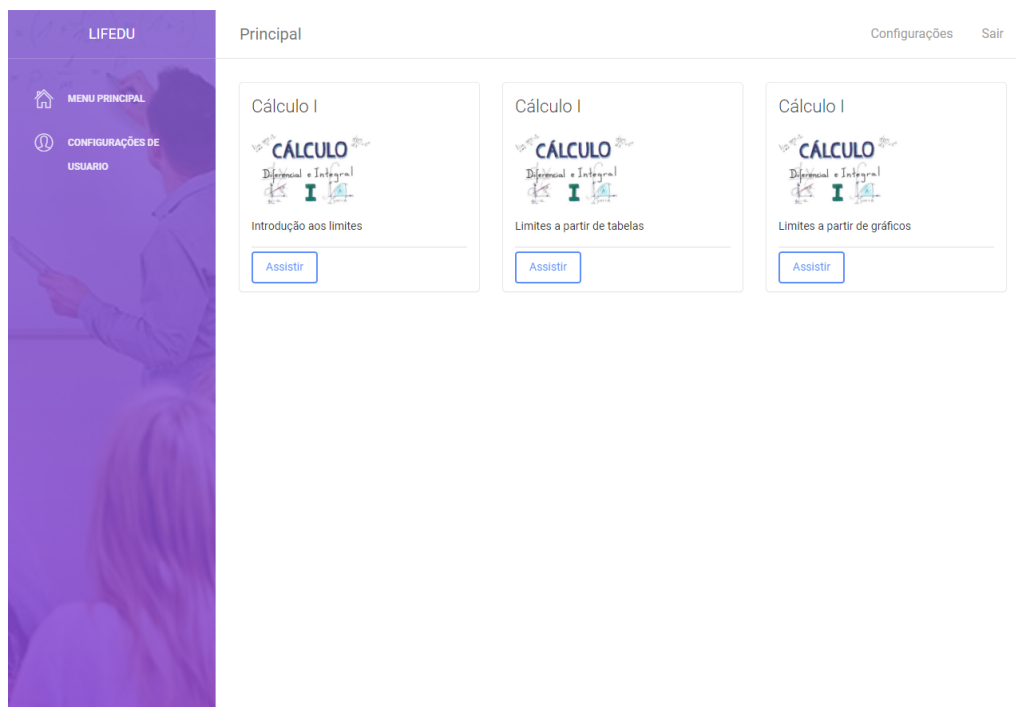


Figura A.18: Tela Principal (Aluno)

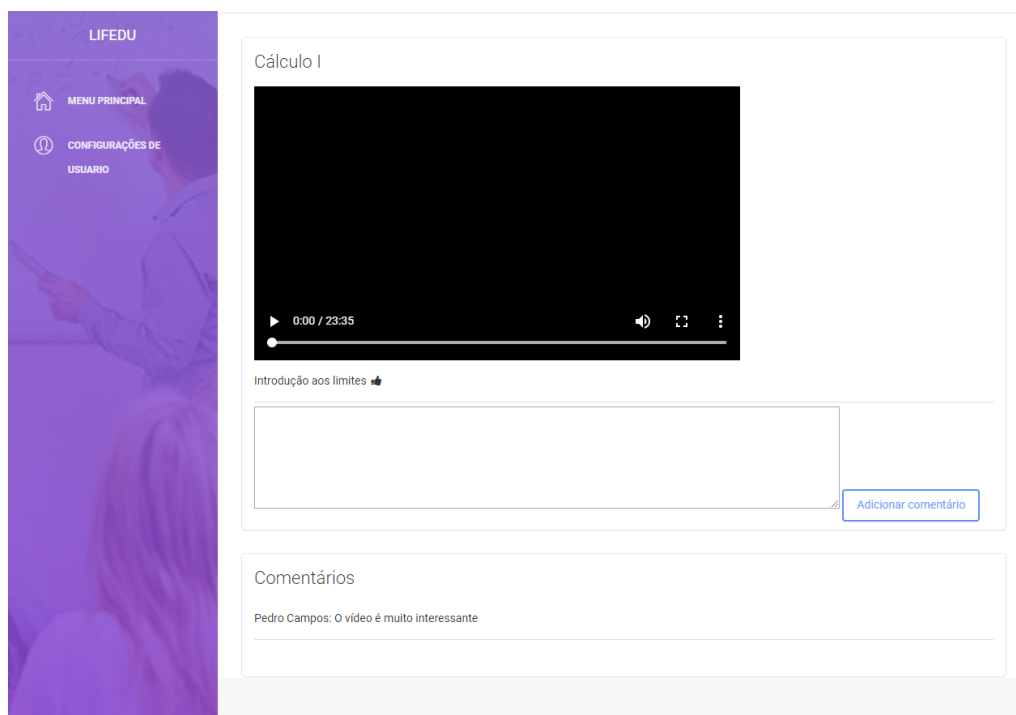


Figura A.19: Tela Principal (Aluno)