

SÃO LUCAS
J I - P A R A N Á • R O



DAVID GUTTENBERG MEDEIROS DOS SANTOS

LUIZ HENRIQUE DE OLIVEIRA MOURA

**DESENVOLVIMENTO DE UM WEBSITE: FIXAR - BIBLIOTECA DE
CURSOS ONLINE**

JI-PARANÁ

2021

DAVID GUTTENBERG MEDEIRO DOS SANTOS

LUIZ HENRIQUE DE OLIVEIRA MOURA

**DESENVOLVIMENTO DE UM WEBSITE: FIXAR - BIBLIOTECA DE CURSOS
ONLINE**

Monografia apresentada a Banca Examinadora do Centro Universitário São Lucas Ji-Paraná, como requisito de aprovação para obtenção do Título de Bacharel em Sistema de Informação.

Orientador: Prof. Esp. José Rodolfo Milazzotto Olivas

Coorientador: Prof. Esp. Pedro Vinicius Rodrigues Dias

JI-PARANÁ

2021

Dados Internacionais de Catalogação na Publicação - CIP

S237d Santos, David Guttenberg Medeiros dos.

Desenvolvimento de um Website: Fixar - Biblioteca de Cursos Online. / David Guttenberg Medeiros dos Santos; Luiz Henrique de Oliveira Moura. – Ji-Paraná, 2021.
61 p.; il.

Monografia (Curso de Sistemas de Informação) – Centro Universitário São Lucas Ji-Paraná, 2021.

Orientador: Prof. Esp. José Rodolfo Milazzotto Olivas

1. Desenvolvimento. 2. Aplicação Web. 3. Biblioteca de Cursos Online. 4. Software de aplicação - desenvolvimento. I. Moura, Luiz Henrique de Oliveira. II. Olivas, José Rodolfo Milazzotto. III. Título.

CDU 004.41

RESUMO

O presente trabalho especifica e descreve a criação do Software Fixar Biblioteca de Cursos Online um Software voltado para a plataforma Web. Essa ferramenta busca facilitar a busca por cursos de seu interesse de modo mais simples e intuitivo. O projeto será desenvolvido utilizando a metodologia de desenvolvimento cascata, e para a programação foi utilizada a linguagem Javascript em conjunto ao *Software Visual Code* da Microsoft. Também utilizaremos Node.js para execução de códigos no *back-end* e o Yarn, que nos permite gerenciar bibliotecas Node.js, onde podemos adicionar bibliotecas ao nosso projeto como express.js que facilita a construção da aplicação. ReactJS será usado para criar interfaces de usuário, e como base de dados será utilizado o banco de dados PostgreSQL. Em sua base teórica apresentada foi utilizada a modelagem do sistema por meio de diagramas UML, nos quais foram descritos os casos de uso, especificados os requisitos funcionais e, finalmente, foi criado o diagrama de classes do sistema.

Palavras Chaves: Desenvolvimento, Aplicação Web, Biblioteca de Cursos Online.

ABSTRACT

The present work specifies and describes the creation of the Software Fixar online course library, a Software aimed at the Web platform. This tool seeks to facilitate the search for courses of interest in a simpler and more intuitive way. The project will be developed using the cascade development methodology, and for programming the Javascript language was used together with Microsoft's Visual Code Software. We will also use Node.js for code execution on the backend and Yarn, which allows us to manage Node.js libraries, where we can add libraries to our project like express.js which makes building the application easier. ReactJS will be used to create user interfaces, and the PostgreSQL database will be used as a database. In its theoretical basis presented, the modeling of the system through UML diagrams was used, in which the use cases were described, the functional requirements specified and, finally, the system class diagram was created.

Keywords: Development, Web Application, Online Course Libra

LISTA DE TABELAS

Tabela 1 – Ferramentas / Tecnologias	43
--	----

LISTA DE ILUSTRAÇÕES

Figura 1 - Sistema Operacional Cheetah Apple.....	16
Figura 2 - MacOS X BigSur	16
Figura 3 - Sistema Operacional Ubuntu.....	17
Figura 4 - Janela do Visual Studio Code.....	22
Figura 5 – Interface Whimsical	23
Figura 6 - Interface Diagramas.net	25
Figura 7 – Interface Insomnia REST.....	26
Figura 8 - Diagrama de Casos de Uso	27
Figura 9 - Metodologia de desenvolvimento Cascata	28
Figura 10 – Diagrama de caso de uso.....	31
Figura 11 – Diagrama de classe.....	42
Figura 12 – Terminal com comandos.....	44
Figura 13 – Terminal com comandos.....	45
Figura 14 – Tela de login do site.....	50
Figura 15 – Tela de login com exceções	51
Figura 16 – Tela de cadastro de usuário	51
Figura 17 – Tela de cadastro de usuário com exceções.....	52
Figura 18 – Tela de cursos	52
Figura 19 – Tela de cursos do administrador.....	53
Figura 20 – Tela de curso.....	53
Figura 21 – Tela de avaliações.....	54
Figura 22 – Tela de adicionar cursos.....	54
Figura 23 – Tela de cursos pendentes	55
Figura 24 – Tela de aprovar cursos	55
Figura 25 – Tela de editar perfil.....	56
Figura 26 – Tela de informações de privacidade	56
Figura 27 – Tela de suporte.....	57

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas.
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de HiperTexto).
WWW	World Wide Web (Rede de Alcance Mundial).
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)
ES8	ECMAScript 2018.
API	Application Programming Interface (Interface de Programação de Aplicações).
REST	Representational State Transfer (Transferência Representacional de Estado).
SPA	Single-page application (Aplicativo de Página Única).
CSS	Cascading Style Sheets (Folha de Estilos em Cascata).
CRUD	Create Read Update Delete (Criar Ler Atualizar Excluir).
JSON	JavaScript Object Notation (Notação de Objeto Javascript).
REA	Recursos Educacionais Abertos
NPM	Node Package Manager (Gerenciador de Pacotes Node)
LTS	Long-term support (Suporte de longo prazo)

SUMÁRIO

1. INTRODUÇÃO	12
1.1. JUSTIFICATIVA.....	13
1.2. PROBLEMATIZAÇÃO	13
1.3. OBJETIVOS	14
1.3.1. Objetivo Geral	14
1.3.2. Objetivos Específicos	14
2. REFERENCIAL TEÓRICO	14
2.1.1. Windows.....	15
2.2.2. Mac Os.....	15
2.3.3. Ubuntu.....	17
2.2. NAVEGADORES DE INTERNET	18
2.2.1. Google Chrome.....	18
2.2.2. Mozilla Firefox	18
2.2.3. Safari.....	19
2.3. DESENVOLVIMENTO DE APLICAÇÕES WEB	19
2.4. FERRAMENTAS DE DESENVOLVIMENTO	19
2.4.1. Javascript	19
2.4.2. ReactJS.....	20
2.4.3. HTML5	20
2.4.4. Visual Studio Code.....	21
2.4.5. PostgreSQL.....	22
2.4.6. CSS.....	23
2.4.7. Whimsical.....	23
2.4.8. Yarn.....	24
2.4.9. Express.js.....	24

2.4.10. GitHub	24
2.4.11. Diagramas.net	24
2.4.12. Insomnia REST	25
2.4.13. Node.js	26
2.5. UML	26
2.6. METODOLOGIA DE DESENVOLVIMENTO CASCATA.....	28
2.7. EDUCAÇÃO ABERTA	28
2.8. A IMPORTANCIA DOS CURSOS ONLINE	29
2.9 NOVAS POSSIBILIDADES DE APRENDIZAGEM	29
3. MATERIAIS E MÉTODOS.....	30
3.1. FASES E ESTUDOS	30
3.1.1. Modelagem	30
3.1.2. Programação.....	30
3.2. CASOS DE USO.....	30
3.3. DOCUMENTAÇÃO DO CASO DE USO	31
3.4. REQUISITOS FUNCIONAIS.....	38
3.5. REQUISITOS NÃO FUNCIONAIS	38
3.6. DIAGRAMA DE CLASSES	41
5. RECURSOS.....	43
6. Desenvolvimento.....	44
6.1 Configuração do projeto.....	44
6.2.1. Instalação do NODEJS	44
6.2.2. Instalação do Yarn	44
6.2.3. Instalação do PostgreSQL	45
6.2.4. Instalação do Visual Studio Code.....	45
7.3 Padrão de comunicação	46
6.3.1. Serviço de usuários.....	46

6.3.2. Serviço de categorias.....	47
6.3.3. Serviço de plataformas.....	47
6.3.4. Serviço de professores.....	48
6.3.5. Serviço de cursos.....	48
6.3.6. Serviço de avaliação	49
6.4 Funcionalidades.....	50
7. CONSIDERAÇÕES FINAIS	58
REFERÊNCIAS.....	59

1. INTRODUÇÃO

Os seres humanos vêm evoluindo, e com o passar dos anos essa exponencial de crescimento vem trazendo uma chuva informações, onde, por vezes desordenadas. Sendo assim, a troca de informações virou algo crucial para o desenvolvimento humano, tanto no âmbito empresarial quanto pessoal, e percebe-se, que as pessoas sempre estão em busca criam novas formas de se comunicar e transmitir conhecimento.

A transmissão de conhecimento é um fator chave para a sociedade moderna, e por sua vez são difundidas por veículos de comunicação, sendo eles desde livros, panfletos, áudio-books ou até mesmo por cursos disponíveis na rede mundial de computadores.

Esses cursos representam uma nova abordagem do conhecimento e exige uma preparação diferenciada para os três atores do processo de ensino e aprendizagem: instituições de ensino, localizadas no virtual; professores, colocados como colaboradores e capacitados a desenvolver trabalhos no ambiente digital; e alunos de uma geração digital que ainda necessitam saber como desenvolver a aprendizagem independente e superar a superficialidade com que a velocidade das transformações os faz encarar as coisas (MUNHOZ, 2016, p. 10).

Dessa forma, a organização de conteúdo em rede para os estudos é uma necessidade crescente nos últimos anos, devido ao grande número dos mais variados tipos de cursos dos mais diversos assuntos que, diariamente, são incluídos na rede.

Diante deste cenário, pela busca da otimização do conhecimento o presente estudo procura desenvolver um WebSite, para catalogação e disponibilização de cursos online, utilizando as tecnologias JavaScript com o back-and NodeJS e front-end em ReactJS. O WebSite de busca de Cursos Online irá proporcionar um sistema para os administradores e para os usuários comuns, irão proporcionar campos para cadastro e disponibilização dos cursos cadastrados na plataforma. O WebSite irá funcionar tanto em computadores desktop como dispositivos móveis desde que tenha um navegador previamente instalado.

1.1. JUSTIFICATIVA

No cenário atual da pandemia da Covid-19, a educação a distância passou a ser a única alternativa para que aulas não fossem cessadas durante o período de quarentena. Estes eventos vivenciados nos mostram que a educação sofreu e está sofrendo uma grande mudança desde a dinâmica a rotina escolar de maneira abrupta.

Sendo assim houve um crescimento na disponibilização de cursos gratuitos em oferta na internet, disponibilizados por diversos sites e plataformas de curso. Os alunos pesquisam pelo curso desejado e encontram diversos resultados na rede, por muitas vezes esses resultados encontrados podem estar embaralhados ou confusos para o entendimento.

Sendo assim para que o mesmo encontre o curso mais adequado é necessário acessar muitos sites e verificar as especificações, os requisitos cada curso, se possuem ou não avaliações de usuários sobre o tema a ser escolhido e suas devidas justificativas.

Assim, com toda procura, por muitas vezes os usuários acabam desistindo de escolher algum curso pela dificuldade de escolha por existir uma grande gama de oferta dos mesmos. Dessa maneira o intuito do sistema Fixar Biblioteca de Cursos Online é facilitar a escolha de cursos online com auxílio de avaliações dos usuários por meio de uma plataforma simples e intuitiva.

Neste trabalho de pesquisa, serão discutidos conceitos de vão desde a configuração do ambiente para o desenvolvimento partindo da instalação e configuração de aplicações até o desenvolvimento. Pretende-se desenvolver uma aplicação utilizando a plataforma Web, para criar uma nova experiência na procurar de cursos disponíveis na internet.

1.2. PROBLEMATIZAÇÃO

Com a situação pandêmica da Covid-19 instaurada desde o ano de 2020, os serviços online tiveram um grande crescimento, além disso os decretos governamentais forçaram as pessoas a permanecerem em suas casas. Com esse

cenário a possibilidade de cursos via internet foi uma forma encontrada para que as pessoas se mantenham informadas e atualizadas. Contudo há uma dificuldade em encontrar um portal que disponibiliza de maneira condensada informações sobre cursos de curta e média duração, bem como uma boa visualização de áreas, precificação e certificações. De que maneira poderá se automatizar esse processo de consulta visando atender um público que cada vez mais cresce?

1.3. OBJETIVOS

1.3.1. Objetivo Geral

Desenvolver um WebSite que visa ser uma plataforma buscadora de cursos online.

1.3.2. Objetivos Específicos

- Desenvolver uma biblioteca de cursos online na linguagem Javascript.
- Esquematizar de forma que o site possa dar feedback às pessoas.
- Construir o back-end do WebSite em NodeJS e o front-end em ReactJS.

2. REFERENCIAL TEÓRICO

2.1. SISTEMAS OPERACIONAIS

Para Silberschatz (1998), o sistema operacional é um programa computacional que possui o papel de intermediador entre o usuário e o hardware do computador. O objetivo do sistema operacional é executar outros softwares, que ajudam os usuários a resolverem problemas comuns do seu dia-a-dia de maneira simples e fácil, em si fazer de uma maneira que o computador seja conveniente na utilização de maneira eficiente.

Segundo Stallings (2012), um sistema operacional (OS) aproveita-se dos recursos de hardware para prover aos usuários um conjunto de serviços para a sua

utilização. O Sistema Operacional realiza todo o gerenciamento do computador, desde o gerenciamento das memórias, periféricos sendo I/O (Input – Entrada e Out – Saída), tais quanto os dispositivos de armazenamento, mouse, teclado, vídeo do sistema.

2.1.1. Windows

O Windows foi criado em 1985 por Bill Gates, também criador da empresa Microsoft que hoje é uma das maiores do mundo. O Windows teve seu propósito inicial pensando que os computadores deixassem de ser apenas do mundo universitário e corporativo para o uso de pessoas comuns.

A Microsoft já lançou várias versões do seu SO. Em 2001 a Microsoft lançou o seu primeiro sistema que popularizou da marca o Windows XP, onde em 2009 veio com o lançamento Windows 7 trazendo com uma nova roupagem visual agradável e com ferramentas otimizadas onde ganhou ainda uma atenção do público.

Com o intuito de um maior controle de sua marca e a melhoria contínua de suas aplicações, em 2005 a Microsoft tomou a decisão de lançar uma versão final do seu sistema operacional para os consumidores de seus sistemas o Windows 10, onde mais tarde viria se chamar somente Windows. Este sistema vem com intuito de reunir as melhores experiências da empresa e empregarem em um local só, atendendo desde empresas a consumidores finais.

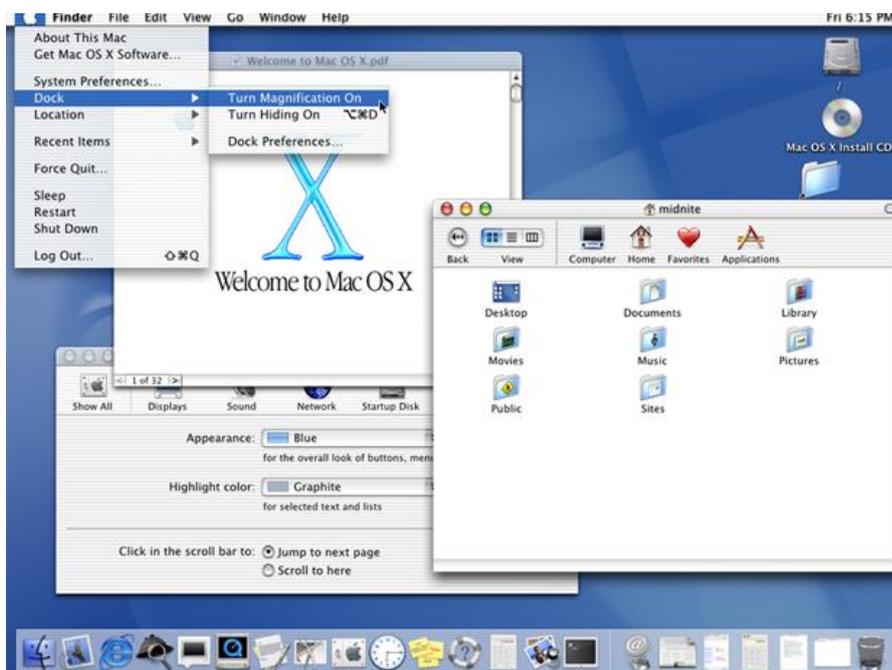
2.2.2. Mac Os

Mac se deu início no ano de 1984 junto ao primeiro Macintosh publicados pela empresa Apple inc. seu sistema operacional era conhecido como *system*.

A partir do *system* a Apple trabalhou em várias versões de seu sistema o decorrer dos anos. Em 1999 houve o lançamento do Mac Os X, em 2001 houve uma nova atuação, o nascimento do sistema “Cheetah” e em 2016 e o nome foi alterado para MacOS para simplificar a marca das demais marcas da empresa iOS, watchOS, tvOS e o MacOS.

Atualmente a última versão do MacOS conhecida como “MacOS 11.0 “Big Sur” lançada em 2020.

Figura 1 - Sistema Operacional Cheetah Apple



Fonte: Techtudo, 2021

Na figura 1, podemos ver a representação do sistema operacional MacOS X Cheetah.

Figura 2 - MacOS X BigSur



Fonte: Apple, 2021

A Figura 2 faz referência ao sistema MacOS BigSur de 2020, para demonstrar a evolução feita pelo sistema operacional MacOS.

2.3.3. Ubuntu

Desde 1984, um projeto chamado de GNU criado por Richard Stallman tinha como meta o desenvolvimento de um sistema operacional livre baseado no Unix. O projeto então criou uma licença de software chamada de GPL. Ela permitia a modificação livre do código de um programa, desde que distribuído posteriormente desta mesma forma e mantida os créditos dos desenvolvedores. Ao longo de cinco anos o projeto já tinha criado a maior parte dos programas essenciais para um sistema operacional mas faltava um Kernel livre (4.LINUX,2021).

O Linux desenvolvido por Linux Torvalds em 1991, usando kernel Linux que é o núcleo de todo sistema operacional, decidiu distribuir livremente o código fonte para que qualquer pessoa possa utilizar estudar modificar, com esse Fato foram surgindo vários modelos de Linux personalizados o mais famoso deles hoje é o Ubuntu da empresa Canonical.

Figura 3 - Sistema Operacional Ubuntu



Fonte: Próprio Autor

2.2. NAVEGADORES DE INTERNET

Segundo MOZILLA FOUNDATION (2021), um navegador web te leva para qualquer lugar na internet. Ele obtém informações de outras partes da web e as exibe em seu computador ou dispositivo móvel. As informações são transferidas usando o Protocolo de Transferência de Hipertexto, que define como textos, imagens e vídeos são transmitidos na web.

2.2.1. Google Chrome

Google Chrome é um Navegador teve seu lançamento pela Google de sua versão web para Microsoft em 2008, em 2010 teve o lançamento para os Sistemas operacionais Linux e Mac OS X e em 2012 a google lança a versão para Android e iOS.

Atualmente Google Chrome segue como sendo o Navegador mais popular do mundo, possui uma vasta loja de extensões que somam muito a experiencia do usuário, tendo também um dos melhores sistema de tradução para navegador onde a página inteira é traduzida de forma simples e intuitiva.

Um domínio tão amplo quanto o do Chrome é perigoso porque pode levar a um cenário em que determinadas aplicações só rodam bem nesse navegador ou permitir que o Google colete uma infinidade de dados sobre nossos hábitos de navegação (ALECRIN, 2018).

2.2.2. Mozilla Firefox

O navegador Firefox teve seu lançamento em 2004 para Windows pela Mozilla Foundation foi uma plataforma desenvolvida de forma onde centenas de colaboradores ajudaram em seu desenvolvimento, com a intenção de criar um navegador leve, seguro e intuitivo, em 2008 teve seu lançamento para Mac OS X e em 2020 sua versão para Android, segundo MOZILLA FOUNDATION (2021) no primeiro ano, novos membros da comunidade de todo o mundo já tinha contribuído com novas funcionalidades, aprimorando recursos existentes e se envolvendo na gestão e planejamento do projeto.

2.2.3. Safari

Safari é um navegador desenvolvido pela Apple inc. teve seu início em 2003 e ele foi feito para ser o navegador padrão da Apple inc. podendo ser usado em seus respectivos produtos seja em Mac OS ou em iOS, foi desenvolvido uma versão para Windows em 2006, mas a mesma foi descontinuada em 2012 e não possui mais o suporte da Apple inc.

2.3. DESENVOLVIMENTO DE APLICAÇÕES WEB

O Desenvolvimento de sites é o conjunto de atividades necessárias para se criar um website, contempla desde a criação de *layout(design* de interface), programação de interface (codificação html), construção de arquivos de estilo css(estilização de conteúdo) até programação de sistema e programação de interações *client side* (javascript). Existem varias tecnologias através das quais você pode desenvolver um website, não importante se estamos falando de um Site Institucional ou de uma Plataforma de *Ecommerce* (Agência KAIZEN).

Segundo Agência KAIZEN, Para Desenvolver um site é conveniente passar por todas as fazes da Criação do Site até chegar na parte técnica de programação, as fases são basicamente divididas em:

- Direção de Arte de um Site: onde se cria o conceito visual do site
- Arquitetura de Informação: onde se cria a estrutura do conteúdo do site
- Design de Interface: onde se aplica a direção de arte em todas as telas criadas pela arquitetura de informação
- Programação de Interface: onde se tranforma as telas em codigo navegavel html
- Programação de Sistema: onde se transforma o site e uma ferramenta com interação com o servidor de aplicação e banco de dados

2.4. FERRAMENTAS DE DESENVOLVIMENTO

2.4.1. Javascript

JavaScript é a linguagem de programação da Web. A ampla maioria dos sites modernos usam JavaScript e todos os navegadores modernos em computadores de mesa, consoles de jogos, tablets e smartphones - incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas (FLANAGAN, 2013).

Se você já conhece outras linguagens de programação, talvez ajude a saber que JavaScript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais. A sintaxe de JavaScript é derivada da linguagem Java, das funções de primeira classe de *Scheme* e da herança baseada em protótipos de Self (FLANAGAN, 2013).

2.4.2. ReactJS

ReactJS é uma biblioteca JavaScript, do Facebook, que evoluiu originalmente de *framework* chamado BoltJS e FaxJS (KOPPALA, 2018). A biblioteca tem seu código aberto, voltada para aplicações *web* e *sites*, que tem como objetivo superar desafios encontrados no desenvolvimento de aplicativos de uma só página (SPA), com códigos flexíveis e eficientes (VIPUL; SONPATKI, 2016).

ReactJS é orientado a componentes. Estes podem ser encapsulados, e também gerenciam o próprio estado (REACT, 2019). Não é um framework, mas pode ser combinado com alguns, como AngularJS, Ember e Meteor (ROBBESTAD, 2016).

2.4.3. HTML5

Segundo Mazza (2012), o HTML5 é uma evolução do HTML (Hypertext Markup Language) que está em sua 5^o versão. O HTML5 introduziu diversos elementos, atributos e funcionalidades propiciando aos desenvolvedores criar melhores experiências para os usuários e simplificações para o desenvolvimento de aplicações para a World Wide Web.

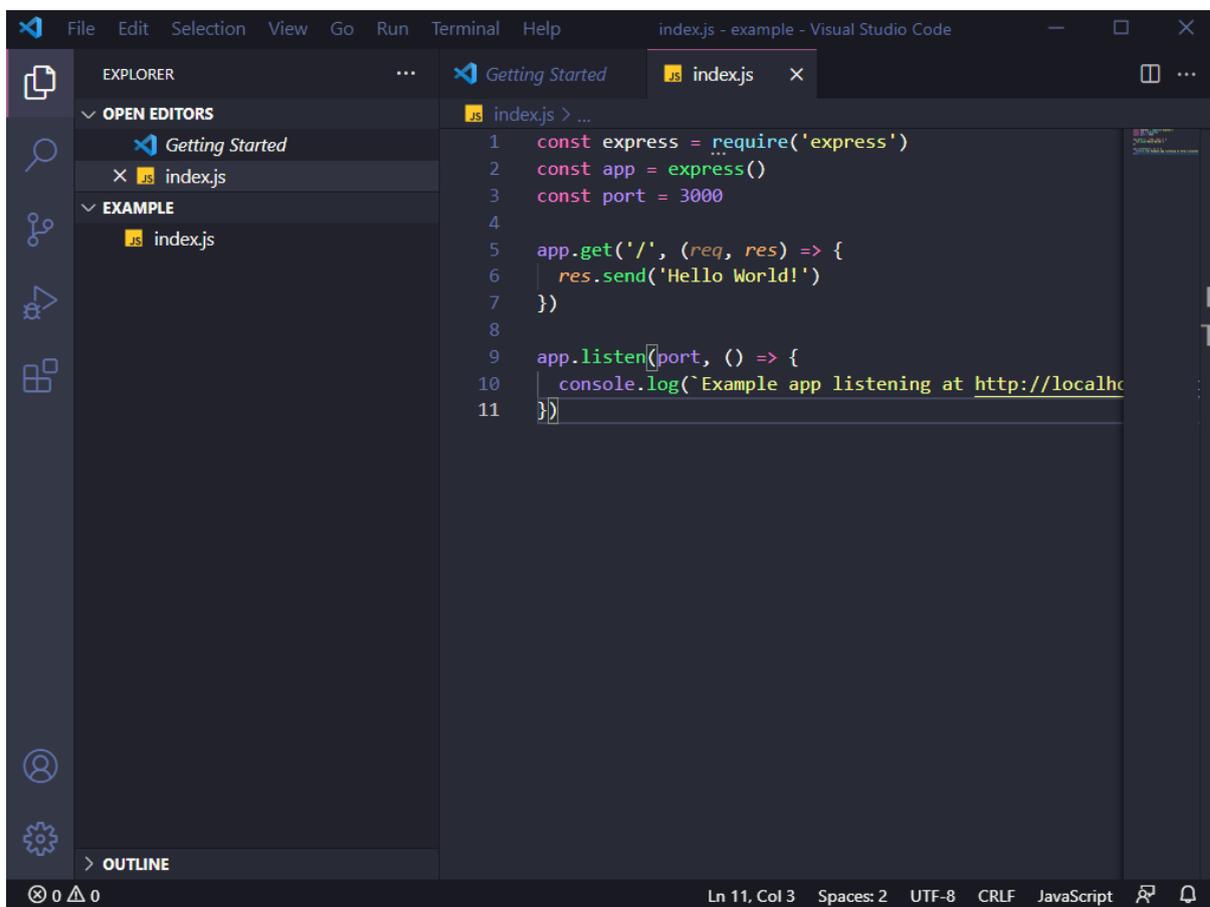
Segundo a Mozilla Foundation (2016), o HTML5 ou HTML5 & *friends* é um conjunto de recursos que possibilita aos desenvolvedores WEB criar novas experiências e sites mais robustos. Os novos recursos são:

- Semântica: permite descrever mais precisamente o seu conteúdo.
- Conectividade: permite uma comunicação com o servidor de formas modernas e inovadoras.
- Off-line e armazenamento: Permite que páginas web armazenem dados localmente do lado do cliente e operem de forma off-line mais eficientemente.
- Multimídia: Viabiliza a utilização de áudio e vídeo de forma primorosa na Web aberta.
- Gráficos e efeitos 2D/3D: viabiliza um leque diversificado de opções de representação gráfica. 13
- Performance e integração: fornece grande otimização de velocidade e melhor utilização do hardware do computador.
- Acesso ao dispositivo: viabiliza a utilização de diversos métodos e dispositivos de entrada e saída.
- Estilização: permite aos autores a escrita de temas mais sofisticados.

2.4.4. Visual Studio Code

O *Visual Studio Code* (figura4) é um editor de texto leve e poderoso disponível para as plataformas Windows, MacOS e Linux. Por padrão já vem com o suporte para linguagens de 14 programação como Javascript, TypeScript entre outras e possui suporte a extensões possibilitando a integração com outras diversas linguagens e idiomas (MICROSOFT, 2017).

Figura 4 - Janela do Visual Studio Code



Fonte: Próprio Autor

2.4.5. PostgreSQL

A ferramenta conhecida atualmente como PostgreSQL teve origem em um projeto chamado de POSTGRES na Universidade Berkeley, na Califórnia (EUA), em 1986. Uma equipe orientada pelo professor Michael Stonebraker foi designada para criar o modelo e as regras de um novo sistema de armazenamento de dados, com o apoio de diversos órgãos, entre eles o Army Research Office (ARO) e o National Science Foundation (NSF) (MILANI, 2008).

O PostgreSQL é um poderoso sistema gerenciador de banco de dados objeto relacional de código aberto. Por muito tempo, foi discriminado no mundo dos bancos de dados, e o seu recente aumento de popularidade veio de usuários de outros bancos de dados em busca de um sistema com melhores garantias de confiabilidade, melhores recursos de consulta, mais operação previsível, ou simplesmente querendo algo mais fácil de aprender, entender e usar (CARVALHO, 2017).

2.4.6. CSS

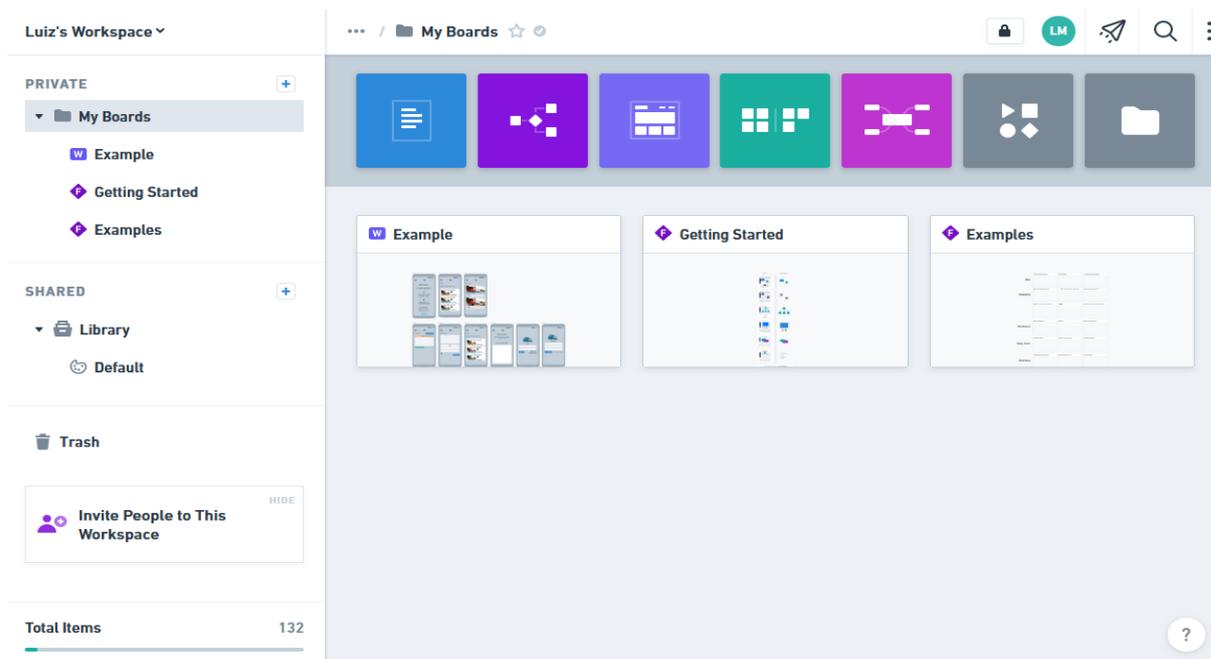
CSS é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante (ARIANE, 2021).

CSS foi desenvolvido pelo W3C (World Wide Web Consortium) em 1996, por uma razão bem simples. O HTML não foi projetado para ter *tags* que ajudariam a formatar a página. Você deveria apenas escrever a marcação para o site (ARIANE, 2021).

2.4.7. Whimsical

Whimsical pode ser utilizada para fazer *Wireframes*, *Flowcharts*, *Mind Maps* (Mapas Mentais), e recentemente tem recursos novos para utilização de *Sticky Notes* que pode ser utilizado para criar *canvas*, diagramas, kanban e muito mais.

Figura 5 – Interface Whimsical



Fonte: Próprio Autor

2.4.8. Yarn

O Yarn é um gerenciador de pacotes que trouxe mais funcionalidades e vantagens para programadores. Com uma estrutura já conhecida e utilizada por ferramentas renomadas, como o NPM, essa aplicação tem se destacado por sua simplicidade e segurança (SOUZA, 2020).

Como é fruto de um trabalho de colaboração, programadores contribuem com pacotes prontos, tornando esse gerenciador cada vez mais enriquecido em variedade de recursos já com códigos devidamente concluídos (SOUZA, 2020).

2.4.9. Express.js

Segundo MOZILLA FOUNDATION (2021), Express é o *framework* Node mais popular e a biblioteca subjacente para uma série de outros *frameworks* do Node. O Express oferece soluções para:

- Gerenciar requisições de diferentes verbos HTTP em diferentes URLs.
- Integrar "*view engines*" para inserir dados nos *templates*.

Definir as configurações comuns da aplicação web, como a porta a ser usada para conexão e a localização dos modelos que são usados para renderizar a resposta.

Adicionar novos processos de requisição por meio de "*middleware*" em qualquer ponto da "fila" de requisições.

2.4.10. GitHub

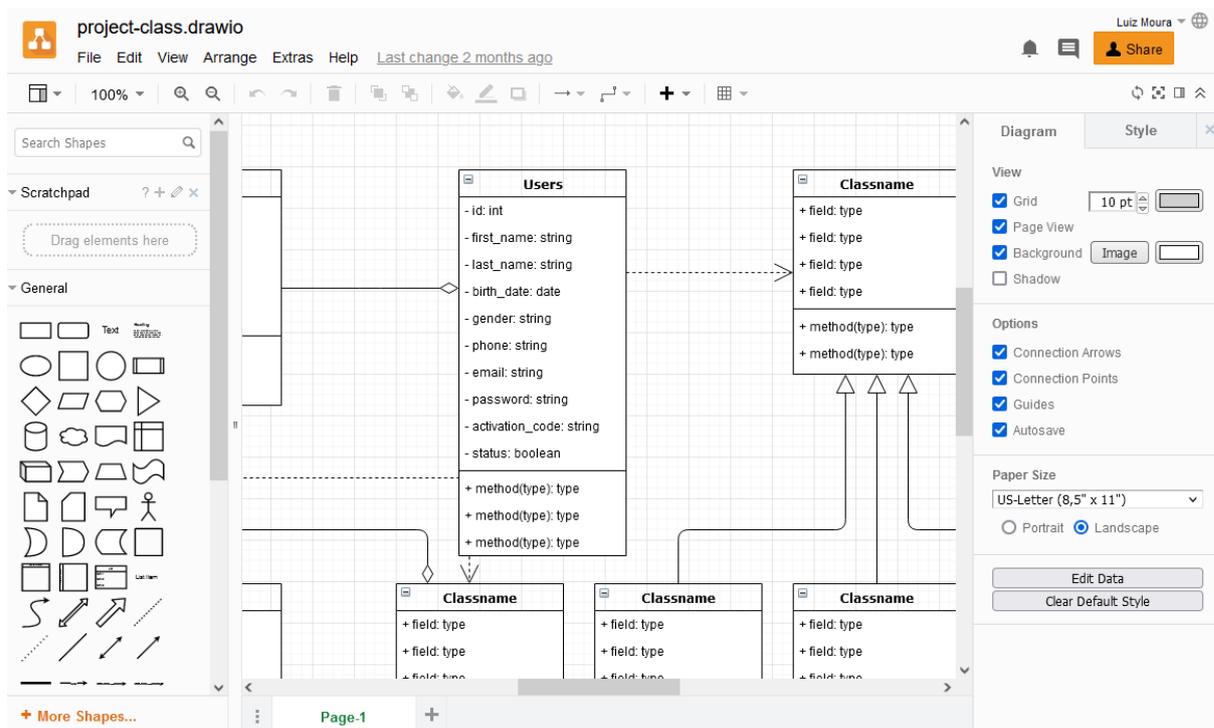
Segundo Augusto (2017), Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.

2.4.11. Diagramas.net

É uma ferramenta de colaboração visual com base em HTML5 que torna a criação de diagramas mais rápida e fácil. Com ela você pode trabalhar junto a um número ilimitado de pessoas para criar e editar diagramas em tempo real, com

alterações incorporadas e sincronizadas instantaneamente, ótimo para trabalhos em equipe e com clientes.

Figura 6 - Interface Diagramas.net



Fonte: Próprio Autor

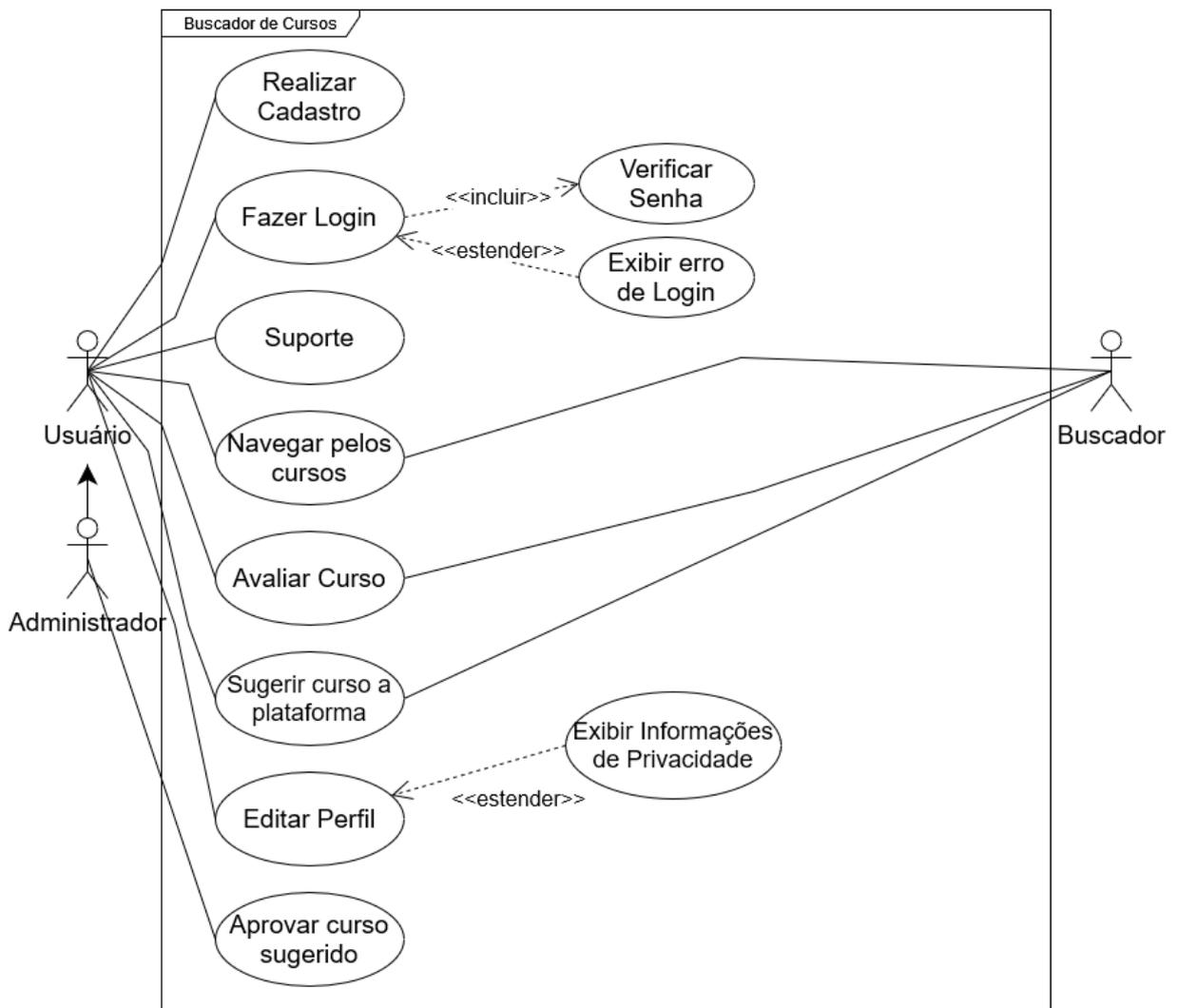
2.4.12. Insomnia REST

Quando estamos desenvolvendo *APIs*, a fase de testes é muito importante. Além do teste propriamente dito também é importante que utilizemos um *REST Client* prático, com boas ferramentas e bem organizado. Insomnia é um *API Client* para aplicações *REST*.

anos, a linguagem-padrão de modelagem adotada internacionalmente pela indústria de engenharia de software (GUEDES, 2011).

O diagrama de casos de uso é o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar (GUEDES, 2011).

Figura 8 - Diagrama de Casos de Uso



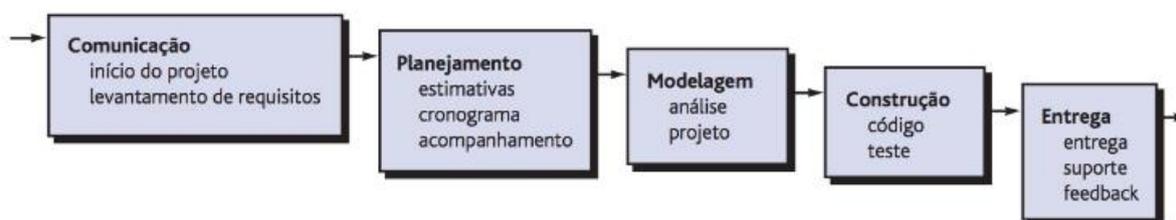
Fonte: Próprio Autor

Como podemos ver na Figura 8, temos o diagrama de casos de Uso referente ao nosso projeto de desenvolvimento.

2.6. METODOLOGIA DE DESENVOLVIMENTO CASCATA

O modelo cascata, algumas vezes chamado ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, começando com a especificação dos requisitos do cliente, avançando pelas fases de planejamento, modelagem, construção e disponibilização, e culminando no suporte contínuo do software concluído (PRESSMAN, 2016).

Figura 9 - Metodologia de desenvolvimento Cascata



Fonte: PRESSMAN, 2016

2.7. EDUCAÇÃO ABERTA

A Educação Aberta é aplicada em diferentes contextos, envolvendo diversas práticas, algumas bem atuais e outras não. Portanto, cabe ressaltar, contudo, que não é a utilização de um REA que constitui a Educação Aberta como tal, mas sim o modo como ela é planejada e executada. Desta forma, a Educação Aberta pode ser entendida de diversos modos, entretanto, sua real caracterização em todos os conceitos e aplicações necessita de um conjunto de práticas que dependem do contexto e do processo de ensino e aprendizagem para que venha a acontecer (FERREIRA, JACQUES L; CORRÊA, YGOR, 2019).

A Educação Aberta está relacionada à flexibilização do processo de ensino e aprendizagem compatível ao estilo de vida do aluno. No contexto virtual de Educação Aberta, o aluno precisa desenvolver-se a partir da sua autoaprendizagem, tendo o reconhecimento do curso/disciplina realizada de modo formal ou informal por meio de certificação opcional. Nesta modalidade educacional o aluno está isento de pagamento e de qualificações ou pré-requisitos, os quais poderiam configurar-se

como barreiras para o acesso à Educação. Diante de sua caracterização, a Educação Aberta centra-se no desenvolvimento de práticas pedagógicas focadas no aluno, assim como na utilização de materiais criados pelos mesmos e, principalmente, pela acessibilidade à educação diante da diversidade humana (AIRES, 2016).

2.8. A IMPORTANCIA DOS CURSOS ONLINE

O acesso móvel tem ampliado de forma indiscutível e incomensurável a possibilidade de uma comunicação pervasiva e total; assim, os cursos que colocam seu foco na interação do aluno com o conteúdo e com a comunidade podem trazer resultados satisfatórios para os participantes, uma vez que o diálogo está posto como elemento importante na teoria e na prática pedagógicas, consideração que tem sua validade total no ambiente da aprendizagem ubíqua. O móvel parece ser uma abordagem ideal para a efetivação de um processo de comunicação que, se ainda tinha alguma barreira em outras abordagens, encontra campo aberto para efetivação nesta abordagem de aprendizagem (MUNHOZ, 2016, p. 37).

2.9 NOVAS POSSIBILIDADES DE APRENDIZAGEM

Na atualidade, os recursos tecnológicos estão cada vez mais presentes no contexto educacional, visando proporcionar um maior desenvolvimento e maior facilidade de acesso à informação e à comunicação, estendida a todos os participantes de processos de ensino e aprendizagem (MUNHOZ, 2016, p. 11).

Estudiosos do assunto consideram que, com o suporte e o apoio das tecnologias é possível conectar alunos e educadores para obtenção, de forma extensiva e diferenciada, de acesso a um grande volume de recursos coletivos e conhecimentos disseminados na grande rede mundial de comunicações, a internet (MUNHOZ, 2016, p. 12).

Assim, a sociedade da aprendizagem utiliza a tecnologia como potente ferramenta para enfrentar a explosão do conhecimento e, impulsionada pela força que os ambientes em rede demonstram, atender aos processos de formação, sejam eles formais ou informais, necessários frente as demandas das sociedades atuais, que estão centradas no conhecimento (MUNHOZ, 2016, p. 12).

3. MATERIAIS E MÉTODOS

Para realização deste projeto, optou-se por utilizar o método de pesquisa exploratória aplicada e método de desenvolvimento de software cascata.

3.1. FASES E ESTUDOS

3.1.1. Modelagem

Utilizamos o diagrams.net para criarmos os diagramas de casos de uso e de classe, o diagrama de casos de uso descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários no sistema, definimos todas as classes que o sistema precisa ter nos diagramas de classe. Será utiliza o Whimsical que é uma ferramenta para criação de protótipos visuais.

3.1.2. Programação

Será utilizado o *Visual Studio Code* para escrever o código pois é uma ferramenta simples, possui suporte para diversas linguagens incluindo Javascript que é a linguagem escolhida para o desenvolvimento do projeto, utilizaremos Node.js para execução de códigos no *back-end*, ele é responsável por tudo o que acontece por trás da tela, será utilizado o Yarn que nos permite gerenciar bibliotecas Node.js, com ele podemos adicionar *frameworks* ao nosso projeto como o express.js que facilita no desenvolvimento de aplicações *back-end*, ReacJS será utilizado para criar interfaces de usuário em páginas web, o banco de dados do website será PostgreSQL pois possui uma forte reputação devido a sua estabilidade e integridade de dados.

3.2. CASOS DE USO

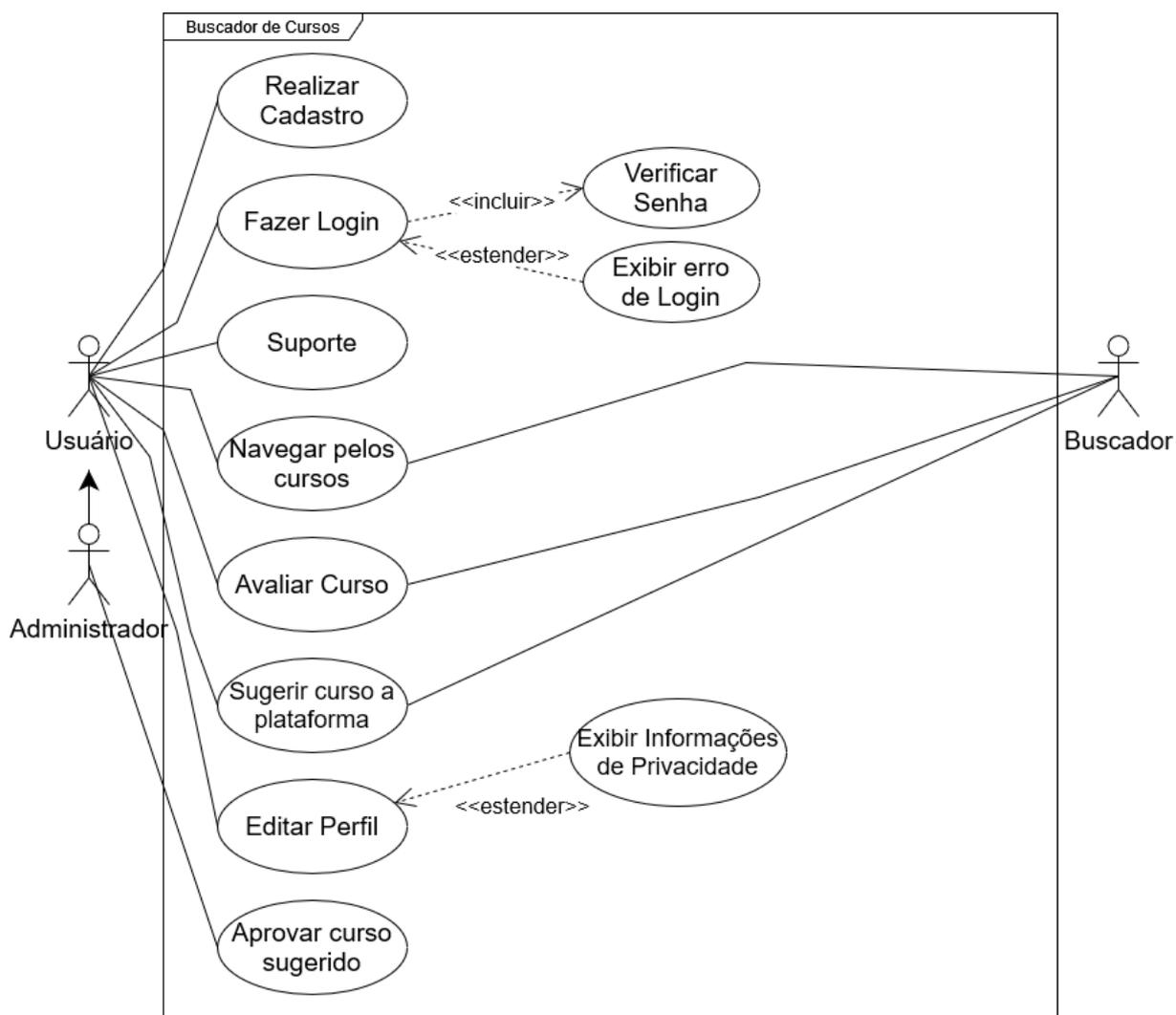
Segundo Guedes (2008), o diagrama de caso de uso é o mais abstrato dentre todos os diagramas da UML (Unified Modeling Language ou linguagem de modelagem unificada) e tem como objetivo apresentar de forma simples o comportamento e principais funcionalidades do sistema a ser desenvolvido para qualquer pessoa.

Para Silva (2007), o diagrama de caso de uso utiliza as especificações funcionais do sistema modelando os elementos externos ao software com os quais se interagem. Trata-se de funcionalidade completa que deve interagir com uma entidade

externa (atores) e representa apenas uma identificação, sem necessariamente descrever de forma completa como uma determinada funcionalidade irá ser executada.

O diagrama de caso de uso é supra valia na análise de requisitos possibilitando o analista a identificar e documentar as funcionalidades, características pessoas envolvidas no em cada caso de utilização do sistema (GUEDES, 2008).

Figura 10 – Diagrama de caso de uso



Fonte: Próprio Autor

3.3. DOCUMENTAÇÃO DO CASO DE USO

Para Guedes (2008), a documentação do caso de uso é uma descrição bem simples das etapas, parâmetros, pós- condições, pré-condições, restrições e validações que um determinado ator irá executar para que o objetivo do caso de uso seja alcançado.

Seguindo o modelo de Guedes (2008), o Quadro 1 define o modelo de documentação de caso de uso que será utilizado para a documentação do sistema a ser desenvolvido.

Quadro 1 - Modelo de documentação de caso e uso

Nome		
Caso de uso geral		
Ator principal		
Atores secundários		
Resumo		
Pré-condições		
Pós-condições		
	Ações do ator	Ações do sistema
Restrições/Validações		

Tendo realizado essas definições os quadros a seguir especificam as funcionalidades do sistema.

Quadro 2 – Documentação do caso de realizar cadastro

Nome	Realizar Cadastro
Caso de uso geral	Realizar Cadastro
Ator principal	Usuário
Atores secundários	

Resumo	Este caso de uso descreve as etapas para que o sistema possa realizar o cadastro do usuário no sistema	
Pré-condições		
Pós-condições	Uma mensagem de sucesso ou erro deverá ser exibida	
	Ações do ator	Ações do sistema
	1. O usuário clica na opção de cadastrar	
		2. Exibe tela de cadastro
	3. Exibe tela de cadastro	
		4. Realiza o cadastro
Restrições/Validações	A combinação dos parâmetros deverá ser valida	

Quadro 3 – Documentação do caso de fazer login

Nome	Fazer login	
Caso de uso geral	Fazer login	
Ator principal	Usuário	
Atores secundários		
Resumo	Este caso de uso descreve as etapas para que o sistema possa realizar a autenticação do usuário no sistema	
Pré-condições	O usuário deverá existir	
Pós-condições	Uma mensagem de sucesso ou erro deverá ser exibida	
	Ações do ator	Ações do sistema
	1. O usuário clica na opção de entrar	
		2. Exibe tela de autenticação
	3. Insere dados de acesso	
		4. Inicializa sessão
Restrições/Validações	A combinação dos parâmetros deverá ser valida	

Quadro 4 – Documentação do caso de verifica senha

Nome	Verifica senha
Caso de uso geral	Fazer Login

Ator principal	Usuário
Atores secundários	
Resumo	Verifica os dados do usuário
Pré-condições	O usuário deverá existir
Pós-condições	
Ações do ator	Ações do sistema
Restrições/Validações	A combinação dos parâmetros deverá ser válida

Quadro 5 – Documentação do caso de exibir erro de login

Nome	Exibir erro de login
Caso de uso geral	Fazer Login
Ator principal	Usuário
Atores secundários	
Resumo	Retorna o erro no caso de os dados informados serem incorretos.
Pré-condições	O usuário deverá existir
Pós-condições	Uma mensagem de sucesso ou erro deverá ser exibida
Ações do ator	Ações do sistema
Restrições/Validações	A combinação dos parâmetros deverá ser válida

Quadro 6 – Documentação do caso de avaliar curso

Nome	Avaliar curso
Caso de uso geral	Avaliar curso

Ator principal	Usuário
Atores secundários	Buscador
Resumo	Este caso de uso descreve as etapas para que o usuário possa avaliar os cursos.
Pré-condições	O usuário deve estar logado
Pós-condições	
Ações do ator	Ações do sistema
1. O usuário clica no curso	
	2. Exibe a tela do curso
3. O usuário clica na opção de avaliar curso	
	4. Exibe a tela de avaliação de curso
5. O usuário insere os dados da avaliação	
	6. Cadastra avaliação
Restrições/Validações	

Quadro 7 – Documentação do caso de sugerir curso a biblioteca

Nome	Sugerir curso a biblioteca
Caso de uso geral	Sugerir curso a plataforma
Ator principal	Usuário
Atores secundários	Buscador
Resumo	Este caso de uso descreve as etapas para que o usuário possa sugerir a adição de cursos a plataforma.
Pré-condições	O usuário deve estar logado
Pós-condições	
Ações do ator	Ações do sistema
1. O usuário clica em sugerir curso	
	2. Exibe a tela de sugestão de cursos
3. Insere os dados do curso	
	4. Cadastra a sugestão

Restrições/Validações	
------------------------------	--

Quadro 8 – Documentação do caso de editar perfil

Nome	Editar perfil	
Caso de uso geral	Editar perfil	
Ator principal	Usuário	
Atores secundários	Buscador	
Resumo	Este caso de uso descreve as etapas para que o usuário possa editar as informações do seu perfil.	
Pré-condições	O usuário deve estar logado	
Pós-condições		
	Ações do ator	Ações do sistema
	1. O usuário clica em editar perfil	
		2. Exibe a tela de editar perfil
	3. Insere os dados a serem editados	
		4. Atualiza o cadastro do usuário
Restrições/Validações		

Quadro 9 – Documentação do caso de exibir informações de privacidade

Nome	Exibir informações de privacidade	
Caso de uso geral	Exibir informações de privacidade	
Ator principal	Usuário	
Atores secundários	Buscador	
Resumo	Este caso de uso descreve as etapas para que o usuário possa exibir informações de privacidade.	
Pré-condições		
Pós-condições		
	Ações do ator	Ações do sistema
	1. O usuário clica em exibir informações de privacidade	

	2. Exibe a tela de informações de privacidade
3. Clica em declarar ciência dos termos de privacidade	
	4. Cadastra aceitação do usuário
Restrições/Validações	

Quadro 10 – Documentação do caso de aceitar sugestão de curso

Nome	Aceitar sugestão de curso	
Caso de uso geral	Aprovar sugestão de curso	
Ator principal	Administrador	
Atores secundários	Buscador	
Resumo	Este caso de uso descreve as etapas para que o usuário possa exibir informações de privacidade.	
Pré-condições	O usuário deve estar logado	
Pós-condições		
	Ações do ator	Ações do sistema
	1. O usuário clica em exibir sugestões de cursos	
		2. Exibe a tela de sugestões de cursos
	3. Confere os dados do curso e clica em aprovar sugestão	
		4. Cadastra curso a biblioteca
Restrições/Validações		

Quadro 11 – Documentação do caso de uso de suporte

Nome	Entrar em contato com o suporte
Caso de uso geral	Suporte
Ator principal	Usuário
Atores secundários	Buscador

Resumo	Este caso de uso descreve as etapas para que o usuário possa entrar em contato com o suporte.	
Pré-condições	O usuário deve estar logado	
Pós-condições		
	Ações do ator	Ações do sistema
	1. O usuário clica em exibir sugestões de cursos	
		2. Exibe a tela de sugestões de cursos
	3. Confere os dados do curso e clica em aprovar sugestão	
		4. Cadastra curso a biblioteca
Restrições/Validações		

3.4. REQUISITOS FUNCIONAIS

Para Wazlawick (2006), requisitos funcionais corresponde a tudo que seja necessário para que o sistema deve realizar e os classifica em dois tipos:

- Evidentes: São os que necessitam da interação com o usuário, ou seja, corresponde a todas as entradas e saídas de informações.
- Ocultas: São os que não necessitam de interação com o usuário, ou seja, os que são executados de forma automática pelo sistema.

3.5. REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais correspondem a restrições que os requisitos funcionais devem seguir podendo ser obrigatórios ou não obrigatórios, podendo também ser classificados por atributos: requisitos de implementação, interface. Os requisitos funcionais também podem ser transitórios e permanentes, ou seja, podem ser alterados conforme o tempo ou nunca serão necessários realizar alterações (WAZLAWICK, 2006).

Seguindo os conceitos de Wazlawick (2006) o define o modelo que será utilizado para apresentar as especificações do aplicativo que será desenvolvido. Para as categorias das especificações não funcionais seguem os seguintes conceitos:

- Usabilidade – Nessa categoria encaixa-se quaisquer fatores que terão a influência da interação com seus utilizadores, exemplo: ajuda do sistema, manuais, tipo de informações que o sistema irá prove.
- Confiabilidade – Essa categoria diz respeito ao gerenciamento de falhas, exemplo: Quais possíveis falhas o sistema deverá gerenciar, Comunicação, energia, entrada de dados entre outros.
- Performance – Nessa categoria são estabelecidos quais serão os princípios de eficiência e precisão o software deverá seguir, como exemplo podemos definir o tempo para a abertura de uma determinada tela ou execução de uma funcionalidade.

Quadro 12 - Exemplo de especificações funcionais e não funcionais

F1 – Nome da Especificação				Oculto ()	
Descrição: Descrever de forma detalhada o requisito					
Requisitos não funcionais					
Nome	Restrição	Categoria	Desejável	Permanente	
NF1.1	Descrever a restrição do requisito não funcional	--	Sim/Não	Sim/Não	

Quadro 13 - Especificações não funcionais

Nome	Restrição	Categoria	Desejável	Permanente
S1 abordagem de desenvolvimento	A abordagem para o desenvolvimento deverá seguir a abordagem de desenvolvimento cascata	Implementação	Sim	Sim
S2 Framework para desenvolvimento	A aplicação deverá fazer o uso do ExpressJS	Implementação	Sim	Sim

S3 Armazenamento dos dados	A persistências dos dados deveram ser enviadas para o servidor	Persistência	Sim	Sim
S4 Interação com usuário	O sistema deverá fornecer avisos visuais para quaisquer erros que ocorrem para os usuários	Usabilidade	Sim	Sim

Quadro 14 – Segurança de dados do usuário

F1 – Segurança dos dados do usuário				Oculto ()
Descrição: Todas as senhas e dados sensíveis do usuário serão criptografadas e não poderão ser enviadas novamente ao usuário.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1	O sistema criptografara todos os dados sensíveis do cadastro do usuário.	Segurança	Sim	Sim

Quadro 15 – Compatibilidade com navegadores de internet

F2 – Compatibilidade com navegadores de internet				Oculto ()
Descrição: O web site a ser desenvolvido terá compatibilidade com os seguintes navegadores: Google Chrome, Mozilla Firefox, Opera, Edge, Safari				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1	O sistema deverá ser compatível com os seguintes navegadores	Compatibilidade	Sim	Sim

	Google Chrome, Mozilla Firefox, Opera, Edge, Safari.			
--	--	--	--	--

Quadro 16 - Uso de Design responsivo nas interfaces gráficas

F3 – Uso de Design responsivo nas interfaces gráficas				Oculto ()
Descrição: Deverá possuir um design responsivo.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1	A interface do sistema deverá se comportar adequadamente independente do front-end que será utilizado. para acesso – Browser, Smartphone ou Tablet.	Usabilidade	Sim	Sim

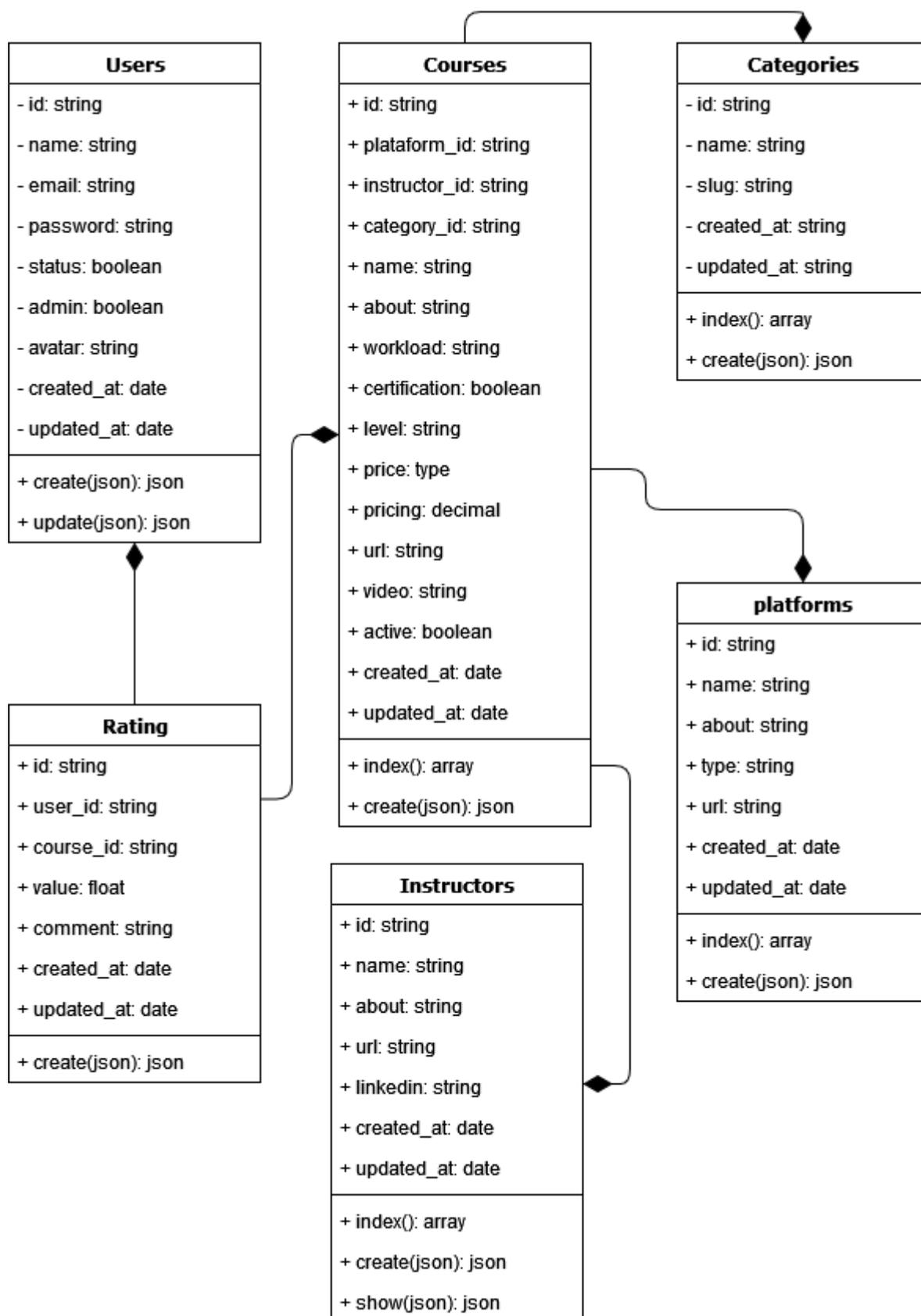
3.6. DIAGRAMA DE CLASSES

Para Guedes (2008), o diagrama de classe é, sem dúvidas, um dos principais diagramas da UML. É composto por classes que possuem atributos e métodos podendo haver relacionamentos com outras classes. O diagrama de classe visa demonstrar como as classes desse diagrama se comunicam e expõem seus dados para as demais.

Segundo Silva (2007), o diagrama de classe corresponde a um modelo gráfico de um programa orientado a objeto, descrevendo sua estrutura e os relacionamentos entre suas classes³⁶ de forma legível, modelo que pode ser diretamente implementado em uma linguagem de programação.

Na Figura 11 podemos ver o diagrama de classe da aplicação de coleta de dados a ser desenvolvida.

Figura 11 – Diagrama de classe



Fonte: Próprio Autor

5. RECURSOS

Tabela 1 – Ferramentas / Tecnologias

Ferramenta / Tecnologia	Versão	Finalidade
NodeJS	14.17.0 LTS	Permite a execução de códigos JavaScript fora de um navegador web
Yarn	1.22.5	Gerenciar bibliotecas NodeJS
Javascript	ES8	Linguagem utilizada para o desenvolvimento
HTML	5	Linguagem de marcação de hipertexto para apresentar e estruturar o conteúdo na web
CSS	3	Linguagem para adicionar estilo a um documento web
Express.js	4.17.1	Framework NodeJS
ReactJS	17.0.2	Criar interfaces de usuário em páginas web
PostgreSQL	13.3	Banco de dados
Whimsical	Não definida pelo desenvolvedor	Ferramenta utilizada para protótipo visual
Diagrams.net	Não definida pelo desenvolvedor	Ferramenta utilizada para criação de diagramas
Insomnia REST	2021.3.0	Testar API
Visual Studio Code	1.56	Escrever os códigos do site
Github	Não definida pelo desenvolvedor	Versionamento do código

6. DESENVOLVIMENTO

De acordo com os objetivos estabelecidos do presente trabalho, faz-se necessário o desenvolvimento das funcionalidades descritas nas seções anteriores. Para tanto iremos demonstrar como realizar a configuração do ambiente de desenvolvimento do projeto, interfaces de comunicação, e as funcionalidades e telas do aplicativo desenvolvido.

6.1 CONFIGURAÇÃO DO PROJETO

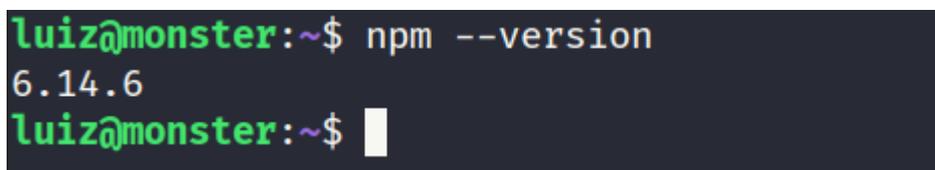
Devemos levar em consideração o sistema operacional ao configurar o projeto. Utilizamos o Ubuntu 20.04 LST que é um sistema operacional livre baseado no Unix, usando kernel Linux que é o núcleo do sistema operacional.

6.2.1. Instalação do NODEJS

```
$ curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash
$ sudo apt-get update
$ sudo apt-get install -y nodejs
```

Após realizada a instalação do nodeJS teremos acesso ao seu gerenciador de pacotes NPM.

Figura 12 – Terminal com comandos



```
luiz@monster:~$ npm --version
6.14.6
luiz@monster:~$
```

Fonte: Próprio Autor

6.2.2. Instalação do Yarn

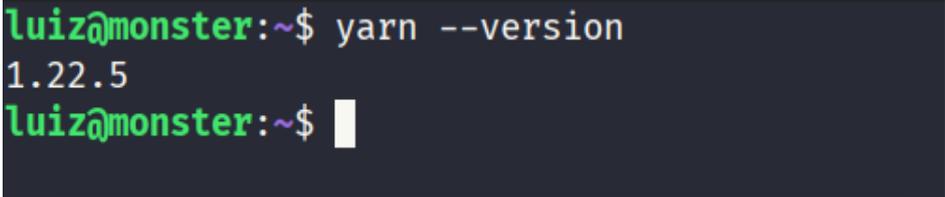
Utilizaremos o Yarn para gerenciar as bibliotecas Javascript em nosso projeto, utilizaremos o gerenciador de pacotes do Nodejs.

```
$ npm install --global yarn
```

Agora temos disponível o Yarn que é o gerenciador de bibliotecas utilizado no projeto

```
$ yarn --version
```

Figura 13 – Terminal com comandos



```
luiz@monster:~$ yarn --version
1.22.5
luiz@monster:~$
```

Fonte: Próprio Autor

6.2.3. Instalação do PostgreSQL

Para instalar o PostgreSQL é necessário executar o seguinte comando no terminal.

```
$ sudo apt install postgresql postgresql-contrib
```

Agora teremos acesso à linha de comando do PostgreSQL através do terminal e executar comandos SQL.

```
$ sudo -u postgres psql
```

Devemos criar uma base de dados para o projeto utilizando os comandos abaixo:

```
# createdb fixa
```

6.2.4. Instalação do Visual Studio Code

Etapas para instalação do Visual Studio Code devemos acessar o terminal e executar o seguinte comando para a instalação da aplicação.

```
$ sudo snap install --classic code
```

7.3 PADRÃO DE COMUNICAÇÃO

Para uma melhor identificação e entendimento das requisições, seus retornos e entrada de dados, iremos apresentá-las na seguinte estrutura:

Serviço	Curta explicação do serviço Rota
Rota	Curta explicação do serviço Rota
Método	Rota para o serviço que iremos consumir
Retorno	Padrão e exemplo de resposta

```
{
  "exemplo": "Exemplo de resposta"
}
```

6.3.1. Serviço de usuários

Serviço	Responsável por retornar os dados do usuário
Rota	/users
Método	GET
Retorno	JSON

```
{
  "usuarios": [{
    "id": "Identificação",
    "name": "Nome",
    "email": "E-mail",
    "avatar_url": "Endereço do avatar do usuário"
  }, ...]
}
```

6.3.2. Serviço de categorias

Serviço	Responsável por retornar os dados de categorias
Rota	/categories
Método	GET
Retorno	JSON

```
{  
  "categories": [{  
    "id": "Identificação",  
    "name": "Nome"  
  }, ...]  
}
```

6.3.3. Serviço de plataformas

Serviço	Responsável por retornar os dados das plataformas
Rota	/platforms
Método	GET
Retorno	JSON

```
{  
  "platforms": [{  
    "id": "Identificação",  
    "name": "Nome",  
    "about": "Sobre",  
    "type": "Tipo da plataforma",  
    "url": "Url da plataforma"  
  }, ...]  
}
```

6.3.4. Serviço de professores

Serviço	Responsável por retornar os dados dos professores
Rota	/instructors
Método	GET
Retorno	JSON

```
{  
  
  "instructors": [{  
  
    "id": "Identificação",  
  
    "name": "Nome",  
  
    "about": "Sobre",  
  
    "url": "Url da plataforma",  
  
    "linkedin": "Url do perfil no linkedin",  
  
  }, ...]  
  
}
```

6.3.5. Serviço de cursos

Serviço	Responsável por retornar os dados dos cursos
Rota	/courses
Método	GET
Retorno	JSON

```
{  
  
  "courses": [{  
  
    "id": "Identificação",  
  
    "platform_id": "Plataforma",  
  
    "category_id": "Categoria",  
  
    "instructor_id": "Instrutor",  
  
  }  
  
}
```

```

    "name": "Nome",
    "about": "Sobre",
    "workload": "Carga horaria",
    "certification": "Certificação",
    "level": "Nível de conhecimento",
    "pricing": "Precificação",
    "url": "Url",
    "vídeo": "Video",
    "active": "Ativo",
  }, ...]
}

```

6.3.6. Serviço de avaliação

Serviço	Responsável por retornar os dados dos cursos
Rota	/courses
Método	GET
Retorno	JSON

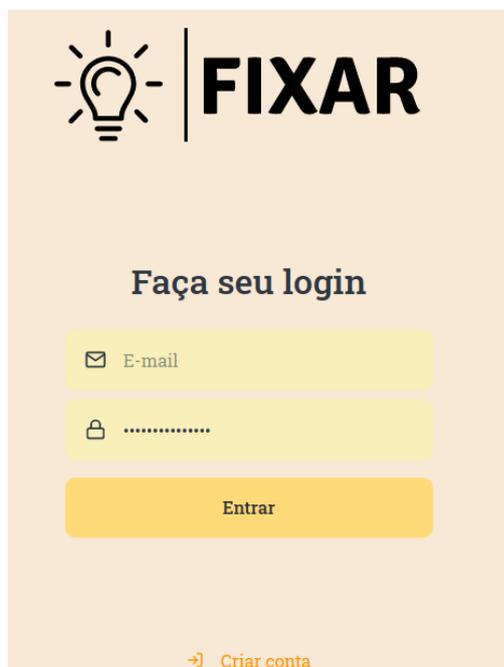
```

{
  "courses": [{
    "id": "Identificação",
    "user_id": "Usuário",
    "course_id": "Curso",
    "value": "Instrutor",
    "comment": "Comentário"
  }, ...]
}

```

6.4 FUNCIONALIDADES

Figura 14 – Tela de login do site



A tela de login do site 'FIXAR' apresenta o seguinte layout:

- Logo: Um ícone de lâmpada acesa à esquerda de uma barra vertical, seguida pelo nome 'FIXAR' em letras maiúsculas e negritadas.
- Título: 'Faça seu login' em negrito.
- Formulário de entrada:
 - Um campo de texto amarelo claro com um ícone de envelope e o rótulo 'E-mail'.
 - Um campo de texto amarelo claro com um ícone de cadeado e caracteres ocultos por pontos.
 - Um botão amarelo sólido com o texto 'Entrar'.
- Link: Um link amarelo com um ícone de seta e o texto 'Criar conta'.

Fonte: Próprio Autor

A tela de autenticação do sistema é a entrada do site. Conforme demonstrado na Figura 14, nela é possível digitar o usuário e senha, para autenticar-se e realizar a utilização do site.

Figura 15 – Tela de login com exceções



A tela de login do sistema FIXAR apresenta o logo de uma lâmpada e o nome 'FIXAR' no topo. O título 'Faça seu login' está centralizado. Abaixo dele, há dois campos de entrada: 'E-mail' com ícone de envelope e 'Senha' com ícone de cadeado. Cada campo possui um ícone de alerta (exclamação dentro de um círculo) no canto inferior direito. Um botão amarelo 'Entrar' está posicionado abaixo dos campos. Na base da tela, há um link '→ Criar conta'.

Fonte: Próprio Autor

Figura 16 – Tela de cadastro de usuário



A tela de cadastro de usuário do sistema FIXAR apresenta o mesmo logo de lâmpada e o nome 'FIXAR' no topo. O título 'Faça seu cadastro' está centralizado. Abaixo dele, há três campos de entrada: 'Nome' com ícone de pessoa, 'E-mail' com ícone de envelope e um campo de senha representado por pontos. Um botão amarelo 'Cadastrar' está posicionado abaixo dos campos. Na base da tela, há um link '← Voltar para logon'.

Fonte: Próprio Autor

Figura 17 – Tela de cadastro de usuário com exceções



A tela de cadastro de usuário apresenta o logo 'FIXAR' com um ícone de lâmpada. O título principal é 'Faça seu cadastro'. Há três campos de entrada: 'Nome', 'E-mail' e 'Senha', cada um com um ícone de erro (exclamação dentro de um círculo vermelho) no canto superior direito. Abaixo dos campos está um botão amarelo 'Cadastrar'. No rodapé, há um link '← Voltar para logon'.

Fonte: Próprio Autor

Figura 18 – Tela de cursos



A tela de cursos possui um cabeçalho amarelo com o logo 'FIXAR' e um botão 'Adicionar curso'. À direita, há uma seção de usuário com o nome 'Bem-vindo, David' e um ícone de perfil. O conteúdo principal é dividido em duas colunas. A esquerda, sob o título 'Todos os cursos disponibilizado Cursos', há dois cartões de curso. O primeiro cartão é para 'Curso de JavaScript e TypeScript do básico ao avançado 2021', com plataforma 'Udemy', categoria 'Desenvolvimento desktop', instrutor 'Abraão' e duração de 145 horas. O segundo cartão é para 'JavaScript do básico ao avançado (c/ Node.js e projetos)', com plataforma 'Udemy', categoria 'Desenvolvimento Web', instrutor 'Abraão' e duração de 22,5 horas. A direita, há duas listas de categorias e plataformas. As categorias são 'Desenvolvimento desktop', 'Desenvolvimento Web' e 'Desenvolvimento mobile'. As plataformas são 'Udemy', 'Sebrae', 'Alura' e 'Fundação Bradesco'.

Fonte: Próprio Autor

Figura 19 – Tela de cursos do administrador



Fonte: Próprio Autor

A barra de navegação lateral permite que o usuário navegue por categorias de cursos e por plataformas que disponibilizam os cursos.

Figura 20 – Tela de curso



Fonte: Próprio Autor

Figura 21 – Tela de avaliações

Curso disponibilizado por **Udemy**

Avaliações 1

☆☆☆☆☆

 David em 25/11/2021 11:59

Ótimo curso

Avalie esse curso

Escreva aqui seu comentário

 Avaliar

Fonte: Próprio Autor

Figura 22 – Tela de adicionar cursos

Adicione o curso desejado

Contribua compartilhando cursos que você gostou 🥰

Nome do curso

Descrição do curso

Carga horária

Endereço do curso (site)

Vídeo de apresentação do curso

Precificação

Grátis

Certificação válida

Com certificação

Nível de conhecimento desejável

Básico

Categoria

Desenvolvimento desktop

Instrutor

Abraão

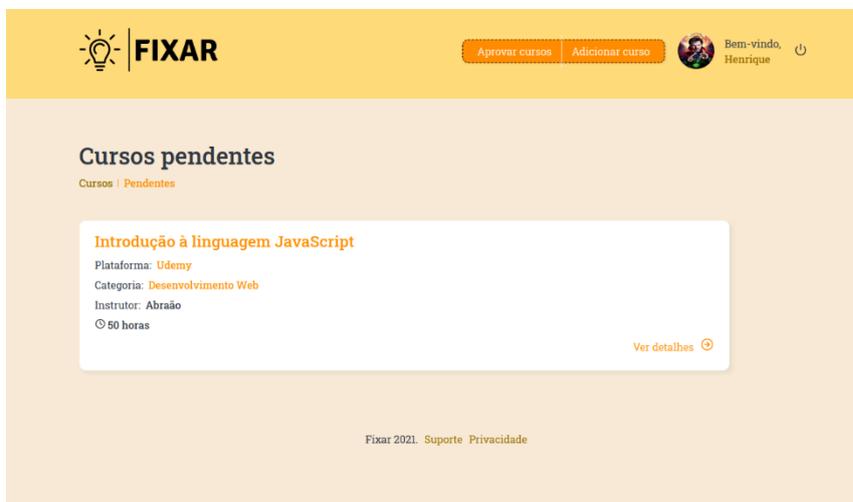
Plataforma

Udemy

Cadastrar

Fonte: Próprio Autor

Figura 23 – Tela de cursos pendentes



Fonte: Próprio Autor

O administrador pode navegar pelos cursos pendentes que foram publicados pelos usuários da plataforma.

Figura 24 – Tela de aprovar cursos



Fonte: Próprio Autor

O administrador pode verificar as informações do curso e após validar ele pode aprovar o curso e disponibilizar o mesmo para todos os usuários.

Figura 25 – Tela de editar perfil

←



Meu perfil

 Henrique Moura

 luizhom@outlook.com

 Senha atual

 Nova senha

 Confirmar senha

Confirmar mudanças

Fixar 2021. Suporte Privacidade

Fonte: Próprio Autor

Figura 26 – Tela de informações de privacidade

←  **FIXAR**

Coleta e usos de informações pessoais

O Usuário está ciente de que fornece informação de forma consciente e voluntária por meio de [FORMULÁRIO/ETC]. Quando o Usuário realiza o cadastro e/ou preenche formulários oferecidos pela FIXAR, determinados Dados Pessoais solicitados serão mantidos em sigilo e serão utilizadas apenas para o propósito que motivou o cadastro.

Compartilhamento e tratamento de informações pessoais

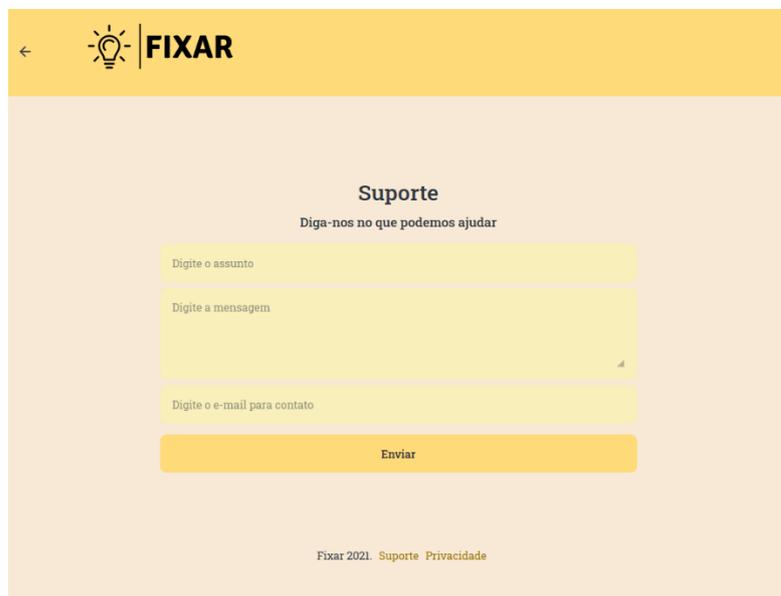
A Fixar não disponibilizará Dados Pessoais coletados em seus sites para corretores de lista de e-mail sem seu expresso consentimento.

A Fixar poderá divulgar os Dados Pessoais coletados a terceiros, nas seguintes situações e nos limites exigidos e autorizados pela Lei:

1. Com os seus clientes e parceiros quando necessário e/ou apropriado à prestação de serviços relacionados;
2. Com as empresas e indivíduos contratados para a execução de determinadas atividades e serviços em nome da Fixar;
3. Com empresas do grupo
4. Com fornecedores e parceiros para consecução dos serviços contratados com a Fixar (como tecnologia da informação, contabilidade, entre outros);
5. Para propósitos administrativos como: pesquisa, planejamento, desenvolvimento de serviços, segurança e gerenciamento de risco.
6. Quando necessário em decorrência de obrigação legal, determinação de autoridade competente, ou decisão judicial. Nas hipóteses de compartilhamento de Dados Pessoais com terceiros, todos os sujeitos mencionados nos itens 1 a VI deverão utilizar os Dados Pessoais partilhados de maneira consistente e de acordo com os propósitos para os quais foram coletados (ou com os quais o Usuário consentiu previamente) e de acordo com o que foi determinado por esta Política de Privacidade, outras declarações de privacidade de website ou países, e todas as leis de privacidade e proteção de dados aplicáveis.

Fonte: Próprio Autor

Figura 27 – Tela de suporte



A tela de suporte do aplicativo Fixar apresenta um cabeçalho amarelo com um ícone de lâmpada e o nome 'FIXAR'. O conteúdo principal é centralizado e contém o título 'Suporte' e o subtítulo 'Diga-nos no que podemos ajudar'. Há três campos de entrada amarelos para 'Digite o assunto', 'Digite a mensagem' e 'Digite o e-mail para contato'. Um botão amarelo 'Enviar' está posicionado abaixo dos campos. No rodapé, há o texto 'Fixar 2021. Suporte Privacidade'.

←  **FIXAR**

Suporte
Diga-nos no que podemos ajudar

Digite o assunto

Digite a mensagem

Digite o e-mail para contato

Enviar

Fixar 2021. Suporte Privacidade

Fonte: Próprio Autor

7. CONSIDERAÇÕES FINAIS

O propósito deste trabalho foi desenvolver uma biblioteca de cursos online. O usuário poderá navegar pelos cursos e também poderá cadastrar novos cursos a plataforma que serão sujeitos a aprovação dos administradores, o conteúdo aqui apresentado como base teórica foi utilizado para fazer a modelagem do sistema por meio de um diagrama UML, no qual descrevemos o caso de uso, especificamos os requisitos funcionais e criamos o diagrama de classes que foram utilizados para a modelagem do projeto e das bases de dados. Utilizamos o NodeJS para gerenciar todas as funcionalidades da aplicação, o responsável por armazenar todas informações da website é o PostgreSQL, o NodeJS serve os dados que serão acessadas por meio da aplicação front-end desenvolvida em ReactJS que ficou encarregado por toda a parte de interação e coleta de dados do usuário.

A utilização do Express framework no NodeJS como tecnologia de desenvolvimento foi de grande valia, pois propiciou a facilidade da escrita de código ao logo do projeto graças à quantidade de recursos que ele oferece. O visual do aplicativo final ficou responsivo para aprimorar a experiência dos usuários em dispositivos móveis que possuem navegador de internet. O desenvolvimento do website de biblioteca de cursos online sugeridos pelos usuários foi concluído conforme o que foi planejado no diagrama de UML.

Desenvolvemos o website: Fixar biblioteca de cursos online, uma plataforma buscadora de cursos online utilizando a linguagem de programação Javascript, fizemos de forma que o site forneça feedback aos usuários em todas as interações do usuário com o sistema como foi estabelecido nos objetivos.

REFERÊNCIAS

4.LINUX. **O que é Linux?**. 4.LINUX. SÃO PAULO. 1 p. Disponível em: [https://4linux.com.br/o-que-e-](https://4linux.com.br/o-que-e-linux/#:~:text=Linux%20é%20um%20Sistema%20Operacional,ser%20livremente%20modificado%20e%20distribuído.&text=O%20Kernel%20Linux%20foi%20criado,versão%20oficial%20lançada%20em%201991)

[linux/#:~:text=Linux%20é%20um%20Sistema%20Operacional,ser%20livremente%20modificado%20e%20distribuído.&text=O%20Kernel%20Linux%20foi%20criado,versão%20oficial%20lançada%20em%201991](https://4linux.com.br/o-que-e-linux/#:~:text=Linux%20é%20um%20Sistema%20Operacional,ser%20livremente%20modificado%20e%20distribuído.&text=O%20Kernel%20Linux%20foi%20criado,versão%20oficial%20lançada%20em%201991). Acesso em: 03 jun. 2021.

Agência KAIZEN. **Desenvolvimento de Sites**. Disponível em: <https://www.agenciakaizen.com.br/desenvolvimento-de-sites/>. Acesso em: 6 jun. 2021.

AIRES, L. E-learning, **Educação Online e Educação Aberta: contributos para uma reflexão teórica**. 2016. Disponível em: <http://revistas.uned.es/index.php/ried/article/view/14356>. Acesso em: 06 junho. 2021.

AUGUSTO, Rafael. **Introdução a Git**, 08 nov. 2017. Disponível em: <https://medium.com/@rafaelvicio/introdu%C3%A7%C3%A3o-a-git-5ae36c303850>. Acesso em: 05 jun. 2021.

ALECRIN, Emerson. **Dez anos de Google Chrome: como o navegador dominou o mercado**. tecnoblog. 2018. 1 p. Disponível em: <https://tecnoblog.net/258619/google-chrome-aniversario-dez-anos/>. Acesso em: 6 jun. 2021.

ARIANE, G. **O que é CSS? Guia Básico para Iniciantes**, 11 maio. 2021. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 06 jun. 2021.

Apple. **Apple apresenta macOS Big Sur com design renovado e sofisticado**. 2020. 1 p. Disponível em: <https://www.apple.com/br/newsroom/2020/06/apple-introduces-macos-big-sur-with-a-beautiful-new-design/>. Acesso em: 6 jun. 2021.

CARVALHO, Vinícius. **PostgreSQL: Banco de dados para aplicações web modernas**. 1 ed. São Paulo: Casa do Código. 2017. 204 pag.

FERREIRA, JACQUES L; CORRÊA, YGOR. **Educação online e educação aberta: avanços, lacunas e desafios**. 2019.

FLANAGAN, David. **Javascript: The Definite Guide, 6E**. Tradução: João Eduardo Nóbrega Tortello. 6 ed. Rio Grande do Sul: Bookman Companhia Ltda, 2013. 1080 p.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4a Ed. – Atlas, São Paulo – SP, 2002.

GUEDES, G. T. **UML: uma abordagem prática**. [S.l.]: Novatec Editora, 2008.

GUEDES, G. **UML 2 – Uma Abordagem Prática** 2 ed. São Paulo: Novatec Editora Ltda. 2011. 45 pag.

KOPPALA, J. **Erp solution with reactjs. Metropolia Ammattikorkeakoulu**, 2018.

MAZZA, L. **HTML5 e CSS domine a web do futuro**. [S.l.]: Casa do Código, 2012.

MICROSOFT. Getting Started. **Visual Studio Code**, 18 jun. 2017. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 05 jun. 2021.

MILANI, André. **PostgreSQL - Guia do Programador**. 1 ed. São Paulo: Novatec Editora Ltda. 2008. 392 pag.

MOZILLA FOUNDATION. HTML5. **MDN - Mozilla Developer Network**, 22 set. 2016. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5>. Acesso em: 05 jun. 2021.

MOZILLA FOUNDATION. Introdução Express/Node. **MDN - Mozilla Developer Network**, 03 jun. 2021. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction. Acesso em: 05 jun. 2021.

MUNHOZ, ANTONIO S. **MOOCS: Produção de conteúdo educacionais**. 1 ed. São Paulo: SARAIVA Editora Ltda. 2016.

PEREIRA, C. R. **Node.js: Aplicações web real-time com Node.js**. São Paulo: Casa do Código, 2013.

PRESSMAN, ROGER; MAXIM, BRUCE. **Engenharia de Software: uma abordagem profissional**. 8 ed. São Paulo: Techbooks Editora Ltda. 2016. 42 pag.

REACT. **React - A JavaScript library for building user interfaces**. 2019. Disponível em: <https://reactjs.org/>.

ROBBESTAD, S. A. **ReactJS Blueprints**. [S.l.]: Packt Publishing Ltd, 2016.

SILBERSCHATZ, A. **Operating system concepts**. Addison-Wesley Reading, v. 4, 1998.

SILVA, R. P. E. **UML 2: modelagem orientada a objetos**. [S.l.]: Visual Books, 2007.

SOUZA, Ivan. **O que é Yarn e como funciona seu gerenciamento de pacotes**, 29 out. 2020. Disponível em: <https://rockcontent.com/br/blog/yarn/>. Acesso em: 05 jun. 2021.

STALLINGS, W. **Operating Systems: Internals and Design Principles**. Prentice Hall, 2012. ISBN ISBN: 9780132309981LCCN: 2010048597

VIPUL, A.; SONPATKI, P. **ReactJS by Example-Building Modern Web Applications with React**. [S.l.]: Packt Publishing Ltd, 2016.

WAZLAWICK, R. **Análise e Projeto de Sistemas de Informação Orientados a objetos**. [S.l.]: Elsevier Brasil, 2006.