**CENTRO PAULA SOUZA**

**ETEC ANTONIO FURLAN**

**Técnico em Desenvolvimento de Sistemas**

**Fabiana Neri Mendes**

**Lucas Santos Silva**

**Mateus Silva dos Santos**

**Samuel Cursino Moya de Souza**

**ART.ME**

**Barueri**

**ART.ME**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da Etec Antônio Furlan, orientado pelo Prof. Bruno e Prof. Rinaldo, como requisito para obtenção do título de técnico em Desenvolvimento de Sistemas.

**Barueri**

**2023**

**Dedicatória**

Dedico este trabalho primeiramente a Deus, que me sustentou até o presente momento, a minha família por todo o apoio e assistência e aos prezados colegas de curso.

**Agradecimento**

Gostaríamos de expressar nossa sincera gratidão a todos que nos apoiaram, tanto familiares, quanto amigos e professores, pois sem tal apoio e ajuda nosso trabalho não seria concretizado.

Gostaria de agradecer aos orientadores, por sua orientação e paciência, principalmente em momentos decisivos e aos mentores, por toda assistência em compromisso no período de elaboração do projeto.

Por fim agradeço a instituição de ensino Etec Antônio Furlan que me proporcionou um ambiente propício para o desenvolvimento do nosso projeto e pelo conhecimento que foi adquirido.

*“A primeira regra de qualquer tecnologia utilizada nos negócios é que a automação aplicada a uma operação eficiente aumentará a eficiência. A segunda é que a automação aplicada a uma operação ineficiente aumentará a ineficiência.”*

*-Bill Gates*

**Resumo**

Hoje vemos uma deficiência no mercado de autores iniciantes no ramo das artes, onde eles tendem a publicar e anunciar suas obras em grupos fechados, boca a boca ou em redes sociais, assim ficando desfalcado em um mercado que tende a crescer. Nosso projeto almeja criar um aplicativo na plataforma *Android* baseado em *Java* usando *React Native* para ajudar pessoas venderem, comprarem ou solicitarem projetos de arte, sejam eles fotógrafos, pintores ou músicos, as tratativas serão feitas pelo aplicativo, desde a proposta, negociação e pagamento, assim assegurando o lado do solicitante e do autor. O aplicativo conta com recursos simples desde a exibição do perfil do autor até a sua contratação e fechamento do respectivo serviço com o cliente.

Palavras-chave:

**Abstract**

Today we see a deficiency in the market for beginner authors in the field of arts, where they tend to publish and advertise their works in closed groups, word of mouth or on social networks, thus being depleted in a market that tends to grow. Our project aims to create an application on the Android platform based on Java using React Native to help people sell, buy or request art projects, whether they are photographers, painters or musicians, the negotiations will be done by the application, from the proposal, negotiation, and payment, thus securing the side of the applicant and the author. The application has simple features from displaying the author's profile to hiring and closing the respective service with the customer.

Key words:

Sumário

Sumário

[INTRODUÇÃO 10](#_Toc134991339)

[Problema 10](#_Toc134991340)

[Justificativa 10](#_Toc134991341)

[Objetivo geral 10](#_Toc134991342)

[Objetivos específicos 10](#_Toc134991343)

[Metodologia 10](#_Toc134991344)

[1.1.2 Requisitos funcionais 12](#_Toc134991345)

[1.1.2 Requisitos não funcionais 14](#_Toc134991346)

[CAPÍTULO – 2 FERRAMENTAS TÉCNOLÓGICAS 15](#_Toc134991347)

[2.1 Android Studio 15](#_Toc134991348)

[2.2 HTML 15](#_Toc134991349)

[2.3 CSS 16](#_Toc134991350)

[2.4 JavaScript 16](#_Toc134991351)

[2.5 React Native 17](#_Toc134991352)

[2.6 Descrições sobre o banco de dados 17](#_Toc134991353)

[2.7 MySQL 17](#_Toc134991354)

[CAPÍTULO – 3 DIAGRAMAS 19](#_Toc134991355)

[3.1 Diagrama de Caso de Uso 19](#_Toc134991356)

[CDU01 – FUNCIONAMENTO DO SOFTWARE (diagrama de sequência) 20](#_Toc134991357)

[CDU02 – Cadastro de cliente (Caso de uso) 21](#_Toc134991358)

[3.2 Diagrama de Entidade e Relacionamento 22](#_Toc134991359)

[3.3.1 Diagrama EER 23](#_Toc134991360)

# 

# INTRODUÇÃO

Em vista sobre a dificuldade dos artistas em compartilhar seus trabalhos e conhecimentos nas redes sociais, o aplicativo mobile ART.ME ajudará os artistas a mostrar sua arte e trabalho com pessoas do mesmo ramo, assim, ajudando na procura de empregos e trabalhos.

# Problema

Ajudar os artistas em geral, assim, facilitando o compartilhamento de seus trabalhos.

# Justificativa

Visto a necessidade de facilitar o processo dos artistas pequenos e independentes, surgiu a ideia da criação de um software para ajudar em todas as etapas.

# Objetivo geral

A criação de um app mobile para artistas independentes para facilitar todo o processo de trabalho, desde compartilhar o portifólio até o fechamento de contrato.

# Objetivos específicos

O objetivo desse projeto é auxiliar o profissional de forma rápida e tecnológica com a criação de um aplicativo mobile que permite o compartilhamento de seus trabalhos.

# Metodologia

A metodologia irá foi realizada através de pesquisas quantitativas e qualitativas onde foi possível identificar que existe uma certa deficiência no mercado de artistas autônomos, tano na parte de postagem do portifólio quanto na divulgação do mesmo e dos seus serviços, sendo assim, é possível trazer um aplicativo que faça essas funções.

**Cronograma**

Para a melhor organização criamos um cronograma dividindo as tarefas de cada um do grupo.

**CAPÍTULO – 1 ANÁLISE DE REQUISITOS**

* 1. **regras de negócio**

O projeto tem como foco principal auxiliar artistas que encontram dificuldade em compartilhar o seu portifólio, e abordará todas as variedades de pessoas que trabalham nesse ramo.

Ao decorrer do tempo o software contará com atualizações futuras e/ou adições de novas informações e implementações de novas funcionalidades.

* 1. **Requisitos**

Um requisito é um aspecto que o sistema proposto deve fazer ou uma restrição no desenvolvimento do sistema. Vale ressaltar que em ambos os casos devemos sempre contribuir para resolver os problemas do cliente e não o que o programador ou um arquiteto deseja. Dessa forma, o conjunto dos requisitos como um todo representa um acordo negociado entre todas as partes interessadas no sistema. Isso também não significa que o programador, arquiteto ou um analista bem entendido no assunto de tecnologia não possam contribuir com sugestões e propostas que levem em conta o desejo do cliente.

Além disso, ainda temos um documento de requisitos que é uma coleção dos requisitos.

Por fim, os requisitos possuem alguns objetivos centrais como estabelecer e manter uma concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer, deve oferecer aos desenvolvedores, projetistas e testadores do sistema uma compreensão melhor dos requisitos do sistema, definir fronteiras do sistema definindo o que deve ser incluído e o que não deve fazer parte do sistema, fornece uma base para estimar o custo e o tempo de desenvolvimento do sistema e por fim definir uma interface de usuário para o sistema.

# 1.1.2 Requisitos funcionais

**RF01:** **Cadastrar usuário**

O Software deverá permitir a inclusão do usuário no sistema. Para isso, ele deve permitir a entrada dos seguintes dados:

* Nome: nome do cliente que será cadastrado (\*) “máximo 100 caracteres”;
* Sexo: sexo do cliente;
* Data de nascimento: data de nascimento do cliente (\*) “máximo 8 caracteres”;
* UF: unidade federal do cliente;
* Email: e-mail de escolha aleatória para o cliente;
* Senha: senha de escolha aleatória para o cliente;

**RF02: Incluir/ Excluir / Alterar dados na tela de perfil do usuário**

O software deve permitir a consulta de dados, exclusão, inclusão e alteração de dados inseridos, tais como, postagens, upload de fotos e vídeos, e categorias inseridas.

**RF03: Geração de uma linha do tempo (feed**)

O software deve permitir uma geração e interação com publicações em um feed que permitirá a visualização do serviço desejado para que seja feita a sua contratação.

**RF06: Comunicação via chat**

O software deve permitir uma comunicação via chat dentro do aplicativo para esclarecimento de dúvidas ou mesmo para ajuda.

**RF07: Contratação de um serviço**

O software deve permitir uma interação para a contratação de um serviço no momento que o cliente seleciona a opção de serviço do contratante.

**RF08: Buscas dentro do aplicativo**

O software deve permitir buscas dentro do aplicativo, para maior variedade ou para pesquisa de perfis, postagens ou serviços oferecidos.

# 1.1.2 Requisitos não funcionais

**RFN01: Perspectiva de usuário:**

O software deverá conter a opção de contratar o serviço e assim levará para a tela de chat onde ocorrerá todo uma negociação até o fechamento do serviço.

**RFN01: Hardware:** Ter compatibilidade com o sistema, e suprir as necessidades de desempenho do software para melhor proveito.

# CAPÍTULO – 2 FERRAMENTAS TÉCNOLÓGICAS

# 2.1 Android Studio

Segundo Lecheta (2015, p.25), o Android Studio é fruto da parceria entre um grupo de empresas de tecnologia e a Google, que formaram a Open Handset Alliance (OHA). Foi graças a essa aliança que o Android se tornou uma realidade.

O Android Studio é uma plataforma de desenvolvimento para dispositivos móveis que é totalmente gratuita e de código aberto. O sistema operacional do Android é baseado no kernel do Linux, o que traz muitos benefícios para o usuário. O próprio sistema é responsável por gerenciar a memória e os processos, permitindo a execução de diversas aplicações simultaneamente.

Além disso, o Android Studio possui uma interface visual rica, GPS e diversas aplicações já instaladas. O ambiente de desenvolvimento do Android Studio é poderoso, inovador e flexível.

# 2.2 HTML

HTML (Hypertext Markup Language) é uma linguagem de marcação utilizada na criação de páginas web. Embora seja mais comumente associada ao desenvolvimento de sites, o HTML também é utilizado no desenvolvimento de software.

Na criação de softwares, o HTML é utilizado para a construção da interface gráfica do usuário (GUI). Através do uso de elementos HTML, como botões, menus, caixas de seleção e campos de texto, é possível criar uma interface interativa e fácil de usar.

Para o desenvolvimento de software, o HTML pode ser utilizado em conjunto com outras tecnologias, como CSS (Cascading Style Sheets) para a estilização da interface e JavaScript para a criação de funcionalidades dinâmicas.

Além disso, com a evolução do HTML, surgiram novas tecnologias que permitem o desenvolvimento de softwares mais complexos e sofisticados, como o HTML5 e o framework AngularJS.

Em resumo, o HTML é uma ferramenta importante no desenvolvimento de software, permitindo a criação de interfaces gráficas intuitivas e funcionais. Combinado com outras tecnologias, é possível criar softwares ainda mais avançados e eficientes.

# 2.3 CSS

CSS (Cascading Style Sheets) é uma linguagem de estilo utilizada para a estilização de páginas web e, assim como o HTML, também é utilizada no desenvolvimento de softwares.

No desenvolvimento de software, o CSS é utilizado para a definição da aparência visual da interface do usuário (UI), permitindo a personalização de cores, fontes, layout, tamanho de fonte e outros elementos visuais.

Com o CSS, é possível criar uma aparência coesa e consistente em todo o software, garantindo que a interface do usuário seja fácil de usar e agradável aos olhos.

O CSS também permite a criação de interfaces responsivas, que se ajustam automaticamente a diferentes tamanhos de tela e dispositivos, tornando a experiência do usuário mais intuitiva e amigável.

Além disso, o CSS pode ser utilizado em conjunto com outras tecnologias, como o HTML e o JavaScript, para criar animações e efeitos visuais, tornando o software ainda mais atraente e dinâmico.

Logo CSS é uma ferramenta importante no desenvolvimento de software, permitindo a personalização da aparência visual da interface do usuário, tornando-a mais agradável e intuitiva. Combinado com outras tecnologias, é possível criar softwares ainda mais avançados e atraentes.

# 2.4 JavaScript

JavaScript é uma linguagem de programação utilizada no desenvolvimento de software para a criação de funcionalidades dinâmicas na interface do usuário (UI). É uma das tecnologias mais populares e amplamente utilizadas no desenvolvimento de software atualmente.

No desenvolvimento de software, o JavaScript é utilizado para a criação de funções que podem ser executadas quando o usuário interage com a interface, como clicar em botões, preencher formulários ou navegar em menus. Com o JavaScript, é possível criar interfaces dinâmicas e responsivas, que se ajustam automaticamente às ações do usuário.

O JavaScript também é utilizado para a validação de formulários e para o envio de dados para servidores, tornando-o uma ferramenta essencial no desenvolvimento de aplicações web.

Além disso, o JavaScript pode ser utilizado em conjunto com outras tecnologias, como o HTML e o CSS, para criar animações e efeitos visuais, tornando a interface do usuário ainda mais atraente e dinâmica.

Com a popularidade crescente do JavaScript, surgiram muitos frameworks e bibliotecas que facilitam o desenvolvimento de software, como o React, Vue.js e AngularJS.

O JavaScript é uma linguagem de programação essencial no desenvolvimento de software, permitindo a criação de interfaces dinâmicas, responsivas e funcionais. Combinado com outras tecnologias, é possível criar softwares ainda mais avançados e sofisticados.

# 2.5 React Native

React Native é um framework de desenvolvimento de aplicativos móveis que permite a criação de aplicativos para iOS e Android usando a linguagem JavaScript. Com o React Native, os desenvolvedores podem criar aplicativos móveis nativos de alto desempenho com uma única base de código.

Uma das principais vantagens do React Native é a sua capacidade de reutilizar o código existente, permitindo que os desenvolvedores criem aplicativos móveis mais rapidamente. Além disso, o React Native permite o desenvolvimento de interfaces de usuário avançadas e personalizadas, com alta velocidade e desempenho.

Outra vantagem do React Native é a sua comunidade ativa e crescente, que contribui com novos componentes e bibliotecas, tornando o desenvolvimento de aplicativos móveis ainda mais fácil e rápido.

# 2.6 Descrições sobre o banco de dados

Nosso banco de dados armazena as informações que são importante para que não haja alguma perca ou danos de dados sensíveis como dados do cliente e do contratante.

# 2.7 MySQL

MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto que utiliza a linguagem SQL (Structured Query Language) para manipulação de dados. É um dos sistemas de gerenciamento de banco de dados mais populares do mundo, sendo amplamente utilizado em aplicações web e empresariais.

Uma das principais vantagens do MySQL é a sua facilidade de uso e instalação, além da sua escalabilidade e segurança. Ele também oferece recursos avançados, como a capacidade de manipulação de dados em tempo real, o controle de transações e a replicação de dados, tornando-o uma escolha popular para aplicações que exigem alto desempenho e confiabilidade.

# CAPÍTULO – 3 DIAGRAMAS

Diagramas são representações gráficas de informações, conceitos, processos, fluxos ou sistemas. Eles são amplamente utilizados em diversas áreas, como engenharia, arquitetura, negócios, tecnologia da informação e outros campos que exigem uma representação visual de informações complexas.

Os diagramas podem ser usados para simplificar informações complexas, facilitar a compreensão de processos, identificar problemas e oportunidades de melhoria, bem como planejar e visualizar projetos e sistemas. Existem diversos tipos de diagramas, incluindo diagramas de fluxo, diagramas de rede, diagramas de causa e efeito, diagramas de Gantt, diagramas de sequência, entre outros.

# 3.1 Diagrama de Caso de Uso

Diagramas de caso de uso são uma técnica da engenharia de software para modelar as interações entre um sistema e seus usuários. Eles são usados para descrever os requisitos funcionais de um sistema, identificar as necessidades dos usuários e as tarefas que o sistema deve executar para atender a essas necessidades.

Um diagrama de caso de uso é composto por atores (usuários do sistema ou outros sistemas externos) e casos de uso (tarefas ou ações que o sistema deve realizar). Os atores representam os usuários que interagem com o sistema, enquanto os casos de uso descrevem as interações entre o sistema e seus usuários.

# CDU01 – FUNCIONAMENTO DO SOFTWARE (diagrama de sequência)

**Diagrama

Descrição gerada automaticamente**

# CDU02 – Cadastro de cliente (Caso de uso)

**Diagrama

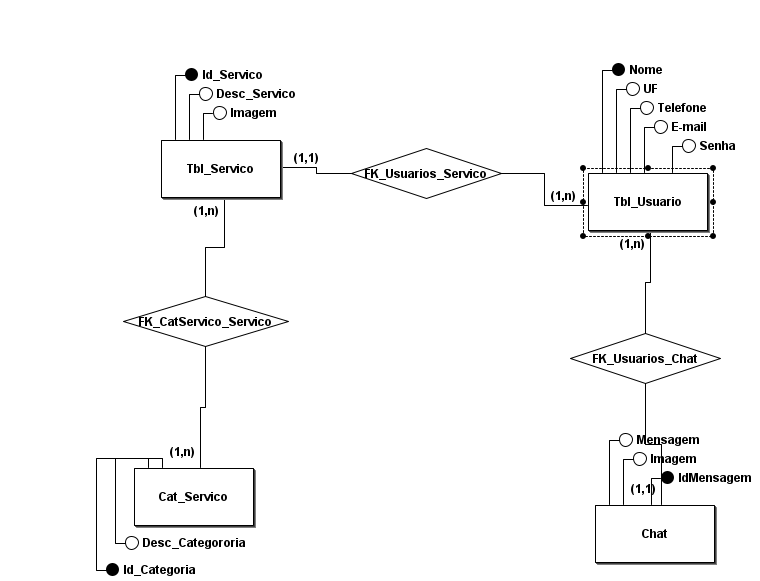
Descrição gerada automaticamente**

# 3.2 Diagrama de Entidade e Relacionamento

Diagramas de Entidade e Relacionamento (ER) são uma técnica para modelar dados em um sistema de informação. Eles representam entidades (objetos ou conceitos) e relacionamentos entre essas entidades.

Um diagrama ER é composto por entidades, que representam objetos ou conceitos do mundo real, e relacionamentos, que descrevem como essas entidades estão relacionadas. Os relacionamentos podem ser de diferentes tipos, como um-para-um, um-para-muitos ou muitos-para-muitos.

Os diagramas ER são importantes porque ajudam a garantir que o design do banco de dados atenda aos requisitos do sistema, facilitando a comunicação entre os desenvolvedores e os usuários finais. Eles também ajudam a identificar possíveis problemas de modelagem de dados e a definir os requisitos para a implementação do banco de dados.



# 3.3.1 Diagrama EER

Diagramas EER (Enhanced Entity-Relationship) são uma extensão dos Diagramas ER (Entidade-Relacionamento) que permitem modelar conceitos mais complexos e relacionamentos entre entidades em um sistema de informação.

Os diagramas EER adicionam novos conceitos, como herança, especialização e generalização, para permitir que o modelo de dados seja mais preciso e expressivo. Por exemplo, é possível criar uma entidade mais abstrata que outras entidades podem herdar atributos ou criar um relacionamento especializado que só pode ocorrer entre um subconjunto específico de entidades.

Os diagramas EER são úteis para modelar sistemas de informação mais complexos, como sistemas de gerenciamento de inventário ou sistemas de gerenciamento de relacionamentos com clientes. Eles permitem uma modelagem mais precisa e eficiente dos dados, ajudando a evitar problemas de redundância e inconsistência de dados no banco de dados.

**Interface gráfica do usuário

Descrição gerada automaticamente**