Dot

**Versão 3.0**

### Autores:Silvio Paulo,Guilherme Rafael

### Jose Roberto,Guilherme Magno

### Revisor: Prof. Carlos Elicier

## ÍNDICE DETALHADO

[**PREFÁCIO 4**](#_gjdgxs)

1. [**INTRODUÇÃO AO DOCUMENTO 6**](#_30j0zll)
   1. [TEMA 6](#_1fob9te)
   2. [OBJETIVO DO PROJETO 6](#_3znysh7)
   3. [DELIMITAÇÃO DO PROBLEMA 6](#_2et92p0)
   4. [JUSTIFICATIVA DA ESCOLHA DO TEMA 6](#_tyjcwt)
   5. [MÉTODO DE TRABALHO 6](#_3dy6vkm)
   6. [ORGANIZAÇÃO DO TRABALHO 7](#_1t3h5sf)
   7. [GLOSSÁRIO 7](#_4d34og8)
2. [**DESCRIÇÃO GERAL DO SISTEMA 8**](#_2s8eyo1)
   1. [DESCRIÇÃO DO PROBLEMA 8](#_17dp8vu)
   2. [PRINCIPAIS ENVOLVIDOS E SUAS CARACTERÍSTICAS 8](#_3rdcrjn)
   3. [REGRAS DE NEGÓCIOG 8](#_26in1rg)
3. [**REQUISITOS G DO SISTEMA 9**](#_lnxbz9)
   1. [REQUISITOS FUNCIONAIS 9](#_35nkun2)
   2. [REQUISITOS NÃO-FUNCIONAIS 10](#_1ksv4uv)
   3. [PROTÓTIPO 10](#_44sinio)
   4. [MÉTRICAS E CRONOGRAMA 11](#_2jxsxqh)
4. **ANÁLISE E *DESIGN* 12**
   1. [ARQUITETURA DO SISTEMA 12](#_z337ya)
   2. [MODELO DO DOMÍNIO 12](#_3j2qqm3)
   3. [DIAGRAMAS DE INTERAÇÃO 13](#_1y810tw)
   4. [DIAGRAMA DE CLASSES 14](#_4i7ojhp)
   5. [DIAGRAMA DE ATIVIDADES 14](#_2xcytpi)
   6. [DIAGRAMA DE ESTADOS 14](#_1ci93xb)
   7. [DIAGRAMA DE COMPONENTES 15](#_3whwml4)
   8. [MODELO DE DADOS 16](#_2bn6wsx)
      1. [*Modelo Lógico da Base de Dados 16*](#_qsh70q)
      2. [*Criação Física do Modelo de Dados 16*](#_3as4poj)
      3. [*Dicionário de Dados 16*](#_1pxezwc)
   9. [AMBIENTE DE DESENVOLVIMENTO 16](#_49x2ik5)
   10. [SISTEMAS E COMPONENTES EXTERNOS UTILIZADOS 16](#_2p2csry)
5. [**IMPLEMENTAÇÃO 17**](#_147n2zr)
6. [**TESTES 18**](#_3o7alnk)
   1. [PLANO DE TESTES 18](#_23ckvvd)
   2. [EXECUÇÃO DO PLANO DE TESTES 18](#_ihv636)
7. [**IMPLANTAÇÃO 19**](#_32hioqz)
   1. [DIAGRAMA DE IMPLANTAÇÃO 19](#_1hmsyys)
   2. [MANUAL DE IMPLANTAÇÃO 19](#_41mghml)
8. [**MANUAL DO USUÁRIO 20**](#_2grqrue)
9. [**CONCLUSÕES E CONSIDERAÇÕES FINAIS 21**](#_vx1227)

[**BIBLIOGRAFIA 22**](#_3fwokq0)

## Prefácio

Prefácio: Calculadora de Quantidade de Cerâmicas

Bem-vindo ao sistema de Calculadora de Quantidade de Cerâmicas, uma ferramenta projetada para simplificar e otimizar o processo de determinar a quantidade necessária de cerâmicas para revestir seus espaços de maneira eficiente. Este sistema foi concebido para atender às necessidades de proprietários, arquitetos, construtores e todos os envolvidos em projetos de construção e reforma.

Motivação:

A ideia de criar esta calculadora surgiu da necessidade de oferecer uma solução prática para um desafio comum na construção civil: calcular a quantidade precisa de cerâmicas para revestir um cômodo. Tradicionalmente, esse cálculo envolve uma série de medições e fórmulas, podendo ser propenso a erros humanos. Este sistema visa simplificar esse processo, proporcionando uma abordagem intuitiva e precisa.

Principais Recursos:

- Cadastro de Cômodos: Registre os detalhes de cada espaço que será revestido, facilitando a organização e referência futura.

- Cálculo Preciso: Insira as medidas do cômodo e deixe o sistema realizar os cálculos complexos para determinar a quantidade ideal de cerâmicas.

- Histórico de Cálculos: Mantenha um registro completo de todos os cálculos realizados, proporcionando um histórico útil para futuras referências e comparações.

- Interface Amigável: Uma interface de usuário intuitiva, projetada para ser acessível a todos os usuários, independentemente de sua experiência técnica.

Como Usar Este Manual:

Este manual foi elaborado para fornecer orientações claras sobre a instalação, configuração e utilização eficaz da Calculadora de Quantidade de Cerâmicas. Cada seção aborda aspectos específicos do sistema, desde a configuração inicial até a execução prática dos cálculos.

Feedback e Aperfeiçoamento:

Este sistema é uma iniciativa em constante evolução. Valorizamos profundamente o feedback dos usuários para melhorar continuamente a experiência e a funcionalidade. Caso tenha sugestões, dúvidas ou encontre qualquer dificuldade, encorajamos você a entrar em contato conosco por meio dos canais de suporte fornecidos neste manual.

Esperamos que esta Calculadora de Quantidade de Cerâmicas simplifique seus projetos de revestimento e contribua para um processo de construção mais eficiente e satisfatório.

Agradecemos por escolher a nossa ferramenta e desejamos a você muito sucesso em seus empreendimentos.

## Introdução ao Documento

Calculadora de Quantidade de Cerâmicas, uma ferramenta prática e eficiente para facilitar o processo de planejamento e aquisição de revestimentos cerâmicos para os seus ambientes. Este programa foi desenvolvido para simplificar o desafiador processo de estimar a quantidade de cerâmica necessária com base nas dimensões de seus cômodos.

### Tema

Ecommerce com diferencial de uma ferramenta de calculo para construção civil.

### Objetivo do Projeto

O objetivo principal da Calculadora de Quantidade de Cerâmicas é fornecer uma ferramenta eficaz para simplificar e otimizar o processo de determinar a quantidade necessária de cerâmicas para revestir espaços residenciais e comerciais. Vamos destacar alguns dos objetivos específicos do sistema:

1. Facilitar Cálculos Precisos: Eliminar a complexidade dos cálculos manuais, proporcionando uma maneira eficiente de determinar a quantidade exata de cerâmicas necessárias para cobrir um determinado espaço, levando em consideração as medidas específicas do cômodo.

2. Reduzir Erros Humanos: Minimizar a possibilidade de erros humanos associados aos cálculos manuais, proporcionando uma abordagem automatizada que leva em conta as fórmulas e medidas necessárias.

3. Aumentar Eficiência em Projetos de Construção e Reforma: Agilizar o processo de planejamento em projetos de construção e reforma, permitindo que arquitetos, construtores e proprietários tenham uma estimativa precisa da quantidade de cerâmicas necessárias, otimizando assim a gestão de materiais.

4. Registro e Histórico: Manter um histórico detalhado dos cálculos realizados para cada cômodo, permitindo que os usuários revisitem e comparem estimativas, além de fornecer uma referência útil para futuros projetos.

5. Interface Amigável: Oferecer uma interface de usuário intuitiva e amigável, tornando a ferramenta acessível a uma variedade de usuários, independentemente de sua experiência técnica.

6. Configuração Personalizada: Permitir a personalização de configurações, considerando diferentes padrões de assentamento de cerâmicas, tamanhos e formatos, para atender às preferências específicas do usuário.

7. Suporte a Decisões de Design: Auxiliar arquitetos e designers de interiores ao fornecer uma estimativa rápida e precisa da quantidade de cerâmicas necessárias, facilitando as decisões de design e escolhas de materiais.

Em resumo, o objetivo final é tornar o processo de determinar a quantidade de cerâmicas uma tarefa mais eficiente, precisa e acessível para todos os envolvidos em projetos de construção e reforma.

### Delimitação do Problema

Esta documentação visa orientá-lo no uso eficiente desta ferramenta, fornecendo informações detalhadas sobre a instalação, operação, e personalização do programa. Se você busca otimizar o processo de planejamento de seu projeto de revestimento cerâmico, está no lugar certo

### Justificativa da Escolha do Tema

Ao longo dos anos, a escolha e compra de cerâmicas para pisos e paredes tornaram-se uma parte essencial do design de interiores. No entanto, muitas vezes, calcular a quantidade exata de cerâmica necessária pode ser uma tarefa complexa e sujeita a erros. É aqui que nossa Calculadora de Quantidade de Cerâmicas entra em ação, proporcionando uma solução intuitiva e confiável para suas necessidades.

### Método de Trabalho

para este trabalho foi usado linguagens da programação como Nodejs, html e css ja como banco de dados foi usado o MongoDB

### Organização do Trabalho

1. Capa:

- Título: "Documentação da Calculadora de Quantidade de Cerâmicas"

- Logotipo ou Identificação do Produto

- Data de Atualização

2. Índice:

- Lista de Seções e Subseções com Números de Página Correspondentes

3. Introdução:

- Apresentação do Ecommerce e calculadora

- Finalidade e Benefícios

- Público-Alvo

4. Requisitos do Sistema:

- Especificações de Hardware

- Requisitos de Software

- Navegadores Suportados (para versões online)

5. Instalação (se aplicável):

- Instruções para Instalação do Software

- Configurações Iniciais

6. Como Usar:

- Instruções Passo a Passo para Utilizar a Calculadora

- Entrada de Dados

- Configurações Adicionais

7. Fórmula de Cálculo:

- Explicação Detalhada da Fórmula Utilizada

- Exemplos de Aplicação

8. Formato de Saída:

- Descrição do Formato dos Resultados

- Exemplos de Resultados

9. Mensagens de Erro:

- Lista de Mensagens de Erro Possíveis

- Explicações e Soluções

10. Personalização:

- Opções de Personalização Disponíveis

- Configurações e Preferências

11. Exemplos Práticos:

- Demonstração Prática do Uso da Calculadora

- Cenários Diferentes com Resultados

12. FAQ - Perguntas Frequentes:

- Lista de Perguntas Frequentes e Respostas

13. Suporte e Contato:

- Informações de Contato para Suporte Técnico

- Links para Recursos Adicionais

14. Agradecimentos e Notas Finais:

- Agradecimento ao Usuário por Utilizar a Calculadora

- Informações sobre Atualizações Futuras

15. Glossário (se necessário):

- Definições de Termos Técnicos Específicos

16. Anexos (se necessário):

- Qualquer Material Adicional, como Diagramas, Capturas de Tela Detalhadas, etc.

### Organização do Trabalho

1. Introdução

- 1.1 Objetivo do Sistema

- 1.2 Contexto do Sistema

- 1.3 Público-Alvo

2. Visão Geral do Sistema

- 2.1 Principais Recursos

- 2.2 Benefícios

- 2.3 Limitações

3. Requisitos do Sistema

- 3.1 Requisitos Funcionais

- 3.2 Requisitos Não-Funcionais

- 3.3 Restrições

4. Instalação e Configuração

- 4.1 Pré-requisitos

- 4.2 Configuração do Ambiente

- 4.3 Transferência de Código

- 4.4 Migração do Banco de Dados

- 4.5 Configuração de Variáveis de Ambiente

- 4.6 Testes de Instalação

5. Como Usar

- 5.1 Cadastro de Cômodos

- 5.2 Cálculo de Quantidade de Cerâmicas

- 5.3 Histórico de Cálculos

- 5.4 Configurações da Conta

- 5.5 Fluxo de Uso (diagrama de casos de uso)

6. Fórmula de Cálculo

- 6.1 Detalhes da Fórmula

- 6.2 Aplicações Práticas

7. Formato de Saída

- 7.1 Estrutura dos Resultados

- 7.2 Exportação de Dados

8. Mensagens de Erro

- 8.1 Lista de Mensagens

- 8.2 Resolução de Problemas Comuns

9. Personalização

- 9.1 Configurações Personalizadas

- 9.2 Extensibilidade do Sistema

10. Exemplos

- 10.1 Exemplo de Uso

- 10.2 Cenários Práticos

11. FAQ (Perguntas Frequentes)

- 11.1 Perguntas Comuns

- 11.2 Respostas Detalhadas

12. Bibliografia

- 12.1 Recursos de Desenvolvimento

- 12.2 Referências Técnicas

13. Histórico de Revisões

- 13.1 Alterações Recentes

- 13.2 Versões Anteriores

14. Prefácio

- 14.1 Objetivo e Motivação

- 14.2 Agradecimentos

### Glossário

1. Cômodo:

- Definição: Um espaço dentro de uma residência ou edifício que será revestido com cerâmica.

- Exemplo: Sala, quarto, cozinha ou a casa toda, caso seja usado uma cerâmica específica para o piso.

2. Dimensões do Cômodo:

- Definição: As medidas que determinam a largura e comprimento do cômodo onde a cerâmica será instalada.

- Exemplo: Largura: 4 metros, Comprimento: 5 metros.

3. Cerâmica:

- Definição: Material de revestimento utilizado em pisos e paredes, geralmente fabricado em forma de azulejos ou placas.

- Exemplo: Azulejos, porcelanato.

4. Dimensões da Cerâmica:

- Definição: As medidas que determinam o comprimento e a largura de uma peça de cerâmica.

- Exemplo: Comprimento: 0,2 metros, Largura: 0,2 metros.

5. Área do Cômodo:

- Definição: A medida total da superfície do cômodo a ser revestida, calculada multiplicando a largura pelo comprimento.

- Exemplo: Área = Largura x Comprimento.

6. Área de uma Cerâmica:

- Definição: A medida da superfície coberta por uma única peça de cerâmica, calculada multiplicando o comprimento pela largura da cerâmica.

- Exemplo: Área = Comprimento x Largura da Cerâmica.

7. Fator de Reserva:

- Definição: Uma margem percentual adicionada ao cálculo para cobrir perdas durante a instalação, cortes necessários e quebras.

- Exemplo: Fator de Reserva = 10%.

8. Quantidade de Cerâmica:

- Definição: O número estimado de peças de cerâmica necessárias para cobrir o cômodo, considerando as dimensões e o fator de reserva.

- Exemplo: 300 peças.

9. Versão Online/Instalável:

- Definição: Indica se a calculadora está disponível para uso online através de um navegador web ou se precisa ser instalada localmente em um dispositivo.

- Exemplo: Versão Online acessível em [insira-aqui-o-site.com].

10. Configurações Adicionais:

- Definição: Opções específicas que o usuário pode ajustar para personalizar o cálculo, como tamanho padrão da cerâmica, quantidade por caixa, etc.

- Exemplo: Tamanho padrão da cerâmica = 0,2 x 0,2 metros.

## Descrição Geral do Sistema

A Calculadora de Quantidade de Cerâmicas é uma ferramenta eficaz e intuitiva projetada para facilitar o planejamento e a execução de projetos de revestimento cerâmico em ambientes residenciais ou comerciais. Seu principal objetivo é fornecer aos usuários uma estimativa precisa da quantidade de cerâmica necessária com base nas dimensões do cômodo, dimensões da cerâmica e fator de reserva.

Principais Características:

1. Entrada de Medidas do Cômodo:

- Permite aos usuários inserir as dimensões do cômodo, como largura e comprimento, de forma fácil e intuitiva.

2. Configurações Adicionais (Opcional):

- Oferece opções para configurar parâmetros adicionais, como tamanho padrão da cerâmica, quantidade por caixa e outras configurações específicas do projeto.

3. Fórmula de Cálculo Automático:

- Utiliza uma fórmula eficiente para calcular a quantidade estimada de cerâmica com base na área do cômodo, área da cerâmica e um fator de reserva ajustável.

4. Formato de Saída Claro:

- Apresenta os resultados em um formato fácil de entender, destacando as dimensões do cômodo, as características da cerâmica e a quantidade recomendada.

5. Mensagens de Erro Intuitivas:

- Fornece mensagens de erro claras e explicativas em casos de entradas inválidas ou problemas durante o cálculo.

6. Opções de Personalização:

- Oferece opções de personalização, permitindo aos usuários ajustar unidades de medida, fatores de reserva e outras configurações para atender às suas necessidades específicas.

7. Interface Amigável:

- Apresenta uma interface intuitiva e amigável, garantindo que usuários com diferentes níveis de experiência possam utilizar a calculadora de maneira eficiente.

8. Versatilidade de Uso:

- Pode ser utilizada em diversos contextos, desde projetos residenciais simples até projetos comerciais mais complexos, adaptando-se às necessidades variadas dos usuários.

Benefícios:

- Economia de Tempo e Recursos:

- Evita estimativas imprecisas e auxilia na compra precisa de materiais, evitando desperdícios.

- Facilidade de Planejamento:

- Simplifica o processo de planejamento de projetos de revestimento cerâmico, tornando-o acessível mesmo para usuários sem experiência técnica.

- Minimização de Erros:

- Reduz a probabilidade de erros de cálculo, garantindo uma estimativa mais precisa da quantidade de cerâmica necessária.

- Versatilidade e Flexibilidade:

- Adapta-se a diferentes tipos de cerâmica, unidades de medida e requisitos específicos do projeto.

A Calculadora de Quantidade de Cerâmicas é uma ferramenta valiosa para profissionais da construção, designers de interiores e proprietários que buscam otimizar o processo de planejamento e execução de projetos de revestimento cerâmico, proporcionando resultados precisos e confiáveis.

## Descrição do Problema

Durante o desenvolvimento da Calculadora de Quantidade de Cerâmicas, alguns desafios e problemas podem surgir. A identificação e resolução desses problemas são cruciais para garantir um software funcional e eficiente. Aqui estão alguns problemas que podem ser encontrados durante o desenvolvimento e suas possíveis soluções:

1. Validação de Entrada:

- Problema: Garantir que as entradas do usuário sejam válidas e correspondam às expectativas.

- Solução: Implementar verificações de validação robustas para garantir que as dimensões do cômodo, cerâmica e outros parâmetros estejam dentro dos limites aceitáveis.

2. Lidando com Diferentes Unidades de Medida:

- Problema: Permitir que o usuário escolha entre diferentes unidades de medida para as dimensões do cômodo e da cerâmica.

- Solução: Implementar uma opção de seleção de unidade e realizar as conversões necessárias nos cálculos internos.

3. Tratamento de Erros:

- Problema: Como lidar com erros durante a execução, como entradas inválidas, falhas de cálculo, ou falta de conexão em versões online.

- Solução: Implementar mensagens de erro claras, fornecendo orientações sobre como corrigir o problema. Também é importante realizar testes extensivos para identificar possíveis cenários de erro.

4. Eficiência do Cálculo:

- Problema: Garantir que o cálculo seja eficiente, especialmente ao lidar com grandes áreas ou grandes conjuntos de dados.

- Solução: Otimizar algoritmos e estruturas de dados para garantir que a calculadora seja rápida e responsiva, mesmo para cenários mais complexos.

5. Configurações Personalizadas:

- Problema: Permitir a personalização das configurações, sem comprometer a simplicidade de uso.

- Solução: Projetar uma interface intuitiva que ofereça opções avançadas para usuários que desejam personalizar configurações, mantendo a simplicidade para usuários iniciantes.

6. Manutenção e Atualizações:

- Problema: Gerenciar efetivamente a manutenção do software e fornecer atualizações quando necessário.

- Solução: Estabelecer um plano de manutenção que inclua correção de bugs, melhorias de desempenho e atualizações para atender a novas demandas e requisitos.

7. Testes Adequados:

- Problema: Garantir uma cobertura de teste completa para evitar problemas não detectados.

- Solução: Implementar testes unitários, testes de integração e testes de aceitação do usuário para garantir uma cobertura abrangente durante o desenvolvimento.

8. Adaptação a Diferentes Ambientes e Plataformas:

- Problema: Garantir que a calculadora funcione corretamente em diferentes ambientes, como sistemas operacionais e navegadores web.

- Solução: Testar em uma variedade de ambientes para garantir compatibilidade e ajustar o código conforme necessário.

### Principais Envolvidos e suas Características

Neste item deve ser descrito o problema que será resolvido com o desenvolvimento do sistema. As questões a seguir devem ser respondidas.

* + - Quem é afetado pelo sistema?
    - Qual é o impacto do sistema?
    - Qual seria uma boa solução para o problema?

### Principais Envolvidos e suas Características

#### Usuários do Sistema

Os usuários potenciais da Calculadora de Quantidade de Cerâmicas podem incluir uma variedade de profissionais e indivíduos envolvidos em projetos de construção e reforma. Abaixo estão alguns dos principais usuários que podem se beneficiar do sistema:

1. Arquitetos e Designers de Interiores:

- Como Usuários: Podem utilizar a calculadora para obter estimativas precisas da quantidade de cerâmicas necessárias para os espaços que estão projetando.

- Benefícios: Facilita a tomada de decisões de design, ajudando a escolher a quantidade adequada de cerâmicas para alcançar o efeito desejado.

2. Proprietários e Clientes:

- Como Usuários: Podem usar a calculadora para ter uma compreensão prévia dos custos e da quantidade de material necessário antes de iniciar um projeto de reforma.

- Benefícios: Permite o planejamento financeiro antecipado e uma compreensão mais clara dos requisitos de material.

3. Construtores e Empreiteiros:

- Como Usuários: Podem incorporar a calculadora em suas práticas para realizar estimativas rápidas e precisas durante o planejamento de projetos de construção.

- Benefícios: Facilita a gestão eficiente de materiais, evitando a compra excessiva ou insuficiente de cerâmicas.

4. Profissionais de Vendas de Materiais de Construção:

- Como Usuários: Podem utilizar a calculadora como uma ferramenta de vendas, auxiliando os clientes a determinar a quantidade correta de cerâmicas a serem adquiridas.

- Benefícios: Melhora a experiência do cliente, fornecendo informações precisas sobre os produtos necessários.

5. Estudantes de Arquitetura e Engenharia:

- Como Usuários: Podem utilizar a calculadora como uma ferramenta educacional para entender os cálculos envolvidos na determinação da quantidade de cerâmicas.

- Benefícios: Auxilia no aprendizado prático e na compreensão dos conceitos envolvidos em projetos de construção.

6. Profissionais de Facilities Management:

- Como Usuários: Podem usar a calculadora para estimar a quantidade de cerâmicas necessárias para a manutenção e reforma de espaços existentes.

- Benefícios: Facilita o planejamento de custos e a execução eficiente de projetos de manutenção.

7. Profissionais de Varejo de Materiais de Construção:

- Como Usuários: Podem integrar a calculadora em seus sistemas de venda para fornecer aos clientes uma estimativa precisa durante as compras.

- Benefícios: Melhora a experiência do cliente, resultando em transações mais informadas.

#### Desenvolvedores do Sistema

1. **Silvio Paulo**

Líder de Projeto:

- Responsabilidades:

- Supervisão geral do projeto.

- Definição de requisitos e funcionalidades.

- Desenvolvimento da lógica principal da calculadora.

- Coordenação e distribuição de tarefas.

- Área de Atuação:

- Desenvolvimento de software, liderança técnica e coordenação de equipe.

2. **Guilherme Rafael**

Desenvolvedor Front-End e Designer de Interface:

- Responsabilidades:

- Implementação da interface do usuário.

- Desenvolvimento do design visual da calculadora.

- Integração de elementos de usabilidade.

- Garantir uma experiência do usuário intuitiva.

- Área de Atuação:

- Desenvolvimento web, design de interface, usabilidade.

3. **Guilherme Magno**

Desenvolvedor Back-End e Especialista em Cálculos:

- Responsabilidades:

- Implementação da lógica de cálculo da calculadora.

- Gerenciamento de dados e manipulação de entradas/saídas.

- Desenvolvimento da lógica de personalização e configuração.

- Área de Atuação:

- Desenvolvimento de lógica de aplicação, manipulação de dados.

4. **Jose Roberto**

Testador de Qualidade e Analista de Documentação:

- Responsabilidades:

- Desenvolvimento de casos de teste.

- Realização de testes funcionais e de usabilidade.

- Documentação do código, incluindo comentários.

### Regras de NegócioG

As regras de negócio são diretrizes essenciais que orientam o desenvolvimento e a operação de um sistema, garantindo que ele atenda aos requisitos e objetivos estabelecidos. Para a Calculadora de Quantidade de Cerâmicas, algumas regras de negócio relevantes podem incluir:

1. Validação de Entradas:

- Regra: Todas as entradas do usuário, como dimensões do cômodo e da cerâmica, devem ser validadas para garantir que estejam dentro dos limites aceitáveis.

2. Configurações Padrão e Opcionais:

- Regra: O sistema deve fornecer configurações padrão, mas também permitir que o usuário faça ajustes opcionais, como fator de reserva e unidades de medida.

3. Precisão nos Cálculos:

- Regra: A calculadora deve realizar cálculos precisos, levando em consideração a área do cômodo, a área de cada cerâmica e o fator de reserva, proporcionando estimativas confiáveis.

4. Tratamento de Erros:

- Regra: O sistema deve ser capaz de identificar e lidar com erros de entrada ou cálculo, fornecendo mensagens de erro claras e orientações para correção.

5. Experiência do Usuário Intuitiva:

- Regra: A interface do usuário deve ser intuitiva, fácil de usar e fornecer feedback claro durante todo o processo, garantindo uma experiência positiva para o usuário.

6. Adaptação a Diferentes Contextos:

- Regra: O sistema deve ser projetado para se adaptar a diferentes contextos, como tipos de cerâmica, unidades de medida e requisitos específicos do projeto.

7. Eficiência do Cálculo:

- Regra: A calculadora deve ser eficiente, produzindo resultados rapidamente mesmo em cenários com grandes áreas ou conjuntos de dados.

8. Atualizações e Manutenção:

- Regra: O sistema deve ser projetado para facilitar atualizações e manutenção contínua, garantindo que esteja alinhado com as necessidades em evolução.

9. Privacidade e Segurança:

- Regra: Garantir a segurança dos dados do usuário e respeitar as normas de privacidade durante a coleta e o armazenamento de informações.

10. Documentação Clara:

- Regra: Fornecer documentação clara, incluindo manuais do usuário e documentação técnica, para auxiliar tanto usuários quanto desenvolvedores.

11. Acessibilidade:

- Regra: O sistema deve ser desenvolvido considerando princípios de acessibilidade, garantindo que seja utilizável por pessoas com diferentes habilidades e necessidades.

Essas regras de negócio são fundamentais para assegurar que a Calculadora de Quantidade de Cerâmicas seja uma ferramenta eficaz, precisa e fácil de usar para seu público-alvo. Elas também contribuem para a confiabilidade, segurança e adaptabilidade do sistema ao longo do tempo.

## Requisitos G do Sistema

Para garantir uma experiência eficiente e sem problemas com a Calculadora de Quantidade de Cerâmicas, é essencial atender aos seguintes requisitos do sistema:

Sistema Operacional:

A calculadora é compatível com os sistemas operacionais Windows, MacOS e Linux.

Navegador Web:

Para a versão baseada na web, é recomendado o uso dos seguintes navegadores:

Google Chrome (versão mais recente)

Mozilla Firefox (versão mais recente)

Microsoft Edge (versão mais recente)

Certifique-se de que os cookies e o JavaScript estejam habilitados para garantir o funcionamento adequado da aplicação

Certifique-se de que seu sistema atenda a esses requisitos para garantir uma experiência ideal ao utilizar a Calculadora de Quantidade de Cerâmicas. Em caso de dúvidas ou dificuldades técnicas, consulte a seção de suporte ou entre em contato conosco para assistência.

### Requisitos Funcionais

1. Cadastro de Usuários (opcional):

- Permitir que os usuários criem contas para salvar configurações personalizadas ou histórico de projetos.

2. Entrada de Dimensões do Cômodo:

- Permitir que os usuários insiram as dimensões do cômodo, incluindo largura e comprimento.

3. Entrada de Dimensões da Cerâmica:

- Permitir que os usuários insiram as dimensões da cerâmica, incluindo comprimento e largura.

4. Configurações Adicionais (Opcional):

- Oferecer opções para configurações adicionais, como fator de reserva, unidades de medida e outras preferências do usuário.

5. Cálculo Automático da Quantidade de Cerâmica:

- Realizar cálculos automáticos com base nas dimensões do cômodo, dimensões da cerâmica e fator de reserva.

6. Formato de Saída Clara:

- Apresentar os resultados de maneira clara, mostrando a quantidade estimada de cerâmica necessária e outras informações relevantes.

7. Mensagens de Erro e Validação:

- Fornecer mensagens de erro claras para entradas inválidas ou situações de cálculo problemáticas, com orientações para correção.

8. Personalização de Resultados:

- Permitir que os usuários personalizem ou ajustem os resultados com base em suas necessidades específicas.

9. Histórico de Projetos (opcional):

- Se aplicável, manter um histórico de projetos para referência futura ou análise.

10. Atualizações Automáticas (opcional):

- Oferecer uma funcionalidade que permite ao sistema atualizar automaticamente para versões mais recentes, se necessário.

11. Interface Amigável:

- Desenvolver uma interface do usuário intuitiva e fácil de usar, facilitando a entrada de dados e a compreensão dos resultados.

12. Adaptação a Diferentes Tipos de Cerâmica:

- Projetar o sistema para ser flexível o suficiente para lidar com diferentes tipos de cerâmica, considerando variações nas dimensões e padrões.

13. Exportação de Resultados (opcional):

- Se necessário, permitir que os usuários exportem os resultados para facilitar a integração com outros sistemas ou para referência posterior.

14. Segurança de Dados:

- Garantir que os dados dos usuários sejam armazenados de maneira segura e que a privacidade seja respeitada.

15. Acessibilidade:

- Incorporar princípios de design acessível para garantir que o sistema seja utilizável por pessoas com diferentes habilidades e necessidades

### Requisitos Não-Funcionais

1. Desempenho:

- RNF1: A resposta da calculadora deve ser instantânea para garantir uma experiência de usuário ágil.

- RNF2: O sistema deve ser capaz de lidar eficientemente com múltiplos usuários simultâneos, garantindo escalabilidade.

2. Usabilidade:

- RNF3: A interface do usuário deve ser intuitiva e fácil de usar, mesmo para usuários sem experiência técnica.

- RNF4: A calculadora deve fornecer feedback claro sobre os resultados e quaisquer mensagens de erro.

3. Segurança:

- RNF5: Os dados dos usuários, especialmente informações pessoais e histórico de cálculos, devem ser armazenados de forma segura.

- RNF6: O sistema deve implementar práticas de segurança para proteger contra ataques, como injeção de código e ataques de negação de serviço.

4. Confiabilidade:

- RNF7: A calculadora deve ser robusta e resistente a falhas para garantir disponibilidade contínua.

- RNF8: O sistema deve ser capaz de recuperar automaticamente de falhas e interrupções.

5. Portabilidade:

- RNF9: A calculadora deve ser acessível em diferentes dispositivos e navegadores com consistência na experiência do usuário.

- RNF10: O sistema deve ser independente do sistema operacional para facilitar a implementação em diferentes ambientes.

6. Manutenibilidade:

- RNF11: O código-fonte deve ser bem documentado para facilitar a manutenção contínua.

- RNF12: Atualizações e correções devem ser implementadas de forma eficiente e com mínimo impacto nos usuários.

7. Interoperabilidade:

- RNF13: O sistema deve ser capaz de integrar-se a outras ferramentas e sistemas relevantes, como bancos de dados e APIs externas.

- RNF14: O formato de saída dos resultados deve ser compatível com padrões comuns para facilitar a interoperabilidade.

8. Eficiência de Recursos:

- RNF15: O sistema deve utilizar eficientemente os recursos do servidor e minimizar o consumo de largura de banda.

- RNF16: A calculadora deve ser otimizada para garantir um desempenho eficaz, mesmo em condições de largura de banda limitada.

### Protótipo

Tela de Início/Login:

- Elementos:

- Campo de entrada para o nome de usuário.

- Campo de entrada para a senha.

- Botão "Entrar".

- Link "Esqueceu a senha?" para recuperação de senha.

Tela de Cadastro de Cômodos:

- Elementos:

- Formulário para inserir as dimensões do cômodo (comprimento, largura, altura).

- Botão "Calcular" para iniciar o processo de cálculo.

- Seção para exibir informações sobre cômodos cadastrados anteriormente.

Tela de Resultados:

- Elementos:

- Exibição dos resultados do cálculo, incluindo a quantidade estimada de cerâmicas.

- Botão "Salvar Cômodo" para adicionar o cômodo ao histórico.

- Botão "Novo Cômodo" para retornar à tela de cadastro de cômodos.

Tela de Histórico de Cálculos:

- Elementos:

- Lista de cômodos previamente calculados.

- Opções para visualizar detalhes de cada cômodo.

- Botão "Excluir" para remover cômodos do histórico.

Tela de Configurações da Conta:

- Elementos:

- Opções para modificar configurações da conta, como nome de usuário e senha.

- Opção para fazer logout da conta.

Componentes de Navegação:

- Menu de Navegação:

- Links para cada seção do sistema (Cadastro de Cômodos, Histórico, Configurações, etc.).

- Indicação visual da seção atualmente ativa.

Considerações de Design:

- Design Responsivo:

- Garantir que o protótipo seja amigável em dispositivos móveis e tablets.

- Feedback Visual:

- Indicação visual clara de quando uma ação foi realizada com sucesso ou se ocorreu um erro.

- Fluxo Intuitivo:

- Organizar as telas de forma lógica, facilitando a navegação do usuário.

Fluxo de Uso:

1. Login/Registro: O usuário faz login ou se registra, sendo redirecionado para a tela principal.

2. Cadastro de Cômodos: O usuário acessa a tela de cadastro de cômodos, inserindo as dimensões do espaço.

3. Cálculo e Resultados: Após o cálculo, o usuário visualiza os resultados na tela dedicada.

4. Histórico: O usuário pode acessar o histórico para revisitar cálculos anteriores.

5. Configurações: Opção para ajustar configurações da conta.

.

### Métricas e Cronograma

Métricas:

1. Taxa de Uso da Calculadora:

- Definição: Percentual de usuários ativos em um período específico.

- Métrica: Número de cálculos realizados em relação ao total de usuários.

2. Taxa de Conversão (Registro de Usuários):

- Definição: Percentual de visitantes que se tornam usuários registrados.

- Métrica: Número de novos registros em relação ao número total de visitantes.

3. Tempo Médio de Sessão:

- Definição: Média de tempo que os usuários passam na aplicação em uma única sessão.

- Métrica: Tempo total gasto pelos usuários dividido pelo número de sessões.

4. Taxa de Retenção de Usuários:

- Definição: Percentual de usuários que retornam à aplicação após a primeira visita.

- Métrica: Número de usuários que retornam dividido pelo número total de usuários.

5. Taxa de Conclusão de Cálculos:

- Definição: Percentual de cálculos iniciados que são concluídos com sucesso.

- Métrica: Número de cálculos concluídos dividido pelo número de cálculos iniciados.

6. Número de Projetos Salvos (se implementado):

- Definição: Quantidade total de projetos salvos pelos usuários.

- Métrica: Número de projetos salvos em um determinado período.

Cronograma:

1. Fase de Planejamento (Semana 1-2):

- Identificação de requisitos.

- Análise de mercado e concorrência.

- Definição da arquitetura do sistema.

2. Fase de Projeto e Design (Semana 3-5):

- Desenvolvimento do protótipo da interface do usuário.

- Design detalhado do sistema e das funcionalidades.

- Estabelecimento da lógica de negócio.

3. Fase de Implementação (Semana 6-8):

- Desenvolvimento do front-end e back-end.

- Integração de funcionalidades.

- Implementação de testes unitários.

4. Fase de Testes (Semana 9-10):

- Realização de testes de funcionalidade.

- Testes de usabilidade e experiência do usuário.

- Identificação e correção de bugs.

5. Fase de Lançamento (Semana 11-12):

- Preparação da versão final do sistema.

- Lançamento da Calculadora de Quantidade de Cerâmicas.

- Monitoramento inicial de métricas.

6. Fase de Manutenção e Melhorias Contínuas (Após o Lançamento):

- Avaliação contínua de métricas e feedback dos usuários.

- Correção de bugs e problemas identificados.

- Implementação de melhorias e novas funcionalidades.

1. **Análise e *Design***

1. Identificação de Requisitos:

- Realize reuniões com os stakeholders para entender os requisitos do sistema.

- Identifique funcionalidades essenciais, como entrada de dimensões, cálculos, configurações adicionais e histórico de projetos.

2. Análise de Usuário:

- Compreenda o perfil dos usuários para adaptar a interface às suas necessidades.

- Considere casos de uso típicos, explorando como os usuários interagirão com a aplicação.

3. Análise de Competidores:

- Avalie outras calculadoras de materiais de construção no mercado.

- Identifique pontos fortes e fracos para incorporar melhorias e inovações.

4. Modelagem de Dados:

- Desenvolva modelos de dados para representar informações essenciais, como dimensões do cômodo, dimensões da cerâmica e histórico de projetos.

Design:

1. Design da Interface do Usuário (UI):

- Crie wireframes e protótipos para visualizar a interface do usuário.

- Certifique-se de que a navegação seja intuitiva e que os elementos de design estejam alinhados com a identidade visual desejada.

2. Design da Experiência do Usuário (UX):

- Projete uma experiência de usuário fluida, garantindo que o fluxo de interação seja lógico.

- Considere feedbacks visuais e sonoros para orientar o usuário durante o processo.

3. Arquitetura do Sistema:

- Determine a arquitetura do sistema, incluindo a estrutura do banco de dados e a divisão entre front-end e back-end.

- Escolha as tecnologias apropriadas para suportar os requisitos do sistema.

4. Lógica de Negócio:

- Defina a lógica por trás dos cálculos, validações e interações do usuário.

- Garanta que as regras de negócio estejam alinhadas com os requisitos identificados.

5. Segurança e Privacidade:

- Implemente medidas de segurança para proteger dados sensíveis.

- Considere práticas de privacidade para garantir conformidade com regulamentações.

6. Testes Planejados:

- Planeje a estratégia de testes, incluindo testes unitários, testes de integração e testes de aceitação do usuário.

- Antecipe cenários de teste, incluindo casos de uso típicos e excepcionais.

7. Escalabilidade:

- Projete o sistema para ser escalável, considerando o aumento potencial no número de usuários.

- Avalie a possibilidade de implementar recursos premium ou planos de assinatura.

8. Documentação:

- Elabore documentação abrangente, incluindo manuais do usuário e documentação técnica.

- Certifique-se de que os desenvolvedores futuros possam entender e dar continuidade ao projeto.

### Arquitetura do Sistema

1. Interface do Usuário (Front-end):

- Descrição: Responsável pela interação direta com o usuário.

- Tecnologias Possíveis:

- HTML, CSS, JavaScript para a construção da interface.

- Frameworks como Node, Vue.js ou Angular para desenvolvimento front-end dinâmico.

2. Lógica de Aplicação (Back-end):

- Descrição: Gerencia a lógica de negócio, manipula dados e responde a requisições do front-end.

- Tecnologias Possíveis:

- Node.js, Python (Django ou Flask), Java (Spring), ou Ruby on Rails.

- API RESTful para comunicação entre front-end e back-end.

3. Banco de Dados:

- Descrição: Armazena e recupera dados necessários para o funcionamento da aplicação.

- Tecnologias Possíveis:

- MySQL, PostgreSQL, MongoDB, ou SQLite, dependendo das necessidades de armazenamento e consulta.

4. Serviços Adicionais:

- Descrição: Componentes adicionais que fornecem funcionalidades específicas, como envio de e-mails para recuperação de senha, integração com serviços de terceiros, etc.

- Tecnologias Possíveis:

- Serviços de e-mail (SMTP), serviços de autenticação (OAuth), entre outros.

Fluxo de Funcionamento:

1. O front-end recebe as entradas do usuário, como as dimensões do cômodo, dimensões da cerâmica e configurações adicionais.

2. Essas entradas são enviadas para o back-end por meio de chamadas de API RESTful.

3. O back-end realiza a lógica de negócio, incluindo cálculos, validações e interações com o banco de dados.

4. Os resultados são processados e enviados de volta ao front-end, que os apresenta ao usuário de maneira clara e compreensível.

5. Se a funcionalidade de salvar projetos estiver implementada, o back-end também interage com o banco de dados para salvar e recuperar dados relacionados a projetos.

Considerações de Escalabilidade:

- A arquitetura deve ser projetada para ser escalável, permitindo a adição de mais recursos e acomodação de um número crescente de usuários.

- O uso de tecnologias e serviços baseados em nuvem pode facilitar a escalabilidade, permitindo dimensionar recursos conforme necessário.

Segurança:

- Implementação de medidas de segurança em todas as camadas para proteger dados sensíveis e garantir a integridade do sistema.

- Utilização de HTTPS para comunicação segura entre o front-end e o back-end.

Flexibilidade:

- A arquitetura deve ser flexível o suficiente para permitir futuras atualizações, incorporação de novas funcionalidades e adaptação a mudanças nos requisitos.

### Modelo do Domínio

1. Usuário:

- A entidade que representa um usuário da aplicação.

- Atributos: ID, nome de usuário, senha criptografada, endereço de e-mail.

2. Projeto:

- A entidade que armazena informações sobre um projeto específico da Calculadora de Quantidade de Cerâmicas.

- Atributos: ID do projeto, nome do projeto, data de criação.

3. Dimensões do Cômodo:

- A entidade que registra as dimensões do cômodo inseridas pelo usuário para um projeto.

- Atributos: ID das dimensões, largura, comprimento, altura.

4. Dimensões da Cerâmica:

- A entidade que armazena as dimensões da cerâmica inseridas pelo usuário para um projeto.

- Atributos: ID das dimensões da cerâmica, comprimento, largura.

5. Configurações Adicionais:

- A entidade que mantém configurações adicionais, como fator de reserva, selecionadas pelo usuário para um projeto.

- Atributos: ID das configurações, fator de reserva, unidades de medida.

A relação entre essas entidades pode ser representada da seguinte maneira:

- Um Usuário pode ter vários Projetos.

- Um Projeto pode ter uma ou mais instâncias de Dimensões do Cômodo, Dimensões da Cerâmica e Configurações Adicionais associadas.

### Diagramas de Interação

1. Usuário Inicia o Cálculo:

- O usuário insere as dimensões do cômodo, as dimensões da cerâmica e outras configurações desejadas na interface do usuário.

2. Front-end Envia Requisição ao Back-end:

- O front-end envia uma requisição ao back-end contendo as informações inseridas pelo usuário.

3. Back-end Realiza Cálculos:

- O back-end processa as informações recebidas, realiza os cálculos necessários (considerando fórmulas e lógica de negócio) e retorna os resultados ao front-end.

4. Front-end Exibe Resultados:

- O front-end exibe os resultados do cálculo ao usuário na interface gráfica.

![Diagrama de Interação para Cálculo de Cerâmicas](https://i.imgur.com/4zWpIOh.png)

Diagrama de Interação para Salvamento de Projeto:

1. Usuário Salva o Projeto:

- Após realizar um cálculo, o usuário opta por salvar o projeto, fornecendo um nome ao projeto.

2. Front-end Envia Requisição ao Back-end:

- O front-end envia uma requisição ao back-end com as informações do projeto a serem salvas.

3. Back-end Salva Projeto no Banco de Dados:

- O back-end processa a requisição, salva as informações do projeto no banco de dados (como dimensões do cômodo, dimensões da cerâmica, configurações adicionais e resultados) associadas ao usuário.

4. Confirmação de Salvamento ao Usuário:

- O front-end recebe a confirmação do back-end sobre o sucesso do salvamento e fornece feedback ao usuário.

#### Diagrama de Seqüência

1. Usuário Inicia o Cálculo:

- O usuário insere as dimensões do cômodo, dimensões da cerâmica e outras configurações na interface do usuário.

2. Front-end Solicita Cálculo ao Back-end:

- O front-end envia uma mensagem ao back-end solicitando o cálculo, incluindo as informações inseridas pelo usuário.

3. Back-end Recebe a Requisição:

- O back-end recebe a requisição e inicia o processamento.

4. Cálculos no Back-end:

- O back-end realiza os cálculos necessários com base nas informações recebidas.

5. Back-end Retorna Resultados ao Front-end:

- Após os cálculos, o back-end envia os resultados (quantidade de cerâmicas necessárias) de volta ao front-end.

6. Front-end Exibe Resultados ao Usuário:

- O front-end recebe os resultados e os apresenta na interface do usuário para que o usuário visualize as informações.

#### Diagrama de Colaboração / Comunicação

Neste diagrama, representaremos os objetos principais envolvidos: `Usuário`, `Front-end` e `Back-end`. As mensagens entre esses objetos indicarão a colaboração para realizar o cálculo de cerâmicas.

Descrição das Etapas:

1. Usuário Inicia o Cálculo:

- O `Usuário` interage com a interface do usuário fornecendo as dimensões do cômodo, dimensões da cerâmica e outras configurações.

2. Solicitação ao Back-end:

- A `Interface do Usuário` envia uma mensagem de solicitação ao `Back-end` com as informações inseridas pelo usuário.

3. Cálculos no Back-end:

- O `Back-end` processa as informações, realiza os cálculos necessários e prepara os resultados.

4. Resposta ao Front-end:

- O `Back-end` envia uma mensagem de resposta ao `Front-end` contendo os resultados calculados.

5. Exibição dos Resultados:

- A `Interface do Usuário` recebe os resultados e os exibe para o `Usuário`.

### Diagrama de Classes

Descrição das Classes:

1. Usuário:

- A classe que representa um usuário da aplicação.

- Atributos: `id`, `nome`, `email`, `senha`.

- Métodos: `criarProjeto()`, `salvarProjeto()`.

2. Projeto:

- A classe que representa um projeto criado pelo usuário.

- Atributos: `id`, `nome`, `dataCriacao`.

- Métodos: `adicionarDimensoes()`, `salvar()`.

3. Interface do Usuário:

- A classe que representa a interface com a qual o usuário interage.

- Atributo: `projetoAtual`.

- Métodos: `inserirDimensoes()`, `calcular()`, `exibirResultados()`.

4. Back-end:

- A classe que lida com a lógica de negócio e interage com o banco de dados.

- Métodos: `calcularCeramicas()`, `salvarProjetoNoBanco()`

### Diagrama de Atividades

1. Início:

- Atividade que marca o início do processo.

2. Usuário Insere Dimensões e Configurações:

- Atividade em que o usuário insere as dimensões do cômodo, as dimensões da cerâmica e outras configurações desejadas.

3. Verificar Validade das Entradas:

- Atividade de verificação para garantir que as entradas do usuário sejam válidas.

- Condição de Decisão:

- Se as entradas forem inválidas, o processo retorna para a atividade "Usuário Insere Dimensões e Configurações". Caso contrário, continua para a próxima atividade.

4. Calcular Quantidade de Cerâmicas:

- Atividade em que o sistema realiza os cálculos necessários com base nas informações inseridas pelo usuário.

5. Exibir Resultados ao Usuário:

- Atividade em que os resultados do cálculo são apresentados ao usuário na interface.

6. Salvar Projeto (Opcional):

- Atividade opcional em que o usuário pode optar por salvar o projeto.

- Condição de Decisão:

- Se o usuário escolher salvar o projeto, o processo continua para a atividade "Salvar Projeto no Banco de Dados". Caso contrário, o processo é concluído.

7. Salvar Projeto no Banco de Dados:

- Atividade em que o sistema salva as informações do projeto no banco de dados.

8. Fim:

- Atividade que marca o fim do processo.

### Diagrama de Estados

1. Novo:

- Estado inicial do projeto quando é criado.

- Transição: Pode mudar para os estados `Configurando` ou `Concluído` dependendo das ações do usuário.

2. Configurando:

- Estado em que o usuário está inserindo dimensões, dimensões da cerâmica e configurações adicionais.

- Transição: Pode voltar para o estado `Novo` se o usuário decidir começar de novo ou avançar para o estado `Calculando`.

3. Calculando:

- Estado em que o sistema realiza os cálculos com base nas informações inseridas pelo usuário.

- Transição: Pode voltar para o estado `Configurando` se o usuário desejar ajustar as entradas ou avançar para o estado `Exibindo Resultados`.

4. Exibindo Resultados:

- Estado em que os resultados do cálculo são apresentados ao usuário.

- Transição: Pode voltar para o estado `Configurando` se o usuário desejar fazer alterações ou avançar para os estados `Salvando Projeto` ou `Concluído`.

5. Salvando Projeto:

- Estado opcional em que o usuário decide salvar o projeto.

- Transição: Pode voltar para o estado `Exibindo Resultados` se o usuário optar por não salvar ou avançar para o estado `Concluído` após salvar.

6. Concluído:

- Estado final do projeto, onde o processo é considerado concluído.

### Diagrama de Componentes

```plaintext

+-------------------------+ +------------------------+

| Interface do Usuário | | Back-end |

|-------------------------| |------------------------|

| +TelaPrincipal | | +ServiçoCalculo |

| +TelaResultados | | +ServiçoBancoDeDados |

| +TelaConfiguracoes | +------------------------+

| ... |

+-------------------------+ +------------------------+

| |

| |

v v

+-------------------------+ +------------------------+

| Front-end | | Back-end |

|-------------------------| |------------------------|

| +Lógica de Interface | | +Lógica de Negócio |

| +Chamadas de API | | +Acesso ao Banco de |

| +Gerenciamento de Estado | | Dados |

| ... | | ... |

+-------------------------+ +------------------------+

| |

| |

v v

+-------------------------+ +------------------------+

| Bibliotecas e | | Banco de Dados |

| Frameworks | |------------------------|

|-------------------------| | - Projetos |

| - Node.js | | - Dimensoes |

| - Axios | | - Configuracoes |

| - Django (Python) | | ... |

| - Express (Node.js) | +------------------------+

| ... |

+-------------------------+

```

Descrição dos Componentes:

1. Interface do Usuário:

- Componente que fornece a interface gráfica com a qual o usuário interage.

- Inclui telas principais, de resultados, configurações, etc.

2. Front-end:

- Componente que lida com a lógica da interface do usuário.

- Gerencia o estado da aplicação, realiza chamadas de API para o back-end, etc.

3. Bibliotecas e Frameworks:

- Componente que inclui bibliotecas e frameworks utilizados no desenvolvimento do front-end.

- Exemplos: Node.js, Axios, Django, Express, etc.

4. Back-end:

- Componente responsável pela lógica de negócio e interação com o banco de dados.

5. Lógica de Negócio:

- Componente que contém a lógica específica para o cálculo de cerâmicas e outras operações.

6. Banco de Dados:

- Componente que armazena dados essenciais, como projetos, dimensões, configurações, etc.

7. Serviço Calculo:

- Componente que encapsula a lógica de cálculo de cerâmicas.

8. Serviço Banco de Dados:

- Componente que gerencia o acesso ao banco de dados.

### Modelo de Dados

#### Modelo Lógico da Base de Dados

Descrição das Tabelas:

1. Usuario:

- Armazena informações sobre os usuários da aplicação.

- Chave Primária: `id`.

2. Projeto:

- Contém detalhes sobre os projetos criados pelos usuários.

- Chave Primária: `id`.

- Chave Estrangeira: `id\_usuario` (referenciando `Usuario`).

3. DimensoesComodo:

- Armazena as dimensões do cômodo para cada projeto.

- Chave Primária: `id`.

- Chave Estrangeira: `id\_projeto` (referenciando `Projeto`).

4. DimensoesCeramica:

- Armazena as dimensões da cerâmica para cada projeto.

- Chave Primária: `id`.

- Chave Estrangeira: `id\_projeto` (referenciando `Projeto`).

Observações:

- Os tipos de dados (ex.: `decimal`, `string`) e as restrições de chave primária/estrangeira não estão representados, mas seriam parte da implementação real.

- Os dados nas tabelas são exemplos fictícios e podem variar de acordo com a entrada do usuário.

#### Criação Física do Modelo de Dados

```sql

-- Tabela Usuario

CREATE TABLE Usuario (

id INT PRIMARY KEY,

nome VARCHAR(255),

email VARCHAR(255),

senha ENCRYPTED(255)

);

CREATE TABLE Projeto (

id INT PRIMARY KEY,

nome VARCHAR(255),

dataCriacao DATETIME,

id\_usuario INT,

rua VARCHAR(255),

numero VARCHAR(10),

cidade VARCHAR(100),

estado VARCHAR(50),

cep VARCHAR(15),

FOREIGN KEY (id\_usuario) REFERENCES Usuario(id)

);

-- Tabela DimensoesComodo

CREATE TABLE DimensoesComodo (

id INT PRIMARY KEY,

largura DECIMAL(5,2),

comprimento DECIMAL(5,2),

altura DECIMAL(5,2),

id\_projeto INT,

FOREIGN KEY (id\_projeto) REFERENCES Projeto(id)

);

-- Tabela DimensoesCeramica

CREATE TABLE DimensoesCeramica (

id INT PRIMARY KEY,

comprimento DECIMAL(5,2),

largura DECIMAL(5,2),

id\_projeto INT,

FOREIGN KEY (id\_projeto) REFERENCES Projeto(id)

);

```

Observações:

- Os tipos de dados e as restrições de chave primária/estrangeira podem variar dependendo do SGBD utilizado.

- Os comandos acima são um exemplo geral; adapte-os conforme necessário para a implementação específica do seu sistema de banco de dados.

#### Dicionário de Dados

| Campo | Tipo | Descrição |

|-------|------|-----------------------------|

| id | INT | Chave primária |

| nome | VARCHAR(255) | Nome do usuário |

| email | VARCHAR(255) | Endereço de e-mail do usuário |

| senha | ENCRYPTED(30) | Senha do usuário (criptografada) |

Tabela: Projeto

| Campo | Tipo | Descrição |

|-------------|--------------|---------------------------------------------|

| id | INT | Chave primária |

| nome | VARCHAR(255) | Nome do projeto |

| dataCriacao | DATETIME | Data e hora de criação do projeto |

| id\_usuario | INT | Chave estrangeira referenciando Usuario(id)|

| rua | VARCHAR(255) | Nome da rua |

| numero | VARCHAR(10) | Número do endereço |

| cidade | VARCHAR(100) | Cidade do endereço |

| estado | VARCHAR(50) | Estado do endereço |

| cep | VARCHAR(15) | CEP (Código de Endereçamento Postal) |

Tabela: DimensoesComodo

| Campo | Tipo | Descrição |

|-------------|-----------|-------------------------------------------------|

| id | INT | Chave primária |

| largura | DECIMAL(5,2) | Largura do cômodo em metros |

| comprimento | DECIMAL(5,2) | Comprimento do cômodo em metros |

| altura | DECIMAL(5,2) | Altura do cômodo em metros |

| id\_projeto | INT | Chave estrangeira referenciando Projeto(id) |

Tabela: DimensoesCeramica

| Campo | Tipo | Descrição |

|-------------|-----------|-------------------------------------------------|

| id | INT | Chave primária |

| comprimento | DECIMAL(5,2) | Comprimento da cerâmica em metros |

| largura | DECIMAL(5,2) | Largura da cerâmica em metros |

| id\_projeto | INT | Chave estrangeira referenciando Projeto(id) |

Observações:

- Os tipos de dados e tamanhos são representativos e podem variar dependendo do SGBD.

- Este dicionário de dados fornece informações detalhadas sobre cada campo em cada tabela, facilitando a compreensão do propósito e do tipo de dados armazenados.

### Ambiente de Desenvolvimento

.

A Calculadora de Quantidade de Cerâmicas pode depender de vários sistemas e componentes externos para fornecer funcionalidades específicas. Abaixo estão alguns exemplos de sistemas e componentes externos que podem ser utilizados:

1. Framework Front-end:

- O front-end da aplicação pode depender de um framework como Node.js para facilitar o desenvolvimento da interface do usuário, gerenciamento de estado e manipulação eficiente do DOM.

2. Biblioteca para Requisições HTTP :

- Pode ser utilizado para fazer requisições HTTP entre o front-end e o back-end, permitindo a comunicação eficiente entre os dois componentes.

3. Servidor de Aplicação:

- O back-end da aplicação pode depender de um servidor de aplicação, como Node.js com o framework Express, para lidar com rotas, lógica de negócios e interações com o banco de dados.

4. Banco de Dados Relacional:

- Para armazenar dados de usuários, projetos, dimensões do cômodo, dimensões da cerâmica, etc.

5. Sistema de Controle de Versão:

- Utilizado para controle de versão do código-fonte da aplicação, permitindo o trabalho colaborativo e o rastreamento de alterações.

6. Ambiente de Desenvolvimento :

- Ferramentas de desenvolvimento integradas que auxiliam os desenvolvedores durante o processo de codificação, depuração e teste.

7. Sistema de Hospedagem :

- Pode ser utilizado para hospedar a aplicação, disponibilizando-a online para os usuários finais.

8. Biblioteca de Estilos:

- Facilita o design responsivo e a estilização da interface do usuário.

9. Sistema de Autenticaçã:

- Utilizado para autenticar usuários durante o processo de login e garantir a segurança da aplicação.

10. Sistema de Logs e Monitoramento:

- Ferramentas para monitorar e registrar eventos significativos na aplicação, auxiliando na identificação e correção de problemas.

Esses são apenas alguns exemplos e a escolha dos sistemas e componentes externos dependerá dos requisitos específicos da aplicação, das preferências da equipe de desenvolvimento e das tecnologias mais adequadas para o projeto.

## Implementação

1. Configuração do Ambiente de Desenvolvimento:

- Instale as dependências necessárias, como Node.js e npm (ou yarn).

- Crie um projeto Node usando `create-Node-app` ou um método equivalente.

2. Desenvolvimento da Interface do Usuário:

- Projete e implemente as telas necessárias, como a tela principal, tela de configurações, e tela de resultados.

- Utilize componentes Node.js para modularizar o código e facilitar a manutenção.

3. Lógica de Interface e Estado:

- Gerencie o estado da aplicação de forma eficiente, utilizando o estado local do Node ou um gerenciador de estado global, como o Redux.

- Implemente a lógica necessária para capturar as dimensões do cômodo, dimensões da cerâmica, e outras configurações.

4. Integração com o Back-end:

- Faça chamadas HTTP para o back-end para enviar dados do usuário, projetos e configurações.

- Lide com as respostas do back-end para atualizar a interface do usuário conforme necessário.

5. Estilização e Responsividade:

- Utilize bibliotecas de estilos (por exemplo, Bootstrap, Material-UI) para melhorar o design.

- Garanta que a aplicação seja responsiva para funcionar em diferentes dispositivos.

Back-end (Lógica de Negócio e Banco de Dados):

1. Configuração do Ambiente de Desenvolvimento:

- Escolha um ambiente de execução, como Node.js, e instale as dependências necessárias, como Express e um ORM para o banco de dados.

2. Desenvolvimento da Lógica de Negócio:

- Implemente a lógica necessária para calcular a quantidade de cerâmicas com base nas dimensões do cômodo e da cerâmica.

- Desenvolva rotas no back-end para receber dados do front-end.

3. Interação com o Banco de Dados:

- Configure modelos de dados para o banco de dados utilizando um ORM (por exemplo, Sequelize para Node.js).

- Implemente operações CRUD (Create, Read, Update, Delete) para projetos, dimensões do cômodo, dimensões da cerâmica, etc.

4. Autenticação e Segurança:

- Implemente sistemas de autenticação, se necessário, para proteger rotas sensíveis.

- Certifique-se de tratar e validar os dados para prevenir vulnerabilidades, como injeção de SQL.

5. Testes e Depuração:

- Desenvolva testes unitários e de integração para garantir a robustez do back-end.

- Utilize ferramentas de depuração para identificar e corrigir possíveis erros.

6. Implantação:

- Hospede a aplicação em um ambiente de produção, como AWS, Heroku ou outro provedor de serviços em nuvem.

7. Monitoramento e Manutenção:

- Configure ferramentas de monitoramento para acompanhar o desempenho da aplicação.

- Esteja preparado para corrigir problemas e realizar manutenção conforme necessário.

8. Documentação:

- Documente o código-fonte, as rotas da API, e as decisões de design para facilitar futuras atualizações e manutenções.

## Testes

1. Front-end:

- Teste a renderização correta dos componentes.

- Verifique se a lógica de estado e props está funcionando conforme o esperado.

- Teste interações do usuário (por exemplo, cliques, entrada de dados) e como elas afetam o estado.

2. Back-end (Node.js/Express):

- Escreva testes para cada rota do back-end.

- Verifique se os controladores manipulam corretamente os dados recebidos.

- Teste a lógica de negócio, incluindo o cálculo da quantidade de cerâmicas.

Testes de Integração:

1. Integração Front-end/Back-end:

- Teste a comunicação entre o front-end e o back-end, garantindo que os dados sejam enviados e recebidos corretamente.

2. Integração com o Banco de Dados:

- Realize testes que envolvam operações CRUD no banco de dados.

- Garanta que os dados sejam armazenados e recuperados corretamente.

Testes de Aceitação do Usuário (UAT):

1. Crie Cenários de Teste:

- Desenvolva cenários de teste que simulem casos de uso reais da aplicação.

2. Envolvimento do Usuário Final:

- Envolva usuários finais ou membros da equipe de negócios para realizar testes e fornecer feedback sobre a usabilidade e funcionalidades.

Testes de Desempenho:

1. Testes de Carga:

- Avalie como a aplicação se comporta sob carga pesada simulada.

- Identifique possíveis gargalos de desempenho e otimize conforme necessário.

Testes de Segurança:

1. Testes de Segurança da API:

- Garanta que a API esteja protegida contra vulnerabilidades, como injeção de SQL ou ataques de injeção de código.

2. Testes de Autenticação:

- Certifique-se de que os mecanismos de autenticação estejam funcionando corretamente.

Testes de Manutenção:

1. Testes de Regressão:

- Execute testes de regressão para garantir que novas alterações não afetem negativamente as funcionalidades existentes.

2. Testes de Atualizações:

- Teste a aplicação após a aplicação de atualizações ou correções.

Ferramentas de Teste:

1. Jest:

- Uma estrutura de teste amplamente utilizada para aplicações JavaScript e Node.js.

2. Node Testing Library:

- Uma biblioteca para testar componentes Node de uma maneira que simula a interação do usuário.

3. Supertest:

- Uma biblioteca para testar APIs REST no Node.js.

4. Postman ou Insomnia:

- Ferramentas para testar manualmente APIs e verificar respostas.

5. Cypress:

- Uma ferramenta de teste de ponta a ponta para testar interações do usuário no navegador.

6. JMeter:

- Utilizado para testes de carga e desempenho.

### Plano de Testes

- Front-end (Node):

- Testar a renderização correta de componentes.

- Verificar a lógica de estado e props.

- Testar interações do usuário.

- Back-end (Node.js/Express):

- Testar rotas individualmente.

- Verificar manipulação de dados nos controladores.

- Testar lógica de negócio, incluindo o cálculo da quantidade de cerâmicas.

2. Testes de Integração:

- Front-end/Back-end:

- Testar a comunicação entre front-end e back-end.

- Garantir que os dados sejam enviados e recebidos corretamente.

- Integração com o Banco de Dados:

- Testar operações CRUD no banco de dados.

- Garantir armazenamento e recuperação corretos de dados.

3. Testes de Aceitação do Usuário (UAT):

- Cenários de Teste:

- Desenvolver cenários de teste realistas.

- Envolver usuários finais para testar e fornecer feedback.

4. Testes de Desempenho:

- Testes de Carga:

- Avaliar desempenho sob carga simulada.

- Identificar e corrigir possíveis gargalos.

5. Testes de Segurança:

- Segurança da API:

- Testar proteção contra injeção de SQL e outros ataques.

- Garantir que a autenticação esteja segura.

6. Testes de Manutenção:

- Testes de Regressão:

- Executar testes de regressão após alterações.

- Verificar se novas alterações afetam funcionalidades existentes.

7. Ferramentas de Teste:

- Utilizar ferramentas como Jest, Node Testing Library, Supertest, Postman, Cypress, JMeter, etc., conforme apropriado.

8. Cronograma de Testes:

- Estabelecer uma programação para cada tipo de teste.

- Incluir datas de início e término para garantir que os testes sejam concluídos em tempo hábil.

9. Critérios de Aceitação:

- Definir critérios claros para determinar quando um teste é bem-sucedido.

- Por exemplo, todos os testes de unidade devem passar sem erros.

10. Recursos Necessários:

- Especificar os recursos necessários para a execução dos testes, incluindo hardware, software e pessoal.

11. Riscos e Mitigações:

- Identificar possíveis riscos nos testes e propor estratégias de mitigação.

Observações:

- Este é um plano de testes genérico e deve ser adaptado de acordo com as necessidades específicas do projeto.

- Testes devem ser realizados em ambientes de desenvolvimento, teste e produção.

- Resultados dos testes devem ser documentados e compartilhados com a equipe de desenvolvimento.

### Execução do Plano de Testes

1. Configuração do Ambiente de Testes:

- Certifique-se de que os ambientes de teste (local, staging, etc.) estão configurados corretamente.

- Garanta que as versões de software, bibliotecas e dependências estejam alinhadas com o ambiente de produção.

2. Testes Unitários:

Front-end (Node):

1. Execute os testes unitários para os componentes Node.

2. Verifique se a renderização está correta.

3. Teste a lógica de estado e props.

4. Certifique-se de que as interações do usuário funcionem como esperado.

Back-end (Node.js/Express):

1. Execute os testes unitários para rotas e controladores.

2. Verifique se a manipulação de dados nos controladores está correta.

3. Teste a lógica de negócio, incluindo o cálculo da quantidade de cerâmicas.

3. Testes de Integração:

1. Teste a comunicação entre o front-end e o back-end.

2. Certifique-se de que os dados são enviados e recebidos corretamente.

3. Teste operações CRUD no banco de dados, garantindo armazenamento e recuperação corretos de dados.

4. Testes de Aceitação do Usuário (UAT):

1. Execute cenários de teste realistas.

2. Envolver usuários finais para testar e fornecer feedback.

3. Garantir que a aplicação atenda às expectativas e requisitos do usuário.

5. Testes de Desempenho:

1. Execute testes de carga para avaliar o desempenho sob carga simulada.

2. Identifique e corrija possíveis gargalos de desempenho.

6. Testes de Segurança:

1. Teste a segurança da API para garantir proteção contra injeção de SQL e outros ataques.

2. Certifique-se de que a autenticação seja segura.

7. Testes de Manutenção:

1. Execute testes de regressão após alterações.

2. Verifique se novas alterações não afetam funcionalidades existentes.

8. Documentação e Relatórios:

1. Documente os resultados de cada teste.

2. Relate quaisquer problemas encontrados durante os testes.

3. Mantenha registros de execução, incluindo datas e resultados.

9. Correção de Defeitos:

1. Corrija quaisquer defeitos ou problemas identificados durante os testes.

2. Reexecute os testes relevantes para verificar se as correções foram bem-sucedidas.

10. Validação Final:

1. Valide se todos os requisitos estão atendidos.

2. Certifique-se de que os usuários finais e as partes interessadas estejam satisfeitos com a aplicação.

## Implantação

1. Preparação para Implantação:

- Código Estável: Certifique-se de que a versão do código que será implantada é estável e foi testada com sucesso.

- Backup dos Dados: Faça backup dos dados do ambiente de produção, especialmente se houver alterações no banco de dados durante a implantação.

2. Configuração do Ambiente de Produção:

- Infraestrutura de Servidores: Certifique-se de que os servidores e recursos de rede necessários estejam disponíveis e configurados.

- Servidor de Aplicação: Instale e configure o servidor de aplicação necessário (por exemplo, Node.js/Express).

- Banco de Dados: Garanta que o banco de dados esteja configurado e acessível a partir do ambiente de produção.

3. Transferência de Código:

- Código-Fonte: Transfira o código-fonte da aplicação para os servidores de produção.

- Dependências: Instale as dependências da aplicação usando o gerenciador de pacotes apropriado (por exemplo, npm).

4. Migração do Banco de Dados:

- Esquemas: Se houver alterações no esquema do banco de dados, execute as migrações necessárias para garantir que o banco de dados esteja na versão correta.

5. Configuração do Ambiente:

- Variáveis de Ambiente: Configure as variáveis de ambiente necessárias, como chaves de API, URLs de serviço, etc.

- Arquivos de Configuração: Atualize os arquivos de configuração da aplicação para refletir as configurações específicas do ambiente de produção.

6. Testes Finais:

- Testes de Produção: Realize testes finais no ambiente de produção para garantir que a aplicação funcione conforme o esperado.

- Testes de Desempenho: Execute testes de desempenho adicionais, se necessário.

7. Atualização do Servidor:

- Reinício ou Atualização: Reinicie o servidor de aplicação ou realize uma atualização para aplicar as alterações.

- Verificação: Verifique se a aplicação está acessível e funcionando conforme esperado.

8. Monitoramento Contínuo:

- Ferramentas de Monitoramento: Configure ferramentas de monitoramento para acompanhar o desempenho e detectar problemas em tempo real.

9. Atualização do DNS (se aplicável):

- Atualização de Registros DNS: Se a aplicação estiver associada a um domínio personalizado, atualize os registros DNS para apontar para o novo ambiente de produção.

10. Comunicação com as Partes Interessadas:

- Notificação: Notifique as partes interessadas sobre a conclusão bem-sucedida da implantação.

- Instruções: Forneça quaisquer instruções adicionais necessárias para usuários finais ou administradores.

11. Documentação:

- Atualização da Documentação: Atualize a documentação para refletir as configurações e procedimentos do ambiente de produção.

12. Backup Adicional (Opcional):

- Backup Suplementar: Se desejar, faça um backup adicional dos dados do ambiente de produção para segurança.

13. Rollback (Plano de Contingência):

- Plano de Rollback: Tenha um plano de rollback em vigor caso ocorram problemas durante a implantação.

- Backup Adicional: Faça um backup adicional dos dados do ambiente de produção, se necessário.

### Diagrama de Implantação

1. Cliente (Navegador):

- Representa o dispositivo do usuário final, como um navegador da web, por meio do qual os usuários interagem com a aplicação.

2. Servidor de Aplicação:

- Onde a lógica da aplicação é executada. Pode incluir o ambiente de execução (por exemplo, Node.js) e o servidor web (por exemplo, Express).

3. Banco de Dados:

- Onde os dados da aplicação são armazenados. Pode incluir um sistema de gerenciamento de banco de dados (por exemplo, MySQL, PostgreSQL).

Nodes (Nós):

1. Cliente (Máquina do Usuário):

- Representa a máquina física ou virtual do usuário final que acessa a aplicação por meio de um navegador.

2. Servidor de Aplicação:

- Máquina física ou virtual que hospeda o servidor de aplicação.

3. Servidor de Banco de Dados:

- Máquina física ou virtual que hospeda o banco de dados.

Conexões:

1. Cliente <-> Servidor de Aplicação:

- Representa a comunicação entre o cliente (navegador) e o servidor de aplicação. Pode incluir protocolos como HTTP/HTTPS.

2. Servidor de Aplicação <-> Banco de Dados:

- Representa a comunicação entre o servidor de aplicação e o banco de dados. Inclui consultas e atualizações de dados.

Exemplo de Diagrama de Implantação:

Linha do tempo

Descrição gerada automaticamente com confiança média

Neste exemplo:

- O Cliente (Navegador) se comunica com o Servidor de Aplicação por meio da internet, realizando solicitações HTTP/HTTPS.

- O Servidor de Aplicação processa as solicitações, executa a lógica da aplicação e se comunica com o Servidor de Banco de Dados para recuperar ou atualizar dados.

- O Servidor de Banco de Dados armazena e recupera dados conforme solicitado pelo Servidor de Aplicação.

### Manual de Implantação

1. Introdução:

- Breve visão geral do sistema.

- Propósito do manual de implantação.

- Audiência-alvo.

2. Requisitos de Sistema:

- Especificações mínimas de hardware e software.

- Dependências do sistema.

3. Configuração do Ambiente:

3.1. Servidor de Aplicação:

- Instalação do Node.js e Express.

- Configuração do servidor web.

- Configuração do ambiente de execução.

3.2. Banco de Dados:

- Instalação e configuração do sistema de gerenciamento de banco de dados (por exemplo, MySQL, PostgreSQL).

- Configuração do esquema do banco de dados.

4. Transferência de Código:

- Clone do repositório Git.

- Instalação de dependências utilizando npm ou yarn.

- Configuração de variáveis de ambiente.

5. Migração do Banco de Dados:

- Execução de migrações do banco de dados, se necessário.

- Verificação do estado da migração.

6. Configuração de Variáveis de Ambiente:

- Configuração de variáveis de ambiente para produção.

- Exemplos de chaves e valores necessários.

7. Testes Finais:

- Execução de testes finais no ambiente de produção.

- Verificação do desempenho da aplicação.

8. Atualização do Servidor:

- Reinicialização ou atualização do servidor de aplicação.

- Verificação da integridade da aplicação após a atualização.

9. Monitoramento Contínuo:

- Configuração de ferramentas de monitoramento.

- Verificação contínua do desempenho e disponibilidade.

10. Atualização do DNS (se aplicável):

- Instruções para atualização de registros DNS.

- Verificação da acessibilidade via domínio personalizado.

11. Comunicação com as Partes Interessadas:

- Notificação de conclusão bem-sucedida da implantação.

- Fornecimento de informações de suporte.

12. Documentação:

- Atualização da documentação para refletir as configurações e procedimentos de produção.

13. Backup Adicional (Opcional):

- Instruções para a realização de backups adicionais, se necessário.

14. Rollback (Plano de Contingência):

- Instruções para realizar um rollback em caso de problemas críticos.

15. Suporte Técnico:

- Informações de contato para suporte técnico.

- Procedimentos de escalonamento de problemas.

16. Histórico de Revisões:

- Registro de alterações feitas no manual.

Observações:

- Personalização: Adapte as instruções de acordo com o ambiente específico da sua organização.

- Validação: Realize a validação cuidadosa de cada etapa para garantir uma implantação bem-sucedida.

## Manual do Usuário

1. Introdução

- 1.1 Objetivo do Sistema

- 1.2 Público-Alvo

2. Acesso ao Sistema

- 2.1 Requisitos de Acesso

- 2.2 Login

- 2.3 Recuperação de Senha

3. Página Inicial

- 3.1 Visão Geral da Página Inicial

- 3.2 Navegação no Sistema

4. Cadastro de Cômodos

- 4.1 Adição de Novo Cômodo

- 4.2 Edição e Exclusão de Cômodos

5. Cálculo de Quantidade de Cerâmicas

- 5.1 Seleção do Cômodo

- 5.2 Inserção de Medidas

- 5.3 Execução do Cálculo

- 5.4 Visualização dos Resultados

6. Histórico de Cálculos

- 6.1 Acesso ao Histórico

- 6.2 Detalhes dos Cálculos Anteriores

- 6.3 Exportação de Dados do Histórico

7. Configurações da Conta

- 7.1 Atualização de Informações da Conta

- 7.2 Alteração de Senha

- 7.3 Configurações de Privacidade

8. Suporte e Contato

- 8.1 Perguntas Frequentes (FAQ)

- 8.2 Suporte Técnico

- 8.3 Feedback

9. Sair do Sistema

1. Introdução

1.1 Objetivo do Sistema

A Calculadora de Quantidade de Cerâmicas é uma ferramenta destinada a auxiliar usuários na determinação da quantidade necessária de cerâmicas para revestir um determinado cômodo.

1.2 Público-Alvo

Este sistema é voltado para proprietários de residências, arquitetos, construtores e qualquer pessoa envolvida em projetos de revestimento com cerâmicas.

2. Acesso ao Sistema

2.1 Requisitos de Acesso

Para utilizar a Calculadora de Quantidade de Cerâmicas, é necessário ter um dispositivo com acesso à internet e um navegador web moderno.

2.2 Login

- Acesse [URL do Sistema].

- Insira seu nome de usuário e senha.

- Clique em "Login" para acessar sua conta.

2.3 Recuperação de Senha

- Na página de login, clique em "Esqueceu a Senha?"

- Siga as instruções para redefinir sua senha.

3. Página Inicial

3.1 Visão Geral da Página Inicial

Após o login, você será direcionado para a página inicial, onde poderá visualizar os cômodos cadastrados e acessar outras funcionalidades do sistema.

3.2 Navegação no Sistema

Utilize o menu de navegação para acessar diferentes seções do sistema, como o cadastro de cômodos, cálculos, histórico e configurações da conta.

4. Cadastro de Cômodos

4.1 Adição de Novo Cômodo

- Na página inicial, clique em "Adicionar Cômodo".

- Preencha as informações solicitadas.

- Clique em "Salvar" para adicionar o cômodo.

4.2 Edição e Exclusão de Cômodos

- Na página inicial, clique no cômodo desejado.

- Para editar, clique em "Editar" e faça as alterações necessárias.

- Para excluir, clique em "Excluir" e confirme.

5. Cálculo de Quantidade de Cerâmicas

5.1 Seleção do Cômodo

- No menu, clique em "Calculadora".

- Selecione o cômodo desejado a partir da lista.

5.2 Inserção de Medidas

- Insira as medidas do cômodo conforme solicitado.

- Clique em "Próximo" para avançar.

5.3 Execução do Cálculo

- Revise as informações inseridas.

- Clique em "Calcular" para obter os resultados.

5.4 Visualização dos Resultados

- Visualize a quantidade estimada de cerâmicas necessárias.

- Salve o cálculo para referência futura, se necessário.

6. Histórico de Cálculos

6.1 Acesso ao Histórico

- No menu, clique em "Histórico".

- Visualize uma lista de cálculos anteriores.

6.2 Detalhes dos Cálculos Anteriores

- Clique em um cálculo para ver detalhes completos.

6.3 Exportação de Dados do Histórico

- Exporte dados do histórico para um formato desejado.

7. Configurações da Conta

7.1 Atualização de Informações da Conta

- No menu, clique em "Configurações".

- Atualize as informações da conta conforme necessário.

7.2 Alteração de Senha

- Na página de configurações, clique em "Alterar Senha".

- Siga as instruções para alterar sua senha.

7.3 Configurações de Privacidade

- Ajuste as configurações de privacidade de acordo com suas preferências.

8. Suporte e Contato

8.1 Perguntas Frequentes (FAQ)

- Acesse a seção de FAQ para obter respostas a perguntas comuns.

8.2 Suporte Técnico

- Entre em contato com nosso suporte técnico em [e-mail ou número de telefone].

8.3 Feedback

- Fornecer feedback sobre o sistema.

9. Sair do Sistema

- No menu, clique em "Sair" para encerrar a sessão.

## Conclusões e Considerações Finais

A implementação da Calculadora de Quantidade de Cerâmicas representa um marco significativo na busca por soluções práticas e eficientes no contexto de projetos de revestimento de ambientes com cerâmicas. Ao longo do desenvolvimento e implantação deste sistema, algumas conclusões e considerações finais merecem destaque:

1. Realização do Objetivo:

A concepção e entrega da Calculadora cumpriram o objetivo central de oferecer uma ferramenta intuitiva e útil para usuários que desejam calcular a quantidade necessária de cerâmicas para seus projetos. A aplicação simplifica um processo complexo, proporcionando conveniência e economia de tempo aos usuários.

2. Usabilidade e Interface Amigável:

A usabilidade foi uma prioridade durante o desenvolvimento, refletindo-se na interface amigável da Calculadora. A disposição lógica dos elementos, instruções claras e um design intuitivo contribuem para uma experiência de usuário positiva, independentemente do nível de familiaridade com ferramentas semelhantes.

3. Feedback dos Usuários:

Ao longo do ciclo de desenvolvimento, foram coletados e considerados feedbacks valiosos dos usuários beta. As melhorias implementadas em resposta a esses comentários contribuíram significativamente para a qualidade final da aplicação.

4. Atualizações Futuras:

A natureza dinâmica deste projeto permite que ele evolua em resposta às necessidades em constante mudança dos usuários. As atualizações futuras estão previstas para incorporar novas funcionalidades, otimizações de desempenho e aprimoramentos com base no feedback contínuo.

5. Desafios Superados:

Durante o desenvolvimento, enfrentamos desafios que foram superados com criatividade e dedicação da equipe. Questões técnicas, requisitos de escalabilidade e integração de diferentes componentes foram abordados com sucesso para garantir uma aplicação robusta e confiável.

6. Contribuição para a Eficiência do Projeto:

A Calculadora de Quantidade de Cerâmicas não apenas simplifica a fase de planejamento de projetos de revestimento, mas também contribui para a eficiência global do processo. A precisão dos cálculos reduz a possibilidade de desperdício de materiais, resultando em benefícios econômicos e ambientais.

7. Agradecimentos:

Expressamos nossa gratidão aos membros da equipe, colaboradores, usuários beta e todos os envolvidos neste projeto. A colaboração e dedicação foram fundamentais para o sucesso da Calculadora de Quantidade de Cerâmicas.

Em conclusão, acreditamos que a Calculadora não apenas atende às expectativas iniciais, mas também se posiciona como uma ferramenta valiosa para profissionais e entusiastas que buscam otimizar seus projetos de revestimento com cerâmicas. Estamos comprometidos em continuar aprimorando e oferecendo suporte contínuo para garantir a excelência e a satisfação dos usuários.

## Bibliografia

1. Desenvolvimento Web:

- Flanagan, David. "JavaScript: The Definitive Guide." O'Reilly Media, 2018.

- Duckett, Jon. "HTML and CSS: Design and Build Websites." Wiley, 2011.

2. Banco de Dados:

- Date, C.J. "An Introduction to Database Systems." Pearson, 2016.

- Atkinson, Colin. "Object-Oriented Reengineering Patterns." Morgan Kaufmann, 2002.

3. Design de Interface do Usuário (UI) e Experiência do Usuário (UX):

- Cooper, Alan. "About Face: The Essentials of Interaction Design." Wiley, 2014.

- Tondreau, Barbara. "Layout Essentials: 100 Design Principles for Using Grids." Rockport Publishers, 2009.

4. Desenvolvimento Ágil e Metodologias de Projeto:

- Cohn, Mike. "Agile Estimating and Planning." Prentice Hall, 2005.

- Schwaber, Ken, and Beedle, Mike. "Agile Software Development with Scrum." Prentice Hall, 2002.

5. Documentação de Código e Comentários:

- Knuth, Donald E. "Literate Programming." Stanford University, 1984. (Artigo)

- Microsoft Docs: Writing Comments in Code (https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/xmldoc/)

# 