

Desenvolvimento de um Sistema Web para Divulgação e Gestão de Orçamentos em uma Empresa de Serviços de Gesso

Diogo Alves Graciano¹
Orientador: Dr. Héctor Azpúrua ¹

Abstract—Este trabalho apresenta o desenvolvimento de um sistema web para divulgação, simulação e gestão de orçamentos de serviços de gesso. O problema tratado envolve a ausência de uma presença digital estruturada e a dependência de processos manuais para elaboração de orçamentos preliminares, situação comum em pequenas empresas de prestação de serviços. O sistema desenvolvido integra uma vitrine pública de serviços, um simulador de orçamento acessível ao cliente e um painel administrativo para manutenção de serviços, categorias, opcionais, imagens e contatos. A aplicação foi construída em arquitetura de três camadas, utilizando React, TypeScript e Vite no frontend, ASP.NET Core no backend, PostgreSQL como banco de dados relacional e Liquibase para versionamento do esquema. O sistema contempla autenticação administrativa com JWT e refresh token, validações de entrada, regras de cálculo por modo de medição, separação de valores de mão de obra e material, aplicação de valor mínimo e envio da estimativa pelo WhatsApp. Como resultado, obteve-se um MVP funcional que reduz o esforço inicial de atendimento, padroniza o cálculo de estimativas e oferece à empresa uma base para ampliar sua atuação digital.

I. INTRODUÇÃO

O setor de serviços de gesso, especialmente no contexto de pequenas e médias empresas, ainda é marcado por práticas administrativas pouco digitalizadas. Em muitas situações, a divulgação dos serviços ocorre de maneira dispersa, por indicações, redes sociais ou conversas em aplicativos de mensagem, enquanto o orçamento inicial depende de medições presenciais, cálculos manuais e troca contínua de informações entre cliente e prestador. Embora a visita técnica continue sendo essencial para a confirmação de detalhes de execução, esse

modelo torna o primeiro contato mais lento, exige esforço operacional antes mesmo da qualificação da demanda e dificulta a padronização das propostas.

Ao mesmo tempo, clientes passaram a esperar experiências digitais mais imediatas, com acesso a informações claras, imagens, valores de referência e canais de contato simples. A ausência de uma vitrine digital estruturada limita a capacidade de uma empresa apresentar seus serviços de forma profissional e também reduz a autonomia do cliente para compreender previamente o tipo de serviço desejado, as formas de medição e a ordem de grandeza do investimento necessário. Conforme discutido por Kotler e Keller [1], canais de relacionamento e apresentação de valor exercem papel importante na experiência do cliente e na competitividade das organizações.

Diante desse cenário, este trabalho propõe e implementa um sistema web para uma empresa de serviços de gesso, combinando catálogo público, simulação de orçamento e gestão administrativa. A proposta não substitui a visita técnica, mas antecipa uma estimativa padronizada e melhora a qualidade da informação trocada no primeiro contato. O cliente pode visualizar serviços, informar medidas, selecionar opcionais e enviar a simulação pelo WhatsApp [11]. O administrador, por sua vez, mantém os dados que alimentam o catálogo e o simulador, como categorias, preços, imagens, formas de medição e contatos públicos.

A. Contexto e Problema

Empresas de pequeno porte que atuam com serviços de gesso costumam concentrar sua capacidade operacional na execução das obras, mantendo

¹ Universidade Federal de Minas Gerais - UFMG - Departamento de Ciência da Computação

processos comerciais e administrativos simples. Essa característica é compatível com a realidade de negócios familiares ou equipes enxutas, mas pode gerar gargalos quando a empresa precisa atender muitos contatos iniciais, explicar diferenças entre serviços, solicitar medidas e calcular valores preliminares.

O processo tradicional de orçamento apresenta três limitações principais. A primeira é a baixa escalabilidade do atendimento, pois cada novo cliente exige interação manual para coleta das mesmas informações. A segunda é a possibilidade de inconsistência no cálculo, já que diferentes atendentes podem interpretar medidas e opcionais de formas distintas. A terceira é a perda de oportunidades comerciais, pois clientes que buscam uma estimativa rápida podem desistir antes de receber um retorno.

O problema central deste trabalho pode ser sintetizado na seguinte questão: como apoiar uma empresa de serviços de gesso na divulgação estruturada de seus serviços e na geração de orçamentos preliminares de forma mais ágil, padronizada e integrada ao processo de atendimento?

B. Objetivos

O objetivo geral deste trabalho é desenvolver um sistema web funcional para divulgação e gestão de orçamentos de serviços de gesso, contemplando uma interface pública para clientes e uma interface administrativa para manutenção do conteúdo e dos parâmetros de cálculo.

Os objetivos específicos são:

- levantar e organizar os requisitos funcionais e não funcionais do sistema;
- modelar os principais casos de uso, entidades, componentes e fluxos de interação;
- implementar uma aplicação frontend responsiva para catálogo, detalhamento de serviços, contato e simulação de orçamento;
- implementar uma API REST com regras de negócio, autenticação, validações e persistência;
- estruturar o banco de dados relacional com controle versionado de mudanças;
- disponibilizar um painel administrativo para gerenciar serviços, categorias, contatos, imagens, opcionais e status de publicação;

- validar as principais regras por meio de testes automatizados e execução dos fluxos centrais do sistema.

C. Justificativa

O projeto se justifica pela oportunidade de aplicar conceitos de engenharia de software a um problema real de uma pequena empresa. A digitalização do processo inicial de orçamento pode reduzir retrabalho, padronizar cálculos, melhorar a experiência do cliente e aumentar a percepção de profissionalismo da empresa. Além disso, a centralização dos parâmetros de preço em um painel administrativo permite que alterações comerciais sejam refletidas imediatamente no catálogo e no simulador, sem a necessidade de intervenção manual de um usuário técnico especializado.

Do ponto de vista acadêmico, o trabalho integra atividades típicas do ciclo de desenvolvimento de sistemas de informação, incluindo elicitação de requisitos, modelagem, projeto arquitetural, implementação, testes e discussão de resultados. A aplicação também oferece um estudo de caso sobre a adoção de tecnologias web em um domínio de prestação de serviços que ainda depende de processos informais.

D. Organização do Trabalho

O restante do texto está organizado da seguinte forma. A Seção II apresenta o referencial teórico e trabalhos relacionados. A Seção III descreve a metodologia adotada. A Seção IV detalha a especificação do sistema. A Seção V apresenta a arquitetura, a modelagem e as decisões de implementação. A Seção VI descreve as interfaces e os principais fluxos de uso. A Seção VII apresenta a validação, os testes e os resultados obtidos. Por fim, a Seção VIII reúne as conclusões e possibilidades de trabalhos futuros.

II. REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

A. Orçamentação em Serviços de Construção

O orçamento em serviços de construção civil consiste na estimativa dos custos necessários para executar uma atividade, considerando mão de obra, materiais, quantidades, perdas, complexidade e condições específicas do local. Em serviços de gesso, é comum que a precificação inicial seja

baseada em medidas como metro quadrado, metro linear ou unidade, dependendo do tipo de serviço. Rebaixamentos de teto, paredes e revestimentos costumam usar áreas; molduras e sancas podem usar metragem linear; e alguns itens adicionais podem ser cobrados por unidade.

Em orçamentos preliminares, a precisão absoluta não é o objetivo principal. O propósito é oferecer uma estimativa útil para apoiar a decisão inicial do cliente e orientar a continuidade do atendimento. Mesmo assim, a estimativa precisa ser consistente. Valores muito baixos podem comprometer a margem de lucro, enquanto valores muito altos podem reduzir a competitividade da proposta. Guias de estimativa de custos destacam que a clareza dos critérios de cálculo e a organização dos insumos são fundamentais para que a estimativa seja compreendida e revisada [3].

Neste trabalho, o orçamento é tratado como uma simulação preliminar. O sistema calcula valores a partir de medidas informadas pelo cliente e parâmetros cadastrados pelo administrador, mas comunica explicitamente que o valor final depende de visita técnica. Essa decisão preserva o caráter comercial da ferramenta sem assumir que todos os detalhes de obra possam ser capturados por formulário.

B. Sistemas Web e Arquitetura em Três Camadas

Aplicações web modernas frequentemente são organizadas em três camadas principais: apresentação, lógica de negócio e dados. A camada de apresentação é responsável pela interação com o usuário no navegador. A camada de lógica de negócio concentra regras, validações, autenticação e exposição de endpoints. A camada de dados persiste as informações em um mecanismo de armazenamento, frequentemente um banco relacional.

Essa separação de responsabilidades facilita manutenção, testes e evolução. Alterações visuais podem ser realizadas no frontend sem modificar diretamente as regras de negócio, enquanto ajustes de persistência podem ser encapsulados na API. Sommerville [2] destaca que a modularidade e a separação de responsabilidades são princípios relevantes para reduzir complexidade e aumentar a manutenibilidade de sistemas.

Neste projeto, a camada de apresentação foi construída como uma Single-Page Application em React [5]. O React é uma biblioteca JavaScript para construção de interfaces baseada em componentes reutilizáveis e combináveis. Em vez de manipular diretamente cada elemento da página, a interface é descrita por componentes que recebem propriedades, mantêm estado e produzem a representação visual correspondente. Essa abordagem favorece a decomposição da aplicação em páginas, formulários, cards, estados de carregamento e componentes administrativos reaproveitáveis. A escolha do React se justifica pela aderência ao modelo de aplicação interativa adotado no sistema, pela integração com TypeScript e pelo ecossistema consolidado para roteamento, organização de estado e construção de interfaces responsivas.

O Vite foi utilizado como ferramenta de desenvolvimento e empacotamento do frontend [6]. Ele fornece servidor local de desenvolvimento, atualização rápida durante a edição do código e processo de geração dos arquivos finais da aplicação. No contexto deste trabalho, o Vite foi selecionado por simplificar a configuração de um projeto React com TypeScript, reduzir o tempo de feedback durante a implementação e permitir o uso de proxy para encaminhar chamadas locais à API durante o desenvolvimento.

A comunicação com a API ocorre por HTTP em formato JSON e foi centralizada com Axios [7]. O Axios é um cliente HTTP para JavaScript que permite organizar requisições, respostas, tratamento de erros e configurações comuns em uma camada própria da aplicação. Essa escolha facilitou a definição de uma URL base, o envio de credenciais de autenticação, o tratamento de respostas da API e a renovação de sessão por meio de interceptadores. O backend foi desenvolvido em ASP.NET Core [8], expondo endpoints REST e utilizando Entity Framework Core para acesso ao PostgreSQL [9]. O versionamento do banco é realizado com Liquibase [10], o que permite registrar a evolução do esquema em arquivos controlados pelo repositório.

C. Segurança em Painéis Administrativos

Como o sistema possui uma área administrativa capaz de alterar preços, imagens, contatos e disponibilidade dos serviços, é necessário proteger esses

fluxos contra acesso não autorizado. Para isso, adotou-se autenticação baseada em tokens JWT, mecanismo de renovação de token de autenticação (refresh token) para renovação de sessão e política de autorização que exige usuário autenticado por padrão nos endpoints não públicos. As rotas públicas, como catálogo e simulação, permanecem acessíveis sem login.

O cadastro de novos administradores não foi implementado na interface pública, sendo feita apenas por inserção direta no banco de dados, ou via rota administrativa protegida por senha mestra. Essa decisão simplifica o escopo do MVP, evitando a necessidade de fluxos de registro, recuperação de senha e gerenciamento de múltiplos perfis administrativos, uma vez que o sistema é voltado para uma pequena empresa com poucos responsáveis pela manutenção do conteúdo.

Além da autenticação, foram adotadas validações nos DTOs (*Data Transfer Objects*, Objetos de Transferência de Dados) de entrada e tratamento padronizado de erros. Essa decisão reduz a chance de persistir dados inválidos e facilita a apresentação de mensagens compreensíveis no frontend. A validação ocorre tanto na interface quanto na API, evitando que a qualidade dos dados dependa apenas do navegador.

D. Soluções Similares

No mercado de construção civil existem ferramentas de cálculo de materiais e custos, como aplicativos voltados para drywall, planilhas de levantamento e sistemas de gestão de obras. Entretanto, muitas delas são direcionadas a profissionais da construção e não à experiência de um cliente final que deseja conhecer os serviços e obter informações sobre preços preliminares.

A lacuna explorada neste trabalho está na integração entre vitrine pública, simulação orientada ao cliente e gestão administrativa simples. A proposta não pretende substituir sistemas completos de ERP (*Enterprise Resource Planning*, Planejamento de Recursos Empresariais), CRM (*Customer Relationship Management*, Gestão de Relacionamento com o Cliente) ou gestão de obras, mas atender ao momento inicial do relacionamento comercial de uma pequena empresa de gesso.

III. METODOLOGIA

A. Tipo de Trabalho

Este trabalho caracteriza-se como uma pesquisa aplicada de desenvolvimento tecnológico. O sistema foi construído para resolver um problema prático de uma empresa de serviços de gesso, utilizando técnicas de engenharia de software e desenvolvimento web. O foco não é apenas descrever uma proposta conceitual, mas entregar um MVP funcional que possa ser executado, testado e evoluído.

B. Etapas Executadas

O desenvolvimento foi conduzido em etapas. Inicialmente, foram levantadas as necessidades do negócio e organizados os requisitos funcionais, requisitos não funcionais, regras de negócio e casos de uso. Em seguida, foram elaborados artefatos de modelagem, incluindo diagramas de casos de uso, classes, componentes, entidade-relacionamento, atividades e sequência.

Após a modelagem, foi definida a arquitetura técnica em três camadas. A implementação começou pela estrutura de dados e migrations do banco de dados, seguiu para a API e posteriormente para o frontend. Essa ordem buscou consolidar os contratos de dados e as regras de negócio antes da construção das telas, que dependem diretamente dessas definições estruturais. Por fim, foram realizados testes automatizados em serviços críticos e verificações de compilação do frontend e backend.

C. Tecnologias e Ferramentas

O frontend foi implementado com React, TypeScript, Vite, React Router, Axios e Tailwind CSS. O código foi organizado em páginas, componentes reutilizáveis, layouts, chamadas de API, tipos de dados e funcionalidades específicas, como catálogo de serviços, autenticação, simulador, área administrativa e listagem de contatos.

O backend foi implementado com ASP.NET Core, C#, Entity Framework Core, PostgreSQL, FluentValidation, AutoMapper, autenticação JWT e documentação local por Swagger/OpenAPI e ReDoc. A estrutura da API separa controllers, services, repositories, entidades, DTOs, validações, middlewares e extensões de registro de dependências.

O banco de dados utiliza PostgreSQL e Liquibase. Os changelogs versionam a criação de papéis de acesso, tabelas, permissões, registros base e evoluções do esquema durante a implementação. Entre as mudanças incorporadas ao longo do desenvolvimento estão a inclusão de modo de medição, contatos públicos, refresh token e separação de valores de mão de obra e material.

D. Critérios de Avaliação

A avaliação do sistema considerou os seguintes critérios:

- aderência aos requisitos definidos;
- funcionamento dos fluxos públicos de catálogo, detalhe, contato e simulação;
- funcionamento dos fluxos administrativos de autenticação e CRUD (*Create, Read, Update, Delete*, criação, leitura, atualização e exclusão) de serviços, categorias e contatos;
- coerência das regras de cálculo com os modos de medição;
- persistência adequada dos dados e evolução controlada do banco;
- execução de testes automatizados para regras de negócio e validações;
- responsividade e clareza das principais interfaces.

IV. ESPECIFICAÇÃO DO SISTEMA

A. Visão Geral

O sistema desenvolvido, denominado Gestão de Orçamentos, é composto por duas áreas principais. A área pública é voltada aos clientes e oferece página inicial, catálogo de serviços, detalhamento de cada serviço, página de contatos e simulador de orçamento. A área administrativa é voltada aos responsáveis pela empresa e oferece uma página inicial, login, gestão de serviços, categorias e contatos.

A área pública busca reduzir o esforço necessário para o cliente compreender os serviços oferecidos. A página inicial serve para recepcionar o usuário e apresentar informações gerais sobre a empresa. O catálogo exibe os serviços publicados, imagens, descrições resumidas, unidades de cobrança e valores de referência. O detalhe do serviço apresenta as informações completas, além de imagens, serviços opcionais e formulário para

simulação rápida de orçamento para aquele serviço específico. Já o simulador geral é mais completo, ele permite organizar o orçamento por cômodos, adicionar serviços repetidos ou diferentes em cada ambiente, informar medidas e selecionar opcionais, sendo recomendado para quem necessita de uma estimativa que contemple mais de um serviço ou mais de um cômodo.

A área administrativa concentra a manutenção dos dados que alimentam a interface pública. Um serviço pode ser cadastrado com nome, slug (identificador único), descrição, categoria, unidade de medida, modo de medição, preços de mão de obra e material, valor mínimo de serviço, imagens e opcionais. Categorias de serviços e contatos também são gerenciados pelo painel.

B. Requisitos Funcionais

Os principais requisitos funcionais da visão do cliente são:

- exibir página inicial com apresentação da empresa, destaques e chamadas para catálogo e simulador;
- listar serviços publicados no catálogo;
- apresentar detalhe completo de um serviço publicado;
- permitir simulação de orçamento com base em medidas, serviços e opcionais;
- validar entradas inválidas, negativas ou incompatíveis com o modo de medição;
- exibir resumo com itens, quantitativos, sub-totais, totais com e sem material e aviso de estimativa preliminar;
- gerar mensagem de contato pelo WhatsApp com dados da simulação;
- exibir contatos públicos ativos.

Os principais requisitos funcionais da visão administrativa são:

- autenticar administrador com login, logout, access token (token de acesso) e refresh token (token de atualização de sessão);
- proteger rotas administrativas no frontend e endpoints administrativos na API;
- criar, listar, editar, publicar, despublicar e deletar serviços;
- configurar preços, valor mínimo, modo de medição, imagens e opcionais de cada serviço;

- criar, listar, editar, ativar, desativar e remover categorias quando permitido;
- criar, listar, editar, ativar, desativar e remover contatos públicos;
- refletir alterações administrativas no catálogo e no simulador.

C. Requisitos Não Funcionais

Entre os requisitos não funcionais, destacam-se:

- arquitetura separada em frontend, backend e banco de dados;
- interface responsiva para uso em desktop e dispositivos móveis;
- validações consistentes no frontend e no backend;
- armazenamento seguro de credenciais por hash de senha;
- endpoints administrativos protegidos por autenticação;
- banco relacional com migrations versionadas;
- código organizado em camadas e componentes reutilizáveis;
- documentação local da API em ambiente de desenvolvimento.

D. Casos de Uso

A Fig. 1 apresenta a visão geral dos casos de uso. O ator Cliente acessa a página inicial, navega pelo catálogo, consulta detalhes, realiza simulação e entra em contato com a empresa. O ator Administrador realiza login e executa operações de CRUD de serviços, categorias e contatos.

Os casos de uso mais relevantes são a simulação de orçamento e a gestão de serviços. No primeiro, o cliente informa as medidas do ambiente e seleciona os opcionais desejados. No segundo, o administrador define os parâmetros que serão utilizados pela simulação. Esses dois fluxos são complementares, pois a precisão e a utilidade do orçamento público dependem diretamente da qualidade do cadastro administrativo.

E. Regras de Negócio

A principal regra de negócio é o cálculo da estimativa. Cada serviço possui uma unidade de medida e um modo de medição. Para serviços cuja unidade é o metro quadrado, o modo de medir pode ser teto, parede individual, cômodo inteiro ou uma escolha do usuário entre parede e cômodo.

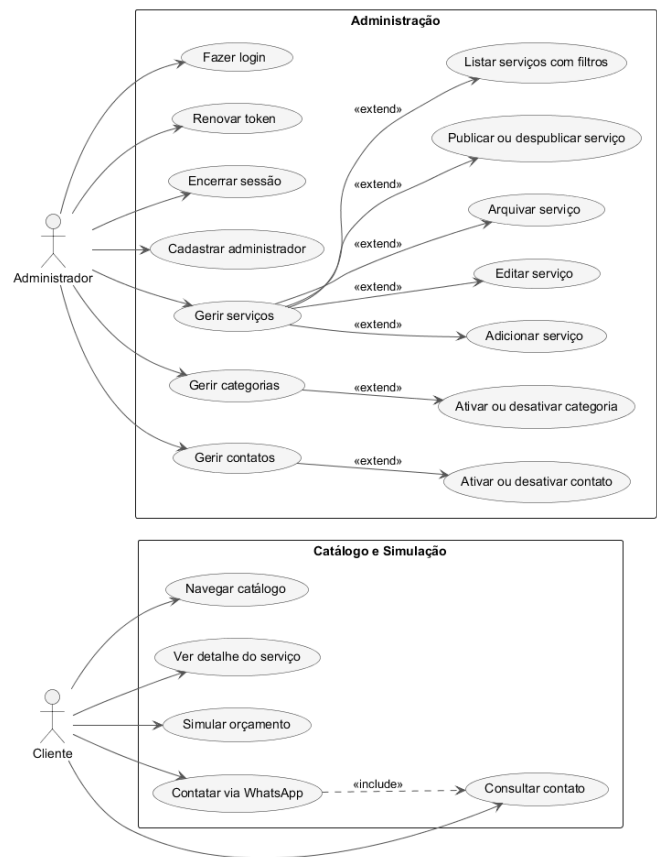


Fig. 1. Diagrama de casos de uso do sistema.

No caso de teto, como em um serviço de rebai-xamento, o cálculo multiplica comprimento por largura. Para parede individual, multiplica largura por altura. Para cômodo inteiro, soma as áreas de todas as paredes do ambiente. Quando a escolha do usuário é permitida, ele informa as medidas de uma parede específica ou do cômodo inteiro, e o sistema calcula a área de acordo com a opção selecionada.

Para serviços em metro linear, o sistema utiliza a metragem manual multiplicada pelo valor unitário.

O cálculo final separa os preços entre mão de obra e material, permitindo apresentar ao cliente duas perspectivas de preço. Essa divisão reflete a possibilidade de contratar apenas a execução do serviço ou incluir também o fornecimento dos materiais pela empresa. O sistema calcula o subtotal sem material, o subtotal com material e o total final, aplicando valor mínimo quando necessário.

Entretanto, o sistema não oferece subtotal apenas com material, pois a mão de obra é parte

obrigatória do serviço prestado pela empresa.

O sistema também aplica valor mínimo por serviço quando o subtotal sem material fica abaixo do mínimo configurado. O objetivo é representar situações em que o deslocamento e a execução de um serviço pequeno não justificam a cobrança estritamente proporcional à área.

Outra regra importante é a publicação de serviços. Serviços em rascunho ou arquivados não aparecem no catálogo público, mas podem ser mantidos no banco para edição posterior. Essa abordagem evita exclusão física desnecessária e permite controlar o que está visível ao cliente.

V. ARQUITETURA, MODELAGEM E IMPLEMENTAÇÃO

A. Arquitetura Geral

A arquitetura segue o modelo em três camadas. O frontend é uma aplicação React executada no navegador web. Ele consome a API via HTTP, obtendo ou enviando dados, exibe os dados públicos, gerencia os formulários e protege as páginas administrativas. O backend é uma API REST em ASP.NET Core, concentrando as regras de negócio e os fluxos de autenticação, validação e persistência de dados. A camada de dados é composta pelo PostgreSQL, cujo esquema é versionado por Liquibase.

Em ambiente local de desenvolvimento, o frontend é servido pelo Vite e utiliza proxy para as rotas da API.

O acesso da API ao banco PostgreSQL é feito por meio do Entity Framework Core. O Liquibase executa os changelogs e garante que o banco esteja atualizado antes da execução completa do sistema.

B. Frontend

O frontend foi organizado em rotas públicas, rota de login e rotas administrativas protegidas. As rotas públicas incluem:

- `/`: página inicial;
- `/catalogo`: catálogo de serviços;
- `/servicos/:slug`: detalhe de serviço específico;
- `/simulador`: simulador de orçamento;
- `/contato`: contatos públicos.

As rotas administrativas ficam sob `/admin` e incluem a página inicial, os serviços, as categorias

e os contatos. O componente de rota protegida verifica o estado de autenticação e redireciona usuários não autenticados para a página de login.

A interface utiliza componentes reutilizáveis para botões, campos, seletores, cards (cartões de informação), badges (indicadores de status), switches (interruptores de ativação de funcionalidades), estados vazios, estados de erro, skeletons (estruturas de carregamento) e mensagens de validação. Essa organização reduz duplicação de código e mantém consistência visual entre telas públicas e administrativas.

No simulador, a interface permite criar múltiplos cômodos, renomeá-los, adicionar serviços, selecionar opcionais, informar medidas e calcular a estimativa consolidada. Cada item do orçamento é enviado à API conforme o serviço e suas medidas. Depois, o frontend consolida os resultados por cômodo e exibe totais gerais.

C. Backend

O backend expõe endpoints públicos e administrativos. Entre as rotas públicas, destacam-se:

- `GET /public/servicos`;
- `GET /public/servicos/{slug}`;
- `GET /public/contatos`;
- `POST /public/simulacoes`;
- `GET /api/health`.

As rotas de autenticação incluem:

- `POST /auth/login`;
- `POST /auth/refresh`;
- `POST /auth/logout`.

As rotas administrativas incluem:

- `/admin/servicos`;
- `/admin/categorias`;
- `/admin/contatos`.

Esses endpoints exigem token Bearer válido, gerado a partir do login administrativo.

A API foi estruturada com controllers (controladores de entrada de requisições), services (serviços para regras de negócio) e repositories (repositórios para acesso a dados).

O middleware de exceção padroniza respostas de erro caso ocorra alguma exceção não tratada, e o FluentValidation valida os DTOs de entrada.

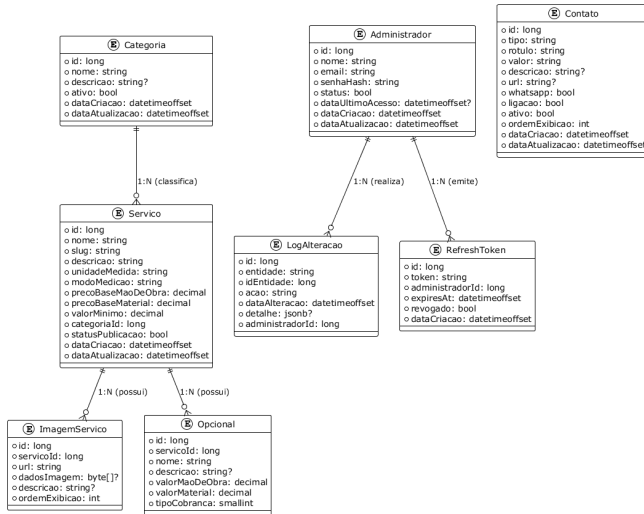


Fig. 2. Diagrama Entidade-Relacionamento.

D. Banco de Dados

O modelo de dados inclui entidades para categoria, serviço, imagem de serviço, opcional, administrador, refresh token, contato e log de alteração. A Fig. 2 apresenta o diagrama entidade-relacionamento.

A entidade *Servico* concentra os dados necessários para catálogo e cálculo: nome, slug, descrição, unidade de medida, modo de medição, preço base de mão de obra, preço base de material, valor mínimo, categoria e status de publicação. A entidade *ImagemServico* armazena imagens por URL ou dados binários. A entidade *Opcional* representa adicionais configuráveis por serviço, com valores separados de mão de obra e material e tipo de cobrança unitária ou por metro.

A entidade *Categoria* organiza os serviços. A entidade *Contato* armazena os canais públicos da empresa, com tipo, rótulo, valor, URL, flags para WhatsApp e ligação, status e ordem de exibição. As entidades *Administrador* e *RefreshToken* dão suporte à autenticação administrativa. A entidade *LogAlteracao* foi modelada como base para rastreabilidade futura.

E. Modelagem de Classes e Componentes

A Fig. 3 resume as classes de domínio e seus relacionamentos. O modelo reflete a separação entre dados cadastrais, parâmetros comerciais e autenticação. Essa organização permite que o catálogo público seja derivado diretamente de serviços

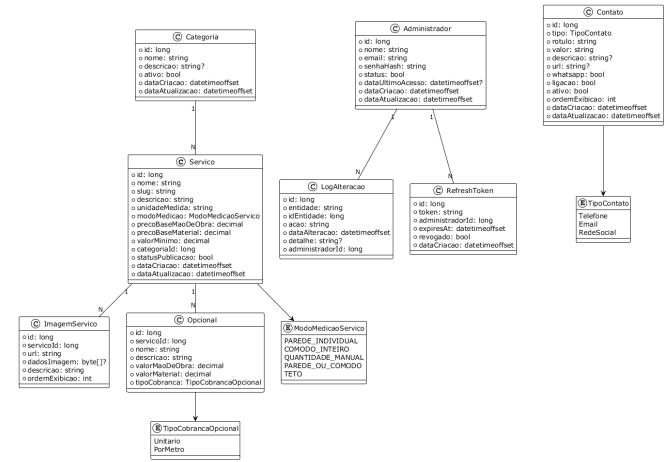


Fig. 3. Diagrama de classes do sistema.

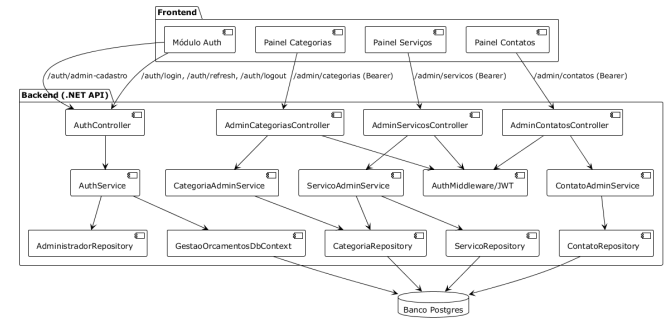


Fig. 4. Componentes envolvidos na gestão administrativa de serviços.

publicados e que o simulador utilize os mesmos parâmetros mantidos no painel administrativo.

Os diagramas de componentes também foram utilizados para representar a separação entre frontend, API, services, repositories e banco de dados. Esses diagramas foram elaborados com apoio da notação PlantUML [4]. A Fig. 4 destaca os componentes envolvidos no fluxo administrativo de serviços.

F. Fluxos de Interação

No fluxo de simulação, o cliente seleciona ou pré-carrega um serviço, informa medidas compatíveis com o modo de medição, seleciona opcionais e solicita o cálculo. O frontend realiza validações iniciais e envia os dados para a API. A API carrega o serviço publicado, resolve o modo de medição, calcula quantitativos, soma itens de mão de obra e material, aplica valor mínimo quando necessário e retorna o resumo da estimativa.

No fluxo administrativo, o administrador realiza login, recebe tokens de sessão e passa a acessar endpoints protegidos. Ao criar ou editar um serviço, a API valida categoria, nome, slug, preços, imagens e opcionais. A publicação pode ser alterada independentemente, permitindo manter rascunhos antes de tornar o serviço visível no catálogo.

G. Modelo de Cálculo

O cálculo parte dos símbolos C para comprimento, L para largura, H para altura, Q para quantidade, P_{mo} para preço de mão de obra, P_{mat} para preço de material e V_{min} para valor mínimo. As principais fórmulas são:

$$\begin{aligned} A_{teto} &= C \times L \\ A_{parede} &= L \times H \times Q \\ A_{comodo} &= 2 \times (L + C) \times H \\ S_{sem} &= (Q_b \times P_{mo}) + O_{mo} \\ S_{com} &= S_{sem} + (Q_b \times P_{mat}) + O_{mat} \\ T_{sem} &= \max(S_{sem}, V_{min}) \\ T_{com} &= \begin{cases} V_{min}, & \text{se o valor mínimo for aplicado} \\ S_{com}, & \text{caso contrário} \end{cases} \end{aligned}$$

Em que Q_b representa a quantidade base calculada para o serviço. Para serviços por metro linear, Q_b é a quantidade manual informada. Para serviços por metro quadrado, Q_b é a área calculada conforme o modo de medição. Opcionais por metro utilizam Q_b ; opcionais unitários utilizam a quantidade informada pelo cliente ou uma unidade quando não houver quantidade específica.

VI. INTERFACES E FUNCIONAMENTO

A. Página Inicial e Catálogo

A página inicial apresenta a empresa, chamadas para simular orçamento e acessar o catálogo, serviços em destaque e um resumo do funcionamento do processo. O objetivo é conduzir o cliente rapidamente para a escolha do serviço ou para a simulação. A Fig. 5 mostra a tela inicial implementada.

O catálogo lista serviços publicados, exibindo imagem, nome, descrição resumida, unidade de

cobrança, preços de referência e botões de navegação. A Fig. 6 apresenta essa tela no sistema implementado.

B. Detalhamento de Serviço

A tela de detalhe apresenta descrição completa, imagens, unidade de medida, valores, opcionais e chamada para iniciar uma simulação daquele serviço. Com essas informações, o cliente consegue compreender melhor o serviço antes de informar medidas. A Fig. 7 mostra a tela implementada.

C. Simulador de Orçamento

O simulador é o principal fluxo público do sistema. Ele permite que o cliente organize o orçamento por cômodos, adicione serviços a cada ambiente e informe medidas compatíveis com o modo de medição. A interface também permite limpar os campos, adicionar novos cômodos e refazer a simulação. A Fig. 8 apresenta o formulário preenchido.

Após o cálculo, a tela de resumo exibe os itens calculados, subtotais, totais com e sem material, taxa mínima quando aplicada e botão de envio pelo WhatsApp. A Fig. 9 apresenta o resumo gerado pelo sistema.

D. Contato Público

A página de contato exibe canais ativos cadastrados no painel administrativo. O objetivo é evitar contato desatualizado no código e permitir que a empresa altere telefone, e-mail ou redes sociais diretamente pelo painel. A Fig. 10 apresenta a tela pública de contato.

E. Login e Painel Administrativo

O acesso ao painel administrativo ocorre por login. Após autenticação, o administrador acessa um layout próprio, com navegação para serviços, categorias e contatos. A proteção ocorre no frontend por rota protegida e no backend por autorização dos endpoints. As Fig. 11 e 12 mostram a autenticação e a tela inicial do painel.

F. Gestão de Serviços

A gestão de serviços é o fluxo administrativo mais completo. A listagem permite busca, filtro por status, paginação e ações de publicar, despublicar, editar e arquivar. A Fig. 13 apresenta a listagem



Fig. 5. Tela inicial do sistema público, com apresentação da empresa e chamadas para catálogo e simulador.

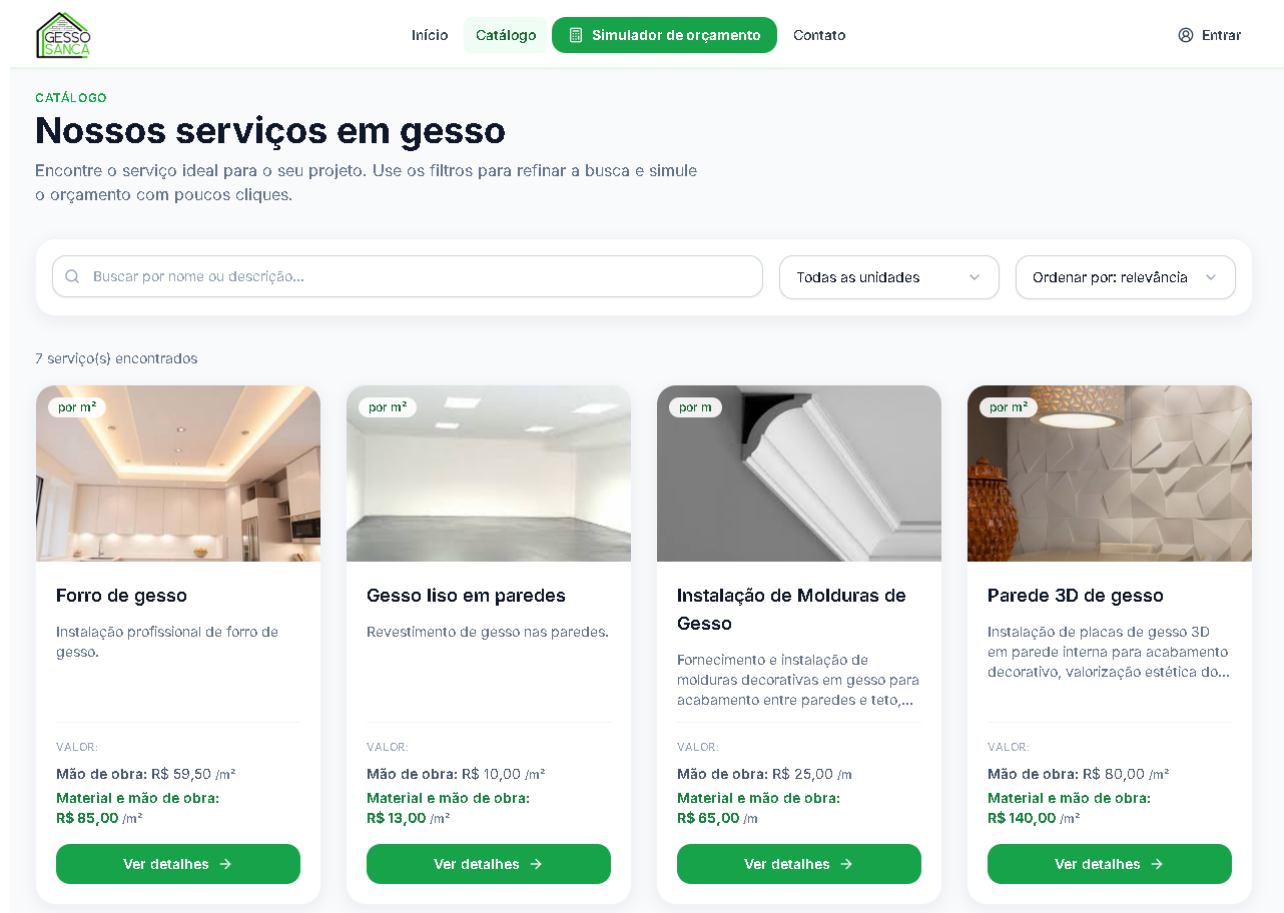


Fig. 6. Tela de catálogo de serviços no sistema implementado.

[← Voltar ao catálogo](#)


Sanca de gesso aberta

Execução de sanca aberta em gesso, preparada para receber iluminação indireta, fita de LED ou mangueira luminosa. Ideal para criar efeito decorativo no teto, destacar ambientes e trazer uma iluminação mais aconchegante e moderna.

PREÇO BASE

Sem material:

R\$ 100,00 /m

Com material:

R\$ 150,00 /m

Valor mínimo do serviço: R\$ 150,00

Informe suas medidas

Metros lineares (m)

Opcionais disponíveis



Fita de led **POR METRO**

+ R\$ 20,00 /m

Fita de led que fica dentro da sanca, provendo uma iluminação indireta. Multiplicado pela metragem calculada do serviço.



Spots de led

+ R\$ 20,00

Lâmpadas de led anexadas à sanca, provendo iluminação direta e visual moderno.

[Simular orçamento](#)

[Adicionar ao simulador multi-serviços](#)

Fig. 7. Tela de detalhamento de serviço no sistema implementado.

administrativa de serviços. A tela de cadastro e edição inclui seções para informações do serviço, preços, modo de medição, status de publicação, galeria de imagens e opcionais, conforme mostrado nas Fig. 14 e 15.

G. Gestão de Categorias e Contatos

As telas de categorias e contatos complementam a administração do catálogo. Categorias permitem organizar serviços e controlar quais grupos permanecem ativos. Contatos permitem manter os canais públicos exibidos ao cliente e utilizados no envio de simulações pelo WhatsApp. As Fig. 16 e 17 apresentam essas telas administrativas.

[Início](#)[Catálogo](#)[Simulador de orçamento](#)[Contato](#)[Entrar](#)[← Voltar ao catálogo](#)

Simulador de orçamento

Organize seu orçamento por cômodos (ex.: "Banheiro do andar de cima", "Quarto do João"). Em cada cômodo você pode adicionar quantos serviços quiser, inclusive repetidos.

COMODO 1

Banheiro do andar de cima

OK X

1

Forro de gesso

m² Ver detalhes

Informe as medidas

Como medir?

Largura do teto *

2,5

m

Comprimento do teto *

2

m

Opcionais

☒ Luminária embutida

Instalação de ponto para luminária embutida

Quantidade: 1

+ R\$ 55,00

☐ Saco

+ R\$ 10,00

Calcular estimativa

Limpar todos os campos

Fig. 8. Formulário de simulação preenchido, com serviço, medidas e opcionais selecionados.

[Início](#)[Catálogo](#)[Simulador de orçamento](#)[Contato](#)[Entrar](#)

Organize seu orçamento por cômodos (ex.: "Banheiro do andar de cima", "Quarto do João"). Em cada cômodo você pode adicionar quantos serviços quiser, inclusive repetidos.



CÔMODO 1

Banheiro do andar de cima

SÓ MÃO DE OBRA

R\$ 484,75

MATERIAL + MÃO DE OBRA

R\$ 692,50

ITEM	QTD.	UNIDADE	VALOR UNIT.	SÓ MÃO DE OBRA	MATERIAL + MÃO DE OBRA
Forro de gesso	7,50	m²	R\$ 59,50	R\$ 446,25	R\$ 446,25
↳ Luminária embutida (mão de obra)	1,00	serviço	R\$ 38,50	R\$ 38,50	R\$ 38,50
↳ Luminária embutida (material)	1,00	serviço	R\$ 16,50	—	R\$ 16,50
Total do serviço				R\$ 484,75	R\$ 692,50

Total da estimativa

Subtotal dos serviços (mão de obra)

R\$ 484,75

Subtotal dos serviços (material + mão de obra)

R\$ 692,50

TOTAL ESTIMADO SÓ MÃO DE OBRA

R\$ 484,75

TOTAL ESTIMADO MATERIAL + MÃO DE OBRA

R\$ 692,50

Estimativa válida por 15 dias.

**Importante**

Estimativa preliminar sujeita à visita técnica Esta estimativa **não é vinculante** e serve apenas como referência inicial. Os valores podem variar após visita técnica ao local.



Enviar pelo WhatsApp

[Refazer simulação](#)[Voltar ao catálogo](#)

Fig. 9. Tela de resumo do orçamento estimado após o cálculo da simulação.

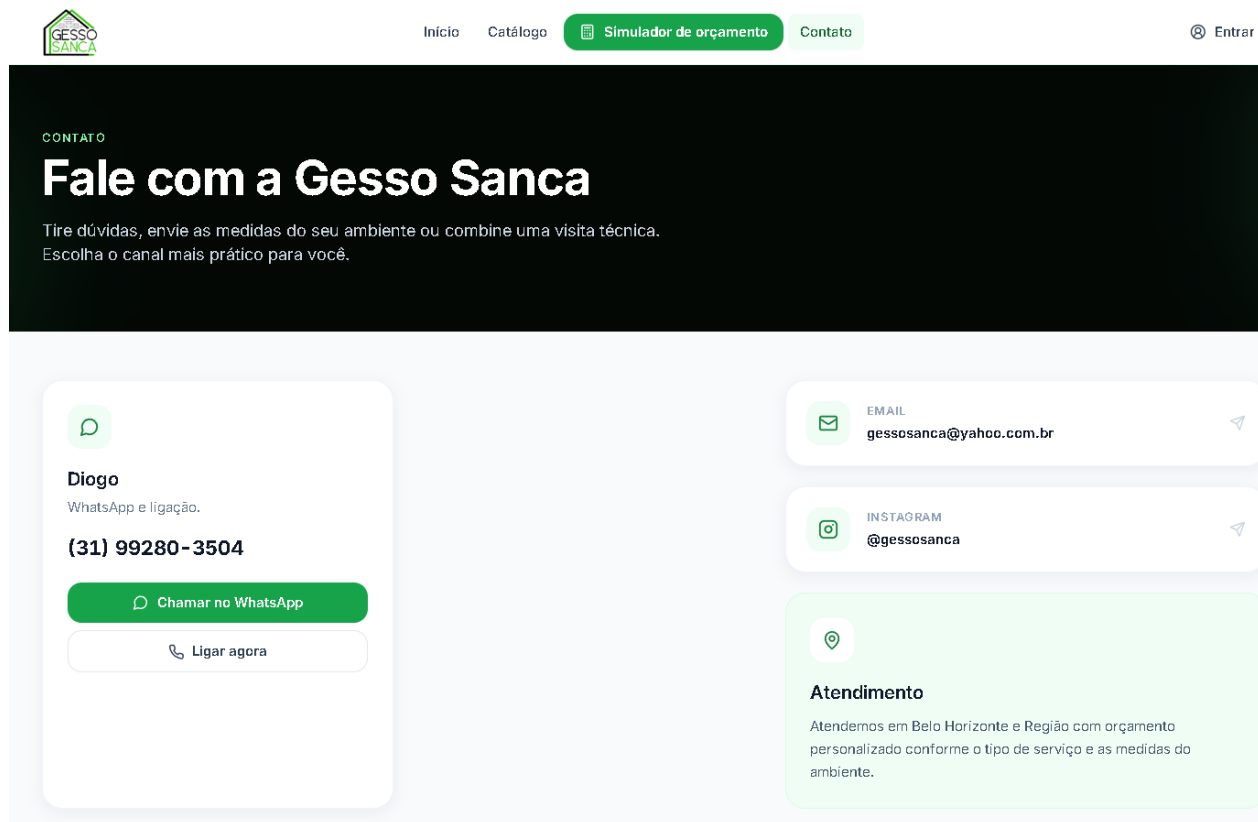


Fig. 10. Tela pública de contato com canais de atendimento da empresa.

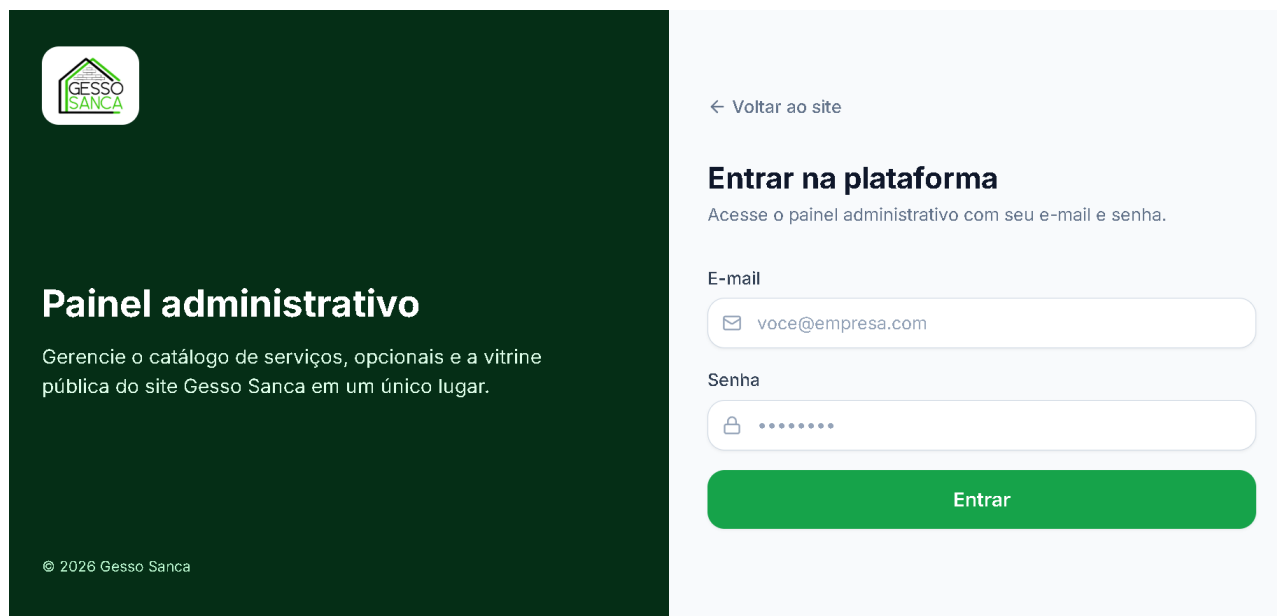


Fig. 11. Tela de autenticação do administrador.

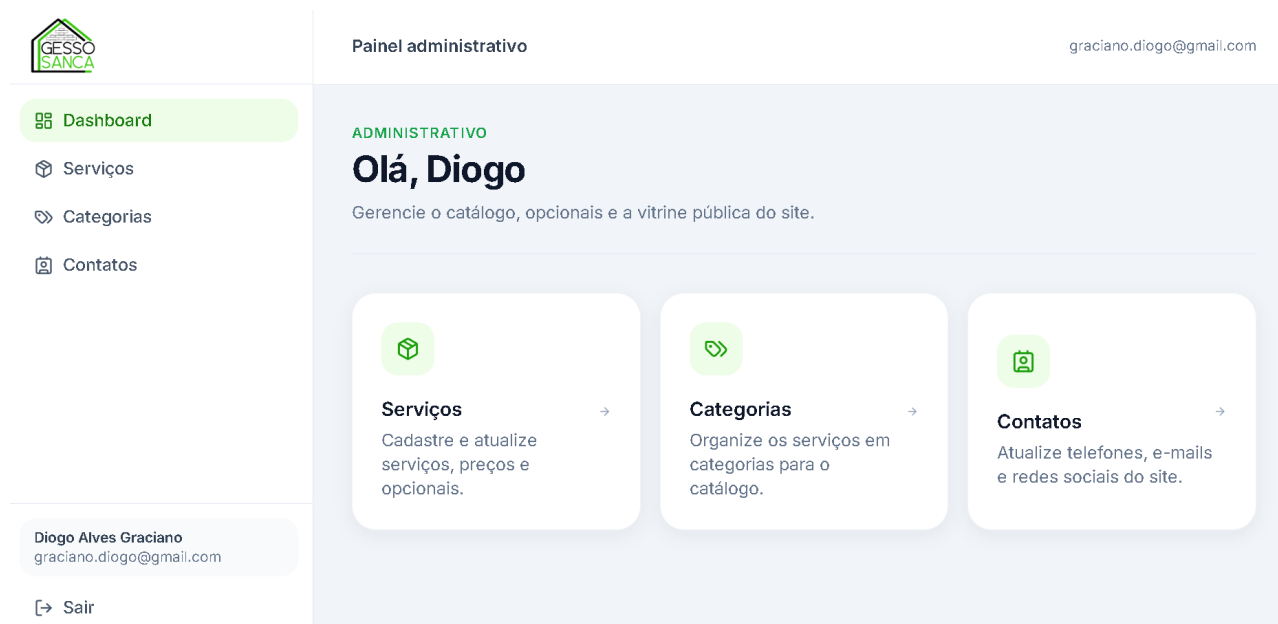


Fig. 12. Tela inicial da área administrativa após login.



Fig. 13. Tela de listagem administrativa de serviços cadastrados.



Dashboard

Serviços

Categorias

Contatos

Painel administrativo

graciano.diogo@gmail.com

NOVO SERVIÇO

Cadastrar novo serviço

Preencha os dados, organize imagens e configure opcionais. Você pode salvar como rascunho ou publicar diretamente.

← Voltar

Os campos marcados com * são obrigatórios. Adicione imagens de qualidade para melhor visualização do serviço.

Informações do Serviço

Nome do Serviço *

Ex.: Forro de gesso liso

0/150 caracteres

Descrição *

Descreva detalhadamente o serviço, incluindo acabamentos, aplicação e diferenciais...

Categoria *

Selecione a categoria

Selecione a categoria do serviço.

Unidade de Cobrança *

Selecione a unidade

Cancelar

Salvar

Salvar e Publicar

Fig. 14. Tela de cadastro ou edição de serviço, com dados gerais, medição e preços.

Painel administrativo

graciano.diogo@gmail.com

Selecione a categoria do serviço.

Forma de medição *

Selecione a forma de medição

Defina quais campos o cliente vai preencher no simulador.

Preço Base - Mão de Obra **

R\$ R\$ 0,00

Valor mão de obra por unidade selecionada

Preço Base - Material **

R\$ R\$ 0,00

Valor material por unidade selecionada

Valor Mínimo *

R\$ R\$ 0,00

Valor mínimo cobrado por orçamento

Galeria de Imagens

Adicione até 10 imagens. Formatos: JPEG, PNG, WebP. Tamanho máximo: 2 MB por imagem.

Anaste imagens aqui ou
selecione do seu computador

JPG, PNG ou WebP até 2 MB - máximo de 10 imagens

Cancelar

Salvar

Salvar e Publicar

Fig. 15. Continuação da tela de cadastro ou edição de serviço, com imagens, opcionais e ações de salvamento.

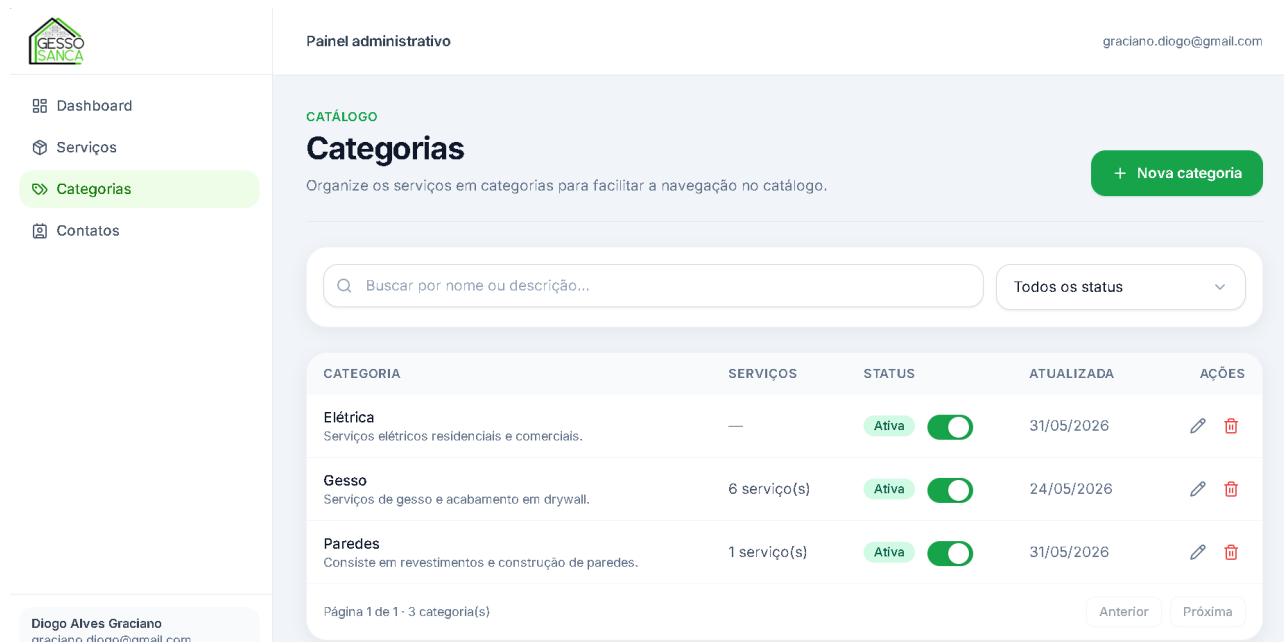


Fig. 16. Tela administrativa de gestão de categorias de serviço.

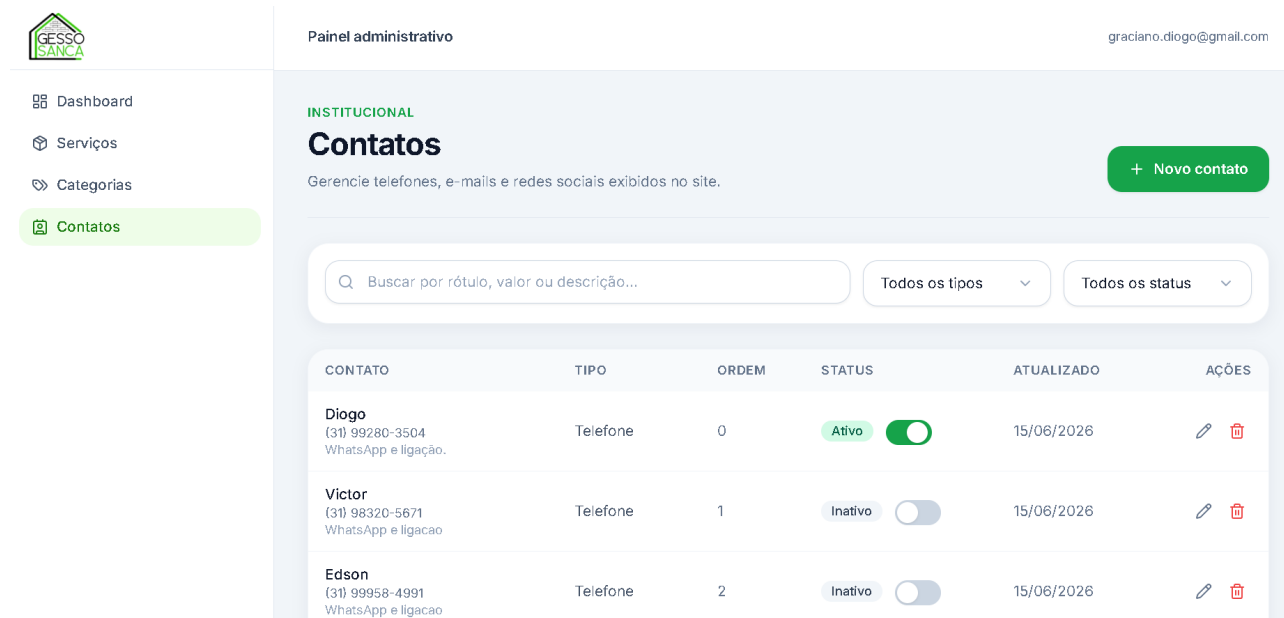


Fig. 17. Tela administrativa de gestão dos contatos públicos.

VII. VALIDAÇÃO, TESTES E RESULTADOS

A. Testes Automatizados

Foram implementados testes automatizados no projeto de testes da API. A cobertura se concentrou nas regras de negócio presentes na API. As principais funcionalidades testadas incluem:

- regras de listagem, criação, atualização e publicação de serviços administrativos;
- comportamento do catálogo público e obtenção de serviço publicado;
- conversão de imagens em base64;
- validações de DTOs de imagens, serviços, categorias e contatos.

Os testes utilizam as ferramentas xUnit e NSubstitute, permitindo isolar dependências de repositório e validar o comportamento das services. Essa estratégia cobre as regras de negócio mais importantes, verifica o funcionamento esperado e reduz o risco de regressões em alterações futuras.

B. Validação Funcional

A validação funcional considerou os fluxos principais do sistema. No fluxo público, verificou-se a navegação pela página inicial, listagem de serviços publicados, abertura de detalhes, preenchimento de simulação, cálculo da estimativa e geração de mensagem para WhatsApp. No fluxo administrativo, verificou-se login, listagem de serviços, criação e edição de serviço, publicação/despublicação, gestão de categorias e gestão de contatos.

Também foram observados cenários de erro, como possibilidade de catálogo vazio, falha de rede ou de servidor, formulário sem campos obrigatórios preenchidos, medidas inválidas, serviço inexistente e tentativa de acesso administrativo sem autenticação. O frontend apresenta estados de carregamento, erro e vazio, enquanto a API retorna mensagens já tratadas para cada caso.

C. Resultados Obtidos

O principal resultado foi a entrega de um MVP funcional que integra o processo de divulgação e orçamento inicial. O sistema permite que a empresa mantenha seus serviços sem alterar código, publique ou oculte itens do catálogo, cadastre opcionais e atualize preços. Para o cliente, oferece uma jornada completa: conhecer a empresa, consultar serviços, estimar valores e iniciar contato.

Em relação ao processo anterior, o sistema reduz a quantidade de perguntas repetitivas no primeiro atendimento e fornece uma estimativa imediata. A padronização dos modos de medição também reduz inconsistências, pois cada serviço define previamente quais campos devem ser usados no cálculo.

Do ponto de vista técnico, a arquitetura adotada mostrou-se adequada ao escopo. A separação entre frontend, API e banco permitiu organizar responsabilidades e testar regras de negócio sem depender da interface. O uso de Liquibase registrou a evolução do banco e facilitou a etapa de desenvolvimento, evitando que fosse necessário recriar o banco de dados a cada alteração de esquema.

D. Limitações

Apesar dos resultados obtidos, algumas limitações permanecem. A estimativa gerada depende da qualidade das medidas fornecidas pelo cliente, que pode não saber medir corretamente o ambiente. Além disso, fatores como estado da obra, acesso ao local, necessidade de acabamento especial e condições estruturais não são capturados integralmente pelo formulário.

Outra limitação é que o sistema ainda não registra historicamente cada orçamento gerado como oportunidade comercial. O envio pelo WhatsApp facilita o contato, mas não substitui uma gestão completa de funil de vendas. Também não foram implementados relatórios analíticos para mensurar conversão ou demanda por serviço.

Também não foram implementados gestão dos orçamentos gerados, status de atendimento ou registro de conversas, funcionalidades que poderiam aproximar o sistema de um módulo simples de CRM. A integração com meios de pagamento, envio de propostas em PDF e agendamento de visita técnica também não foram contemplados.

Todas essas funcionalidades são consideradas para evoluções futuras do sistema, mas não foram inseridas no MVP por limitações de escopo e tempo.

E. Discussão

A experiência de desenvolvimento evidencia que sistemas para pequenos negócios precisam equilibrar simplicidade operacional e flexibilidade. Um

painel muito complexo poderia dificultar o uso pela empresa, enquanto um sistema rígido demais exigiria alterações de código para cada mudança comercial. A modelagem com serviços, categorias, opcionais, modos de medição e contatos configuráveis buscou esse equilíbrio.

Também se observou que o simulador precisa comunicar claramente seu caráter preliminar. A estimativa é útil para reduzir incerteza inicial, mas não elimina a necessidade de avaliação técnica. Por isso, o sistema exibe aviso de que os valores estão sujeitos à confirmação após visita técnica.

VIII. CONCLUSÕES E TRABALHOS FUTUROS

A. Conclusões

Este trabalho apresentou o desenvolvimento de um sistema web para divulgação e gestão de orçamentos de serviços de gesso. O projeto enfrentou um problema prático: a dependência de processos manuais e pouco padronizados para apresentar serviços e gerar estimativas iniciais. O sistema implementado integra catálogo público, detalhamento de serviços, simulador de orçamento, contato via WhatsApp e painel administrativo.

Os objetivos definidos foram atendidos. Os requisitos foram levantados e documentados, os principais artefatos de modelagem foram produzidos, a arquitetura em três camadas foi implementada e os fluxos centrais foram disponibilizados em um MVP funcional. A aplicação permite manutenção administrativa dos dados, cálculo padronizado de estimativas e separação entre valores de mão de obra e material.

Do ponto de vista acadêmico, o projeto permitiu aplicar conceitos de engenharia de software em um cenário real, percorrendo desde a análise do problema até a implementação e validação. Do ponto de vista prático, o sistema oferece à empresa uma base para melhorar sua presença digital, agilizar o atendimento inicial e evoluir seus processos comerciais.

B. Trabalhos Futuros

Como trabalhos futuros, propõe-se implementar histórico de orçamentos gerados, permitindo que cada simulação se torne uma oportunidade acompanhável pela empresa. Também seria útil criar cadastro de clientes, status de atendimento

e registro de conversas, aproximando o sistema de um módulo simples de CRM.

Outra evolução relevante é a criação de relatórios gerenciais, como serviços mais simulados, ticket médio, taxa de conversão e períodos de maior demanda. Também podem ser adicionadas funcionalidades de agendamento de visita técnica, integração com meios de pagamento, envio de propostas em PDF e personalização do simulador para outros segmentos da construção civil.

Por fim, recomenda-se ampliar os testes automatizados, incluindo testes de integração da API com banco de dados e testes end-to-end dos principais fluxos do frontend. Essas melhorias aumentariam a confiabilidade do sistema à medida que novas funcionalidades fossem adicionadas.

REFERÊNCIAS

- [1] P. Kotler and K. L. Keller, *Marketing Management*, 14th ed. Upper Saddle River, New Jersey, EUA: Pearson Education, Inc., 2012. [Online]. Disponível em: https://library.uniq.edu.iq/storage/books/file/kotler_keller_-_marketing_management_14th_edition/1666787488kotler_keller_-_marketing_management_14th_edition.pdf Acesso em: 28 jun. 2026.
- [2] I. Sommerville, *Engenharia de Software*, 9a ed. São Paulo: Pearson, 2011. [Online]. Disponível em: <https://archive.org/details/engenharia-de-software-10e> Acesso em: 28 jun. 2026.
- [3] B. Pasini and D. Gray, “Construction Cost Estimating: A Step-By-Step Guide,” Procore, 2025. [Online]. Disponível em: <https://www.procore.com/library/construction-estimating> Acesso em: 28 jun. 2026.
- [4] PlantUML, “PlantUML Language Reference Guide.” [Online]. Disponível em: <https://plantuml.com> Acesso em: 28 jun. 2026.
- [5] React, “React Documentation.” [Online]. Disponível em: <https://react.dev/learn> Acesso em: 28 jun. 2026.
- [6] Vite, “Vite Documentation.” [Online]. Disponível em: <https://vite.dev/guide/> Acesso em: 2 jul. 2026.
- [7] Axios, “Axios Documentation.” [Online]. Disponível em: <https://axios-http.com/docs/intro> Acesso em: 2 jul. 2026.
- [8] Microsoft, “Documentação da API Web ASP.NET Core.” [Online]. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/web-api/> Acesso em: 28 jun. 2026.
- [9] PostgreSQL Global Development Group, “PostgreSQL Documentation.” [Online]. Disponível em: <https://www.postgresql.org/docs/> Acesso em: 28 jun. 2026.
- [10] Liquibase, “Liquibase Documentation.” [Online]. Disponível em: <https://docs.liquibase.com/> Acesso em: 28 jun. 2026.
- [11] WhatsApp, “How to use click to chat.” [Online]. Disponível em: <https://faq.whatsapp.com/5913398998672934> Acesso em: 28 jun. 2026.