

**SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL**

**“ ALMIRANTE TAMANDARÉ ”**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**ANA BEATRIZ QUEIROZ COSTA**

**ARTHUR PASCOAL RIBEIRO**

**ENZO GOLDNER CASTRO**

**FELIPE PEREIRA NOVAIS**

**ISAAC MENEZES SILVA**

**ÁRTEMIS: SISTEMA MÓVEL PARA GESTÃO DE PEDIDOS EM RESTAURANTES**

Tecnologia para otimização do fluxo de pedidos

São Bernardo do Campo, SP

2025

ANA BEATRIZ QUEIROZ COSTA

ARTHUR PASCOAL RIBEIRO

ENZO GOLDNER CASTRO

FELIPE PEREIRA NOVAIS

ISAAC MENEZES SILVA

## **ÁRTEMIS: SISTEMA MÓVEL PARA GESTÃO DE PEDIDOS EM RESTAURANTE**

Tecnologia para otimização do fluxo de pedidos

Trabalho de conclusão de curso apresentado ao curso técnico em desenvolvimento de sistemas do SENAI - “Almirante Tamandaré”, como requisito para obtenção do título de técnico em desenvolvimento de sistemas.

Orientador: Prof. Pedro Castilho Cardoso

São Bernardo do Campo, SP

2025

COSTA, Ana Beatriz Queiroz.  
RIBEIRO, Arthur Pascoal.  
CASTRO, Enzo Goldner.  
NOVAIS, Felipe Pereira.  
SILVA, Isaac Menezes.

**ÁRTEMIS: SISTEMA MÓVEL PARA GESTÃO DE PEDIDOS EM RESTAURANTES** - Tecnologia para otimização do fluxo de pedidos /Ana Beatriz Queiroz Costa, Arthur Pascoal Ribeiro, Enzo Goldner Castro, Felipe Pereira Novais, Isaac Menezes Silva.

São Bernardo do Campo, 2025. 67 p.

Trabalho de Conclusão de Curso (Técnico em Desenvolvimento de Sistemas) – Serviço Nacional de Aprendizagem Industrial – SENAI Almirante Tamandaré, 2025.

I. Aplicativo móvel. 2. Pizzaria. 3. Pedidos online.

I. Orientador: Pedro Castilho Cardoso.

II. Serviço Nacional de Aprendizagem Industrial – SENAI Almirante Tamandaré.

III. Desenvolvimento de aplicação mobile.

## **ERRATA**

ANA BEATRIZ QUEIROZ COSTA, ARTHUR PASCOAL RIBEIRO, ENZO GOLDNER CASTRO,  
FELIPE PEREIRA NOVAIS, ISAAC MENEZES SILVA.

### **ÁRTEMIS: SISTEMA MÓVEL PARA GESTÃO DE PEDIDOS EM RESTAURANTES.**

Tecnologia para otimização do fluxo de pedidos. 2025. 67 p. Trabalho de Conclusão de Curso – Serviço Nacional de Aprendizagem Industrial – SENAI Almirante Tamandaré, São Bernardo do Campo, 2025.

**Folha**

**Linha**

**Onde de Lê**

**Leia- se**

ANA BEATRIZ QUEIROZ COSTA

ARTHUR PASCOAL RIBEIRO

ENZO GOLDNER CASTRO

FELIPE PEREIRA NOVAIS

ISAAC MENEZES SILVA.

## **ÁRTEMIS: SISTEMA MÓVEL PARA GESTÃO DE PEDIDOS EM RESTAURANTES**

Tecnologia para otimização do fluxo de pedidos

Trabalho de conclusão de curso apresentado ao curso técnico em desenvolvimento de sistemas do SENAI - “Almirante Tamandaré”, como requisito para obtenção do título de técnico em desenvolvimento de sistemas.

Aprovados em: \_\_\_/\_\_\_/\_\_\_

Prof. Orientador Pedro Castilho Cardoso – SENAI Almirante Tamandaré

Julgamento \_\_\_\_\_ Assinatura \_\_\_\_\_

Profº Edgard Coutinho Silva – SENAI Almirante Tamandaré

Julgamento \_\_\_\_\_ Assinatura \_\_\_\_\_

Profº Ms. Vinícius Baratieri Suterio – SENAI Almirante Tamandaré

Julgamento \_\_\_\_\_ Assinatura \_\_\_\_\_

Profº Thiago Francisco Andrade de Lima – SENAI Almirante Tamandaré

Julgamento \_\_\_\_\_ Assinatura \_\_\_\_\_

Dedicamos este trabalho aos nossos familiares, que foram a base e a fonte de inspiração durante toda a nossa caminhada no curso Técnico em Desenvolvimento de Sistemas.

Em especial, aos nossos professores e ao SENAI Almirante Tamandaré, por nos fornecerem o conhecimento e as ferramentas necessárias para a concretização deste projeto.

## **AGRADECIMENTOS**

A prática deste Trabalho de Conclusão de Curso representa a realização de uma importante etapa em nossa formação. Gostaríamos de expressar nossa sincera gratidão às pessoas e a instituição que contribuíram diretamente para este resultado.

Ao Professor Pedro Castilho Cardoso, nosso Orientador, por sua inestimável dedicação e pelos conhecimentos técnicos e direcionamentos que foram cruciais para o desenvolvimento e finalização deste trabalho.

Ao Professor Edgard Coutinho Silva, pelo apoio e valiosas contribuições oferecidas ao longo da jornada do projeto.

A todos os demais professores do Serviço Nacional de Aprendizagem Industrial – SENAI Almirante Tamandaré, pela excelência na formação e por compartilharem seu saber.

Aos nossos colegas de curso, pelo companheirismo e troca de experiências.

E, por fim, às nossas famílias, pela compreensão e incentivo contínuo.

## **RESUMO**

A crescente demanda por atendimento personalizado tem impulsionado o uso de soluções digitais que ampliem a autonomia do cliente e mitiguem intervenções manuais. Este trabalho apresenta o desenvolvimento de uma aplicação móvel para o gerenciamento de pedidos personalizados em ambiente interno de pizzaria, fundamentada na arquitetura cliente-servidor e comunicação via APIs REST. A solução foi desenvolvida em React Native com TypeScript, com modelagem de dados via MER/DER e persistência no Supabase integrado ao Prisma ORM. O processo de concepção adotou práticas de UX Design e metodologias ágeis. Embora validada através de um estudo de caso no setor de pizzarias, a arquitetura proposta é agnóstica e escalável, permitindo adaptação a outros contextos de varejo que demandem fluxos complexos de personalização. Os resultados demonstram a viabilidade técnica da solução em promover eficiência operacional, usabilidade e redução de falhas humanas no processamento de pedidos.

**Palavras-chave:** desenvolvimento de *software*; atendimento individualizado; sistema de pedidos; usabilidade.

## **ABSTRACT**

The growing demand for personalized customer service has driven the adoption of digital solutions that enhance user autonomy and reduce manual interventions. This work presents the development of a mobile application for managing customized orders within a pizzeria environment, grounded in a client–server architecture and communication through REST APIs. The solution was developed in React Native with TypeScript, employing data modeling through ERM/ERD and persistence using Supabase integrated with the Prisma ORM. The design process adopted UX Design practices and agile methodologies. Although validated through a case study in the pizzeria sector, the proposed architecture is agnostic and scalable, allowing adaptation to other retail contexts that require complex customization flows. The results demonstrate the technical feasibility of the solution in promoting operational efficiency, usability, and the reduction of human error in order processing.

Keywords: software development; individualized service; order system; usability.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Brainstorm e Referências	20
Figura 2 - Processo do Fluxograma	21
Figura 3 - Fluxograma da aplicação	22
Figura 4 - Fluxograma da aplicação	23
Figura 4 - Diagrama de Casos de Uso	25
Figura 6 - Organograma	33
Figura 7 – Quadro Kanban (Plataforma Trello)	35
Figura 8 - Guia de Estilos	49
Figura C.1 - Páginas de Cadastro (Placeholder e Preenchido)	64
Figura C.2 - Páginas de Login (Placeholder e Preenchido)	65
Figura C.3 - Páginas de Menu (Categorias: Bebidas e Pizzas)	66
Figura C.4 - Páginas de Detalhes e Personalização	67
Figura C.5 - Páginas de Leitura do QR Code e Validação da Mesa	68
Figura C.6 - Páginas de Sacola e Checkout com Pagamento	69
Figura C.7 - Alerta de Pagamento bem-sucedido e Página de Pedidos Relacionados à Mesa	70
Figura C.8 - Páginas de Status do Pedido (Enviado à Cozinha e Item Entregue)	71
Figura C.9 - Alerta de Pedidos Finalizados e Página de Produtos Favoritos (Vazio)	72
Figura C.10 - Card de Produto Favoritado na Página de Favoritos e no Dashboard	73
Figura C.11 - Páginas de Pedir Ajuda “Chamar Garçom” e Erro de Leitura do QR Code	74
Figura C.12 - Páginas de Pedir Ajuda Bem-Sucedida e Perfil	75
Figura C.13 - Página web de cadastro	77
Figura C.14 - Página web de login	78
Figura C.15 - Página web de painel de pedidos	79
Figura C.16 - Página web de status do pedido e produto	80
Figura C.17 - Página web de criação de categoria	81
Figura C.18 - Página web de solicitação de ajuda	81
Figura C.19 - Página web de criação do produto	82
Figura C.20 - Página web de criação de ingredientes	83
Figura C.21 - Figma do projeto	83
Figura C.22 - Protótipo do projeto	83
Figura D.1 - Qr code Repositório do Projeto	84

## **LISTA DE TABELAS**

Tabela 1 - Regras de Negócio para o Gerenciamento de Pedidos dos Clientes	26
Tabela 2 - Requisitos funcionais	28
Tabela 3 - Requisitos Não-Funcionais da Aplicação	30
Tabela 5 - Cronograma de Execução do Projeto (Sprints e Focos).	36
Tabela 6 - Entidade Usuário Cliente	39
Tabela 7 - Entidade Usuário Funcionário	39
Tabela 8 - Entidade Função	40
Tabela 9 - Entidade Mesa	41
Tabela 10 - Entidade Produto	41
Tabela 11 - Entidade Pagamento	42
Tabela 11 - Entidade Pagamento	43
Tabela 12 - Entidade Pedido	43
Tabela 14 - Entidade Itens	44
Tabela 14 - Entidade Itens	45
Tabela 15 - Entidade Favoritos	45
Tabela 16 - Entidade Categorias	46
Tabela 17 - Entidade Produto Ingrediente	46
Tabela 17 - Entidade Produto Ingrediente	47
Tabela 18 - Entidade Extra	47
Tabela 19 - Entidade Item Extra	48
Tabela 20 - Entidade Produto Ingrediente Removido	48

## **LISTA DE ABREVIATURAS E SIGLAS**

ANSI	American National Standards Institute
API	Application Programming Interface
BA	Business Analyst
BYOD	Bring Your Own Device
CDN	Content Delivery Network
CPF	Cadastro de Pessoas Físicas
DBA	Database Administrator
DER	Diagrama Entidade-Relacionamento
FK	Foreign Key
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
JSX	JavaScript XML
MER	Modelo Entidade-Relacionamento
MVP	Minimum Viable Product
ORM	Object-Relational Mapping
PK	Primary Key
QA	Quality Assurance
RF	Requisitos Funcionais
RN	Regras de Negócio
RNF	Requisitos Não Funcionais
SQL	Structured Query Language
TCC	Trabalho de Conclusão de Curso
UI	User Interface
UX	User Experience

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>14</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
<b>3. METODOLOGIA DE DESENVOLVIMENTO.....</b>	<b>19</b>
3.1 CONCEPÇÃO INICIAL (BRAINSTORMING).....	20
3.2 LEVANTAMENTO DE REQUISITOS.....	21
3.2.1 Fluxograma.....	21
3.2.2 Diagrama de Casos de Uso.....	24
3.2.3 Regras de negócio.....	26
3.2.4 Requisitos funcionais.....	28
3.2.5 Requisitos não-funcionais.....	29
3.3 BACKLOG DO PRODUTO E HISTÓRIA DE USUÁRIOS.....	30
3.4 PLANEJAMENTO E GERENCIAMENTO DO DESENVOLVIMENTO.....	32
3.4.1 Organograma.....	32
3.4.2 Ferramentas de Gestão do Fluxo.....	34
3.4.3 Cronograma de Execução.....	35
3.5 MODELAGEM DE DADOS.....	36
3.5.1 Modelo Entidade-Relacionamento (MER).....	37
3.5.2. Diagrama Entidade-Relacionamento (DER).....	37
3.5.3. Dicionário de Dados( Modelo físico de dados).....	38
3.6. DESIGN GRÁFICO.....	48
3.6.1. Identidade visual.....	49
3.6.2. Prototipagem da imagem e validação visual.....	50
3.7. TECNOLOGIAS UTILIZADAS.....	51
3.7.1. Plataforma Front-end e Desenvolvimento Móvel.....	51
3.7.2. Backend e Gerenciamento de Dados.....	53
3.7.3. Ferramentas de Apoio, Versionamento e Gestão.....	54
<b>4. CONCLUSÃO.....</b>	<b>56</b>
<b>REFERÊNCIAS.....</b>	<b>57</b>
<b>GLOSSÁRIO.....</b>	<b>58</b>
<b>APÊNDICE A - MODELO DE ENTIDADE-RELACIONAMENTO (MER).....</b>	<b>62</b>
<b>APÊNDICE B - DIAGRAMA DE ENTIDADE-RELACIONAMENTO (DER).....</b>	<b>63</b>
<b>APÊNDICE C - DOCUMENTAÇÃO VISUAL DA INTERFACE.....</b>	<b>64</b>
<b>APÊNDICE D - REPOSITÓRIO DO PROJETO.....</b>	<b>84</b>

## 1. INTRODUÇÃO

O panorama mercadológico contemporâneo é marcado por uma crescente demanda por atendimento individualizado e personalizado. Essa tendência acompanha o que Ghobril, Benedetti e Fragoso (2021) descrevem como a “virada digital” na experiência de consumo, na qual o usuário passa a buscar soluções que se ajustem às suas preferências e ofereçam maior autonomia. Clientes esperam cada vez mais serviços que se adaptem às suas necessidades específicas, o que, por sua vez, resulta na redução da dependência de intervenções constantes de equipes externas e no fortalecimento de sistemas que ampliam a participação ativa do consumidor no processo de atendimento.

Esse movimento não se limita aos serviços tradicionais, mas permeia profundamente o desenvolvimento de soluções digitais, que devem ser responsivas, acessíveis e orientadas à experiência do usuário. Norman (2013) aponta que, em interfaces modernas, a expectativa central do usuário é poder realizar suas tarefas de forma clara, fluida e com mínima fricção, o que reforça a necessidade de sistemas bem projetados e intuitivos.

Nesse cenário de customização e alta exigência, a criação de sistemas eficientes torna-se um imperativo, especialmente em setores que lidam diretamente com a gestão de pedidos sob medida. A ineficiência no processamento de solicitações individualizadas pode gerar atrasos, erros e insatisfação — problemas amplamente discutidos por Sommerville (2011), ao tratar da importância da definição precisa de requisitos para garantir consistência e confiabilidade no desenvolvimento de software. Assim, emerge a necessidade de sistemas capazes de automatizar, organizar e otimizar fluxos de trabalho complexos.

Com o objetivo de atender a essa demanda, o presente trabalho descreve a criação de uma aplicação para o gerenciamento de pedidos individualizados em uma pizzaria. O desenvolvimento buscou conceber um sistema que não apenas acomodasse a complexidade dos pedidos customizados — como tamanhos, combinações e ingredientes — mas que também oferecesse alta usabilidade, clareza e autonomia ao usuário. Para isso, foram aplicadas metodologias e tecnologias modernas, conforme discutido por autores que abordam desenvolvimento ágil e engenharia de software, como Sutherland (2014) e Sommerville (2011), integrando os conhecimentos adquiridos ao longo da formação técnica em Desenvolvimento de Sistemas.

## 2. FUNDAMENTAÇÃO TEÓRICA

A transformação digital tem exercido influência profunda e contínua sobre o modo como consumidores interagem com estabelecimentos gastronômicos, sobretudo em pizzarias e restaurantes. A busca por rapidez, autonomia e previsibilidade tem conduzido o setor à adoção de tecnologias capazes de eliminar gargalos, evitar falhas humanas e melhorar a experiência do cliente. Como apontam Ghobril, Benedetti e Fragoso (2021), soluções digitais inseridas no contexto alimentar contribuem para otimizar operações internas, reduzir ruídos de comunicação e aumentar a qualidade do atendimento, características cada vez mais valorizadas em ambientes de consumo contemporâneos. Dessa forma, sistemas de autoatendimento passam a ser percebidos não como um luxo tecnológico, mas como um componente natural das expectativas de quem frequenta estabelecimentos físicos.

Essa tendência pode ser observada em grandes redes que incorporaram totens de autoatendimento como parte estruturante de suas operações. Empresas como *McDonald's* e *Bob's* utilizaram esses dispositivos para reduzir filas e redistribuir a carga de trabalho da equipe, permitindo que os funcionários focassem em funções de preparo e entrega. No segmento das pizzarias, marcas como *Domino's* e *Pizza Hut* ampliaram o alcance de seus aplicativos, embora tradicionalmente voltados ao *delivery*. Contudo, novas abordagens têm ganhado espaço ao deslocar o autoatendimento de equipamentos fixos para dispositivos pessoais. Nos últimos anos, observa-se o avanço de modelos que eliminam a necessidade de *totens* físicos ao permitir que o próprio *smartphone* do cliente — uma tendência conhecida como *BYOD* (*Bring Your Own Device*) — se torne o canal primário de interação. Segundo Quintela (2016), a apropriação cotidiana de tecnologias digitais cria cenários em que o próprio usuário assume papel ativo no consumo de informações e serviços, o que torna estratégias *BYOD* especialmente atraentes em ambientes comerciais.

Essa evolução direciona o autoatendimento para experiências mais flexíveis, personalizáveis e acessíveis, integrando recursos como QR *Codes*, cardápios dinâmicos, notificações em tempo real e sistemas adaptados ao contexto de uso. Nesse ecossistema emergente insere-se o projeto Ártemis, desenvolvido especificamente para o ambiente interno de pizzarias. Diferente de aplicações voltadas apenas ao *delivery*, a solução adota o modelo *BYOD* ao transformar o *smartphone* do cliente em seu próprio terminal de pedidos, sem exigir

totens ou equipamentos adicionais. Essa abordagem acompanha o que Santos (2017) descreve como o novo paradigma da interação digital: o usuário não deseja apenas consumir conteúdo, mas participar ativamente do processo. Assim, ao possibilitar que o cliente visualize o cardápio, personalize o pedido, acompanhe o status e receba atualizações diretamente em seu dispositivo, o Ártemis representa um passo significativo em direção à modernização do atendimento presencial.

A fim de viabilizar uma solução com essas características, torna-se necessário compreender fundamentos essenciais de arquitetura de software. O modelo *cliente-servidor* permanece como referência para aplicações distribuídas, conforme descreve Tanenbaum (2011). Nesse modelo, a camada cliente é responsável pela interação visual e pela captura de ações do usuário, enquanto o servidor realiza o processamento da lógica de negócios, a validação de informações e o gerenciamento dos dados persistentes. A comunicação entre essas partes ocorre por meio de *APIs*, em geral estruturadas segundo os princípios REST, formulados por Fielding (2000). Por serem *stateless*, essas interfaces tratam cada requisição de modo independente, garantindo escalabilidade, simplicidade conceitual e baixo acoplamento entre os componentes do sistema.

Para o tráfego de dados entre cliente e servidor, utiliza-se frequentemente o formato JSON, que apresenta leveza, legibilidade e compatibilidade com diferentes plataformas. No lado do cliente, bibliotecas como *Axios* tornam esse processo mais confiável ao simplificar o envio de requisições HTTP, tratando erros, tempos de resposta e padronizando o consumo das APIs. Essa camada de comunicação desempenha papel significativo em sistemas de autoatendimento, nos quais notificações e atualizações em tempo real influenciam diretamente a expectativa do usuário sobre velocidade e precisão.

A construção da interface é sustentada por tecnologias específicas para desenvolvimento móvel. O *framework React Native* destaca-se por permitir a criação de aplicações nativas para *Android* e *iOS* utilizando componentes reutilizáveis e uma arquitetura baseada em estados. A abordagem declarativa evita comportamentos inesperados e facilita a manutenção, especialmente em projetos que evoluem rapidamente. De acordo com Sommerville (2011), reutilização e clareza são elementos fundamentais para a qualidade do software, e *frameworks* multiplataforma contribuem para alcançar esses resultados.

Complementando essa previsibilidade, o *TypeScript* adiciona tipagem estática ao *JavaScript*, permitindo que inconsistências sejam detectadas em tempo de desenvolvimento(antes da execução) e oferecendo maior integridade ao código.

Ferramentas como *Expo Go* otimizam esse processo ao permitir testes imediatos em dispositivos reais e fornecer acesso facilitado a APIs nativas, reduzindo a complexidade de configurações e acelerando o ciclo de desenvolvimento. A combinação entre *React Native*, *TypeScript* e *Expo Go* compõe um conjunto moderno de tecnologias centrado em produtividade, consistência e escalabilidade.

Antes da etapa de implementação, entretanto, é indispensável compreender o processo de levantamento de requisitos. Essa fase identifica o que o sistema deve realizar e sob quais condições deve funcionar. Sommerville (2011) classifica requisitos funcionais como aqueles que descrevem comportamentos específicos — registrar pedidos, exibir cardápios e atualizar status, por exemplo — enquanto os requisitos não funcionais determinam qualidades essenciais, como desempenho, acessibilidade, segurança e usabilidade. Regras de negócio complementam esse universo ao definir políticas, restrições e parâmetros operacionais que regem o funcionamento interno de uma pizzaria, como disponibilidade de produtos, horários de atendimento, associações entre mesas e pedidos e regras para formas de pagamento.

A documentação desses elementos é frequentemente representada por Diagramas de Casos de Uso, que mapeiam interações de alto nível entre atores e funcionalidades, e por fluxogramas, que representam graficamente o fluxo de ações. Esses recursos auxiliam na visualização da lógica e no alinhamento entre equipe técnica, gestores e usuários, constituindo ferramentas essenciais para reduzir ambiguidades antes da implementação.

No campo da modelagem de dados, o Modelo Entidade-Relacionamento (MER) constitui a representação conceitual das entidades relevantes para o sistema, como Cliente, Pedido, Produto, Mesa e Categoria. Conforme Coronel e Rob (2011), esse modelo auxilia na organização de informações complexas ao identificar atributos e relacionamentos relevantes para o domínio. O Diagrama Entidade-Relacionamento (DER), por sua vez, transforma essa visão abstrata em estruturas lógicas, definindo cardinalidades, chaves primárias e chaves estrangeiras que orientarão a implementação física no banco de dados.

Ferramentas modernas auxiliam na transição entre a modelagem lógica e a estrutura final de armazenamento. O *Prisma* ORM se destaca como intermediário entre a aplicação e o banco de dados, traduzindo comandos em SQL otimizado e adicionando camadas de segurança, tipagem e padronização. Já o Supabase fornece uma plataforma completa baseada em *PostgreSQL*, com autenticação, APIs automáticas e escalabilidade integrada. Essa combinação reflete o que Perkovic (2015) identifica como tendência contemporânea: abstrações que reduzem a complexidade e permitem que o desenvolvedor se concentre na lógica de negócio.

Para aplicações que incluem imagens — como fotografias de produtos, banners ou elementos visuais do cardápio — o armazenamento de mídias torna-se uma necessidade. Plataformas como *Cloudinary* oferecem mecanismos de compressão, manipulação, otimização e distribuição por CDN, assegurando carregamento rápido e consistente mesmo em conexões instáveis. Em aplicações móveis, a performance do conteúdo visual influencia diretamente a satisfação do usuário, tornando o gerenciamento de mídias parte central da arquitetura.

A experiência do usuário (UX) constitui outro pilar fundamental. Segundo Norman (2013), interfaces eficazes devem reduzir a carga cognitiva, tornando as ações previsíveis e coerentes com as expectativas do usuário. No contexto de autoatendimento presencial, isso significa oferecer navegação clara, legibilidade adequada, hierarquia visual consistente e respostas imediatas às interações. A prototipação complementa essa abordagem ao permitir validação precoce de fluxos e telas, evitando retrabalho e garantindo que a solução atenda às necessidades reais do público.

A construção de sistemas modernos também se apoia em práticas de desenvolvimento adequadas. As Metodologias Ágeis, principalmente o *Scrum*, organizam o trabalho em ciclos iterativos que facilitam adaptação e melhoria contínua (SUTHERLAND, 2014). O *Kanban* complementa essa estrutura ao fornecer visão visual do fluxo, evitando sobrecargas e garantindo previsibilidade. O uso do Git como ferramenta de controle de versão registra o histórico completo de modificações do código, permitindo colaboração segura entre desenvolvedores e preservando a integridade do projeto.

Assim, observa-se que a construção de uma aplicação de autoatendimento integrando dispositivos móveis exige a articulação de diversos fundamentos teóricos: evolução do consumo digital, paradigma BYOD, arquitetura *cliente-servidor*, APIs *REST*, bibliotecas de requisição como *Axios*, frameworks móveis como *React Native*, linguagens tipadas como *TypeScript*, ferramentas de apoio como *Expo Go*, processo de Engenharia de Requisitos, modelagem de dados por MER e DER, uso de plataformas modernas como Prisma, Supabase e *Cloudinary*, além de princípios de UX e metodologias ágeis. A integração desses elementos sustenta o desenvolvimento de uma solução capaz de atender às demandas de usuários e estabelecimentos, refletindo o estado atual das tecnologias aplicadas ao setor gastronômico.

### **3. METODOLOGIA DE DESENVOLVIMENTO**

Este capítulo tem como objetivo detalhar o processo de execução do projeto Ártemis, definindo as etapas, a abordagem e os artefatos documentais utilizados para transformar o referencial teórico em uma solução funcional. Dada a natureza dinâmica do desenvolvimento de *software* e a necessidade de adaptação contínua, o projeto foi guiado pelos princípios das Metodologias Ágeis, utilizando o framework *Scrum* para a gestão das *sprints* e o *Kanban* para a visualização do fluxo de trabalho.

O ciclo de desenvolvimento foi estruturado em uma sequência lógica, que garantiu a evolução controlada do produto: da Concepção Inicial à Implementação. O processo se iniciou com a Geração de Ideias (*Brainstorming*), que estabeleceu a visão do produto. Em seguida, as ideias foram formalizadas por meio do Levantamento de Requisitos e Modelagem de Dados, etapas cruciais para definir o comportamento esperado e a estrutura de persistência do sistema.

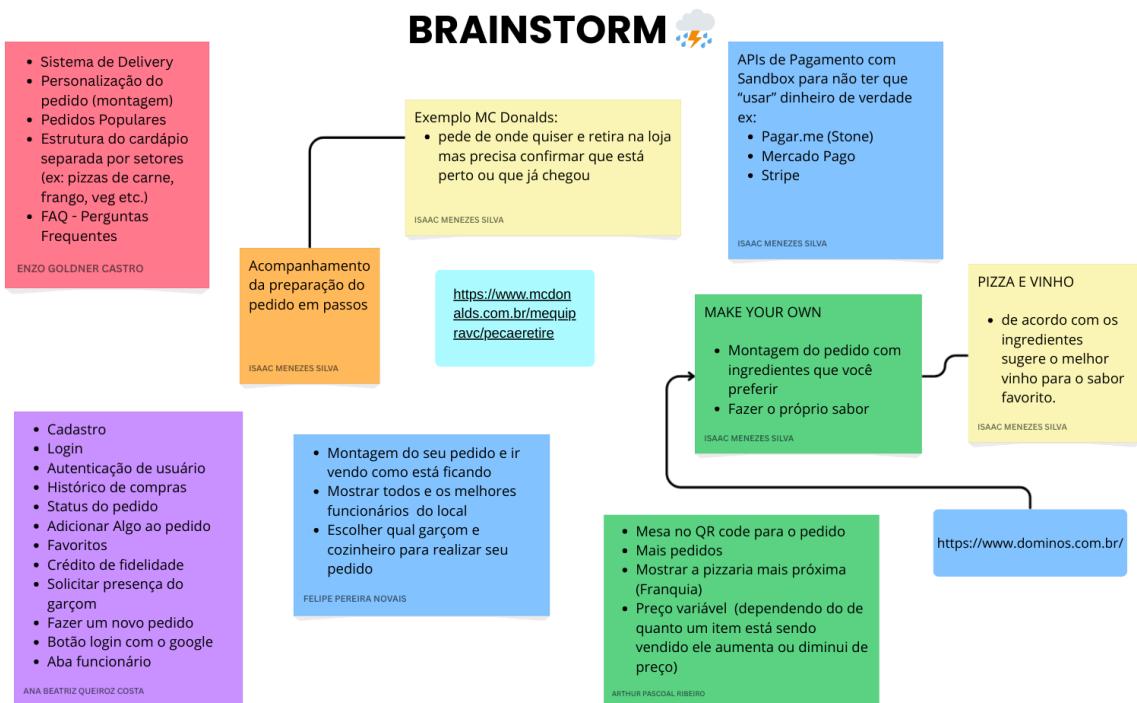
Essa progressão do abstrato ao concreto assegurou que toda a arquitetura fosse planejada e documentada (por meio de requisitos funcionais, não funcionais e diagramas) antes do início da codificação, minimizando retrabalho e garantindo o alinhamento do produto final aos objetivos definidos.

### 3.1 CONCEPÇÃO INICIAL (*BRAINSTORMING*)

A fase inicial do projeto consistiu em uma sessão de *brainstorming* (tempestade de ideias) para definir as funcionalidades centrais da aplicação, além de possíveis diferenciais, sem mapear custos, tempo assim como capacidade e conhecimento técnico, recursos esses que seriam levados em consideração em futuras etapas.

Durante essa etapa, foram explorados módulos cruciais, como os sistemas de cadastro e autenticação de usuário (incluindo *login* social), gerenciamento de pedidos (com histórico e acompanhamento de *status*) e personalização do cardápio (como a montagem de pedidos customizados e filtros). Adicionalmente, foram analisadas as necessidades de infraestrutura, como a integração de APIs de pagamento (a exemplo de Pagar.me e *Stripe*) e o uso de APIs de geolocalização (*Google Maps*) para o sistema de delivery e localização de lojas. Essa discussão inicial foi fundamental para consolidar a visão do produto, inspirando-se em fluxos de mercado já estabelecidos, como os modelos do *McDonald's* e *Domino's* (Figura 1).

**Figura 1 - Brainstorm e Referências**



Fonte: Autoria (2025).

### 3.2 LEVANTAMENTO DE REQUISITOS

A Fundamentação Teórica estabeleceu que a construção de um sistema de autoatendimento eficiente, como o projeto Ártemis, exige uma compreensão precisa dos requisitos e das regras de negócio que regem o funcionamento de uma pizzaria. Conforme discutido com base em Sommerville (2011), a fase de levantamento é crucial para a definição do comportamento do sistema – os requisitos funcionais – e de suas qualidades, como desempenho e usabilidade – os requisitos não funcionais. Para a correta tradução desses requisitos e para o alinhamento da lógica de negócio, a documentação se apoia em ferramentas visuais. O seu Referencial Teórico aponta que a representação por fluxogramas e Diagramas de Casos de Uso é essencial para mapear as interações e reduzir ambiguidades antes da implementação.

#### 3.2.1 Fluxograma

Dando continuidade a este processo, a seguir, são apresentados o fluxo de trabalho essencial da aplicação Ártemis, os quais detalham a sequência lógica de ações e decisões tomadas tanto pelo usuário quanto pelo sistema. Para garantir a clareza e a padronização na leitura desses diagramas, a simbologia adotada segue o padrão universalmente aceito baseado nas Normas ANSI/ISO para diagramas de processo, conforme detalhado na Figura 2.

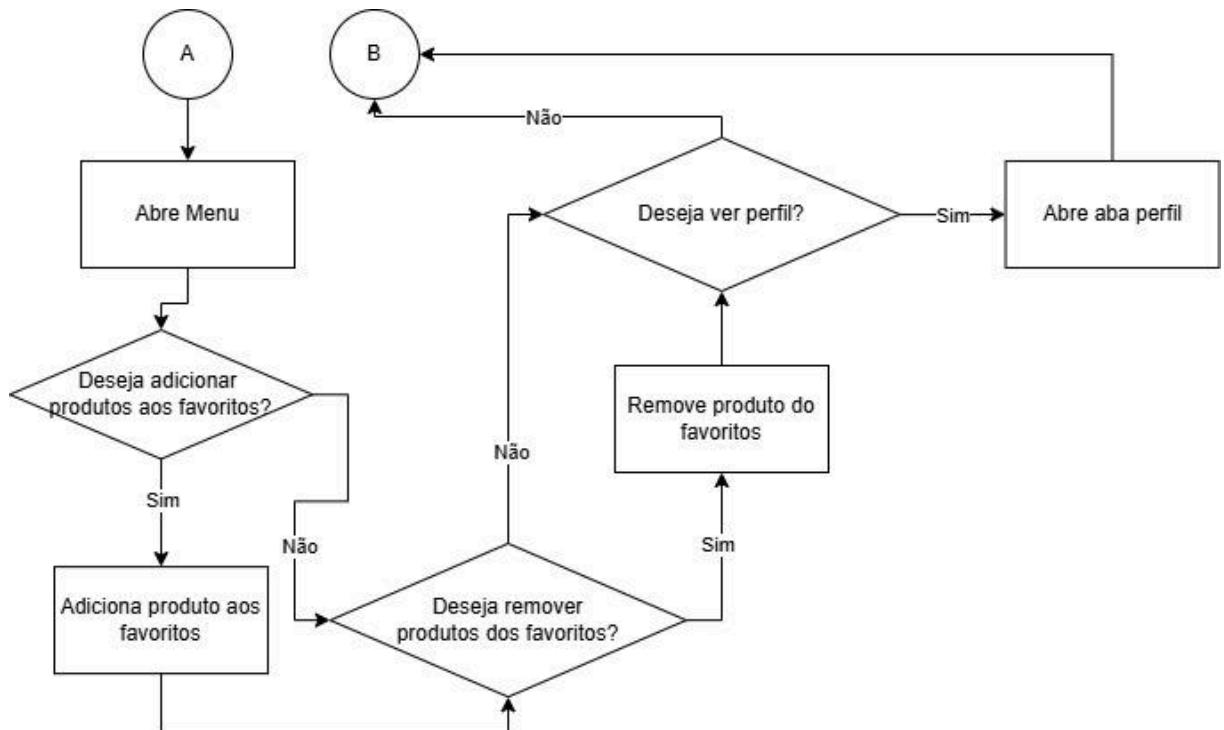
**Figura 2 - Processo do Fluxograma**

	Indica o inicio ou fim do processo
	Indica cada atividade que precisa ser executada
	Indica um ponto de tomada de decisão
	Indica a direção do fluxo
	Indica os documentos utilizados no processo
	Indica uma espera
	Indica que o fluxograma continua a partir desse ponto em outro círculo, com a mesma letra ou número, que aparece em seu interior

Fonte: Não Informada.

Essa simbologia define que o Oval (Terminal) marca o início e o fim do processo; o Retângulo representa uma Ação ou Processo; e o Losango indica um ponto de Decisão no fluxo. Dessa forma, é possível visualizar o percurso do usuário e a lógica interna do sistema, abrangendo as duas principais modalidades de acesso. O fluxograma da aplicação (Figura 3 e 4) detalha as ações disponíveis para usuários, desde o *login* até a conclusão do pagamento e acompanhamento do status. Em suma, este diagrama fornece a base para a compreensão da lógica que será implementada nas camadas Cliente (*React Native*) e Servidor (API REST).

**Figura 3 - Fluxograma da aplicação**



Fonte: Autoria(2025).

**Figura 4 - Fluxograma da aplicação**

### 3.2.2 Diagrama de Casos de Uso

Com o detalhamento dos fluxos operacionais de acesso, com e sem autenticação, o passo seguinte na engenharia de requisitos foi representar graficamente as funcionalidades do sistema sob a perspectiva do usuário. Para isso, empregou-se o Diagrama de Casos de Uso, ferramenta essencial para mapear as interações de alto nível entre os atores envolvidos (Cliente e Funcionário/Sistema) e as funcionalidades principais do Ártemis. Essa representação, conforme discute Sommerville (2011), é fundamental para visualizar a lógica das operações e alinhar expectativas entre os integrantes do projeto antes da etapa de desenvolvimento.

No contexto do Ártemis, o diagrama não apenas descreveu as funcionalidades, mas serviu como instrumento de validação dos requisitos. Ele permitiu identificar redundâncias, dependências e ações obrigatórias, como o fluxo mínimo necessário para realizar um pedido. Além disso, foi possível confirmar que usuários não autenticados deveriam ter acesso simplificado às operações essenciais — decisão que influenciou diretamente a arquitetura do protótipo e o design das telas do totem.

Os elementos gráficos do diagrama — atores, casos de uso e associações — foram utilizados para estruturar de maneira objetiva as ações possíveis dentro do sistema. Da mesma forma, as relações Extend (<<extend>>) e Include (<<include>>) permitiram organizar etapas comuns entre funcionalidades, como “Selecionar Produto”, que aparece repetidamente em diferentes cenários e pôde ser reutilizado sem gerar inconsistências. A organização das relações também evidenciou funcionalidades condicionais, como a etapa de “Adicionar Observações ao Pedido”, ativada apenas quando o cliente opta por personalizações.

Esse mapeamento visual possibilitou sintetizar o escopo funcional do Ártemis e identificar claramente as ações essenciais para seu funcionamento, como Realizar Pedido, Customizar Item e Acompanhar Status. A Figura 4 apresenta essa estrutura de maneira consolidada, oferecendo uma visão sistêmica que guiou o desenvolvimento das telas e a definição dos fluxos de interação.

**Figura 4 - Diagrama de Casos de Uso**

### 3.2.3 Regras de negócio

Antes de formalizar os requisitos do sistema, é fundamental estabelecer as Regras de Negócio que governam o fluxo de trabalho e o comportamento lógico da pizzaria. Estas regras definem as políticas, limitações e parâmetros operacionais do estabelecimento – como a capacidade máxima de customização e a associação de pedidos à mesa –, e servem como premissas não negociáveis que orientam toda a concepção do *software* Ártemis. A Tabela 1 resume as principais regras de negócio que foram identificadas e consideradas durante a fase de análise do projeto.

Tabela 1 - Regras de Negócio para o Gerenciamento de Pedidos dos Clientes

(continua)

CÓD	Regra	Descrição	RN relacionados
RN01	<i>Login</i> de Usuário	O usuário precisa realizar <i>login</i> no aplicativo <i>mobile</i> para gerenciar pedidos.	
RN02	Gestão de Pedidos	O funcionário autenticado no aplicativo <i>web</i> pode gerenciar pedidos.	
RN03	Registro de Pedido por garçom	O sistema fornece a opção de registrar o pedido pelo garçom caso o cliente não tenha acesso a um dispositivo próprio.	RN01, RN02
RN04	Escolha de Itens	O cliente escolhe itens do cardápio para compor o pedido.	RN01, RN02, RN10
RN05	<i>Login</i> da Gerência	Apenas usuários com credenciais de gerente podem acessar funções administrativas.	
RN07	Gerenciamento do Catálogo	Apenas a gerência pode criar, editar ou remover itens do cardápio.	RN06

Fonte: Elaborado pelo grupo (2025).

Tabela 1 - Regras de Negócio para o Gerenciamento de Pedidos dos Clientes

(continuação)

CÓD	Regra	Descrição	RN relacionados
RN09	Informações Obrigatórias dos Itens	Cada item do cardápio deve ter, no mínimo, nome e descrição.	
RN10	Vinculação de Pedido à Mesa	Todo pedido deve estar vinculado a uma mesa, que, por sua vez, deve estar associada a um ou mais clientes.	RN01, RN03
RN11	O pedido só é enviado após pagamento	O pagamento deve ser efetuado e validado antes de o pedido ser enviado para a cozinha	RN01
RN12	Ler QR code com número da mesa	Para realizar o pedido, o cliente deve ler o QR code da mesa	RN10
RN13	Registro de pedido por cliente	Após a escolha dos itens é criado um pedido	RN05
RN14	Manipulação do pedido	O pedido do cliente pode ter itens deletados, adicionados e alterados.	
RN15	Tamanho da Pizza	Toda pizza tem que ter seu tamanho especificado no pedido.	

Fonte: Elaborado pelo grupo (2025).

Tabela 1 - Regras de Negócio para o Gerenciamento de Pedidos dos Clientes

(conclusão)

CÓD	Regra	Descrição	RN relacionados
RN16	Histórico e favoritos	O cliente autenticado pode favoritar e rever seus pedidos anteriores.	
RN17	Cadastro do cliente	O cadastro inclui nome, idade e a opção de adicionar formas de pagamento.	
RN18	<i>Status</i> do pedido	Todos os pedidos devem ser executados em ordem crescente	

Fonte: Autoria (2025).

### 3.2.4 Requisitos funcionais

Com as premissas do negócio estabelecidas, o foco é direcionado para a definição do comportamento do software. Os Requisitos Funcionais (RF) definem, de forma clara e objetiva, as funções e serviços que o sistema Ártemis deve oferecer ao usuário e às equipes da pizzaria. Conforme a classificação de Sommerville (2011), estes requisitos descrevem o que o sistema deve fazer, abrangendo desde a consulta de itens no cardápio até o processamento do pagamento e a atualização do status do pedido. A Tabela 2 detalha e organiza os requisitos funcionais essenciais que nortearam o desenvolvimento de cada funcionalidade da aplicação.

Tabela 2 - Requisitos funcionais

(continua)

CÓD	Regra	Descrição	RN	RF	RNF
RF01	Criação da conta	Cadastrar conta com e-mail e senha do usuário para a criação de conta.	RN02		
RF02	Menu	Exibir menu ao usuário.	RN06, RN07, RN08		RNF05
RF03	Ler QR code com número da mesa	O QR code da mesa informa o número da mesa para o aplicativo.	RN12		

Fonte: Autoria (2025)

Tabela 2 - Requisitos Funcionais da Aplicação

(conclusão)

CÓD	Regra	Descrição	RN	RF	RNF
RF04	Remover itens do cardápio	Exclusão de itens do cardápio do aplicativo.		RN07	
RF05	Adicionar itens no carrinho	Adiciona itens ao carrinho do cliente.		RN14	
RF06	Remover ingredientes	Remover ingredientes de um produto existente.		RN14	
RF07	Chamar um Garçom	Chamar um garçom para a mesa que solicitar.		RN10	
RF08	Acompanhar <i>status</i> do pedido	Mostra em que parte do processo está o pedido.		RN18	
RF09	Favoritar itens	Adiciona itens na página de favoritos.		RN16	
RF10	Ver histórico de pedidos	Abrir página de históricos.		RN16	
RF11	Fazer observações	Adiciona observação no pedido.		RN14	
RF12	Adicionar ingredientes a um produto	Adiciona ingredientes a um produto			
RF13	Realizar o Pagamento	Reliza o pagamento do pedido			

Fonte: Autoria (2025).

### 3.2.5 Requisitos não-funcionais

Após a definição das funcionalidades centrais do sistema, é igualmente crucial estabelecer os requisitos não-funcionais. Estes requisitos, diferentemente dos funcionais, não descrevem o que o sistema faz, mas sim a qualidade com que ele deve executar suas tarefas. Eles englobam aspectos como desempenho (velocidade de resposta), usabilidade (facilidade

de uso e clareza da interface, essencial para o autoatendimento), acessibilidade e segurança (proteção dos dados do cliente e da transação). A Tabela 3 detalha os requisitos não-funcionais estabelecidos para o projeto Ártemis, garantindo que a aplicação não seja apenas funcional, mas também eficiente, confiável e satisfatória para o usuário.

Tabela 3 - Requisitos Não-Funcionais da Aplicação

CÓD	Nome	Descrição	RN
RNF01	Segurança	A senha do usuário deve ser armazenada de forma criptografada para garantir segurança dos dados.	RN01, RN02, RN06
RNF02	Dependência de Internet	O sistema necessita de conexão com a internet para funcionar corretamente.	
RNF03	Autenticação Obrigatória	O acesso ao sistema só será liberado após autenticação via login bem-sucedida.	RN06, RN07, RN08
RNF04	Interface Responsiva	A interface deve se adaptar a diferentes resoluções de tela, garantindo usabilidade em dispositivos variados.	
RNF05	Gerenciamento de Mídias	As imagens e mídias do cardápio serão gerenciadas por meio da plataforma Cloudinary.	RN07, RN09

Fonte: Autoria (2025).

### 3.3 BACKLOG DO PRODUTO E HISTÓRIA DE USUÁRIOS

A partir dos requisitos funcionais e não-funcionais estabelecidos, a aplicação dos conceitos de metodologias ágeis, conforme abordado na fundamentação teórica (Sutherland, 2014), foi essencial para a gestão e priorização do escopo. O *product backlog* foi estruturado com base no formato de histórias de usuário (*user stories*), que transformam o requisito em uma perspectiva de valor para o cliente, seguindo a estrutura padrão. Para garantir que as entregas fossem incrementais e focadas no valor de negócio, as funcionalidades foram classificadas em três níveis de prioridade. As funcionalidades de Alta Prioridade definiram o *Minimum Viable Product* (MVP), focando no *core* da aplicação (fluxo de pedido e autenticação básica). As funcionalidades de Média Prioridade visam o aprimoramento da transação e usabilidade (customização e integração de pagamento), enquanto as de Baixa Prioridade foram alocadas para iterações futuras, focando em enriquecimento da experiência

(como favoritos e histórico). A Tabela 4 detalha o *backlog* inicial do sistema Ártemis, apresentando as histórias de usuário e a respectiva priorização estratégica.

Tabela 4 - Histórias de Usuário e Prioridades do Sistema.

(continua)

Recurso	Prioridade	História de Usuário
Cardápio Digital	Alta	Como cliente, desejo acessar o cardápio digital para visualizar opções de pratos e bebidas.
Integração de Pagamento	Média	Como cliente, desejo efetuar o pagamento diretamente pelo aplicativo, garantindo rapidez e segurança de dados financeiros e pessoais.
Status do Pedido	Média	Como cliente, desejo acompanhar o progresso do pedido em tempo real enquanto aguardo.
Personalização do Pedido	Média	Como cliente, desejo modificar ingredientes do meu pedido, considerando alergias ou preferências.
Criar Pedido do Zero	Média	Como cliente, desejo montar meu pedido manualmente, salvando a configuração para pedidos futuros.
Cadastro e Login	Alta	Como cliente, desejo criar uma conta ou acessar minha conta para salvar dados e histórico de pedidos.
Login com Google	Média	Como cliente, desejo autenticar com minha conta Google para agilizar o acesso.
Favoritos	Baixa	Como cliente, desejo favoritar itens do cardápio para agilizar pedidos futuros.
Solicitar Garçom	Alta	Como cliente no restaurante, desejo solicitar a presença do garçom pelo aplicativo.
Carrinho	Alta	Como cliente, desejo visualizar e gerenciar itens no carrinho, ajustando quantidades e conferindo o total.
Selecionar Mesa via QR Code	Alta	Como cliente no restaurante, desejo escanear o QR Code da mesa para que o pedido seja enviado ao local correto.

Fonte: Autoria (2025)

Tabela 4 - Histórias de Usuário e Prioridades do Sistema.

(continuação)

Recurso	Prioridade	História de Usuário
Cancelar Pedido	Média	Como cliente, desejo cancelar um pedido feito por engano, evitando cobranças ou desperdício de comida.
Autenticação de Usuário	Alta	Como cliente, desejo garantir acesso seguro ao aplicativo por meio de autenticação com código criptografado enviado por e-mail.
Remover Produto do Pedido	Média	Como cliente, desejo remover itens do carrinho antes de finalizar a compra.
Perguntas Frequentes	Baixa	Como cliente, desejo acessar respostas para dúvidas sem depender de suporte.
Perfil e Configurações	Média	Como cliente, desejo editar meus dados e preferências no aplicativo.
Média de Tempo	Baixa	Como cliente, desejo visualizar o tempo estimado de preparo e entrega.

Fonte: Elaborado pelo grupo (2025).

### 3.4 PLANEJAMENTO E GERENCIAMENTO DO DESENVOLVIMENTO

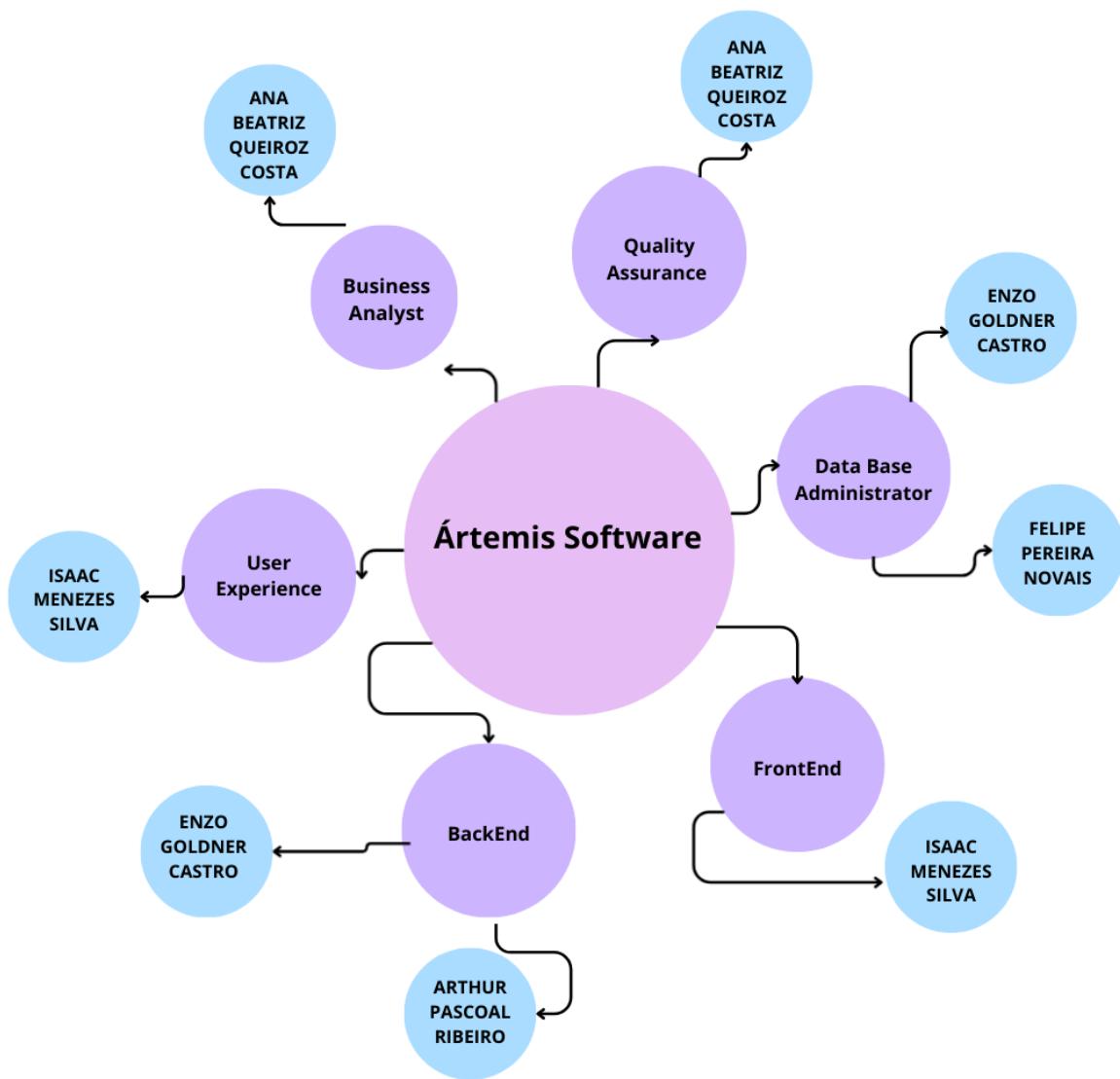
O desenvolvimento da aplicação Ártemis foi orientado pelos princípios do Desenvolvimento Ágil, conforme referenciado na Fundamentação Teórica (Sutherland, 2014). Para garantir a flexibilidade, o foco contínuo no valor de negócio e a adaptação rápida a eventuais mudanças nos requisitos, foi adotada uma abordagem híbrida de Scrum com Kanban (Scrumban). O Scrum forneceu a estrutura de ciclos iterativos (*Sprints*) e papéis bem definidos, enquanto o Kanban complementou essa estrutura ao oferecer uma gestão visual do fluxo de trabalho, limitando o *Work in Progress* (WIP) e otimizando a passagem de tarefas entre as etapas de desenvolvimento, teste e implantação.

#### 3.4.1 Organograma

A gestão eficiente do modelo Scrumban adotado exigiu uma estrutura organizacional clara. O Organograma da Equipe da Ártemis Software foi elaborado com o objetivo de representar de maneira clara a estrutura do projeto, seguindo o modelo funcional, no qual cada

área é definida e separada de acordo com suas funções específicas dentro do processo de desenvolvimento de software. Essa representação é vital para o alinhamento das comunicações e das responsabilidades técnicas, garantindo que os papéis do Scrum (*Product Owner* e *Development Team*) fossem cumpridos. A Figura 6

**Figura 6 - Organograma**



Fonte: Autoria (2025).

ilustra a estrutura da equipe e as interconexões entre seus membros, cujas responsabilidades são detalhadas a seguir:

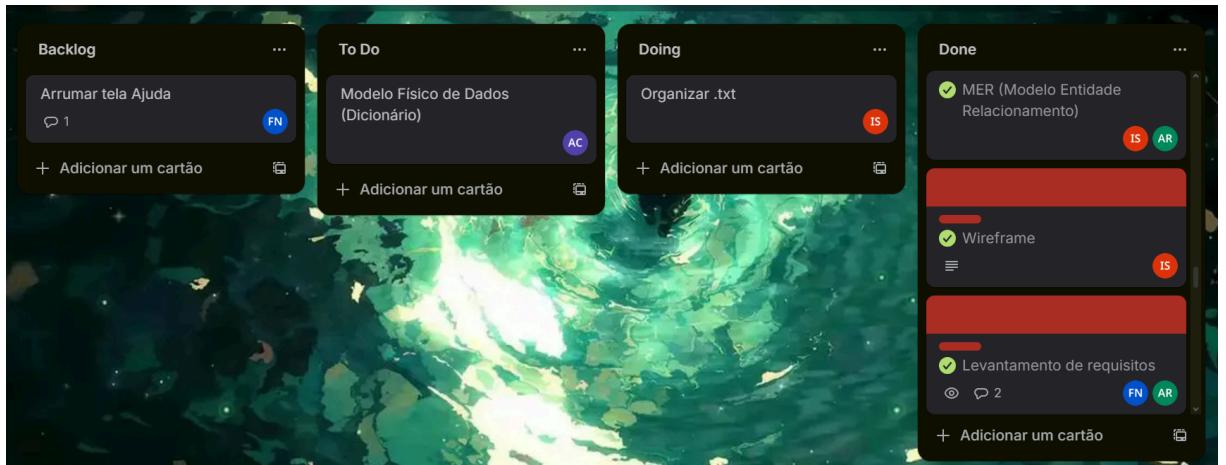
- ***Business Analyst (BA)*** e ***Quality Assurance (QA)***: A integrante Ana Beatriz Queiroz Costa acumulou as funções de BA, responsável pela análise de requisitos e comunicação com o usuário, e QA, atuando na garantia da qualidade por meio de testes e validações.
- ***Database Administrator (DBA)***: A modelagem, organização e administração dos bancos de dados foram de responsabilidade dos integrantes Enzo Goldner Castro e Felipe Pereira Novais.
- ***Front-End e User Experience (UX)***: O desenvolvimento da interface gráfica, garantindo acessibilidade, responsividade e experiência visual do usuário (UX), foi a principal função do integrante Isaac Menezes Silva.
- ***Back-End***: A lógica de negócio, o controle das funcionalidades internas do sistema e a comunicação com o banco de dados foram desenvolvidos pelos integrantes Enzo Goldner Castro e Arthur Pascoal Ribeiro.

Essa estrutura permitiu que os membros da equipe se concentrarem em suas especialidades funcionais, otimizando a produtividade em cada *Sprint* de desenvolvimento, em conformidade com o foco na especialização técnica esperado em projetos ágeis.

### 3.4.2 Ferramentas de Gestão do Fluxo

A fim de operacionalizar o modelo *Scrumban* adotado e garantir a gestão visual do *Product Backlog*, foram utilizadas ferramentas digitais específicas para o gerenciamento do fluxo de trabalho. O uso da plataforma *Trello* foi central nesse processo, uma vez que ela se adequa perfeitamente ao método *Kanban*, facilitando o monitoramento contínuo das tarefas em suas diferentes fases: *Backlog* (tarefas a fazer), *To Do* (prontas para começar), *Doing* (em execução) e *Done* (concluídas). A Figura 7 ilustra o Quadro Kanban digital implementado no *Trello* pela equipe. Esse recurso permitiu visualizar e controlar o estado das Histórias de Usuário durante o desenvolvimento, identificar rapidamente gargalos, balancear a carga de trabalho e assegurar que o desenvolvimento seguisse um fluxo contínuo e eficiente, conforme o princípio *pull* do Kanban. Este controle visual e a limitação implícita do *Work in Progress* (WIP) foram cruciais para a produtividade da equipe e para garantir a entrega das *Sprints* com a qualidade exigida.

**Figura 7 – Quadro Kanban (Plataforma Trello)**



Fonte: Autoria (2025).

### 3.4.3 Cronograma de Execução

O planejamento temporal do projeto foi estabelecido com base na priorização do *Product Backlog* (Tabela 4) e na estimativa de esforço necessária para completar cada Histórias de Usuário. O Cronograma de Execução (Tabela 5) reflete o sequenciamento das *Sprints*, alinhado à metodologia Scrumban. Este cronograma foi utilizado como uma ferramenta central para monitorar o progresso, garantindo que o escopo de alta prioridade fosse concluído dentro do prazo estipulado. Ele define os períodos e os objetivos de cada iteração, promovendo a previsibilidade e a transparência do processo, aspectos cruciais das metodologias ágeis. Os marcos estabelecidos no cronograma serviram para a comunicação do *status* do projeto e para orientar os *Daily Scrums*, permitindo que a equipe se adaptasse rapidamente a qualquer desvio de rota, mantendo o foco na entrega contínua de valor.

Tabela 5 - Cronograma de Execução do Projeto (*Sprints* e Focos).

(Continua)

Task	Responsável	Duração	Início	Fim
Brainstorm	Equipe	1h	11/08/2025	11/08/2025
Regras de Negócio	Isaac Menezes	2h	11/08/2025	11/08/2025
Requisitos Funcionais	Felipe Pereira	3h	11/08/2025	11/08/2025
Requisitos Não Funcionais	Enzo Goldner	3h	11/08/2025	11/08/2025
<i>Backlog</i>	Equipe	2h	11/08/2025	12/08/2025
<i>Wireframe</i>	Isaac Menezes	4h	11/08/2025	12/08/2025
Configuração de Ambiente	Equipe	8h	12/08/2025	18/08/2025
Prototipagem Figma	Isaac Menezes	16h	18/08/2025	19/08/2025
Implementação Banco de Dados no SupaBase	Ana Beatriz	24h	12/08/2025	19/08/2025
Modelo Entidade Relacionamento	Equipe	4h	12/08/2025	15/11/2025
DER (Diagrama Entidade Relacionamento)	Equipe	4h	12/08/2025	15/11/2025
Perfil do Usuário	Isaac Menezes	5h	10/11/2025	11/11/25

Fonte: Autoria (2025).

### 3.5 MODELAGEM DE DADOS

Concluído o levantamento de requisitos, torna-se essencial a criação de uma estrutura lógica para o armazenamento das informações do sistema Ártemis. A modelagem de dados é o processo de definir e representar as entidades relevantes e os relacionamentos entre elas, servindo como a fundação para a implementação física do banco de dados.

### **3.5.1 Modelo Entidade-Relacionamento (MER)**

O Modelo Entidade-Relacionamento (MER) constituiu a primeira etapa conceitual da modelagem de dados do projeto. Sua principal finalidade foi realizar a abstração dos dados, identificando as entidades essenciais para o domínio da pizzaria (como Cliente, Pedido, Produto e Mesa) e os atributos que as descrevem. Conforme referenciado na Fundamentação Teórica (Coronel e Rob, 2011), o MER é crucial por permitir a organização de dados complexos em um nível de alto entendimento, garantindo que todas as regras de negócio definidas (Tabela 1) fossem mapeadas.

No projeto Ártemis, o MER serviu como um esboço conceitual, permitindo a visualização clara dos principais componentes de informação antes da definição das estruturas lógicas do banco. Para a leitura do diagrama, utilizou-se a notação padrão, na qual as entidades são representadas por retângulos, os atributos por ovais, e os relacionamentos por losangos.

Além disso, a interpretação das cardinalidades nos relacionamentos é fundamental, pois elas definem as restrições e a quantidade de instâncias que podem se relacionar. Por exemplo, a notação 1:N (um para muitos), comum no modelo, indica que uma instância de uma entidade se relaciona com múltiplas instâncias da outra. Essa notação permite compreender as regras de negócio em relação à participação dos elementos – como o fato de uma Mesa pode ter N Pedidos. O diagrama completo do Modelo Entidade-Relacionamento (MER) pode ser consultado no Apêndice A.

### **3.5.2. Diagrama Entidade-Relacionamento (DER)**

O Diagrama Entidade-Relacionamento (DER) é a etapa subsequente, que traduz o modelo conceitual (MER) em uma estrutura lógica detalhada, orientada diretamente à implementação no banco de dados. O DER foi fundamental para definir os aspectos técnicos do *schema* de dados e garantir a integridade referencial do sistema Ártemis, evitando inconsistências de dados. Em contraste com o MER, o DER em sua fase lógica prescinde dos detalhes de tipos, tamanhos e restrições dos atributos. Ele trabalha com os elementos cruciais para a construção física do banco de dados, sendo o foco de interpretação as Cardinalidades Lógicas e a identificação das Chaves Primárias (PK) e Estrangeiras (FK)."As FK, por

exemplo, são o mecanismo que liga fisicamente as tabelas e impõe a integridade referencial, conforme a Fundamentação Teórica sugere (Coronel e Rob, 2011). As cardinalidades, representadas por notações mais precisas para a implementação (*e.g., pé de galinha*), definem as restrições de participação máxima e mínima das entidades nos relacionamentos. Essa clareza na especificação permitiu que a estrutura de dados fosse diretamente utilizada para a criação das tabelas no *PostgreSQL*, com auxílio do *Prisma* ORM (mencionado na Fundamentação Teórica). A representação completa do Diagrama Entidade-Relacionamento (DER), que ilustra a arquitetura de dados final do projeto, está disponível para consulta no Apêndice B.

### **3.5.3. Dicionário de Dados( Modelo físico de dados)**

Complementando a modelagem, o Dicionário de Dados foi elaborado para detalhar com descrições e valores, de forma tabular, todos os atributos das entidades definidas no DER. Esta documentação é crucial para manter a consistência e a clareza do projeto, pois especifica o nome da tabela, o nome do campo (atributo), o tipo de dado (*e.g., string, integer, boolean*), se o campo é obrigatório (Null ou Not Null) e uma breve descrição da sua função. As tabelas que compõem o Dicionário de Dados do sistema Ártemis estão dispostas no Apêndice C.

O Dicionário de Dados, essencial para a manutenção e a clareza do projeto, inicia sua especificação pela entidade central de interação com o sistema. A tabela 6 detalha a estrutura da entidade usuário cliente, apresentando os atributos fundamentais necessários para o cadastro e a autenticação do consumidor, como o id, nome, email e a senha, além dos campos de auditoria criado\_em e atualizado\_em. Esta especificação garante a integridade dos dados, estabelecendo o tipo de dado e a obrigatoriedade (*Null/Not Null*) de cada campo, conforme o DER (Apêndice B) requer.

Tabela 6 - Entidade Usuário Cliente

(Continua)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do usuário, gerado automaticamente pelo sistema.	1a821476-5bc1-458a-93a8-01de7a06358b
nome	<i>text</i>	65.535 caracteres	-	Nome completo do usuário	Beatriz Queiroz Costa
email	<i>text</i>	65.535 caracteres	-	Endereço de e-mail do usuário	beatriz@teste.com
cpf	<i>text</i>	65.535 caracteres	-	Identificador único da pessoa	758.547.320-69
data_nascimento	<i>text</i>	65.535 caracteres	-	data de nascimento do usuário	01/11/1990
senha	<i>text</i>	65.535 caracteres	-	Senha de acesso do usuário	\$2b\$08\$YuJIQ.WVuRqmYtbHxbn8HOWN9Ane mEwJeFAcPdzaa3jOQk MHhvi6K
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 18:03:16.993
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do usuário.	2025-06-23 18:03:16.993

Fonte: Autoria (2025)

Tabela 7 - Entidade Usuário Funcionário

(Continua)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do usuário, gerado automaticamente pelo sistema.	1a821476-5bc1-458a-93a8-01de7a06358b
Funçãoid	<i>text</i>	65.535 caracteres	FK	Identificador único da função de trabalho do usuário, gerado automaticamente pelo sistema.	1a820930-9mk3-358w-93a8-01de7a06358b
nome	<i>text</i>	65.535 caracteres	-	Nome completo do Funcionário	Beatriz Queiroz Costa

Fonte: Autoria (2025).

Tabela 7 - Entidade Usuário Funcionário

(Continuação)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
cpf	<i>text</i>	65.535 caracteres	-	Identificador único da pessoa	758.547.320-69
email	<i>text</i>	65.535 caracteres	-	Endereço de e-mail do Funcionário	beatriz@teste.com
senha	<i>text</i>	65.535 caracteres	-	Senha de acesso do usuário	\$2b\$08\$YuJIQ.WVuRqmYtbHxbn8HOWN9AneWJeFAcpdzaa3jOQkMHhvi6K
data_nascimento	<i>text</i>	65.535 caracteres	-	Data de nascimento do funcionário	01/11/1990
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 18:03:16.993
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do usuário.	2025-06-23 18:03:16.993
id	<i>text</i>	65.535 caracteres	PK	Identificador único da função, gerado automaticamente pelo sistema.	1a821476-5bc1-458a-93a8-01de7a06358b
nome_funcao	<i>text</i>	65.535 caracteres	-	Nome da função	Garçom
nivel_acesso	<i>Int(32,0)</i>	4 bytes	-	Nível de acesso ao sistema	1

Fonte: Autoria (2025).

Tabela 8 - Entidade Função

(Continuação)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 18:03:16.993
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do usuário.	2025-06-23 18:03:16.993

Fonte: Autoria (2025).

Tabela 9 - Entidade Mesa

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único da mesa, gerado automaticamente pelo sistema	106566a0-d7b5-45e1-a0f6-2fdd dfa0dd7e
usuario_id	<i>text</i>	65.535 caracteres	FK	Identificador único do usuário, gerado automaticamente pelo sistema	106566a0-d7b5-45e1-a0f6-2fdd dfa0dd7e
numero	<i>int4(32,0)</i>	4 bytes	-	Número da mesa lido pelo QRCode	12
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-10 11:11:41.297
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações da atualização	2025-06-10 11:11:41.297

Fonte: Autoria(2025).

Tabela 10 - Entidade Produto

(Continua)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do produto, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d154 2dccc15d
nome	<i>text</i>	65.535 caracteres	-	Nome completo do produto	Pizza de Mussarela
preco	<i>text</i>	65.535 caracteres	-	Preço de cada produto	20
imagem	<i>text</i>	65.535 caracteres	-	Banner da imagem do Ingrediente	<a href="http://res.cloudinary.com/cov1750679291/tfawrojptx2bcxwhjheo.jpg">http://res.cloudinary.com/cov1750679291/tfawrojptx2bcxwhjheo.jpg</a>
descricao	<i>text</i>	65.535 caracteres	-	Detalhamento do produto	Pizza Com Mussarela

Fonte: Autoria (2025).

Tabela 10 - Entidade Produto

(Continuação)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 11:48:11.699
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 11:48:11.699
categoria_id	<i>text</i>	65.535 caracteres	FK	id da categoria a qual o produto pertence	b79c1fa2-0b7b-4cc7-bf69-565af74d31e5

Fonte: Autoria (2025).

Tabela 11 - Entidade Pagamento

(Continua)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do pagamento, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d1542dcc15d
pedido_Id	<i>text</i>	65.535 caracteres	FK	id do pedido a qual o pagamento pertence	0048678b-1dc7-430e-8382-d1542dcc92f
tipoDePagamento	<i>int(32,0)</i>	4 bytes	-	Tipo do pagamento	Cartão de débito
precoTotal	<i>numeric(65,30)</i>	32 bytes	-	Quantidade de caracteres no pagamento	22,5
transacao_Id	<i>text</i>	65.535 caracteres	-	id da transação do pagamento do pedido	0063288b-2hm7-490e-0142-d1542dcc92f
status	<i>text</i>	65.535 caracteres	-	status do andamento do pagamento do pedido	Pagamento finalizado!
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 11:48:11.699

Fonte: Autoria (2025).

Tabela 11 - Entidade Pagamento

(Continuação)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
CodigoPix	<i>text</i>	65.535 caracteres		código do pix do pagamento	b79c1fa2-0b7b-4cc7-bf69-565af74d31e5
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 11:48:11.699

Fonte: Autoria (2025).

Tabela 12 - Entidade Pedido

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único da order, gerado automaticamente pelo sistema.	b79c1fa2-0b7b-4cc7-bf69-565af74d31e5
nome	<i>text</i>	65.535 caracteres	-	Nome completo do usuário da order	Beatriz
mesa_id	<i>text</i>	65.535 caracteres	FK	Identificador único da mesa, gerado automaticamente pelo sistema	b79c1fa2-0b7b-4cc7-bf69-565af74d31e5
observacao	<i>text</i>	65.535 caracteres	-	Observação do pedido	Quero que tire o queijo, pois sou intolerante a lactose
<i>status</i>	<i>bool</i>	1 bit	-	Status do pedido	TRUE
rascunho	<i>bool</i>	1 bit	-	Rascunho do Pedido	FALSE
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-10 11:11:41.297
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 12:53:59.349

Fonte: Autoria (2025).

Tabela 13 - Entidade Ingredientes

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do ingrediente, gerado automaticamente pelo sistema	106566a0-d7b5-45e1-a0f6-2fdddfa0dd7e
nome	<i>text</i>	65.535 caracteres	-	Nome do ingrediente	Calabresa
preco	<i>numeric(65,30)</i>	65.535 caracteres	-	Preço do Ingrediente	20.00
imagem	<i>text</i>	65.535 caracteres	-	imagem do Ingrediente	<a href="http://res.cloudinary.com/dsq3pnxaf/image/upload/v1750679291/tfawrojptx2bcxwhjheo.jpg">http://res.cloudinary.com/dsq3pnxaf/image/upload/v1750679291/tfawrojptx2bcxwhjheo.jpg</a>
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-10 11:11:41.297
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações da atualização	2025-06-10 11:11:41.297

Fonte: Autoria (2025).

Tabela 14 - Entidade Itens

( Continua )

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do item, gerado automaticamente pelo sistema.	1a821476-5bc1-458a-93a8-01de7a06358b
quantidade	<i>int4(32,0)</i>	4 bytes	-	Quantos itens	1
status	<i>bool</i>	1 bit	-	Rascunho do Pedido	FALSE
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 18:03:16.993
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações da categoria.	2025-06-23 12:53:59.349

Fonte: Autoria (2025).

Tabela 14 - Entidade Itens

(Continuação)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
pedido_id	<i>text</i>	65.535 caracteres	FK	Identificador único da order, em que o item esta	b79c1fa2-0b7b-4cc7-bf69-565af74d31e5
produto_id	<i>text</i>	65.535 caracteres	FK	Identificador único do produto, que está na ordem.	0048678b-1dc7-430e-8382-d1542dcc15d

Fonte: Autoria (2025).

Tabela 15 - Entidade Favoritos

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do favorito, gerado automaticamente pelo sistema	106566a0-d7b5-45e1-a0f6-2fdd dfa0dd7e
produto_id	<i>text</i>	65.535 caracteres	FK	Identificador único do produto que foi favoritado	027413ac-f418-4df8-a86e-8a2a dca77093
usuarioCliente_id	<i>text</i>	65.535 caracteres	FK	Identificador único do favorito, gerado automaticamente pelo sistema	106566a0-d7b5-45e1-a0f6-2fdd dfa0dd7e
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-10 11:11:41.297
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações da atualização	2025-06-10 11:11:41.297

Fonte: Autoria (2025).

Tabela 16 - Entidade Categorias

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único da categoria, gerado automaticamente pelo sistema.	b79c1fa2-0b7b-4cc7-bf69-565af74d31e5
nome	<i>text</i>	65.535 caracteres	-	Nome da categoria	Pizzas
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 18:03:16.993
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações da categoria.	2025-06-23 12:53:59.349

Fonte: Autoria (2025).

Tabela 17 - Entidade Produto Ingrediente

(Continua)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do ProductIngredient, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d1542dcc15d
quantidade	<i>int4(32,0)</i>	4 bytes	-	Quantos ingredientes	1
produto_Id	<i>text</i>	65.535 caracteres	FK	Identificador único do produto, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d1542dcc15d
ingrediente_Id	<i>text</i>	65.535 caracteres	FK	Identificador único do ingrediente, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d1542dcc15d

Fonte: Autoria(2025).

Tabela 17 - Entidade Produto Ingrediente

(Continuação)

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 11:48:11.699
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 11:48:11.699

Fonte: Autoria(2025).

Tabela 18 - Entidade Extra

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do ProductIngredient, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d1542dccc15d
nome	<i>text</i>	65.535 caracteres		Nome do adc.	0048678b-1dc7-430e-8382-d1542dccc15d
preço	<i>text</i>	65.535 caracteres		Preço do adc.	R\$ 200,00
Categoria_id	<i>text</i>	65.535 caracteres	FK	identificador da categoria de extra	e9ab564b-e1bd-4a9d-a4bf-74a1539c2748
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 11:48:11.699
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 11:48:11.699

Fonte: Autoria (2025).

Tabela 19 - Entidade Item Extra

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do ProductIngredient, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d1542dccc15d
item_id	<i>text</i>	65.535 caracteres	FK	identificador da tabela item	0048678b-1dc7-430e-8382-d1542dccc15d
extra_id	<i>text</i>	65.535 caracteres	FK	identificador da tabela extra	e9ab564b-e1bd-4a9d-a4bf-74a1539c2748
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 11:48:11.699
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 11:48:11.699

Fonte: Autoria (2025).

Tabela 20 - Entidade Produto Ingrediente Removido

Atributo	Tipo de Dado	Tamanho Aproximado	PK / FK	Descrição	Valor
id	<i>text</i>	65.535 caracteres	PK	Identificador único do ProductIngredient, gerado automaticamente pelo sistema.	0048678b-1dc7-430e-8382-d1542dccc15d
ProdutoIngrediente_Id	<i>text</i>	65.535 caracteres	FK	Identificador único da tabela productIngredient	b78be45c-9d20-4e1d-adc7-b12ee00cc731
item_Id	<i>text</i>	65.535 caracteres	FK	Identificador único da tabela item	c9258818-7525-4681-ac33-ba84e66415cc
produto_id	<i>text</i>	65.535 caracteres	FK	Identificador único da tabela product	0a9cc645-2458-404e-9554-2fa5bf328581
criado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora de criação do registro no sistema.	2025-06-23 11:48:11.699
Atualizado_em	<i>timestamp(3)</i>	8 bytes	-	Data e hora da última atualização das informações do pedido.	2025-06-23 11:48:11.699

Fonte: Autoria (2025).

### 3.6. DESIGN GRÁFICO

O desenvolvimento do design do projeto teve como propósito integrar referências consolidadas do mercado a uma solução visual simples, intuitiva e moderna, adequada ao ambiente de autoatendimento via dispositivo móvel (BYOD). Como o sistema se destina a um ambiente de autoatendimento presencial, tornou-se fundamental criar uma interface que

orientasse o usuário por meio de uma liberdade assistida, reduzindo erros, acelerando o fluxo de ações e oferecendo uma experiência clara e objetiva.

Para fundamentar as decisões tomadas, foram analisados modelos de empresas reconhecidas internacionalmente pela maturidade de seus processos de design. Um dos principais referenciais foi o *iFood* (IFOOD, 2023), cuja comunicação institucional disponibiliza artigos sobre experiência do usuário, enfatizando detalhes que influenciam diretamente na percepção de eficiência e satisfação durante o uso do serviço. Outro referencial utilizado foi o material disponibilizado pela *Uber* (UBER, 2023) sobre seu processo interno de design, que descreve metodologias, uso de tecnologias como o Figma e a construção de sistemas escaláveis para evolução contínua da plataforma. O aspecto de escalabilidade, especialmente, serviu de inspiração para estruturar o design do projeto de maneira que possa ser ampliado no futuro sem descaracterizar sua identidade principal. Assim, o propósito central foi alinhar funcionalidade e estética, construindo um sistema visual consistente, acessível e tecnicamente sustentável.

### 3.6.1. Identidade visual

A identidade visual do projeto foi desenvolvida a partir de um Guia de Estilos próprio, elaborado especificamente para garantir coerência entre todos os elementos que compõem o protótipo. Esse guia funciona como base técnica para a aplicação das cores, tipografias, componentes, imagens e estrutura visual, assegurando uniformidade no uso e facilitando a manutenção futura do design (Figura 8).

**Figura 8 - Guia de Estilos**



Fonte: Autoria (2025)

A escolha tipográfica priorizou legibilidade e hierarquia clara. Foram selecionadas fontes com traços geométricos e boa leitura em ambientes digitais, adequadas ao uso em telas de dispositivos móveis. Títulos utilizam uma tipografia mais robusta, de impacto visual, com a função de orientar rapidamente o usuário em cada etapa do fluxo. Textos de corpo adotam uma fonte neutra e equilibrada, garantindo conforto visual e reduzindo a carga cognitiva durante a navegação.

A paleta cromática foi planejada para equilibrar identidade gastronômica com boa acessibilidade visual. Tons quentes — como vermelho profundo e amarelo — foram escolhidos por sua forte associação com o universo de pizzarias, remetendo ao calor do forno, à massa e aos ingredientes tradicionais. Esses tons provocam sensações de apetite, acolhimento e proximidade. Para contrabalançar a intensidade dessas cores, foram incluídos tons neutros e um azul escuro, que proporcionam contraste adequado, reforçam a modernidade da interface e ajudam a criar zonas de foco visual. A presença do *off-white* cumpre a função de "respiro" na tela, permitindo que os elementos principais se destaquem.

As imagens presentes no guia de estilos foram selecionadas para reforçar a comunicação visual e favorecer o reconhecimento imediato das categorias e produtos. Optou-se por fotografias de pizzas, ingredientes e elementos associados ao ambiente de restaurantes, transmitindo autenticidade e facilitando o processo de decisão do cliente. A decisão de usar fotografias condiz com a necessidade de se aproximar do usuário por meio do imaginário cotidiano.

### **3.6.2. Prototipagem da imagem e validação visual**

Após a definição da Identidade Visual e do Guia de Estilos (3.6.1), a prototipagem foi conduzida para materializar os Requisitos Funcionais em um produto visual e navegável. Esta fase utilizou a criação de *wireframes* de baixa fidelidade para organizar o fluxo de telas, seguida pelo desenvolvimento do protótipo de alta fidelidade no *Figma*.

A prototipagem validou o conceito de autoatendimento via dispositivo móvel (BYOD), um pilar da Fundamentação Teórica (QUINTELA, 2016), ao simular a navegação do cliente pelo seu *smartphone*. O principal objetivo foi garantir que a interface fosse intuitiva e previsível

(NORMAN, 2013), reduzindo a carga cognitiva e promovendo a autonomia do cliente na visualização do cardápio dinâmico e na personalização dos pedidos.

A aplicação das diretrizes de tipografia e da paleta cromática definidas no Guia de Estilos foi testada no protótipo, assegurando legibilidade e coerência visual em todos os elementos, como o fluxo de leitura do QR *Code* para vinculação à mesa.

Para uma compreensão aprofundada da organização das telas, da evolução visual da interface e do fluxo de navegação proposto, os *wireframes* e o protótipo de alta fidelidade estão disponíveis para consulta no APÊNDICE C – DOCUMENTAÇÃO VISUAL DA INTERFACE.

### **3.7. TECNOLOGIAS UTILIZADAS**

A escolha da *stack* tecnológica do projeto Ártemis foi orientada por critérios de produtividade, escalabilidade e capacidade de desenvolvimento *cross-platform*, essenciais para uma aplicação que adota o modelo BYOD (*Bring Your Own Device*). O projeto demandou uma arquitetura moderna e escalável, baseada em JavaScript, visando desenvolver uma aplicação *mobile* nativa de alta performance e um *backend* robusto para a gestão eficiente dos dados do restaurante. As ferramentas selecionadas compõem um conjunto moderno focado em otimizar o ciclo de desenvolvimento (alinhado à metodologia Ágil) e garantir a consistência e integridade da aplicação.

#### **3.7.1. Plataforma *Front-end* e Desenvolvimento Móvel**

A camada de interface e interação com o usuário (*Frontend*) foi desenvolvida em React Native (framework) com o auxílio do *Expo Go* (ambiente de desenvolvimento), garantindo a portabilidade nativa para os sistemas *Android* e *iOS*. Essa escolha tecnológica foi estratégica para otimizar o trabalho em equipe, uma vez que o *React Native* permite a manutenção de uma base de código única, evitando a fragmentação do desenvolvimento em linguagens nativas distintas. O uso do *Expo*, por sua vez, mitiga problemas de compatibilidade de ambiente entre os diferentes sistemas operacionais utilizados pelos desenvolvedores, além de agilizar a etapa de testes em dispositivos físicos por meio da leitura de QR *Codes*, sem a necessidade de compilações demoradas.

Para aumentar a segurança e a integridade do código, utilizou-se o *TypeScript* para tipagem estática, permitindo a detecção precoce de erros. O desenvolvimento da interface segue o paradigma declarativo e baseado em componentes do *React Native*. A Listagem 1 exemplifica a estrutura de um componente de exibição de produto, onde elementos como `<View>` e `<Text>` organizam o conteúdo e consomem as propriedades do item por meio da sintaxe JSX.

#### **Listagem 1 – Exemplo de Estrutura de Componente de Cartão de Produto**

```
<View style={styles.textContainer}>
  <Text style={styles.cardTitle} numberOfLines={1}>{item.product.name}</Text>
  <Text style={styles.cardDescription} numberOfLines={1}>{item.product.description}</Text>
</View>
```

Fonte: Autoria(2025)

A comunicação com o *backend* foi estabelecida via requisições HTTP, gerenciadas pela biblioteca *Axios*, conhecida por simplificar o consumo de APIs e tratar retornos de forma padronizada. A Listagem 2 ilustra a criação de uma instância configurada do *Axios*, permitindo que todas as chamadas subsequentes à API utilizem o mesmo endereço base e as mesmas configurações.

#### **Listagem 2 – Configuração Base da Instância Axios**

```
const api = axios.create({
  baseURL: "http://192.168.15.16:3333",
});
```

Fonte: Autoria(2025)

Adicionalmente, o projeto utilizou bibliotecas essenciais para funcionalidades específicas de *hardware* e UX: o React Navigation foi implementado para gerenciar o fluxo de navegação complexo entre as telas; o Async Storage garantiu a persistência local de dados (como *tokens* de autenticação); e o conjunto *Expo Camera* e *Expo Barcode Scanner* foi crucial para viabilizar a funcionalidade de leitura do QR *Code*, essencial para a vinculação da

mesa ao pedido. Por fim, o *React Native Gesture Handler* foi incluído para gerenciar interações tátteis mais avançadas na interface.

### 3.7.2. Backend e Gerenciamento de Dados

O *backend* foi construído em ambiente *Node.js* utilizando o *framework* “*Express*” para roteamento de APIs e lógica de negócio. A persistência e o gerenciamento dos dados foram realizados com o serviço *Supabase*, que fornece uma plataforma completa baseada em *PostgreSQL*. Os testes de API foram feitos utilizando o *insomnia*, programa que permite a utilização das APIs sem o uso da interface gráfica do aplicativo em desenvolvimento.

O *Express* é fundamental para gerenciar as rotas e executar a lógica de negócios crítica, como validações e verificações de status. A Listagem 3 ilustra um trecho de validação essencial na API, garantindo que a requisição de finalização de pedido contenha todos os dados obrigatórios antes de prosseguir com o processamento.

#### Listagem 3 – Exemplo de Validação da Lógica de Negócio (*Express*)

```
if (!orderId || !paymentType) {  
  
    return res.status(400).json({ error: 'Dados essenciais (ID do pedido e tipo de pagamento) incompletos.' });  
  
}
```

Fonte: Autoria(2025)

A interação entre o *backend* e o banco de dados foi intermediada pelo *Prisma* ORM (*Object-Relational Mapper*). O *Prisma* foi escolhido por facilitar a modelagem de dados, prover tipagem segura e automatizar consultas complexas, alinhando a estrutura do código à estrutura do banco. A Listagem 4 demonstra a estrutura de uma *model* no *schema* do *Prisma*, definindo os atributos, tipos e mapeamentos (como a chave `@id` e o relacionamento com `UsuarioFuncionario`), que servem como base para a criação das tabelas no *PostgreSQL*.

#### Listagem 4 – Definição de Model (Prisma ORM)

```
model Funcao{
    id      String      @id @default(uuid())
    funcao_name  String      @db.Text
    nivel_acesso Int
    criado_em   DateTime     @default(now())
    atualizado_em  DateTime     @default(now()) @updatedAt
    funcionarios  UsuarioFuncionario[]
    @@map("Funcao")
}
```

Fonte: Autoria(2025)

### 3.7.3. Ferramentas de Apoio, Versionamento e Gestão

Para além da *stack* de desenvolvimento (Frontend e Backend), o projeto Ártemis exigiu um conjunto de ferramentas de apoio focadas em governança, colaboração e organização do trabalho, essenciais para o sucesso na abordagem ágil. O *Git* foi adotado como sistema de controle de versão distribuído, sendo fundamental para a integridade e rastreabilidade do código-fonte. Em um ambiente de desenvolvimento colaborativo, ele permitiu que a equipe trabalhasse de forma assíncrona, gerenciando *branches* de desenvolvimento e garantindo que o histórico completo de todas as modificações fosse registrado. O uso do *GitHub* como repositório remoto assegurou que o código estivesse centralizado e acessível, facilitando a revisão e a manutenção da estabilidade do sistema ao longo da evolução das *sprints*.

Complementando o versionamento, ferramentas de gestão de fluxo como o Trello foram cruciais para a organização e rastreabilidade das tarefas, seguindo o *framework* Kanban e a estrutura de *sprints* do Scrum. No que tange à concepção visual e modelagem, o Figma foi a ferramenta central para a criação dos *wireframes*, do Guia de Estilos e do protótipo de alta fidelidade. Para a modelagem de dados e elaboração de diagramas conceituais, como o Modelo Entidade-Relacionamento (MER) e Diagramas de Casos de Uso, utilizou-se o

*Draw.io* e o *Miro* durante as sessões de *brainstorming*. Toda a documentação formal, incluindo a elaboração colaborativa do projeto e o armazenamento centralizado dos artefatos do projeto, foi gerenciada através do *Google Docs/Drive*.

A inspiração para a concepção visual do projeto baseou-se em referências do mercado, demandando a participação ativa do *design* para traduzir as necessidades de simplicidade, interatividade e modernidade em uma interface que, simultaneamente, guiasse o usuário (restringindo a liberdade para evitar erros), conforme o objetivo de autonomia assistida estabelecido na Fundamentação Teórica.

## 4. CONCLUSÃO

O presente Trabalho de Conclusão de Curso atingiu seus objetivos ao conceber e desenvolver o Ártemis, um sistema mobile de autoatendimento baseado no modelo BYOD (*Bring Your Own Device*). Embora ainda não tenha sido implantado em um ambiente produtivo, o protótipo funcional construído demonstra a viabilidade técnica da proposta e evidencia o potencial do uso de dispositivos pessoais no fluxo de atendimento gastronômico.

A aplicação foi estruturada com tecnologias que favoreceram a portabilidade, a organização do código e a escalabilidade da solução. O uso de *React Native* e *TypeScript* no *frontend*, aliado ao *backend* desenvolvido com *Node.js*, *Express*, *Prisma ORM* e *Supabase*, possibilitou a criação de uma arquitetura coerente com as necessidades do projeto. Além disso, a adoção de metodologias ágeis e do versionamento contínuo em *Git* contribuiu para o controle das etapas, para a rastreabilidade das decisões e para o avanço consistente do desenvolvimento.

Com a base técnica consolidada, o Ártemis apresenta condições para evoluir em direção a um produto completo. Como aprimoramentos futuros, destacam-se a implementação de um histórico detalhado de pedidos, o desenvolvimento de um sistema de fidelidade, o aperfeiçoamento do cardápio para suportar customizações complexas e a criação de um painel administrativo que ofereça indicadores financeiros e operacionais. Tais extensões têm potencial para ampliar a utilidade da aplicação tanto para os usuários quanto para a gestão do estabelecimento.

Assim, conclui-se que o Ártemis constitui uma solução promissora no contexto da digitalização do atendimento em restaurantes, contribuindo para processos mais eficientes, autônomos e alinhados às demandas contemporâneas de modernização.

## REFERÊNCIAS

- CORONEL, Carlos; ROB, Peter. *Sistemas de banco de dados: projeto, implementação e administração*. 9. ed. Rio de Janeiro: LTC, 2011.
- FIELDING, Roy Thomas. *Architectural styles and the design of network-based software architectures*. 2000. Tese (Doutorado) – University of California, Irvine, 2000.
- GHOBRIL, Alexandre Nabil; BENEDETTI, Mauricio Henrique; FRAGOSO, Nelson Destro. Práticas inovadoras no setor de bares, restaurantes e lanchonetes. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 18., 2021, Resende. Anais [...]. Resende: AEDB, 2021.
- NORMAN, Donald A. *Design emocional: por que amamos (ou odiamos) os objetos do dia a dia*. Rio de Janeiro: Rocco, 2013.
- PERKOVIC, Ljubomir. *Introdução à programação com Python*. Rio de Janeiro: Elsevier, 2015.
- QUINTELA, Adriana. *Comunicação e novas mídias*. São Paulo: Atlas, 2016.
- SANTOS, Flávia. *Mídias digitais e interação*. São Paulo: PUC-SP, 2017.
- SOMMERVILLE, Ian. *Engenharia de software*. 9. ed. São Paulo: Pearson Education, 2011.
- SUTHERLAND, Jeff. *Scrum: a arte de fazer o dobro do trabalho na metade do tempo*. São Paulo: LeYa, 2014.
- CLOUDINARY. *Digital Asset Management (DAM)*. [S. l.]: Cloudinary, [2025]. Disponível em: <https://cloudinary.com/>. Acesso em: 10 nov. 2025.
- LOCAWEB. *Wireframe: 7 ferramentas para agilizar seus projetos*. [S. l.], 19 dez. 2023. Disponível em: <https://www.locaweb.com.br/blog/temas/de-olho-no-digital/o-que-e-wireframe/>. Acesso em: 15 nov. 2025.
- SUPABASE. *The Open Source Firebase Alternative*. [S. l.]: Supabase, [2025]. Disponível em: <https://supabase.com/>. Acesso em: 15 nov. 2025.

## GLOSSÁRIO

*API (Application Programming Interface): Conjunto de rotinas e padrões que permitem a comunicação entre sistemas, possibilitando integrações como pagamento e geolocalização.*

*API REST: Estilo arquitetural que define padrões para criação de APIs utilizando o protocolo HTTP.*

*Autenticação: Processo de verificação de identidade do usuário para garantir acesso seguro ao sistema.*

*Back-End: Camada lógica e estrutural da aplicação responsável pelo processamento dos dados e comunicação com o banco.*

*Brainstorming: Técnica colaborativa utilizada para gerar ideias e possibilidades relacionadas ao projeto, sem restrições iniciais.*

*Cadastro de Usuário: Processo de criação de conta no sistema, envolvendo dados pessoais e credenciais.*

*Cardápio Digital: Interface que apresenta os itens disponíveis para seleção e pedido.*

*Carrinho: Função que reúne os itens escolhidos pelo usuário antes da finalização do pedido.*

*CDN (Content Delivery Network): Rede de servidores usada para distribuir conteúdo com maior velocidade e desempenho.*

*Cloudinary: Plataforma de gerenciamento e hospedagem de mídias utilizada para armazenar imagens da aplicação.*

*CPF: Cadastro de Pessoa Física, identificador numérico do cidadão brasileiro.*

*Customização de Pedido: Função que permite ajuste de ingredientes, tamanhos e observações pelo usuário.*

*DBA (Database Administrator): Profissional responsável pela administração e manutenção do banco de dados.*

*DER (Diagrama Entidade-Relacionamento): Modelo visual que representa entidades e seus relacionamentos.*

*Development Team: Equipe de desenvolvimento responsável pela implementação do produto.*

*Dicionário de Dados: Documento que descreve atributos, tipos e regras das tabelas do banco de dados.*

*Entidade:* Objeto representado no banco, como usuários, mesas ou pedidos.

*Express:* Fornece uma estrutura de desenvolvimento para o API REST

*Figma:* Ferramenta para design de interfaces e prototipação.

*Frontend:* Camada da aplicação responsável pela interface gráfica com o usuário.

*Framework:* Estrutura que fornece ferramentas e padrões para desenvolvimento de software.

*Git:* Sistema de controle de versões.

*GitHub:* Plataforma para hospedagem e colaboração em projetos que utilizam Git.

*Google Docs:* Serviço de edição de documentos em nuvem.

*Google Drive:* Armazenamento em nuvem da Google.

*Google Maps:* Serviço de localização e mapas integrado à aplicação para dados geográficos.

*História de Usuário (User Story):* Descrição curta que representa uma necessidade do usuário.

*HTTP (Hypertext Transfer Protocol):* Protocolo usado para comunicação entre cliente e servidor na web.

*Interface Responsiva:* Capacidade da interface se adaptar a diferentes tamanhos de tela.

*JavaScript:* Linguagem de programação utilizada no desenvolvimento web e mobile.

*JSX:* Extensão de sintaxe do JavaScript usada no React e React Native.

*Kanban:* Método visual de organização do fluxo de trabalho por cartões e colunas.

*Login Social:* Autenticação realizada por meio de serviços externos, como Google.

*Mesa (Table):* Entidade que identifica a mesa vinculada ao pedido em restaurantes.

*MER (Modelo Entidade-Relacionamento):* Representação lógica das entidades e relacionamentos do sistema.

*Metodologia Ágil:* Conjunto de práticas que priorizam entregas rápidas, adaptação e trabalho iterativo.

*Miro:* Ferramenta colaborativa para organização de ideias, fluxos e quadros.

*Model:* Representação de um objeto de dados no contexto da aplicação.

*Mobile: Aplicações desenvolvidas para dispositivos móveis.*

*MVP (Minimum Viable Product): Versão mínima do produto com funções essenciais para testes iniciais.*

*Node.js: Ambiente de execução de JavaScript no servidor.*

*ORM (Object-Relational Mapping): Técnica que mapeia estruturas do banco para objetos na aplicação.*

*Payload: Conjunto de dados enviados em uma requisição ou resposta.*

*Pedido: Conjunto de itens selecionados que será enviado para preparo e pagamento.*

*Pix: Sistema de pagamento instantâneo brasileiro.*

*PostgreSQL: Sistema gerenciador de banco de dados relacional.*

*Prisma ORM: ORM utilizado para modelar e manipular o banco de dados.*

*Product Backlog: Lista priorizada de funcionalidades, demandas e melhorias do projeto.*

*Pull (Kanban): Sistema de puxar tarefas conforme capacidade do time, sem sobrecarregar o fluxo.*

*QA (Quality Assurance): Profissional responsável pela garantia de qualidade do software.*

*QR Code: Código que permite identificação automática da mesa ou usuário.*

*Regras de Negócio (RN): Conjunto de políticas que determinam o funcionamento do sistema.*

*Requisitos Funcionais (RF): Funcionalidades obrigatórias que o sistema deve cumprir.*

*Requisitos Não Funcionais (RNF): Propriedades relacionadas a desempenho, segurança, usabilidade etc.*

*Schema: Estrutura que define a forma dos dados em um banco ou aplicação.*

*Scrum: Framework ágil de desenvolvimento baseado em ciclos curtos (sprints).*

*Scrumban: Híbrido entre Scrum e Kanban usado para organizar o fluxo de desenvolvimento.*

*Sprints: Ciclos de tempo definidos para desenvolvimento de incrementos do produto.*

*Stack de Desenvolvimento: Conjunto de linguagens, ferramentas e tecnologias usadas no projeto.*

*Supabase: Plataforma de banco de dados com autenticação e APIs prontas.*

*TypeScript: Linguagem baseada em JavaScript com tipagem estática.*

*UI (User Interface): Interface visual com a qual o usuário interage.*

*UX (User Experience): Experiência total do usuário ao utilizar o sistema.*

*Wireframe: Desenho estrutural simplificado das telas antes da prototipação.*

**APÊNDICE A - MODELO DE ENTIDADE-RELACIONAMENTO (MER)**

**APÊNDICE B - DIAGRAMA DE ENTIDADE-RELACIONAMENTO (DER)**

## APÊNDICE C - DOCUMENTAÇÃO VISUAL DA INTERFACE

Este apêndice reúne o protótipo de alta fidelidade desenvolvido para o projeto Ártemis. O material visual tem como objetivo documentar a estrutura final da interface, o fluxo de navegação e a aplicação prática do Guia de Estilos definido na Seção 3.6.1. O conteúdo demonstra a aplicação dos princípios de autonomia do usuário e usabilidade no contexto do modelo BYOD.

Cada ilustração, organizada na sequência lógica de navegação do usuário, representa uma tela-chave ou um estado importante do fluxo de interação. Isso permite evidenciar as decisões tomadas em relação à hierarquia visual, tipografia e paleta cromática, conforme as diretrizes do *design* focado em dispositivos móveis.

**Figura C.1 - Páginas de Cadastro (*Placeholder* e *Preenchido*)**

**Cadastre-se**

Bem-vindo à Artemis Pizzaria

Nome

Email

CPF (apenas números)

Data de Nascimento

Senha

Confirmar Senha

Finalizar Cadastro

Já tem uma conta? [Faça Login](#)

**Cadastre-se**

Bem-vindo à Artemis Pizzaria

Erick Sales

ericksales@email.com

80023592087

01/04/2000

Senha

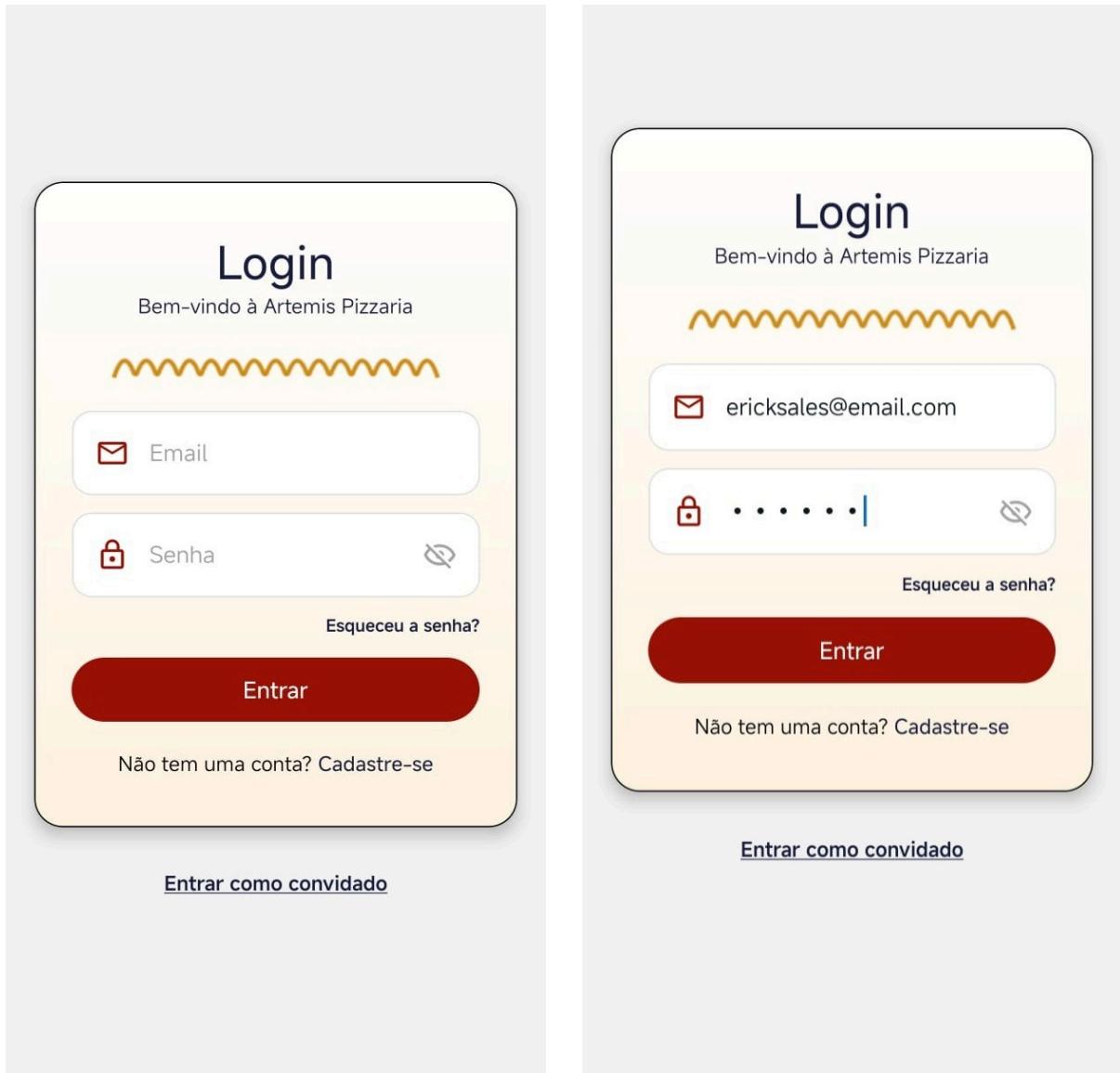
Confirmar Senha

Finalizar Cadastro

Já tem uma conta? [Faça Login](#)

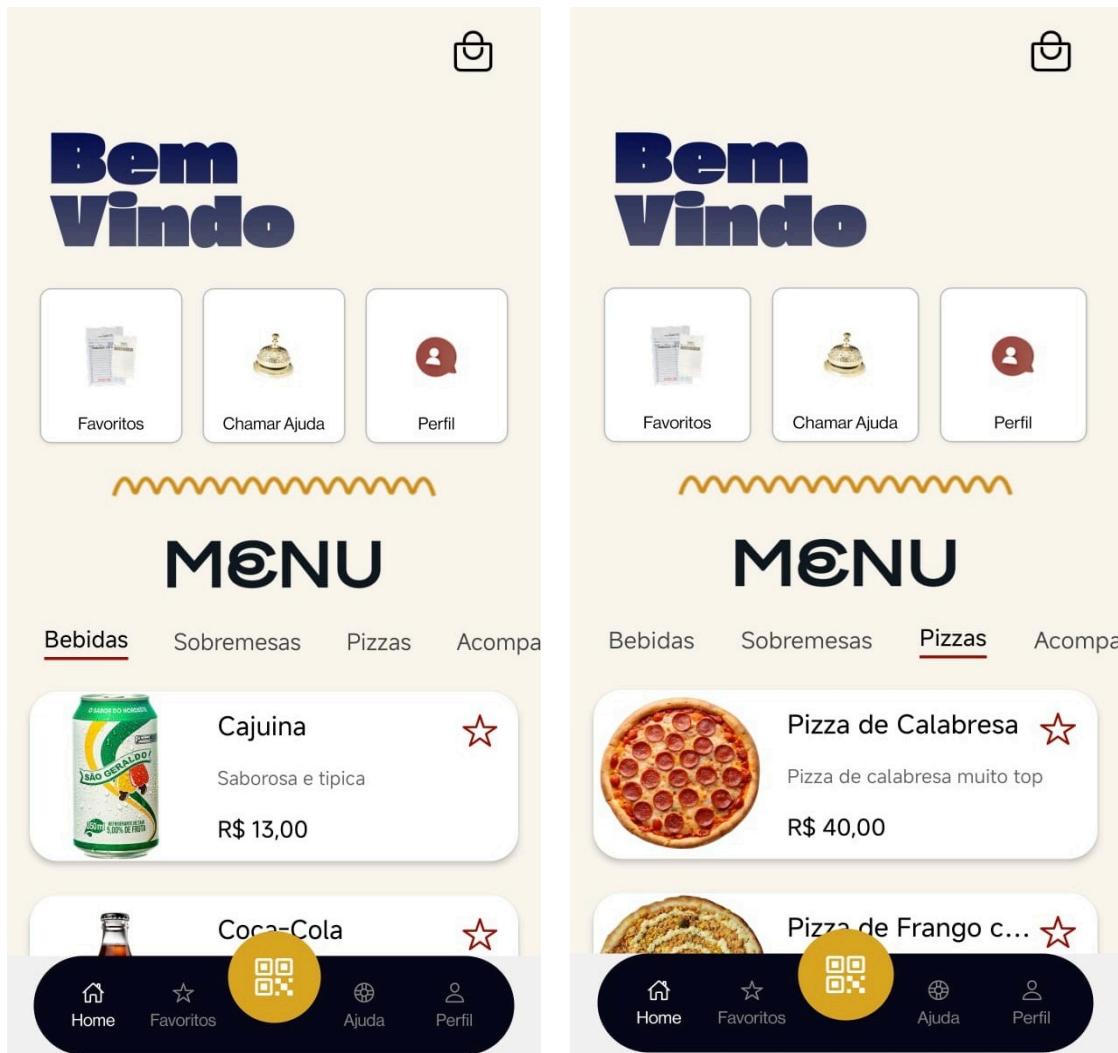
Fonte: Autoria (2025)

Figura C.2 - Páginas de Login (Placeholder e Preenchido)



Fonte: Autoria (2025)

**Figura C.3 - Páginas de Menu (Categorias: Bebidas e Pizzas)**



Fonte: Autoria (2025)

**Figura C.4 - Páginas de Detalhes e Personalização**



**Margherita Clássica**

A simplicidade e o sabor da tradição italiana.  
Uma combinação leve e aromática.

**Ingredientes**

Molho de Tomate	<input checked="" type="checkbox"/>
Manjericão fresco	<input type="checkbox"/>
Muçarela	<input checked="" type="checkbox"/>
Rodelas de Tomate	<input checked="" type="checkbox"/>

**Adicionais**

Borda de catupiry + R\$ 5,50	<input type="checkbox"/>
Borda de mussarela + R\$ 5,50	<input checked="" type="checkbox"/>
Mussarela + R\$ 5,50	<input type="checkbox"/>
Bacon + R\$ 5,50	<input type="checkbox"/>

- 1 +      Adicionar R\$ 45,00

## Margherita Clássica

A simplicidade e o sabor da tradição italiana.  
Uma combinação leve e aromática.

**Ingredientes**

Molho de Tomate	<input checked="" type="checkbox"/>
Manjericão fresco	<input checked="" type="checkbox"/>
Muçarela	<input checked="" type="checkbox"/>
Rodelas de Tomate	<input checked="" type="checkbox"/>

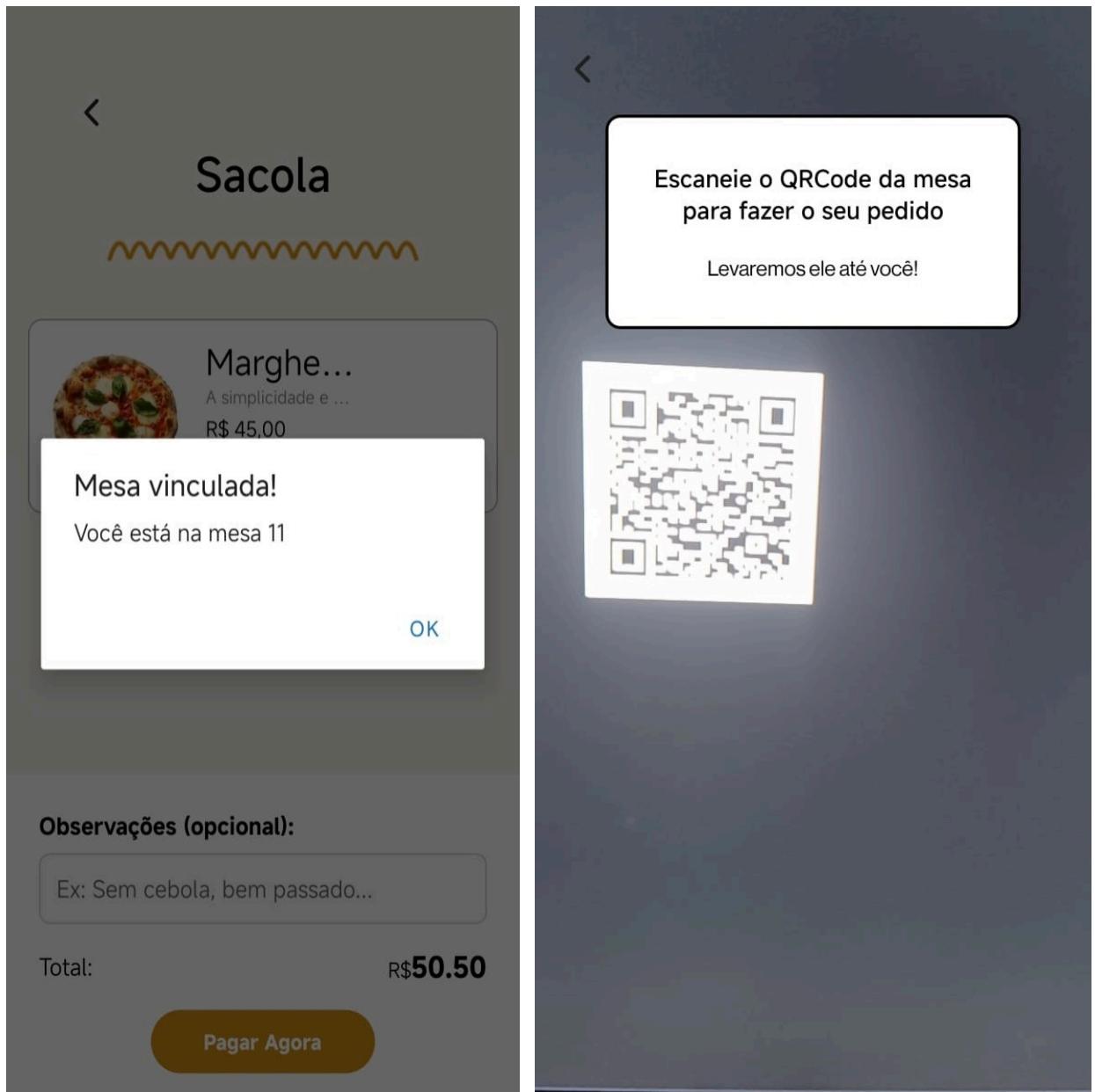
**Adicionais**

Borda de catupiry + R\$ 5,50	<input type="checkbox"/>
Borda de mussarela + R\$ 5,50	<input checked="" type="checkbox"/>
Mussarela + R\$ 5,50	<input type="checkbox"/>
Bacon + R\$ 5,50	<input type="checkbox"/>

- 1 +      Adicionar R\$ 50,50

Fonte: Autoria (2025)

**Figura C.5 - Páginas de Leitura do QR Code e Validação da Mesa**



Fonte: Autoria (2025)

Figura C.6 - Páginas de Sacola e Checkout com Pagamento

The figure consists of two side-by-side screenshots of a mobile application.

**Screenshot 1: Sacola (Cartão)**

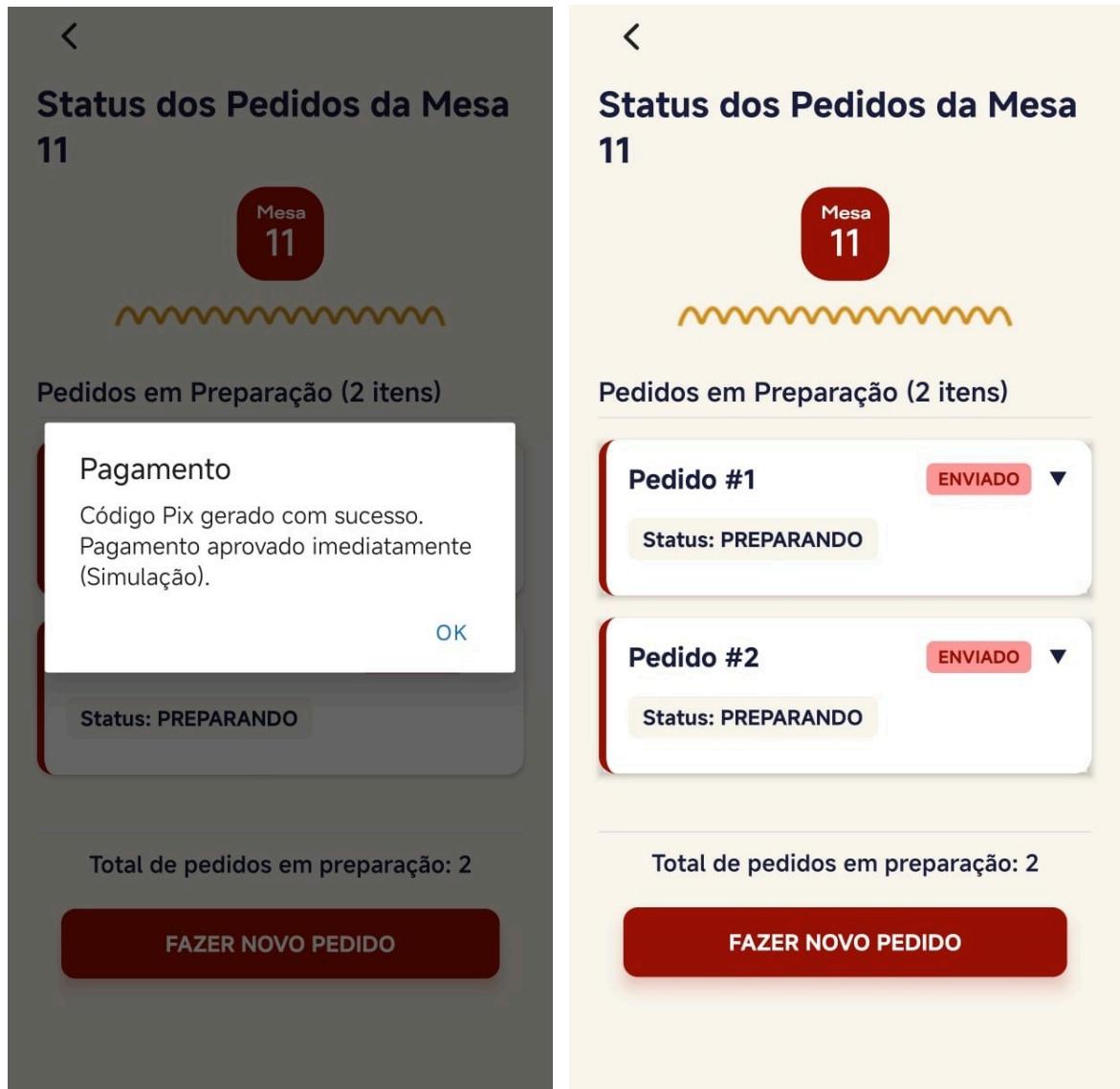
This screen shows the shopping cart. At the top, there is a back arrow and the word "Sacola". Below this is a decorative wavy line. A card item is listed: "Marghe..." (Margherita), described as "A simplicidade e ..." (The simplicity and ...), priced at "R\$ 45,00". There are minus and plus buttons next to the quantity, which is currently "1". Below the card is a section for "Observações (opcional)" (Optional notes) with a placeholder "Ex: Sem cebola, bem passado...". At the bottom, it says "Total: R\$ 50.50" and features a yellow button labeled "Pagar Agora" (Pay Now).

**Screenshot 2: Checkout**

This screen shows the payment method selection. At the top, there is a back arrow and the word "Checkout" with a decorative wavy line below it. It displays the order ID "Pedido: 07c95342-5f20-4a8c-aade-448a2392a021". Below this are three payment method options: "Cartão de Crédito" (Credit Card), "Cartão de Débito" (Debit Card), and "Pix". Each option has a "Detalhes" (Details) button. A modal window titled "Pix" is open, containing the text "Após confirmar, um QR Code será gerado para pagamento." (After confirming, a QR code will be generated for payment.) and a QR code icon with the amount "50.50". At the bottom of the modal is a yellow button labeled "Gerar QR Code Pix" (Generate QR Code Pix).

Fonte: Autoria (2025)

Figura C.7 - Alerta de Pagamento bem-sucedido e Página de Pedidos Relacionados à Mesa



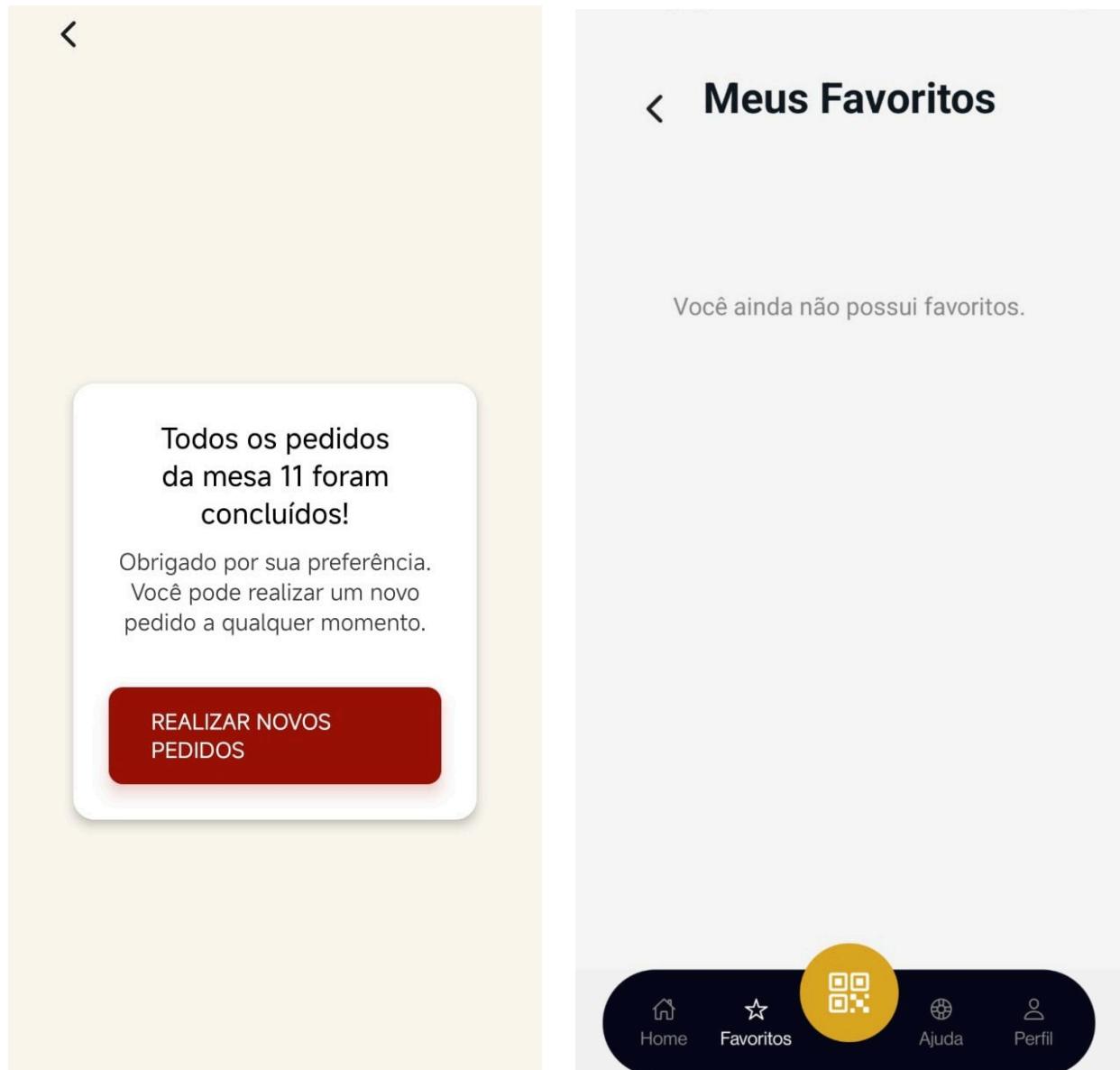
Fonte: Autoria (2025)

**Figura C.8 - Páginas de Status do Pedido (Enviado à Cozinha e Item Entregue)**



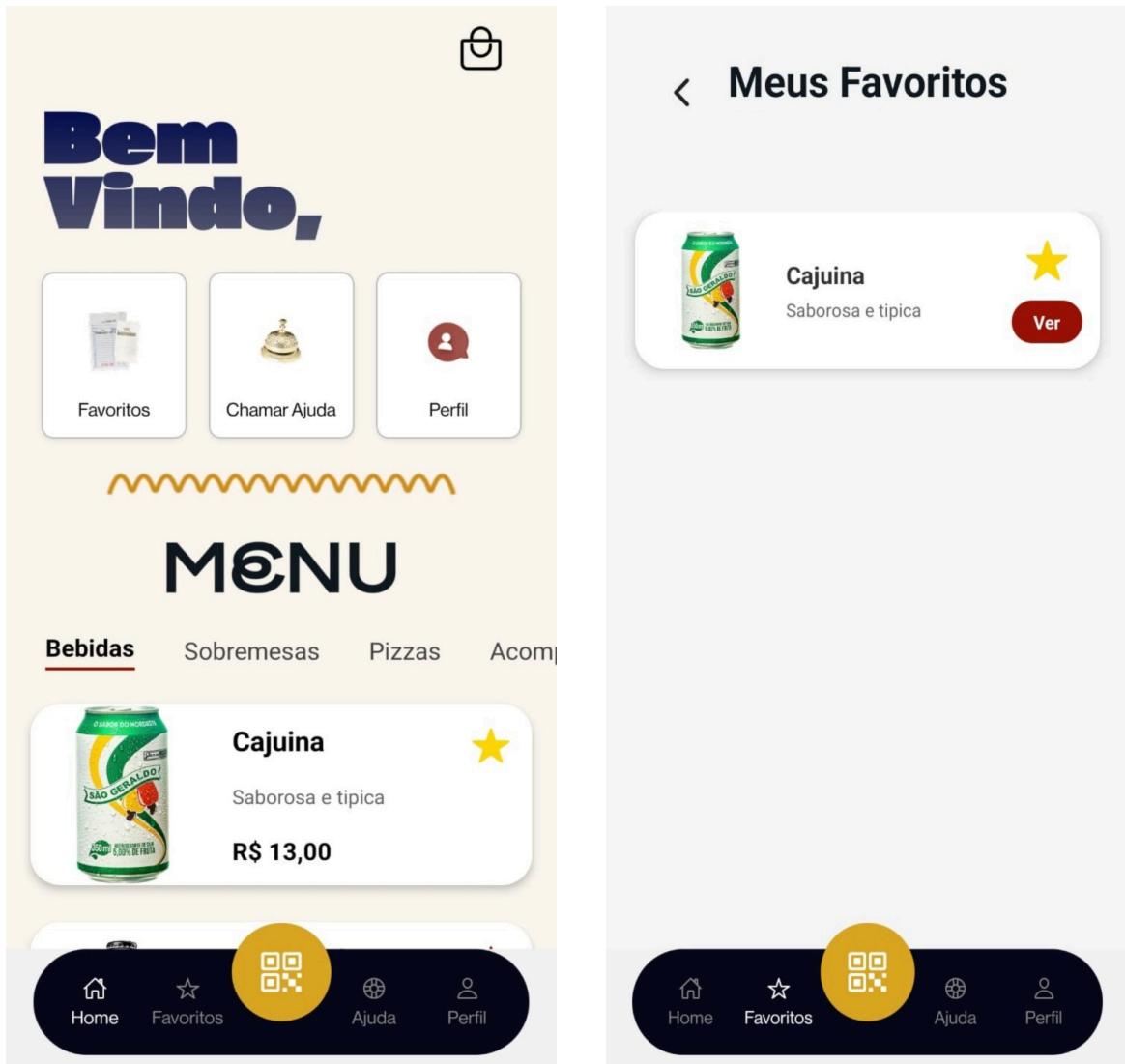
Fonte: Autoria (2025)

**Figura C.9 - Alerta de Pedidos Finalizados e Página de Produtos Favoritos (Vazio)**



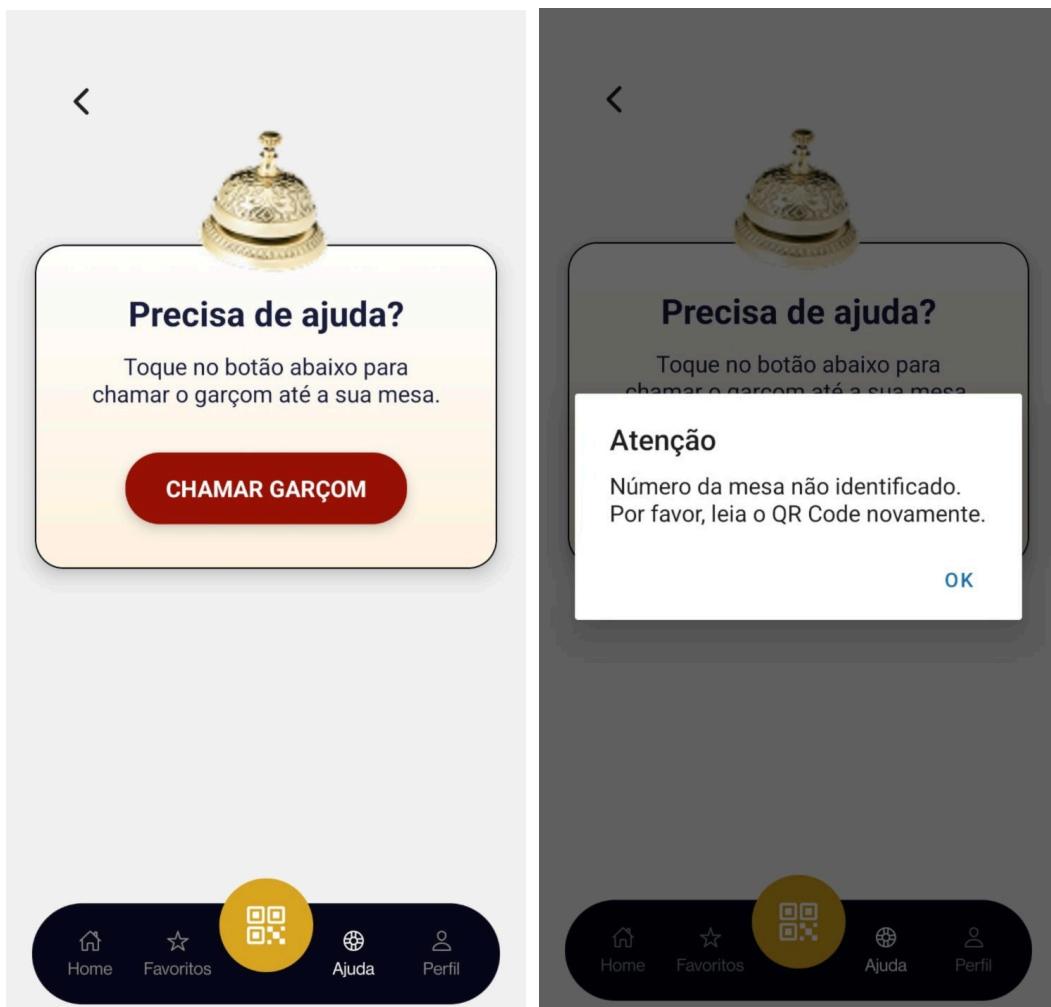
Fonte: Autoria (2025)

**Figura C.10 - Card de Produto Favoritado na Página de Favoritos e no Dashboard**



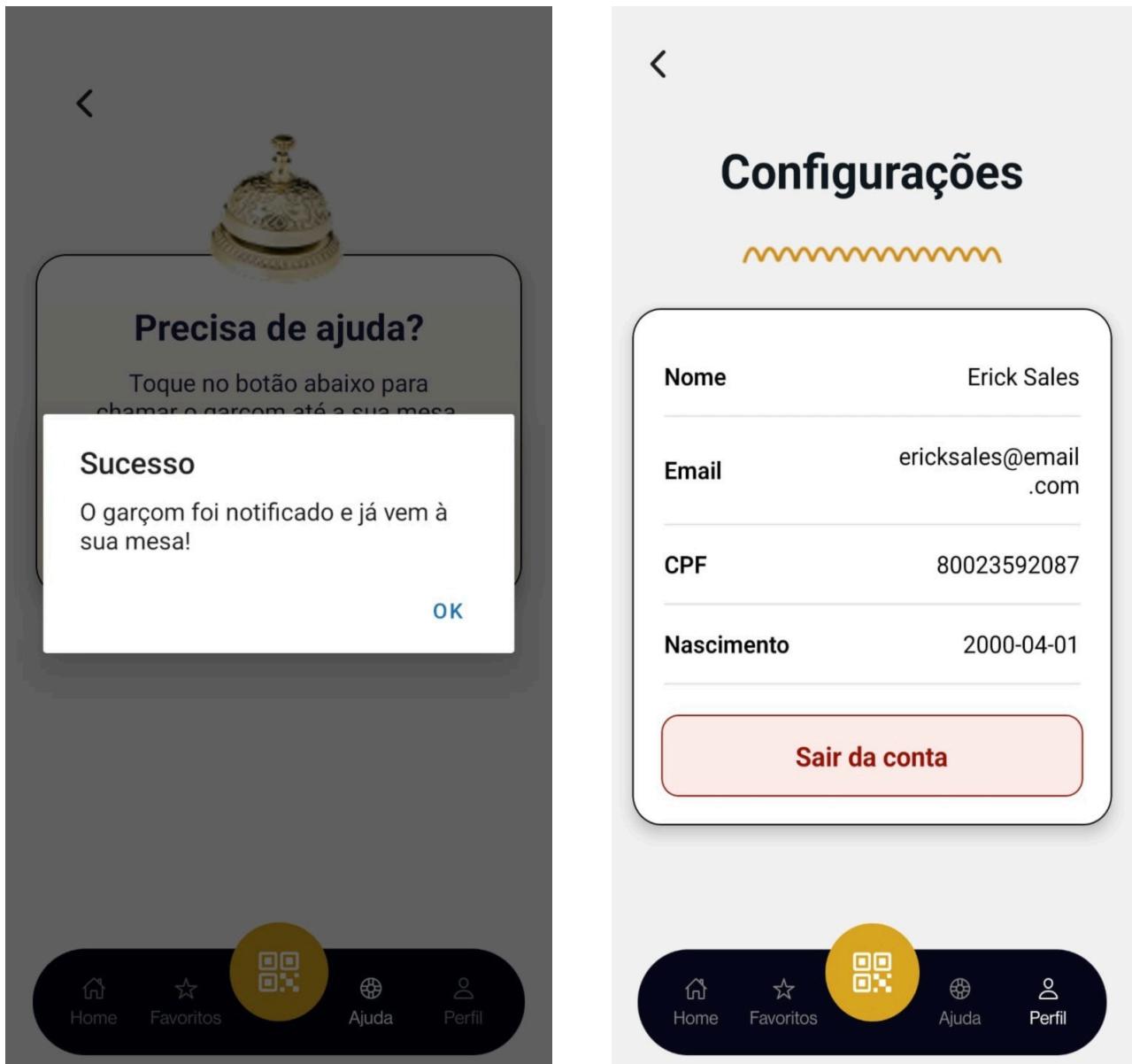
Fonte: Autoria (2025)

**Figura C.11 - Páginas de Pedir Ajuda “Chamar Garçom” e Erro de Leitura do QR Code**



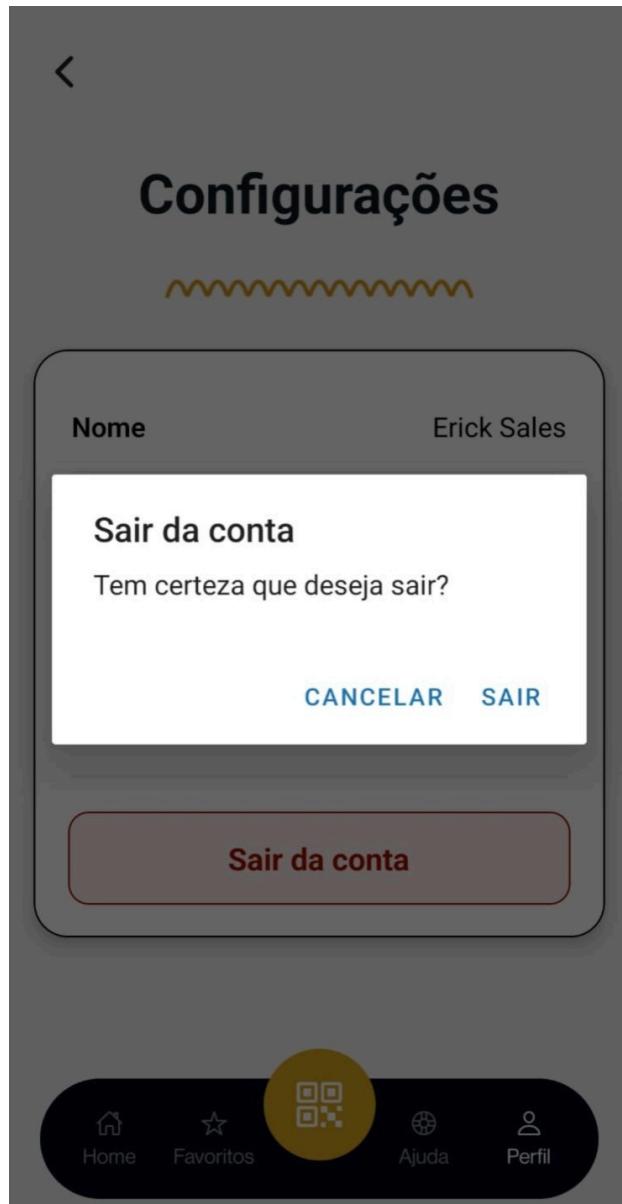
Fonte: Autoria (2025)

Figura C.12 - Páginas de Pedir Ajuda Bem-Sucedida e Perfil



Fonte: Autoria (2025)

Figura C.12 - Alerta de Logout



Fonte: Autoria (2025)

Dando continuidade ao fluxo de navegação, esta seção detalha o protótipo de alta fidelidade da interface Web, projetada para os usuários de Gestão (Gerência) e Produção (Cozinha). O material demonstra a aplicação dos princípios de design na construção de ferramentas administrativas e operacionais, essenciais para o controle completo do sistema Ártemis.

**Figura C.13 - Página web de cadastro**

The figure consists of two screenshots of a web-based registration form for 'ARTEMIS'.  
The top screenshot shows an empty registration form. The fields are as follows:

- Name: Digite seu nome...
- Email: Digite seu email...
- CPF: Digite seu CPF...
- Date of Birth: dd/mm/aaaa (with a calendar icon)
- Password: \*\*\*\*\*
- Role: Erro ao carregar cargos (dropdown menu)

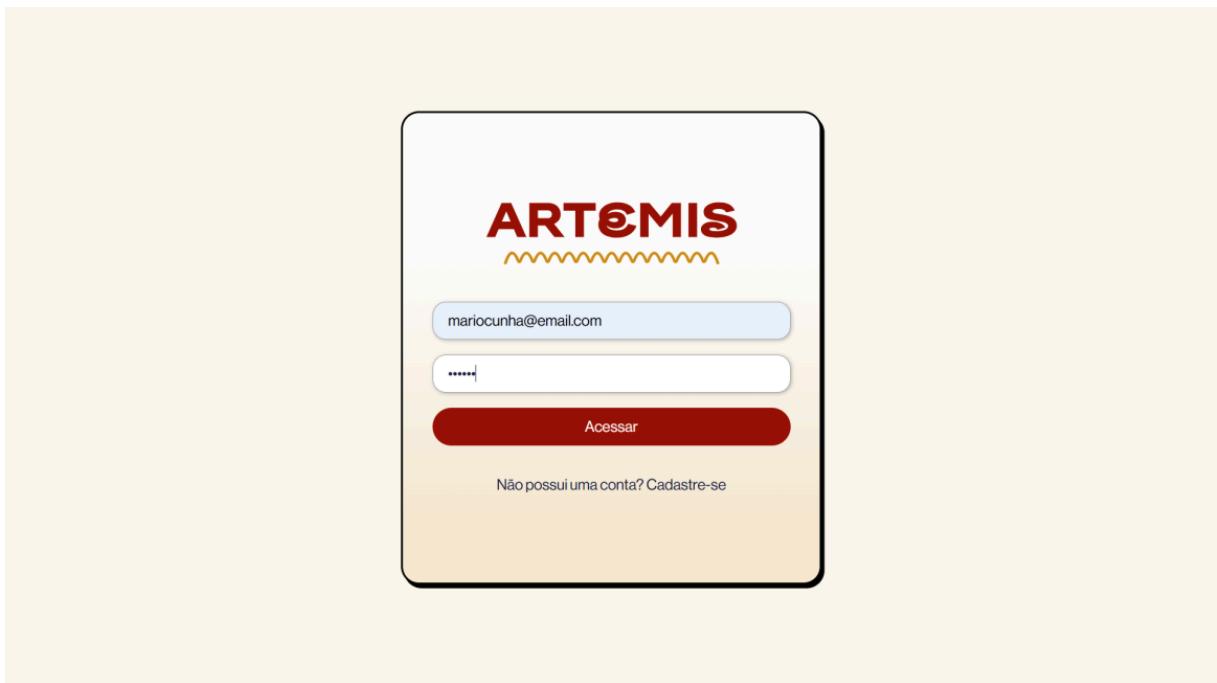
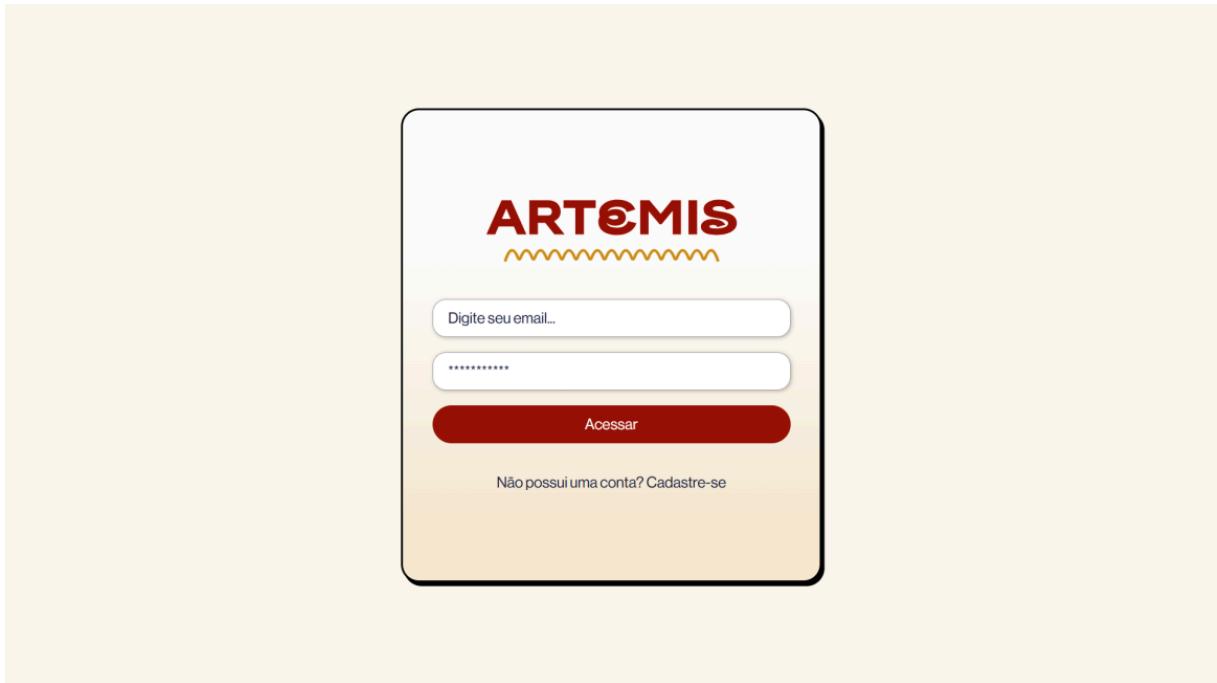
A red 'Cadastrar' button is at the bottom, and a link 'Já possui uma conta? Faça o login' is at the bottom right.  
The bottom screenshot shows the same form with the following data entered:

- Name: Mario Cunha
- Email: mariocunha@email.com
- CPF: 704.241.530-25
- Date of Birth: 21/07/2005
- Password: \*\*\*\*\*
- Role: Gerente (selected from dropdown)

The red 'Cadastrar' button and the 'Já possui uma conta? Faça o login' link are also present.

Fonte: Autoria (2025)

Figura C.14 - Página web de login



Fonte: Autoria (2025)

Figura C.15 - Página *web* de painel de pedidos

The figure consists of two vertically stacked screenshots of a web-based dashboard titled "ARTEMIS".

**Screenshot 1:** This screenshot shows a very sparse "Últimos pedidos" (Last orders) section. It contains a single, isolated text entry: "Mesa 13".

**Screenshot 2:** This screenshot shows a more populated "Últimos pedidos" section. It lists several order entries, each with a small circular icon next to it. The entries include:

- Mesa 13
- Mesa 14
- Mesa 15
- Mesa 16
- Mesa 17
- Mesa 18
- Mesa 19
- Mesa 20
- Mesa 21
- Mesa 22
- Mesa 23
- Mesa 24
- Mesa 25
- Mesa 26
- Mesa 27
- Mesa 28
- Mesa 29
- Mesa 30
- Mesa 31
- Mesa 32
- Mesa 33
- Mesa 34
- Mesa 35
- Mesa 36
- Mesa 37
- Mesa 38
- Mesa 39
- Mesa 40
- Mesa 41
- Mesa 42
- Mesa 43
- Mesa 44
- Mesa 45
- Mesa 46
- Mesa 47
- Mesa 48
- Mesa 49
- Mesa 50
- Mesa 51
- Mesa 52
- Mesa 53
- Mesa 54
- Mesa 55
- Mesa 56
- Mesa 57
- Mesa 58
- Mesa 59
- Mesa 60
- Mesa 61
- Mesa 62
- Mesa 63
- Mesa 64
- Mesa 65
- Mesa 66
- Mesa 67
- Mesa 68
- Mesa 69
- Mesa 70
- Mesa 71
- Mesa 72
- Mesa 73
- Mesa 74
- Mesa 75
- Mesa 76
- Mesa 77
- Mesa 78
- Mesa 79
- Mesa 80
- Mesa 81
- Mesa 82
- Mesa 83
- Mesa 84
- Mesa 85
- Mesa 86
- Mesa 87
- Mesa 88
- Mesa 89
- Mesa 90
- Mesa 91
- Mesa 92
- Mesa 93
- Mesa 94
- Mesa 95
- Mesa 96
- Mesa 97
- Mesa 98
- Mesa 99
- Mesa 100

Fonte: Autoria (2025)

Figura C.16 - Página web de status do pedido e produto

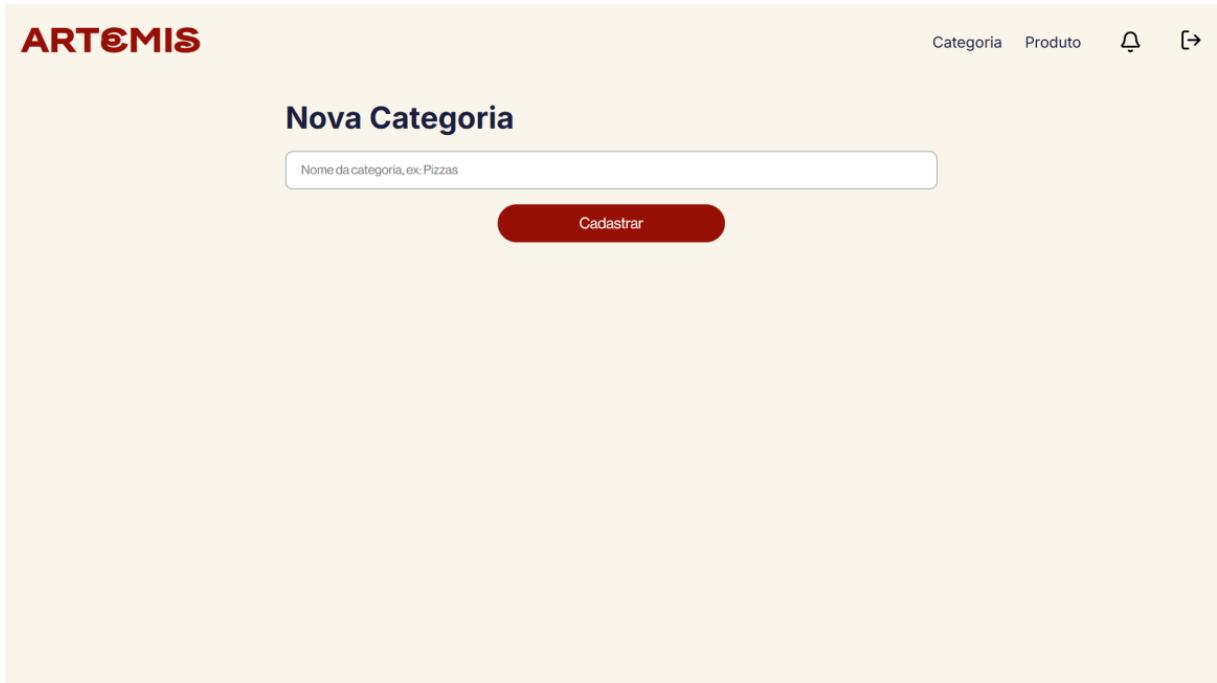
The figure consists of two vertically stacked screenshots of a software application interface, likely for a restaurant or food service. Both screenshots show a modal window titled "Detalhes do pedido" (Order Details) over a background screen labeled "Últimos pedidos" (Last Orders).

**Screenshot 1 (Top):** The modal shows an order for "Mesa 13". It lists "Qtd: 1 Portuguesa - R\$ 40" and includes a note: "Status: PENDENTE | Um recheio farto e saboroso, trazendo o gostinho caseiro da culinária lusa." A yellow button labeled "Concluir item" is visible. Below the note, it says "Valor Total: R\$ 40" and has a grey button labeled "Concluir pedido (PENDENTE)".

**Screenshot 2 (Bottom):** The modal shows the same order for "Mesa 13". The note now reads: "Status: CONCLUÍDO | Um recheio farto e saboroso, trazendo o gostinho caseiro da culinária lusa." The yellow "Concluir item" button is still present. Below the note, it says "Valor Total: R\$ 40" and has a red button labeled "Concluir pedido (PRONTO)".

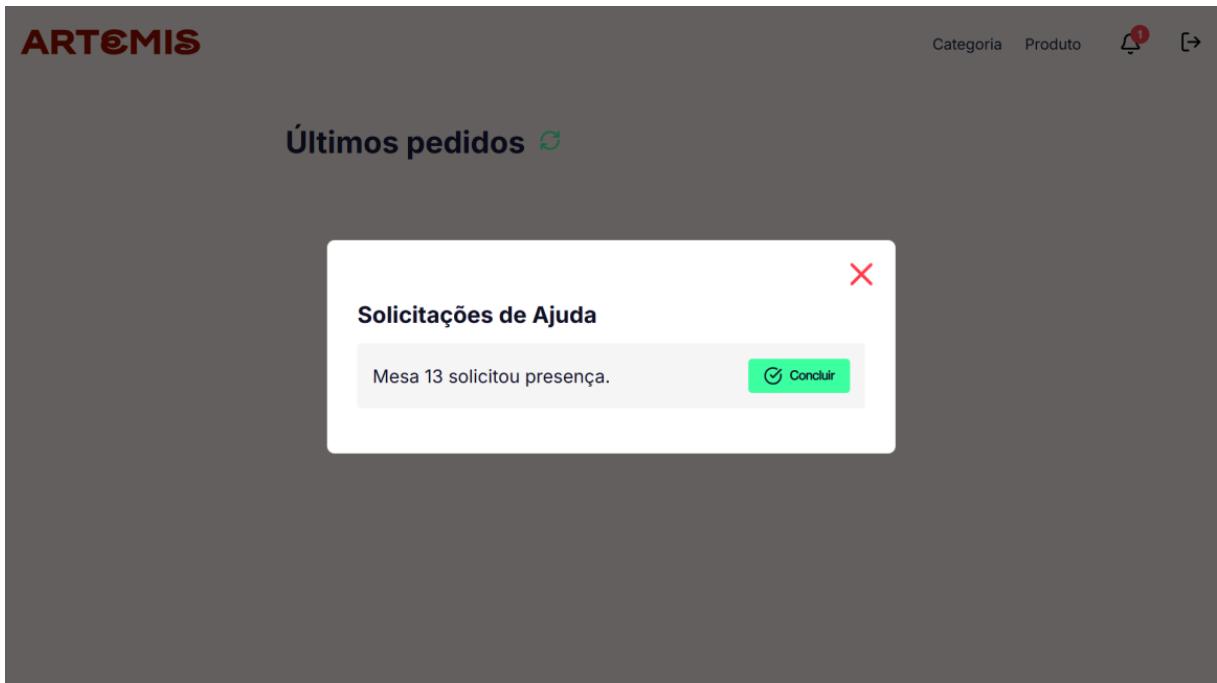
Fonte: Autoria (2025)

Figura C.17 - Página web de criação de categoria



Fonte: Autoria (2025)

Figura C.18 - Página web de solicitação de ajuda



Fonte: Autoria (2025)

Figura C.19 - Página web de criação do produto

The screenshot displays a web application interface for creating new products. At the top, there is a header with the brand name "ARTEMIS" and navigation links for "Categoria", "Produto", a bell icon, and a user profile icon.

The main area is titled "Novo produto". It features a large input field for a product image, followed by four input fields for basic product information: category (set to "Bebidas"), product name, price, and description. Below this section, there is a repeatable row for adding more products, which includes identical input fields for category, name, price, and description.

The bottom section is titled "Ingredientes" (Ingredients) and lists three ingredients with quantity controls (minus, zero, plus buttons):

- Calabresa R\$
- Molho de Tomate R\$
- Rodelas de Tomate R\$

At the bottom right of the ingredient list, there is a "Novo Ingrediente" (New Ingredient) button.

Fonte: Autoria (2025)

Figura C.20 - Página web de criação de ingredientes

The screenshot shows a web page titled "Novo ingrediente" (New ingredient) under the "Novo ingrediente" section of the ARTEMIS platform. At the top right are navigation links for "Categoria", "Produto", a bell icon, and a user profile icon. Below the title is a large input field with a placeholder arrow icon. Underneath are two smaller input fields: one for "Nome do ingrediente..." and another for "Preço adicional...". A red button at the bottom right labeled "Cadastrar ingrediente" (Register ingredient) is centered between the two input fields.

Fonte: Autoria (2025)

Figura C.21 - Figma do projeto



Fonte: Autoria (2025)

Figura C.22 - Protótipo do projeto



Fonte: Autoria (2025)

## APÊNDICE D - REPOSITÓRIO DO PROJETO

Para complementar a visão do processo de desenvolvimento e organização do projeto, este apêndice reúne a referência ao repositório digital principal utilizado ao longo do ciclo de vida da aplicação. A inclusão deste artefato serve como evidência central para o controle de versão e a colaboração da equipe, permitindo a rastreabilidade da estrutura de código e das decisões de desenvolvimento, e oferecendo um panorama técnico completo da execução e da governança do projeto.

**Figura D.1 - Qr code Repositório do Projeto**



Fonte: Elaborado pelo grupo (2025)