

## 1. Introdução

O Divvi é uma plataforma de gestão de pedidos presenciais que pretende inovar e automatizar o sistema de realização de pedidos em estabelecimentos alimentícios. A plataforma possui dois módulos, uma aplicação mobile para os usuários dos estabelecimentos, e o módulo web que funciona como um sistema PDV no gerenciamento de pedidos realizados no aplicativo. A aplicação móvel leva a forma de um cardápio online do estabelecimento, em que o usuário para obter acesso, deve efetuar a leitura de um QR Code que identifica a mesa que o mesmo está solicitando entrada. No Divvi, o usuário é capaz de realizar pedidos e dividir o mesmo com qualquer outro usuário presente na mesa, também é possível visualizar o andamento de seus pedidos, assim como a sua conta individual. Minimizando equívocos que geralmente ocorrem na divisão de conta em mesas com grande quantidade de clientes.

O módulo Web corresponde a um sistema de controle no progresso e administração dos pedidos que estão sendo efetuados no momento. Ao acessar o sistema é possível realizar o cadastro das mesas do estabelecimento, no qual cada mesa gerada possui o próprio QR Code, disponibilizado para acesso dos usuários. Ademais, o sistema permite a visualização completa da quantidade de mesas que se encontram livres e ocupadas, sendo capaz de observar o detalhamento de cada uma delas, contendo as informações dos usuários presentes e dos pedidos efetuados na mesma. Do mesmo modo que é possibilitado a visualização das mesas, é disponibilizado no módulo obter as informações detalhadas de cada pedido efetuado no sistema, sendo capaz de controlar o andamento de cada uma delas.

### 1.1. Delimitação do Problema

A motivação para o desenvolvimento desse projeto surgiu a partir da identificação de problemas recorrentes encontrados em restaurantes e lanchonetes. Dentre esses problemas, alguns geram muito desconforto e insatisfação nos clientes, como por exemplo, a demora no atendimento e entrega dos pedidos, principalmente quando o estabelecimento se apresenta lotado, inconsistências nos registros dos pedidos e até mesmo problemas na divisão da conta, quando há muitos clientes na mesa.

Assim foi entendido que o problema em comum dos usuários em relação a maneira atual de fazer pedidos, era a comunicação entre cliente e atendente. Dessa forma a proposta do Divvi é proporcionar uma interação mais eficiente entre o estabelecimento e o cliente, de forma que o mesmo não se sinta negligenciado ou desassistido durante o atendimento.

### 1.2. Objetivos do Projeto

#### 1.2.1. Objetivo Geral

O projeto tem por objetivo construir uma plataforma para o gerenciamento de pedidos presenciais, de forma automatizada, para estabelecimentos alimentícios, como restaurantes e lanchonetes. A plataforma é composta por dois módulos, um módulo web acessado pelos gestores dos estabelecimentos que permite gerenciar os pedidos feitos pelos seus clientes, e

um módulo mobile, que possibilita aos clientes solicitar e monitorar seus pedidos em tempo real.

### 1.2.2. Objetivos Específicos

- 1.2.2.1. Diminuir as falhas e inconsistências entre cliente e atendente;
- 1.2.2.2. Padronizar o sistema de pedidos presenciais por aplicativos;
- 1.2.2.3. Reduzir custos no estabelecimento.

### 1.3. Metodologia do Projeto e de Desenvolvimento do Software

A metodologia utilizada no desenvolvimento do projeto foi a SCRUM, sendo um tipo de metodologia ágil especializada em desenvolvimento de softwares. Conforme é feito no SCRUM, o projeto foi dividido em ciclos chamados de *Sprints*, em que consiste no período de tempo que um determinado conjunto de atividades é executado pela equipe. Cada subdivisão do projeto (API, Web e Mobile) tem seus próprios *Sprints*, para melhor organização e composição da equipe. Os padrões de projeto utilizados durante o desenvolvimento da aplicação foram:

- DDD (*Domain-Driven Design* - Design orientado a domínio), utilizado na estrutura da API;
- MVC (Model-View-Controller), utilizado em conjunto com o DDD na estrutura da API;
- Micro Serviços, cada funcionalidade da API foi desenvolvida como um serviço a ser consumido pelas funcionalidades dos módulos Mobile e Web;

### 1.4. Cronograma das atividades

Atividades		Descrição	Envolvidos	Duração (Dias)
Protótipo	Baixa Fidelidade	Prototipagem do modelo de baixa fidelidade do sistema a partir dos requisitos.	Bruna Cardoso Andrade; Brenda Cardoso Andrade;	5
	Alta Fidelidade	Prototipagem do modelo de alta fidelidade do sistema com base nas pesquisas de experiência do usuário.	Brenda Cardoso Andrade;	35
Modelagem de Domínio		Modelagem do domínio da aplicação a partir dos requisitos.	Bruna Cardoso Andrade; Gabriel Luiz Santos de Santana; Leonardo Sena do Vale da Silva;	20

			Lucas Cassimiro Tranzillo Nogueira; Tércio Carneiro de Carvalho.	
	<b>Modelagem de Dados</b>	Modelagem do modelo de dados da aplicação a partir dos requisitos e domínio criado.	Gabriel Luiz Santos de Santana; Tércio Carneiro de Carvalho;	10
	<b>Criação Física do Modelo de Dados</b>	Criação do script para gerar o banco de dados com base no modelo de dados e domínio.	Gabriel Luiz Santos de Santana; Tércio Carneiro de Carvalho;	6
<b>Desenvolvimento</b>	<b>API</b>	Desenvolvimento da API do projeto que será consumida pelos módulos Web e Mobile.	Bruna Cardoso Andrade; Gabriel Luiz Santos de Santana; Leonardo Sena do Vale da Silva; Lucas Cassimiro Tranzillo Nogueira; Tércio Carneiro de Carvalho.	150
	<b>Web</b>	Desenvolvimento FrontEnd do módulo Web.	Bruna Cardoso Andrade; Lucas Cassimiro Tranzillo Nogueira; Vitor Carvalho Pontes;	44
	<b>Mobile</b>	Desenvolvimento do módulo mobile.	Bruna Cardoso Andrade; Leonardo Sena do Vale da Silva; Tércio Carneiro de Carvalho.	82

Tabela 1: Cronograma das atividades realizadas

## 2. Descrição Geral do Sistema

### 2.1. Descrição do Problema

Estabelecimentos alimentícios como bares e restaurantes, geralmente são frequentados por grupos numerosos, e por conta disso é comum ocorrerem equívocos durante o atendimento. Ao finalizar a conta em uma mesa volumosa, por exemplo, é habitual que surjam problemas, já que normalmente a separação entres os itens consumidos ocorre de forma manual pelos próprios clientes. Erros como divisão incorreta de valores para pagamento, quantidade de itens inseridos na comanda de forma equivocada, falta de atenção ou comunicação para registrar observações relacionada ao pedido, itens trocados ou incorretos.

## 2.2. Principais Envolvidos e suas Características

### 2.2.1. Utilizadores do Sistema

Utilizador do Sistema	Descrição
Proprietário(a) e funcionários de estabelecimentos alimentícios.	Esses utilizadores, terão acesso ao módulo Web.
Clientes dos estabelecimentos alimentícios.	Esses utilizadores, terão acesso ao módulo Mobile.

Tabela 2: Descrição dos utilizadores do sistema.

### 2.2.2. Desenvolvedores do Sistema

Nome da Parte Interessada	Papel no Projeto	Dados de Contato
Bruna Cardoso Andrade	Gerente do projeto e Desenvolvedora	(71) 9 9950-9445
Gabriel Luiz Santos de Santana	Desenvolvedor	(71) 9 9253-9794
Leonardo Sena do Vale da Silva	Desenvolvedor	(71) 9 9253-9794
Lucas Cassimiro Tranzillo Nogueira	Desenvolvedor	(71) 9 9926-2746
Tércio Carneiro de Carvalho	Desenvolvedor	(71) 9 9228-4977
Márcio Renê Brandão Sousa	Orientador Técnico	(71) 9 9123-8870
Brenda Cardoso Andrade	Design UX/UI	(71) 9 99721725
Danilo José Pereira Delgado	Representante da Empresa	(71) 9 9199-9003

Tabela 3: Descrição dos desenvolvedores (diretos e indiretos) do sistema.

## 2.3. Regras de Negócio

<b>Identificador</b>	RN-001
----------------------	--------

<b>Nome</b>	Acesso ao aplicativo.		
<b>Data de criação</b>	02/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	02/05/2020	<b>Autor</b>	Bruna Andrade
<b>Versão</b>	N/A	<b>Dependências</b>	RF-001
<b>Descrição</b>	A única forma de cadastro ou acesso ao aplicativo será utilizando o número de celular do usuário. Após submeter o número, o usuário deve ser redirecionado para a tela de ativação, no qual deve receber uma mensagem (SMS) com o código de validação. O status do usuário deve ser preenchido como "Pendente" e o mesmo deve adicionar o código de ativação nos determinados campos da tela. Após submeter os dados, o status do usuário deve ser alterado para "Validado" e o aplicativo deve verificar a existência de um cadastro com os dados fornecidos pelo usuário, caso tenha o mesmo deve ser redirecionado para "Home" do aplicativo. Caso não seja encontrado dados semelhantes, o usuário deve ser direcionado para preencher as informações de cadastro.		

<b>Identificador</b>	RN-002		
<b>Nome</b>	Cadastro de dados do usuário.		
<b>Data de criação</b>	02/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	03/05/2020	<b>Autor</b>	Bruna Andrade
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	O cadastro do usuário só é possível após a ativação da conta pelo número e o mesmo com o status de "Validado". Deve ser requerido do usuário, obrigatoriamente, as informações de nome completo, email, estado e cidade. Como informação opcional, o usuário é capaz de submeter ao campo, uma imagem de perfil. Após a realização do		

cadastro, o status do usuário é alterado para “Aprovado” e o mesmo deve ser redirecionado para a “Home”. O aplicativo deve permitir que o usuário consiga editar suas informações de cadastro.

<b>Identificador</b>	RN-003		
<b>Nome</b>	Acesso ao estabelecimento.		
<b>Data de criação</b>	02/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	03/05/2020	<b>Autor</b>	Tércio Carvalho
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	<p>O acesso do usuário ao estabelecimento deve ser feito através da leitura de um QR Code. Ao fazer a leitura, o sistema deve verificar se o usuário é a primeira pessoa a entrar na mesa, caso seja, este se torna “Gerente” e inicia a sessão na determinada mesa. Caso já exista uma sessão em andamento, deve ser enviada uma requisição ao “Gerente” da mesa, para permitir ou negar a entrada deste usuário. Ao entrar na sessão o status do usuário deve ser preenchido como “Vinculado”, caso a solicitação de entrada seja negada, deve receber o status de “Rejeitado”. Apenas o “Gerente” da sessão, deve possuir permissão para retirar outros usuários presentes na mesa. Ao remover um usuário, o mesmo deve ter o status alterado para “Removido”. A opção de saída da sessão só deve ser permitida quando o usuário não possuir pendência financeira na sessão. Caso o usuário saia da sessão, seu status deve ser substituído para “Desconectado”. Caso o usuário possua pedidos em aberto, o mesmo deve receber o status de “Conta Pendente”.</p>		

<b>Identificador</b>	RN-004		
<b>Nome</b>	Realizar pedido.		
<b>Data de criação</b>	13/05/2020	<b>Autor</b>	Bruna Andrade

<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	<p>A visualização dos itens do cardápio do estabelecimento deve ser separada por categorias e exibidos para o usuário. O sistema deve permitir que o usuário adicione qualquer item do cardápio, e que o mesmo possa ser dividido entre qualquer usuário presente na mesa. Ao adicionar um ou mais itens, esses devem ser preenchidos na “Bandeja”, que deve exibir todos os itens selecionados com os seus respectivos valores, usuários que participaram da divisão e quantidade. A “Bandeja” deve possuir uma opção de acrescentar ou reduzir a quantidade de um item selecionado. Caso a quantidade atinja zero, o item deve ser removido da “Bandeja”. Deve ser exibido para o usuário o valor total dos itens presentes na “Bandeja”, assim como a opção de realizar o pedido. Caso essa opção seja selecionada, o pedido deve ser enviado ao estabelecimento com o status de “Pendente”.</p>		

<b>Identificador</b>	RN-005		
<b>Nome</b>	Visualização das informações.		
<b>Data de criação</b>	13/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	14/05/2020	<b>Autor</b>	Tércio Carvalho
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	<p>O usuário deve ter total visualização de todos os usuários que estão presentes na mesa, assim como os pedidos realizados. Identificando quem foi o(a) criador(a) de cada pedido, seus participantes, valor total, valor dividido (caso tenha participantes) e o status atual. Os pedidos poderão ter cinco status diferentes, referente ao tipo de ação tomada pelo usuário ou estabelecimento. Após o pedido ser realizado pelo usuário, o mesmo permanece como “Pendente” até ser aprovado pelo estabelecimento, que é altera para “Confirmado” quando esta ação ocorre. “Em Preparo” deve ser o status do pedido quando o</p>		

estabelecimento iniciar o preparo. Caso o pedido seja cancelado ou rejeitado, o status deve ser alterado para "Cancelado", assim como a justificativa deve ser enviada ao usuário. Por fim, o status de "Entregue" deve ser aplicado, quando o pedido for enviado ao usuário. O sistema também deve fornecer para cada usuário a visualização detalhada do consumo total individual na mesa.

<b>Identificador</b>	RN-006		
<b>Nome</b>	Pagamento.		
<b>Data de criação</b>	13/05/2020	<b>Autor</b>	Tércio Carvalho
<b>Data da última alteração</b>	14/05/2020	<b>Autor</b>	Bruna Andrade
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	O usuário deverá ser capaz de solicitar o pagamento da conta através da aplicação, o que acionará os responsáveis do estabelecimento para dirigir-se ao usuário e realizar o pagamento de forma presencial. Após a confirmação de pagamento do estabelecimento, o usuário deverá ser retirado da sessão e receber um recibo digital. A sessão só poderá ser finalizada após todos os usuários presentes na mesma, tiverem pagamento confirmado pelo estabelecimento.		

<b>Identificador</b>	RN-007		
<b>Nome</b>	Acesso ao Sistema		
<b>Data de criação</b>	13/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	14/05/2020	<b>Autor</b>	Bruna Andrade



<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	A única forma de acesso ao sistema, será utilizando email e senha do usuário. Após submeter esses dados, o sistema deve validar se o usuário está ativado ou possui informações pendentes que necessitam de análise. Caso não haja dados incompletos, o sistema deve redirecionar para a DashBoard. Caso existam informações pendentes, o usuário deve permanecer em uma tela de validação referente a informação que falta.		

<b>Identificador</b>	RN-008		
<b>Nome</b>	Cadastro das informações do usuário		
<b>Data de criação</b>	13/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	14/05/2020	<b>Autor</b>	Bruna Andrade
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	O cadastro do usuário, é iniciado apenas com a submissão de email e senha. Após a finalização desse cadastro inicial o sistema deve enviar um email de confirmação para o usuário, para que o mesmo ative sua conta no sistema. Caso o usuário ative a sua conta, é possível finalizar o cadastro colocando as suas informações pessoais: Nome completo, CPF, RG, documento com foto e opcionalmente uma foto de perfil para o usuário. Caso todas as informações estejam corretas, o usuário deve ser direcionado para o cadastro do estabelecimento, neste deve conter: Nome do estabelecimento, nome fantasia, CNPJ, email, descrição, contrato social e a imagem com a logo do estabelecimento. Com a finalização desta etapa do cadastro, o usuário deve ser mantido em uma tela de validação, já que o estabelecimento permanecerá com o status de "Em Análise". O usuário só poderá ter acesso a DashBoard do sistema, após ser analisado e aprovado pela equipe.		

<b>Identificador</b>	RN-009		
<b>Nome</b>	Cadastro das informações do usuário		
<b>Data de criação</b>	15/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	<p>O cadastro do usuário, é iniciado apenas com a submissão de email e senha. Após a finalização desse cadastro inicial o sistema deve enviar um email de confirmação para o usuário, para que o mesmo ative sua conta no sistema. Caso o usuário ative a sua conta, é possível finalizar o cadastro colocando as suas informações pessoais: Nome completo, CPF, RG, documento com foto e opcionalmente uma foto de perfil para o usuário. Caso todas as informações estejam corretas, o usuário deve ser direcionado para o cadastro do estabelecimento, neste deve conter: Nome do estabelecimento, nome fantasia, CNPJ, email, descrição, contrato social e a imagem com a logo do estabelecimento. Com a finalização desta etapa do cadastro, o usuário deve ser mantido em uma tela de validação, já que o estabelecimento permanecerá com o status de "Em Análise". O usuário só poderá ter acesso a DashBoard do sistema, após ser analisado e aprovado pela equipe.</p>		

<b>Identificador</b>	RN-010		
<b>Nome</b>	Visualização das informações pessoais.		
<b>Data de criação</b>	15/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A

<b>Descrição</b>	Após o usuário ter realizado login, o sistema deve direcioná-lo para a tela da DashBoard. Na página inicial os usuários devem visualizar as informações de um histórico semanal dos pedidos, contendo quantidade e valor total por cada dia. O sistema também deve fornecer a opção de visualizar as suas informações pessoas e alterá-las, assim como as do estabelecimento.
------------------	---

<b>Identificador</b>	RN-011		
<b>Nome</b>	Visualização das informações das mesas.		
<b>Data de criação</b>	15/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	Na tela de Mesas, o sistema deve permitir que o usuário adicione o número de mesas desejado, e gerar um QR Code para cada uma delas. Após a criação das mesas, o sistema deve listar as mesas geradas pelo usuário, e possibilitar que o mesmo visualize todas as mesas, apenas as ocupadas e somente as livre. Também deve ser proporcionado um filtro de busca pelo número da mesa e um detalhamento dela, contendo uma lista de pessoas presentes na mesa e os pedidos que estão sendo feitos nela, caso a mesma esteja ocupada. Já a mesa estando livre, deve ser possibilitado ao usuário as informações da mesa e a opção de imprimir novamente o QRCode.		

<b>Identificador</b>	RN-012		
<b>Nome</b>	Visualização das informações dos pedidos.		
<b>Data de criação</b>	15/05/2020	<b>Autor</b>	Bruna Andrade

<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	<p>Na tela de Pedidos, o sistema deve listar todos os pedidos que estão sendo realizados no momento no estabelecimento. Os pedidos devem ser separados por status: Pendente, Confirmado, Entregue e Cancelado. Para os pedidos que estão como Pendente, o sistema deve possibilitar duas ações ao usuário, a de confirmar ou cancelar o pedido. Caso o pedido seja confirmado, o mesmo deve ter o status alterado para "Confirmado" e ser realocado para lista deste status. Caso o mesmo seja cancelado, o usuário deve informar o motivo do cancelamento e o mesmo será também realocado para a lista deste status. Nos pedidos confirmados, o sistema deve fornecer a opção de entregar o pedido ao cliente, e da mesma forma que os outros, caso seja entregue, o pedido deve ser realocado para lista deste status. Em todas as listagens, independente do status devem ser visualizados: o nome da pessoa que realizou o pedido, os itens solicitados, a quantidade de cada item e o valor total do pedido.</p>		

### 3. Requisitos do Sistema

#### 3.1. Requisitos Funcionais

<b>Identificador</b>	RF-001		
<b>Nome</b>	Realizar login.		
<b>Data de criação</b>	13/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RN-001

<b>Descrição</b>	O aplicativo deve permitir ao usuário preencher o campo de telefone, para realizar o login no aplicativo.
------------------	---

<b>Identificador</b>	RF-002		
<b>Nome</b>	Cadastro e alterações de dados.		
<b>Data de criação</b>	20/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RN-002
<b>Descrição</b>	O aplicativo deve permitir ao usuário, cadastrar seus dados pessoais e alterá-los.		

<b>Identificador</b>	RF-003		
<b>Nome</b>	Leitura do QR Code.		
<b>Data de criação</b>	20/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RN-003

<b>Descrição</b>	O QR Code deve conter um identificador que reconheça o número da mesa do determinado estabelecimento.
------------------	---

<b>Identificador</b>	RF-004		
<b>Nome</b>	Realizar pedido.		
<b>Data de criação</b>	20/05/2020	<b>Autor</b>	Tércio Carvalho
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	O usuário deverá ser capaz de realizar seus pedidos no estabelecimento através da aplicação.		

<b>Identificador</b>	RF-005		
<b>Nome</b>	Visualização das informações.		
<b>Data de criação</b>	20/05/2020	<b>Autor</b>	Tércio Carvalho
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A

<b>Descrição</b>	O usuário deverá ser capaz de acompanhar todas as informações de suas atividades no estabelecimento através da aplicação (Pedidos, valores, pessoas vinculadas a sua mesa).
------------------	---

<b>Identificador</b>	RF-006		
<b>Nome</b>	Solicitação de login.		
<b>Data de criação</b>	15/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RN-008
<b>Descrição</b>	O sistema deve permitir ao usuário preencher os campos de email e senha, para realizar o login na aplicação.		

<b>Identificador</b>	RF-007		
<b>Nome</b>	Cadastro e alteração de informações		
<b>Data de criação</b>	15/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RN-009

<b>Descrição</b>	O sistema deve permitir o usuário, cadastrar seus dados pessoais e a alterá-los. caso seja necessário
------------------	---

<b>Identificador</b>	RF-008		
<b>Nome</b>	Visualização das informações.		
<b>Data de criação</b>	15/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RN-010, RN-011 e RN-012
<b>Descrição</b>	O usuário deverá ser capaz de acompanhar e administrar todas as informações das ações ocorridas no estabelecimento.		

### 3.2. Requisitos Não-Funcionais

<b>Identificador</b>	RNF-001		
<b>Nome</b>	Envio do código de validação.		
<b>Data de criação</b>	20/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RF-001



<b>Descrição</b>	O envio do código de validação ao usuário para a ativação do número, deve ser feito utilizando o serviço Amazon SNS (Amazon Simple Notification Service).
------------------	---

<b>Identificador</b>	RNF-002		
<b>Nome</b>	Upload de imagem.		
<b>Data de criação</b>	20/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	RF-002
<b>Descrição</b>	As imagens submetidas pelo usuário no cadastro devem ser feitas através do serviço Amazon S3 (Servidor de homologação para imagens e documentos).		

<b>Identificador</b>	RNF-002		
<b>Nome</b>	Envio de email.		
<b>Data de criação</b>	20/05/2020	<b>Autor</b>	Bruna Andrade
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	v.01	<b>Dependências</b>	N/A
<b>Descrição</b>	O envio de email de confirmação de cadastro ou alteração de senha, devem ser feitos através do serviço Amazon SES (Serviço de envio de email para homologação).		

### 3.3. Protótipo

O protótipo de baixa fidelidade foi projetado utilizando o software WireframePro, desenvolvido a partir dos requisitos iniciais do projeto. Este protótipo está apresentado nos Anexo A e B. Já o protótipo de alta fidelidade, foi desenvolvido baseando no de baixa e a partir de pesquisas relacionadas à experiência de usuário. O mesmo está presente nos Anexos C e D, e foram preparados no software de prototipação Figma e reproduzido no desenvolvimento final.

## 4. Análise e Design

### 4.1. Arquitetura do Sistema

O modelo de arquitetura utilizado no desenvolvimento da API, foi o Domain Driven Design(DDD). Essa arquitetura consiste na divisão do projeto em camadas que já são padronizadas, e que possuem finalidades bem específicas dentro do projeto. A implementação dessa arquitetura possibilita que o projeto seja orientado pelo domínio, no qual o objetivo principal é que a implementação de complexas regras e processos de negócio, se tornem mais simples.

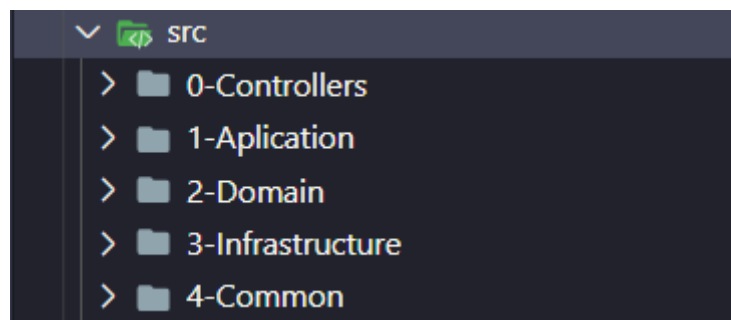


Figura 1: Arquitetura da API.

Fonte: Própria

### 4.2. Arquitetura DDD

As camadas do DDD implementadas no projeto são: apresentação, aplicação, infraestrutura e comum. Através dessas camadas existe um fluxo por onde as informações percorrem no sistema, entrando pela camada de apresentação e sendo permeada na camada de infraestrutura.

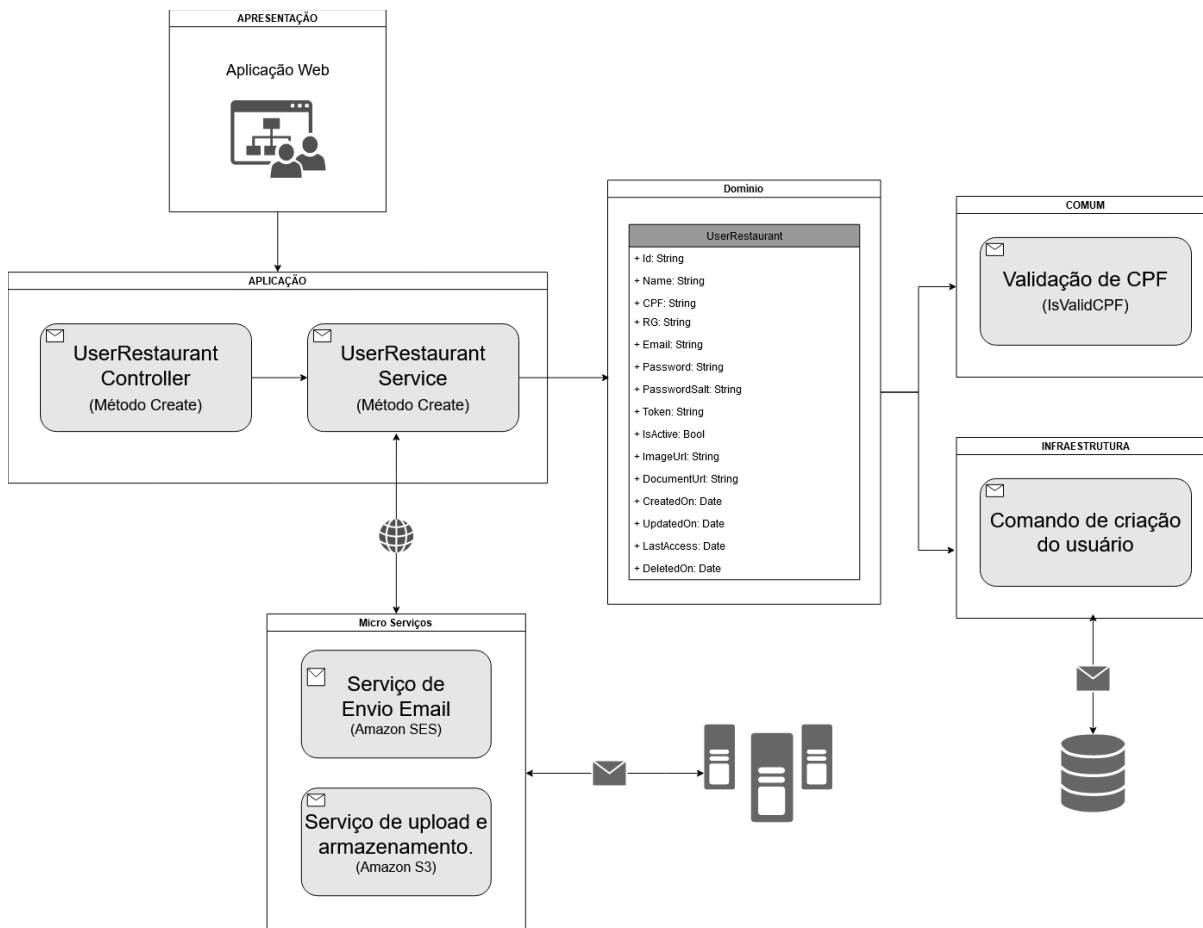


Figura 2: Arquitetura do fluxo de criação do usuário.

Fonte: Própria

#### 4.2.1. Camadas do DDD

- 4.2.1.1. **Apresentação:** É a porta de entrada para o sistema, nessa camada ficam as telas, formulários que são preenchidos pelos usuários e os controladores que são responsáveis por receber requisições HTTP e direcionar para a camada de aplicação.
- 4.2.1.2. **Aplicação:** É responsável pela comunicação e gerenciamento entre as camadas, até que o objetivo da requisição do usuário seja atendido. Também é possível realizar chamadas na camada de infraestrutura e no domínio do projeto.
- 4.2.1.3. **Domínio:** É responsável por representar lógicas do negócio e conceitos do sistema, nele são implementadas as entidades e relacionamentos do sistema. O foco da arquitetura DDD gira em torno das interações dessa camada.

- 4.2.1.4. Infraestrutura: A camada de infraestrutura é a camada responsável por acessar o banco de dados, seja no caso de consultas, atualizações, remoções ou inserções.
- 4.2.1.5. Comum: Nesta camada, estão presentes os métodos universais que não fazem parte do domínio e podem ser acessados por qualquer camada, como métodos de validação de CPF, CNPJ, email e telefone.

### 4.3. Ambiente de Desenvolvimento

O projeto foi desenvolvido na linguagem de programação JavaScript, utilizando o editor de código VSCode. A escolha desta linguagem ocorreu de forma estratégica, para que todas as frentes de desenvolvimento utilizassem a mesma, facilitando assim a rotatividade dos integrantes ao decorrer do processo.

No desenvolvimento FrontEnd, também foram utilizados os frameworks React e React Native, nos módulos Web e Mobile respectivamente. Além desse framework, durante o desenvolvimento foram adicionadas bibliotecas para a comunicação com a API (Axios), de leitura de QrCode (qrcode-scanner), para animações de componentes de carregamento (lottie), para *inputs* que necessitam de máscara (react-input-mask), para o tratamento de dados do tipo data (moment) e para a geração de PDFs contendo informações da mesa (jspdf).

Na construção da base de dados do projeto, foi usado o MySQL na versão 10.2, em conjunto com o modelo de mapeamento objeto-relacional (ORM) Sequelize. No controle de versionamento de todos os módulos do projeto, foi utilizado o Github.

### 4.4. Sistemas e componentes externos utilizados

O projeto utilizou três serviços externos da amazon em algumas funcionalidades do sistema. O AWS S3, para o armazenamento de imagens e documentos necessários pelo usuário e estabelecimento, o serviço de envio de email AWS SES utilizado na na confirmação de cadastro, e o AWS SNS servindo o envio de mensagens SMS para o usuário realizar a ativação de sua conta no aplicativo.

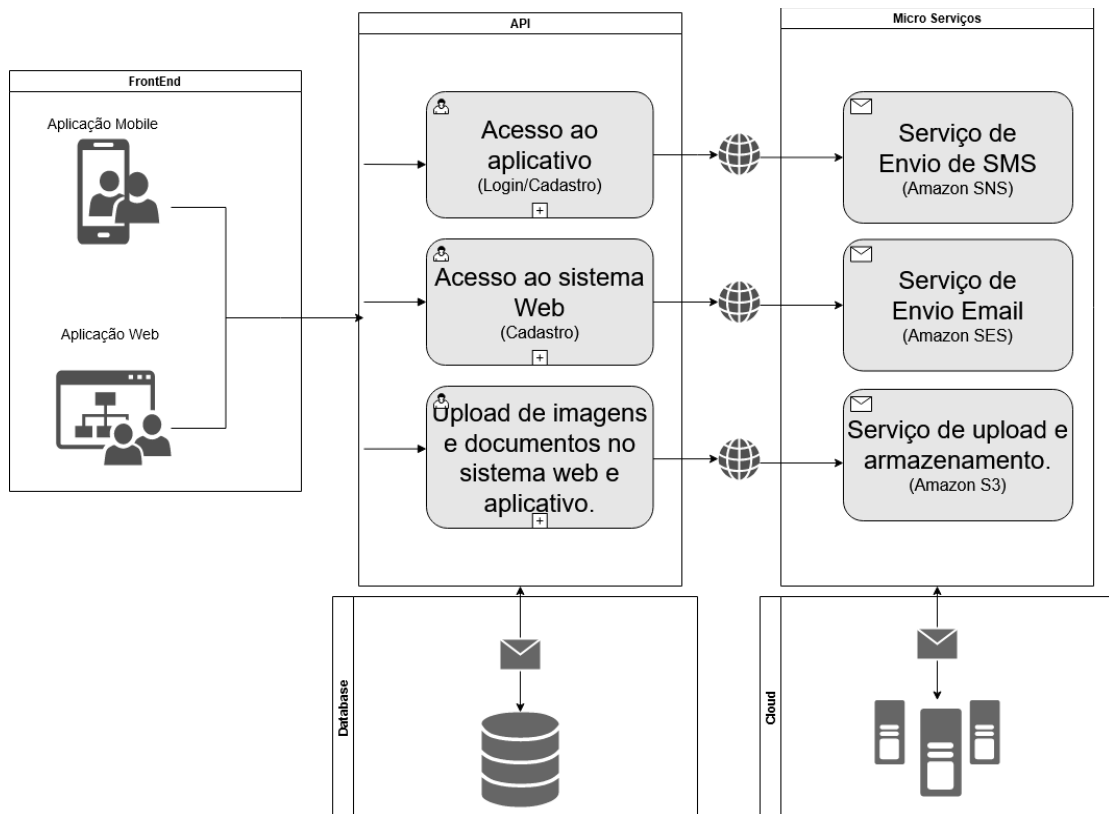


Figura 3: Arquitetura dos serviços externos utilizados.

Fonte: Própria

## 5. Implementação

A partir das diretrizes estabelecidas pelos requisitos e opções arquiteturais, o projeto foi desenvolvido a priori sendo subdividido em três grandes módulos, que integrados contemplam todas as funcionalidades necessárias na solução.

Para a implementação da solução base, do ponto de vista estrutural, seriam necessárias a hospedagem dos módulos web e servidor em servidores dedicados, hospedagem do banco de dados, pelo menos 1 computador simples no estabelecimento para acesso ao sistema web, publicação da aplicação mobile nas lojas de aplicativo dos sistemas (Android e iOS) e impressão das tags para as mesas.

Além disso, também deve ser considerado o treinamento dos funcionários na utilização e orientação sobre o sistema.

Nas seções seguintes estão detalhadas diretrizes básicas e funcionamento de cada módulo.

### 5.1. Servidor

Módulo responsável por toda intercomunicação e regras de negócio da aplicação. A plataforma escolhida para o seu desenvolvimento foi a Node Js, caracterizado como um ambiente de execução em Javascript baseado em eventos que funciona em um único thread,

economizando recursos de servidor. Além disso, ser baseado em javascript garantiu uma melhor comunicação com os demais módulos (também baseados em javascript). Entre as suas principais funcionalidades estão:

- Gerenciar as requisições dos módulos Web e Mobile;
- Aplicar todas regras de negócio e validações correspondentes;
- Gerenciar as interações com APIs e Serviços terceiros;
- Gerenciar a conexão e persistir as informações no banco de dados do sistema;

## 5.2. Mobile

Desenvolvido em react-native, o módulo mobile funciona tanto nas plataformas iOS quanto Android, sendo interligado ao módulo servidor através de conexões http utilizando a biblioteca axios e responsável pela interação do cliente com o restaurante. Através dele, o cliente pode:

- Fazer seu cadastro no sistema;
- Atualizar informações de perfil;
- Realizar seu vínculo a uma mesa do restaurante;
- Gerenciar e vincular outros participantes na mesa;
- Visualizar o cardápio;
- Realizar pedidos, adicionando suas observações e com a possibilidade de dividi-los entre os participantes da mesa, gerando contas separadas para cada um;
- Visualizar o status de cada um dos pedidos;
- Visualizar o valor da sua conta parcial de forma detalhada;
- Fazer o pedido de finalização da conta;
- Chamar o responsável pelo restaurante caso exista algum problema.

### 5.3. Web

O módulo web foi desenvolvido na linguagem javascript através do framework react, o que garantiu além das vantagens estruturais e de execução de um sistema em SPA (Single Page Application) nos permitiu ter uma velocidade de produção maior devido a sua fácil interligação com os demais módulos, também baseados em javascript, além do reaproveitamento de código e conhecimentos. Assim como o módulo mobile, este também é interligado ao módulo servidor através de conexões http utilizando a biblioteca axios. Esse módulo tem por principal finalidade o gerenciamento e atualização dos pedidos por parte do restaurante. Além disso, os funcionários também fazem o cadastro das mesas, impressão das tags de conexão e visualização das estatísticas dos últimos dias

## 6. Testes

Nesta etapa do projeto, ocorreu a revisão geral do sistema, com o intuito de validar as regras de negócio e as funcionalidades desenvolvidas para cada requisito.

### 6.1. Plano de Testes

Durante todo o processo de desenvolvimento, foram realizados testes a fim de verificar possíveis erros durante o desenvolvimento, baseado na estrutura TDD (*Test Driven Development*), com o intuito de diminuir refatorações no código em correções pontuais de bugs no processo final de desenvolvimento. Para a finalização do plano de testes, foram definidas duas rodadas para validação e correção do escopo geral do sistema.

### 6.2. Execução do Plano de Testes

Inicialmente foram feitos apenas testes baseados na metodologia TDD, focando apenas em refatorar possíveis problemas gerados em uma feature desenvolvida naquele momento. Em seguida, foram feitas duas rodadas de testes para verificação do sistema no geral, identificando e corrigindo problemas impeditivos para o funcionamento do sistema. Os não impeditivos por sua vez foram corrigidos em outro momento. Para a segunda rodada de testes, foram validadas as correções feitas na primeira, com o intuito de gerar a primeira release do Divvi, estável e sem bugs.

## 7. Considerações Finais e Recomendações

O desenvolvimento do projeto foi uma experiência muito edificante profissionalmente, através dele foi possível pôr em prática e verificar a importância de cada parte do planejamento de um projeto de grande porte. A análise de requisitos, pesquisa de similares e principalmente o planejamento do tempo de desenvolvimento das atividades são pontos cruciais que devem ser bem enfatizados para alcançar os resultados esperados.

Além disso, a escolha e aplicação de arquiteturas e padrões de desenvolvimento foi de grande relevância na organização das tarefas em grupo e do projeto como um todo.

Um outro ponto de grande importância para o andamento do projeto foi o estudo e definição das escolhas técnicas de maior impacto com antecedência, sendo possível uma escolha mais precisa que atendesse as necessidades do projeto de forma eficiente sem impactar o prazo de desenvolvimento dessa etapa com uma curva de aprendizado maior sobre a ferramenta.

Ao final, conseguimos desenvolver um produto que, com baixo custo estrutural para os estabelecimentos, é capaz de atender as necessidades e suprir as falhas no fluxo de atendimento atual estabelecidas nos objetivos iniciais.

É válido ressaltar que nesse projeto ainda existe espaço para novas funcionalidades que, apesar de não comporem o funcionamento base, podem complementá-lo e tornar a solução mais robusta e melhorar sua usabilidade para o restaurante e os clientes, como a integração ou implementação de um sistema de gerenciamento completo para o restaurante e a integração com um serviço de pagamento pelo aplicativo, que removeria a necessidade da interação direta com o restaurante no processo de pagamento, dando autonomia ao cliente e agilidade na finalização do serviço.



## 8. Bibliografia

CARVALHO, Bernardo; MELLO, Carlos. **Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica.** Gest. Prod., São Carlos, v. 19, n. 3, p. 557-573, 2012.

CHICHINELLI, Micheli; CAZARINI, Edson. **REQUISITOS NÃO FUNCIONAIS E SUA IMPORTÂNCIA NO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO.** Universidade de São Paulo – Escola de Engenharia de São Carlos - São Carlos.

SCRUM. Página inicial. Disponível em: <<https://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 08 de jun de 2020.

## ANEXO A - Protótipo de baixa fidelidade módulo Mobile



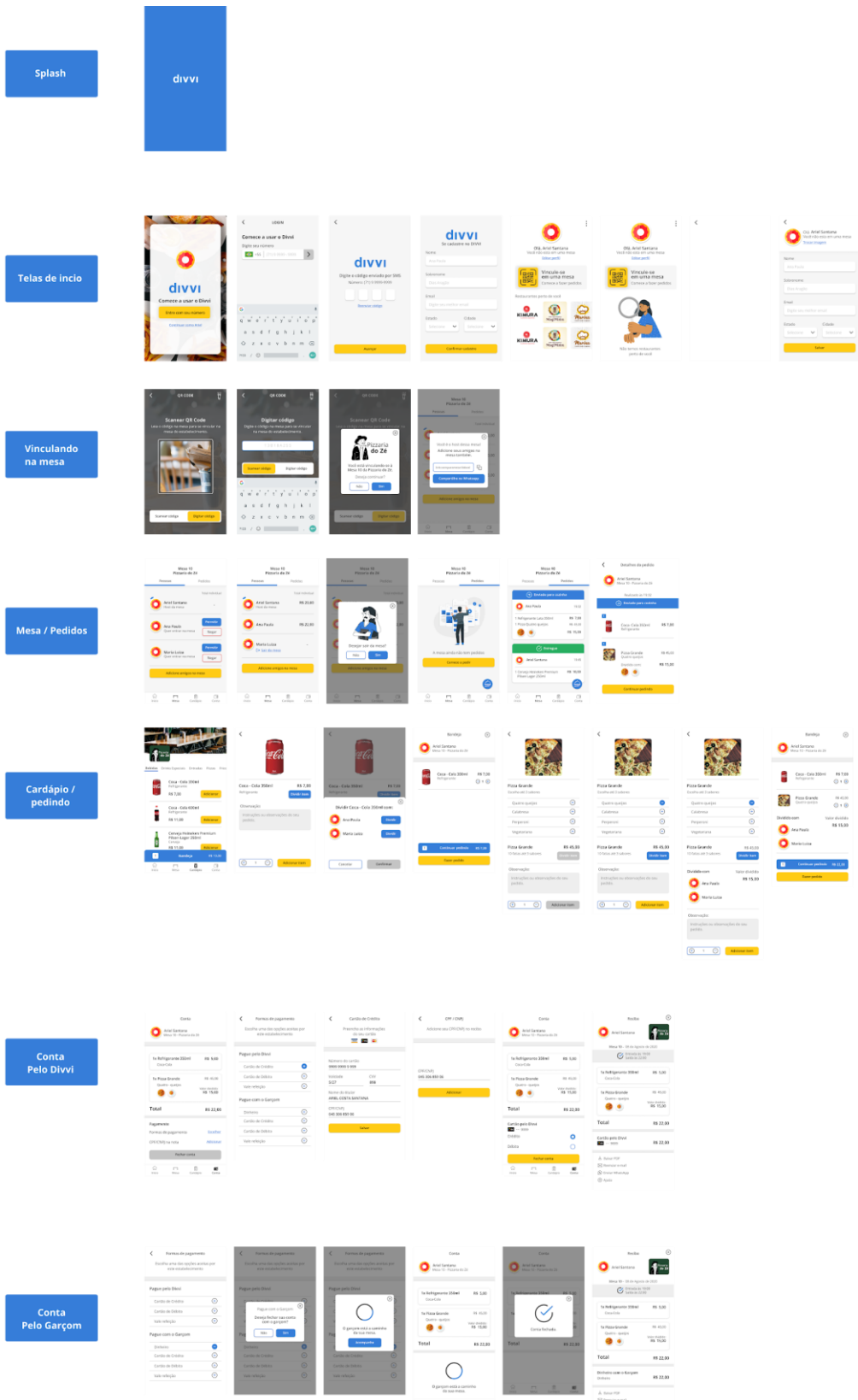
## ANEXO B - Protótipo de baixa fidelidade módulo Web





Documentação do Divvi, a plataforma de gerenciamento para pedidos presenciais. Bruna Cardoso Andrade, Gabriel Luiz Santos de Santana, Leonardo Sena do Vale da Silva, Lucas Cassimiro Tranzillo Nogueira e Tércio Carneiro de Carvalho.

## ANEXO C - Protótipo de alta fidelidade módulo Mobile





Documentação do Divvi, a plataforma de gerenciamento para pedidos presenciais. Bruna Cardoso Andrade, Gabriel Luiz Santos de Santana, Leonardo Sena do Vale da Silva, Lucas Cassimiro Tranzillo Nogueira e Tércio Carneiro de Carvalho.

### ANEXO D - Protótipo de alta fidelidade módulo Web

