

Universidade Paulista - UNIP

Gabriel Perinotto Lopes

**APLICAÇÃO PARA IMPLEMENTAÇÃO DE UM MODELO DE
RESTAURANTE SEM COZINHA**

**Limeira
2022**

Universidade Paulista - UNIP

Gabriel Perinotto Lopes

**APLICAÇÃO PARA IMPLEMENTAÇÃO DE UM MODELO DE
RESTAURANTE SEM COZINHA**

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciência da computação sob a orientação do professor Sérgio Eduardo Nunes.

**Limeira
2022
Sergio Eduardo Nunes**

APLICAÇÃO PARA IMPLEMENTAÇÃO DE UM MODELO DE RESTAURANTE SEM COZINHA

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em Ciência da Computação sob a orientação do professor Sérgio Eduardo Nunes.

Aprovada em XX de XXXXX de 201X.

BANCA EXAMINADORA

Prof. Dr. Nome completo

Prof. Me. Nome completo

Prof. Esp. Nome completo

DEDICATÓRIA

Dedico este trabalho a Deus
principalmente, também a minha família,
colegas e professores que me permitiram
chegar até aqui

*“Cada sonho que você deixa para trás é um
pedaço do seu futuro que deixa de existir”.*

(Steve Jobs)

RESUMO

Texto em parágrafo único, no máximo 500 palavras...

Palavra-Chave: até cinco palavras, separadas por ponto-e-vírgula.

ABSTRACT

Text...

Key Words: ...

LISTA DE FIGURAS

Figura 01 – Interação de Valores na Distribuição Normal no GeoGebra

13

LISTA DE QUADROS

Quadro 01 – Tipos de Distribuição Estatística

13

LISTA DE ABREVIATURAS

SUMÁRIO

INTRODUÇÃO	12
1.1 Objetivo	12
1.2 Justificativa	13
1.3 Metodologia	14
2. PRIMEIRO NÍVEL	13
2.1 Segundo Nível	13
2.1.1 Terceiro nível	13
CONCLUSÃO	14
REFERÊNCIAS BIBLIOGRÁFICAS	15

1. INTRODUÇÃO

Uma crise recente, que foi a pandemia ocasionada pelo Covid-19 abalou muito esse ramo na parte física, forçando diversos estabelecimentos a fecharem as portas por não conseguirem manter os custos e serem forçados a mudar o modelo de negócios para Delivery que acabou não dando certo para todos, sendo mais de 300 mil em 2020 segundo a Associação Brasileira de Bares e Restaurantes (Abrasel) que acabaram fechando.

Muitos restaurantes têm grande dificuldade em se manter, principalmente no início de seu funcionamento quando o negócio ainda é de pequeno ou médio porte, às vezes muito deles vem sem planejamento por empreendedores aspirantes, e segundo a Abrasel, 50% desses empreendimentos fecham as portas em questão de 2 anos, sendo no maior caso por parte das dívidas e com maior impacto ainda durante a pandemia.

Podemos ter um exemplo disso na atualidade por exemplo com o reality show “Pesadelo na Cozinha”, o qual foi ao ar em 2017, e teve 3 temporadas até o período de 2021, e diversos restaurantes foram avaliados, sendo um total de 24 estabelecimentos, em sua primeira temporada, lançada em 2017, houveram 13 avaliados, dos quais apenas 3 se mantiveram abertos no período estabelecido da pesquisa do Abrasel de 2 anos, uma pequena parte chegou até a trocar o foco dos alimentos depois da crise da pandemia e o aumento dos preços de alimentos.

Outro fator que podemos apontar é que diversos locais tem como principal foco oferecer uma experiência ao seu usuário que quer ir em um restaurante pela experiência e com isso acabar tendo uma boa refeição de pratos diversos no lugar, tendo tudo isso em mente teremos uma forma de atender estabelecimentos sem a necessidade de uma cozinha interna focado no ambiente.

1.1 Objetivo

O objetivo deste trabalho é por meio de um sistema composto por 2 aplicações, sendo uma delas um site de gerenciamento dos endereços restaurante via Web e o outro uma aplicação desenvolvida em Flutter para realizar pedidos de refeições dos

clientes a uma API, unidas de uma forma que um restaurante consiga realizar suas vendas e oferecer seus serviços focados no estabelecimento on entretenimento em si sem a necessidade de haver cozinha ou gastos com alimentos dentro do local para servir seus clientes.

Tudo será feito por meio de um sistema de Delivery o qual poderá ser usado por seus clientes ou pela própria empresa, permitindo que o restaurante ou até mesmo um buffet, se foque em seu estabelecimento e experiência que quer oferecer ao usuário, enquanto a comida será oferecida à parte por um outro sistema integrado a este com uma parte que já tenha conhecimento nessa área de entrega de alimento, como o *iFood*®, mas o nosso foco será apenas oferecer uma aplicação que forneça códigos das mesas que seus clientes vão pedir junto ao endereço do estabelecimento para ser enviado a API de Delivery e assim será feita a entrega ao local, quanto ao pagamento poderá ser deixado a par do cliente ou oferecer um saldo para cada mesa usar em qualquer refeição como bônus.

Com isso poderíamos cortar todo o custo de manutenção de uma cozinha, além de cortar a necessidade de cozinheiros e manter estoque de alimentos evitando custos de salário, desperdícios por parte da comida e redução da parte de um aluguel ou aproveitamento de espaço extra ao retirar a parte que seria para a cozinha, com isso auxiliando a prover um alívio nas contas do restaurante ou até mesmo, além é claro de permitir que seus clientes comam diversos pratos variados em seu restaurante enquanto aproveitam o entretenimento oferecido comendo o que quer que gostem, seja comida vegana, massas, doces, tudo em um lugar que irá se focar em trazer uma experiência a ele e mudar seu ambiente usando nossa aplicação como a ponte de acesso aos alimentos.

Por fim, será proporcionada uma boa experiência ao usuário permitindo ao restaurante focar em manter o cliente entretido e bem tratado com o ambiente, pois o aplicativo poderá oferecer formas de indicar as tratativas que o usuário quer ao restaurante durante a entrega do pedido.

1.2 Justificativa

A razão por trás desse trabalho vem de um problema que pode ser observado bem principalmente durante a pandemia ou com alguns documentários que demonstram a dificuldade pela qual certos restaurantes e pequenos negócios do ramo alimentício passam quando se trata de dívidas e gastos, e a grande maioria deles envolvem gastos com a cozinha e os alimentos.

Certos restaurantes tem parte do seu foco em entreter o seu cliente junto a comida, como é o caso de alguns estabelecimentos com show ao vivo, ou os buffets de festa infantil ou locações para eventos, mas parte do serviço é oferecer a comida, que gera um aumento das despesas de mantimento e manutenção, mesmo as vezes não sendo o principal que o negócio tem a oferecer ou não sendo a fonte de renda principal.

A minha ideia procura se não resolver essa parte, amenizar o impacto de preços, é pensado nesse impacto que futuras crises podem ocasionar e nos estabelecimentos com renda principal focada na ambientação além da alimentação que este projeto foi idealizado para atender a demanda de refeições e cortar os custos.

1.3 Metodologia

A primeira etapa consiste em levantar os requisitos do que vai ser necessário para as aplicações funcionarem e como seria a cultura do restaurante, pensando em software para executar a mesma será necessário uma máquina que possa executar um ambiente com um servidor web com linguagem em PHP e gerenciadores de dependências, além da capacidade para executar um banco em MySQL, um simulador para Android ou que execute aplicações em Dart usando Flutter e finalmente um editor de texto para montar o código.

Com o ambiente pronto iremos montar as aplicações, que são o site web gerencial para o restaurante, um aplicativo mobile aos seus clientes, uma aplicação irá conversar com a outra além de conversar com certas APIs durante o processo.

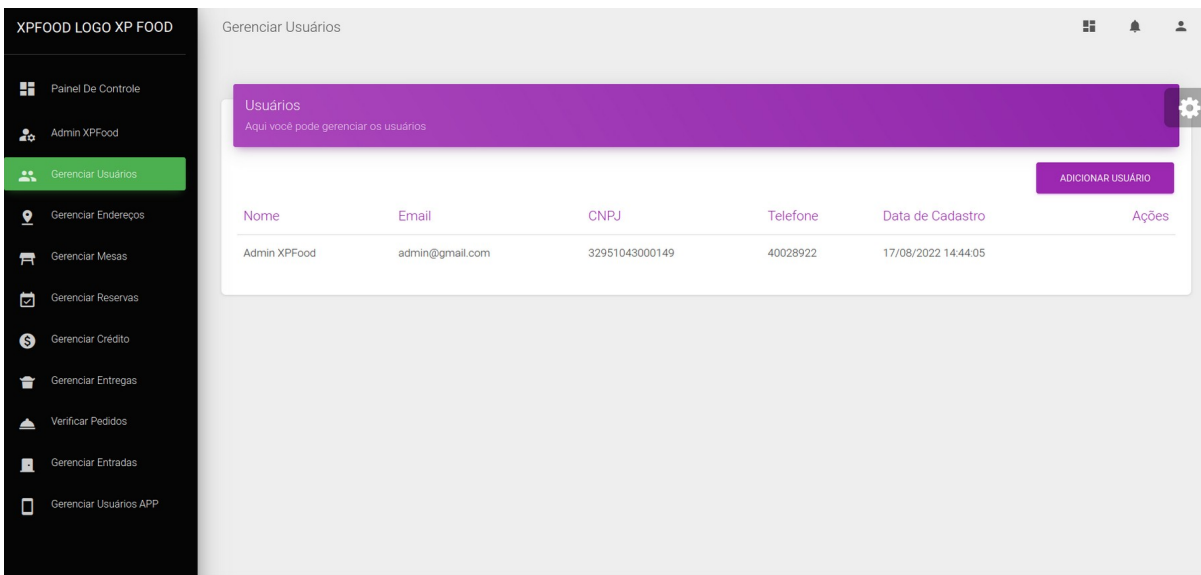
Pensando no site web gerencial, usaremos frameworks para conseguir agilizar sua produção, como o Laravel para a questão de organização e melhor

funcionamento do backend, e o Bootstrap para organização e responsividade do frontend somado a um padrão de design disponibilizado pela internet. A aplicação web em si deverá ser capaz de se comunicar com a base para realizar alterações na mesma e fazer validações.

A primeira ação que o site deve realizar a partir de sua página inicial, é permitir que o usuário possa se cadastrar ou acessar a aplicação usando suas credenciais, com a possibilidade de alterar sua senha via e-mail caso necessário, o cadastro é feito com um simples validação que vai verificar se os dados enviado por formulário são aceitos, os quais são precisos o nome, um email único, o CNPJ, um telefone e por fim a senha de acesso da empresa que quer se tornar um parceiro da nossa aplicação, e caso sejam, o backend vai salvar os registros na base e então redirecionar o usuário ao dashboard da aplicação com a permissão de acesso sendo salva em sessão para o usuário.

Em caso de acesso com credenciais já cadastradas, a aplicação irá verificar se elas estão cadastradas na nossa base, por meio do e-mail e senha, e então cria uma sessão a nós caso passe pela validação, com este acesso poderemos passar por um Middleware de verificação de autenticação de usuário para acessar as rotas da aplicação específicas para apenas quem é cadastrado, e essas rotas vão permitir o gerenciamento de seu restaurante em diversas funções.

Figura 01 – Página de Gerenciamento de Usuário da Aplicação Web



The screenshot shows the 'Gerenciar Usuários' page. On the left is a dark sidebar with the 'XPFOOD LOGO XP FOOD' at the top and a list of menu items: 'Painel De Controle', 'Admin XPFood', 'Gerenciar Usuários' (highlighted in green), 'Gerenciar Endereços', 'Gerenciar Mesas', 'Gerenciar Reservas', 'Gerenciar Crédito', 'Gerenciar Entregas', 'Verificar Pedidos', 'Gerenciar Entradas', and 'Gerenciar Usuários APP'. The main content area has a header 'Gerenciar Usuários' and a purple banner that says 'Usuários' and 'Aqui você pode gerenciar os usuários'. Below this is a table with columns: 'Nome', 'Email', 'CNPJ', 'Telefone', 'Data de Cadastro', and 'Ações'. The table contains one row for 'Admin XPFood' with email 'admin@gmail.com', CNPJ '32951043000149', phone '40028922', and registration date '17/08/2022 14:44:05'. A purple button 'ADICIONAR USUÁRIO' is located to the right of the table.

Nome	Email	CNPJ	Telefone	Data de Cadastro	Ações
Admin XPFood	admin@gmail.com	32951043000149	40028922	17/08/2022 14:44:05	

Fonte: Elaborado pelo autor.

Teremos diversas opções a ser oferecida pelo sistema web quanto o gerenciamento, sendo elas:

- Um menu com um painel com informações gerais e úteis da loja no formato de um Dashboard;
- Um menu para gerenciar seu perfil de acesso do site, podendo alterar e-mail, senha e telefone de seu perfil;
- Um menu para o gerenciamento de usuários do ambiente web apenas para administradores do próprio sistema (Figura 01), onde podemos adicionar novos usuários, editar informações de um já existente ou excluir os mesmos, e claro não será permitido que o perfil usado para o acesso da aplicação seja excluído ou alterado;
- Um menu para gerenciar os endereços de seus estabelecimentos, onde podemos cadastrar novos endereços e nosso código vai vai por meio dos dados cadastrados chamar uma API de geolocalização para retornar as coordenadas aproximadas do local cadastrado, também podemos editar e excluir os mesmos;
- Um menu para gerenciar as mesas que estarão em seus estabelecimentos, podendo adicionar uma anexando a um endereço já existente, e também editar ou excluir a mesma, além de gerar os QR Codes para leitura com o aplicativo mobile dos clientes ao fazer um pedido e também permitir fazer uma reserva de uma mesa por um usuário do aplicativo mobile através de seu CPF;
- Um menu para visualizar ou excluir as reservas das mesas após a reserva ser feita no menu de mesas;
- Um menu para gerenciar de crédito da empresa, onde é possível ver seu crédito atual e comprar crédito por meio de uma API de pagamento com diversas opções para se realizar a transação, e após isso poderá transferir o seu crédito a usuários do aplicativo, desde que tenham uma reserva de mesa ativa, para fazerem compras em seus restaurantes;
- Um menu para gerenciar as entregas que serão feitas ao seus restaurantes de acordo o que os clientes pediram pela aplicação mobile e assim confirmar a mesma durante o recebimento. Os pedidos vão ser feitas por meio de outra

API que já trabalhe com a questão de Delivery de comida, após o pagamento do cliente e leitura do Qr Code;

- Um menu para visualizar os pedidos já feitos e entregues ao seu restaurante, os pedidos aparecem aqui depois de confirmar o recebimento da entrega;
- Um menu para gerenciar as entradas de seu restaurante usando o CPF dos cliente e também gerenciar sua saída, aonde quando alguém entra no restaurante sua entrada deve ser registrada pelo CPF junto a saída;
- Um menu para gerenciar os usuários do aplicativo mobile que se cadastraram no sistema, apenas para os administradores do próprio sistema;

Além destas diversas opções em menu, ainda haverá opções para personalizar o fundo, tema de cor e uma pequena aba para notificações quando um pedido é realizado.

Todos os menus tem uma rota registrada na aplicação com um método, que são direcionadas a um método de Controller que ou vão fazer uma alteração na base ou vão redirecionar o usuário a uma página conhecida como View, cada Controller é baseado em um Model, que direciona a uma tabela em nossa base para permitir as alterações, isso compõem um modelo de aplicação MVC.

Figura 02 – Rotas do Menu de Gerenciamento de Usuários da Aplicação Web

```

42 //Usuario
43 Route::middleware([IsAdmin::class])->group(function() {
44     Route::get('/user', 'App\Http\Controllers\UserController@index')->name('user.index');
45     Route::post('/userForm', 'App\Http\Controllers\UserController@store')->name('register.form');
46     Route::put('/userForm/{id}', 'App\Http\Controllers\UserController@edit')->name('register.edit');
47     Route::delete('/userForm/{id}', 'App\Http\Controllers\UserController@delete')->name('register.delete');
48 });
49

```

Fonte: Elaborado pelo autor.

A rota de cada aplicação permite chegarmos a uma classe e função da mesma que pode nos direcionar para uma View através de uma URL junto ao nosso domínio, podendo ter um apelido para chamar a mesma dentro do código mais a frente, um exemplo seria uma tela com a lista de todos os usuários do sistema, um index (Figura 02 - Linha 44). Dentro dessas rotas podemos usar alguns Middlewares, sendo o principal para autenticação de usuário autenticado ou um que apenas permita que usuários administradores acessem uma rota (Figura 02 - Linha 43), esses Middlewares servirão como um meio de campo de acesso entre um

usuário e uma função do sistema, bloqueando ou liberando seu acesso de acordo alguns parâmetros para as funções.

Figura 03 – Controller de Usuários do Sistema

```

10 class UserController extends Controller
11 {
12     public function index(User $model)
13     {
14         return view('users.index', ['users' => $model->paginate(15)]);
15     }
16
17     public function store(UserRequest $request)
18     {
19         $data = $request->all();
20
21         try {
22             User::create([
23                 'name' => $data['name'],
24                 'email' => $data['email'],
25                 'cnpj' => $data['cnpj'],
26                 'phone' => $data['phone'],
27                 'isAdmin' => 0,
28                 'password' => Hash::make($data['password']),
29             ]);
30
31             return redirect()->back()->withErrors(['msg' => 'Usuário Cadastrado com Sucesso!']);
32         } catch (\Throwable $th) {
33             return redirect()->back()->withErrors(['error' => 'Houve um Erro ao Cadastrar o Usuário!'])->withInput();
34         }
35     }
36 }
37

```

Fonte: Elaborado pelo autor.

Passando para a classe, a grande maioria se trata de um Controller (Figura 03 - Linha 10) ou classe com funções gerais seguindo a Orientação a Objetos, dentro dessas classes poderemos manipular as informações e alterar a base usando um Model, uma classe que guarda os dados de uma tabela da base e permite chamar funções que modificam as mesmas, como o exemplo de usuários e sistema, podemos exibir uma lista deles em uma página (Figura 03 - Linha 14), adicionar a tabela de usuários (Figura 03 - Linha 17 a 36), editar ou remover os mesmos, seguindo um Model declarado antes de cada modificação ou exibição da base (Figura 03 - Linhas 14 e 22).

Por fim, sempre que vamos mexer com dados vindos do front end nas funções do backend via requisições, iremos fazer um arquivo de Request para esses dados, onde ele vai validar se os dados enviados a função de uma rota estão corretos e devidamente declarados, se não estiverem ele nem entra na função e retorna um erro, ele age como praticamente um Middleware, no caso de um Request para usuários (Figura 03 - Linha 17), iríamos validar todas as informações obrigatórias que teríamos na base, o seu tipo de variável, caso os dados precisarem ser únicos, entre outras validações de acordo a tabela.

A aplicação web é o que vai gerenciar toda nossa aplicação enquanto a aplicação mobile será formado usuário interagir com ela por meio de seu celular, a aplicação mobile se comunicará com a aplicação web por meio de requisições as rotas para pegar informações da base nossa, pegara informações das chamadas de API usando a web como intercessor, a aplicação desempenha diversas funções, dentre elas:

- Uma tela inicial onde o usuário poderá fazer login ou se cadastrar a aplicação, tudo sendo feito por chamadas de rotas da nossa aplicação web, e guardando a informação de login e token de autenticação para requisição;
- Após a autenticação, a primeira tela exibida deverá conter uma lista de refeições e alimentos que podem ser adicionados ao carrinho e então pedidos via delivery que vem por meio de uma requisição da nossa aplicação web a uma API já especializada em delivery, e ela apenas receberá nossas informações para nos fornecer os pedidos que podem ser feitos;
- Uma tela que permite editar seu perfil de usuário via requisições a nossa API da aplicação web;
- Uma tela de carrinho onde o usuário vai poder ver seus pedidos e então prosseguir ao pagamento após confirmar sua localização através de um Qr Code que deverá estar na mesa do restaurante, a verificação de localização será feita usando as coordenadas do celular e comparando com o endereço salvo da base usando a API de geolocalização
- A tela para realizar pagamentos deve aparecer depois que a localização foi confirmada, assim abrindo uma API para realizar os pagamentos por diversos meios, após a confirmação do pagamento serão enviados os dados a API de delivery para preparar e enviar o prato ao restaurante, mantendo um registro dos pedidos para o ambiente web;
- A comida será recebida em uma parte que não interfira na experiência do cliente e servida a ele por meio dos funcionários do restaurante seguindo o tratamento que eles desejam, essas informações serão analisadas e manipuladas pelo ambiente web;

O pedido será feito pela aplicação mobile, que fará contato com a web por API usando Tokens para proteger seus dados, a aplicação web fará o

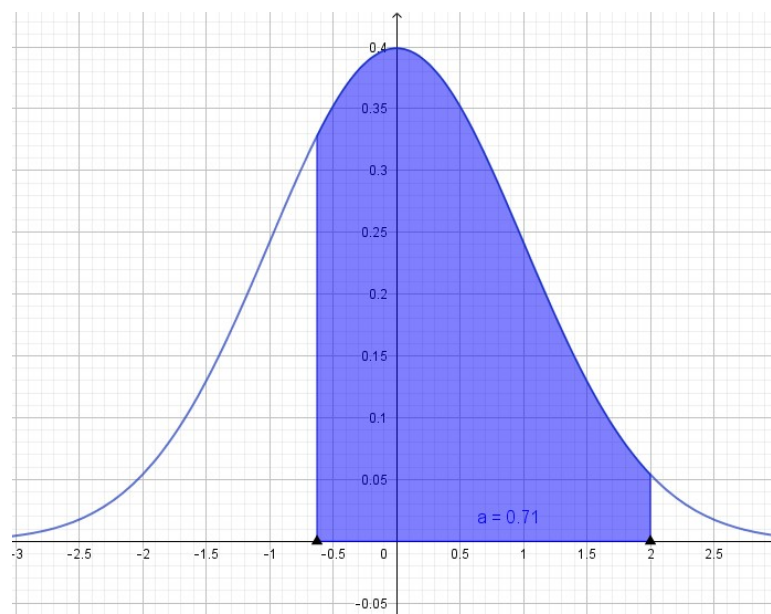
gerenciamento dos restaurante e cobrirá as funções de API para o mobile, além de se comunicar com APIs terceiras para pedidos, geolocalização e pagamentos.

Após toda a parte tecnológica, será apenas necessário um treino para os funcionários que iriam servir a mesa, mantendo a cordialidade e respeito pelos tratamentos quanto ao usuário.

2. PRIMEIRO NÍVEL

Texto...

Figura 01 – Interação de Valores na Distribuição Normal no GeoGebra



Fonte: Elaborado pelo autor, print software GeoGebra.

2.1 Segundo Nível

Texto...

2.1.1 Terceiro nível

Texto...

Quadro 01 – Tipos de Distribuição Estatística

Distribuições Contínuas	Distribuições Discretas
Normal	Poisson
Uniforme	Uniforme discreta
Triangular	*****
Exponencial	*****
Weibull	*****

Fonte: Adaptado Filho (2001, p. 173)

3. REDES CONVERGENTES

Jhfjdsjhfhdsjhfkjhsjkdhfkjhsjkhdjkfhjshjfhj jhsdhkjfhjkshdkjhfkjhjkdh
jdhsjfhkjsjhdfhjhsjkhkjfhjh jdhfjhjks hjdhkjhsjh sjhdfjhsjhdjh...

CONCLUSÃO

O trabalho permitiu...

REFERÊNCIAS BIBLIOGRÁFICAS

BARBETTA, P. A.; BORNIA, A. C. R. **Estatística para cursos de engenharia e informática**. 3. ed. São Paulo: Atlas, 2010.

FAZENDA, Ivani C. A. **Integração e Interdisciplinaridade no Ensino Brasileiro: Efetividade ou Ideologia**. 5 ed., São Paulo: Loyola, 2002. Pág. 22.

HOHENWATER, M. Manual Oficial do GeoGebra. Disponível em:<https://app.geogebra.org/help/docupt_PT.pdf>. Acesso em 17 set. 2017.