



José Victor Costa de Araújo

**Projeto e desenvolvimento de um sistema web voltado ao
controle de pagamento de faturas**

Fortaleza, Brasil

2022



José Victor Costa de Araújo

Projeto e desenvolvimento de um sistema web voltado ao controle de pagamento de faturas

Trabalho de conclusão de curso
apresentado ao Centro Universitário
Estácio do Ceará, como requisito
exigido parcial à obtenção do grau de
Tecnólogo em Análise e
Desenvolvimento de Sistemas.

Orientador: Jackanderson Menezes

Centro Universitário Estácio do Ceará
Faculdade de Tecnologia em Análise e Desenvolvimento de Sistemas

Fortaleza, Brasil

2022

Dedico este trabalho a todos que estão ao meu lado e sempre acreditaram em mim, mesmo com todas as dificuldades, me deram forças para prosseguir o meu caminho.

Resumo

Este trabalho tem como objetivo o desenvolvimento de um sistema Web, destinado a organização de pagamentos das faturas. Para evitar perdas ou atrasos nas faturas, o sistema foi criado, com intuito de melhorar a organização dos pagamentos. substituindo o antigo caderno de anotações de dívidas. O sistema será mais prático, onde, os usuários poderão estar cadastrando as faturas e seus recibos e verificando a situação de cada uma delas de forma rápida e prática.

Por fim, todos os propósitos estabelecidos inicialmente foram cumpridos, visto que o sistema foi criado, desenvolvido suas funcionalidades e enviado à plataforma web, podendo ser verificada na seguinte url abaixo.

<https://secure-brushlands-23209.herokuapp.com/>

Sumário

1 INTRODUÇÃO	8
1.1 Método	9
1.2 Organização do Trabalho	9
2 DESENVOLVIMENTO	10
2.1 Visão Geral	10
2.2 Requisitos Funcionais	11
2.3 Diagrama do sistema	12
2.4 Protótipo das telas	13
2.4.1 Login	13
2.4.2 Cadastro de Usuários	14
2.4.3 Recuperar Senha	14
2.4.4 Página Inicial	15
2.4.5 Cadastrar Editar Fatura	15
2.4.6 Detalhes do Usuário	16
2.5 Banco de Dados	17
2.6 Implantação da Aplicação ao Web Service	18
3 CONCLUSÃO	19
REFERÊNCIAS	20

1 Introdução

O avanço exponencial da tecnologia traz uma alta demanda de criações de aplicações mais simples e informatizadas, com o objetivo de tornar mais fácil com menos esforços a vida das pessoas.

As aplicações web são exemplos desse avanço, pois podemos nos conectar de forma online ao um mundo de funcionalidades e operabilidades que antes eram realizadas de forma presencial. Isso vai desde compras, pagamentos de contas, envio de mensagens, acesso as notícias, empregos, dentre outros. O que antigamente era encarado com algo improvável, graças a tecnologia, hoje se tornou algo comum e vantajoso na vida das pessoas.

Com o avanço da globalização, o dia a dia se torna mais corrido na vida das pessoas. Com isso, se torna muito comum o esquecimento em relação à determinadas tarefas. Visto isto, foi desenvolvido um protótipo de uma aplicação web, com o abstrair as ideias para a criação da aplicação. O sistema foi criado utilizando as linguagens PHP, CSS e Javascript e usando o framework web Laravel.

1.1 Método

Com base em meus conhecimentos em implementações de sistemas anteriores, comecei à princípio, verificar às necessidades que as pessoas ainda enfrentam para realizadas determinadas tarefas. A partir de então, verifiquei que alguns de meus familiares organizavam as contas de luz, água, telefone etc. em um caderninho de anotações e o papel da fatura com seu comprovante eram guardados em suas gavetas.

Assim, verifiquei um possível risco que poderia acontecer se o papel de uma determinada fatura ou comprovante viesse a perder, resultando na precisão de gerar uma segunda via, como também, a dúvida que a pessoa teria se foi pago ou não uma determinada fatura.

A partir disso, comecei a entrevistá-los com o objetivo de recolher as principais dificuldades enfrentadas, para está criando sistema que possa auxiliá-los.

Então comecei a abstrair as informações repassadas para a criação do novo sistema. Nesta etapa comecei a desenhar os processos a serem adotados e as ferramentas a serem utilizadas no processo de desenvolvimento da aplicação. Quanto a plataforma, optou-se pelo desenvolvimento de um sistema WEB.

Após ser realizado a implementação do sistema na área de desenvolvimento, foi realizado o deploy da aplicação para o servidor heroku.

1.2 Organização do Trabalho

Este trabalho está organizado nos seguintes capítulos:

Desenvolvimento

No capítulo 2, são apresentadas as tecnologias utilizadas no desenvolvimento do sistema, os requisitos funcionais e não-funcionais, descrevendo detalhadamente cada funcionalidade implementada, os diagramas utilizados no desenvolvimento do sistema, os protótipos, bem como a forma de comunicação com o banco de dados.

Conclusão.

No capítulo 3, são expostas as considerações finais sobre o sistema e são propostas algumas mudanças que podem ser realizadas em trabalhos futuro

2 Desenvolvimento

Neste capítulo, as etapas do desenvolvimento deste trabalho são detalhadas. Inicialmente, é mostrada uma visão geral sobre como o sistema foi desenvolvido, detalhando as técnicas e recursos utilizados. Depois, são apresentados detalhadamente, todas as funcionalidades e restrições do sistema. Posteriormente, o diagrama e os protótipos das telas e o banco de dados utilizado no desenvolvimento e na produção. Na última sessão, explica-se sobre o processo de implantação da aplicação no servidor.

2.1 Visão Geral

Para utilizar às funcionalidades finais do sistema, foram utilizados HTML5, PHP e Javascript, com framework Laravel. Na estilização, foi utilizado CSS com a biblioteca Bootstrap e para o banco de dados foi utilizado o mysql no ambiente de desenvolvimento e postgres no ambiente de produção.

HTML5: Hypertext Markup Language 5 é uma linguagem de marcação para a World Wide Web e é uma tecnologia chave da Internet, originalmente proposto por Opera Software. É a quinta versão da linguagem HTML. Esta nova versão traz consigo importantes mudanças quanto ao papel do HTML no mundo da Web, através de novas funcionalidades como semântica e acessibilidade.

PHP: Personal Home Page é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web.

Javascript: Javascript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web.

Laravel: é um framework PHP livre e open-source criado por Taylor B. Otwell para o desenvolvimento de sistemas web que utilizam o padrão MVC (model, view, controller). Algumas características proeminentes do Laravel são sua sintaxe simples e concisa, um sistema modular com gerenciador de dependências dedicado, várias formas de acesso a banco de dados relacionais e vários utilitários indispensáveis no auxílio ao desenvolvimento e manutenção de sistemas.

MYSQL: É um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface. É atualmente um dos sistemas de

gerenciamento de bancos de dados mais populares da Oracle Corporation, com mais de 10 milhões de instalações pelo mundo

Postgres: é um sistema gerenciador de banco de dados objeto relacional (SGBD), desenvolvido como projeto de código aberto.

2.2 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades que cada usuário deve ser capaz ao acessar o sistema. Esses requisitos estão relacionados às atividades que o sistema realiza.

Cadastro de Usuários: O sistema deve disponibilizar uma área de cadastro, onde o novo usuário poderá realizar o cadastro com suas respectivas credenciais. Cada usuário, terá as suas cadastradas. Nenhum usuário poderá ver uma fatura cadastrada ou outra informação de outro usuário

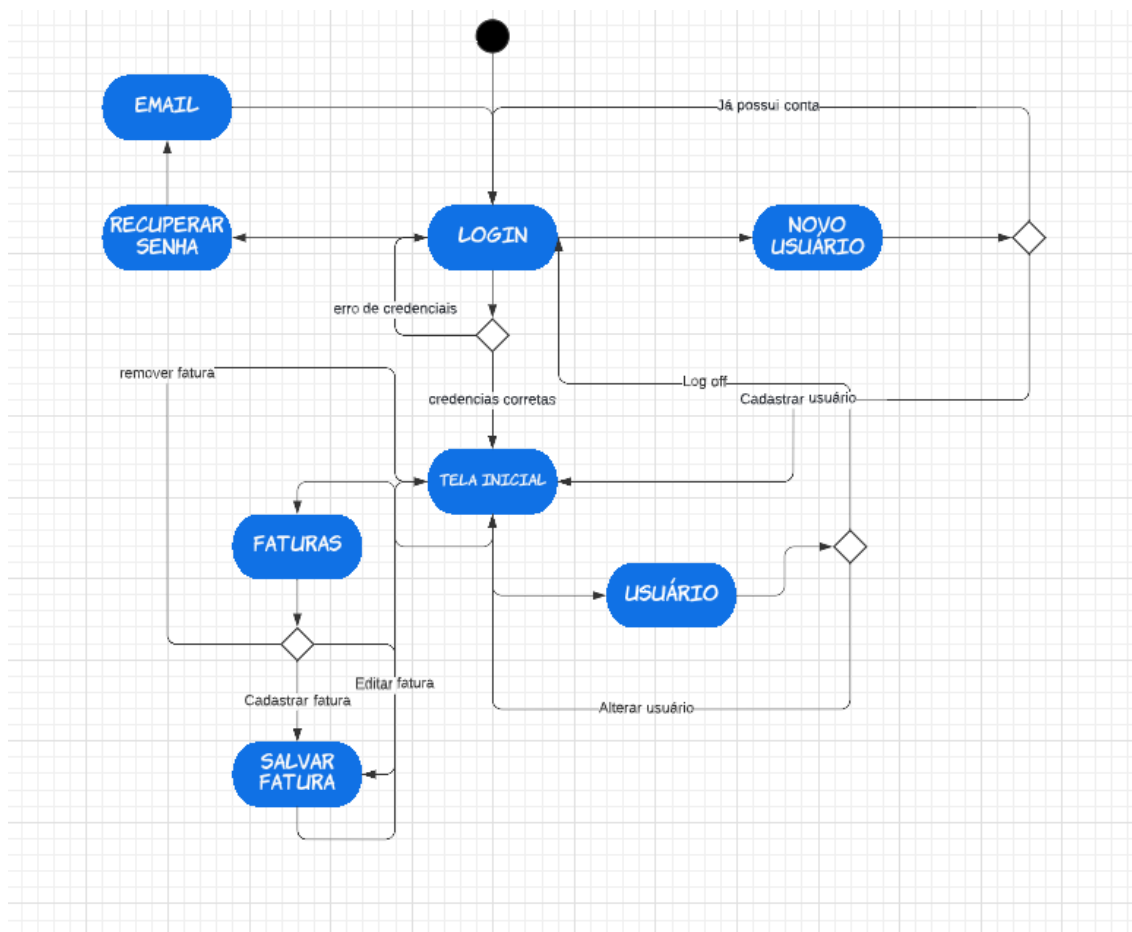
Cadastro de Faturas: O sistema deve ser capaz de salvar as faturas registradas pelo usuário no sistema. As informações salvas deverão ser as mesmas informações que contém na fatura, sendo inteira responsabilidade do usuário o preenchimento das respectivas informações.

Edição de Faturas Cadastradas: O sistema deverá disponibilizar de uma interface em que o usuário possa estar editando as faturas cadastradas, alterando as suas de acordo com a sua necessidade

Remover Fatura Cadastradas: O sistema deverá oferecer uma opção de remoção, em que, o usuário poderá remover as faturas cadastradas. A remoção das informadas registradas no sistema será de responsabilidade de usuário.

Saída do Usuário do Sistema (log off): O sistema deverá oferecer uma opção de saída, em que o usuário possa estar encerrando a sessão.

2.3 Diagrama do Sistema



Na imagem é mostrada o fluxo de como a aplicação deverá ser comportada. Após o início, o fluxo irá para o login, caso as credenciais de acesso estejam corretas, o fluxo irá continuar para a tela inicial, caso contrário o fluxo voltará para tela de login. O usuário então terá três opções para logar. Realizar uma nova tentativa de acesso, recuperar a senha, cadastrar um novo usuário.

Ao recuperar a senha, o fluxo irá ser direcionado para o e-mail, onde será realizado o procedimento de redefinição de senha. Depois disso, o fluxo irá retornar a tela de login.

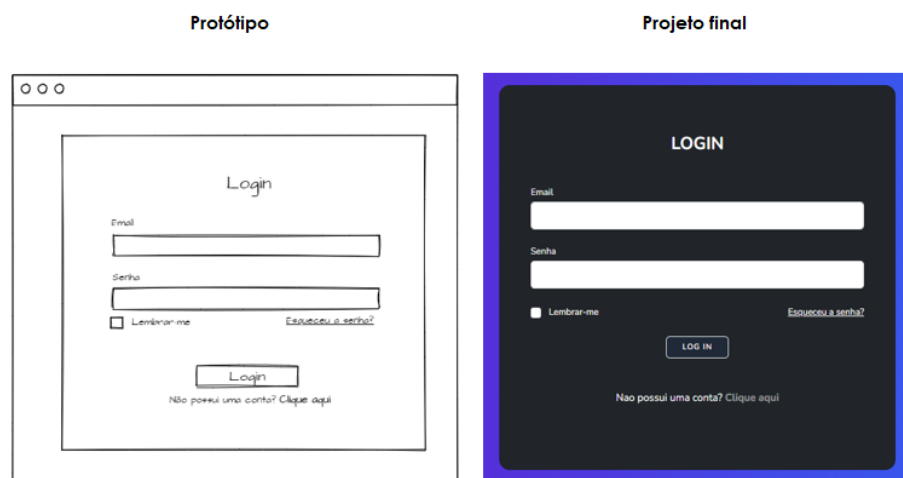
Na opção de novo usuário, será criado um usuário de acesso. Se a criação for realizada com sucesso, o fluxo irá para tela inicial, caso contrário o usuário poderá realizar uma nova tentativa de cadastro. Se o usuário já possui uma conta, o fluxo retornará a tela de login.

Na tela inicial, o usuário a opção de faturas e de usuário. Na opção de faturas, o usuário poderá verificar todas as faturas cadastradas, realizar o cadastro, alterar e remover uma fatura. Já na opção usuário, o mesmo poderá estar editando as credenciais do usuário e realizar o encerramento da sessão.

2.4 Protótipo das Telas

Quando falamos em protótipo, estamos falando em tentar passar, na maioria das vezes para o cliente, como o software ficará visualmente, permitindo que o cliente possa entender e opinar sobre os aspectos visuais e funcionais do sistema. A prototipagem pode ter vários níveis de fidelidade. desde o mais baixo (wireframe) até o mais com muitos níveis de detalhes dando impressão do software funcionando. Neste trabalho serão mostrados os protótipos wireframe

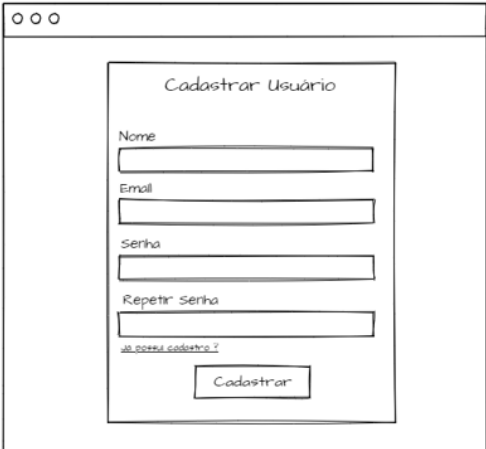
2.4.1 Login



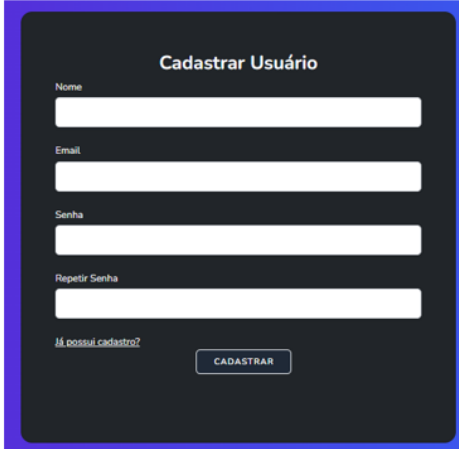
Na tela de login, o usuário deverá inserir suas credenciais de acesso. Caso o usuário inserir algum campo de forma inválida, o sistema irá trazer um retorno informando o erro. Na opção “esqueceu a senha?”, o usuário será direcionado a tela de recuperar, onde seguirá os procedimentos de recuperação. No link “Clique aqui”, o usuário poderá criar um de perfil de acesso, enquanto na opção “lembra-me”, o usuário poderá realizar o acesso de forma direta, sem precisar realizar o login.

2.4.2 Cadastro de Usuários

Protótipo



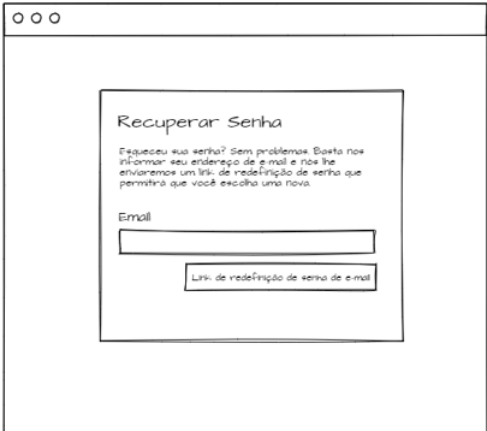
Projeto final



Ao realizar o cadastro, o usuário poderá inserir seus parâmetros de acesso. Ao clicar no botão cadastrar, o usuário será redirecionado para a tela principal da aplicação. Clicando no botão “já possui cadastro”, o usuário retornará para a tela de login.

2.4.3 Recuperar Senha

Protótipo

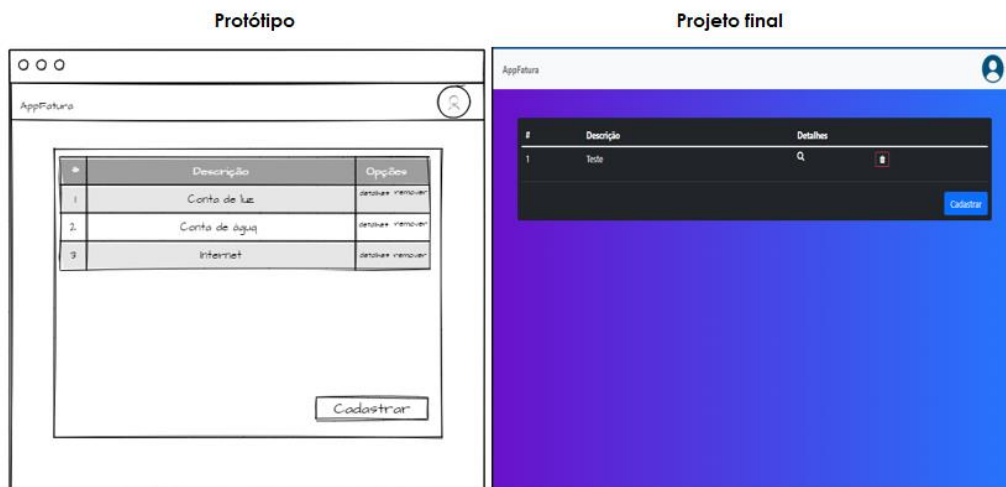


Projeto final



Na tela de recuperação de senha o usuário deve inserir o e-mail em que foi cadastrado anteriormente. Ao clicar no botão “link de redefinição de senha de e-mail”, será enviado um link para o e-mail do usuário informando os passos a serem seguidos para realizar a redefinição.

2.4.4 Página Inicial



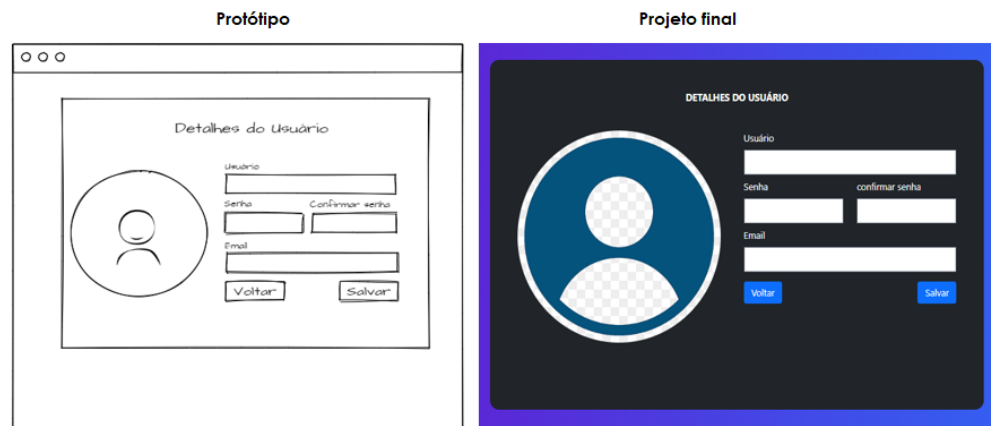
Na tela principal, serão informadas todas as faturas cadastradas pelo usuário. Na coluna “Opções”, o usuário poderá ver os detalhes de cada fatura e recibo cadastrado. Na opção “Remover”, o irá remover a fatura cadastrada. No canto superior direito da tela o usuário poderá ter as opções de “detalhes do usuário” e de realizar o log-off.

2.4.5 Cadastrar | Editar Fatura



Ao realizar o cadastro de uma nova fatura, o usuário deverá informar obrigatoriamente todos os campos com exceção dos recibos e data do pagamento. É extrema importância que, as informações inseridas sejam as mesmas informações que constam na fatura e no recibo. O cadastro de todos é de responsabilidade do usuário. Ao clicar no botão “salvar”, o usuário irá salvar o registro, ao clicar no botão “voltar”, ele irá redirecionar o usuário para a tela principal.

2.4.6 Detalhes do Usuário



Nesta tela, o usuário poderá estar editando seu credenciais de usuário. Ao salvar, o usuário será redirecionado para a tela principal.

2.5 Banco de Dados

Ao ser realizada a ideia da aplicação, é necessário está implementando um banco de dados para armazenar as informações de usuários e das faturas. O banco do sistema, no lado do desenvolvimento foi o mysql, enquanto no lado da produção, postgres. O motivo de usar bancos distintos em ambientes diferentes, é que no ambiente servidor heroku, o postgres no momento, é único banco que pode ser utilizado de forma gratuita até 10 mil registros no banco.

Estrutura do banco: Com a ajuda da ORM do Laravel: eloquent foi realizado a criação das tabelas. O eloquent realiza as migrações de todas as tabelas e seus relacionamentos para o banco. O banco está estruturado em duas tabelas, usuários e faturas.

Usuários:

```
CREATE TABLE `usuarios` (
  `idUsuarios` int(11) NOT NULL AUTO_INCREMENT,
  `login` varchar(100) NOT NULL,
  `senha` varchar(32) NOT NULL,
  `nome` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  PRIMARY KEY (`idUsuarios`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=latin1 PACK_KEYS=0;
```

Na tabela de usuários, todos os campos serão obrigatórios para o salvamento de um registro. A chave primária será “idUsuarios”.

Faturas:

```
CREATE TABLE `faturas` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `descricao` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `valorFatura` double(8,2) NOT NULL,
  `imgFatura` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `vencimento` date NOT NULL,
  `imgRecibo` text COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `dataPagamento` date DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `user_id` bigint(20) unsigned NOT NULL,
  PRIMARY KEY (`id`),
  KEY `faturas_user_id_foreign` (`user_id`),
  CONSTRAINT `faturas_user_id_foreign` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Na tabela de faturas, foi adicionado uma foreign key “user_id”, que a realizará a conversação com “idUsuarios”, da tabela usuários. Assim, cada fatura criada terá um usuário associado.

2.6 Implantação da Aplicação ao Web Service

Após realizar a criação da aplicação, é feita a implantação (deploy) ao web service. Esse processo, consiste em trazer a aplicação do lado do desenvolvimento para produção. Neste projeto, o web service utilizado será o heroku, que é um servidor bem prático e gratuito, desde que, a aplicação não ultrapasse o valor de 10 mil registros no banco. Além disso, o servidor já realiza a criação de uma máquina virtual e do domínio de forma gratuita. A partir disso é necessário visitar a página do servidor e verificar a documentação que contém o passo a passo da implantação. Feito isso, é realizado o processo de deploy da aplicação ao servidor.

3 Conclusão

Este trabalho foi desenvolvido com a finalidade de mostrar as funcionalidades que a aplicação realiza, com o objetivo de ajudar as pessoas no quesito de organizar a vida financeira. O sistema foi de forma que os usuários não terão dificuldades em utilizar nenhuma das funcionalidades. As ideias que foram pensadas, e não incluídas no sistema atual, mas que podem ser implementados futuramente são:

- Implementar uma integração com um API de envio de mensagens, com o objetivo de oferecer uma opção em que o usuário possa enviar os seus boletos para o e-mail.
- Geração de uma tela que mostre o valor total de todas as faturas cadastradas
- Criar uma validação em que o valor do vencimento e do recibo não possam ser maiores que as data atual do cadastro, para evitar erros de inserção de dados.
- Correções de bugs no lado da produção.

Referências

- HTML5: <https://pt.wikipedia.org/wiki/HTML5>. Citado na página 10.
- PHP: <https://pt.wikipedia.org/wiki/PHP>. Citado na página 10.
- Javascript: <https://pt.wikipedia.org/wiki/JavaScript>. Citado na página 10.
- Laravel: <https://pt.wikipedia.org/wiki/Laravel>. Citado na página 10.
- MYSQL: <https://pt.wikipedia.org/wiki/MySQL>. Citado na página 10 e 11.
- Postgres: <https://pt.wikipedia.org/wiki/PostgreSQL>. Citado na página 11.