



2022

**PROJETO DE CONSULTORIA  
EMPRESARIAL**



UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROJETO DE CONSULTORIA EMPRESARIAL**  
GESTÃO FINANCEIRA  
**FISCON**

SÃO JOÃO DA BOA VISTA, SP  
NOVEMBRO 2022

UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**  
  
**PROJETO DE CONSULTORIA EMPRESARIAL**  
  
**GESTÃO FINANCEIRA**  
  
**FISCON**

MÓDULO MODELAGEM E DESENVOLVIMENTO DE SISTEMAS

Gestão Financeira – Profa. Renata Alencar Marcondes

Programação Orientada a Objeto – Prof. Mauro Glória

Lógica de Programação – Prof. Sidney Gitcoff Telles

Modelagem de Dados – Prof. Max Streicher Vallim

Projeto de Modelagem e Desenvolvimento de Sistemas – Profa Mariângela

Martimbianco Santos

Estudantes:

Davi Pinto Santana Rodrigues, 22000584

Fernando Pagliarini Furlanetto, 22000293

João Otávio Rehder Mendes, 22000508

Kauã da Silva Lima Amorim, 22000589

Kayque Silva Zanelli, 22001114

Lucas de Oliveira Barreiro, 22000100

Victor Castilho, 22001035

Vinicius Miguel Quintino, 22001127

Monitor:

Caio Grilo da Cunha, RA 22000246

Fábio Luiz Barbosa Filho, RA 22000941

SÃO JOÃO DA BOA VISTA, SP NOVEMBRO 2022

# SUMÁRIO

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	5
3. PROJETO DE CONSULTORIA EMPRESARIAL	6
3.1 GESTÃO FINANCEIRA	6
3.1.1 FLUXO DE CAIXA	6
3.2 PROGRAMAÇÃO ORIENTADA A OBJETO	7
3.2.1 CLASSES E OBJETOS	7
3.2.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.	8
3.2.3 MÉTODOS ESTÁTICOS, PÚBLICOS, PRIVADOS E PROTECTED	9
3.3 LÓGICA DE PROGRAMAÇÃO	10
3.3.1 LÓGICA DE PROGRAMAÇÃO O.O.	11
3.3.2 PROTOTIPAÇÃO	11
3.3.3 TEMPLATE	12
3.4 MODELAGEM DE DADOS	13
3.4.1 MODELO CONCEITUAL	13
3.4.2 MODELO LÓGICO	14
3.4.3 SQL	14
4. CONCLUSÃO	18
REFERÊNCIAS	20
ANEXOS	22

# 1. INTRODUÇÃO

A proposta deste projeto é criar um sistema em nível de protótipo que visa fidelizar os funcionários da empresa na qual foi escolhida. O protótipo é responsável por evidenciar as funcionalidades do mesmo.

A importância de um sistema financeiro é simplificar o serviço feito em diversos processos, dos quais devem ser realizados diariamente pelos funcionários, além de tornar possível criar projeções que a empresa deseja fazer.

Este protótipo é feito com base em um banco de dados que recebe as entradas e saídas e o controle dos saldos desta empresa, com estes dados recebidos se torna possível realizar projeções de ao menos 1 mês até um longo período de tempo, para criação deste protótipo será utilizado outros módulos como Modelagem de Dados, Gestão Financeira, Lógica da Programação, e Programação Orientada a Objeto.

## **2. DESCRIÇÃO DA EMPRESA**

A empresa alvo do projeto realizado, tem como FISCON - Empresa Contábil Sociedade Simples LTDA, matriculado sob o CNPJ 48.619.449/0001-69, localizado sob o município de São João da Boa Vista – SP, rua Joaquim Valim, 98 - Jardim Satélite (13.870-399).

Fundada em 21 de janeiro de 1977, a Fiscon traz como sua principal atividade, o ramo de contabilidade no segmento de controle financeiro corporativo, onde o mesmo visa proporcionar com qualidade e excelência, serviços na área contábil, fiscal e de recursos humanos.

### 3. PROJETO DE CONSULTORIA EMPRESARIAL

Durante este módulo foram apresentadas plataformas e métodos que foram utilizados para construção do projeto, além do conhecimento necessário para lidar com o mesmo, seja o conhecimento financeiro, ou o conhecimento técnico.

Para se construir o projeto primeiro você deve ter conhecimento do tema e como manipulá-lo para isso foi necessário a disciplina de Gestão Financeira, ministrada pela professora Renata E. de Alencar Marcondes, onde foi ensinado como funciona um fluxo de caixa e como manipular ou ter controle do mesmo, algo que é de extrema importância, visto que além do conhecimento técnico necessário para criação do projeto, deve-se ter conhecimento sobre a necessidade do cliente, pois isso passa credibilidade, confiança e facilita no desenvolvimento do projeto.

Durante o período de criação do projeto as disciplinas de Programação Orientada a Objeto e Lógica da Programação ministradas respectivamente pelos professores Mauro Glória Júnior e Sidney Gitcoff Telles, e se fizeram presente durante a criação tanto do *Front End* quanto *Back End*, auxiliando na manipulação de variáveis, como guardar valores, trazer valores, e manipular os mesmos, e também na estética do software, onde foram utilizados linguagens e extensões para criar uma interface agradável e principalmente simples e funcional.

Quanto a disciplina de modelagem de dados ministrada pelo professor Max Streicher Vallim, é de extrema importância para o projeto, pois ela passa o conhecimento necessário para se construir um banco de dados que possa armazenar todos esses dados de uma forma funcional e simples, evitando problemas no software, e também de forma segura, por se tratar de dados pessoais e financeiros do usuário é de extrema importância a segurança desse banco de dados.

#### 3.1 GESTÃO FINANCEIRA

Para criação deste software é necessário certo conhecimento financeiro, por se tratar do tema central do projeto, a partir das aulas, é ensinado como se realizar futuras projeções de lucros, porcentagem de juros, juros simples e composto, montante, dados estes que são de

extrema importância quando se vai criar um software de finanças, pois o desenvolvedor de um software deve possuir conhecimento também da área em que o sistema irá operar, para que possa atender de melhor forma as necessidades do cliente.

### **3.1.1 FLUXO DE CAIXA**

Desta forma foi apresentado pela professora Renata Marcondes dizendo que o fluxo de caixa é uma ferramenta de gestão que permite ao empresário, ou ao analista, planejar sua situação financeira no intuito de agir antecipadamente a possíveis desequilíbrios entre os recebimentos e os pagamentos.

Para Zdanowicz (1998, p.33) o fluxo de caixa é o instrumento que permite demonstrar as operações financeiras que serão realizadas pela empresas, facilitando a análise e a decisão de comprometer os recursos financeiros, de relacionar o uso das linhas de créditos menos onerosas, de determinar o quanto a organização dispõe de capital próprio, bem como utilizar as disponibilidades da melhor forma possível.

Porém deve-se ter em mente que fluxo de caixa e controle de caixa são diferentes formas de administração de recursos dentro de uma empresa, sendo o controle de caixa responsável pela saúde do atual presente da empresa tendo em vista manter-se no mercado, e o fluxo de caixa uma organização visando o futuro da mesma assim como dito pela professora Renata Marcondes..

Tanto no fluxo de caixa quanto na projeção de fluxo de caixa, o detalhamento de entradas e saídas de capital permite uma visão ampla e clara sobre vários aspectos do funcionamento da empresa. Uma das vantagens desse demonstrativo é facilitar a administração financeira das empresas. Com ele se pode saber se os problemas financeiros têm origem no Operacional, nos investimentos, nos financiamentos, ou ainda numa combinação dos três grupos. Filho (1999. p. 32).

## **3.2 PROGRAMAÇÃO ORIENTADA A OBJETO**

A programação orientada a objetos é um modelo de programação onde diversas classes possuem características que definem um objeto na vida real, além disso, ela é bem requisitada no contexto das aplicações mais atuais, que o mercado demanda. Entre os motivos dessa preferência, estão a possibilidade de reutilização de código. Cada classe determina o



comportamento do objeto definido por métodos e seus estados possíveis definidos por atributos. Foi utilizada no projeto de forma que conecta o site escrito em html e php, com o banco de dados, armazenando e definindo as informações adquiridas no site através de formulários que serão preenchidos pelos clientes, obtendo todos os dados necessários para o banco de dados realizar a interpretação e fazer cálculos necessários para o sistema, tendo como objetivo principal o cálculo do fluxo de caixa do cliente e da sua empresa, além de armazenar e gerar informações como o estoque e a possível comissão dos vendedores.

### 3.2.1 CLASSES E OBJETOS

Segundo o professor Albani(2022), técnico em processamento de dados pela escola técnica federal do Espírito Santo, em seu e-book há a seguinte afirmação: “Uma classe é como um tipo de objeto que pode ser definido pelo programador para descrever uma entidade real ou abstrata, contendo o que chamamos de atributos e métodos”.

Nesse sentido, foram adicionadas classes ao nosso projeto com o intuito de, ao instanciar um novo elemento à classe, ou seja, um novo objeto, nele irá ser atribuído instantaneamente os atributos e os métodos da classe na qual foi atribuída, facilitando e agilizando assim o nosso trabalho. Um exemplo disso é, imagine uma classe chamada Pessoa, nela estará presente os atributos nome, CPF, RG e altura, e os métodos falar, pular, cantar e correr. Ao instanciar um novo elemento dentro dessa classe, ou seja, um objeto, na qual apresentará o nome de uma pessoa, como Carlos, será atribuída automaticamente a esse objeto os atributos da classe. Sendo assim, o objeto Carlos passará a ter os atributos nome, CPF, RG e altura, e os métodos falar, pular, cantar e correr.

**Figura 1 - Exemplo de Classe**

```
class Caixa {  
  
}
```

Fonte: Autor do Documento

**Figura 2 - Exemplo de Objeto**

```
$cliente = new Cliente();  
  
$cliente->setCnpj($cnpj);  
$cliente->setRazaosocial($razaosocial);  
$cliente->setEndereco($endereco);  
$cliente->setNomeEmpresa($nome_empresa);  
$cliente->setEmail($email);  
$cliente->setSenha($senha);
```

Fonte: Autor do Documento

### **3.2.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.**

De acordo com Gamma Erich, autor do livro Design Patterns: Elements of Reusable Object-Oriented Software, os atributos são os elementos que definem a estrutura de uma classe. Os atributos também são conhecidos como adjetivos de classe, e podem ser divididos em dois tipos básicos: atributos de instância e de classe. Os valores dos atributos de instância determinam o estado de cada objeto, sendo usado como um armazenador de informações preenchidas no formulário do site, e através do método “POST” encaminhará esses dados para o valor do atributo.

As classes na orientação a objetos funcionam como um molde para os objetos. Os objetos são criados a partir de uma classe e muitos deles podem ser feitos da mesma classe. De acordo com o livro Design Patterns: Elements of Reusable Object-Oriented Software, herança assim como na vida real somos um acúmulo de características de nossos antepassados, na orientação a objetos isto também é possível. Na verdade, isto é até uma boa prática. O conceito de herdar é literalmente ser uma cópia de outra classe com algumas características adicionais.

Já polimorfismo significa muitas formas, então temos a possibilidade de um objeto assumir diversas formas diferentes na orientação a objetos.

**Figura 3 - Exemplo de Atributos**

```
class Cliente{  
    private $cnpj;  
    private $razao_social;  
    private $endereco;  
    private $nome_empresa;  
    private $email;  
    private $senha;
```

Fonte: Autor do Documento

### **3.2.3 MÉTODOS ESTÁTICOS, PÚBLICOS, PRIVADOS E PROTECTED**

Em relação aos métodos, há diferentes tipos, sendo eles: estáticos, públicos, privados e protegidos. Os métodos estáticos são métodos de uma classe declarados pela palavra-chave “static” que podem ser utilizados sem que haja a necessidade de instanciar um objeto a essa classe. Já os métodos públicos são os métodos declarados pela palavra-chave “public” que podem ser acessados por qualquer outra classe. Além disso, têm-se também a definição de métodos privados, que seriam os métodos declarados pela palavra-chave “private” e que somente podem ser acessados por métodos da própria classe, e por fim, os métodos protegidos, que seriam os métodos declarados pela palavra-chave “protected” e que são responsáveis por tornar o membro acessível às classes do mesmo pacote ou através de herança, seus membros herdados não são acessíveis a outras classes fora do pacote em que foram declarados.

Em nosso projeto, foram utilizados e implementados para a realização do encapsulamento apenas os métodos públicos: setters e getters. Pois com esse tipo de método, se torna possível permitir acesso a qualquer código externo à classe, facilitando e agilizando assim o planejamento, a execução e a criação do nosso script.

**Figura 4 - Exemplo de Métodos**

```
class Caixa {
    private $nome;
    private $banco;
    private $agencia;
    private $conta;

    public function setBanco($banco) {
        $this->banco = $banco;
    }

    public function __getBanco(){
        return $this->banco;
    }

    public function setName($nome) {
        $this->nome = $nome;
    }

    public function __getNome(){
        return $this->nome;
    }

    public function setAgencia($agencia) {
        $this->agencia = $agencia;
    }

    public function __getAgencia(){
        return $this->agencia;
    }

    public function setConta($conta) {
        $this->conta = $conta;
    }

    public function __getConta(){
        return $this->conta;
    }
}
```

Fonte: Autor do Documento

### 3.3 LÓGICA DE PROGRAMAÇÃO

A metodologia de lógica da programação é uma parte muito importante ao longo do projeto como um todo ao decorrer do desenvolvimento. A lógica de programação é criada

através de um conjunto de princípios. Essas diretrizes ajudam os desenvolvedores a criar códigos que possam ser compreendidos por máquinas ou programas seguindo um padrão específico, após o código escrito, pode-se converter dados em informações que as máquinas podem compreender.

A lógica de programação ao decorrer desse projeto mostrou-se altamente necessária para o desenvolvimento do programa, onde o mesmo passou por diferentes estágios de idealização até, desde o conceito de algoritmo proposto até a definição de todos os devidos passos da lógica do algoritmo para uma linguagem de programação.

### **3.3.1 LÓGICA DE PROGRAMAÇÃO O.O.**

Essas estruturas são utilizadas para processar dados de entrada no programa, alterando dados até que a saída desejada seja gerada, gerando assim fluxograma que pode ser compreendido como representação esquemática desenvolvidas por gráficos, que no qual mostram o processo que o sistema irá percorrer, até que se chegue no resultado final da exibição a ser gerada.

### **3.3.2 PROTOTIPAÇÃO**

Segundo FILHO (2001), a prototipagem, nos processos de desenvolvimento de software, pode ser entendida como uma técnica aplicável a atividades do fluxo de requisitos. A prototipação é a produção de uma representação gráfica que se aproxima das funcionalidades idealizadas do projeto, o protótipo pode ser realizado com a função de clarear as funcionalidades para execução dos próximos passos, além de como ficará a interação entre homem e máquina. A prototipação possui diversos benefícios onde se pode realizar testes de usabilidade e funcionalidade, evitando assim a eventual possibilidade de lançar um produto com falhas.

No projeto interdisciplinar, a prototipação foi feita para testes de usabilidade, onde se foi criado um esqueleto de projeto que nele continha as funcionalidades presentes no sistema a ser desenvolvido, onde se foram realizados testes de inserção e retorno de informações para o banco de dados.

**Figura 5 - Prototipação de Sistema**



Fonte: Autor do Documento

**Figura 6 - Prototipação de Sistema**

A imagem mostra um formulário de cadastro de empresa com os seguintes campos: CNPJ, Razão Social, Endereço, Nome da Empresa, E-mail e Senha. Abaixo dos campos há dois botões: "Enviar" e "Limpar".

Fonte: Autor do Documento

A tática de realizar a prototipação nas funcionalidades fez com que o grupo conseguisse idealizar soluções para problemas recorrentes das falhas apresentadas nos testes através do protótipo. Para Budde e Zullighoven (1990) seu principal objetivo é auxiliar a especificação e validação de requisitos relevantes ou problemas de implementação, permitindo elaborar e testar interfaces com os usuários de maneira visual e interativa.

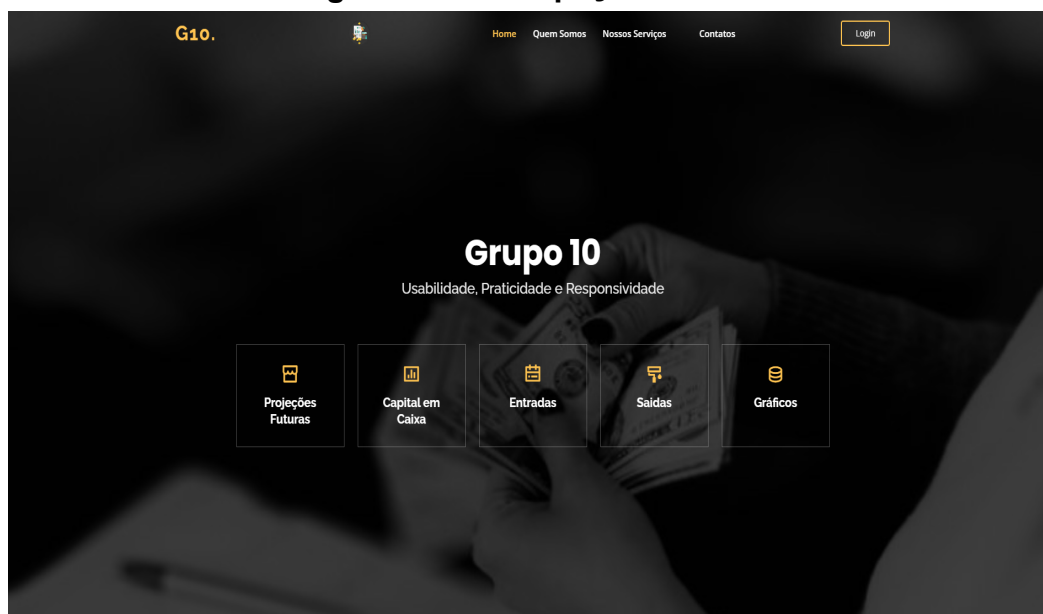
### 3.3.3 TEMPLATE

Toda vez que é acessado um site, além do conteúdo, o que mais chama a atenção é o template que pode ser definido como sendo um modelo de Layout com padrões de elementos de design pré determinado pelo criador, o uso de templates no decorrer do desenvolvimento de um projeto acarreta uma grande otimização de tempo para entrega do produto final, já que o uso de tal ferramenta traz consigo elementos de base que ajudam na execução e idealização

do projeto como um todo. Trazendo consigo também o recurso da responsividade que é um dos elementos contidos junto a ferramenta quando se faz a utilização da mesma, o sistema modula a proporção da tela fazendo com que o sistema na qual ela se faz presente se adapte às mais diversas telas, a utilização do recurso além de aplicar esses elementos de design faz com que o sistema tenha maior relevância ao ser hospedado em um navegador.

Em nosso projeto, a utilização de templates de front-end foi essencial para o desenvolvimento do sistema, já que otimizamos tempo na execução do front-end, o tema de template que foi escolhido tem como características a presença de uma única página para demonstração da maior parte das informações e funcionalidades no mesmo ambiente, esse método de sistema se é classificado como one-page.

**Figura 7 - Prototipação de Sistema**



Fonte: Autor do Documento

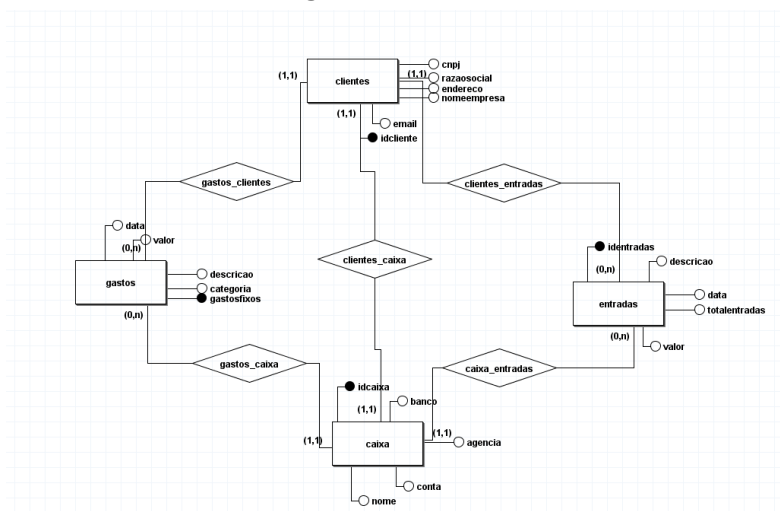
Na figura apresentada acima, pode se notar a presença do método one-page aplicado de maneira prática, onde todos os recursos do sistema se fazem presente em um único local.

### **3.4 MODELAGEM DE DADOS**

Nessa etapa o aluno deve desenvolver o projeto do banco de dados do sistema, iniciando no modelo lógico e terminando no modelo físico.

### 3.4.1 MODELO CONCEITUAL

**Figura 8 - Modelo Conceitual do Sistema**



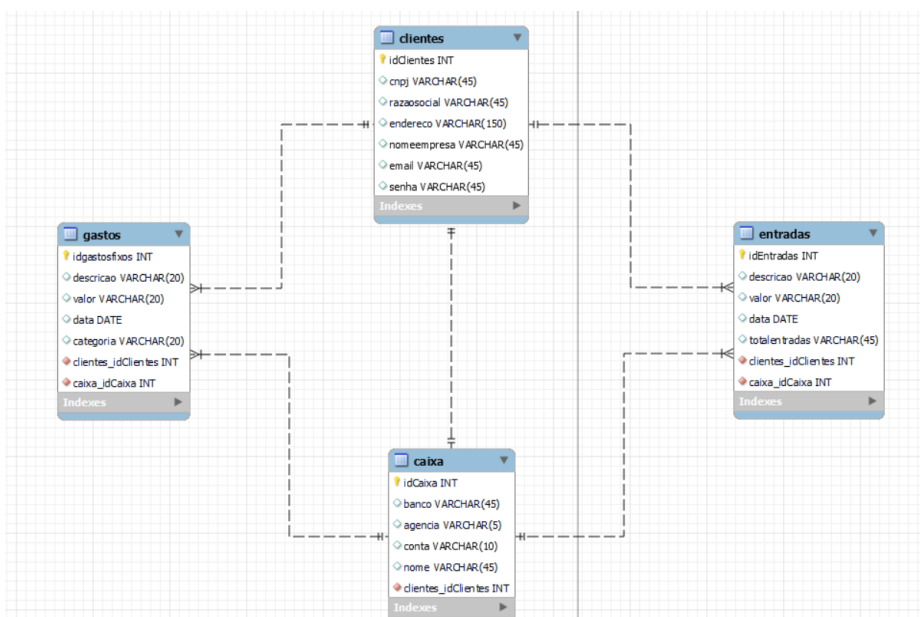
Fonte: Autor do Documento

Foi visto em sala de aula, que os modelos conceituais de dados são usados nas fases iniciais da modelagem de dados para organizar e definir conceitos e regras baseados em requisitos de caso de uso, em resumo, eles são diagramas de alto nível, que representam os conceitos de dados que suportam os negócios de nossa empresa, o objetivo principal de um modelo de dados conceitual no nosso projeto, é fornecer uma visão geral dos requisitos de informação envolvidos no projeto.

### 3.4.2 MODELO LÓGICO

**Figura 9 - Modelo Lógico do Sistema**





Fonte: Autor do Documento

Segundo o professor Osvaldo Kotaro Takai, mestre em Ciências de Computação e Matemática Computacional pela USP de São Carlos (ICMC-USP), em seu livro digital, a finalidade desses dois modelos, é garantir que o resultado atenda a um propósito inicial e, portanto, é necessário definir as bases, o caminho, os recursos e diversos outros aspectos para que seja possível obter um resultado plausível em qualquer tipo de empresa, em seus diversos segmentos.

Utilizamos o modelo lógico para estabelecer a estrutura dos elementos de dados e os relacionamentos entre eles, é por meio dessa estrutura formada por diversas informações sobre clientes, dados e procedimentos que são feitas consultas ao sistema, com os mais diversos objetivos, o modelo de dados lógico leva os elementos de modelagem de dados conceituais um passo adiante, adicionando mais informações a eles.

### 3.4.3 SQL

**Figura 10 - Código SQL do Sistema**

```

1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
6
7  -----
8  -- Schema projetopi
9  -----
10
11 -----
12 -- Schema projetopi
13 -----
14 CREATE SCHEMA IF NOT EXISTS `projetopi` DEFAULT CHARACTER SET utf8 ;
15 USE `projetopi` ;
16
17 -----
18 -- Table `projetopi`.`clientes`
19 -----
20 CREATE TABLE IF NOT EXISTS `projetopi`.`clientes` (
21   `idClientes` INT NOT NULL AUTO_INCREMENT,
22   `cnpj` VARCHAR(45) NULL,
23   `razaosocial` VARCHAR(45) NULL,
24   `endereço` VARCHAR(150) NULL,
25   `nomeempresa` VARCHAR(45) NULL,
26   `email` VARCHAR(45) NULL,
27   `senha` VARCHAR(45) NULL,
28   PRIMARY KEY (`idClientes`))
29 ENGINE = InnoDB;
30
31 -----
32
33 -- Table `projetopi`.`caixa`
34 -----
35 CREATE TABLE IF NOT EXISTS `projetopi`.`caixa` (
36   `idCaixa` INT NOT NULL AUTO_INCREMENT,
37   `banco` VARCHAR(45) NULL,
38   `agencia` VARCHAR(5) NULL,
39   `conta` VARCHAR(10) NULL,
40   `nome` VARCHAR(45) NULL,
41   `clientes_idClientes` INT NOT NULL,
42   PRIMARY KEY (`idCaixa`),
43   INDEX `fk_caixa_clientes1_idx` (`clientes_idClientes` ASC),
44   CONSTRAINT `fk_caixa_clientes1`

```

```

46 REFERENCES `projetopi`.`clientes` (`idClientes`)
47 ON DELETE NO ACTION
48 ON UPDATE NO ACTION)
49 ENGINE = InnoDB;
50
51
52 -----
53 -- Table `projetopi`.`gastos`
54 -----
55 CREATE TABLE IF NOT EXISTS `projetopi`.`gastos` (
56   `idgastosfixos` INT NOT NULL AUTO_INCREMENT,
57   `descricao` VARCHAR(20) NULL,
58   `valor` VARCHAR(20) NULL,
59   `data` DATE NULL,
60   `categoria` VARCHAR(20) NULL,
61   `clientes_idClientes` INT NOT NULL,
62   `caixa_idCaixa` INT NOT NULL,
63   PRIMARY KEY (`idgastosfixos`),
64   INDEX `fk_gastos_clientes1_idx` (`clientes_idClientes` ASC) ,
65   INDEX `fk_gastos_caixa1_idx` (`caixa_idCaixa` ASC) ,
66   CONSTRAINT `fk_gastos_clientes1`
67     FOREIGN KEY (`clientes_idClientes`)
68     REFERENCES `projetopi`.`clientes` (`idClientes`)
69     ON DELETE NO ACTION
70     ON UPDATE NO ACTION,
71   CONSTRAINT `fk_gastos_caixa1`
72     FOREIGN KEY (`caixa_idCaixa`)
73     REFERENCES `projetopi`.`caixa` (`idCaixa`)
74     ON DELETE NO ACTION
75     ON UPDATE NO ACTION)
76 ENGINE = InnoDB;
77
78
79 -----
80 -- Table `projetopi`.`entradas`
81 -----
82 CREATE TABLE IF NOT EXISTS `projetopi`.`entradas` (
83   `idEntradas` INT NOT NULL AUTO_INCREMENT,
84   `descricao` VARCHAR(20) NULL,
85   `valor` VARCHAR(20) NULL,
86   `data` DATE NULL,
87   `totalentradas` VARCHAR(45) NULL,
88   `clientes_idClientes` INT NOT NULL,
89   `caixa_idCaixa` INT NOT NULL,

```

```

66     CONSTRAINT `fk_gastos_clientes1`
67     FOREIGN KEY (`clientes_idClientes`)
68     REFERENCES `projetopi`.`clientes` (`idClientes`)
69     ON DELETE NO ACTION
70     ON UPDATE NO ACTION,
71     CONSTRAINT `fk_gastos_caixa1`
72     FOREIGN KEY (`caixa_idCaixa`)
73     REFERENCES `projetopi`.`caixa` (`idCaixa`)
74     ON DELETE NO ACTION
75     ON UPDATE NO ACTION)
76     ENGINE = InnoDB;
77
78
79     -----
80     -- Table `projetopi`.`entradas`
81     -----
82     CREATE TABLE IF NOT EXISTS `projetopi`.`entradas` (
83     `idEntradas` INT NOT NULL AUTO_INCREMENT,
84     `descricao` VARCHAR(20) NULL,
85     `valor` VARCHAR(20) NULL,
86     `data` DATE NULL,
87     `totalentradas` VARCHAR(45) NULL,
88     `clientes_idClientes` INT NOT NULL,
89     `caixa_idCaixa` INT NOT NULL,
90     PRIMARY KEY (`idEntradas`),
91     INDEX `fk_entradas_clientes_idx` (`clientes_idClientes` ASC) ,
92     INDEX `fk_entradas_caixa1_idx` (`caixa_idCaixa` ASC) ,
93     CONSTRAINT `fk_entradas_clientes`
94     FOREIGN KEY (`clientes_idClientes`)
95     REFERENCES `projetopi`.`clientes` (`idClientes`)
96     ON DELETE NO ACTION
97     ON UPDATE NO ACTION,
98     CONSTRAINT `fk_entradas_caixa1`
99     FOREIGN KEY (`caixa_idCaixa`)
100    REFERENCES `projetopi`.`caixa` (`idCaixa`)
101    ON DELETE NO ACTION
102    ON UPDATE NO ACTION)
103    ENGINE = InnoDB;
104
105
106    SET SQL_MODE=@OLD_SQL_MODE;
107    SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
108    SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
109

```

Fonte: Autor do Documento

## 4. CONCLUSÃO

A partir dos dados assim informados neste artigo, obteve-se um projeto funcional e capaz de atender as necessidades do cliente, o projeto foi realizado a partir da divisão de três pontos, Front End, Back End e documentação.

O primeiro ponto foi o Front End, onde foi realizado a escolha de um template base para idealização do software, a partir deste template foram utilizados os conhecimentos passados em lógica da programação para com um código HTML realizar alterações e atualizações neste software de forma que ele se torna-se simples e muito informativo ao cliente, é possível visualizar este template e mais informações sobre o mesmo no campo **ANEXOS**.

O Back End foi o banco de dados feito de forma com que fosse funcional e seguro, visando não somente receber estes dados do cliente, mas também que seja seguro, pois trata-se de informações financeiras de um cliente, dados estes que são de extrema importância, A partir disto se obtém um modelo lógico com apenas quatro tabelas, um modelo simples com poucas de tabelas de forma que se torna mais simples a manutenção e o reconhecimento de falhas no software, estas tabelas se conectam a partir de conexões simples, formando um looping onde um cliente faz uma entrada, essa entrada é armazenada em seu caixa, deste caixa saem seus gastos, finalizando a assim este looping em cliente possui gastos.

O terceiro ponto, a documentação se faz necessário para explicação e registros de todos os processos realizados dentro do Front e Back End, onde esta documentação visa apresentar de forma detalhada a forma com que o projeto foi idealizado e realizado, o mesmo se faz de extrema importância por pode apontar também dificuldades dentro da criação, dificuldades estas não que houveram muitas. Dentro do projeto a maior dificuldade foi a junção de nossa interface ao banco de dados por ter sido um tema abordado nas últimas aulas, esta questão da linkagem, foi um ponto onde se teve uma dificuldade porém não foi algo em que interferisse negativamente a estrutura do projeto, funcionou como um mecanismo de estudo obrigando a buscar conhecimento também de forma externa de como realizar este projeto.

As maiores dificuldades deste projeto, foram dificuldades de organização, sejam elas em relação ao tempo ou os processos a se realizar, eram muito poucos os horários onde se era possível reunir todos os membros para reuniões ou realização de etapas do projeto, desta

forma foi adotado um método de divisão de funções onde cada dupla de membros era responsável por determinado tema a ser abordado, seja ele estrutural do projeto ou na parte da documentação, a partir desta divisão foi possível concluir todos os processos, realizando assim a entrega de um projeto totalmente funcional, adotando os três pontos que são essenciais para o grupo que são eles **Simplicidade**, **Modular** e **Responsivo**, o cliente possui um software com grande quantidade informações, informações estas que são apresentadas de forma simples a ele sem que seja necessário realizar muitos processos para obtenção das mesmas, sendo possível visualizar informações já na tela inicial deste software, e de forma responsiva, o cliente não precisa necessariamente estar em frente a um computador ou desktop para visualização destes dados, ele possui acesso a estes dados na palma de sua mão seja em um tablet ou smartphone. Com as informações apresentadas a conclusão a ser tomada é de que este projeto foi realizado com sucesso e está pronto para utilização.

## REFERÊNCIAS

ALBANI, Victor. **Programação Orientada a Objetos**. Disponível em: <[http://redeetec.mec.gov.br/images/stories/pdf/eixo\\_infor\\_comun/tec\\_inf/081112\\_prog\\_obj.pdf](http://redeetec.mec.gov.br/images/stories/pdf/eixo_infor_comun/tec_inf/081112_prog_obj.pdf)>. Pág. 38. Acesso em: 13 out. 2022.

BARBOSA, Suria. **Na Prática. –O Criar um protótipo pode ser a chave para garantir o sucesso do seu negócio**. Disponível em: <<https://www.napratica.org.br/prototipagem-garante-sucesso-do-negocio/#:~:text=Prototipar%20-%20ou%20ação%20de%20gerar,de%20interação%20com%20o%20usuário>>. Acesso em: 11 out. 2022.

BESSA, André. **Algoritmos e Lógica de programação: O que são e qual a importância?**. Disponível em: <<https://www.alura.com.br/artigos/algoritmos-e-logica-de-programacao>>. Acesso em: 13 out. 2022.

BUDDE, R., ZULLIGHOVEN, H. (1990) “Prototyping revisited”, Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering, TelAviv, Israel, p. 418-427

CAMPOS FILHO, Ademar. **Demonstração dos fluxos de caixa: uma ferramenta indispensável para administrar sua empresa**. São Paulo: Atlas, 1999

COUTINHO, Thiago. **O Template é a melhor solução para se destacar nas redes sociais?**. Disponível em: <<https://www.voitto.com.br/blog/artigo/template>>. Acesso em: 13 out. 2022.

DAVID, Igor. **O Fluxo de caixa: tudo o que você precisa saber para ter sucesso e potencializar seu negócio**. Disponível em: <<https://tecnologia.com.br/blog/fluxo-de-caixa/>>. Acesso em: 11 out. 2022.

ERICH, Gamma . **Design Patterns: Elements of Reusable Object-Oriented Software. 1994**. Disponível em: <<https://www.amazon.com.br/Design-Patterns-Object-Oriented-Addison-Wesley-Professional-ebook/dp/B000SEIBB8>> Acesso em: 14 out. 2022.

IMPACTA, Redação; Impacta . **Entenda o que é lógica de programação Software. 1994**. Disponível em: <<https://www.impacta.com.br/blog/entenda-o-que-e-logica-de-programacao/>> Acesso em: 13 out. 2022.

FILHO, Paula, W. P. (2001), Engenharia de Software, LTC, 2ª edição.

TAKAI, Osvaldo; DCC-IME-USP . **Introdução a Bancos de Dados**. Disponível em: <<https://www.ime.usp.br/~jef/apostila.pdf>> Acesso em: 13 out. 2022.

MARCONDES, Renata; Gestão Financeira. **Fluxo de Caixa 08.08**  
Acesso em: 17 out. 2022.



# ANEXOS

Nos anexos abaixo, teremos os resultados do Front-End, que serão as interfaces acessíveis aos usuários:

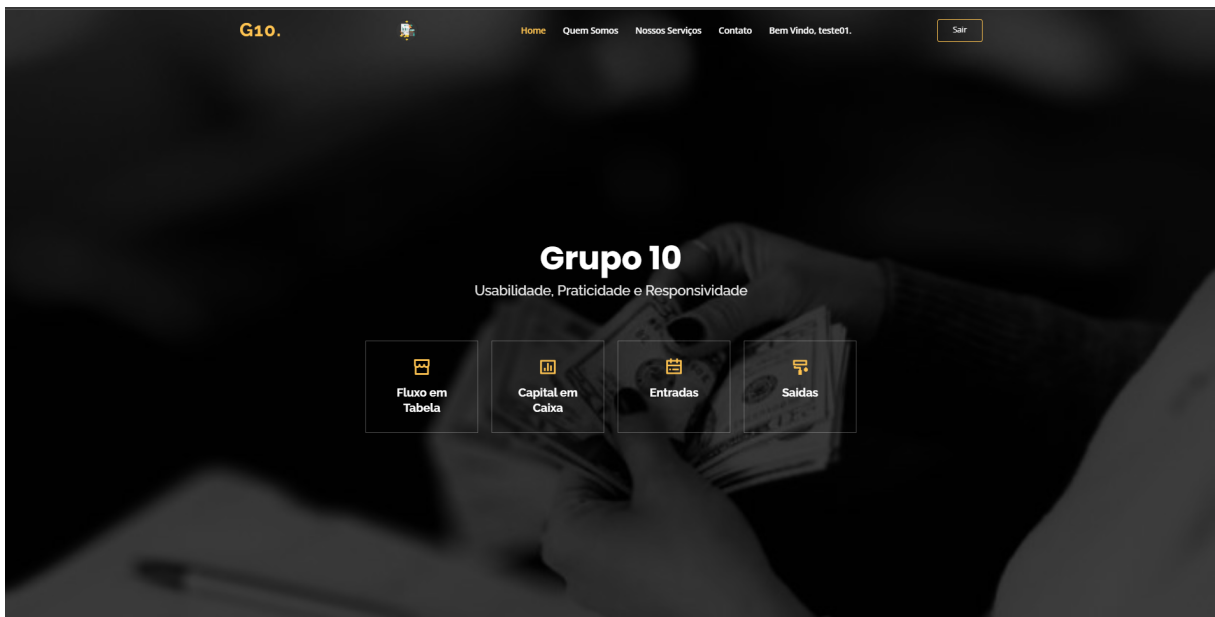


Figura 1: Nesta primeira dashboard, é possível visualizar a home do projeto, a mesma se apresenta de forma simples e direta, com ícones e atalhos de fácil entendimento, que levam o usuário até a área desejada.

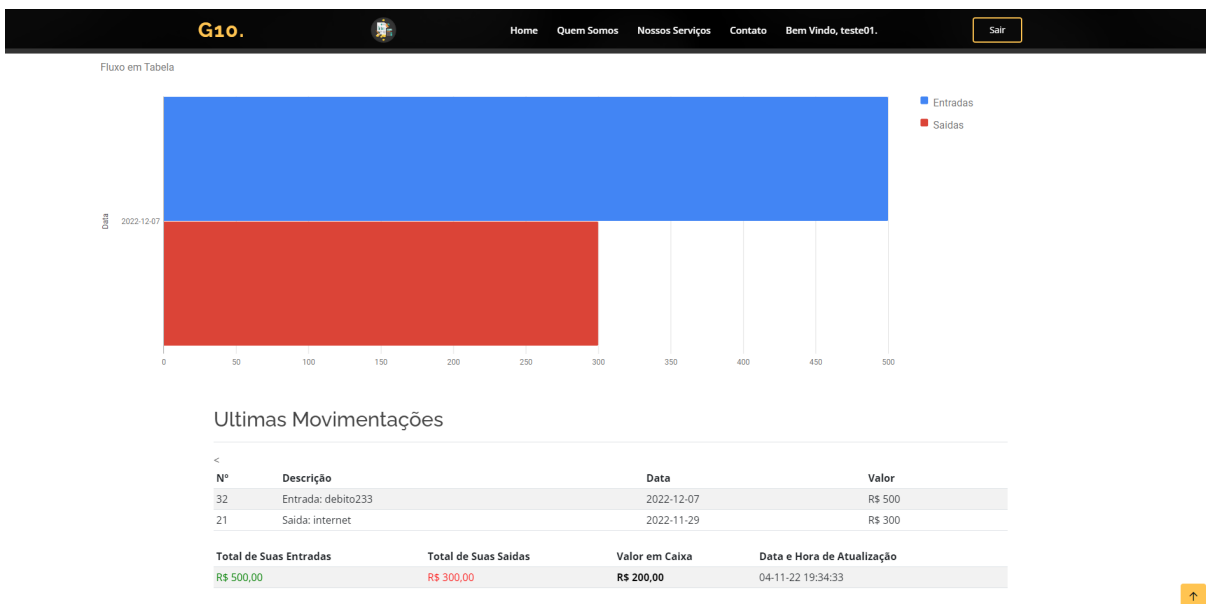


Figura 2: A segunda dashboard, apresenta um gráfico interativo de um fluxo de caixa simples, que representa entradas e saídas de capital, divididas de forma mensal, sendo as linhas azuis, entradas, e as vermelhas saídas, possibilitando uma primeira visualização dos resultados.

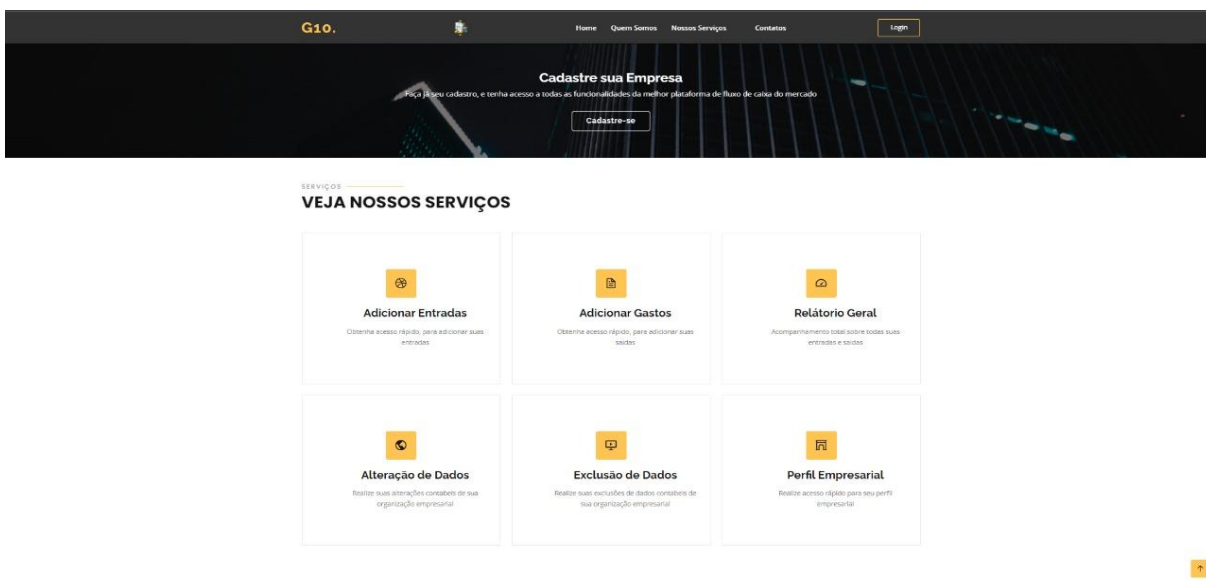


Figura 3: Em seguida, a terceira dashboard apresenta campos interativos, onde o usuário pode adicionar dados de entradas e saídas, consultar relatórios, realizar alterações de dados, e buscar mais informações do perfil empresarial dos desenvolvedores.

Figura 4: Nesta dashboard, o cliente realiza interações diretas com o banco de dados, onde é possível adicionar informações que serão armazenadas, tornando possível visualizar essas informações futuramente, sendo apresentadas em relatórios ou gráficos.

Figura 5: Em seguida, a quarta dashboard apresenta campos interativos, onde o usuário pode adicionar dados de saídas.

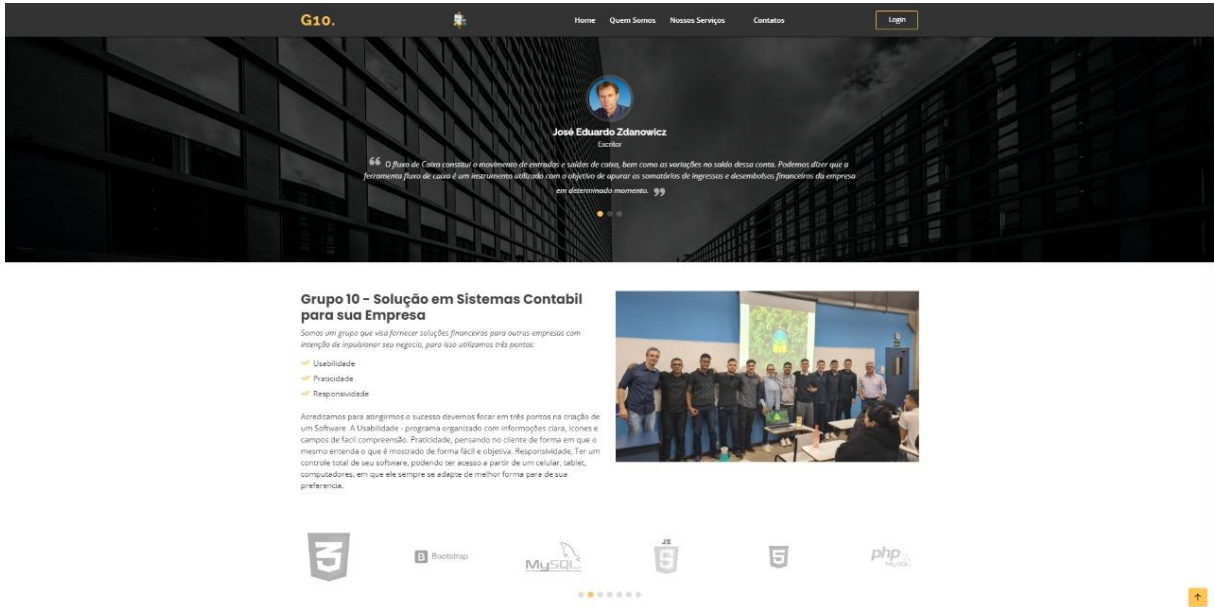


Figura 6: Esta dashboard, apresenta a equipe de desenvolvedores e softwares utilizados na criação do projeto

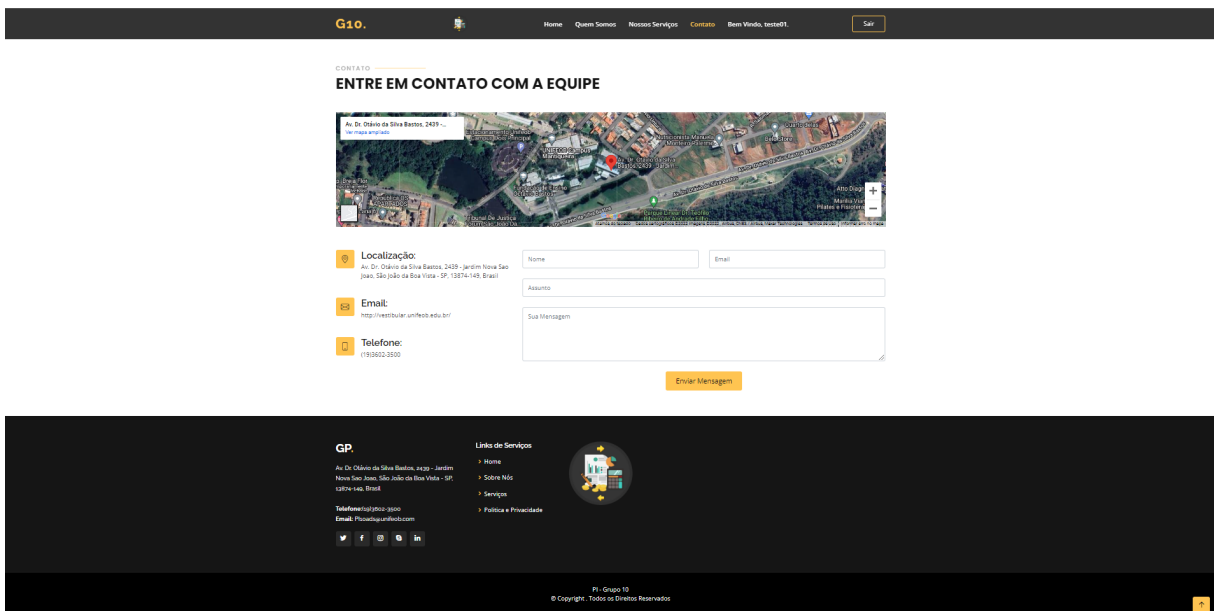


Figura 7: Nessa dashboard, o usuário terá campos interativos, onde será possível realizar contato de forma direta com a equipe de desenvolvedores.

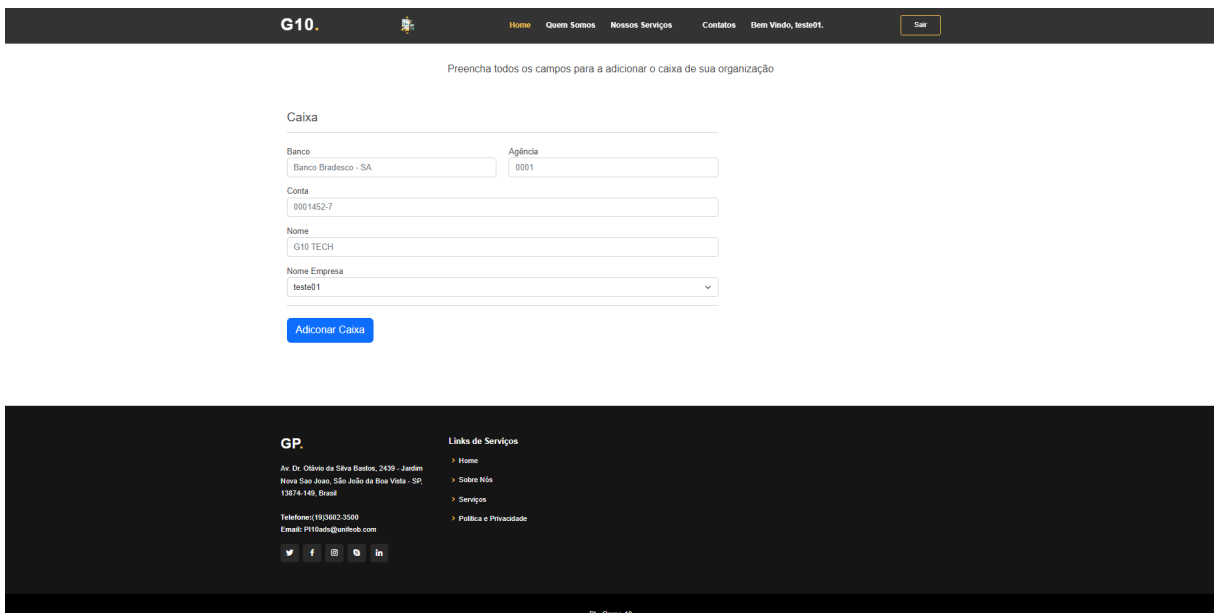


Figura 8: Na oitava dashboard, apresenta-se a tela de cadastro do caixa da empresa onde o mesmo é enviado para o banco de dados para que na inserção de entradas ou saídas o usuário possa selecionar qual caixa deseja.

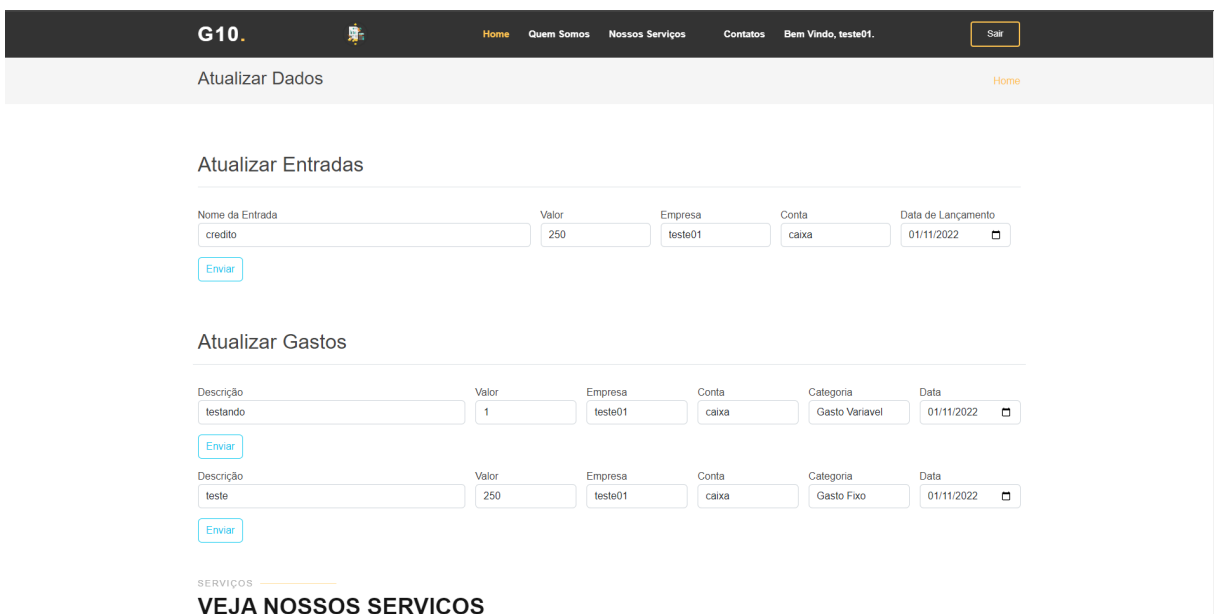


Figura 9: Na nona dashboard, é apresentada a área de alteração de dados, onde a mesma é subdividida em entradas ou saídas, o usuário pode realizar a alteração completa de todos os campos presentes na página.

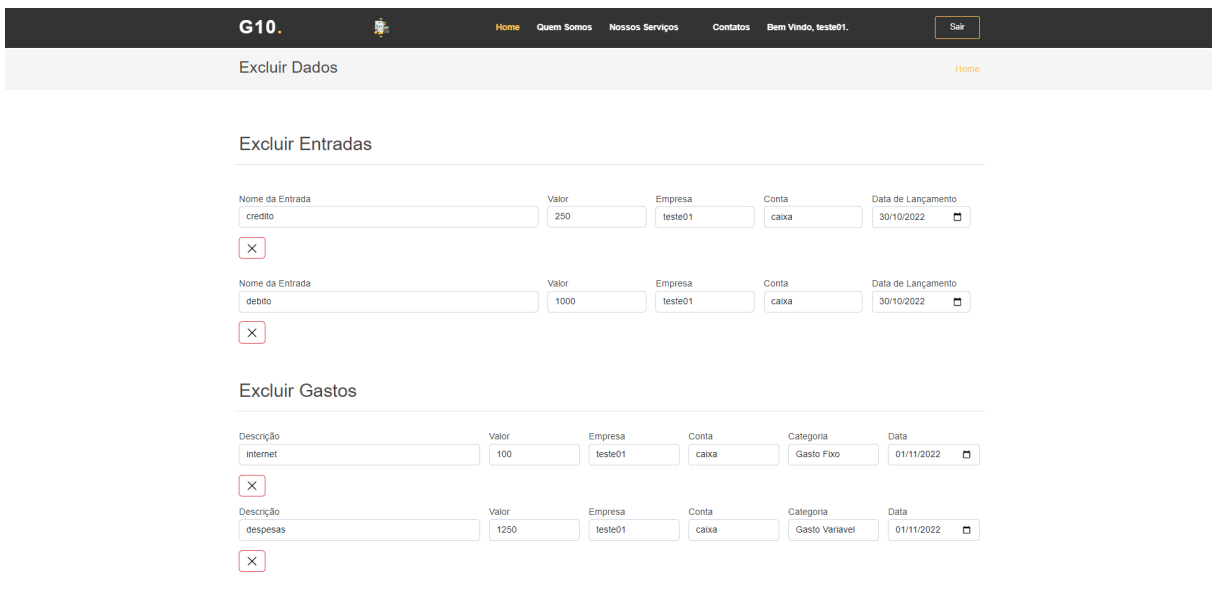


Figura 10: Na décima dashboard, apresenta-se a tela de exclusão de dados, onde as mesmas também se fazem subdivididas em entradas ou saídas.

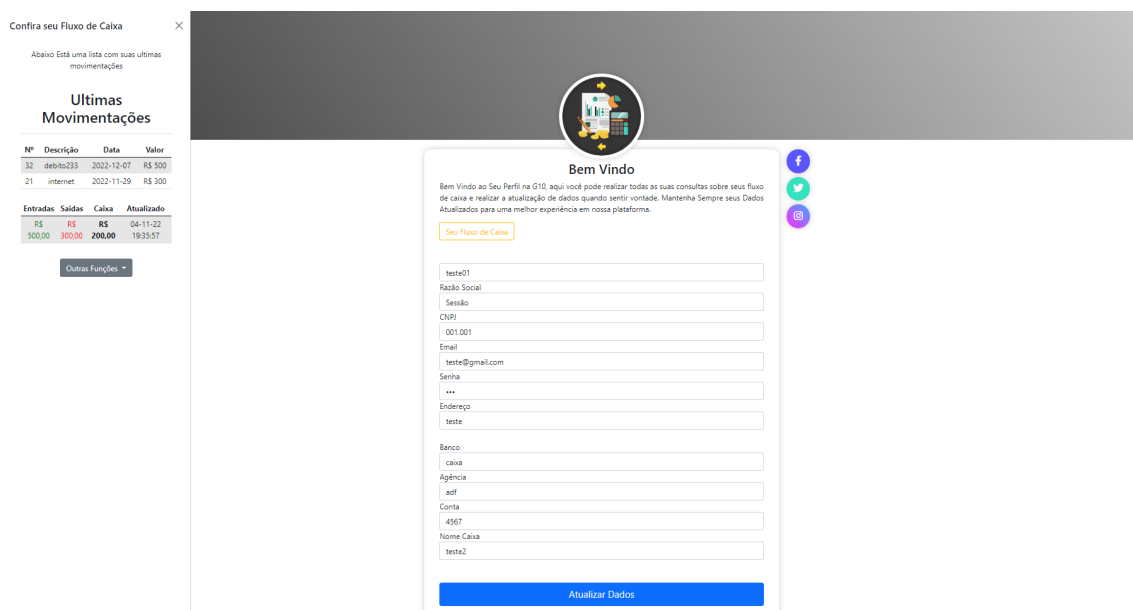




Figura 11: Na décima primeira dashboard, foi realizado o perfil de usuário, onde os mesmos podem realizar atualizações sobre a empresa, caixa e ainda acompanhar suas movimentações através de um menu presente na lateral da página.



**G10.**  [Home](#) [Quem Somos](#) [Nossos Serviços](#) [Contatos](#) [Login](#)

Painel de Relatórios e Notas [Home](#)

### Notas de Saídas

Nº	Descrição	Data	Gerar PDF
3	Relatório Geral	Todos as Datas	

### Notas de Entradas

Nº	Descrição	Data	Gerar PDF
26	credito	2022-10-30	
27	debito	2022-10-30	

### Notas de Saídas




Nº	Descrição	Data	Gerar PDF
14	internet	2022-11-01	
15	despesas	2022-11-01	


Figura 12: Na décima segunda dashboard, mostra-se o painel de relatórios e notas onde o usuário pode consultar relatórios de suas movimentações, ou gerar notas específicas como de entradas ou saídas.

 <p>G10 - Soluções Contábeis São João da Boa Vista - São Paulo Telefone: (19)3602-3300</p>	<b>NFS-e</b>		<p>modelo da nota</p> <p>data e hora de emissão</p> <p>status da nota</p> <p>status de restrição</p>												
	<p>PRESTADOR DE SERVIÇOS</p> <p>CNPJ/CPF: 024.03241/001 Nome/Razão Social: G10 - Soluções Contábeis Endereço: Rua da Oliveira da Silva Bastos, 2430 - Jardim Mirim São João CEP: 13074-400 Bairro: São João da Boa Vista UF: São João da Boa Vista</p>														
<p>TOMADOR DE SERVIÇO</p> <p>Nome/Razão Social: teste010   teste0 CNPJ/CPF: 024.03241/001 Endereço: teste0</p>															
<p>DISCRIMINAÇÃO DOS SERVIÇOS</p> <table border="1"> <thead> <tr> <th>Nº</th> <th>Descrição</th> <th>Data</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>26</td> <td>credito</td> <td>2022-10-30</td> <td>R\$ 250</td> </tr> <tr> <td>27</td> <td>debito</td> <td>2022-10-30</td> <td>R\$ 1000</td> </tr> </tbody> </table>			Nº	Descrição	Data	Valor	26	credito	2022-10-30	R\$ 250	27	debito	2022-10-30	R\$ 1000	
Nº	Descrição	Data	Valor												
26	credito	2022-10-30	R\$ 250												
27	debito	2022-10-30	R\$ 1000												
		internet	2022- R\$ 11-01 100												
		despesas	2022- R\$ 11-01 1250												

Imprima Página .

G10 - Soluções Contábeis

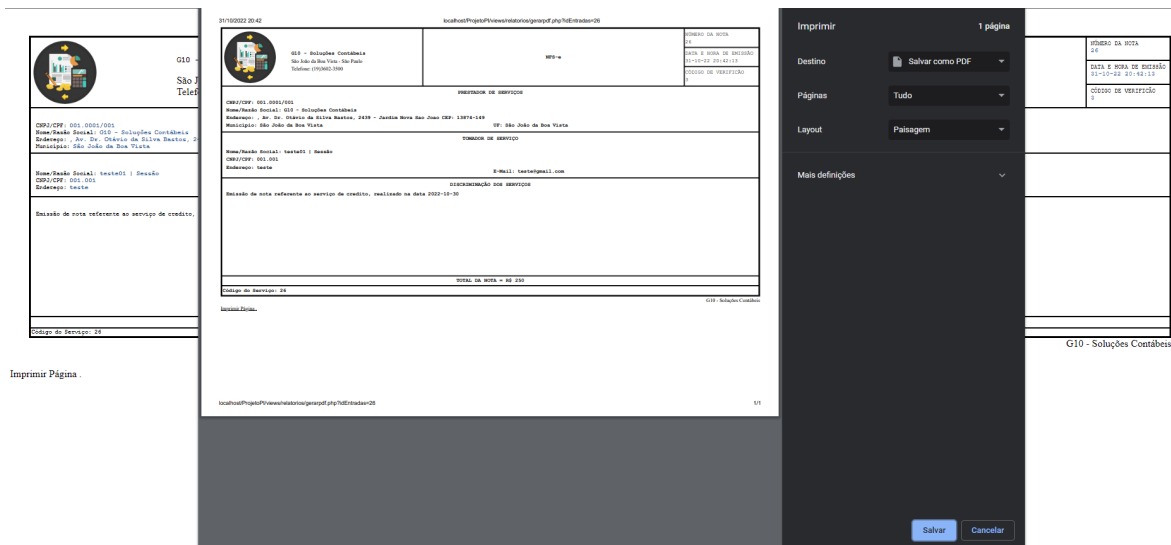
Figura 13: Na décima terceira dashboard, mostra-se como é gerado a o relatório geral de movimentações, onde neste relatório se faz presente todas as movimentações do usuário, classificando por data, valor e descrição de serviço, também é puxado dados da empresa na parte superior, além de mostrar código de relatório e data e hora de solicitação.

 <p>G10 - Soluções Contábeis São João da Boa Vista - São Paulo Telefone: (19)3602-3300</p>	<p>NFS-e</p>	<p>NÚMERO DA NOTA 26</p>
		<p>DATA E HORA DE EMISSÃO 20/10/2022 20:42:03</p>
<p>PRESTADOR DE SERVIÇOS</p> <p>CNPJ/CPF: 051.0051/001 Nome/Razão Social: G10 - Soluções Contábeis Endereço: Av. Dr. Orlando de Silva Bastos, 2419 - Jardim Nova São João CEP: 13874-149 Município: São João da Boa Vista</p>		<p>TIPO DE SERVIÇO</p> <p>UF: São João da Boa Vista</p>
<p>TOMADOR DE SERVIÇO</p> <p>Nome/Razão Social: teste01   teste0 CNPJ/CPF: 051.001 Endereço: teste0</p>		<p>E-MAIL: teste@gmail.com</p>
<p>DISCRIMINAÇÃO DOS SERVIÇOS</p> <p>Descrição de nota referente ao serviço de crédito, realizado na data 2022-10-30</p>		
<p>TOTAL DA NOTA = R\$ 250</p>		
<p>Código de Serviço: 26</p>		

G10 - Soluções Contábeis

Imprimir Página .

Figura 14: Na décima quarta dashboard, mostra o modelo de nota específica, podendo ser de entrada ou saída conforme a solicitação realizada, onde mostra-se na discriminação do serviço a data, a descrição do serviço, a categoria (se houver), e logo abaixo segue o valor total da nota, também é puxado dados da empresa na parte superior, além de mostrar código de relatório e data e hora de solicitação.



The image shows a dashboard interface with a central preview of a PDF invoice. The invoice content is identical to the one in Figure 14. To the right of the preview is a dark-themed print settings panel. The panel has a title 'Imprimir' and a page count '1 página'. It includes several dropdown menus: 'Destino' set to 'Salvar como PDF', 'Páginas' set to 'Tudo', and 'Layout' set to 'Paisagem'. There is also a 'Mais definições' section with a downward arrow. At the bottom of the panel are two buttons: 'Salvar' and 'Cancelar'. The dashboard also shows a 'G10 - Soluções Contábeis' logo and name at the top left and bottom right of the preview area.

Figura 15: Na décima quinta dashboard, mostra a maneira de salvar no modelo PDF, ou impressão dos relatórios.



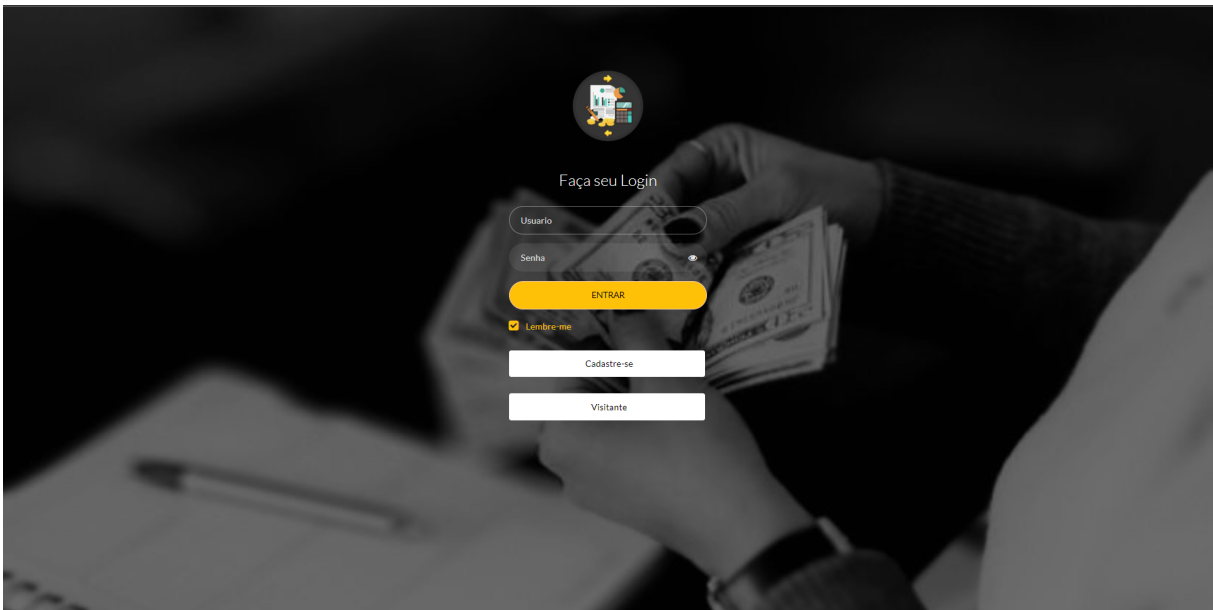


Figura 16: Na décima sexta dashboard, mostra-se a tela de login do sistema, onde o usuário entra com os campos de email e senha, contando também com um modo de visitante..

Figura 17: Na décima sétima dashboard, mostra-se a tela de cadastro de empresa.