





**UNifeob**

ESCOLA DE NEGÓCIOS



**2023**

**PROJETO DE CONSULTORIA  
EMPRESARIAL**



UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

**PROJETO INTEGRADO**  
**SISTEMA EMPRESARIAL**

MÓDULO DESENVOLVIMENTO DESKTOP

Banco de Dados – Prof. Sidney Gitcoff Telles

Programação Orientada a Objeto – Prof. Sidney Gitcoff Telles

Projeto de Desenvolvimento Desktop – Prof. Sidney Gitcoff Telles

Estudantes:

Leonardo Bueno Sossai, RA 1012023100172  
Felipe Larotonda Barbosa da Silva, RA 1012023100269  
Tatiane Daciane De Farias Godoy , RA 1012023100221  
Lucas Rafael candido da Silva, RA 1012023100143  
Gabriele Rigobeli da Silva, RA 1012023100062

SÃO JOÃO DA BOA VISTA, SP  
ABRIL, 2023

**SUMÁRIO**

<b>1 INTRODUÇÃO</b>	<b>4</b>	<b>2 DESCRIÇÃO DA EMPRESA</b>	<b>6</b>	<b>3 PROJETO DE CONSULTORIA EMPRESARIAL</b>	<b>7</b>
<b>3.1 BANCO DE DADOS</b>					<b>7</b>
<b>3.1.1</b>	<b>5</b>	<b>3.1.2</b>	<b>6</b>	<b>3.1.3</b>	<b>TÓPICO 3 7</b>
<b>3.2 PROGRAMAÇÃO ORIENTADA A OBJETOS</b>					<b>8</b>
<b>3.2.1</b>	<b>10</b>	<b>3.2.2</b>	<b>12</b>	<b>3.2.3</b>	<b>14</b>
<b>3.3 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ADAPTANDO-SE A MUDANÇAS</b>					<b>9</b>
<b>3.3.1 ADAPTANDO-SE A MUDANÇAS</b>	<b>9</b>	<b>3.3.2 ESTUDANTES NA PRÁTICA</b>			
	<b>9</b>				
<b>4 CONCLUSÃO</b>	<b>11</b>	<b>REFERÊNCIAS</b>	<b>12</b>	<b>ANEXOS</b>	<b>13</b>

## **1 INTRODUÇÃO**

Este projeto de consultoria visa implementar um sistema empresarial na Vila Tropical Brasil, uma empresa do ramo de confecções de bonecas ciganas artesanais. O projeto envolve a criação de um banco de dados com três tabelas (Contas a Pagar, Contas a Receber e Consulta) e o desenvolvimento do backend e frontend do sistema para interação com o banco de dados criando automatizando o controle financeiro, através do cadastramento de contas a pagar e receber.

## **2 DESCRIÇÃO DA EMPRESA**

Após uma pesquisa regional com pequenos empresários foi identificado a necessidade da implementação de um sistema empresarial na Vila Tropical Brasil uma empresa do ramo de confecções de bonecas ciganas artesanais, empresa localizada na R. Oscar Guanabara N° 236 - Sem. Vasconcelos - Rio de Janeiro. Empresa com ênfase em vendas online e por encomendas de artigos artesanais.

### 3 PROJETO DE CONSULTORIA EMPRESARIAL

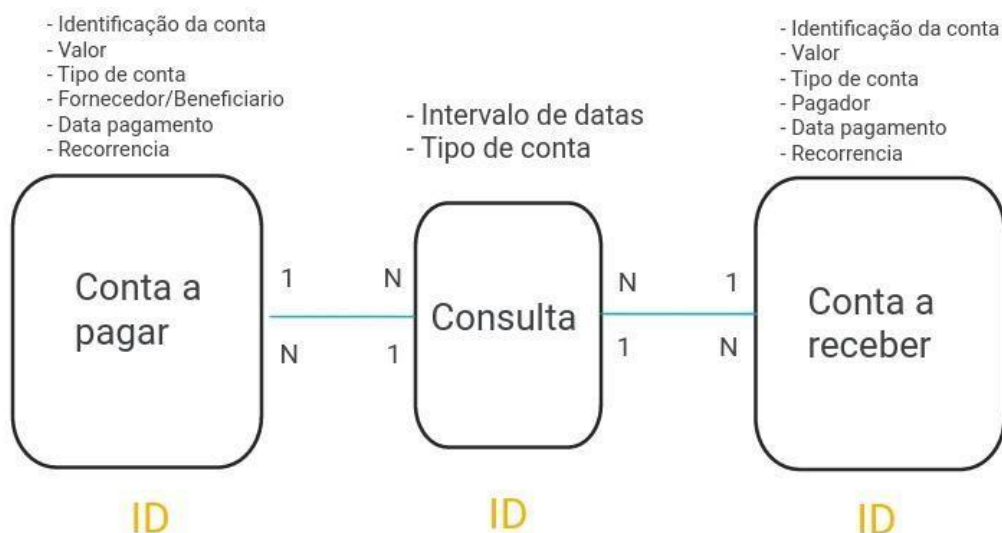
Após uma reunião com representante da empresa foi identificada a necessidade de implementação de um sistema de software para uma melhor organização financeira contemplando banco de dados referente a contas a pagar, contas a receber e consultas das mesmas, otimizando o tempo de buscas com apenas alguns cliques. Para isso será necessário a criação de um micro sistema inicial. O sistema poderá ser otimizado conforme a necessidade do cliente com o passar do tempo.

#### 3.1 BANCO DE DADOS

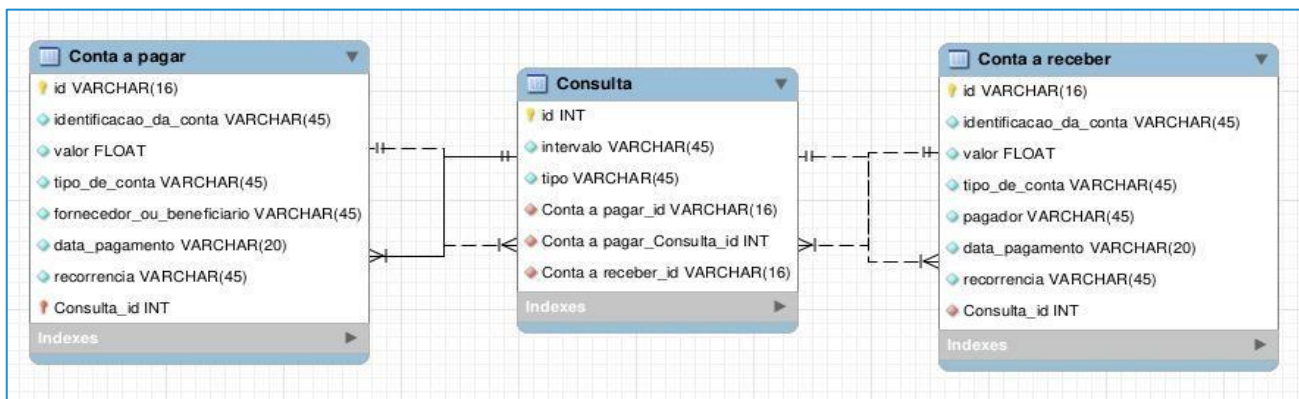
Para criação do banco de dados foi identificada a necessidade da criação de três tabelas são elas:

- Contas a Pagar
- Contas a Receber
- Consulta

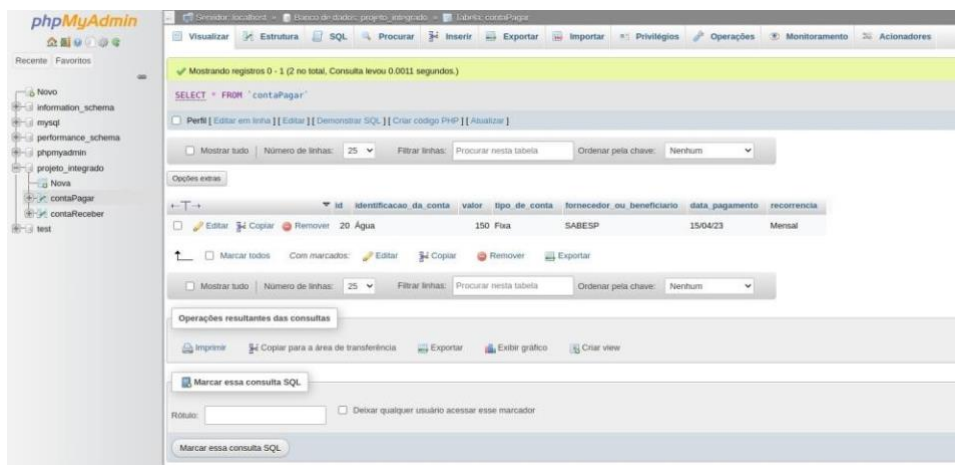
##### 3.1.1 MER - MODELAGEM ENTIDADE RELACIONAMENTO



### 3.1.2 DER - DIAGRAMA ENTIDADE RELACIONAMENTO



### 3.1.3 FÍSICO



phpMyAdmin

Recente Favoritos

- Novo
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- projeto\_integrado
  - Nova
  - contaPagar
  - contaReceber
  - test

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Privilegios Operacoes Monitoramento Acionadores

Estrutura da tabela Visão de relação(ões)

#	Nome	Tipo	Colação	Atributos	Nulo	Padrão	Comentários	Extra	Ação
1	id	int(11)			Não	Nenhum		AUTO_INCREMENT	Alterar Eliminar Mais
2	identificacao_da_conta	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
3	valor	float			Não	Nenhum			Alterar Eliminar Mais
4	tipo_de_conta	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
5	remecedor_ou_beneficiario	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
6	data_pagamento	varchar(20)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
7	reocorrencia	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais

Índices

Ação	Nome da chave	Tipo	Único	Pacote	Coluna	Cardinalidade	Colação	Nulo	Comentário
Editar Renomear Eliminar	PRIMARY	BTREE	Sim	Não	id	1	A	Não	

Partições

Nenhum particionamento definido

phpMyAdmin

Recente Favoritos

- Novo
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- projeto\_integrado
  - Nova
  - contaPagar
  - contaReceber
  - test

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Privilegios Operacoes Monitoramento Acionadores

Estrutura da tabela Visão de relação(ões)

#	Nome	Tipo	Colação	Atributos	Nulo	Padrão	Comentários	Extra	Ação
1	id	int(11)			Não	Nenhum		AUTO_INCREMENT	Alterar Eliminar Mais
2	identificacao_da_conta	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
3	valor	float			Não	Nenhum			Alterar Eliminar Mais
4	tipo_de_conta	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
5	pagador	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
6	data_pagamento	varchar(20)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais
7	reocorrencia	varchar(45)	utf8_general_ci		Não	Nenhum			Alterar Eliminar Mais

Índices

Ação	Nome da chave	Tipo	Único	Pacote	Coluna	Cardinalidade	Colação	Nulo	Comentário
Editar Renomear Eliminar	PRIMARY	BTREE	Sim	Não	id	0	A	Não	

Partições

Nenhum particionamento definido

phpMyAdmin

Recente Favoritos

- Novo
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- projeto\_integrado
  - Nova
  - contaPagar
  - contaReceber
  - test

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Privilegios Operacoes Monitoramento Acionadores

Mostrando registros 0 - 0 (1 no total. Consulta levou 0.0012 segundos.)

SELECT \* FROM `contaReceber`

Perfis [ Editar em linha ] [ Editar ] [ Demonstrar SQL ] [ Criar código PHP ] [ Atualizar ]

Mostrar tudo | Número de linhas: 25 | Filtrar linhas: Procurar nesta tabela

Opções extras

	id	identificacao_da_conta	valor	tipo_de_conta	pagador	data_pagamento	reocorrencia
Editar Copiar Remover	8	Pagamento de bonecas	500	Varável	Joao Silva	12/04/23	Única

Operações resultantes das consultas

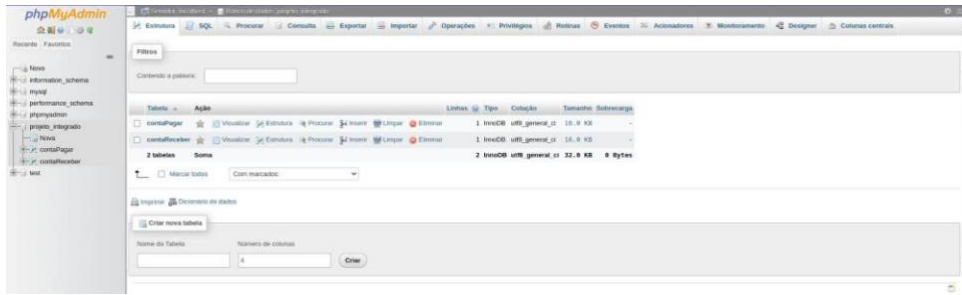
Imprimir Copiar para a área de transferência Exportar Exibir gráfico Criar view

Marcar essa consulta SQL

Rótulo:   Deixar qualquer usuário acessar esse marcador

Marcar essa consulta SQL





3.1.1 e 3.1.2. Tirar print dos comandos de criação e das tabelas do banco criado.

```

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 mydb
9  -----
10
11 -- Schema mydb
12 -----
13
14 CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
15 USE `mydb` ;
16
17 -----
18 -- Table `mydb`.`Conta a receber`
19 -----
20 CREATE TABLE IF NOT EXISTS `mydb`.`Conta a receber` (
21   `id` VARCHAR(10) NOT NULL,
22   `identificacao da conta` VARCHAR(45) NOT NULL,
23   `valor` FLOAT NOT NULL,
24   `tipo de conta` VARCHAR(45) NOT NULL DEFAULT CURRENT_TIMESTAMP,
25   `pagador` VARCHAR(45) NOT NULL,
26   `data pagamento` VARCHAR(20) NOT NULL,
27   `recorrencia` VARCHAR(45) NOT NULL,
28   `Consulta id` INT NOT NULL,
29   PRIMARY KEY (`id`),
30   INDEX `fk_Conta a receber_Consulta_id` (`Consulta_id` ASC) VISIBLE,
31   CONSTRAINT `fk_Conta a receber_Consulta_id`
32     FOREIGN KEY (`Consulta_id`)
33     REFERENCES `mydb`.`Consulta` (`id`)
34     ON DELETE NO ACTION
35     ON UPDATE NO ACTION);
36

```

```

38 -----
39 -- Table `mydb`.`Consulta`
40 -----
41 CREATE TABLE IF NOT EXISTS `mydb`.`Consulta` (
42   `id` INT NOT NULL AUTO_INCREMENT,
43   `intervalo` VARCHAR(45) NOT NULL,
44   `tipo` VARCHAR(45) NOT NULL,
45   `Conta a pagar_id` VARCHAR(16) NOT NULL,
46   `Conta a pagar_consulta_id` INT NOT NULL,
47   `Conta a receber_id` VARCHAR(16) NOT NULL,
48   PRIMARY KEY (`id`),
49   INDEX `fk_consulta_conta a pagar1_idx` (`Conta a pagar_id` ASC, `Conta a pagar_consulta_id` ASC) VISIBLE,
50   INDEX `fk_consulta_conta a receber1_idx` (`Conta a receber_id` ASC) VISIBLE,
51   CONSTRAINT `fk_consulta_conta a pagar1`
52     FOREIGN KEY (`Conta a pagar_id` , `Conta a pagar_consulta_id`)
53     REFERENCES `mydb`.`Conta a pagar` (`id` , `Consulta_id`)
54     ON DELETE NO ACTION
55     ON UPDATE NO ACTION,
56   CONSTRAINT `fk_consulta_conta a receber1`
57     FOREIGN KEY (`Conta a receber_id`)
58     REFERENCES `mydb`.`Conta a receber` (`id`)
59     ON DELETE NO ACTION
60     ON UPDATE NO ACTION);
61
62 -----
63 -- Table `mydb`.`Conta a pagar`
64 -----
65 -----
66 CREATE TABLE IF NOT EXISTS `mydb`.`Conta a pagar` (
67   `id` VARCHAR(16) NOT NULL,
68   `identificacao_da_conta` VARCHAR(45) NOT NULL,
69   `valor` FLOAT NOT NULL,
70   `tipo_de_conta` VARCHAR(45) NOT NULL DEFAULT CURRENT_TIMESTAMP,
71   `fornecedor_ou_beneficiario` VARCHAR(45) NOT NULL,
72   `data_pagamento` VARCHAR(20) NOT NULL,
73   `recorrencia` VARCHAR(45) NOT NULL,
74   `Consulta_id` INT NOT NULL,
75   PRIMARY KEY (`id` , `Consulta_id`),
76   INDEX `fk_conta a pagar_consulta_idx` (`Consulta_id` ASC) VISIBLE,
77   CONSTRAINT `fk_conta a pagar_consulta`
78     FOREIGN KEY (`Consulta_id`)
79     REFERENCES `mydb`.`Consulta` (`id`)
80     ON DELETE NO ACTION

```

## 3.2 PROGRAMAÇÃO ORIENTADA A OBJETOS

Utilizando a IDE do NetBens foi desenvolvido todo o backend e frontend do sistema para ser realizado a interação com o banco de dados criado. Foram criadas três telas sendo elas:

- Tela de cadastro de conta a pagar
- Tela de cadastro conta a receber •

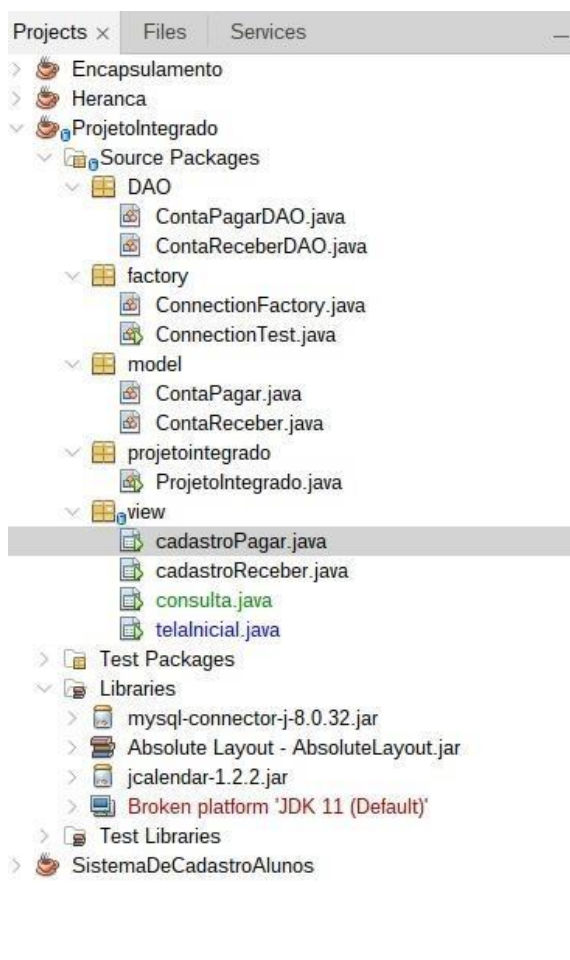
Tela consulta

### 3.2.1 DIAGRAMA DE CLASSES

The image shows two screenshots of an IDE displaying Java source code. The top screenshot shows the beginning of the `ContaPagar` class, including package declaration, a Javadoc comment, and the class definition with its attributes and the first three methods. The bottom screenshot continues the code, showing the remaining methods for the `ContaPagar` class.

```
4  */
5  package model;
6
7
8  /**
9   *
10  * @author ls
11  */
12  public class ContaPagar {
13
14      int id;
15      String identificacao_da_conta;
16      float valor;
17      String tipo_de_conta;
18      String fornecedor_ou_beneficiario;
19      String ocorrencia;
20      String data_pagamento;
21
22      public int getId() {
23          return id;
24      }
25
26      public void setId(int id) {
27          this.id = id;
28      }
29
30      public String getIdentificacao_da_conta() {
31          return identificacao_da_conta;
32      }
33
34      public void setIdentificacao_da_conta(String identificacao_da_conta) {
35          this.identificacao_da_conta = identificacao_da_conta;
36      }
37
38      public float getValor() {
39          return valor;
40      }
41
42      public void setValor(float valor) {
43          this.valor = valor;
44      }
45
46      public String getTipo_de_conta() {
47          return tipo_de_conta;
48      }
49
50      public void setTipo_de_conta(String tipo_de_conta) {
51          this.tipo_de_conta = tipo_de_conta;
52      }
53
54      public String getFornecedor_ou_beneficiario() {
55          return fornecedor_ou_beneficiario;
56      }
57
58      public void setFornecedor_ou_beneficiario(String fornecedor_ou_beneficiario) {
59          this.fornecedor_ou_beneficiario = fornecedor_ou_beneficiario;
60      }
61
62      public String getRecorrencia() {
63          return ocorrencia;
64      }
65
66      public void setRecorrencia(String ocorrencia) {
```

```
ContaPagar.java x ContaReceber.java x cadastroReceber.java x cadastroPagar.java x
Source History
59 |         this.fornecedor_ou_beneficiario = fornecedor_ou_beneficiario;
60 |     }
61 |
62 |     public String getRecorrencia() {
63 |         return recorrencia;
64 |     }
65 |
66 |     public void setRecorrencia(String recorrencia) {
67 |         this.recorrencia = recorrencia;
68 |     }
69 |
70 |     public String getData_pagamento() {
71 |         return data_pagamento;
72 |     }
73 |
74 |     public void setData_pagamento(String data_pagamento) {
75 |         this.data_pagamento = data_pagamento;
76 |     }
77 |
78 |
79 | }
80 |
```



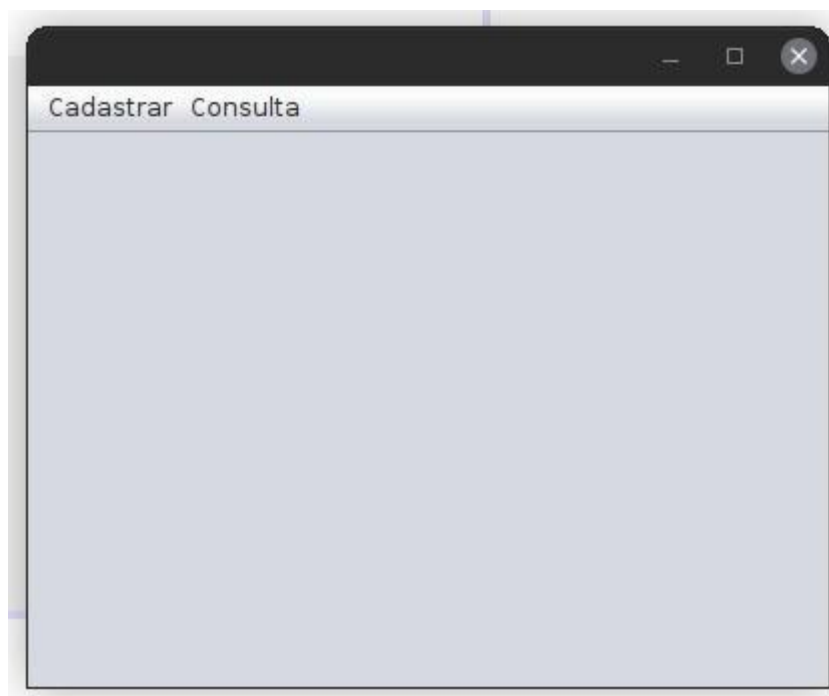
## 3.2.2 CÓDIGOS DO SISTEMA

```
ContaPagar.java x ContaReceber.java x cadastroReceber.java x cadastroPagar.java x ContaReceberDA
Source History
1  /**
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package model;
6
7  /**
8  *
9  * @author ls
10 */
11 public class ContaReceber {
12
13     String identificacao_da_conta;
14     float valor;
15     String tipo_de_conta;
16     String pagador;
17     String ocorrencia;
18     String data_pagamento;
19
20     public String getIdentificacao_da_conta() {
21         return identificacao_da_conta;
22     }
23
24     public void setIdentificacao_da_conta(String identificacao_da_conta) {
25         this.identificacao_da_conta = identificacao_da_conta;
26     }
27
28     public float getValor() {
29         return valor;
30     }
31
32     public void setValor(float valor) {
33         this.valor = valor;
```

```
ContaPagar.java x ContaReceber.java x cadastroReceber.java x cadastroPagar.java x
Source History
31
32     public void setValor(float valor) {
33         this.valor = valor;
34     }
35
36     public String getTipo_de_conta() {
37         return tipo_de_conta;
38     }
39
40     public void setTipo_de_conta(String tipo_de_conta) {
41         this.tipo_de_conta = tipo_de_conta;
42     }
43
44     public String getPagador() {
45         return pagador;
46     }
47
48     public void setPagador(String pagador) {
49         this.pagador = pagador;
50     }
51
52     public String getRecorrencia() {
53         return ocorrencia;
54     }
55
56     public void setRecorrencia(String ocorrencia) {
57         this.occorrencia = ocorrencia;
58     }
59
60     public String getData_pagamento() {
61         return data_pagamento;
62     }
63
```

```
ContaPagar.java x ContaReceber.java x cadastroReceber.java x cadastroPagar.java x
Source History
49 |         this.pagador = pagador;
50 |     }
51 |
52 |     public String getRecorrencia() {
53 |         return recorrencia;
54 |     }
55 |
56 |     public void setRecorrencia(String recorrencia) {
57 |         this.recorrencia = recorrencia;
58 |     }
59 |
60 |     public String getData_pagamento() {
61 |         return data_pagamento;
62 |     }
63 |
64 |     public void setData_pagamento(String data_pagamento) {
65 |         this.data_pagamento = data_pagamento;
66 |     }
67 |
68 | }
69 |
70 |
```

### 3.2.3 IMAGENS DO SISTEMA



### CADASTRO - CONTA A PAGAR

Identificação da conta  Valor R\$

Tipo de conta  Beneficiário / Fornecedor

Data pagamento  Recorrência

### CADASTRO - CONTA A RECEBER

Identificação da conta  Valor R\$

Tipo de conta  Pagador

Data pagamento  Recorrência

### Consulta de contas

Tipo de conta:  Recorrência

Identificação da conta	Valor R\$	Data pagamento	Beneficiário / Fornecedor	Tipo de conta
Água	150.0	15/04/23	SABESP	Fixa
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

## 4 CONCLUSÃO

Em conclusão, o desenvolvimento de um software para uma empresa de artesanato que inclua funcionalidades de contas a pagar e receber pode trazer diversos benefícios para o negócio. Com a possibilidade de automatizar processos financeiros, integrar informações e controlar o fluxo de caixa, a empresa poderá oferecer um atendimento mais rápido e eficiente, otimizando o tempo dos artesãos e melhorando a satisfação dos clientes. Além disso, a possibilidade de gerar relatórios e estatísticas permitirá um maior controle financeiro e uma melhor gestão do negócio. A implementação desse software pode trazer diversos benefícios para a empresa de artesanato, tais como aumento da produtividade, melhoria da qualidade dos serviços prestados, redução de custos e aumento da lucratividade. Com isso, concluímos que o investimento em tecnologia é essencial para a competitividade e sucesso do negócio na atualidade.

Para criar um software que atenda às necessidades de uma empresa de artesanato é importante contar com profissionais especializados em desenvolvimento de sistemas. É necessário que esses profissionais tenham conhecimentos em tecnologias específicas para o desenvolvimento de softwares empresariais e que sejam capazes de entender as necessidades do negócio. Além disso, é importante que o software seja desenvolvido com base nas melhores práticas do mercado e que seja testado antes da implantação.

## **ANEXOS**

Link com video da gravação da apresentação:

<https://github.com/leobsossai/PI-ADS-GTI>

[Projeto Integrado - Banco de dados e Programação Orientada a Objetos](#)







**RELATÓRIO FINAL DAS  
ATIVIDADES DE EXTENSÃO**

**1. IDENTIDADE DA ATIVIDADE**

**RELATÓRIO:**

**CURSO:** Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação

**MÓDULO:** Desenvolvimento Desktop

**PROFESSOR RESPONSÁVEL:** Sidney Gitcoff Telles

**ESTUDANTE:**

**PERÍODO DE REALIZAÇÃO:** 02/2023 a 04/2023

**2. DESENVOLVIMENTO**

Contextualização

Desafio

Cronograma das Ações

Síntese das Ações

a. Aspectos positivos



b. Dificuldades encontradas

c. Resultados atingidos

d. Sugestões / Outras observações

### 3. EQUIPE DOS ESTUDANTES NO PROJETO

RA 1012023100062	NOME Gabriele Rigobeli da Silva
RA 1012023100143	NOME Lucas Rafael candido da Silva
RA 1012023100221	NOME Tatiane Daciane De Farias Godoy
RA 1012023100269	NOME Felipe Larotonda Barbosa da Silva
RA 1012023100172	NOME Leonardo Bueno Sossai,

**Curso de Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação**

**Módulo Desenvolvimento Desktop**

**Cronograma de Validação - Projeto Integrado**

<b>Unidade Estudo</b>	<b>Participação no Projeto</b>	<b>Data da Validação</b>
Banco de Dados	Construção do banco de dados com MER, DER E Físico.	29/03
Programação Orientada a Objetos	Desenvolvimento as telas e da parte lógica do sistema que conectará com o banco de dados	30/03
<p><b><u>Descrição do Projeto:</u></b> criar um sistema, um módulo reduzido, que seja utilizado em qualquer estabelecimento comercial ou empresarial. Esse sistema deverá contemplar atividades básicas da empresa, como controle de produtos, entrada e saída, controle de vendas, módulos menores que possam ser criados e executados neste trimestre.</p>		

