



COLÉGIO EVANGÉLICO PANAMBI

CURSO TÉCNICO EM INFORMÁTICA

RAFAEL MATEUS DALLABRIDA

SISTEMA DE CONTROLE DE FINANÇAS

E OBJETIVOS FINANCEIROS

PANAMBI

2024

RAFAEL MATEUS DALLABRIDA

SISTEMA DE CONTROLE DE FINANÇAS E OBJETIVOS FINANCEIROS

Projeto de conclusão de curso apresentado como requisito parcial para obtenção do título de Técnico em Informática, do Curso de Técnico em Informática do Colégio Evangélico Panambi – Centro Tecnológico e de Formação Profissional.

Orientadores: Prof. Catana
Prof. Rodrigo Steinhorst

PANAMBI

2024

RAFAEL MATEUS DALLABRIDA

SISTEMA DE CONTROLE DE FINANÇAS E OBJETIVOS FINANCEIROS

Projeto de conclusão de curso apresentado como requisito parcial para obtenção do título de Técnico em Informática, do Curso de Técnico em Informática do Colégio Evangélico Panambi – Centro Tecnológico e de Formação Profissional.

Banca Examinadora

Prof. Rodrigo Steinhorst

Prof. Catana Rahmeier

Prof. Sirlei Adriane Wottrich

PANAMBI

2024

Sumário

INTRODUÇÃO	6
GOALSPLAN.....	7
CARACTERÍSTICAS DO SISTEMA.....	8
1. Funcionalidades:	8
2. Tecnologias utilizadas:.....	8
3. Resultados Esperados	9
MODELO CONCEITUAL.....	10
DICIONÁRIO DE DADOS	12
UTILIZANDO O SISTEMA.....	15
1. Login.....	15
2. Padrão de Cadastro	16
3. Página Inicial	17
4. Lista de Registros.....	18
CONCLUSÃO.....	19
REFERÊNCIAS.....	20
FERRAMENTAS UTILIZADAS.....	22
CÓDIGO DO BANCO DE DADOS	23

Índice de Imagens

Figura 1 - Modelo Conceitual Parte 1	10
Figura 2 - Modelo Conceitual Parte 2	11
Figura 3 - Tela de login	15
Figura 4 - Erro no login.....	15
Figura 5 - Cadastro de usuário.....	16
Figura 6 - Página inicial.....	17
Figura 7 - Lista de Registros	18
Figura 8 - Lista de Pagamentos com checkbox.....	18

INTRODUÇÃO

Este sistema tem como objetivo ajudar o usuário a organizar suas finanças, seus pagamentos e recebimentos, de forma que consiga categorizar seus lançamentos e ter um retorno visual de quanto recurso financeiro lhe falta para atingir seus objetivos.

O sistema é uma aplicação web que conta com páginas de cadastro e listagem de registros intuitivas e de fácil utilização e vem para suprir uma falta de sistemas similares que sejam simplificados para o uso cotidiano. Quando comparado a outros sistemas, este apresenta apenas a inclusão de dados essenciais para organizar finanças pessoais, que não sejam vinculadas à uma empresa, tornando essa organização menos burocrática.

Realizando os cadastros corretamente, o usuário terá fácil acesso ao valor monetário que possui e poderá acompanhar seu progresso em relação aos objetivos que deseja alcançar.

GOALSPLAN

O sistema foi nomeado como “GoalsPlan” unindo duas palavras da língua inglesa: *Goals*, que significa “Objetivos” e *Plan*, “Planejar”. Como o próprio nome sugere, este sistema terá a função de auxiliar seus usuários a fazer o planejamento para alcançar seus objetivos (também chamados de metas), função de muita importância, principalmente nos dias de hoje, pois sempre somos bombardeados por diversos anúncios que nos instigam a cada vez mais desviar o foco de nossos objetivos e adquirir algo que muitas vezes nem se faz necessário.

Ter um controle sobre seus gastos e recebimentos é muito vantajoso para aqueles que o conseguem fazer. Porém, nem sempre é uma tarefa fácil manter um caderno ou uma planilha com essas informações. Podem ser facilmente perdidas e só podem ser acessadas com o caderno em mãos ou tendo o arquivo em seu computador. Então é uma boa ideia recorrer à um sistema online para atender a essa demanda, até mesmo por ter uma resposta mais dinâmica dessas informações.

Os sistemas que existem hoje para facilitar esse processo muitas vezes ou são pagos (e nada baratos) ou são muito complexos de se entender, fazendo a pessoa se desanimar com essa atividade que normalmente já não é a de maior interesse em sua rotina.

A *GoalsPlan* surge como uma solução à estes problemas mencionados, um sistema web de fácil acesso e de complexidade reduzida para tornar o processo de organização de finanças mais dinâmico e menos exaustivo.

CARACTERÍSTICAS DO SISTEMA

1. Funcionalidades:

- a) Cadastro de Usuários: Permite a privacidade de dados, todas as tabelas do banco de dados possuem o usuário como chave, garantindo assim que cada usuário possa ver apenas seus próprios dados;
- b) Página Inicial: Apresenta o saldo total do usuário, bem como uma visão geral de todas as suas metas cadastradas que ainda não foram concluídas, com uma barra de progresso indicando quanto falta para atingi-lás;
- c) Cadastro de Entidades: Garante o cadastro personalizado e pessoal das seguintes entidades por usuário:
 - I. Fornecedores - instituições ou pessoas a quem é feito pagamentos;
 - II. Clientes - instituições ou pessoas das quais o usuário recebe;
 - III. Históricos – finalidades recorrentes dos lançamentos (exemplos: salário, lazer, educação, alimentação, investimento, etc);
 - IV. Metas – objetivos que se deseja alcançar (exemplos: carro, computador, casa, etc);
 - V. Pagamentos: valores que saem do saldo do usuário com destino à algum fornecedor;
 - VI. Recebimentos: valores que acrescentam o saldo do usuário vindos de algum cliente.

2. Tecnologias utilizadas:

- a) HTML - usado para estruturar as páginas web;
- b) CSS – usado para deixar as páginas visualmente mais agradáveis e intuitivas;
- c) PHP – linguagem de programação usada para manipular os dados dos campos HTML e fazer a conexão com o banco de dados;

- d) JavaScript – linguagem de programação usada para fazer interações dinâmicas entre o sistema e o usuário;
- e) MariaDB – banco de dados de código aberto usado para armazenar e consultar os dados enviados pelas páginas web;
- f) SQL – linguagem de consulta usado para consultar, inserir, alterar e excluir dados do banco de dados.

3. Resultados Esperados:

- a) Auxiliar o usuário a registrar e organizar suas finanças;
- b) Manter o foco do usuário em relação às suas metas cadastradas apresentando sua progressão de forma visual;
- c) Tornar o processo de organização de finanças menos desgastante.

MODELO CONCEITUAL

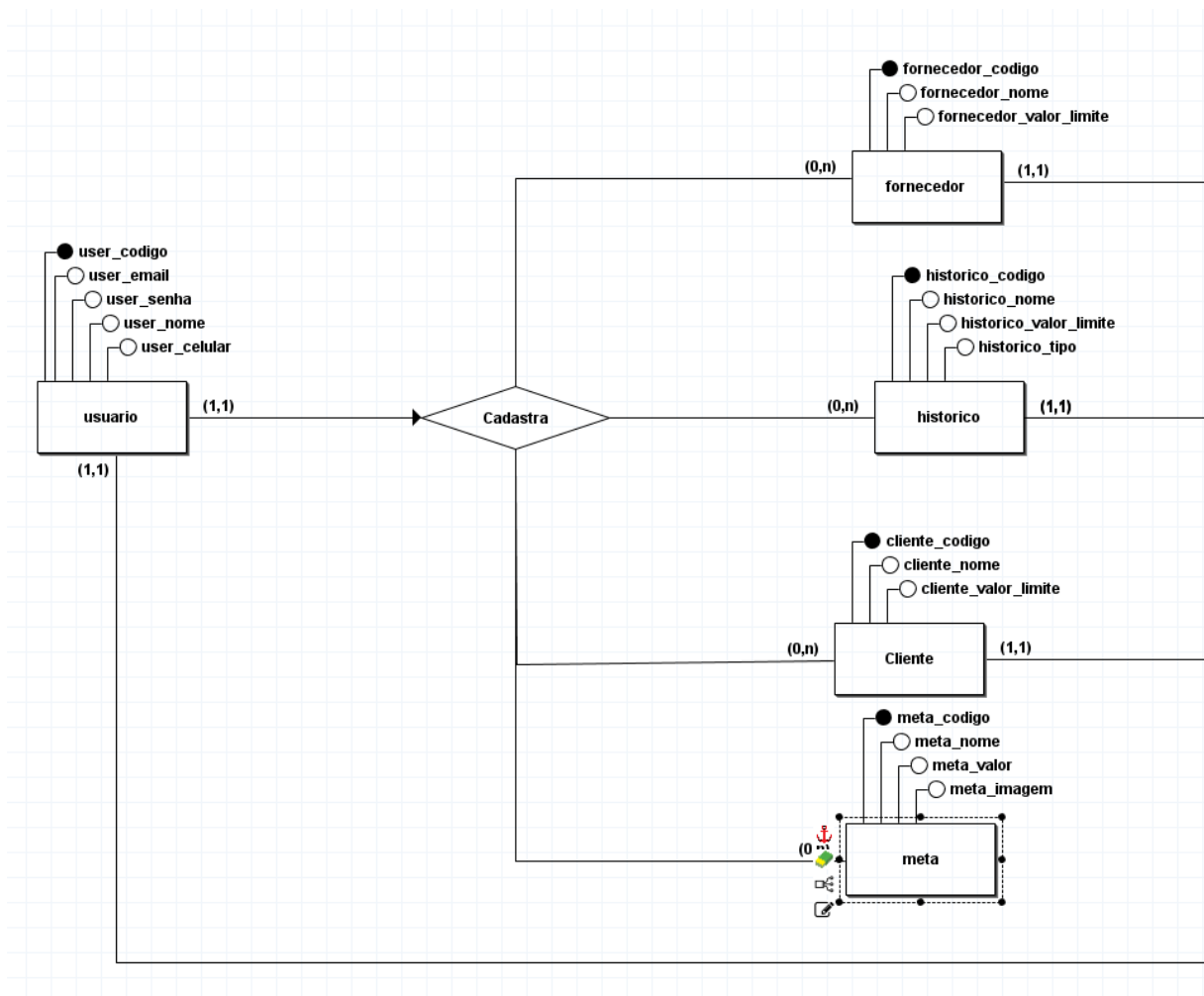


Figura 1 - Modelo Conceitual Parte 1

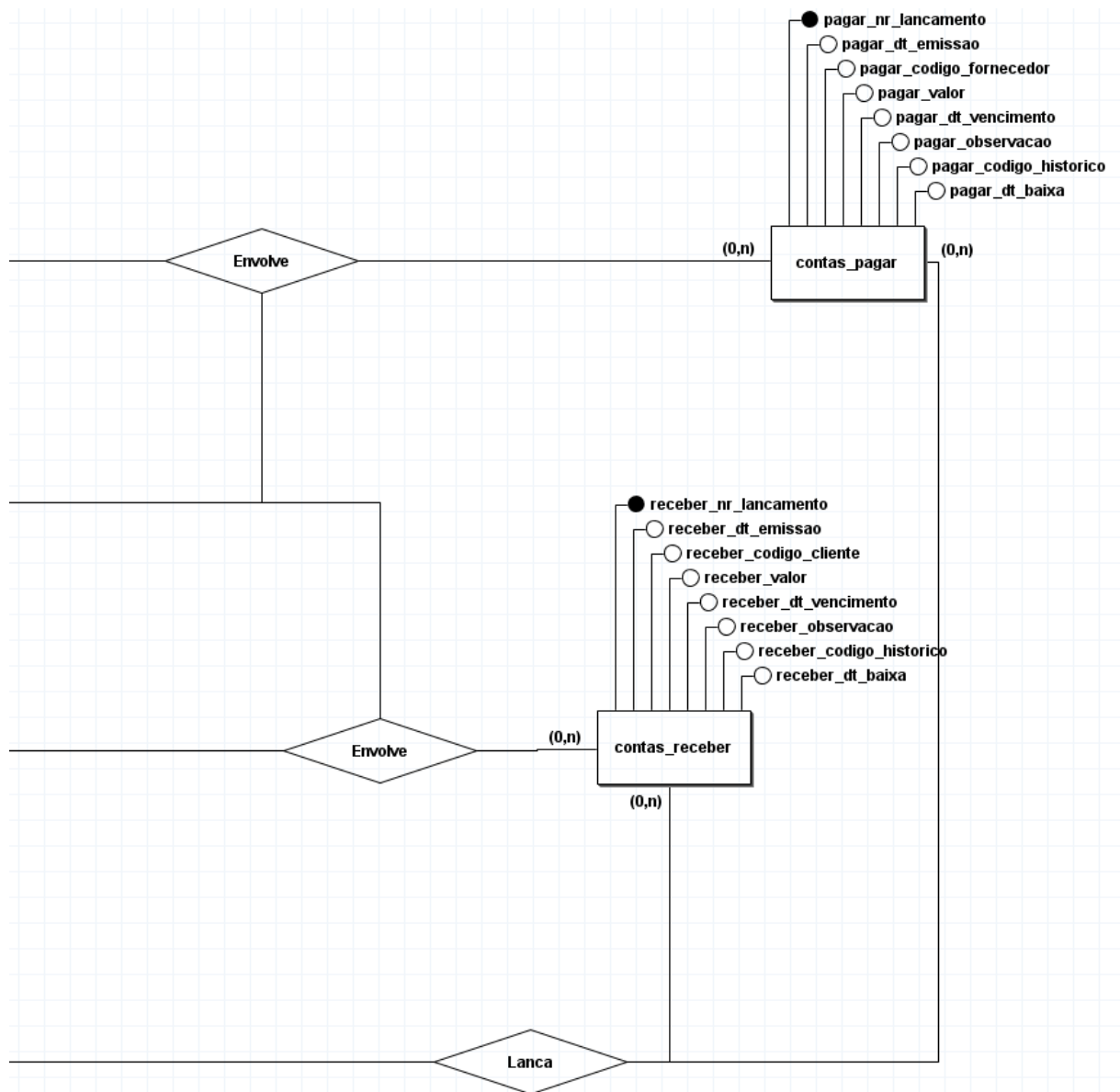


Figura 2 - Modelo Conceitual Parte 2

DICIONÁRIO DE DADOS

NOME	usuario			
DESCRIÇÃO	Tabela de usuários			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
usuario_codigo	INT		S	Chave Principal
usuario_email	VARCHAR	255	S	
usuario_username	VARCHAR	16	S	
usuario_senha	VARCHAR	16	S	
usuario_nome	VARCHAR	255	S	
usuario_celular	INT		N	

NOME	cliente			
DESCRIÇÃO	Tabela de clientes			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
cliente_usuario	INT		S	Chave estrangeira relacionada à tabela usuario
cliente_codigo	INT		S	Chave Principal
cliente_nome	VARCHAR	255	S	
cliente_valor_limite	DECIMAL	10,2	N	

NOME	fornecedor			
DESCRIÇÃO	Tabela de fornecedores			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
fornecedor_usuario	INT		S	Chave estrangeira relacionada à tabela usuario
fornecedor_codigo	INT		S	Chave Principal
fornecedor_nome	VARCHAR	255	S	
fornecedor_valor_limite	DECIMAL	10,2	N	

NOME	tipo_historico			
DESCRIÇÃO	Tabela de tipos de histórico			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
tipo_codigo	INT		S	Chave Principal
tipo_nome	VARCHAR	255	S	

NOME	historico			
DESCRIÇÃO	Tabela de históricos, históricos são as categorias dos lançamentos			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
historico_usuario	INT		S	Chave estrangeira relacionada à tabela usuario
historico_codigo	INT		S	Chave Principal
historico_nome	VARCHAR	255	S	
historico_valor_limite	DECIMAL	10,2	N	
historico_tipo	INT		S	Chave estrangeira relacionada à tabela tipo_historico

NOME	meta			
DESCRIÇÃO	Tabela de metas, objetivos			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
meta_usuario	INT		S	Chave estrangeira relacionada à tabela usuario
meta_codigo	INT		S	Chave Principal
meta_nome	VARCHAR	255	S	
meta_valor	DECIMAL	10,2	N	
meta_imagem	VARCHAR	255	N	

NOME	contas_pagar			
DESCRIÇÃO	Tabela de pagamentos			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
pagar_usuario	INT		S	Chave estrangeira relacionada à tabela usuario
pagar_nr_lancamento	INT		S	Chave Principal
pagar_dt_emissao	DATE		S	
pagar_codigo_fornecedor	INT		S	Chave estrangeira relacionada à tabela fornecedor
pagar_valor	DECIMAL	10,2	S	
pagar_dt_vencimento	DATE		S	
pagar_observacao	VARCHAR	255	N	
pagar_codigo_historico	INT		S	Chave estrangeira relacionada à tabela historico
pagar_dt_baixa	DATE		S	

NOME	contas_receber			
DESCRIÇÃO	Tabela de recebimentos			
CAMPO	TIPO DE DADO	TAMANHO	REQUERIDO	COMENTÁRIO
receber_usuario	INT		S	Chave estrangeira relacionada à tabela usuario
receber_nr_lancamento	INT		S	Chave Principal
receber_dt_emissao	DATE		S	
receber_codigo_cliente	INT		S	Chave estrangeira relacionada à tabela cliente
receber_valor	DECIMAL	10,2	S	
receber_dt_vencimento	DATE		S	
receber_observacao	VARCHAR	255	N	
receber_codigo_historico	INT		S	Chave estrangeira relacionada à tabela historico
receber_dt_baixa	DATE		S	

UTILIZANDO O SISTEMA

1. Login

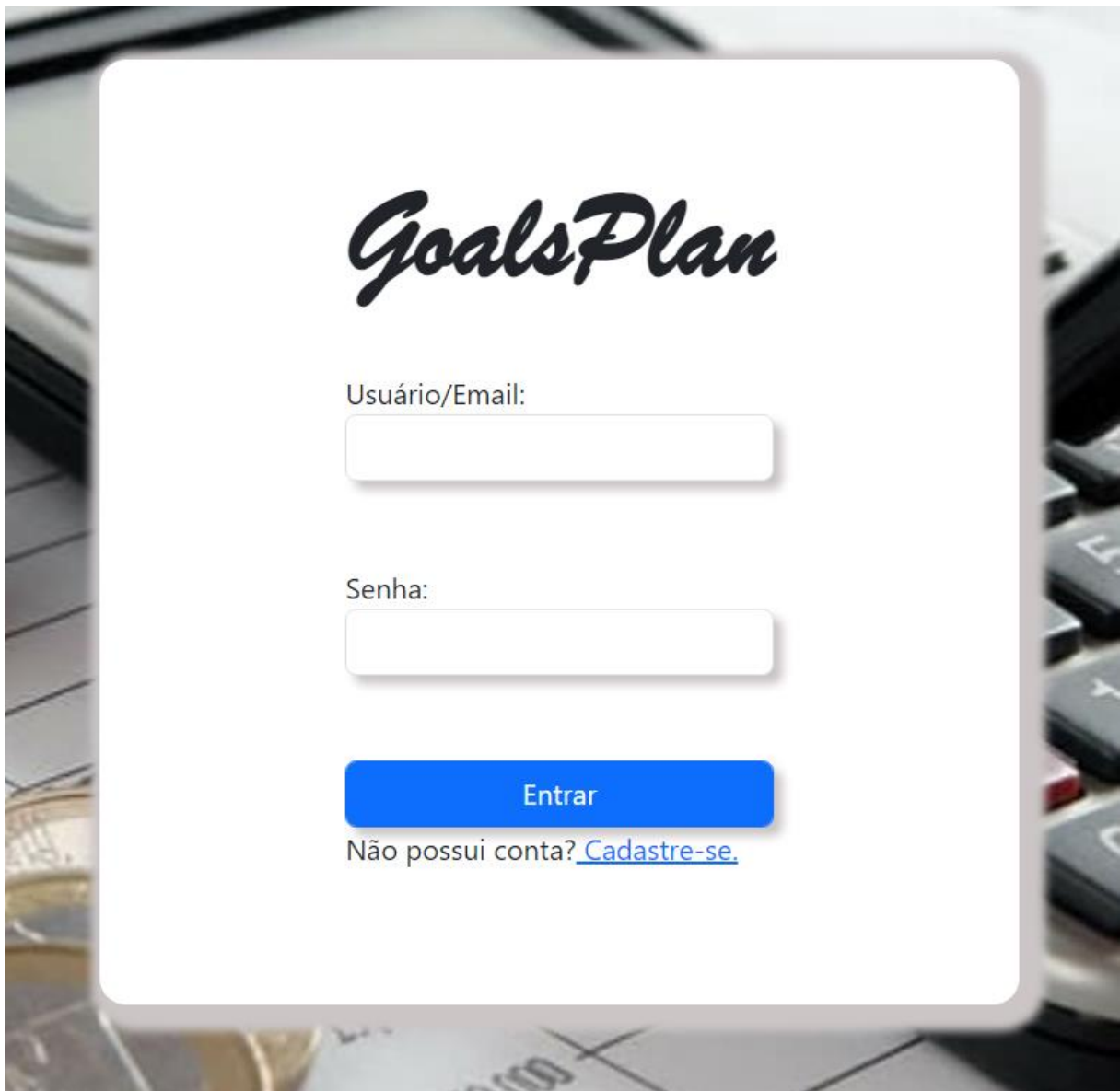


Figura 3 - Tela de login

O login é feito quando já se tem um cadastro registrado no banco de dados, caso tente entrar com email/senha inexistente é apresentado um alerta do erro que ocorreu.

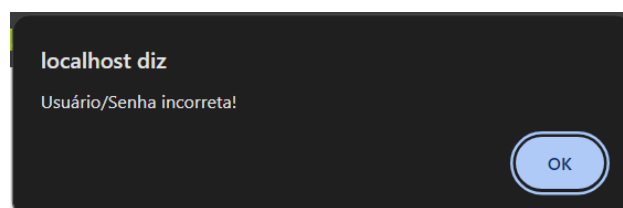


Figura 4 - Erro no login

2. Padrão de Cadastro



Cadastro de usuário

Nome

E-mail

Usuário

Celular

Senha

Salvar

Figura 5 - Cadastro de usuário

A página acima apresenta o padrão de todas as páginas de cadastro/edição de registros, onde usa *inputs* para receber o valor de todos os campos e os salva pelo botão “Salvar”.

3. Página Inicial

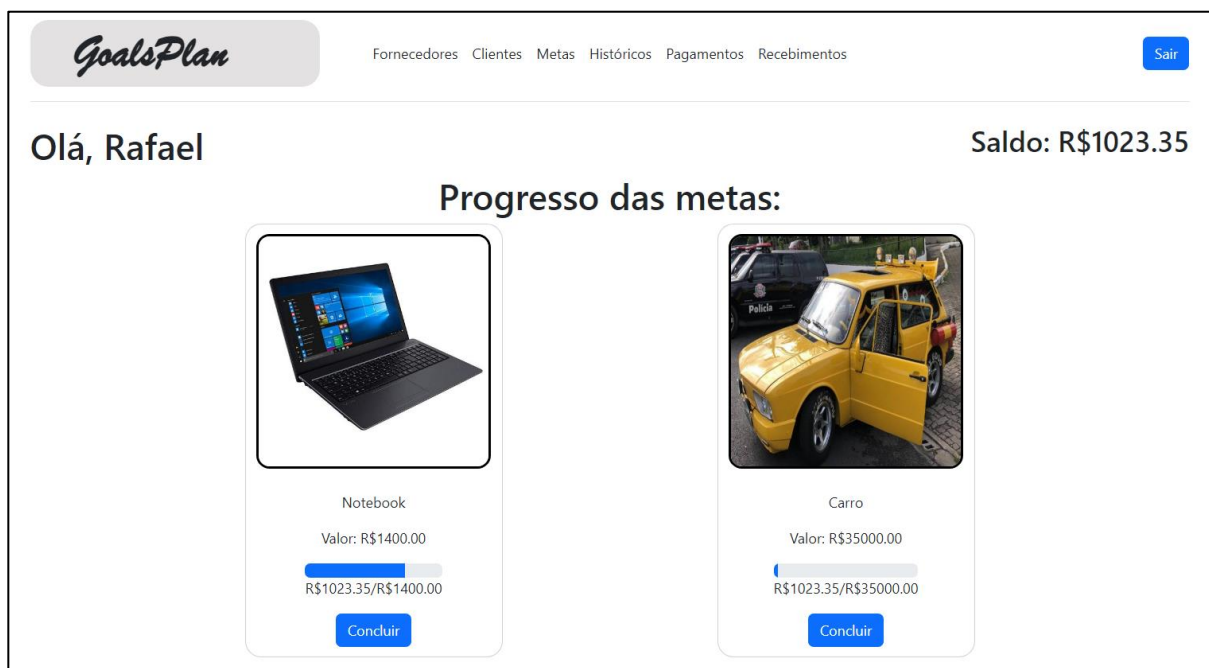
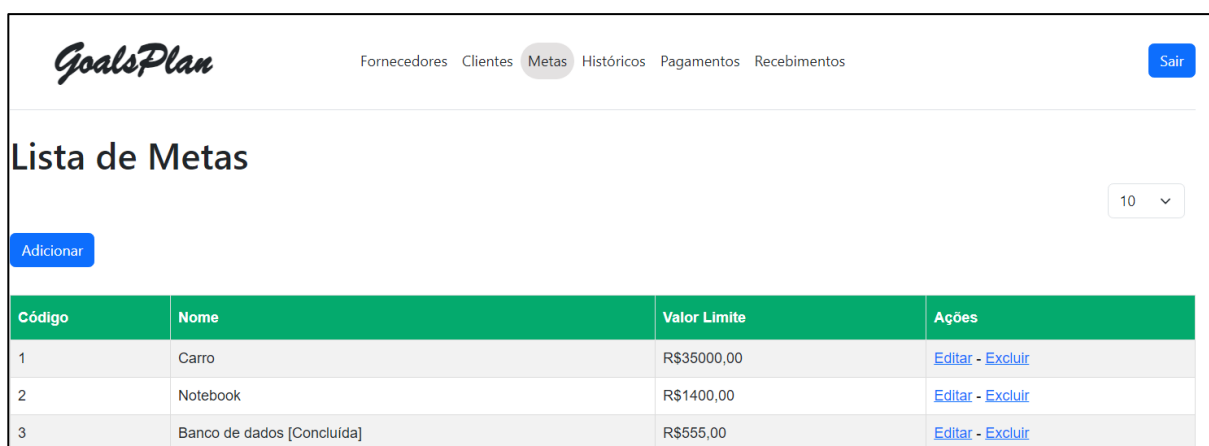


Figura 6 - Página inicial

Na página inicial já temos acesso ao cabeçalho de navegação que se repete em todas as demais páginas, é feita uma saudação ao usuário, é lhe apresentado o saldo que possui, e também aparecem os cartões das metas cadastradas e não concluídas. Nas metas é possível identificar o valor das mesmas junto com uma barra de progresso comparando o valor da meta com o saldo atual do usuário.

Ao concluir a meta, basta clicar no botão “Concluir” da respectiva meta que o cartão será removido da página inicial;

4. Lista de Registros



Código	Nome	Valor Limite	Ações
1	Carro	R\$35000,00	Editar - Excluir
2	Notebook	R\$1400,00	Editar - Excluir
3	Banco de dados [Concluída]	R\$555,00	Editar - Excluir

Em cada uma das páginas presentes no cabeçalho, há uma lista com todos os

Figura 7 - Lista de Registros

registros da respectiva entidade.

O link do cabeçalho fica com cor alterada quando se está na página.

Nestas listas há o botão “Adicionar”, para acessar a página de cadastro da entidade, há um *select* onde pode-se escolher a quantidade de registros que vão aparecer em uma página da lista (caso haja mais registros do que o valor do *select* é feito paginação).

Dentro da lista, além dos dados da entidade, há também os links de ações para editar ou excluir o registro.

As únicas listas diferentes são as de Pagamentos e Recebimentos, onde, além do conteúdo que há nas demais, existe um checkbox para cada registro e um botão “Baixar”, selecionando os registros que se quer e clicando nesse botão o lançamento passa a ser contabilizado no saldo do usuário.



Seleção	Número Lançamento	Data de Vencimento	Fornecedor	Valor	Histórico	Observação	Data de Emissão	Data de Baixa	Ações
	1	13/06/2024	fornecedor	R\$50,70	lazer	atatata 2		15/06/2024	Editar - Excluir
	2	14/06/2024	fornecedor	R\$2,50	lazer			19/06/2024	Editar - Excluir
<input checked="" type="checkbox"/>	9	28/06/2024	fornecedor	R\$100,00	lazer	Teste	27/06/2024		Editar - Excluir

Figura 8 - Lista de Pagamentos com checkbox

CONCLUSÃO

O sistema atende à sua proposta de ser uma ferramenta simplificada de organização de finanças pessoais, possui baixa complexidade e apresenta a útil funcionalidade de poder enxergar o progresso dos objetivos logo ao acessá-lo. Acredito que seja uma ferramenta funcional que pode facilmente atender às necessidades daqueles que a utilizarem para desfrutar dos benefícios de fazer um planejamento de futuros gastos.

Tirar as ideias da cabeça e levá-las a uma aplicação, bem como fazer melhorias posteriormente passaram a ser uma possibilidade plausível com a conquista de uma boa base de aprendizado advinda dos 3 anos e meio de ensino técnico e das experiências obtidas durante as tentativas e erros ao desenvolver esta e outras aplicações.

Durante a realização deste projeto de conclusão de curso, fui percebendo a importância que um bom técnico de informática tem na resolução de problemas, tanto do cotidiano, quanto na área profissional. Este é alguém que deve estar preparado com seu acervo de tecnologias (mas ainda assim disposto a aceitar e entender outras) para solucionar os problemas que lhe são apresentados com eficiência e buscando inovação de processos.

Tive que enfrentar o desafio de ir atrás de ferramentas novas, das quais eu não tinha conhecimento, para otimizar alguns pontos durante o desenvolvimento do sistema, e, em relação as tecnologias que eu conhecia, entender como aplicá-las para cada situação que aparecia até chegar à finalização do projeto. Com ele também experimentei a importância de organizar bem as ideias e o tempo, para fazer um desenvolvimento produtivo e ter um bom resultado.

REFERÊNCIAS

How do I get the root directory path of a PHP project? Disponível em: <<https://stackoverflow.com/questions/8668776/get-root-directory-path-of-a-php-project>>. Acesso em: 09/05/2024.

XAMPP MySQL shutdown unexpectedly – How to Fix It. Disponível em: <<https://kinsta.com/knowledgebase/xampp-mysql-shutdown-unexpectedly/>>. Acesso em: 21/05/2024.

Erro - Mensagem de Login. Disponível em: <<https://cursos.alura.com.br/forum/topico-erro-mensagem-de-login-63155>>. Acesso em: 21/05/2024.

hash - Manual. Disponível em: <https://www.php.net/manual/pt_BR/function.hash.php>. Acesso em: 22/05/2024.

PDOStatement::bindParam - Manual. Disponível em: <https://www.php.net/manual/pt_BR/pdostatement.bindparam.php>. Acesso em: 01/06/2024.

Basic Branching and Merging - Git Branching. Disponível em: <<https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>>. Acesso em: 17/06/2024.

How To Get The Current URL Using JavaScript. Disponível em: <https://www.w3schools.com/howto/howto_js_get_url.asp>. Acesso em: 18/06/2024.

Integração com GitHub: Implantação. Disponível em: <<https://guia-de-uso.tiflux.com/integracoes/projetos/github-1/implantacao>>. Acesso em: 18/06/2024.

Pagination - Bootstrap. Disponível em:

<<https://getbootstrap.com.br/docs/4.1/components/pagination/>>. Acesso em: 20/06/2024.

Como pegar a URL anterior em JavaScript. Disponível em:

<<https://horadecodar.com.br/como-pegar-a-url-anterior-em-javascript/>>. Acesso em: 21/06/2024.

Card - Bootstrap. Disponível em:

<<https://getbootstrap.com.br/docs/4.1/components/card/>>. Acesso em: 24/06/2024.

CSS: Guia do Flexbox. Disponível em: <<https://www.alura.com.br/artigos/css-guia-do-flexbox>>. Acesso em: 26/06/2024/2024.

FERRAMENTAS UTILIZADAS

1. Visual Studio Code – programa de edição de códigos onde há indentação, tabulação, destaque de comandos e muitas outras funcionalidades, principalmente com inclusão de extensões, para facilitar o desenvolvimento;
2. Git/GitHub – ferramenta usada para gerenciar o diretório do projeto, podendo fazer versionamento do mesmo e subir o diretório para um repositório em nuvem. Link para repositório: <<https://github.com/rafaelmdgh/pcc>>;
3. HeideSQL -Sistema de Gerenciamento de Banco de Dados (SGBD) usado para acessar o banco de dados, bem como fazer alterações no mesmo de forma gráfica;
4. Navicat – programa usado para fazer modelos de banco de dados, conceitual, lógico e físico;
5. Br Modelo – programa usado para fazer modelo conceitual de banco de dados;
6. Xampp – facilita a instalação de um servidor local Apache para podermos fazer a conexão entre páginas web e banco de dados;

CÓDIGO DO BANCO DE DADOS

```
CREATE DATABASE finanças_pessoais;  
USE finanças_pessoais;
```

```
CREATE TABLE usuario (  
    usuario_codigo int NOT NULL auto_increment,  
    usuario_email varchar(255) NOT NULL,  
    usuario_username varchar (16) NOT NULL,  
    usuario_senha varchar(16) NOT NULL,  
    usuario_nome varchar(255) NOT NULL,  
    usuario_celular int NULL,  
    PRIMARY KEY (usuario_codigo)  
) ENGINE=MYISAM DEFAULT CHARSET=utf8  
AUTO_INCREMENT=1;
```

```
CREATE TABLE cliente (  
    cliente_usuario int NOT NULL,  
    cliente_codigo int NOT NULL auto_increment,  
    cliente_nome varchar(255) NOT NULL,  
    cliente_valor_limite decimal(10,2) NULL,  
    PRIMARY KEY (cliente_usuario, cliente_codigo)  
) ENGINE=MYISAM DEFAULT CHARSET=utf8  
AUTO_INCREMENT=1;
```

```
CREATE TABLE fornecedor (  
    fornecedor_usuario int NOT NULL,  
    fornecedor_codigo int NOT NULL auto_increment,  
    fornecedor_nome varchar(255) NOT NULL,  
    fornecedor_valor_limite decimal(10,2) NULL,  
    PRIMARY KEY (fornecedor_usuario, fornecedor_codigo)  
) ENGINE=MYISAM DEFAULT CHARSET=utf8  
AUTO_INCREMENT=1;
```

```
CREATE TABLE tipo_historico (  
    tipo_codigo int NOT NULL auto_increment,  
    tipo_nome varchar(255) NOT NULL,  
    PRIMARY KEY (tipo_codigo)  
) ENGINE=MYISAM DEFAULT CHARSET=utf8  
AUTO_INCREMENT=1;
```

```
CREATE TABLE historico (  
    historico_usuario int NOT NULL,  
    historico_codigo int NOT NULL auto_increment,  
    historico_nome varchar(255) NOT NULL,  
    historico_valor_limite decimal(10,2) NULL,  
    historico_tipo INT NOT NULL,  
    PRIMARY KEY (historico_usuario, historico_codigo)
```

```
) ENGINE=MYISAM DEFAULT CHARSET=utf8  
AUTO_INCREMENT=1;
```

```
CREATE TABLE meta (  
    meta_usuario int NOT NULL,  
    meta_codigo int NOT NULL auto_increment,  
    meta_nome varchar(255) NOT NULL,  
    meta_valor decimal(10,2) NULL,  
    meta_imagem varchar(255) NULL,  
    PRIMARY KEY (meta_usuario, meta_codigo)  
) ENGINE=MYISAM DEFAULT CHARSET=utf8  
AUTO_INCREMENT=1;
```

```
CREATE TABLE contas_pagar (  
    pagar_usuario int NOT NULL,  
    pagar_nr_lancamento int NOT NULL auto_increment,  
    pagar_dt_emissao date NOT NULL,  
    pagar_codigo_fornecedor int NOT NULL,
```



```

pagar_valor decimal(10,2) NOT NULL,
pagar_dt_vencimento date NOT NULL,
pagar_observacao varchar(255) NULL,
pagar_codigo_historico int NOT NULL,
pagar_dt_baixa date NOT NULL,
PRIMARY KEY (pagar_usuario, pagar_nr_lancamento)
) ENGINE=MYISAM DEFAULT CHARSET=utf8
AUTO_INCREMENT=1;

```

```

CREATE TABLE contas_receber (
receber_usuario int NOT NULL,
receber_nr_lancamento int NOT NULL auto_increment,
receber_dt_emissao date NOT NULL,
receber_codigo_cliente int NOT NULL,
receber_valor decimal(10,2) NOT NULL,
receber_dt_vencimento date NOT NULL,
receber_observacao varchar(255) NULL,
receber_codigo_historico int NOT NULL,
receber_dt_baixa date NOT NULL,
PRIMARY KEY (receber_usuario, receber_nr_lancamento)
) ENGINE=MYISAM DEFAULT CHARSET=utf8
AUTO_INCREMENT=1;

```

```

ALTER TABLE cliente ADD CONSTRAINT cliente_usuario_codigo
FOREIGN KEY (cliente_usuario) REFERENCES usuario
(usuario_codigo) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE fornecedor ADD CONSTRAINT
fornecedor_usuario_codigo FOREIGN KEY (fornecedor_usuario)
REFERENCES usuario (usuario_codigo) ON DELETE CASCADE
ON UPDATE CASCADE;
ALTER TABLE tipo_historico ADD CONSTRAINT
tipo_historico_codigo FOREIGN KEY (tipo_codigo) REFERENCES
historico (historico_tipo) ON DELETE RESTRICT ON UPDATE
CASCADE;

```

```

ALTER TABLE historico ADD CONSTRAINT
historico_usuario_codigo FOREIGN KEY (historico_usuario)
REFERENCES usuario (usuario_codigo) ON UPDATE CASCADE
ON DELETE CASCADE;
ALTER TABLE meta ADD CONSTRAINT meta_usuario_codigo
FOREIGN KEY (meta_usuario) REFERENCES usuario
(usuario_codigo) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE contas_pagar ADD CONSTRAINT
contas_pagar_usuario_codigo FOREIGN KEY (pagar_usuario)
REFERENCES usuario (usuario_codigo) ON DELETE CASCADE
ON UPDATE CASCADE;
ALTER TABLE contas_pagar ADD CONSTRAINT
contas_pagar_fornecedor_codigo FOREIGN KEY
(pagar_codigo_fornecedor) REFERENCES fornecedor
(fornecedor_codigo) ON DELETE RESTRICT ON UPDATE
CASCADE;
ALTER TABLE contas_pagar ADD CONSTRAINT
contas_pagar_historico_codigo FOREIGN KEY
(pagar_codigo_historico) REFERENCES historico
(historico_codigo) ON DELETE RESTRICT ON UPDATE
CASCADE;
ALTER TABLE contas_receber ADD CONSTRAINT
contas_receber_usuario_codigo FOREIGN KEY (receber_usuario)
REFERENCES usuario (usuario_codigo) ON DELETE CASCADE
ON UPDATE CASCADE;
ALTER TABLE contas_receber ADD CONSTRAINT
contas_receber_cliente_codigo FOREIGN KEY
(receber_codigo_cliente) REFERENCES cliente (cliente_codigo)
ON DELETE RESTRICT ON UPDATE CASCADE;
ALTER TABLE contas_receber ADD CONSTRAINT
contas_receber_historico_codigo FOREIGN KEY
(receber_codigo_historico) REFERENCES historico
(historico_codigo) ON DELETE RESTRICT ON UPDATE
CASCADE;

```

```
insert into tipo_historico (tipo_nome) values ("Receitas");  
insert into tipo_historico (tipo_nome) values ("Despesas Fixas");  
insert into tipo_historico (tipo_nome) values ("Despesas Diversas");  
insert into tipo_historico (tipo_nome) values ("Investimentos");
```