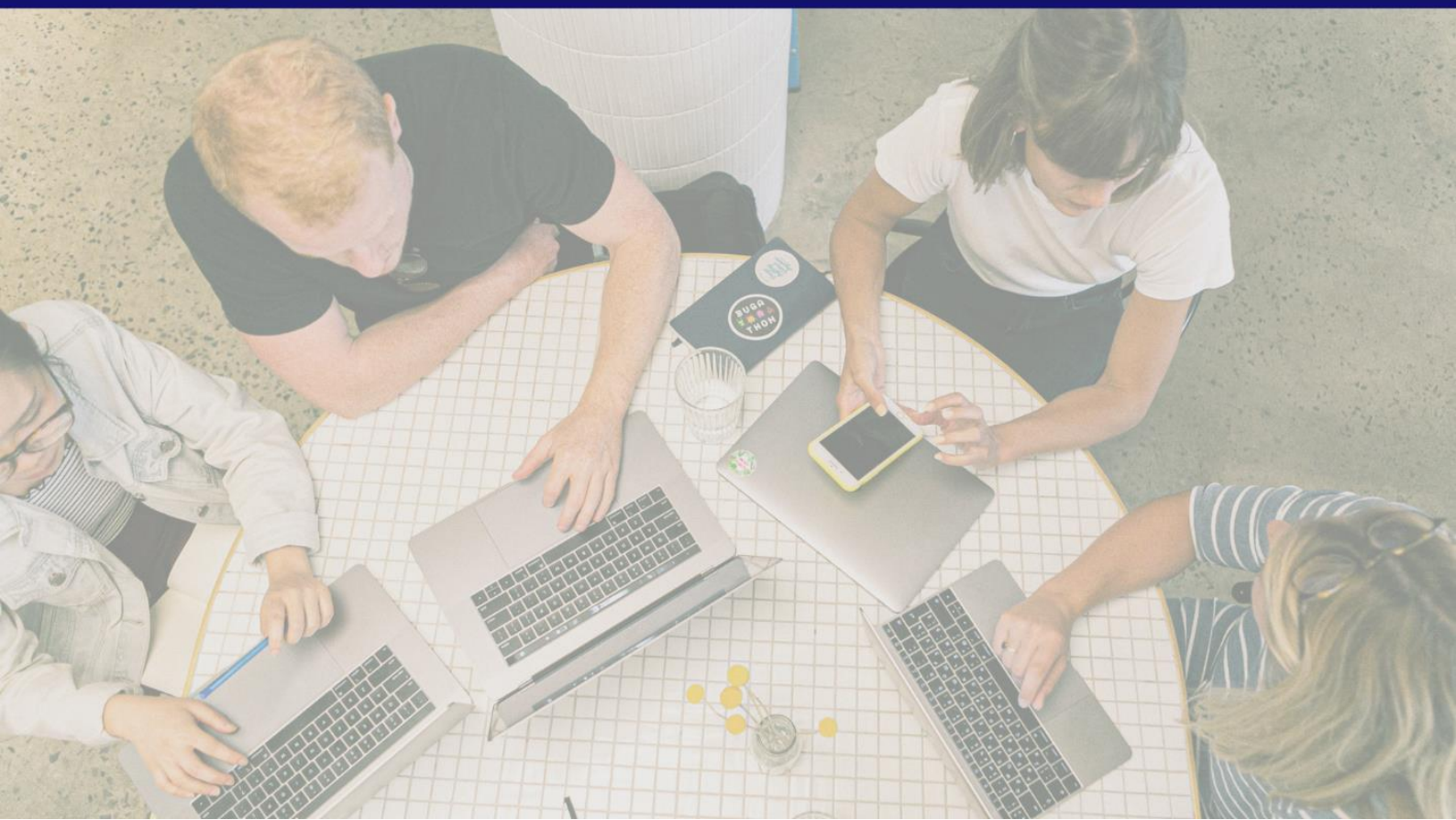
UNIFEOB
ESCOLA DE NEGÓCIOS

2023

PROJETO INTEGRADO



UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

A.D.S. E CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

IOT DATA STREAMER

UNIFEOB

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2023

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

A.D.S. E CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

IOT DATA STREAMER

UNIFEOB

MÓDULO MODELAGEM E DESENVOLVIMENTO DE SISTEMAS

Gestão Financeira – Profa. Renata Elizabeth de Alencar Marcondes

Programação Orientada a Objeto – Prof. Nivaldo Andrade

Lógica de Programação – Prof. Marcelo Ciacco de Almeida

Modelagem de Dados – Prof. Max Streicher Vallim

Projeto de Modelagem e Desenvolvimento de Sistemas – Profª. Mariângela Martimbianco Santos

Estudantes:

Augusto B Sossai, RA 23000707
Erick Simo de Souza Fernandes , RA 23000562
Nathan Heller Torati Salmaso, RA 23000428
Samuel Pontes Portela, RA 23000540
Waldrick Wolak, RA 23000668

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2023

SUMÁRIO

ISSN 1983-6767

1. INTRODUÇÃO

O objetivo principal desse projeto é desenvolver um sistema eficiente e acessível que possa ser aplicado em diferentes ambientes para aprimorar a segurança e a vigilância. A união do Arduino, uma plataforma eletrônica de código aberto, com a biblioteca OpenCV, que fornece recursos avançados de visão computacional, permite a criação de um sistema de segurança inteligente e personalizado.

As câmeras utilizadas no sistema são conectadas ao Arduino, que atua como o cérebro do sistema, gerenciando as funcionalidades e interações. A biblioteca OpenCV é responsável pelo processamento de imagens em tempo real, permitindo a detecção de movimentos, reconhecimento facial, análise de padrões e outras tarefas relacionadas à segurança.

A UNIFEOB está empenhada em desenvolver esse projeto de maneira interdisciplinar, envolvendo professores e estudantes de diversas áreas, como Engenharia, Ciência da Computação e Sistemas de Informação. Essa abordagem colaborativa permite explorar diferentes perspectivas e expertise, enriquecendo a qualidade e a eficiência do sistema de segurança.

O título do PI é “**IoT Data Streamer**", O objetivo do IoT Data Streamer é implementar uma aplicação desktop responsável por gerenciar os dados coletados e trabalhados no "App" mobile, que são provenientes de dispositivos de Internet das Coisas (IoT). A aplicação deve simular a importação de dados gerados, tanto em formato TXT como CSV, e popular banco de dados. Posteriormente, os dados populados serão utilizados para consultas com filtros, dias e datas específicas, que serão exibidos em visualizações adequadas.

Funcionalidades Principais

Simulação de Importação de Dados:

A aplicação permitirá a simulação da importação de dados gerados previamente, podendo ser em formato TXT ou CSV. Esses dados serão processados e armazenados em um banco de dados para uso posterior.

Armazenamento em Banco de Dados:

Os dados importados serão persistidos em um banco de dados, possibilitando a recuperação rápida e organizada dessas informações.

Consulta e Filtros:

A partir dos dados armazenados no banco, o sistema permitirá consultas com filtros específicos, como datas, dias da semana, ou outros critérios definidos pelo usuário.

Visualizações:

As consultas realizadas serão exibidas em visualizações intuitivas e informativas, como gráficos, tabelas ou outros formatos adequados, para facilitar a análise dos dados coletados.

Cadastro Manual:

Além da importação de dados, o sistema permitirá que o usuário possa cadastrar novos dados manualmente, caso necessário.

Recursos Avançados:

A aplicação oferecerá recursos avançados, como a possibilidade de exibir alertas com base em padrões identificados nos dados, bem como enviar notificações por e-mail para o gerenciador quando certas condições forem atendidas.

Login e Segurança:

O sistema contará com um formato de login para garantir a segurança dos dados e permitir acesso apenas a usuários autorizados.

Entrega Mínima:

A aplicação será capaz de simular a importação de dados por meio de arquivos CSV ou TXT, salvando esses dados em um banco de dados. Os dados poderão ser recuperados e exibidos em uma interface gráfica para análise básica. Nesta versão inicial, recursos avançados, como alertas e notificações, poderão ser implementados posteriormente em versões mais avançadas do sistema.

ISSN 1983-6767

2. DESCRIÇÃO DA EMPRESA

UNIFEOB, CNPJ: 69.546.754/0001-48, Avenida Doutor Otávio da Silva Bastos, 2439, São João da Boa Vista - SP,

A Unifeob foi fundada em 1965 com a visão de oferecer educação de qualidade em São João da Boa Vista, promovendo o desenvolvimento da região. Hoje, é uma referência nacional e global. A universidade adota uma abordagem interdisciplinar, envolvendo professores e estudantes de diversas áreas, como Engenharia e Ciência da Computação, para aprimorar o sistema de segurança. Com milhares de estudantes formados anualmente e uma comunidade acolhedora, a Unifeob é um centro de conhecimento e crescimento pessoal.

ISSN 1983-6767

3. PROJETO INTEGRADO

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

A Programação Orientada a Objetos (POO) é um paradigma essencial na informática, onde o código é estruturado em objetos conectados, proporcionando uma modelagem mais próxima da realidade e aumentando a reutilização do código.

Segundo Henrique (2023):

> Como a maioria das atividades que fazemos no dia a dia, programar também possui modos diferentes de se fazer. Esses modos são chamados de paradigmas de programação e, entre eles, estão a programação orientada a objetos (POO) e a programação estruturada.

No projeto foi usado JavaScript e PHP como linguagem de programação, junto dela usamos a orientação a objeto para executar as funções do nosso projeto.

Esse trecho de código utiliza o método fetch para enviar uma solicitação HTTP para o recurso backend/pesquisar-placas.php no servidor. Essa solicitação é do tipo POST e inclui um cabeçalho com o tipo de conteúdo application/x-www-form-urlencoded.

Após a resposta da solicitação estar disponível, o código utiliza a função then para lidar com a resposta.

Dentro da função que lida com a resposta, verifica-se o conteúdo da resposta JSON, especificamente a propriedade data.RETORNO.

Se data.RETORNO for igual a "ERRO", o código cria uma mensagem de alerta de aviso, informando que não foram encontrados registros, e insere essa mensagem no elemento com o ID "retorno" no HTML. A mensagem de alerta inclui um ícone de exclamação e um botão para fechar o alerta.

Se data.RETORNO não for igual a "ERRO", o código não exibe o alerta de erro. Presumivelmente, o código continua com alguma outra ação que não está incluída neste trecho.

Tela 1- Função do Código

```
const Listar = () =>{
  fetch(`backend/pesquisar-placas.php`, {
    credentials: 'same-origin',
    method: 'POST',
    headers: {
        'Content-type': 'application/x-www-form-urlencoded'
    }
}).then(function (response) {
    response.json().then(data => {
        if(data.RETORNO == "ERRO"){
            // Aqui trata o erro da requisição
            let alerta = `
            <div class="alert alert-warning alert-dismissible fade show" role="alert">
                <strong><i class="fas fa-exclamation-triangle"></i> Atenção!</strong> Não foi possivel encontrar registros!
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&times;</span>
                </button>
            </div>
            `
            $("#retorno").html(alerta)
        }else{
            // Aqui trata o sucesso
```

Fonte : Autores

Esse outro trecho inclui um arquivo chamado "conexao.php", que provavelmente contém informações de conexão com o banco de dados.

Define uma consulta SQL que seleciona todos os registros da tabela "usuario" no banco de dados.

Prepara a consulta SQL usando a conexão do banco de dados.

Executa a consulta SQL.

Coleta todos os dados do resultado da consulta no formato de array associativo usando fetchAll com a constante PDO::FETCH_ASSOC.

Verifica se a quantidade de registros obtidos, representada por $dados, é maior que zero. Se houver registros no banco de dados:

Codifica esses dados em formato JSON usando json_encode, com a opção JSON_UNESCAPED_UNICODE para garantir que caracteres especiais sejam mantidos como estão.

Se não houver registros (ou seja, $dados é igual a zero):

Cria um array associativo chamado $retorno com duas chaves: "RETORNO" com o valor "ERRO" e "MENSAGEM" com a mensagem "REGISTROS NÃO ENCONTRADOS!".

Codifica esse array em formato JSON da mesma forma que no passo anterior.

Imprime o JSON resultante na saída, que é a resposta da requisição feita ao arquivo PHP. Isso pode ser consumido por JavaScript ou outro código do lado do cliente.

Fecha a conexão com o banco de dados.

Em resumo, esse código PHP faz uma consulta ao banco de dados para buscar registros da tabela "usuario". Se encontrar registros, ele os converte em JSON e os envia como resposta. Se não encontrar registros, ele envia uma mensagem de erro em JSON como resposta. Esse

código é comumente usado para fornecer dados a partir de um servidor web para uma aplicação do lado do cliente, como um aplicativo web ou móvel. (Anexo TELA: 2)

3.1.1 CLASSES E OBJETOS

Em nosso projeto na parte do PHP criamos várias classes que se comunicam entre si passando as informações necessárias para os INSERTS, SELECT, UPDATE e DELETE.

Como no exemplo, as telas: (Anexo TELA: 3, TELA: 4)

Classe UsuarioService:

A classe UsuarioService é um modelo ou plano que define como as operações relacionadas a usuários devem ser executadas.

Ela encapsula a lógica para atualizar informações de um usuário no banco de dados.

A classe possui uma propriedade privada $conexao, que é uma instância de conexão com o banco de dados. Ela é passada como um parâmetro no construtor da classe.

A classe possui um método público atualizarUsuario. Esse método aceita os parâmetros necessários para atualizar um usuário e executa a operação de atualização no banco de dados. Ele também retorna uma resposta em formato JSON.

Objeto da classe UsuarioService:

A linha $usuarioService = new UsuarioService($conexao); cria uma instância (objeto) da classe UsuarioService. Esse objeto é chamado de $usuarioService.

O objeto da classe UsuarioService é criado com base na classe e recebe uma conexão com o banco de dados como um parâmetro no construtor.

A relação entre a classe e o objeto é a seguinte: classe UsuarioService é como um modelo que define o comportamento esperado.

O objeto $usuarioService é uma instância dessa classe e representa uma entidade real que pode realizar as operações definidas na classe.

Ao chamar métodos no objeto $usuarioService, você executa as operações definidas na classe UsuarioService. Neste caso, o método atualizarUsuario é usado para atualizar as informações do usuário no banco de dados.

As classes são usadas para definir estruturas reutilizáveis e encapsular a funcionalidade. Os objetos são instâncias dessas classes, permitindo a interação com a funcionalidade definida na classe. Isso segue o conceito fundamental da programação orientada a objetos.

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.

Utilizar do conhecimento adquirido sobre atributos, métodos, encapsulamento, herança e polimorfismo através de exemplos práticos.

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Tela 5 - Métodos Privados e Públicos

```php
<?php
class UsuarioService {
    private $conexao;

    public function __construct($conexao) {
        $this->conexao = $conexao;
    }

    public function listarUsuarios() {
        $sql = "SELECT * FROM usuario";
        $comando = $this->conexao->prepare($sql);
        $comando->execute();
        $dados = $comando->fetchAll(PDO::FETCH_ASSOC);

        if (count($dados) > 0) {
            return $dados;
        } else {
            return array("RETORNO" => "ERRO", "MENSAGEM" => "REGISTROS NÃO ENCONTRADOS!");
        }
    }

    public function retornarJSON($data) {
        return json_encode($data, JSON_UNESCAPED_UNICODE);
    }
}

// Uso da classe
include "conexao.php"; // Supondo que você já incluiu o arquivo de conexão
$usuarioService = new UsuarioService($conexao);
$dados = $usuarioService->listarUsuarios();
$json = $usuarioService->retornarJSON($dados);
```

Fonte: Autores

ListarUsuarios é um método público, porque ele foi declarado sem especificar uma visibilidade. Isso significa que ele pode ser acessado fora da classe UsuarioService. Esse método é usado para listar os usuários no banco de dados e retornar os dados em formato de array associativo.

O comando retornarJSON é outro método público, também porque não foi especificada uma visibilidade. Ele é usado para converter os dados em formato JSON e retorná-los.

Ambos os métodos podem ser chamados a partir de instâncias da classe `UsuarioService, bem como a partir de código fora da classe, desde que a instância da classe seja criada.

A visibilidade private e protected são usadas para controlar o acesso a métodos e propriedades dentro da classe e são normalmente usadas para encapsular o comportamento interno da classe e protegê-lo de modificações externas.

3.2 LÓGICA DE PROGRAMAÇÃO

A programação lógica serve como a espinha dorsal do desenvolvimento de software. Neste projeto, investigamos os alicerces que conduzem à elaboração de algoritmos eficazes, capacitando os programadores a abordar desafios de forma organizada e otimizada.

Segundo Roveda (2018):

> Lógica de programação é a organização coesa de uma sequência de instruções voltadas à resolução de um problema, ou à criação de um software ou aplicação.A lógica de programação é o conhecimento anterior a qualquer outro quando falamos em desenvolvimento web porque é a partir dele que os aprendizados posteriores, como por exemplo o das linguagens de programação, farão sentido.
> Cada linguagem tem suas próprias particularidades, como sua sintaxe, seus tipos de dados e sua orientação, mas a lógica por trás de todas é a mesma.Em outras palavras, dominar a lógica de programação é a porta de entrada para tornar-se um programador completo, seja em front-end ou em back-end.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

Um dos principais pilares da lógica de programação é o conceito de algoritmos. Os algoritmos são sequências de instruções bem definidas que indicam como um determinado problema deve ser resolvido. Eles são como receitas passo a passo que orientam o computador sobre o que fazer. Ter a capacidade de projetar algoritmos eficientes é essencial para qualquer programador.

Além disso, a lógica de programação envolve o uso de variáveis. As variáveis são espaços de memória que armazenam informações temporárias, como números, texto ou outros dados. Elas são utilizadas para manter o estado do programa e realizar cálculos.

Os tipos de dados são uma parte fundamental da lógica de programação, pois determinam o tipo de informações que uma variável pode armazenar. Isso inclui tipos como inteiros, números de ponto flutuante, strings (cadeias de caracteres) e booleanos (verdadeiro ou falso).

As funções desempenham um papel importante na lógica de programação, permitindo que blocos de código sejam reutilizados em várias partes de um programa. As funções são como mini-programas dentro do programa principal e podem ser chamadas com diferentes argumentos para executar tarefas específicas.

Estruturas condicionais são usadas para tomar decisões dentro de um programa com base em condições específicas. Elas permitem que o programa escolha entre diferentes caminhos de execução com base em valores avaliados, tornando-o mais flexível e capaz de responder a situações variáveis.

Por fim, os operadores são símbolos ou palavras-chave que permitem realizar operações em variáveis e valores. Eles incluem operadores aritméticos (como adição, subtração, multiplicação e divisão), operadores de comparação (para verificar igualdade, maior ou menor) e operadores lógicos (para combinar condições).

Em resumo, a lógica de programação é a base sobre a qual todo o desenvolvimento de software é construído. Compreender e aplicar esses conceitos fundamentais, como algoritmos, variáveis, tipos de dados, funções, estruturas condicionais e operadores, é essencial para criar programas eficazes e resolver problemas computacionais de maneira eficiente.

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP

JavaScript e jQuery revolucionam o desenvolvimento web, trazendo interatividade e agilidade. Enquanto o JavaScript é a espinha dorsal da interação do usuário, o jQuery simplifica o processo, oferecendo uma sintaxe concisa e facilitando a manipulação do DOM. Essa combinação dinâmica permite desde a validação instantânea de formulários até o carregamento ágil de conteúdo, efeitos visuais atrativos e requisições AJAX assíncronas. Juntos, formam uma ferramenta eficaz para criar experiências web envolventes e eficientes.

3.3 MODELAGEM DE DADOS

A modelagem de dados desempenha um papel crucial na ordenação e formatação de informações. Neste trabalho, exploramos os princípios e métodos necessários para criar representações de dados eficazes, desempenhando um papel fundamental no desenvolvimento de sistemas de informação com alto desempenho.

De acordo com Kodado (2015):

> Os bancos de dados são caracterizados pelo armazenamento de informações em massa e pela utilização de linguagens como SQL. Esses sistemas permitem a coleta de dados e o armazenamento simultâneo a fim de prover acesso simplificado por meio de diversos meios

Sendo assim utilizamos linguagens de programação como SQL para armazenar as informações vinda da análise das placas, como antecedentes, se a placa é cadastrada no sistema, hora de chegada e hora de saída, tempo dentro do estacionamento e etc.

Nessa etapa os estudantes devem desenvolver o projeto do banco de dados do sistema, iniciando no modelo lógico e terminando no modelo físico.

3.3.1 MODELO CONCEITUAL

Nesta etapa pensamos em quais seriam as entidades que iriam compor o DER, as entidades criadas foram cliente, placa, sistema, histórico de placa e câmera. Também nessa etapa colocamos os atributos nas entidades como segue o esquema abaixo(Anexo TELA: 6).

Na tela 1 temos o primeiro modelo desenvolvido no programa BRmodelo, após analisarmos mais minuciosamente os requisitos do projeto recriamos um novo modelo lógico do banco de dados utilizando o MySQL workbench, como podemos ver na tela 2.

3.3.2 MODELO LÓGICO E FÍSICO

Já com o modelo conceitual em mente, prosseguimos para o SQL montando o modelo lógico e físico do DER (Diagrama de Entidade Relacionamento), chegando ao seguinte resultado: (Anexo TELA:7).

No modelo lógico, 5 tabelas sendo elas: segurança, cliente, carro, placa e câmera.

A tabela segurança, é o local onde será cadastrado os administradores do sistema, que ficarão responsáveis pelos cadastros dos clientes.

Na tabela cliente, é o local onde ficarão cadastrados os utilizadores do local.

Já na tabela carro, é o local onde ficarão cadastrados os carros dos clientes, enquanto na tabela placas, é o local onde ficarão as placas dos carros cadastrados. Por fim, a tabela camêra, que é onde fica registrado em qual entrada o carro acessou o estabelecimento.

3.3.3 SQL

Para o desenvolvimento do banco de dados a equipe utilizou o programa MySQL workbench, que é uma ferramenta muito utilizada na criação e desenvolvimento de banco de dados, assim a equipe desenvolveu algumas querys para manipulação de dados do projeto. Segue abaixo umas das principais querys utilizadas para dar corpo ao projeto:

SELECT * FROM carro

Neste comando SQL, está sendo realizada uma consulta na tabela "carro". A cláusula "SELECT *" significa que quer ser selecionada todas as colunas da tabela. Portanto, este

comando retorna uma lista completa de carros armazenados no banco de dados, incluindo todos os detalhes de cada carro.

```
SELECT idcarro, modelo, descrição FROM carro
INNER JOIN cliente ON cliente.idcliente = carro.idcarro
WHERE cliente.idcliente = 1;
```

Neste comando, foi realizada uma junção interna entre as tabelas "carro" e "cliente" com base na correspondência entre as colunas "idcliente" e "idcarro". Em seguida, a condição "WHERE" restringe os resultados apenas ao cliente com o ID 1, e o comando retorna informações detalhadas sobre os carros associados a esse cliente.

```
SELECT * FROM carro
INNER JOIN placa ON placa.idplaca = carro.idcarro
WHERE carro.idcarro = 1;
```

Neste comando, realizamos uma junção interna entre as tabelas "carro" e "placa" com base na correspondência entre as colunas "idplaca" e "idcarro". A condição "WHERE" restringe os resultados aos carros com o ID 1, e o comando retorna informações detalhadas sobre os carros cujo ID corresponde ao ID da placa.

Os comandos SQL apresentados desempenham um papel essencial na extração de informações específicas do banco de dados, permitindo filtragem, junção de tabelas e seleção de colunas para atender às necessidades de aplicativos ou sistemas. Eles oferecem uma abordagem estruturada para recuperar dados de tabelas interligadas.

3.4 GESTÃO FINANCEIRA

A gestão financeira desempenha um papel crítico em organizações e na vida pessoal. Este trabalho explora os princípios e práticas que envolvem o manejo eficaz de recursos financeiros, abordando tópicos essenciais, desde orçamentos até investimentos estratégicos.

Entender a gestão financeira é fundamental para tomar decisões informadas e alcançar estabilidade econômica e sucesso financeiro.

De acordo com o blog TOTVS (2023):

> Gestão financeira é o conjunto de práticas e ações que, dentro de uma empresa, buscam a análise, planejamento e controle de toda e qualquer atividade financeira. Sua importância é enorme para que qualquer negócio tenha condição não só de manter-se ativo como prosperar e crescer gradativamente.

O projeto foi desenvolvido com base nas necessidades da UNIFEOB. Nossa equipe projetou o produto que será utilizado na infraestrutura buscando o maior custo benefício do mesmo. Nossa classificação foi dividida em custo direto e custo indireto.

3.4.1 CLASSIFICAÇÃO DOS CUSTOS

Segundo o site ERPconsultoria os custos diretos são aqueles em que o gestor consegue atribuir facilmente, pois são diretamente ligados à linha de produção de uma empresa.

Custos direto: Câmera de vigilância, cabo de rede,computador onde vai estar rodando o software, arduino e o desenvolvimento do software para uso nas câmeras, mão de obra.

O site ERPconsultoria ainda diz que custos indiretos são aqueles que não conseguimos relacionar diretamente ao produto ou serviço. Eles são gastos que não possuem ligação direta com o produto, sendo impossível medir as quantidades dos insumos utilizados por eles.

Custo indireto/fixo: Gasolina para o transporte, salário do funcionário, aluguel, água, energia elétrica, internet, telefone, IPTU, IPVA do carro da empresa.

De acordo com Schoeps (1961) “Visa o custeio direto dar maior flexibilidade à determinação dos preços de venda e do lucro da empresa, separando as despesas fixas de fabricação, de vendas e de administração dos custos diretamente variáveis com a produção”

3.4.2 CUSTOS DO PRODUTO

Para a UNIFEOB calculamos certos produtos como arduino UNO, câmeras ESP32 CAM, metros de fios de energia e o software. Custando respectivamente R$ 51,75 (podendo variar), R$ 53,90 (podendo variar), R$ 2,59 /m (podendo variar), R$ 2.500,00. Somando os custos indiretos também tivemos o aluguel, funcionários, eletricidade, água, telefone, IPVA e internet, totalizando tudo em um valor aproximado de R$ 4.630,78, o que nos ajudou a fazer o rateio total de R$1469,07 sobre o projeto.

Tabela 1 - Custos Direto

Custo Direto	
ESP-CAM32	R$ 53,90
Metro de Fio	R$ 2,60
Arduino Uno	R$ 51,60
Software	R$ 2.500,00
Total	R$ 2.608,10

Fonte: Autores

Tabela 2 - Custos Indireto

Custo Indireto	
Aluguel	R$ 1.500,00
Funcionário	R$ 1.934,00
Eletricidade	R$ 200,00
Água	R$ 100,00
Telefone	R$ 20,00
IPVA	R$ 776,78
Internet	R$ 100,00
Total	R$ 4.630,78

Fonte: Autores

Somando todos os valores obtidos tivemos um valor total do projeto de R$7.239,03.

Tabela 3 - Valor Total do Projeto

Valor Total do Projeto	
Custo Direto	R$2.608,25
Custo Indireto	R$4.630,78
Total	R$7.239,03

Fonte: Autores

O projeto é sim viável em comparação com o mercado já que os preços variam de R$ 4.200,00 a R$ 23.940,00.

Temos algumas empresas da região como Fortress que oferecem um serviço parecido, trabalhando com uma segurança que utiliza funcionários como guardas, porteiros e etc, e a nossa diferença é que trabalhamos com a verificação de placas 24 horas por dia (ou enquanto a empresa ficar aberta) sem utilização de porteiros e guardas.

3.4.3 PRECIFICAÇÃO

Após analisarmos os custos do projeto obtivemos certos valores de precificação. Os valores foram esses:

Tabela 4 - Tabela Precificação

Produto Serviço	Custo fixo	Materiais	Custo total	Preço	Margem de contribuição	Margem	Mark Up	Indice de MarkUp
Sistema de Vigilancia	R$ 4.630,78	R$ 2.608,25	R$ 7.239,03	R$ 10.000,00	R$ 2.760,97	27,61%	38,14%	1,38%

Fonte: Autores

Segundo Gilles (2014)

> Margem de Contribuição é um indicador econômico-financeiro capaz de dizer exatamente se a receita de uma empresa é suficiente para pagar os custos e as despesas fixas e, ainda assim, lucrar. Trata-se de uma informação fundamental, até porque o volume de vendas não é sinônimo de lucratividade.

O projeto está rendendo uma margem de contribuição de R$2.760,97 com um preço de venda de R$10.000.

De acordo com Sousa (2021) a margem específica é:

> Apesar disso, cada empresa tem uma margem de lucro específica, pois esse percentual (valor) depende de diversas variações, como o setor, o tipo de produto e/ou serviço realizado, o preço, etc. Mas, em média, o lucro que se deve esperar é de: 30% a 20% para serviços. 10% a 20% para comércio.

A porcentagem da margem se trata da divisão da margem de contribuição por preço de venda do produto, com isso obtivemos uma margem de 27,61%.

O markup consiste em um índice que tem por objetivo estipular a precificação de determinados produtos e serviços. Ele basicamente representa a diferença de custo entre o preço de venda e o preço de custo de um produto ou mercadoria.

Segundo o site Resultados Digitais (2017):

> O markup é um índice multiplicador que é aplicado ao custo do serviço prestado ou do produto vendido para formar o preço de venda. Seu principal objetivo é garantir que o preço estabelecido cubra, de fato, todos os custos (fixos e variáveis) envolvidos na produção e entrega de cada serviço prestado pela agência.

Fazendo os cálculos, o projeto obteve um índice de markup total de 1,38%.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

3.5.1 GERENCIANDO FINANÇAS

Está disponível para os estudantes no Classroom, o tema “Gerenciando Finanças”.

Nesta parte do Projeto, os estudantes deverão realizar uma síntese dos 4 (quatro) tópicos deste tema, quais sejam:

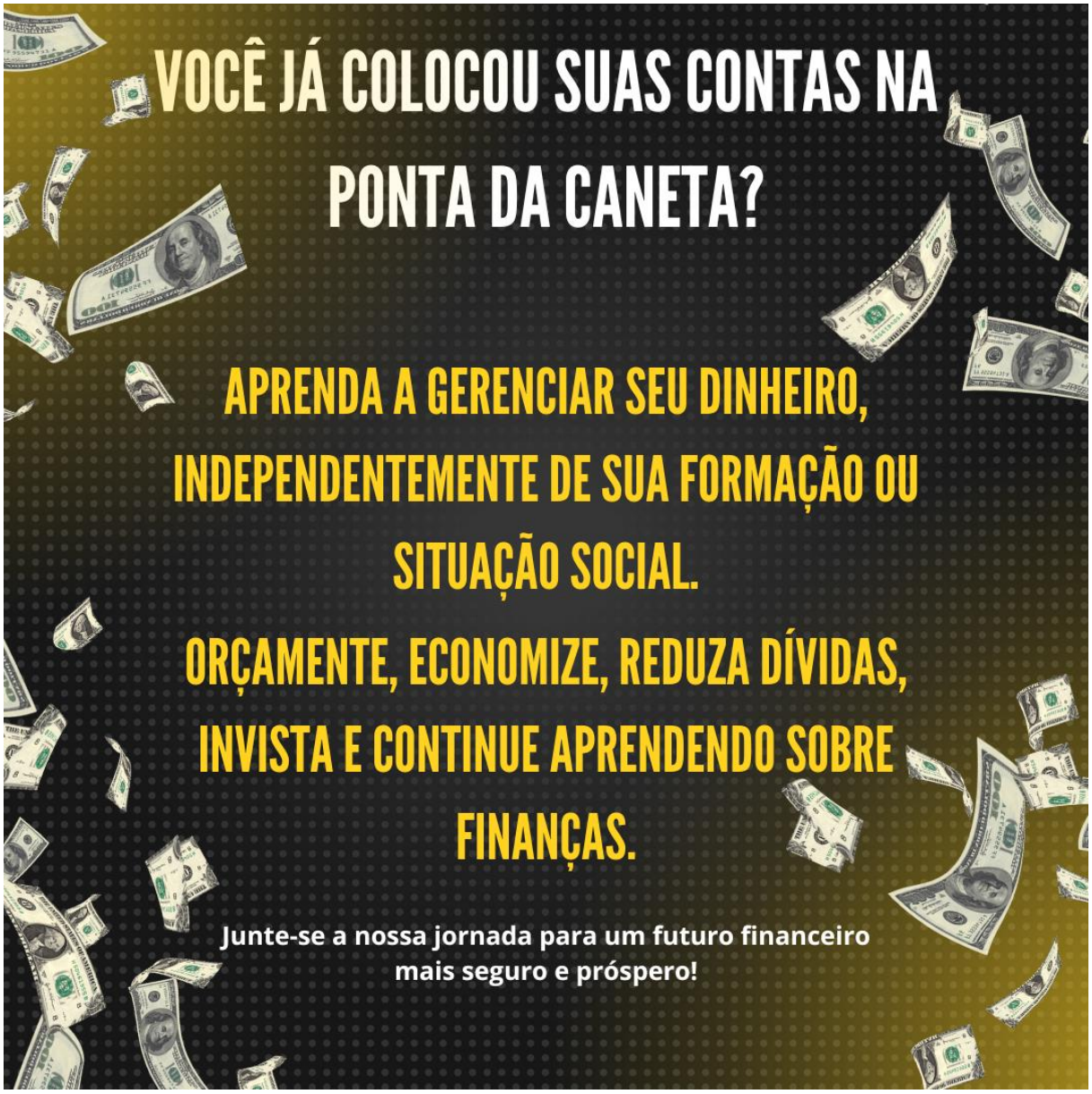
- **Tópico 1:** Introdução aos conceitos econômicos e financeiros básicos
    - Vemos conceitos econômicos que servem tanto para a nossa vida cotidiana quanto para uma vida empresarial, aprendemos que uma empresa é teoricamente igual a uma pessoa física com gastos, despesas e etc, nos é ensinado a tomar cuidado com o quanto é gasto para não ultrapassar o orçamento mensal e criar dívidas com a conta do caixa de fluxo mensal
- **Tópico 2:** Entendendo o ambiente: independência financeira, o valor da minha riqueza e o registro do dia a dia
    - Neste tópico complementa a ideia do tópico passado, mostra o que você pode fazer para gerar mais lucro para você pode ser investindo numa bolsa de valores, em imobiliária ou até mesmo se tornando um acionista, este tópico também nos passa o ensinamentos sobre os cuidados das contas estarem no "vermelho" e estratégias para evitar dívidas e como se afastar delas e como só o controle de gastos não é suficiente você tem que o complementar ele.
- **Tópico 3:** Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado
    - Os objetivos da matemática financeira podem ser considerados sob dois aspectos: de quem aplica - sempre irá procurar as taxas de juros mais altas; e de quem toma emprestado (capta) - sempre irá procurar as taxas de juros mais baixas. A variação das taxas de juros é estabelecida pela oferta e pela demanda de mercado.
    - Os juros compostos determinam que os juros de períodos anteriores são acrescentados ao capital inicial e em cima desse novo valor é realizado um cálculo de juros para o período seguinte.
    - Quando falamos de crédito, estamos associando o tema ao risco do negócio. Um dos objetivos da área financeira é a avaliação e a concessão de crédito a novos clientes. Uma boa análise de crédito pode evitar ou minimizar os riscos em operações de recebimentos duvidosos.

    - O crédito geralmente é concedido após a conferência de alguns documentos, como comprovante de renda, movimentação de conta bancária e declaração de Imposto de Renda.
- **Tópico 4:** Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos
    - Desenvolver um projeto para realizar sonhos envolve estabelecer metas financeiras e engajar o grupo a que se pertence. Isso requer um planejamento sólido e a manutenção de controles financeiros para garantir a estabilidade futura. A gestão financeira pessoal e empresarial compartilha a importância de hábitos e atividades essenciais para evitar problemas financeiros inesperados. A educação financeira desempenha um papel crucial, permitindo o equilíbrio das finanças para a realização de sonhos, independentemente do tamanho. O projeto para atingir esses objetivos envolve priorização, definição de datas-alvo e comprometimento com a criação de um caminho financeiro sólido.

3.5.2 ESTUDANTES NA PRÁTICA

O grupo optou pela realização de um banner sobre gestão financeira para disseminar conhecimento e conscientização sobre a importância do planejamento financeiro. Com um design vibrante e informativo, o banner destaca estratégias para o controle de gastos, a importância de poupar e investir, e aborda conceitos-chave de educação financeira para diferentes faixas etárias. O objetivo é empoderar as pessoas a tomar decisões mais conscientes e inteligentes em relação ao dinheiro, promovendo a estabilidade financeira e o bem-estar econômico.

ISSN 1983-6767



ISSN 1983-6767

4. CONCLUSÃO

No decorrer deste projeto, enfrentamos desafios significativos, como a integração eficiente de objetos em nossa programação orientada a objetos, a criação de algoritmos complexos para lidar com o fluxo de dados das placas, a modelagem e implementação de um banco de dados para armazenar as informações das placas e o gerenciamento financeiro para garantir que nossos recursos fossem utilizados com eficácia.

Ao longo do processo, aprendemos a importância de uma comunicação sólida e colaboração em equipe para superar esses obstáculos. Trabalhar em conjunto para resolver problemas, testar e otimizar nossos algoritmos e manter um controle financeiro rigoroso foram aspectos cruciais do nosso sucesso.

Em resumo, este Projeto Integrado nos desafiou a aplicar e integrar conceitos complexos de programação e gestão de dados em um projeto do mundo real. Essa experiência nos proporcionou habilidades valiosas e conhecimentos práticos que, sem dúvida, serão úteis em nossa futura carreira.

Dificuldades encontradas de cada tópico em específico:

Integração de Objetos: Encontramos desafios ao integrar diferentes objetos em nossa programação orientada a objetos, garantindo que eles se comunicam de maneira eficaz para atingir os objetivos de monitoramento das placas.

Complexidade Algorítmica: Desenvolver algoritmos para rastrear o fluxo de informações das placas de maneira eficiente e precisa foi um desafio, exigindo várias iterações e otimizações.

Modelagem de Banco de Dados: Criar a estrutura de banco de dados ideal para armazenar os dados das placas e os registros de cadastro provou ser uma tarefa complexa, que envolveu a seleção cuidadosa de tipos de dados e relacionamentos

Gestão Financeira: Garantir que nossos recursos financeiros fossem alocados de maneira eficaz e que o projeto permanecesse dentro do orçamento foi uma tarefa que demandou atenção constante.

ISSN 1983-6767

REFERÊNCIAS

HENRIQUE, João. **Programação Orientada a Objetos (POO).** Alura, 2023. Disponível em: https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos. Acesso em: 19 de outubro de 2023.

KONDADO. **Banco de Dados: o que é e quais são os principais tipos.** Kondado, 2017. Disponível em: https://kondado.com.br/blog/blog/2022/09/13/banco-de-dados-o-que-e-e-quais-sao-os-principais-tipos/#:~:text=Os%20bancos%20de%20dados%20s%C3%A3o,por%20meio%20de%20diversos%20meios. Acesso em: 17 de outubro de 2023.

PAUL, Gilles B. de. **Como calcular a margem de contribuição de seus produtos?** Treasy, 2014. Disponível em: https://www.treasy.com.br/blog/como-calcular-a-margem-de-contribuicao-de-seus-produtos/. Acesso em: 15 out.2023.

ROVEDA, Ugo. **O que é lógica de programação?** Kenzie, 2020. Disponível em: https://kenzie.com.br/blog/logica-de-programacao/#:~:text=7.1%20Compartilhe%20isso%3A-,O%20que%20%C3%A9%20l%C3%B3gica%20de%20programa%C3%A7%C3%A3o%3F,de%20um%20software%20ou%20aplicação. Acesso em: 19 out. 2023.

SOUSA, Roberta. **Como Calcular a Margem de Lucro.** Conube, 2021. Disponível em: https://conube.com.br/blog/calcular-margem-de-lucro/#:~:text=Apesar%20disso%2C%20cada%20empresa%20tem,10%25%20a%2020%25%20para%20com%C3%A9rcio. Acesso em: 15 de outubro de 2023.

TOTVS,**Gestão financeira: O que é, pra que serve e dicas.** São Paulo, 06 de Fev de 2023. Disponível em: https://www.totvs.com/blog/servicos-financeiros/gestao-financeira/#:~:text=O%20que%20%C3%A9%20gest%C3%A3o%20financeira%3F,e%20planejar%20suas%20atividades%20financeiras. Acesso em: 22 de Set de 2023.

TURCATO, Augusto. **Gestão financeira. CRM PipeRun**, 2019. Disponível em: https://crmpiperun.com/blog/gestao-financeira/#:~:text=Gest%C3%A3o%20financeira%20%C3%A9%20o%20conjunto,como%20prosperar%20e%20crescer%20gradativamente. Acesso em: 20 out. 2023.

ISSN 1983-6767

ANEXOS

Programação Orientada ao Objeto Tela: 2

```php
<?php
class UsuarioService {
    private $conexao;

    public function __construct($conexao) {
        $this->conexao = $conexao;
    }

    public function listarUsuarios() {
        $sql = "SELECT * FROM usuario";
        $comando = $this->conexao->prepare($sql);
        $comando->execute();
        $dados = $comando->fetchAll(PDO::FETCH_ASSOC);

        if (count($dados) > 0) {
            return $dados;
        } else {
            return array("RETORNO" => "ERRO", "MENSAGEM" => "REGISTROS NÃO ENCONTRADOS!");
        }
    }

    public function retornarJSON($data) {
        return json_encode($data, JSON_UNESCAPED_UNICODE);
    }
}

// Uso da classe
include "conexao.php"; // Supondo que você já incluiu o arquivo de conexão
$usuarioService = new UsuarioService($conexao);
$dados = $usuarioService->listarUsuarios();
$json = $usuarioService->retornarJSON($dados);
```

Fonte : Autores

ISSN 1983-6767

Programação Orientada ao Objeto Tela: 3

```php
<?php
include "conexao.php"; // Suponha que você já incluiu o arquivo de conexão.

class UsuarioService {
    private $conexao;

    public function __construct($conexao) {
        $this->conexao = $conexao;
    }

    public function atualizarUsuario($id, $nome, $modelo, $marca, $placa) {
        $sql = "UPDATE usuario SET nome = :nome, modelo = :modelo, marca = :marca, placa = :plac
        $comando = $this->conexao->prepare($sql);
        $comando->bindParam(':id', $id);
        $comando->bindParam(':nome', $nome);
        $comando->bindParam(':modelo', $modelo);
        $comando->bindParam(':marca', $marca);
        $comando->bindParam(':placa', $placa);
        $this->conexao->beginTransaction();
        $comando->execute();
        $this->conexao->commit();

        $retorno = array("RETORNO" => "SUCESSO", "MENSAGEM" => "DESCRIÇÃO ALTERADA!");
        return json_encode($retorno, JSON_UNESCAPED_UNICODE);
    }
}
```

Fonte : Autores

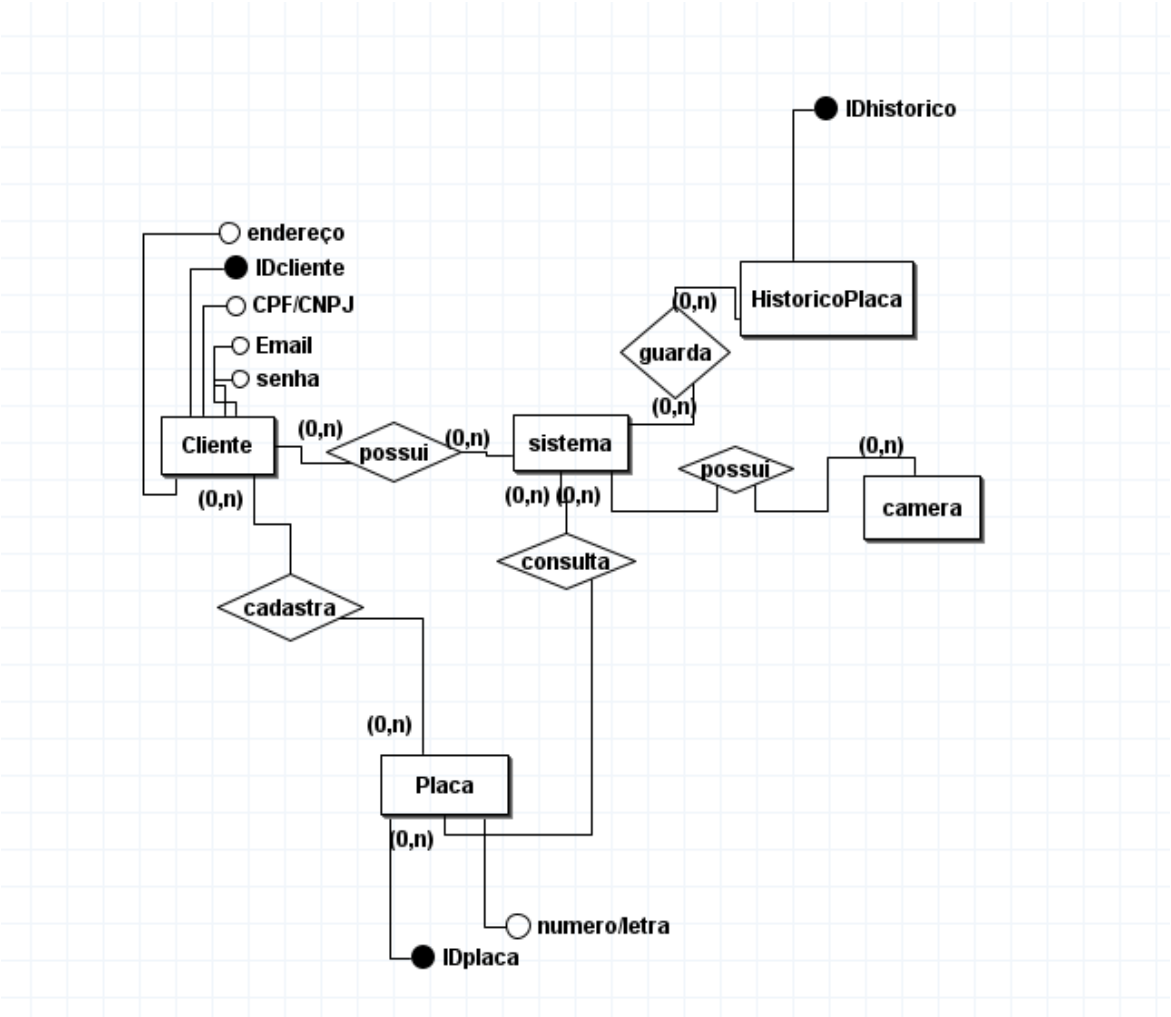
Programação Orientada ao Objeto Tela: 4

```php
// Uso da classe
$usuarioService = new UsuarioService($conexao);

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $id = $_POST['id'];
    $nome = $_POST['nome'];
    $modelo = $_POST['modelo'];
    $marca = $_POST['marca'];
    $placa = $_POST['placa'];

    $json = $usuarioService->atualizarUsuario($id, $nome, $modelo, $marca, $placa);
    echo $json;
}
?>
```

Fonte : Autores

ISSN 1983-6767

Modelagem de Dados Tela: 6



ISSN 1983-6767

Modelagem de Dados Tela: 7