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:

Arthur Jandelli, RA 23001186
Klinsmann Stanguini de Oliveira, RA 23001228;
Lanna Gabriela Greghi Amaro, RA 23001112;
Marcos Henrique Cunha Gonçalves, RA 23000016;
Victor Venturini Neto, RA 23001159.

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2023

SUMÁRIO

ISSN 1983-6767

1. INTRODUÇÃO

O advento da Internet das Coisas (IoT) revoluciona a forma como interagimos com o mundo, possibilitando a criação de soluções inovadoras para diversos desafios. No âmbito acadêmico, a necessidade de otimizar processos, como o registro de presença em instituições de ensino, motiva a concepção do projeto "IoT Data Streamer". Este projeto visa abordar a problemática da fraude de presença em ambientes educacionais por meio da implementação de um dispositivo IoT chamado SmartEye.

O SmartEye é um dispositivo IoT desenvolvido para a autenticação segura e eficiente de usuários por meio de reconhecimento facial e digital. O foco principal é mitigar as práticas fraudulentas relacionadas ao registro de presença em instituições de ensino, garantindo a precisão e integridade das informações coletadas. O dispositivo utiliza tecnologias avançadas de reconhecimento biométrico para assegurar a identificação precisa de cada usuário, proporcionando uma solução robusta e confiável.

O Projeto Integrado (PI) concentra-se na implementação de uma aplicação desktop, utilizando as tecnologias Javascript e Electron JS, destinada a gerenciar os dados coletados associados ao SmartEye. A aplicação visa facilitar a manipulação e análise dos dados provenientes de dispositivos IoT, oferecendo recursos avançados para importação, armazenamento e consulta.

O objetivo central do "IoT Data Streamer" é simular a importação de dados gerados em formato TXT e CSV, provenientes do aplicativo mobile associado ao SmartEye, e populá-los em um banco de dados MySQL. Esses dados, uma vez armazenados, são utilizados para consultas personalizadas, permitindo filtrar informações com base em critérios como dias e datas específicas. A aplicação proporciona visualizações adequadas para facilitar a compreensão e análise dos dados coletados, contribuindo assim para uma gestão eficaz e transparente dos registros de presença.

O projeto está alinhado com as disciplinas do semestre, integrando conceitos de programação orientada a objetos, modelagem de dados com banco de dados em MySQL e gestão financeira. A abordagem multidisciplinar visa proporcionar uma experiência abrangente e prática no desenvolvimento de soluções tecnológicas, preparando os estudantes para enfrentar desafios do mundo real.

Em síntese, o projeto "IoT Data Streamer" emerge como uma resposta inovadora para aprimorar o registro de presença em ambientes educacionais, integrando tecnologias IoT através do SmartEye. Ao aliar conceitos de programação, modelagem de dados e gestão financeira, o projeto não apenas atende às exigências acadêmicas, mas também promete oferecer uma solução prática e eficiente. A aplicação desktop não apenas gerencia dados provenientes do aplicativo mobile, mas sinaliza um passo significativo em direção a uma gestão transparente e segura, contribuindo não só para o ambiente acadêmico, mas também delineando perspectivas para futuras inovações e aplicações em diferentes setores.

ISSN 1983-6767

2. DESCRIÇÃO DA EMPRESA

Com uma sólida trajetória de mais de 50 anos dedicados à educação superior, a UNIFEOB, CNPJ 59.764.555/0001-52, destaca-se como um centro universitário comprometido com a excelência acadêmica e a formação integral de seus estudantes. Ao longo de sua história, a instituição tem se pautado por uma abordagem inovadora no desenvolvimento de projetos pedagógicos, buscando constantemente aprimorar a qualidade do ensino oferecido.

Além do reconhecimento obtido através do Índice Geral de Cursos do MEC, que atesta a qualidade dos cursos oferecidos pela UNIFEOB com um conceito 4 em uma escala que vai até 5, a instituição orgulha-se de ter sido agraciada com o Prêmio Nacional de Gestão Educacional em 2019. Este prestigioso prêmio reconheceu os esforços da UNIFEOB na implementação do projeto pedagógico de formação por competências, destacando-a como referência no cenário educacional brasileiro.

Esse comprometimento constante com a inovação e a busca pela excelência reforçam a posição da UNIFEOB como uma instituição de ensino superior comprometida com o desenvolvimento acadêmico e profissional de seus alunos.

ISSN 1983-6767

3. PROJETO INTEGRADO

Nesta etapa do projeto SmartEye, foram aplicados os conhecimentos adquiridos nas diversas unidades de estudo para garantir sua realização bem-sucedida. A unidade de Modelagem de Dados desempenhou um papel fundamental, pois os conceitos aprendidos em banco de dados e linguagem MySQL, permitiram a implementação eficiente do armazenamento e gerenciamento dos dados de biometria facial e digital do dispositivo.

A unidade de Programação Orientada a Objetos teve relevância na implementação do SmartEye, uma vez que os conceitos de POO e a prática da linguagem Python foram aplicados para o desenvolvimento do software do dispositivo, garantindo uma arquitetura robusta e de fácil manutenção.

Adicionalmente, a unidade de Gestão Financeira contribuiu para o projeto ao fornecer as habilidades necessárias para administrar os custos envolvidos no desenvolvimento e produção do SmartEye. Conceitos como custo direto, indireto, despesa, margem e índice de markup foram fundamentais para a gestão eficaz dos recursos financeiros durante o desenvolvimento do projeto.

Por fim, a unidade de Lógica de Programação teve um papel crucial na implementação da interface do usuário do SmartEye. Os conhecimentos adquiridos em JavaScript e a prática de HTML, CSS e JS foram aplicados para desenvolver uma interface amigável e responsiva para o dispositivo de detecção de presença por biometria facial e digital, proporcionando uma experiência de usuário aprimorada.

De acordo com Moon (2016), a Internet das Coisas (IOT) é um ecossistema que interconecta objetos físicos, permitindo a troca e armazenamento de dados por meio de endereços de IP ou outras redes. O fluxo de dados de um IOT data streamer envolve a coleta inicial por sensores em dispositivos como smartphones, roteadores e wearables, o armazenamento e análise na nuvem, a gestão por aplicativos (software) e o compartilhamento de informações entre os usuários.

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

A aplicação do SmartEye, desenvolvida no âmbito do projeto deste semestre, é uma solução inovadora para a gestão de presença em ambientes acadêmicos. Para atender aos requisitos de robustez, flexibilidade e manutenção, a equipe adotou uma abordagem baseada em Programação Orientada a Objetos (POO) no desenvolvimento do sistema.

A Programação Orientada a Objetos é uma metodologia de desenvolvimento de software que se baseia na definição e interação de objetos, permitindo uma representação mais fiel e modularizada do mundo real. Ao aplicar os princípios da POO, a equipe estruturou a lógica do projeto em entidades discretas, como classes e objetos, facilitando a gestão de dados e operações complexas inerentes ao reconhecimento biométrico e à integração com o banco de dados MySQL.

A modelagem de classes permitiu a representação das funcionalidades do SmartEye por meio de entidades distintas, facilitando a gestão de dados e operações complexas. Essa abordagem promoveu a estruturação do projeto e a reutilização eficiente de código.

O uso de encapsulamento restringiu o acesso direto a certos componentes, aumentando a segurança e a modularidade do sistema. A implementação de herança estabeleceu relações hierárquicas entre as classes, possibilitando a reutilização de funcionalidades comuns e a extensão de comportamentos específicos.

A implementação da Programação Orientada a Objetos foi essencial para a criação de um sistema coeso, modular e de fácil manutenção. A adoção desses conceitos permitiu o desenvolvimento de uma aplicação SmartEye robusta e eficiente, integrando perfeitamente o dispositivo IOT de reconhecimento biométrico com a lógica de programação em Javascript e o banco de dados MySQL, conforme descrito nas seções anteriores desta documentação.

3.1.1 CLASSES E OBJETOS

Durante a implementação do projeto SmartEye, a equipe pôde aplicar de maneira significativa os conhecimentos adquiridos sobre Classes e Objetos, resultando em uma arquitetura bem estruturada e funcionalidades robustas. A seguir, são apresentados exemplos práticos que ilustram a aplicação desses conceitos fundamentais.

Abreu *et al*. (2022) propõem uma abordagem orientada a objetos para a gestão de projetos de software. Essa abordagem baseia-se na ideia de que os projetos de software podem ser modelados como objetos, que interagem entre si para atingir um objetivo comum.

Uma das maneiras pelas quais o conhecimento sobre Classes foi aplicado no projeto foi na representação do próprio dispositivo SmartEye. Por meio da definição de uma classe central que engloba as funcionalidades principais do dispositivo, foi possível modularizar as operações relacionadas ao reconhecimento biométrico, como o registro e a verificação da presença dos usuários. Isso permitiu uma gestão mais eficiente das operações complexas inerentes ao sistema, ao mesmo tempo em que facilitou a manutenção e a extensão do código.

Além disso, o conhecimento sobre Objetos foi aplicado na instância e manipulação de objetos específicos no contexto do projeto SmartEye. Por exemplo, cada usuário que fosse registrado no sistema seria representado como um objeto individual, contendo informações biométricas e de identificação. Esses objetos foram então manipulados para realizar operações de verificação de presença e gerenciamento de dados. A abordagem baseada em Objetos permitiu uma representação mais fiel e uma gestão mais eficiente das entidades envolvidas no processo de reconhecimento de presença.

A utilização prática do conhecimento adquirido sobre Classes e Objetos desempenhou um papel crucial no sucesso da implementação do projeto SmartEye. Através de uma estrutura bem definida e da manipulação eficiente de objetos, foi possível desenvolver uma aplicação desktop integrada com o dispositivo IOT e o banco de dados MySQL, fornecendo uma solução eficaz para o problema de controle de presença em ambientes acadêmicos.

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO E HERANÇA.

A aplicação do conhecimento adquirido sobre atributos, métodos, encapsulamento e herança desempenhou um papel fundamental no desenvolvimento do projeto SmartEye, permitindo a criação de uma estrutura coesa e funcionalidades otimizadas. A seguir, são apresentados exemplos práticos que demonstram a aplicação desses conceitos essenciais.

Através da definição de atributos específicos, como informações biométricas e dados de identificação, e métodos adequados, como a verificação de presença e o registro de usuários, a equipe pôde estruturar eficientemente o funcionamento do SmartEye. Por exemplo, a criação de métodos para registrar usuários, juntamente com atributos para armazenar dados relevantes, contribuiu para a integridade e eficácia do sistema no reconhecimento de presença.

A aplicação de encapsulamento é essencial para potencializar a segurança e a integridade dos dados do projeto SmartEye. Ao restringir o acesso direto a certos componentes do código-fonte e disponibilizar interfaces específicas para interações externas, a equipe garantiu a proteção e o correto funcionamento das funcionalidades críticas do

dispositivo. O encapsulamento também permitiu a criação de uma barreira protetora em torno de informações sensíveis, contribuindo para a confiabilidade e estabilidade do sistema como um todo.

A implementação de herança no projeto SmartEye foi realizada de forma a promover a reutilização eficiente de funcionalidades comuns e a extensão de comportamentos específicos. Por exemplo, a definição de uma classe base para funcionalidades genéricas do dispositivo, juntamente com subclasses especializadas para operações específicas, permitiu uma organização hierárquica clara e a facilitação do desenvolvimento de novas funcionalidades. A herança também contribuiu para a redução da redundância de código e para a manutenção simplificada do sistema.

A incorporação do princípio de herança na tela de Missão, Visão e Valores da UNIFEOB proporciona uma estrutura organizada e coerente para transmitir a identidade institucional. Ao aplicar a herança, é possível estabelecer relações hierárquicas entre esses elementos, facilitando a gestão e manutenção das informações. Dessa forma, a tela se torna mais modular e adaptável, refletindo de maneira eficaz os princípios fundamentais que guiam a instituição.

A aplicação prática do conhecimento adquirido sobre atributos, métodos, encapsulamento e herança foi substancial para o sucesso do projeto SmartEye. Através da implementação cuidadosa desses conceitos-chave, foi possível desenvolver uma solução eficiente e segura para o registro de presença em ambientes acadêmicos, integrando perfeitamente o dispositivo IOT e o banco de dados MySQL, conforme descrito nas seções anteriores desta documentação.

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

A aplicação do conhecimento adquirido sobre métodos estáticos, públicos e privados desempenhou um importante papel no desenvolvimento eficaz e na funcionalidade otimizada do projeto. A seguir, são apresentados exemplos práticos que ilustram a aplicação desses conceitos no contexto do projeto.

A equipe incorporou métodos estáticos no projeto SmartEye para facilitar a execução de operações que não dependem de instâncias específicas da classe. A aplicação de métodos estáticos permitiu que determinadas funcionalidades fossem acessadas sem a necessidade de criar instâncias adicionais da classe, resultando em um desempenho aprimorado do sistema como um todo.

Métodos públicos foram implementados para facilitar a interação com outros componentes do sistema, permitindo operações essenciais, como o registro de presença e a autenticação de usuários. Por outro lado, métodos privados foram utilizados para realizar operações internas específicas que não precisavam ser acessíveis externamente, contribuindo para a segurança e a integridade dos dados sensíveis manipulados pelo dispositivo.

Através da implementação cuidadosa desses conceitos-chave, a equipe conseguiu criar uma aplicação desktop coesa e segura, integrando perfeitamente o dispositivo IOT de reconhecimento biométrico com a lógica de programação em JavaScript e o banco de dados MySQL, como descrito nas seções anteriores desta documentação.

3.2 LÓGICA DE PROGRAMAÇÃO

De acordo com o autor Michael Sipser (2019), a lógica de programação é um ramo da ciência da computação que estuda a estrutura e a semântica de programas de computador. Ela fornece uma base formal para o desenvolvimento de programas corretos e eficientes.

O projeto da interface do IOT Data Streamer foi feito baseado em Electron JS para a criação de uma aplicação desktop para ser integrada com o IOT. Essa aplicação representa uma solução inovadora e eficaz para o controle de presença em ambientes educacionais. Ela oferece diferentes níveis de acesso para administradores, professores e alunos, permitindo um gerenciamento eficiente de cursos, alunos, professores e registros de presença. A utilização de tecnologias modernas e a integração com o IOT agregam valor à experiência de controle de presença, tornando-a mais precisa e segura.

O propósito central desta aplicação é substituir as metodologias tradicionais de registro de presença em ambientes educacionais, adotando o reconhecimento facial e digital como meios de controle de presença e acesso.

A aplicação foi desenvolvida para atender às demandas de gestão e controle de presença em instituições de ensino, levando em consideração diferentes níveis de acesso entre administradores, professores e alunos. A conexão com o dispositivo IOT, SmartEye, desempenha um papel fundamental na coleta de informações biométricas para fins de autenticação e controle de presença.

O uso do framework Electron JS foi selecionado devido à sua capacidade de criar aplicações desktop multiplataforma, tornando-a compatível com sistemas operacionais Windows, macOS e Linux, aproveitando tecnologias web, como HTML, CSS e JavaScript, já o sistema de gerenciamento de banco de dados relacional MySQL foi escolhido para a gestão e armazenamento de dados do sistema, devido à sua robustez e ampla aceitação em contextos de sistemas de informação.

Conforme Sanchez, o Electron é um framework que permite o desenvolvimento de aplicativos desktop usando JavaScript, HTML e CSS. Ele é baseado no Chromium e no Node.js, o que significa que os aplicativos Electron podem acessar as APIs do sistema operacional e usar recursos da Web. Por exemplo, respectivamente, o script que controla a instância da janela do Electron JS (main.js) e o script inicial indicado no arquivo principal da aplicação (index.html).

No que se refere aos diferentes níveis de acesso na aplicação, os administradores podem realizar o cadastro e gerenciamento de cursos, professores e alunos, além de monitorar e gerar relatórios de presença e controlar o acesso ao sistema. Os professores têm a capacidade de registrar a presença em suas turmas por meio do SmartEye, acessar registros de presença e visualizar informações relativas aos alunos e cursos relacionados ao seu perfil. Os alunos, por sua vez, podem checar qual a aula atual e se estão presentes ou não, acessar seus próprios registros de presença e informações pessoais, bem como detalhes dos cursos nos quais estão matriculados.

A operação da aplicação se baseia na integração entre o software desktop e o dispositivo SmartEye, com os alunos e professores fazendo uso de seus dados biométricos; reconhecimento facial ou digital; para efetuar o registro de presença. Os dados resultantes são armazenados no banco de dados MySQL (MariaDB) e podem ser acessados por webmasters, administradores e professores para fins de monitoramento e geração de relatórios.

Em resumo, a aplicação desktop desenvolvida com Electron JS e integrada ao SmartEye representa uma solução inovadora e eficaz para o controle de presença em instituições de ensino. Ela oferece diferentes níveis de acesso, permitindo um gerenciamento eficiente de cursos, alunos, professores e registros de presença, e a integração com um dispositivo IOT agrega valor ao processo, tornando-o mais preciso e seguro.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

Compreender os conceitos essenciais que regem a lógica de programação é crucial para a construção de aplicativos eficientes e funcionais. Neste tópico, serão explorados os principais elementos da lógica de programação, destacando a sua aplicação prática no contexto do projeto SmartEye.

Algoritmos são empregados para o processamento e armazenamento dos dados biométricos coletados, assegurando um desempenho eficiente e uma experiência de usuário aprimorada.

As variáveis desempenham um papel essencial no armazenamento e manipulação de informações como configurações do dispositivo, dados de usuários e registros de presença, contribuindo para o gerenciamento eficaz dos recursos do SmartEye.

Dada a diversidade de tipos de dados em JavaScript, incluindo números, strings, booleanos, arrays e objetos, compreender esses tipos é vital para a correta manipulação e armazenamento das informações necessárias para o funcionamento adequado do projeto.

Funções são usadas para modularizar o código da aplicação desktop, melhorando sua legibilidade e manutenção. Elas executam tarefas específicas, como o processamento de dados biométricos, verificação de identidade e interação com o dispositivo de Internet das Coisas (IOT) SmartEye.

As estruturas condicionais são empregadas para controlar o fluxo do programa com base em condições específicas, garantindo a segurança e a precisão na verificação de identidade dos usuários, contribuindo para a integridade do sistema como um todo.

Os operadores em JavaScript desempenham um papel fundamental no processamento e manipulação de dados na aplicação desktop. Eles são utilizados para realizar operações específicas em variáveis e valores

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP

Electron JS é um framework de JavaScript de desenvolvimento de aplicações desktop que permite a criação de aplicativos usando HTML, CSS e JavaScript. Ele é baseado na tecnologia Chromium, o que significa que as aplicações desenvolvidas com Electron JS são compatíveis com todos os principais sistemas operacionais, incluindo Windows, macOS e Linux. O Framework é fundamentado em tecnologias web, o que torna mais fácil para os desenvolvedores criarem aplicações desktop ágeis e responsivas.

A criação de interfaces de usuário (UI) é uma parte crucial no desenvolvimento de aplicativos de software. Uma UI bem projetada é fundamental para garantir que os usuários possam interagir de maneira intuitiva e descomplicada. Durante a criação da aplicação desktop SmartEye, foram utilizados HTML para a estruturação do conteúdo das interfaces, CSS para o design visual das telas e JavaScript para a interatividade do projeto.

Além do foco na criação da UI, habilidades fundamentais foram aplicadas para garantir a robustez da aplicação, incluindo:

Manipulação de arquivos: Essencial para tarefas como a gravação de dados em formatos como JSON, CSV ou TXT, o carregamento de arquivos de configuração e a geração de relatórios. A manipulação de arquivos foi implementada de forma eficaz usando JavaScript.

Conexão e operações de banco de dados: Fundamental para o desenvolvimento do SmartEye, permitindo a inserção de dados relevantes para o registro de presença do dispositivo IoT, consultas de dados, atualizações e exclusões de informações cruciais do projeto.

Empacotamento e publicação: Essas etapas são essenciais no desenvolvimento de aplicativos desktop com Electron JS. O empacotamento envolve a criação de um arquivo executável que pode ser instalado e executado em um sistema operacional. A publicação, por sua vez, diz respeito ao processo de disponibilizar o aplicativo para outros usuários.

Com a aplicação efetiva dessas habilidades, o projeto SmartEye foi capaz de alcançar uma interface amigável, funcionalidades robustas e uma presença sólida no ambiente de desenvolvimento de aplicações desktop.

3.3 MODELAGEM DE DADOS

De acordo com Milani, o MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional,, de licença dupla (sendo uma delas um software livre), projetado inicialmente para trabalhar com aplicações de pequenos e médios portes, mas hoje atendendo a aplicações de grande porte e com mais vantagens do que seus concorrentes. Possui todas as características que um banco de dados de grande porte precisa, sendo reconhecido por algumas entidades como o banco de dados Open Source com maior capacidade para concorrer com programas similares de código fechado, tais como SQL Server (da Microsoft) e Oracle.

No desenvolvimento do banco de dados, foram levados em consideração vários fatores, incluindo a complexidade da estrutura educacional da faculdade, que envolve a gestão de um grande número de alunos, aulas, professores e semestres. Além disso, reconheceu-se a necessidade de estabelecer diferentes níveis de acesso ao software de controle, de acordo com as funções desempenhadas por diferentes usuários, tais como administradores, professores e alunos.

Em MySQL, assim como em muitos outros bancos de dados, uma hierarquia de estruturas de dados complexas é utilizada para armazenar informações. No contexto desse sistema, a unidade fundamental é a tabela, que mantém os registros ou blocos de dados (Gonzaga & Birckan, 2000). Esses registros, por sua vez, consistem em objetos menores conhecidos como tipos de dados, que podem ser manipulados pelos usuários. A combinação de um ou mais tipos de dados forma um registro. Portanto, a hierarquia de um banco de dados pode ser concebida como Banco de dados > Tabela > Registro > Tipo de dados. Os tipos de dados apresentam uma variedade de formas e tamanhos, proporcionando aos programadores a flexibilidade para criar tabelas específicas de acordo com suas necessidades (Gonzaga & Birckan, 2000, p. [4]).

O processo de criação do banco de dados começou com a compreensão abrangente desses requisitos. Isso implicou a elaboração de um modelo conceitual do banco de dados, que mapeou as entidades-chave, seus relacionamentos e os atributos relevantes. Em seguida, o modelo conceitual foi refinado para um modelo lógico, onde as estruturas de dados foram detalhadas, incluindo a definição de chaves primárias e estrangeiras, bem como restrições de integridade referencial.

Finalmente, para implementação prática, o modelo físico do banco de dados foi desenvolvido utilizando o sistema de gerenciamento de banco de dados MySQL. Neste estágio, as tabelas, campos, índices e outras características técnicas do banco de dados foram

definidos de forma a acomodar eficientemente os dados e garantir o desempenho adequado do sistema.

No que diz respeito aos diferentes níveis de acesso, o banco de dados foi configurado para refletir as permissões e privilégios associados às funções desempenhadas por cada usuário. Isso permitiu a segmentação e controle apropriados, garantindo que apenas informações autorizadas fossem acessadas e manipuladas por administradores, professores e alunos, de acordo com suas respectivas responsabilidades e necessidades.

Em suma, o desenvolvimento do banco de dados para este projeto incorporou uma análise abrangente dos requisitos específicos da faculdade, a modelagem conceitual, lógica e física do banco de dados e a configuração de níveis de acesso com base nas funções dos usuários. Essa abordagem meticulosa garante a eficácia, segurança e confiabilidade do sistema no controle de presença e gerenciamento acadêmico.

A aplicação em questão apresenta dependências essenciais nas bibliotecas "mysql" e "mysql2" para facilitar a interação com um banco de dados relacional, mais especificamente o MariaDB. A gestão desse banco é realizada através do software XAMPP. No script responsável pela tela de login (index.html), um script JavaScript (index.js) é invocado.

No início do script, uma constante denominada "connection" é declarada, a qual recebe o módulo "createConnection" da biblioteca "mysql". Essa constante é então configurada com os parâmetros necessários para estabelecer a conexão com o banco de dados MariaDB. Subsequentemente, a cada transação com o banco, utiliza-se uma estrutura de consulta baseada no modelo "connection.query", seguido pela ação desejada.

Esse padrão é mantido consistente nos demais scripts da aplicação, garantindo uma abordagem coesa e uniforme em todas as interações com o banco de dados. Este método estruturado facilita a manutenção do código, melhorando a legibilidade e a compreensão das transações relacionadas ao banco de dados em toda a aplicação.

3.3.1 MODELO CONCEITUAL

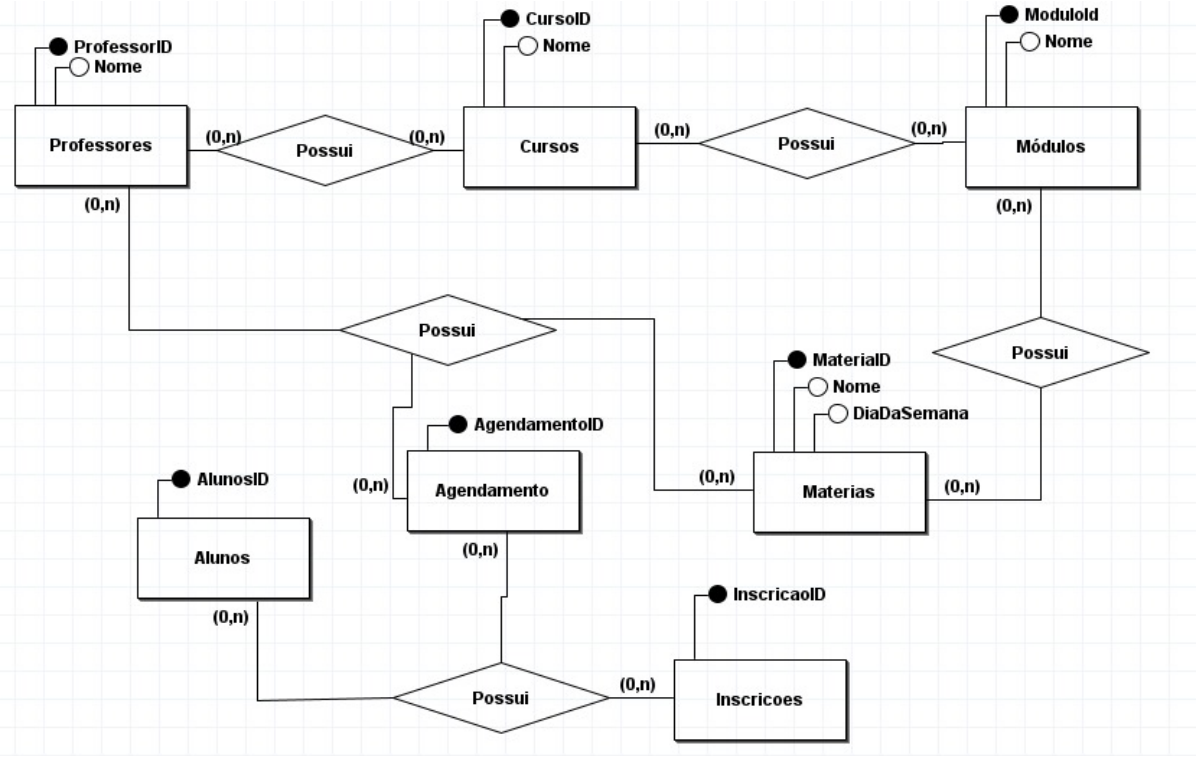
No modelo conceitual apresentado nos anexos, criado no BrModelo, foi criado uma primeira versão do banco de dados utilizado em nossa aplicação. Foram criadas sete tabelas, cada uma com um propósito específico para a estrutura do banco de dados. A tabela "Professores" inclui os campos "ID do Professor" e "Nome do Professor", identificando exclusivamente os professores e seus nomes. Na tabela "Cursos", são registrados os "ID do Professor", "ID do Curso" e "Nome do Curso", permitindo a associação de cursos a professores específicos. A tabela "Módulos" contém os campos "ID do Curso", "ID do

Módulo" e "Nome do Módulo", o que possibilita a organização dos cursos em módulos distintos.

A tabela "Matérias" se relaciona com os módulos e inclui os campos "ID do Módulo", "Dia da Semana", "ID da Matéria" e "Nome da Matéria", permitindo a atribuição de matérias a módulos específicos em dias específicos da semana.

A tabela "Agendamento" tem o propósito de conectar professores, matérias e dias da semana. Ela é composta pelos campos "Dia da Semana", "ID do Professor", "ID da Matéria" e "ID do Agendamento", possibilitando a criação de agendamentos para a programação de aulas. A tabela "Alunos" é responsável pelo registro de informações dos alunos e inclui os campos "ID do Aluno" e "Nome do Aluno".

Por fim, a tabela "Inscrições" serve para associar alunos a agendamentos, definindo horários de aula. Ela possui os campos "ID do Aluno", "ID do Agendamento" e "ID da Inscrição". Essa abordagem estruturada e bem definida no modelo conceitual visa a organização eficiente e coerente das informações no banco de dados, permitindo um controle rigoroso e uma administração precisa das atividades acadêmicas e dos agendamentos.



3.3.2 MODELO LÓGICO E FÍSICO

O desenho meticuloso do banco de dados para o Sistema de Controle Acadêmico reflete uma abordagem estruturada e eficiente na gestão de informações essenciais para instituições educacionais. O âmago dessa lógica reside na representação detalhada de entidades cruciais, abrangendo administradores, professores, alunos, cursos, matérias, módulos, agendamentos, inscrições e registros de presença. Vamos agora explorar minuciosamente a arquitetura e a lógica que norteiam esse banco de dados.

A tabela denominada administrativo serve como repositório para informações relativas aos administradores do sistema. Nela, são armazenados dados vitais, como nome, login, senha e perfil de cada administrador. A unicidade de cada administrador é garantida pelo identificador exclusivo denominado AdministrativoID.

Na esfera dos educadores, a tabela professores contém pormenores relevantes, como nome, login, senha e perfil individual. A identificação única de cada professor é preservada por meio do ProfessorID, proporcionando uma estrutura sólida para a gestão desses profissionais.

Quanto aos estudantes, a tabela alunos incorpora uma abrangência de dados, incluindo nome, login, senha, perfil, foto e informações acadêmicas pertinentes. O mecanismo de singularidade é garantido pelo identificador AlunoID, conferindo solidez à individualização de cada aluno.

A contextualização acadêmica se estende para a tabela cursos, na qual são registradas informações relativas aos cursos oferecidos, incluindo o docente responsável. De maneira integrada, a tabela modulos detalha os módulos associados a cada curso, estabelecendo uma relação consolidada por meio da chave estrangeira CursoID.

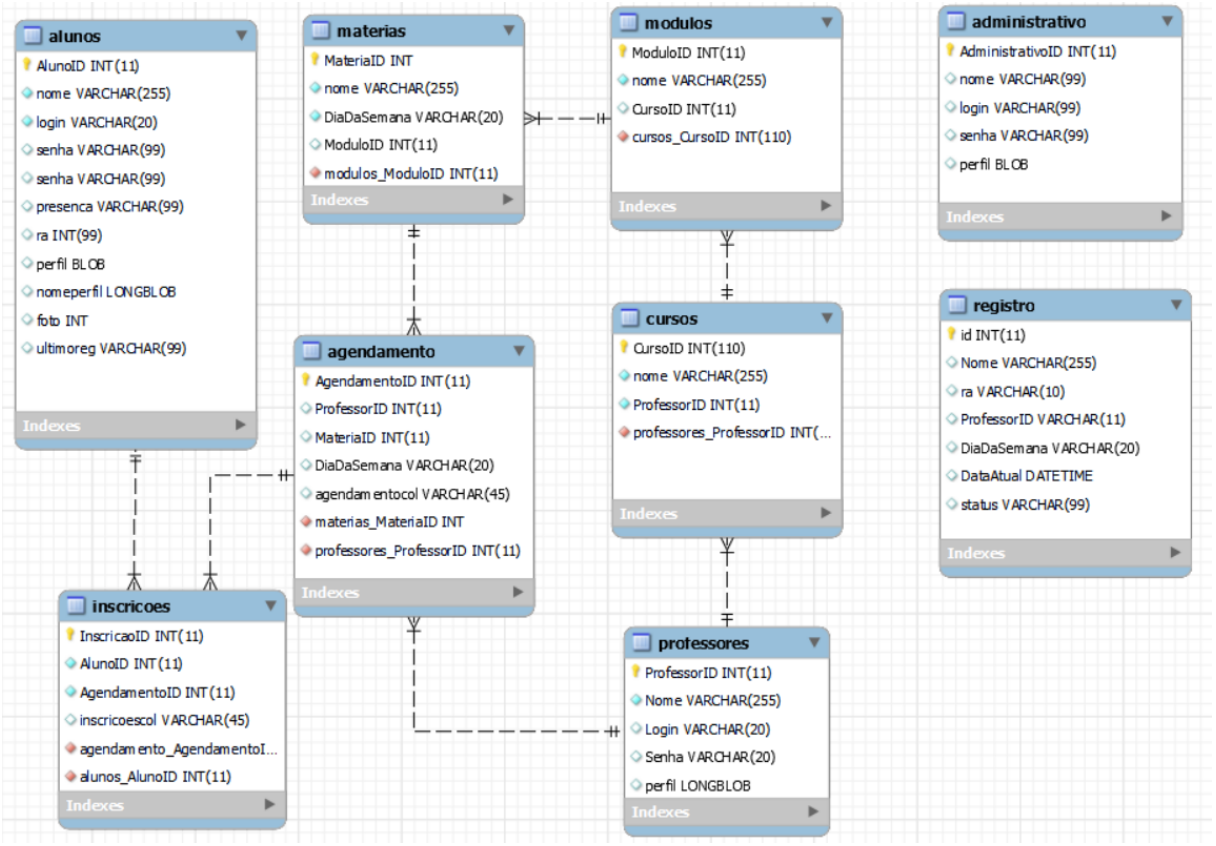
O domínio das disciplinas encontra expressão na tabela materias, que abarca informações cruciais como nome, dia da semana e a vinculação ao módulo específico. A relação entre materias e modulos é concretizada pela chave estrangeira ModuloID, proporcionando uma conexão essencial entre essas entidades acadêmicas.

O agendamento de aulas é registrado na tabela agendamento, na qual professores, matérias e dias da semana são meticulosamente vinculados. As chaves estrangeiras ProfessorID e MateriaID estabelecem vínculos precisos com as tabelas professores e materias, respectivamente, consolidando assim a integralidade do agendamento acadêmico.

A rastreabilidade das inscrições estudantis em agendamentos específicos encontra sua expressão na tabela de inscrições. As chaves estrangeiras AlunoID e AgendamentoID

estabelecem referências cuidadosas às tabelas alunos e agendamento, proporcionando uma visão abrangente das participações estudantis.

Explorando a faceta dos Registros de Presença, a tabela registro emerge como um repositório de detalhes minuciosos, associando alunos, professores, dias da semana e os estados de presença. Essa estrutura visa manter a coesão referencial entre as entidades, garantindo a consistência das informações e respeitando as relações entre as tabelas. A adoção perspicaz de chaves primárias e estrangeiras facilita a navegação fluida entre as tabelas, capacitando a execução eficiente de consultas complexas para a obtenção de informações específicas e fundamentais.



3.3.3 SQL

A eficácia de um sistema depende, em grande parte, da integridade e da qualidade dos dados que o alimentam. No contexto do projeto SmartEye, a implementação do banco de dados MySQL é um componente crucial para a gestão eficiente da presença de professores e alunos por meio do dispositivo IoT. Neste cenário, a capacidade de manipular dados por meio de comandos SQL é fundamental, não apenas para a população inicial do banco, mas também para a realização de testes e o desenvolvimento de consultas estratégicas.

A inserção de dados fictícios no banco de dados é um passo vital no processo de desenvolvimento, possibilitando simulações realistas e testes eficazes. Abaixo, destacamos os comandos SQL utilizados para a população inicial das tabelas do projeto.

A integração destes comandos nas unidades de programação permite que os alunos compreendam profundamente a interação entre o banco de dados MySQL e a aplicação desktop desenvolvida em Javascript com Electron JS. Essa visão prática fortalece a base teórica adquirida durante a unidade de estudos de modelagem de dados e programação orientada a objetos, promovendo uma compreensão mais abrangente e aplicada do projeto SmartEye.

ISSN 1983-6767

3.4 GESTÃO FINANCEIRA

Segundo Drucker (1990), o planejamento financeiro é crítico para o sucesso financeiro de qualquer empresa, nas suas próprias palavras sendo o mapa que guia o caminho para a lucratividade, por isso o custo do projeto é um elemento fundamental em nossa iniciativa atual. É uma ferramenta insubstituível na tomada de decisões que impactam o sucesso de um negócio. Estamos realizando uma pesquisa teórica abrangente sobre as principais classificações de custos, com base nas informações da planilha, que incluem categorias de custos de matéria-prima, como "ESP32 cam" e "leitor digital," custos indiretos como "energia" e "água," e salários associados a cargos como "Supervisores" e "Dev Junior."

Vamos calcular o valor total do projeto da empresa com base nessas informações e, em seguida, conduzir uma análise de viabilidade para determinar se o projeto é viável. Essa abordagem baseada em dados permitirá tomar decisões informadas e maximizar as chances de sucesso do empreendimento.

3.4.1 CLASSIFICAÇÃO DOS CUSTOS

Custos Diretos e Custos Indiretos:

Antes de falar sobre custo direto e indireto especificamente, é importante mencionar o quão fundamental é a diferenciação entre ambos, pois nos permite identificar onde estão os principais impulsionadores de despesas e tomar decisões informadas para otimizar a rentabilidade.

De acordo com Eliseu Martins, a classificação dos custos em diretos e indiretos diz respeito ao produto fabricado ou serviço prestado, e não à produção como um todo ou aos departamentos da empresa.

Custos Diretos: São os custos que podem ser diretamente atribuídos a um produto específico ou ao projeto em questão. No contexto da sua empresa, os custos diretos são aqueles relacionados diretamente à fabricação ou desenvolvimento do seu projeto. Um exemplo disso é o custo das matérias-primas, como ESP32 CAM, protoboard, leitor digital, jumpers, FTDI conversor, case e display 16x2. Esses custos são incorridos diretamente devido à produção do projeto e podem ser rastreados diretamente até ele.

Custos Diretos	
Produto	**Valor**
Dev Senior	R$12.000,00
Dev Pleno	R$7.000,00
Dev Junior	R$3.600,00
Windows (pg único)	R$563,20
Firebase	R$126,00
ESP32 CAM	R$80,00
Leitor Digital	R$70,00
Figma	R$60,00
Display 16x2	R$22,00
FTDI Conversor	R$20,00
Protoboard	R$15,00
Case	R$15,00
8 Jumpers	R$1,50
Total	R$23.572,00

Custos Indiretos: São custos que não podem ser atribuídos diretamente a um produto específico ou projeto, mas são necessários para manter as operações da empresa funcionando. No seu caso, os custos indiretos incluem despesas gerais, como energia e água. Esses custos são necessários para manter o ambiente de trabalho operacional, mas não podem ser vinculados diretamente a um único projeto.

Custos Indiretos

Produto	Custo
Energia Elétrica	R$80,00
Água	R$60,00
Total	R$15.140,00

Mão de Obra	
Supervisor	R$15.000,00
Dev. Senior	R$12.000,00
Dev. Pleno	R$7.000,00
Dev. Junior	R$3.600,00
Total	R$37.600,00

Custos Fixos e Custos Variáveis:

Custos Fixos: São custos que permanecem relativamente constantes, independentemente do volume de produção ou do projeto. Em sua empresa, os salários dos funcionários, como os supervisores, Dev Senior, Dev Pleno e Dev Junior, são exemplos de custos fixos. Esses salários são pagos regularmente e não variam com o número de projetos produzidos. Mesmo se não houver produção em um determinado mês, esses custos ainda precisam ser pagos.

Custos Variáveis: São custos que variam proporcionalmente com o volume de produção ou a quantidade de projetos realizados. No seu caso, os custos variáveis estão relacionados às matérias-primas, listadas anteriormente. À medida que você produz mais unidades do seu projeto, você precisa adquirir mais dessas matérias-primas, o que resulta em um aumento proporcional nos custos. Por exemplo, se você dobrar a produção, o custo das matérias-primas também dobrará. Essa classificação é fundamental para a gestão financeira da sua empresa, pois ajuda a entender como os custos afetam a rentabilidade e a tomada de decisões relacionadas à produção, precificação e alocação de recursos.

3.4.2 CUSTOS DO PRODUTO

No âmbito do projeto SmartEye, é imperativo realizar uma apuração minuciosa dos custos envolvidos na produção e operação do produto, bem como avaliar sua viabilidade em relação ao atual cenário de mercado. A apuração de custos abrange diversas categorias, sendo os custos diretos os mais relevantes. Estes incluem a matéria-prima, como a ESP32 cam, protoboard, leitor digital, 8 jumpers, FTDI conversor, case e display 16x2, que compõem diretamente o produto final, além dos salários de colaboradores, como supervisores, desenvolvedores seniores, plenos e juniores, que desempenham papéis fundamentais no processo de criação do SmartEye.

Também são considerados custos diretos os gastos com assinaturas de software, como Figma, Windows e Firebase, pois essas ferramentas são utilizadas diretamente no desenvolvimento do projeto. Além disso, os custos indiretos, como energia e água, embora não estejam diretamente ligados à produção do SmartEye, impactam o custo global de operação da empresa.

A análise de viabilidade no mercado de cibersegurança e produtos de ponto eletrônico revela um cenário promissor e desafiador para o projeto SmartEye, especialmente ao considerar os valores dos produtos de empresas concorrentes. No segmento de cibersegurança, a crescente conscientização sobre a importância da segurança de acessos e sistemas impulsiona a demanda por soluções eficazes, como o SmartEye. Ao examinarmos os preços praticados por empresas concorrentes nesse espaço, notamos que produtos semelhantes, como os oferecidos pela "Control ID" são comercializados por R$1.171,00, enquanto a "Manchester Automação" disponibiliza produtos por R$1.530,00. Vale ressaltar que esses valores se referem apenas ao produto, excluindo qualquer software adicional, o que representa uma referência importante para a definição da estratégia de preços do SmartEye. No contexto dos produtos de ponto eletrônico, onde a "Control ID" e a "Manchester Automação" são players significativos, a análise de mercado mostra que os preços praticados por essas empresas estão na faixa de R$1.171,00 a R$1.530,00, excluindo o software. Nesse sentido, o SmartEye pode posicionar-se competitivamente ao oferecer funcionalidades exclusivas a um preço acessível, alinhado com os valores da concorrência. Portanto, estratégias de marketing que destacam os diferenciais do SmartEye em relação às ofertas da "Control ID" e da "Manchester Automação" serão fundamentais para conquistar uma fatia desse mercado em constante evolução. Em resumo, a análise de viabilidade indica que o SmartEye possui oportunidades consideráveis nos mercados de cibersegurança e produtos de ponto eletrônico. A capacidade do projeto de oferecer soluções eficazes a preços competitivos

em relação a esses concorrentes será um fator determinante para o seu sucesso e a entrada nesse mercado em constante evolução.

3.4.3 PRECIFICAÇÃO

Produto	Custo Fixo	Matéria Prima	Impostos	Custo Total	Preço	Margem de Contribuição
SmartEye	R$367,00	R$786,70	R$9,11	R$1.162,81	R$1.750,00	R$587,19

Margem	MarkUp	Índice de MarkUp
33,55%	50,50%	1,50

Quando se trata de estabelecer o preço dos nossos produtos, seguimos as informações já mencionadas neste documento. O custo fixo, que engloba os salários dos nossos funcionários, é rateado entre os produtos que eles fabricam, com uma média de 100 unidades produzidas por mês. Esse custo fixo é fundamental para calcular o custo total de produção.

Os custos variáveis incluem os materiais necessários para a produção e os impostos, como o ICMS a 12% e o PIS a 0,65%. Esses custos variáveis também são distribuídos proporcionalmente entre os produtos com base na quantidade produzida. Eles influenciam diretamente o custo total de produção.

Com todos esses números em mente, chegamos ao custo total do SmartEye, que é de R$1.162,81. Esse é o valor que a empresa gasta para produzir cada unidade do produto.

Com base nesse cálculo, definimos o preço de venda do produto em R$1.750,00, o que nos coloca em um patamar competitivo no mercado, em linha com nossos concorrentes. O preço de venda é crucial para garantir que a empresa cubra seus custos e gere lucro.

Além disso, os outros termos na planilha são importantes para avaliar a rentabilidade. A margem de lucro é obtida ao dividir a margem de contribuição pelo preço do produto. A margem de contribuição é a diferença entre o preço de venda e o custo total, representando o valor disponível para cobrir os custos fixos e gerar lucro.

O markUp, por sua vez, é calculado ao dividir a margem de contribuição pelo custo total e multiplicar o resultado por 100. Ele é usado para determinar a porcentagem de markup necessária para atingir a margem de lucro desejada.

O índice de markUp é obtido dividindo o markUp por 100 e somando 1. Esse índice é valioso, pois ajuda a definir preços que levam em consideração as metas de lucratividade da empresa.

Esses indicadores nos ajudam a entender nossa estratégia de preços e a lucratividade do produto de forma abrangente, garantindo que nossos preços sejam competitivos no mercado e que atinjamos nossos objetivos financeiros.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

O gerenciamento de finanças é uma competência vital que permeia a vida de indivíduos em todas as fases, mas sua relevância é particularmente proeminente no contexto universitário. A importância do gerenciamento financeiro é inegável tanto durante a permanência na faculdade quanto após a graduação, e a aquisição de habilidades sólidas nessa área deve ser uma prioridade para todos os estudantes.

Durante a estadia na faculdade, os alunos frequentemente enfrentam desafios financeiros substanciais. A necessidade de custear mensalidades, moradia, alimentação, materiais didáticos e outras despesas relacionadas à educação é uma realidade onerosa. Nesse ambiente, a capacidade de administrar eficazmente as finanças torna-se uma necessidade premente.

A falta de controle financeiro pode prejudicar o desempenho acadêmico, aumentar os níveis de estresse e levar ao acúmulo de dívidas estudantis, que podem ter consequências adversas a longo prazo. Por isso, é importante que os estudantes adquiram habilidades de gerenciamento financeiro desde cedo, para que possam enfrentar esses desafios com sucesso, vê-se então a importância de uma unidade de estudo que visa o ensino do gerenciamento de finanças.

Além disso, a criação de um orçamento é um dos pilares fundamentais do gerenciamento financeiro na faculdade. Isso ajuda os estudantes a compreender suas entradas e saídas de dinheiro, identificar áreas de gastos excessivos e definir prioridades financeiras. Um orçamento bem elaborado oferece orientação sobre como equilibrar as despesas com as receitas, contribuindo para a responsabilidade e a disciplina financeira.

Além de ajudar os estudantes a lidar com os desafios financeiros imediatos, o orçamento também pode ser uma ferramenta valiosa para alcançar metas de longo prazo, como economizar para a aposentadoria ou comprar uma casa.

As habilidades adquiridas na faculdade continuam a ser relevantes após a graduação, uma vez que a vida pós-universidade traz consigo uma nova série de desafios financeiros. Tomar decisões sobre investimentos, comprar uma casa, planejar a aposentadoria e lidar com despesas do dia a dia são situações que exigem um sólido entendimento das finanças pessoais.

Aqueles que dominaram o gerenciamento financeiro na faculdade estão mais bem preparados para tomar decisões informadas e assertivas, o que pode levar a uma maior estabilidade financeira e prosperidade futura.

Além dos benefícios financeiros, o gerenciamento financeiro também pode proporcionar uma série de benefícios adicionais, como:

- Redução do estresse: a capacidade de controlar as finanças pode ajudar os estudantes a se sentirem mais confiantes e seguros, o que pode reduzir os níveis de estresse.
- Melhor desempenho acadêmico: o estresse financeiro pode prejudicar o desempenho acadêmico, por isso, a capacidade de gerenciar as finanças de forma eficaz pode ajudar os estudantes a se concentrarem nos estudos.
- Maior autonomia: o gerenciamento financeiro capacita os estudantes a tomar decisões sobre suas finanças, o que lhes dá uma sensação de autonomia e controle sobre suas vidas.

Por fim, o gerenciamento de finanças é uma habilidade essencial que desempenha um papel central tanto na faculdade quanto além dela. A aquisição de competências financeiras sólidas é crucial para o sucesso acadêmico, a estabilidade financeira futura e o bem-estar geral. Por isso, têm-se a importância da formação para a vida que capacita os estudantes a enfrentar os desafios financeiros ao longo de suas jornadas acadêmicas e em suas vidas pós-universitárias.

3.5.1 GERENCIANDO FINANÇAS

Falando primeiramente sobre o tópico um, esse tópico aborda os conceitos econômicos e financeiros básicos, como oferta e demanda, taxas de juros, balanço patrimonial e demonstração de resultados.

Quando pensamos em oferta e demanda, temos que pensar na oferta de um produto ou serviço onde há demanda, levando em conta o projeto integrado, nosso grupo viu a necessidade de um aparelho que substituísse a chamada tradicional, devido a diversas fraudes que ocorrem, principalmente no ambiente universitário. Além disso, taxas de juros, balanço patrimonial e a demonstração de resultados foram essenciais para o desenvolvimento do projeto, visto que foram colocados todos os gastos com o projeto no papel, assim, podendo ser visualizado de maneira melhor.

O tópico dois aborda a importância de entender o ambiente financeiro em que se vive, incluindo a independência financeira, o valor da riqueza e o registro do dia a dia.

Como já citado anteriormente, o gerenciamento financeiro é importante para diversas coisas, dentre elas, a conquista da independência financeira, que permite a capacidade de viver sem depender de um emprego, dito isso, a independência financeira está inerentemente relacionada ao valor da riqueza, visto que para alcançar a independência financeira, o

indivíduo tenha que ter ciência do valor de todos os seus bens, direitos e dívidas. Já o registo do dia a dia permite que a independência seja mantida, através do acompanhamento das finanças, podendo assim, ser identificado as áreas na vida de um indivíduo onde é possível economizar dinheiro.

O tópico três aborda os conceitos de dívidas, juros compostos, opções de empréstimo e alternativas ao endividado. As dívidas são obrigações financeiras que devem ser pagas. As dívidas podem ser contraídas para comprar bens ou serviços, ou para investir. As dívidas estão ligadas diretamente aos juros compostos, que são calculados sobre o principal e sobre os juros já acumulados, esse tipo de juros pode fazer com que as dívidas aumentem rapidamente.

Felizmente existem várias opções de empréstimo disponíveis, cada uma com suas próprias vantagens e desvantagens. Além disso, para não correr o risco de se criar novas dívidas, o endividado pode economizar dinheiro antes de comprar algo, buscar financiamento com juros baixos ou optar por bens e serviços mais baratos.

Por fim, o tópico quatro tem como tema estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos.

Para estabelecer metas financeiras, é importante definir o que se deseja alcançar, quando se deseja alcançar e quanto dinheiro é necessário para alcançar o objetivo.

Já o desenvolvimento do grupo pode ajudar a atingir metas financeiras, pois pode fornecer apoio, motivação e ideias. Por exemplo, uma pessoa pode se juntar a um grupo de poupança ou a um grupo de investimento para ajudar a alcançar suas metas financeiras.

3.5.2 ESTUDANTES NA PRÁTICA

O podcast listado abaixo foi gravado para a unidade de ensino e destaca a importância do planejamento financeiro, especialmente para universitários, abordando passos-chave, como conhecer suas finanças, estabelecer metas, criar um orçamento e evitar dívidas. Oferece dicas para estudantes, como pedir ajuda, negociar com a faculdade e buscar oportunidades de renda. Também desmente mitos comuns sobre finanças, enfatizando a necessidade de começar cedo, mesmo com pouco dinheiro, e não adiar a gestão financeira.

ISSN 1983-6767

4. CONCLUSÃO

Ao longo desse projeto, exploramos a integração de tecnologias avançadas para desenvolver um sistema de reconhecimento facial e digital conectado a um IOT, potencializado por um software desktop construído com Electron. A aplicação foi sustentada por um banco de dados MySQL, adotando princípios de programação orientada a objetos para otimizar a estruturação do código e garantir a modularidade do sistema.

O projeto teve origem na necessidade premente de aprimorar a segurança e a autenticidade dos processos acadêmicos, especialmente no contexto de instituições de ensino, onde as chamadas convencionais estavam suscetíveis a diversas formas de fraudes. A aplicação prática deste sistema não apenas oferece uma solução eficaz para substituir os métodos tradicionais, mas também representa um avanço significativo na proteção da integridade acadêmica.

A incorporação de reconhecimento facial e digital proporcionou um método robusto de autenticação, eliminando as brechas que permitiam a prática de fraudes. A interconexão com dispositivos IoT não apenas fortaleceu a segurança, mas também possibilitou a captura e a análise de dados em tempo real, otimizando a eficiência operacional.

O uso do Electron para o desenvolvimento do software desktop não só simplificou a implementação em diferentes sistemas operacionais, mas também proporcionou uma experiência de usuário consistente e intuitiva. A escolha do banco de dados MySQL promoveu eficiência no armazenamento e recuperação de dados, fundamentais para a gestão financeira e acadêmica integrada ao projeto.

A abordagem orientada a objetos adotada na programação foi crucial para criar um código organizado e modular, facilitando a manutenção e permitindo a adaptação contínua às necessidades específicas das instituições de ensino.

Em síntese, este projeto não apenas atendeu aos objetivos propostos, mas também respondeu a uma demanda crescente por métodos mais seguros e eficientes na gestão acadêmica ao substituir as chamadas convencionais por uma solução tecnológica inovadora.

ISSN 1983-6767

REFERÊNCIAS

ABREU, R. D.; OLIVEIRA, M. R.; SILVA, L. F. (2022). Uma abordagem orientada a objetos para a gestão de projetos de software. Revista de Sistemas e Computação, 12(1), 1-12.

DRUCKER, Peter. O gestor eficaz. São Paulo: Editora LTC, 1990.

GONZAGA, F. S., & BIRCKAN, G. (2000). Apostila Curso PHP e MySQL. Florianópolis, SC.

MARTINS, Eliseu. Contabilidade de Custos. São Paulo: Editora Atlas, 2010.

MILANI, André. MySQL Guia do programador. Editora: novatec, 2006.

MOON, B. Internet of Things & Hardware Industry Overview 2016. SparkLabs Global Ventures, 2016.

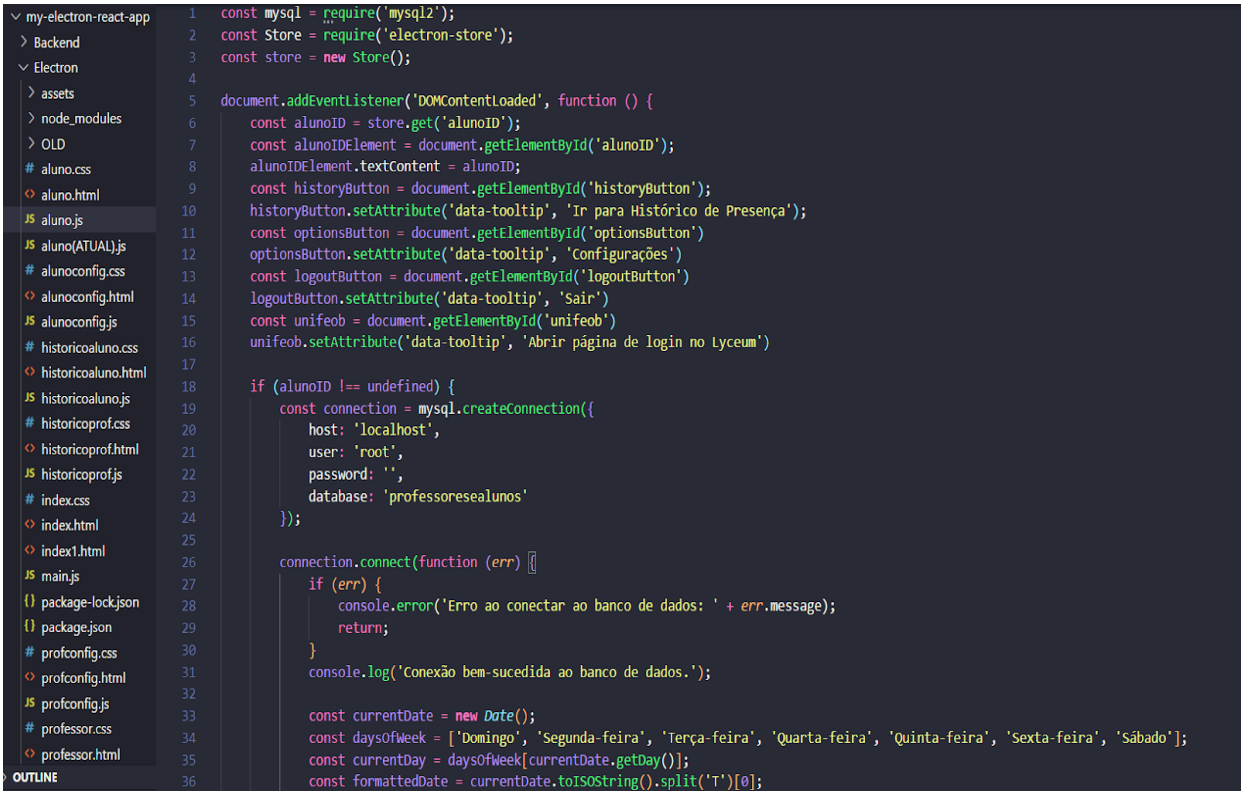
SANCHEZ, Daniel. Desenvolvimento Desktop com ElectronJS. 1ª ed. Novatec Editora, 2022.

SIPSER, Michael. Introdução à Teoria da Computação. 4ª ed. Cengage Learning, 2019.

ISSN 1983-6767

ANEXOS

- Visão geral do Visual Studio Code com a os arquivos utilizados no front-end e back-end:



Screenshot do Visual Studio Code no arquivo “Aluno.js” e mostrando os arquivos utilizados à esquerda do projeto para desenvolvimento da aplicação do SmartEye.

- Scripts de “main.js” de lógica de programação:

```
const { app, BrowserWindow, ipcMain, dialog } = require('electron');
const path = require('path');
const ElectronStore = require('electron-store');
const fs = require('fs');
ElectronStore.initRenderer();
let mainWindow;

function createWindow() {
  mainWindow = new BrowserWindow({
    width: 900,
    height: 700,
    frame: true,
    webPreferences: {
      nodeIntegration: true,
      contextIsolation: false,
      devTools: true,
      enableRemoteModule: true
    },
  });

  mainWindow.professorID = null;
  mainWindow.setMenuBarVisibility(false);
  mainWindow.loadFile('index.html');
  mainWindow.webContents.openDevTools();

  ipcMain.on('update-professorID', (event, professorID) => {
    mainWindow.professorID = professorID;
  });
}
```

```
app.whenReady().then(createWindow);

app.on('window-all-closed', () => {
  if (process.platform !== 'darwin') {
    app.quit();
  }
});

app.on('activate', () => {
  if (BrowserWindow.getAllWindows().length === 0) {
    createWindow();
  }
});
```

Screenshots do Visual Studio Code no arquivo “Main.js”.

- Scripts “index” de lógica de programação:

```
<!DOCTYPE html>
<html lang="pt">
<head>
    <meta charset="UTF-8">
    <title>Página de Login</title>
    <link rel="stylesheet" type="text/css" href="index.css">
</head>
<body>
    <img id="backgroundImage" src="./assets/login/background.png" alt="Background Image">
    <img id="bemvindo" src="./assets/login/bemvindo.png" alt ="Bem-vindo">
    <img id="loginto" src="./assets/login/loginto.png" alt="Log into">
    <img id="retangulo1" src="./assets/login/retangulo1.png" alt="Retangulo 1">
    <img id="retangulo2" src="./assets/login/retangulo2.png" alt="Retangulo 2">
    <img id="cadeado" src="./assets/login/cadeado.png" alt="cadeado">
    <img id="email" src="./assets/login/email.png" alt="E-mail">

    <form id="loginForm">
        <input type="text" id="username" placeholder="ID do Professor (a) ou Estudante:" name="username" required><br><br>

        <input type="password" id="password" placeholder="Senha:" name="password" required><br><br>

        <button type="button" id="loginButton">
            <div class="button-content">
                <span>Login</span>
                <img src="./assets/login/logar.png" alt="Login">
            </div>
        </button>
        <button type="button" id="esqueceusenha">Esqueceu sua senha?</button>
    </form>
    <p id="errorMessage" class="error-message"></p>
    <script>
        const mysql = require('mysql');
        const Store = require('electron-store')
        const store = new Store();
        const connection = mysql.createConnection({
```

```
    <script>
        const mysql = require('mysql');
        const Store = require('electron-store')
        const store = new Store();
        const connection = mysql.createConnection({
            host: 'localhost',
            user: 'root',
            password: '',
            database: 'professoresealunos'
        });

        document.getElementById('loginButton').addEventListener('click', function () {
            const username = document.getElementById('username').value;
            const password = document.getElementById('password').value;
            const errorMessageElement = document.getElementById('errorMessage');
            errorMessageElement.textContent = "";


            connection.query("SELECT * FROM alunos WHERE Login = ? AND Senha = ?", [username, password], function (err, results) {
                if (err) {
                    console.error(err);
                    return;
                }

                if (results.length > 0) {
                    console.log("Login bem-sucedido como aluno!");
                    const alunoID = results[0].AlunoID;
                    store.set('alunoID', alunoID);
                    console.log("AlunoID:", alunoID);
                    errorMessageElement.textContent = "Login efetuado com sucesso!";
                    errorMessageElement.style.color = "green";
                    window.location.href = 'aluno.html';
                } else {

                    connection.query("SELECT professorID FROM professores WHERE Login = ? AND Senha = ?", [username, password], function (err, results) {
                        if (err) {
                            console.error(err);
                            return;
                        }

                        if (results.length > 0) {
                            const professorID = results[0].professorID;
                            store.set('professorID', professorID);
                            errorMessageElement.textContent = "Login efetuado com sucesso!";
                            errorMessageElement.style.color = "green";
                            console.log("Login bem-sucedido como professor!");
                            console.log("professorID:", professorID);
                            window.location.href = 'professor.html';
                        } else {
                            console.log("Credenciais inválidas. Tente novamente.");
                            errorMessageElement.textContent = "Credenciais inválidas";
                            errorMessageElement.style.color = "red"; // Opcional: Estilo de cor vermelha para a mensagem de erro
                        }
                    });
                }
            });
        });
    </script>
</body>
</html>
```

Screenshots do Visual Studio Code no arquivo “Index.html”.

- Script em MySQL do banco de dados

```
-- Tabela `administrativo`
CREATE TABLE `administrativo` (
  `AdministrativoID` int(11) NOT NULL,
  `nome` varchar(99) DEFAULT NULL,
  `login` varchar(99) DEFAULT NULL,
  `senha` varchar(99) DEFAULT NULL,
  `perfil` blob DEFAULT NULL,
  PRIMARY KEY (`AdministrativoID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Tabela `agendamento`
CREATE TABLE `agendamento` (
  `AgendamentoID` int(11) NOT NULL,
  `ProfessorID` int(11) DEFAULT NULL,
  `MateriaID` int(11) DEFAULT NULL,
  `DiaDaSemana` varchar(20) NOT NULL,
  PRIMARY KEY (`AgendamentoID`),
  KEY `ProfessorID` (`ProfessorID`),
  KEY `MateriaID` (`MateriaID`),
  CONSTRAINT `agendamento_ibfk_1` FOREIGN KEY (`ProfessorID`) REFERENCES `professores` (`ProfessorID`),
  CONSTRAINT `agendamento_ibfk_2` FOREIGN KEY (`MateriaID`) REFERENCES `materias` (`MateriaID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;


-- Tabela `alunos`
CREATE TABLE `alunos` (
  `AlunoID` int(11) NOT NULL,
  `Nome` varchar(255) NOT NULL,
  `Login` varchar(20) NOT NULL,
  `Senha` varchar(20) NOT NULL,
  `presenca` varchar(99) DEFAULT NULL,
  `ra` int(99) DEFAULT NULL,
  `perfil` blob DEFAULT NULL,
  `nomeperfil` varchar(99) DEFAULT NULL,
  `foto` longblob DEFAULT NULL,
  `ultimoreg` varchar(99) DEFAULT NULL,
  PRIMARY KEY (`AlunoID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Tabela `cursos`
CREATE TABLE `cursos` (
  `CursoID` int(11) NOT NULL,
  `Nome` varchar(255) NOT NULL,
  `ProfessorID` int(11) DEFAULT NULL,
  PRIMARY KEY (`CursoID`),
  KEY `ProfessorID` (`ProfessorID`),
  CONSTRAINT `cursos_ibfk_1` FOREIGN KEY (`ProfessorID`) REFERENCES `professores` (`ProfessorID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```sql
-- Tabela `inscricoes`
CREATE TABLE `inscricoes` (
  `InscricaoID` int(11) NOT NULL,
  `AlunoID` int(11) DEFAULT NULL,
  `AgendamentoID` int(11) DEFAULT NULL,
  PRIMARY KEY (`InscricaoID`),
  KEY `AlunoID` (`AlunoID`),
  KEY `AgendamentoID` (`AgendamentoID`),
  CONSTRAINT `inscricoes_ibfk_1` FOREIGN KEY (`AlunoID`) REFERENCES `alunos` (`AlunoID`),
  CONSTRAINT `inscricoes_ibfk_2` FOREIGN KEY (`AgendamentoID`) REFERENCES `agendamento` (`AgendamentoID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Tabela `materias`
CREATE TABLE `materias` (
  `MateriaID` int(11) NOT NULL,
  `Nome` varchar(255) NOT NULL,
  `DiaDaSemana` varchar(20) NOT NULL,
  `ModuloID` int(11) DEFAULT NULL,
  PRIMARY KEY (`MateriaID`),
  KEY `ModuloID` (`ModuloID`),
  CONSTRAINT `materias_ibfk_1` FOREIGN KEY (`ModuloID`) REFERENCES `modulos` (`ModuloID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;


-- Tabela `modulos`
CREATE TABLE `modulos` (
  `ModuloID` int(11) NOT NULL,
  `Nome` varchar(255) NOT NULL,
  `CursoID` int(11) DEFAULT NULL,
  PRIMARY KEY (`ModuloID`),
  KEY `CursoID` (`CursoID`),
  CONSTRAINT `modulos_ibfk_1` FOREIGN KEY (`CursoID`) REFERENCES `cursos` (`CursoID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Tabela `professores`
CREATE TABLE `professores` (
  `ProfessorID` int(11) NOT NULL,
  `Nome` varchar(255) NOT NULL,
  `Login` varchar(20) DEFAULT NULL,
  `Senha` varchar(20) DEFAULT NULL,
  `perfil` longblob DEFAULT NULL,
  PRIMARY KEY (`ProfessorID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- Tabela `registro`
CREATE TABLE `registro` (
  `id` int(11) NOT NULL,
  `nome` varchar(255) DEFAULT NULL,
  `ra` varchar(10) DEFAULT NULL,
  `professorid` int(11) DEFAULT NULL,
  `diadasemana` varchar(20) DEFAULT NULL,
  `dataatual` datetime DEFAULT NULL,
  `status` varchar(99) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```