



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
<Ágora>

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

<Ágora>

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:

Carlos Gabriel dos Santos - RA 23000679

Fellipe Kozerski Silva - RA 23000710

Leticia Ramiro Abreu - RA 23000786

Lucas Guimarães Castro Nunes - RA 23000143

Matheus Farnetani Friedrich - RA 23000780

SUMÁRIO

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	7
3. PROJETO INTEGRADO	8
3.1 PROGRAMAÇÃO ORIENTADA A OBJETO	8
3.1.1 CLASSES E OBJETOS	8
3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.	8
3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS	8
3.2 LÓGICA DE PROGRAMAÇÃO	8
3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE	9
3.2.2 DESENVOLVIMENTO DE APLICAÇÕES DESKTOP	9
3.3 MODELAGEM DE DADOS	9
3.3.1 MODELO CONCEITUAL	9
3.3.2 MODELO LÓGICO E FÍSICO	9
3.3.3 SQL	9
3.4 GESTÃO FINANCEIRA	10
3.4.1 CLASSIFICAÇÃO DOS CUSTOS	10
3.4.2 CUSTOS DO PRODUTO / SERVIÇO	10
3.4.3 PRECIFICAÇÃO	10
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS	10
3.5.1 GERENCIANDO FINANÇAS	11
3.5.2 ESTUDANTES NA PRÁTICA	11
4. CONCLUSÃO	13
REFERÊNCIAS	14
ANEXOS	15

1. INTRODUÇÃO

Há muito tempo, temos estabelecido como sociedade que o “normal” é não precisar estar preocupado com a nossa segurança, e que por padrão todos os lugares são seguros, mas infelizmente essa não é mais uma realidade tão sólida.

Apenas nos anos entre 2017 e 2019, os casos de invasões e ataques a escolas aumentaram cerca de 60%, de acordo com os dados apresentados pelo Fórum Brasileiro de Segurança Pública. Recentemente, nos primeiros meses de 2023, fomos testemunhas de uma onda significativa de ataques e tentativas, a maioria dos quais não resultou em vítimas, a instituições de ensino, principalmente aquelas dedicadas ao ensino infantil e fundamental. Isso fez com que o ano de 2023 registrasse o maior aumento de ataques até o momento, gerando uma grande comoção, receio e, em alguns casos, medo na população. Como resultado, a população passou a exigir ações e posicionamentos das instituições, muitas vezes sem uma compreensão real das complexidades e do tempo necessário para implementar medidas eficazes de segurança.

Sendo assim, vimos vários movimentos no que se diz respeito ao estabelecimento e melhoria para serviços e sistemas de segurança, envolvendo a contratação e aumento na presença de seguranças particulares, implementação de sistemas de verificação e protocolos de emergência.

Eis que somos colocados à frente do problema da Unifeob, instituição de ensino superior, técnico e de capacitação, de apresentar uma solução para os pedidos de melhoria da segurança, que surgiram dos estudantes durante e após os últimos acontecimentos citados anteriormente.

Nos fazendo valer dos requisitos básicos para a construção do projeto, nosso produto é um sistema de segurança e identificação modular de baixo custo, projetado para instalação rápida. Usando a tecnologia RFID (Identificação por Radiofrequência), mapeamos o controle de acesso em tempo real de forma simples, com apenas um toque. Os cartões RFID usam ondas eletromagnéticas para acessar os dados armazenados em um microchip. Esse microchip possui uma pequena antena que emite um sinal para identificar as informações armazenadas em um banco de dados MySQL. Cada pessoa cadastrada terá um cartão exclusivo sincronizado com seu perfil de cadastro.

Entre outros objetivos estão:

- Aumentar o sentimento de segurança por parte dos estudantes, adicionando mais uma camada de prevenção contra invasão e violência.
- Tornar a estrutura da Unifeob mais interativa, com mais praticidade e agilidade no uso de diferentes espaços.
- Facilitação do processo de presença dos estudantes (chamada/faltas), para diminuição de presenças falsas, e identificação de estudantes que adentram o campus, mas não as salas de aula.
- Para aqueles estudantes que estão com um número de faltas alto o sistema poderá enviar um Email alertando o mesmo da sua situação.
- Ter um controle sobre o fluxo de pessoas em determinados espaços da instituição, utilizando o banco de dados. Podendo assim saber se o ambiente está sendo bem aceito pelos estudantes, e se é viável a existência daquilo.

Nós, da empresa Ágora, fomos fundados no início de 2023 e atualmente nos dedicamos a oferecer soluções para problemas de segurança e logística. Nosso objetivo é solucionar desafios do cotidiano por meio de ideias inovadoras, visando aprimorar a segurança física e local. A missão da Ágora é ouvir as necessidades das pessoas, identificar problemas que afetam grandes grupos e encontrar soluções eficazes e abrangentes, oferecendo suporte ao longo de todo o período de contratação.

Diante do problema apresentado, surge a seguinte pergunta: por que é importante reestruturar os sistemas de segurança? Essa questão pode ser respondida de forma mais prática quando consideramos o impacto na imagem do ambiente perante seus stakeholders e a comunidade em geral. Em outras palavras, a reestruturação dos sistemas de segurança não é apenas uma questão técnica, mas também uma estratégia de marketing e construção de uma imagem pública positiva.

Desta forma, no trabalho a seguir, descreveremos todos os passos seguidos para a estruturação do modelo lógico por trás do modelo físico. Isso envolverá cada Unidade de Estudo trabalhada neste semestre, desde a identificação da problemática com a proposta de solução, passando pela estruturação de cada código por trás da interface de design até o banco de dados, onde todos os dados serão armazenados e poderão ser consultados em tempo real.

2. DESCRIÇÃO DA EMPRESA

Fundada em 1965, a Unifeob, Centro Universitário Fundação de Ensino Octávio Bastos, sempre teve como intuito e principal objetivo a disseminação da educação de qualidade na região e o estabelecimento de uma padrão de ensino de alto nível.

Inscrita sob o CNPJ 59.764.555/0001-52, a instituição privada é uma referência no mercado de educação superior, disponibilizando cursos de graduação, tecnólogos, pós-graduação, especializações, gestão empresarial e mais.

Se localiza na cidade de São João da Boa Vista, interior de São Paulo, na avenida que leva o mesmo nome do fundador da instituição, Av. Dr. Octávio da Silva Bastos, Jardim Nova São João, próximo à Avenida do Mantiqueira.

Com estrutura modelo e reconhecida com IGC (Índice Geral de Cursos) nota 4, a IE se coloca como uma das melhores universidades da região, capacitando estudantes que hoje se destacam em empresas como Nubank, Red Bull, Ferrero Rocher, entre outras.

3. PROJETO INTEGRADO

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

A Programação Orientada a Objetos revolucionou a maneira como desenvolvemos software, concentrando-se na organização do código em torno de objetos, que são representações de entidades do mundo real, e em suas interações. Antes do advento da POO, a programação era fundamentada em paradigmas procedimentais. O código era estruturado em procedimentos e funções dedicados à execução de tarefas específicas. Não havia uma ênfase significativa na organização em torno de objetos ou entidades, o que gerava alguns malefícios, tais como:

- **Reutilização de Código:** A reutilização de Códigos era uma tarefa árdua e muitas vezes ineficiente para os desenvolvedores. A falta de encapsulamento na programação procedural dificultava a criação de módulos independentes e reutilizáveis pois
- As variáveis e funções frequentemente eram globais ou acessíveis a partir de qualquer lugar no programa. Isso dificultava a criação de módulos independentes, uma vez que variáveis globais podem ser modificadas de forma inesperada.
- **Falta de Abstração:** O paradigma procedural não incentivava a criação de estruturas de dados complexas que representassem objetos do mundo real, isso dificultava a encapsulação de funcionalidades desses objetos em componentes reutilizáveis.
- **Reescrita de Código:** Quando os desenvolvedores desejavam realizar tarefas semelhantes em diferentes partes do programa, frequentemente precisam reescrever o código ou copiá-lo, o que aumentava a chance de erros e tornava o código mais difícil de manter.

Ao discutir as vantagens da Programação orientada a objetos (POO), o renomado especialista em engenharia de software, Dr. Grady Booch, enfatiza que a 'abordagem orientada a objetos não apenas permite a realização eficiente do código, mas também facilita a modelagem intuitiva de sistemas complexos, proporcionando maior modularidade e manutenibilidade (BOOCH, 2007).

3.1.1 CLASSES E OBJETOS

Em POO, uma classe é uma estrutura que define um conjunto de atributos e métodos que objetos daquela classe possuirão. As classes servem como modelos para criação de objetos e ajudam a estruturar o código e a representar entidades do mundo real de forma organizada. Observe nosso exemplo prático:

Figura 1 - Tabelas

```
class Registers(db.Model):  
    __tablename__ = "registers"
```

Fonte: Autores

A classe ‘Registers’ tem como finalidade construir um objeto que represente a entidade “Registros” do banco de dados junto com a biblioteca SQLAlchemy, onde se é possível construir um objeto de mapeamento relacional que permite a criar um banco de dados a partir de linhas de código. Em primeiro momento, é necessário a criação dos objetos, definindo o nome da tabela e seus atributos para que assim, construir uma *engine* e um *session maker* para utilizar os metadados para construir um banco de dados em qualquer opção SQL disponível. A biblioteca também permite a automatização na construção de *queries* através de funções. Assim, com uma maneira simples de referenciar a tabela do banco de dados, e construções de queries é possível garantir um uso eficiente e simplificado, pois com uma linha de código é possível extrair um dicionário contendo os nomes dos atributos da tabela e seus respectivos valores, prontos para serem transformados em JSON e exibidos na rota “/api/<número do gráfico>” e “/api/tabela”.

Também em um exemplo fictício onde criamos uma classe de Livro podemos adicionar atributos a classe como , editora, livro, ano.e métodos como ‘verificarDisponibilidade’ e ‘adicionarCopias’ que representam respectivamente funções do cotidiano de um aluguel de livro, no nosso caso adicionamos os seguintes atributos:

Figura 2 - Atributos

```

# Attributes
id: Mapped[int] = mapped_column(Integer(), primary_key=True, autoincrement=True, nullable=False)
date: Mapped[Date] = mapped_column(Date, nullable=False)
hour: Mapped[int] = mapped_column(Integer(), nullable=False)
minute: Mapped[int] = mapped_column(Integer(), nullable=False)
second: Mapped[int] = mapped_column(Integer(), nullable=False)

# Many-to-one relation with 'cards'
card_id: Mapped[int] = mapped_column(Integer(), ForeignKey("cards.id"), index=True)
card: Mapped["Cards"] = relationship("Cards", back_populates="register")

# Many-to-one relation with 'equipments'
equipment_id: Mapped[int] = mapped_column(Integer(), ForeignKey("equipments.id"), index=True)
equipment: Mapped["Equipments"] = relationship("Equipments", back_populates="register")

```

Fonte: Autores

A tabela de registros representa o registro gerado pelos cartões RFID. Aqui é necessário armazenar as informações referentes ao tempo, ou seja, dia, hora, minuto e segundo. Nas linhas seguintes a definição destas variáveis com seus tipos de dados mapeados, é construído a relação da tabela registros com a tabela cartões. Primeiro é necessário criar o atributo que irá conter o valor e depois constrói-se a relação com a tabela cartões populando a variável *register* dentro da tabela *cards* e a mesma lógica é aplicada, relação às variáveis *equipment_id* e *register* na tabela equipamentos.

Um objeto é uma instância específica de uma classe. Cada objeto possui sua própria cópia dos atributos da classe e pode invocar os métodos definidos na classe. Os objetos são entidades que representam as coisas do mundo real em um sistema de software, são instâncias de classes, incorporando detalhes e características específicos de determinada entidade que estão representando podemos levantar como exemplo um:

Sistema de Gerenciamento de Identificação RFID:

- Uma classe chamada 'CartaoRFID' poderia ser criada para representar os cartões RFID usados no sistema. Cada Objeto 'CartaoRFID' conteria informações como número de identificação e status de acesso.
- Uma classe chamada 'Case' poderia ser criada para representar os cartões RFID usados no sistema. Cada objeto 'Case' conteria informações como localização e estado de operação.
- Uma classe chamada 'Usuario' poderia ser criada para representar os usuários do sistema. Cada objeto 'Usuário' conteria informações e histórico de acesso.

Nesse cenário, os objetos 'CartaoRFID', 'Case' e 'Usuario' seriam instâncias das suas respectivas classes, com cada objeto mantendo os dados relacionados a essas entidades do mundo real. Quando objetos são modelados com precisão, o software se torna uma

representação fiel do sistema de identificação RFID e o acesso a isso torna mais fácil entender,manter e estender o sistema. A interação entre esses objetos,por meio de métodos e atributos ,permite a simulação de processos reais,como monitoramento de acesso,registro de histórico de acesso e gerenciamento de cartões RFID

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.

Atributo: Atributos desempenham um papel fundamental na representação que pertencem a uma classe e representam a característica e propriedades dos objetos criados a partir de uma classe.Você deve definir atributos como variáveis que serão usadas para representar características de um objeto como por exemplo:

```
class Pessoa:
    def __init__(self,nome,idade):
        self.nome = nome
        self.idade = idade
```

Como exemplificado, a classe pessoa tem dois atributos,'nome' e 'idade',que são inicializados quando um objeto da classe é criado no código.

Os atributos desempenham um papel vital na representação do estado de objetos em programação orientada a objetos. eles permitem que você modele e manipule dados de forma eficaz,tornando a POO uma abordagem poderosa para o desenvolvimento de software.

Métodos: Os métodos são como habilidades ou ações que um objeto pode executar,muito semelhante às funções.Eles representam o comportamento de um objeto e permitem que os objetos interajam um com os outros,realizando operações e modificando estados.

```
class Calculadora:
    def somar (self,a,b):
        return a + b
```

A classe Calculadora tem um método chamado 'somar',que aceita dois argumentos,'a' e 'b',e retorna sua soma.Os Métodos são essenciais pois permitem que os objetos realizem ações e interajam uns com os outros tornando a modelagem do comportamento de objeto mais eficaz.

Encapsulamento: O encapsulamento envolve o ocultamento dos detalhes internos de um objeto e o fornecimento de uma interface controlada para interagir com ele. ele desempenha um papel crucial na proteção dos atributos e métodos de uma classe,promovendo

a segurança e a integridade dos dados. Em muitas linguagens como Python existem modificadores de acesso que definem o nível de visibilidade dos atributos e métodos. Os três principais modificadores são:

- **Public:** Atributos e métodos públicos podem ser acessados de qualquer lugar fora da classe. Não há restrições de acesso.
- **Private:** Atributos e métodos privados são acessíveis apenas dentro da própria classe, eles não podem ser acessados fora da classe.
- **Protect:** Atributos e métodos protegidos são semelhantes aos privados mas podem ser acessados por classes derivadas (herança).

O encapsulamento, com seu poder de proteção dos dados e o controle sobre o comportamento dos objetos, não é apenas um pilar da (POO), é o alicerce de sistemas de softwares que requerem segurança. Ele permite que se esconda os detalhes internos e complexos, garantindo que apenas as interfaces designadas possam interagir com objetos, evitando assim o acesso não autorizado e modificações indesejadas. Em um mundo onde a segurança e integridade de dados são inestimáveis, o encapsulamento se destaca como guardião da confiabilidade.

Herança: A herança é um conceito-chave na programação orientada a objetos que permite que uma classe herde atributos e métodos de outra classe, onde uma classe filha ou subclasse herda características da classe pai ou superclasse. Esta relação “é um” entre classes é uma forma poderosa de reutilizar o código e criar modelos que refletem a estrutura do mundo real.

A herança é frequentemente usada para estabelecer uma hierarquia de classes, onde uma classe filha herda os atributos e métodos de sua classe pai. A classe pai, por sua vez, pode ser considerada mais geral, enquanto as classes filhas são mais específicas e especializadas. Isso permite a criação de uma estrutura hierárquica de objetos que compartilham funcionalidades comuns e, ao mesmo tempo, podem ter comportamento específico. Podemos realçar alguns benefícios trazido pela Herança tais como:

- **Reutilização de Código:** A herança permite que você herde e reutilize funcionalidades existentes, evitando a duplicação de código.
- **Modelagem Hierárquica:** Ela possibilita a criação de modelos hierárquicos que refletem a relação “é um” entre objetos do mundo real

Polimorfismo: A herança também está intimamente ligada ao conceito de polimorfismo. O polimorfismo permite que objetos de diferentes classes sejam tratadas de maneira uniforme, desde que compartilhem métodos ou interfaces comuns. Isso promove a

flexibilidade e a extensibilidade do código, permitindo que as subclasses substituam o comportamento da superclasse se necessário. O poliformismo também é uma ferramenta poderosa para a extensibilidade do código. Quando novas subclasses são adicionadas à hierarquia de classes, elas podem herdar e compartilhar métodos e interfaces comuns. Isso permite que o sistema seja facilmente estendido sem modificar o código existente. O polimorfismo promove o princípio do “aberto para extensão, fechado para modificações”, que é um dos princípios fundamentais do design orientado a objetos.

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Na POO, os métodos desempenham um papel central na definição do comportamento de objetos. Além dos métodos convencionais que realizavam ações sobre objetos, há três categorias importantes de métodos que contribuem para a organização e controle do comportamento: métodos públicos, métodos privados e métodos estáticos.

- **Métodos Públicos:** representam a interface principal para interagir com objetos. Eles são acessíveis de fora da classe e são projetados para fornecer funcionalidades específicas, permitindo que os usuários da classe realizem operações ou acessem informações sem se preocupar com os detalhes internos.
- **Métodos Privados:** São destinados a serem usados internamente dentro da classe e não devem ser acessados externamente. Eles são essenciais para encapsular a lógica interna e proteger o estado dos objetos, evitando o acesso direto a funcionalidades sensíveis.
- **Métodos Estáticos:** São métodos que não dependem de instâncias de classe e, portanto não tem acesso aos atributos de instância. Eles são chamados diretamente na classe, em vez de objetos específicos, e são frequentemente utilizados para funções de utilidade ou métodos que não precisam de acesso a atributos de instância.

Compreender quando e como aplicar essas categorias de métodos é fundamental para criação de classes bem estruturadas e sistemas de software que sejam coesos, seguros e eficazes. Portanto, a consideração cuidadosa e a aplicação estratégica de métodos estáticos, públicos e privados são a chave para construção de sistemas de software que atendam às necessidades de hoje e que possam se adaptar e evoluir para enfrentar futuros desafios.

3.2 LÓGICA DE PROGRAMAÇÃO

A lógica de programação consiste em um conjunto de princípios e regras que guiam a criação de algoritmos e a resolução de problemas por meio da programação. Essa lógica atua como alicerce para a elaboração de código em diversas linguagens de programação. Uma vez construído o algoritmo, é possível codificar o programa em diferentes tipos de linguagens, incluindo JavaScript. Uma lógica de programação sólida é fundamental para o desenvolvimento de código eficiente e sustentável. Ao estruturar algoritmos de maneira clara e organizada, os programadores facilitam a compreensão do código, tornando-o mais fácil de dar manutenção e menos propenso a erros. Além disso, uma boa lógica contribui para a eficiência de software, otimizando o consumo de recursos e promovendo um desempenho mais eficaz. O exercício cognitivo colabora e aprimora a resolução de problemas permitindo a o desenvolvedor abordar desafios de forma sistemática e criar soluções mais eficazes.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

O desenvolvimento de software, também caracterizado pela criatividade e lógica, representa um processo complexo que exige a integração cuidadosa dessas características. Dentro deste contexto, é interessante compreender os conceitos fundamentais que compõem o pilar para criação de um programa eficaz. Alguns deles são:

- **Algoritmos:** São sequências de passos organizados para resolver um problema específico. Eles são independentes de linguagens de programação e representam a estrutura lógica de um programa;
- **Estruturas de Controle:** Em JavaScript, assim como em muitas outras linguagens, utiliza-se estruturas de controle para tomar decisões e controlar o fluxo de execução do programa. As estruturas de controle, como condicionais (if/else) e loops (for/while), são fundamentais para implementar a lógica de um programa;
- **Variáveis e Tipos de Dados:** Para manipular informações em JavaScript, você precisa entender como trabalhar com variáveis e tipos de dados. A lógica de programação envolve a escolha adequada de variáveis e a manipulação correta dos tipos de dados para realizar operações desejadas;
- **Funções:** Funções são blocos de código reutilizáveis que realizam tarefas específicas. A lógica de programação ajuda na criação de funções eficientes e na compreensão de como elas se encaixam no programa como blocos de construção lógicos;
- **Estruturas Condicionais:** As estruturas condicionais são usadas para controlar o fluxo de um programa, permitindo que você execute blocos de código com base em condições específicas. No JavaScript, as estruturas condicionais mais comuns são:
 - If:** Utilizado para executar um bloco de código se uma condição for avaliada como verdadeira;

Elif: Usado após um `if` para verificar uma condição alternativa, executando um bloco de código se a condição anterior for falsa, mas a condição atual for verdadeira;

Else: Utilizado para definir um bloco de código que será executado se todas as condições anteriores forem avaliadas como falsas.

- **Operadores de Comparação:** Os operadores de comparação são usados para comparar valores e retornar um valor booleano (verdadeiro ou falso) que indica se a comparação é verdadeira ou falsa. Alguns operadores de comparação em JavaScript incluem:

== (igual a): Verifica se dois valores são iguais;

!= (diferente de): Verifica-se dois valores não são iguais;

=== (estritamente igual a): Verifica se dois valores são estritamente iguais;

!== (estritamente diferente de): Verifica-se dois valores são estritamente diferentes;

> (maior que): Verifica se o valor da esquerda é maior que o valor da direita;

< (menor que): Verifica se o valor da esquerda é menor que o valor da direita;

>= (maior ou igual a): Verifica se o valor da esquerda é maior ou igual ao valor da direita;

<= (menor ou igual a): Verifica se o valor da esquerda é menor ou igual ao valor da direita.

- **Operadores Lógicos:** Os operadores lógicos são usados para combinar ou modificar condições em expressões lógicas. No JavaScript, os operadores lógicos mais comuns são:

&& (E lógico): Retorna verdadeiro se ambas as condições à esquerda e à direita forem verdadeiras;

|| (OU lógico): Retorna verdadeiro se pelo menos uma das condições à esquerda ou à direita for verdadeira;

! (NÃO lógico): Inverte o valor de verdadeiro para falso e vice-versa, aplicado a uma única condição.

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES WEB

A ferramenta de desenvolvimento Flask é uma biblioteca escrita na linguagem de programação Python que permite a criação de aplicativos mínimos, fazendo com que o desenvolvedor tenha que adicionar as bibliotecas complementares, que hoje são desenvolvidas por equipes separadas, algo que antes estava centralizado. Por exemplo, caso queira automatizar a criação de um código HTML de um formulário, é possível adicionar a biblioteca WTForms, que através de definição de classes é possível realizar esta tarefa, sendo necessário apenas a “chamar” a função que verifica a classe, no momento em que a rota recebe um POST.

Um aplicativo desenvolvido em Flask, pode ter inúmeras funções, como por exemplo uma aplicação web criada para exibir dados obtidos de leitores de cartões RFID. A ferramenta

permite a construção de páginas HTML com o uso da ferramenta de *templates* Jinja e junto das bibliotecas complementares do Flask torna muito mais fácil a criação das interfaces.

A ferramenta também permite a definição de rotas e seus respectivos métodos, onde é possível processar informações introduzidas pelo usuário e retorná-las para que o Jinja possa renderizar um *template* com a informação. O Flask também permite a modularização da estrutura do aplicativo quando utilizado o conceito de *blueprint* da ferramenta, onde cria-se múltiplos arquivos de definição de rotas e seus respectivos códigos necessários para o funcionamento.

3.3 MODELAGEM DE DADOS

A Modelagem de Dados desempenha um papel fundamental na criação de modelos de bancos de dados. Dominar a habilidade de organizá-los adequadamente para análise é crucial para garantir a excelência e eficácia do sistema. Os sistemas resultantes da modelagem de dados têm como objetivo principal o armazenamento de dados, a criação de conexões entre elementos nos bancos de dados e a análise de informações de acordo com os requisitos de seu design.

É importante destacar que a modelagem de dados é indispensável, pois esse processo estabelece todos os atributos necessários para que o banco de dados armazene informações relevantes que podem gerar insights valiosos para a empresa. O objetivo principal desse processo é criar representações claras e compreensíveis dos dados, tornando-os úteis para a tomada de decisões e a gestão de informações. Essas representações são desenvolvidas como parte do projeto de banco de dados, podendo ocorrer tanto antes quanto depois da coleta de dados.

A modelagem de dados envolve a criação de um diagrama simplificado com o objetivo de estruturar o armazenamento e o fluxo de informações relevantes para a construção de um banco de dados. Para isso, são utilizados diversos métodos, incluindo diagramas, texto e símbolos, para criar representações virtuais. Essa estrutura deve incluir, por exemplo, os formatos que os dados terão e as relações que estabelecem entre si, bem como outros aspectos lógicos e funcionais. Esse procedimento é realizado antes da construção do sistema de banco de dados e requer um estudo dos requisitos para compreender a finalidade da modelagem e como o banco de dados será utilizado.

Em resumo, ao criar um modelo de dados completo e otimizado, é possível desenvolver um banco de dados lógico e eficiente. Isso ocorre porque o modelo auxilia na eliminação de redundâncias nos dados, na redução dos requisitos de armazenamento e na facilitação da recuperação eficiente de informações quando necessário. Para alcançar esse

objetivo, é importante seguir um processo que geralmente envolve quatro etapas: análise de requisitos, modelagem conceitual, modelagem lógica e modelagem física.

Um banco de dados é um sistema projetado para armazenar e gerenciar dados que estão relacionados e tratam do mesmo assunto ou tema. Em outras palavras, um banco de dados serve como um repositório centralizado para coletar, armazenar e recuperar informações relevantes. Esses dados podem ser organizados de forma eficiente para atender às necessidades específicas de uma pessoa, equipe ou organização, além de poderem ser administrados e acessados por indivíduos, equipes ou até mesmo por uma empresa, dependendo da finalidade e do uso pretendido do banco de dados.

É uma ferramenta essencial em muitos campos, pois permite a gestão eficaz de informações que são cruciais para tomar decisões informadas e realizar operações comerciais de maneira eficiente. Com base nas informações teóricas sobre modelagem de dados e bancos de dados, desenvolvemos um modelo de dados abrangente. Isso nos permitiu construir no MySQL um banco de dados eficaz para rastrear o controle de fluxo de pessoas em uma área específica da instituição. Nosso sistema estabelece a conectividade em tempo real com um sistema de identificação que controla a entrada, permanência e saída das pessoas por meio de cartões ou tags RFID, implementado com a tecnologia IoT.

3.3.1 MODELO CONCEITUAL

Modelos de dados conceituais oferecem uma visão geral dos dados onde elabora uma representação abstrata de um sistema, processo ou conjunto de informações. Durante essa fase, criamos um diagrama que representa visualmente todas as regras de negócio estabelecidas na análise de requisitos. Basicamente, esse diagrama mostra as funcionalidades essenciais do sistema.

Um diagrama de modelagem conceitual deve abranger quatro elementos essenciais: Entidades, Relacionamentos, Cardinalidade e Atributos. Esses elementos são fundamentais para definir o que será armazenado no sistema de banco de dados. Abaixo, fornecerei as definições de cada um desses elementos e demonstrarei como eles se encaixam em nosso modelo conceitual.

- **Entidades:** São objetos, conceitos ou itens importantes em um domínio ou sistema, é a forma como as informações se relacionam dentro do sistema.
 - Lugares, equipamentos, Arduinos, registros, cartões, usuários, estudantes, colaboradores e visitantes.

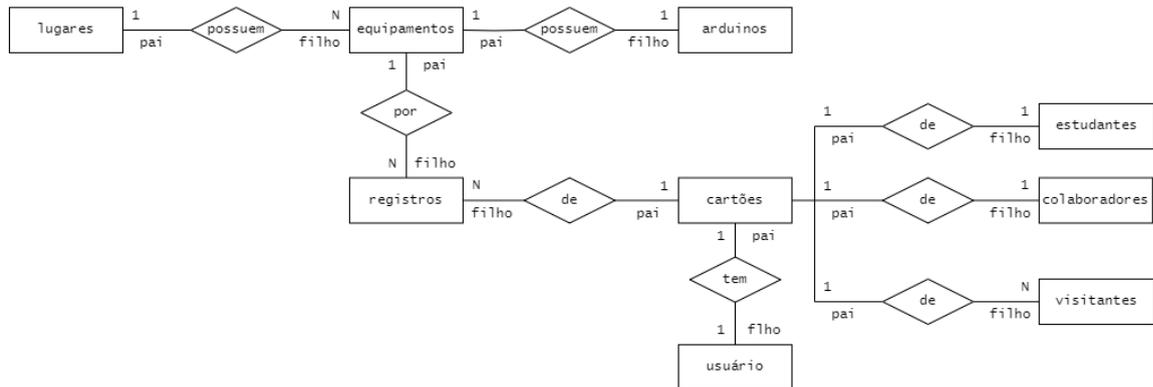
- **Relacionamento:** Representam as conexões ou vínculos entre diferentes entidades. Eles mostram como as entidades se conectam e interagem no contexto do sistema.
 - Lugares - possui - equipamentos - possui - arduinos.
Cada lugar terá um equipamento que conterà um Arduino.
 - Equipamentos - por - registros - de - cartões - tem - usuários.
Cada equipamento terá acesso aos registros de cada cartão de cada usuário.
 - Usuários - de - estudantes, colaboradores e visitantes.
São os usuários, estudantes, colaboradores e visitantes.
- **Cardinalidade:** Descreve quais podem ser os relacionamentos de uma entidade com outras entidades, incluindo um para um, de um para muitos ou de muitos para muitos.
 - (1) lugar possui (MUITOS) equipamentos. E (1) equipamento possui (1) arduino.
 - (1) equipamento possui (MUITOS) registros. E (1) cartão possui (MUITOS) registros.
 - (1) cartão possui (1) usuário.
 - (1) cartão pode possuir (1) estudante OU pode possuir (1) colaborador OU pode possuir (MUITOS) visitantes.
- **Atributos:** São características ou propriedades das entidades. Eles descrevem informações específicas relacionadas a uma entidade.
 - Lugares: ID - Descrição.
 - Equipamentos: ID - Descrição - Tipo - Última inspeção - Próxima inspeção.
 - Arduinos: ID - Descrição - Versão código.
 - Registros: ID - Data - Hora - Minuto - Segundo.
 - Cartões: ID - UID - Tipo - Tem usuário.
 - Usuários: ID - Username - Email - Tipo - Hash.
 - Estudantes: ID - Nome - Nascimento - Curso - Ano ingresso.
 - Colaboradores: Nome - Nascimento - Setor - Turno começa - Turno termina.
 - Visitantes: Nome - Nascimento - Documento.

Uma forma comum de representar modelos conceituais é por meio do "Diagrama Entidade-Relacionamento" (DER ou ERD, na sigla em inglês). Esse tipo de diagrama emprega símbolos e linhas para visualizar entidades, seus atributos e as conexões entre elas.

O diagrama visual é essencial para apresentar de maneira clara e objetiva a visão de negócios do projeto, servindo como alicerce para as próximas etapas na modelagem de um sistema de banco de dados.

Nessa fase, analisamos as informações que precisam ser armazenadas no banco de dados. Além disso, definimos as chaves primárias, ou Primary Keys (PK), que são identificadores únicos para cada registro na tabela.

Figura 1 - Modelo Conceitual.



Fonte: Elaborados pelos autores.

Figura 2 - Registro.

arduinos	cartões	colaboradores	equipamento	lugares
PK <u>UniqueID</u>				
descricao	uid	nome	descricao	descricao
versao_codigo	tipo	nascimento	tipo	
	tem_usuario	setor	ultima_inspecao	
		turno_comeca	proxima_inspecao	
		turno_termina		
registros	estudantes	usuarios	visitantes	
PK <u>UniqueID</u>	PK <u>UniqueID</u>	PK <u>UniqueID</u>	PK <u>UniqueID</u>	
data	nome	username	nome	
hora	nascimento	email	nascimento	
minuto	curso	tipo	documento	
segundo	ano_ingresso	hash		

Fonte: Elaborados pelos autores.

3.3.2 MODELO LÓGICO E FÍSICO

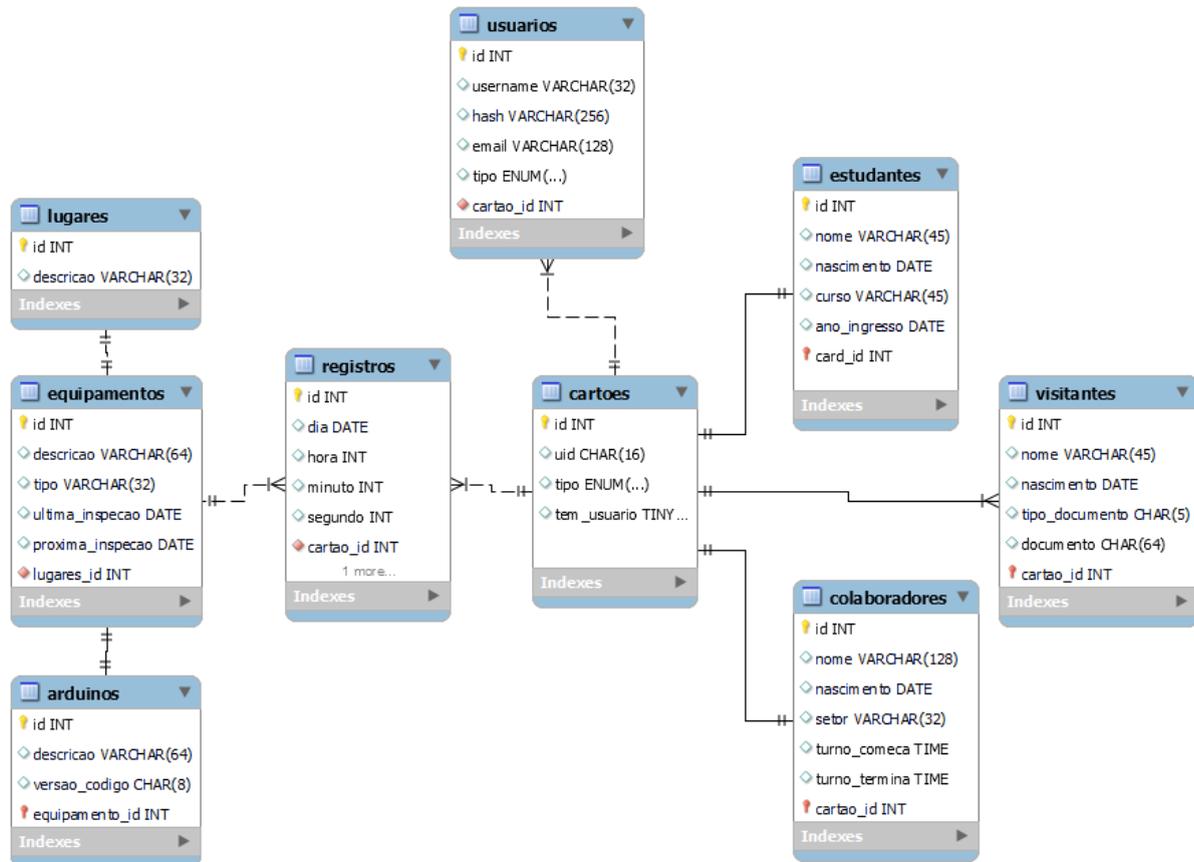
Entre os softwares de gerenciamento de banco de dados, um dos mais populares é o MySQL. O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD), que é um

software que possibilita a interação com os dados armazenados em um banco. Através de um SGBD, é possível manipular e acessar os dados armazenados em bancos de dados. O MySQL utiliza a linguagem SQL (Structured Query Language - Linguagem de Consulta Estruturada) como interface. O MySQL oferece soluções confiáveis e escaláveis para a organização eficiente e segura de grandes volumes de dados.

No modelo lógico, a estrutura dos elementos de modelagem se torna mais detalhada. Nessa fase, não apenas criamos tabelas, mas também especificamos as colunas, os tipos de dados, as chaves primárias e estrangeiras, bem como as restrições. Além disso, estabelecemos minuciosamente os relacionamentos entre as tabelas. Os três componentes principais deste modelo são Entidades, Relacionamentos e Atributos. Este modelo vai além do modelo conceitual, adicionando mais informações. Ele serve como um guia detalhado para os dados usados. O modelo de dados lógicos incorpora todos os elementos de informação essenciais para o funcionamento cotidiano de acordo com sua finalidade.

Neste nível, estabelecemos regras de integridade e normalização para garantir a consistência dos dados. A normalização também ajuda a reduzir a redundância de dados, melhorando a eficiência do sistema. Resumidamente, o modelo de dados lógicos define como o sistema deve ser implementado, independentemente do sistema de gerenciamento de banco de dados utilizado. Isso resulta em um mapa altamente técnico das regras e estruturas de dados subjacentes.

No modelo de dados físico, o modelo conceitual e lógico é traduzido e implementado em um banco de dados real usando a linguagem escolhida. Nesse estágio, escolhemos qual sistema de gerenciamento de banco de dados (SGBD) será utilizado, com base no modelo adotado, para garantir que atenda aos requisitos planejados e suporte eficazmente às necessidades de armazenamento de dados. Além disso, é nesta fase que estabelecemos as regras de segurança, determinando quem terá acesso a cada informação e dado gerado no banco de dados, bem como definindo procedimentos de segurança para os dados armazenados.

Figura 3 - Banco de Dados.

Fonte: Elaborados pelos autores.

3.3.3 SQL

Os estudantes devem utilizar comandos básicos de SQL (Insert, Update, Delete e Select) para popularizar o banco de dados, realizar testes e criar consultas. Tais comandos também serão utilizados nas unidades de programação.

3.4 GESTÃO FINANCEIRA

O conceito de gestão pode ser resumido como a busca pelo alcance de metas. O desafio central está em estabelecer metas eficazes que levem aos resultados estratégicos necessários para o desenvolvimento empresarial. Falconi (2017) acrescenta que a alta administração é viável apenas quando existe uma visão clara do destino desejado. Enfrentar

problemas requer um processo de análise detalhada para compreender suas raízes e resolvê-los de forma consciente, evitando consequências graves. Portanto, é essencial estabelecer metas bem definidas, orientadas para resultados positivos, e comunicá-las a todos os envolvidos na empresa e na organização, incentivando-os a contribuir para alcançar esses resultados.

Segundo Falconi (2017)

Diante de um problema, as empresas já querem a solução, pulam todo o processo de análise das causas, entendimento das razões do problema e já querem ir pra solução. Essas soluções na maioria das vezes não são as mais acertadas, no final você tem métodos que custam caro e não resolvem o problema.

Em gestão financeira, abordamos vários conceitos fundamentais relacionados à administração eficaz dos recursos financeiros de uma empresa. Compreendendo esses conceitos e seus propósitos, somos capazes de tomar decisões mais responsáveis que visam a um retorno financeiro lucrativo, ao mesmo tempo em que minimizem os riscos associados. Consequentemente, os princípios básicos da gestão financeira são cruciais e devem ser rigorosamente seguidos para garantir o sucesso.

3.4.1 CLASSIFICAÇÃO DOS CUSTOS

A classificação de custos é o processo de categorizar despesas e custos em diferentes grupos ou categorias com base em critérios específicos. É de extrema importância compreender como esses gastos estão sendo alocados, a fim de facilitar o planejamento, controle de despesas e análise de rentabilidade para a tomada de decisões estratégicas.

→ **Custo Direto:** O custo direto está diretamente relacionado ao produto final, sendo facilmente atribuível sem a necessidade de rateio, pois está ligado diretamente à produção. Os componentes abaixo são essenciais para o funcionamento do sistema RFID e podem ser atribuídos diretamente a esse projeto.

Exemplo de custo direto no contexto do nosso projeto são: Kit RFID+Arduino, Case (filamento 3d), Servidores, Cabeamento, Licenças de software, Salário dos funcionários dedicados ao projeto, Pasta de Solda, Equipamentos e ferramentas, Servidores e Compra de mais 'TAGS' e Cartões RFID.

→ **Custo Indireto:** Os custos indiretos são aqueles que não podem ser atribuídos diretamente a um produto ou serviço específico, uma vez que não é possível mensurar de forma precisa o custo do produto, mas ainda são necessários para o funcionamento geral da empresa ou projeto. Eles são compartilhados entre vários projetos ou atividades e não podem ser alocados diretamente a nenhum deles.

Exemplo de custo direto no contexto do nosso projeto são: Marketing e Depreciação de equipamentos.

- Custos Fixos: São aqueles que não variam com a produção ou vendas de produtos ou serviços. Esses custos permanecem constantes ao longo do tempo e não dependem do nível de atividade da empresa.

Exemplo de custo direto no contexto do nosso projeto são: Manutenção de equipamentos, Internet, Aluguel e Energia Elétrica.

- Custo Variável: São aqueles que consideramos mais imprevisíveis e sujeitos a oscilações de preço. Eles dependem do resultado da produção e não é tão fácil estipular um valor específico, já que podem sofrer alterações periódicas devido ao aumento ou diminuição na produção. Portanto, são custos que variam de acordo com a quantidade produzida ou do nível de atividade da empresa.

Exemplo de custo direto no contexto do nosso projeto são: Impostos, Manutenção Posterior à Instalação e Contratação de terceiros.

3.4.2 CUSTOS DO PRODUTO

O custo total do nosso projeto foi de R\$5.730,74. Esse valor se baseia em um único plano, em comparação com o produto do nosso concorrente, que é dividido em três planos, sendo que um deles pode chegar a R\$33.500,00.

Na tabela abaixo, classificamos os custos diretos e indiretos do nosso produto, juntamente com os valores dos custos que a nossa empresa incorre.

Tabela 1 - Custos Diretos e Indiretos.

Custos diretos				Custos Indiretos	
Produto	Custo	Unidades	Total	Produto	Custo
Kit RFID+Arduíno	R\$ 151,90	150	R\$ 28.481,25	Marketing	R\$ 2.000,00
		37,5			
Case (filamento 3d)	R\$ 600,00	-	R\$ 600,00	Depreciação de equipamentos	R\$ 150,00
Servidores	R\$ 100,00	-	R\$ 100,00		
Cabeamento	R\$ 15.000,00	-	R\$ 15.000,00		
Licenças de software	R\$ 140,00				
Salário dos funcionários dedicados ao projeto	R\$ 1.500,00				
Compra de Mais 'TAGS' e Cartões RFID	-				
Pasta de Solda	R\$ 33,94				
Equipamentos e ferramentas	R\$ 550,00				
Servidores	R\$ 150,00				

Fonte: Elaborados pelos autores.

Com base em uma pesquisa de mercado, identificamos uma empresa que oferece serviços similares aos nossos. Utilizando o orçamento que nos foi fornecido por essa empresa, obtivemos informações sobre os valores e os serviços, o que nos permitiu realizar uma comparação com os nossos próprios custos, a fim de determinar a viabilidade do nosso produto.

A proposta oferecida pela empresa Afixcode oferece os seguintes produtos e serviços:

- 2000 etiquetas para superfície não metálica - padrão - leitura em até 1,5 metros;

Quantidades e valores na parte de Hardware:

- Tags – Para quantidades acima de 10.000 itens;
- Tag RFID (Não metal – Impressão Preto e Branco) = R\$ 1,87 / unidade;
- Tag RFID (Metal - Impressão Preto e Branco) = R\$ 4,82 / unidade;
- Leitor portátil RFID Opticon (8EU6) = R\$ 9.200,00 / unidade;
- Kit portal, incluindo leitor e duas antenas paralelas, portal personalizado(adaptável ao ambiente)= R\$ 33.500,00 /unidade;

Quantidades e valores na parte de Software:

- Basic: 157 - Valor mensal (Até 500 itens e 1 usuário);

- Standard: 237 - Valor mensal (Até 1500 itens e 3 usuários);

Ao analisar os preços dos produtos em questão, torna-se evidente uma notável discrepância em relação ao nosso protótipo. O produto concorrente apresenta um custo aproximadamente 240% superior ao nosso, além de requerer uma mensalidade que varia entre R\$157 e R\$237, dependendo do plano escolhido. Além disso, eles oferecem apenas um banco de dados com 500 cadastros.

Por outro lado, nosso serviço estabelece uma mensalidade fixa de R\$150, proporcionando acesso a um banco de dados excepcional com mais de 200.000 registros, além de outras vantagens:

- **Escalabilidade:** Além de suportar um grande número de registros, também permitimos a implementação de mais de 100.000 usuários. Isso é uma grande vantagem para empresas que têm planos de expansão, pois podem adicionar mais usuários sem precisar mudar de plataforma.
- **Suporte Incluso:** Ao contrário de muitos concorrentes, que cobram caro pelo suporte e manutenção, nosso projeto inclui suporte mensal por cerca de R\$95, garantindo que nossos clientes tenham acesso a assistência técnica sempre que precisarem, sem custos adicionais elevados.
- **Abordagem Abrangente:** Nosso projeto RFID não se limita apenas a hardware. Também inclui um aplicativo e um site que facilitam o gerenciamento de dados e a interação com o sistema. Isso oferece uma solução completa e integrada para as necessidades de nossos clientes.

Em resumo, com base na análise, nosso projeto RFID oferece uma combinação única de custo acessível, obtendo capacidade de armazenamento de dados excepcional, escalabilidade de grande escala, suporte inclusivo satisfatório e abordagem abrangente com inúmeros benefícios. Essas vantagens tornam nossa solução a escolha ideal para empresas que buscam uma solução eficaz e econômica para suas necessidades de rastreamento e gerenciamento de dados.

3.4.3 PRECIFICAÇÃO

Podemos definir a precificação como o preço que a empresa cobra por seus produtos ou serviços. Na tabela abaixo, apresentamos algumas informações dos custos do nosso produto, como Matéria prima, Comissão, Impostos, Transação Financeira, Custo total, Margem de Contribuição, Margem, Mark Up e Índice de Markup.

Tabela 2 - Tabela de Precificação.

Custo Fixo	Custo Variável				Custo Total	Preço	Margem de Contribuição
	Matéria Prima	Comissão	Impostos	Transação Fin.			
R\$ 2.054,90	R\$ 2.675,84	R\$ 500,00	R\$ 300,00	R\$ 200,00	R\$ 5.730,74	R\$ 9.500,00	R\$3.769,26

Fonte: Elaborados pelos autores.

Todos esses termos são fundamentais no contexto financeiro e de precificação de produtos e serviços. Compreender e utilizar esses conceitos adequadamente é essencial para o sucesso financeiro de qualquer organização.

- **Matéria Prima:** Refere-se aos materiais básicos ou componentes necessários para a produção de um produto. No projeto podemos citar as placas arduino, sensores RFID, cartões e TAGS e os softwares. O gasto com a matéria prima totaliza em R\$2.675,84.
- **Comissão:** É uma recompensa oferecida ao funcionário de uma empresa, a fim de trazer incentivos quando o mesmo cumpre metas ou objetivos definidos previamente e tendo um ótimo desempenho com seu próprio trabalho. A comissão proposta pela nossa empresa, totaliza em R\$500,00.
- **Impostos:** Representam os valores que as empresas devem pagar ao governo sobre suas receitas e lucros. A taxa de imposto mensal totaliza em R\$300,00 (Valor sujeito a possíveis alterações).
- **Transação Financeira:** Refere-se a qualquer atividade que envolve a transferência de dinheiro ou ativos financeiros, como compra, venda, investimento, empréstimo ou pagamento. O custo total da transação financeira é de R\$200,00.
- **Custo total:** O custo total é a soma de todos os custos associados à produção ou oferta de um produto ou serviço. Ele inclui tanto os custos fixos quanto os custos variáveis, para calcular o custo total você precisa somar todos esses elementos. No projeto o custo total foi de R\$5.730,74, além dos custos fixos também podemos citar a comissão, impostos, matéria prima.
- **Margem de contribuição:** É a diferença entre o preço de venda de um produto ou serviço e os custos variáveis associados a ele. Em outras palavras, a margem de contribuição é a quantia disponível para cobrir os custos fixos e, eventualmente, gerar lucro e a fórmula usada para fazer o cálculo é: Margem de Contribuição = Receita Total - Custos Variáveis. Em relação ao nosso projeto, a margem de contribuição totalizou em R\$3.769,26, que representa 39,68% do valor final do protótipo.

A tabela abaixo apresenta as porcentagens de Margem, Mark-Up e Índice de Markup do nosso produto.

Tabela 3 - Tabela de Porcentagem e Índice de Markup.

Margem	Mark Up	Índice de Markup
39,68%	65,77%	1,66

Fonte: Elaborados pelos autores.

- **Margem:** A margem de lucro é a diferença entre a receita total e o lucro líquido de uma empresa. Ela indica quanto dinheiro a empresa mantém como lucro após a dedução de todos os custos, incluindo custos variáveis e custos fixos e a fórmula usada para fazer o cálculo é: $\text{Margem de Lucro (\%)} = (\text{Lucro Líquido} / \text{Receita Total}) \times 100$. Conforme demonstrado na tabela acima, com base nos cálculos, podemos constatar que a margem de lucro de nossa empresa é bastante positiva, atingindo um total de 39,68%.
- **Markup:** O Mark-up é uma porcentagem adicionada ao custo de um produto para determinar seu preço de venda. O Markup é usado para garantir que o preço de venda cubra não apenas os custos, mas também os lucros desejados da empresa. É uma estratégia comum de precificação. A porcentagem adicionada ao nosso produto totaliza 65,77%. Mesmo com esse aumento de quase 80%, nosso produto continua acessível e de baixo custo.
- **Índice de Markup:** O índice de Markup é uma porcentagem usada para determinar o preço de venda de um produto ou serviço. Para calcular o índice de Markup, você pode usar a seguinte fórmula: $\text{Índice de Mark-Up} = \text{preço de venda} \div \text{custo total} \times 100\%$. No projeto conseguimos alcançar um índice de 1,66%.

Com base nas informações acima, podemos concluir que nossa empresa busca excelência em seus resultados. Além de oferecer produtos de qualidade e suporte eficaz, todas as decisões foram minuciosamente discutidas e calculadas para fornecer um serviço acessível aos consumidores e, ao mesmo tempo, lucrativo para a empresa. Alcançar esses resultados positivos envolveu um estudo profundo dos conceitos de economia e finanças, compreensão das categorias de custos e cálculos essenciais, que foram fundamentais para alcançar um resultado surpreendente, até mesmo para nós mesmos. Conforme afirmou Ludwig von Mises

(1949), na economia de mercado não há outro meio de adquirir e preservar a riqueza, a não ser fornecer às massas o que elas querem, da maneira melhor e mais barata possível.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

Nos estágios mais primitivos da civilização, as pessoas não realizavam pagamentos da maneira como conhecemos hoje. Em vez disso, elas trocavam bens e serviços diretamente uns com os outros como uma forma de comércio, um sistema conhecido como "troca direta" ou "escambo". Por exemplo, alguém poderia trocar especiarias por um item artesanal. Os produtos trocados variavam de acordo com as necessidades das partes envolvidas. No entanto, à medida que as sociedades evoluíam, essa forma de pagamento por troca direta começou a causar problemas, principalmente devido à dificuldade de determinar um valor relativo entre diferentes bens.

Com o passar dos anos, a forma de avaliar o valor dos produtos e realizar pagamentos evoluiu consideravelmente. Essa evolução incluiu o uso de diferentes formas de moeda, como moedas metálicas e papel-moeda, bem como a introdução de instituições financeiras, como bancos e cheques. Além disso, surgiram inovações como cartões de crédito, pagamentos digitais e até mesmo criptomoedas. Essas mudanças continuam a acontecer à medida que as necessidades do século atual moldam os sistemas de pagamento e as tecnologias financeiras.

Com base nessas informações, fica evidente a importância de compreender o conceito de "finanças" e como ele está intrinsecamente ligado à experiência humana. Em suma, podemos afirmar que para um indivíduo, e até mesmo para um país, alcançar o sucesso financeiro, não basta apenas entender os conceitos básicos; é fundamental acompanhar o mercado financeiro e saber como gerenciar suas aplicações e economias de forma eficaz.

3.5.1 GERENCIANDO FINANÇAS

- **Tópico 1:** Introdução aos conceitos econômicos e financeiros básicos.

Seja para administrar uma empresa ou para a gestão financeira pessoal, a economia e a contabilidade precisam estar presentes para se alcançar o sucesso econômico. O conhecimento nessas áreas por parte do gestor financeiro pode ser altamente benéfico no dia a dia, uma vez que permite realizar análises que facilitam a tomada de decisões informadas.

Para compreendermos os fenômenos financeiros, é essencial entender a relação entre as entidades. Neste caso, estamos abordando a relação entre pessoas e empresas, em que ambas as partes precisam chegar a um acordo de valor para estabelecer uma relação de troca de bens, serviços ou consumo.

Na visão contábil, podemos classificar os gastos em algumas terminologias, sendo elas gasto, desembolso, despesa, custo, investimento, entre outros. Podemos dizer que os gastos são um sacrifício financeiro que resulta na busca da realização de um objetivo, seja ele de curto, médio ou longo prazo. O papel da contabilidade é nos auxiliar no controle dessas transações financeiras.

São classificados como fenômenos financeiros "Investimento, Custo e Despesas". Compreender suas vantagens contribui para equilibrar uma gestão financeira saudável, levando tanto o indivíduo quanto uma empresa ao sucesso. Em suma, 'Investimento' tem como objetivo alocar recursos financeiros em ativos com a expectativa de obter retornos financeiros positivos ao longo do tempo, visando ao crescimento do patrimônio. Por outro lado, 'Custos' são despesas financeiras associadas à produção de bens ou serviços em uma empresa e desempenham um papel fundamental na determinação do lucro. Por fim, 'Despesas' são gastos financeiros que não estão diretamente relacionados à produção de bens e serviços, mas são necessários para manter as operações em transições, tanto em contexto empresarial quanto pessoal.

Muitas teorias estudadas em administração financeira podem ser facilmente aplicadas na gestão pessoal. Ter conhecimento de quanto se ganha e o quanto se gasta é fundamental para determinar a situação financeira do próximo mês. Portanto, é de extrema importância não gastar mais do que se ganha. Ter controle da própria renda, a curto ou longo prazo, traz grandes benefícios, como a capacidade de planejar uma viagem, comprar algo desejado, trocar um veículo e até mesmo investir no mercado financeiro. Além disso, em caso de imprevistos, estar preparado proporciona paz de espírito, e ao final do dia, ao dormir, você estará mais tranquilo.

- **Tópico 2:** Entendendo o ambiente: independência financeira, o valor da minha riqueza e o registro do dia a dia.

No estudo de administração financeira, um dos princípios abordados é que um gestor financeiro competente pode aplicar conceitos importantes em sua vida pessoal. Se soubermos administrar com sabedoria nossas finanças pessoais, também teremos a capacidade de administrar uma empresa com sucesso.

O primeiro passo para gerar um resultado financeiro positivo é identificar uma fonte de renda que, após deduzir os custos associados, deixará um excedente. Para as empresas, essa fonte de renda provém do faturamento gerado pela venda de produtos ou serviços. Para os indivíduos, a renda geralmente deriva do emprego ou do empreendedorismo.

Outra estratégia é investir o que resta após cobrir as despesas mensais para fazer o dinheiro crescer ao longo do tempo, por exemplo, investindo em aplicações financeiras, obtendo renda com aluguéis ou se tornando acionista de uma empresa, entre outras possibilidades. Em resumo, a disciplina no gerenciamento financeiro é fundamental para o sucesso.

É essencial manter controle sobre os gastos, garantindo que as despesas não excedam as receitas. Alguns compromissos são mensais e fixos, difíceis de evitar, enquanto outros podem ser reduzidos, adiados ou eliminados. Em caso de imprevistos e complicações, é importante tomar medidas para minimizar os danos e, se possível, resolver completamente o problema financeiro.

- **Tópico 3:** Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado.

A Matemática Financeira estuda o valor do dinheiro ao longo do tempo, ou seja, ela realiza uma análise quantitativa das transações financeiras e das decisões relacionadas ao dinheiro ao longo do tempo. Ela é composta por diversos conceitos e fórmulas que auxiliam na avaliação e tomada de decisões financeiras em diversas situações, tanto pessoais quanto empresariais.

Existem dois sistemas para determinação dos juros em uma operação financeira: juros simples e juros compostos. Em matemática financeira, ambos passam por análises de como os juros são calculados ao longo do tempo. Os juros simples incidem apenas sobre o capital inicial, enquanto os juros compostos incidem sobre o capital inicial e os juros acumulados.

Ao contratar créditos, precisamos nos atentar ao montante que desejamos solicitar e avaliar se é viável diante das taxas de juros aplicadas. Em matemática financeira, no conceito de crédito, ela faz uma análise de como funcionam financiamentos e empréstimos, incluindo o cálculo de prestações, amortização, saldo devedor e custo total do financiamento.

A organização financeira é significativa não apenas no meio empresarial, mas também tem sua relevância no campo pessoal. A desorganização financeira não afeta apenas o desempenho operacional, mas também impacta o bem-estar emocional das pessoas, trazendo inúmeras consequências negativas. Portanto, é de extrema importância estabelecer prioridades

e preparar um orçamento financeiro como medidas para manter suas finanças mais organizadas.

- **Tópico 4:** Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos.

Para garantir sucesso no futuro, é preciso ter sabedoria no presente. Construir hábitos saudáveis para manter as finanças organizadas desde cedo, seja no âmbito empresarial ou pessoal, é a primeira linha de raciocínio que devemos colocar em prática, seguida do planejamento. É primordial agir com cautela na tomada de decisões, sempre levando em consideração as consequências a curto, médio e longo prazo. Portanto, quando realizamos planejamentos, podemos colher informações essenciais ao longo da execução que possibilita análises para determinar se estamos ou não no caminho do sucesso.

O sonho e a busca por realizar projetos estão enraizados em nossa mente desde cedo, abrangendo desde planos simples até os mais grandiosos. No entanto, o que eles têm em comum é a maneira como agiremos para concretizar esses desejos e aspirações. Quando aplicamos a educação financeira, os benefícios podem ser enormes. Portanto, saber equilibrar as duas áreas de nossa vida, pessoal e empresarial, é indispensável. Dessa forma, não devemos deixar de questionar como desejamos nos encontrar no futuro e se, no presente atual, estamos conseguindo equilibrar a gestão financeira e como podemos melhorar nesse aspecto.

3.5.2 ESTUDANTES NA PRÁTICA

A partir das informações da apostila 'Gerenciando Finanças', foram produzidos uma série de stories para o Instagram, contendo conteúdo informativo sobre a importância de compreender a necessidade de um bom gerenciamento financeiro desde cedo, tanto na vida pessoal quanto na empresarial. Esses stories incluem dicas baseadas em conceitos econômicos e financeiros básicos, princípios fundamentais que ajudam a compreender a economia e a gestão financeira de forma simples, além de ressaltar a importância de se ter foco no planejamento.

A sequência será postada na conta oficial da empresa no Instagram, e fixada na forma de Destaque, funcionalidade da plataforma, estando sempre disponível de forma prática a quem interessar ou for necessário.

Conteúdo para redes sociais.

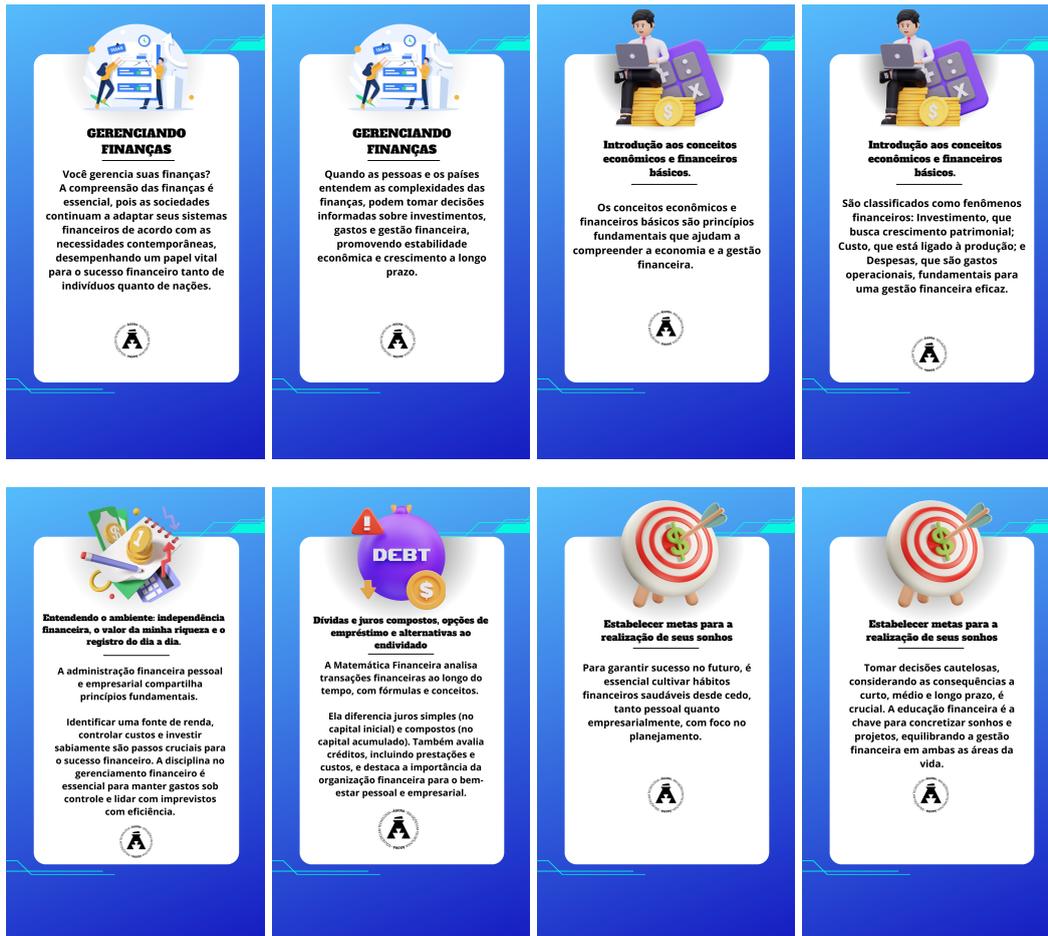


Figura 0 - Sequência de stories com dicas sobre planejamento financeiro e seus conceitos básicos.

Fonte: Elaborado pelos autores.

4. CONCLUSÃO

A segurança verdadeira só se concretiza quando a sensação de proteção está presente. Embora isso pareça óbvio, é a atmosfera de segurança proporcionada pelo ambiente que efetivamente a estabelece. Num mundo onde a garantia de segurança não é mais uma certeza, torna-se essencial que as instituições de ensino reavaliem seus sistemas. Nesse sentido, desenvolvemos um sistema exclusivo para a instituição Unifeob, projetado para assegurar essa sensação a todos que transitam por seus espaços.

Além disso, a satisfação do cliente é um elemento fundamental para qualquer empresa. A segurança, sendo uma característica básica e crucial, é não apenas esperada, mas essencial em qualquer serviço.

Dessa forma, a empresa Ágora propõe um produto baseado na tecnologia RFID para aprimorar a segurança desses locais, proporcionando aos estudantes e funcionários das instituições uma sensação de segurança ampliada. O desenvolvimento consiste em uma interface que possibilita a coleta, gerenciamento e controle dos dados armazenados no banco de dados. Essas informações são geradas pelos Arduinos distribuídos pela área. Todo o mecanismo de filtragem está sincronizado nas páginas web, assegurando uma experiência de usuário agradável.

Com certeza, essa solução versátil pode ser expandida para gerenciar, no futuro, possíveis armários que venham a ser instalados nas dependências da instituição. Além disso, ela simplificará o processo de registro de frequência dos alunos e facilitará a gestão da distribuição de Chromebooks aos estudantes. Essa flexibilidade evidencia a adaptabilidade do sistema, proporcionando uma extensa variedade de funcionalidades que podem ser exploradas para atender às necessidades em constante evolução da instituição de ensino.

Conforme Moonegan (2016), "A Internet das Coisas (IoT - Internet of Things) pode ser utilizada em diversos segmentos da vida humana, seja em âmbitos particulares como a casa, o carro e o entretenimento individual, ou em âmbitos públicos como iluminação das cidades, transporte público, programas de saúde, educação e outros". Com base nisso, o projeto apresentado neste trabalho, por meio da IoT, busca atender às necessidades de todos que frequentam os espaços da Unifeob. Isso proporcionará maior segurança e conforto, além de contribuir para a melhoria da imagem da instituição perante seus stakeholders e a comunidade em geral.

REFERÊNCIAS

Caldeira, R. P. (n.d.). **Banco de dados, database, SGBD: você sabe o que é isso?** Raphael P. Caldeira. Retirado em 14/11/2023 de <https://academy.indicium.tech/blog/banco-de-dados-database-sgbd-voce-sabe-o-que-e-isso>

Coutinho, T. (2023). **Modelagem de Dados: o que é, importância, tipos e exemplos.** Blog Voitto; Voitto. Retirado em 10/11/2023 de <https://www.voitto.com.br/blog/artigo/modelagem-de-dados>

Curso em vídeo- Java POO [40 Horas]. Retirado em 12/9/2023 de <https://www.cursoemvideo.com/curso/java-poo/>

DevMedia-Principais Conceitos da Programação Orientada a Objetos. Retirado em 24/10/2023 de devmedia.com.br

Estrutura de dados do Javascript. Retirado em 5 de Outubro de 2023, de https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Data_structures

Estruturas condicionais e estruturas de repetição em JavaScript. Retirado em 5 de Outubro de 2023, de <https://www.treinaweb.com.br/blog/estruturas-condicionais-e-estruturas-de-repeticao-em-javascript>

KOTLER, P.; KELLER, K. L. (2012). **Administração de marketing.** 14. ed. São Paulo: Pearson Prentice Hall.

Livro "**Aprendendo Python: Programação Orientada a Objetos**". Autor: Mark Lutz.

Lógica de programação: o que é e como aprender? Retirado em 5 de outubro de 2023, de <https://www.hostgator.com.br/blog/logica-de-programacao/>

MySQL: confira o que é e suas vantagens. (n.d.). HostGator. Retirado em 15/11/2023 de <https://www.hostgator.com.br/blog/mysql-e-suas-vantagens/>

O que é e como funciona algoritmo e lógica de programação. Blog da Casa do Desenvolvedor; Casa do Desenvolvedor. Retirado em 19 de setembro de 2023, de <https://blog.casadodesenvolvedor.com.br/logica-de-programacao/>

O que é modelagem de dados? (n.d.). Microsoft.com; Microsoft. Retirado em 23/11/2023 de <https://powerbi.microsoft.com/pt-br/what-is-data-modeling/>

O que é modelagem de dados? (n.d.-b). SAP. Retirado em 09/11/2023 de <https://www.sap.com/brazil/products/technology-platform/datasphere/what-is-data-modeling.html>

Quanto custa um sistema RFID – SMARTX HUB BRASIL – Rastreamento e Controle de Ativos e Pessoas em Tempo Real. (n.d.). Retirado em 15/11/2023 de <https://smartxhub.com.br/2023/01/quanto-custa-um-sistema-rfid/>

Rezende, R. (2006, April 16). **Banco de dados: Conceitos Fundamentais.** DevMedia. Retirado em 13/10/2023 de <https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>

Soares, I. (2022, December 22). **Custos diretos, indiretos, fixos e variáveis: guia completo.** Cobli. Retirado em 15/11/2023 de <https://www.cobli.co/blog/custos-diretos-e-indiretos/>

Soares, I. (2022, December 22). **Custos diretos, indiretos, fixos e variáveis: guia completo.** Cobli. Retirado em 12/11/2023 de <https://www.cobli.co/blog/custos-diretos-e-indiretos/>

Tutorialspoint - **POO em Python.**

Youtube - **O que é Orientação a Objeto em Python?** Retirado em 26/08/2023, Vídeo: <https://www.youtube.com/watch?v=dG7LIYne2VA>

Youtube - **Professor Otávio Miranda - O que é Orientação a Objeto?** Retirado em 22/08/2023, Vídeo: https://www.youtube.com/watch?v=RLVbB91A5-8&list=PLbIBj8vQhvm34qAAEEH_PdL2tMG9rz-P7

ANEXOS

Anexo A - Link para a imagem 1 [link](#)

https://drive.google.com/file/d/12l_6UquBWec2gHiu3voBCXyqZHoprDHC/view

Anexo B - Link para a imagem 2 [link](#)

https://drive.google.com/file/d/1qZS-3yD2qZZK5ciHgiGbsI2-cOkzxDyz/view?usp=share_link



RELATÓRIO FINAL DAS ATIVIDADES DE EXTENSÃO

1. IDENTIDADE DA ATIVIDADE

RELATÓRIO: IOT DATA STREAMER

CURSO: Análise e Desenvolvimento de Sistemas e Ciência da Computação

MÓDULO: MODELAGEM E DESENVOLVIMENTO DE SISTEMAS

PROFESSOR RESPONSÁVEL: Mariangela Martimbianco Santos

ESTUDANTE: Matheus Farnetani Friedrich

PERÍODO DE REALIZAÇÃO: 08/2023 a 11/2023

2. DESENVOLVIMENTO

Contextualização: Para realizar esta atividade de extensão, continuamos o projeto iniciado no primeiro semestre. Naquela fase, nosso objetivo era identificar um problema existente na instituição de ensino Unifeob e desenvolver uma solução tecnológica para o mesmo. O problema identificado dizia respeito à segurança nos espaços da instituição, especificamente à identificação das pessoas que circulam por eles, uma vez que, atualmente, qualquer pessoa pode ter acesso sem a necessidade de se identificar. Para abordar essa questão, desenvolvemos um sistema de identificação que utiliza cartões RFID para controlar o acesso. Neste semestre, implementamos nossa tecnologia, incorporando o "IoT Data Streamer". Em outras palavras, o objetivo do IoT Data Streamer é implementar uma aplicação desktop responsável por gerenciar os dados coletados e processados no "App" mobile, provenientes de dispositivos de Internet das Coisas (IoT). Essa aplicação simula a importação de dados gerados, tanto em formato TXT como CSV, e popula o banco de dados.

Desafio: Os desafios que deparamos no meio do processo de desenvolvimento do sistema, consistiram de estudos para criar uma interface que permitisse o gerenciamento e controle dos dados gerados pela ideia do sistema completo de cartões RFID. Nesta interface, tivemos que aprender a criar sistemas de cadastros e sistemas de coleta de informações no banco de dados, que permitisse o acesso às informações geradas pelos arduinos espalhados pela área. Outro desafio importante a ser ressaltado, foi a questão de criar mecanismos de filtragem assíncronas nas páginas web de apresentação das informações contidas no banco de dados para alimentar o conteúdo presente nas páginas para garantir uma experiência de usuário agradável. A estrutura do banco de dados foi escrita utilizando a biblioteca SQLAlchemy que através de declaração de objetos cria automaticamente a conexão com o banco de dados e suas tabelas, no caso de ser um novo banco de dados.

Cronograma das Ações:

Semana 01: sexta-feira, 11 de agosto de 2023

Semana 02: sexta-feira, 18 de agosto de 2023

Semana 03: sexta-feira, 25 de agosto de 2023

Semana 04: sexta-feira, 01 de setembro de 2023

Semana 05: sexta-feira, 08 de setembro de 2023

Semana 06: sexta-feira, 15 de setembro de 2023
Semana 07: sexta-feira, 22 de setembro de 2023
Semana 08: sexta-feira, 29 de setembro de 2023
Semana 09: sexta-feira, 06 de outubro de 2023
Semana 10: sexta-feira, 13 de outubro de 2023
Semana 11: sexta-feira, 20 de outubro de 2023
Semana 12: sexta-feira, 27 de outubro de 2023
Semana 13: sexta-feira, 03 de novembro de 2023
Semana 14: sexta-feira, 10 de novembro de 2023
Semana 15: sexta-feira, 17 de novembro de 2023
Semana 16: sexta-feira, 24 de novembro de 2023
Semana 17: 27 a 29 de novembro de 2023

Síntese das Ações:

Semana 01: Live de apresentação do cronograma dos Projetos Integrados.
Semana 02: Organização das equipes e definição dos estudantes monitores.
Semana 03: Definição e apresentação das informações sobre a empresa a ser utilizada no projeto.
Semana 04: Descrição do objetivo do projeto (parte provisória da Introdução) e definição do cronograma de trabalho da equipe.
Semana 05: Suspensão de Atividades.
Semana 06: Live sobre a atividade relacionada ao eixo de Formação para a Vida.
Semana 07: Elaboração de pesquisa para o desenvolvimento do tema.
Semana 08: Apresentação da síntese sobre o andamento do projeto em relação ao cronograma de trabalho definido anteriormente, relatando possíveis problemas que podem impactar na realização e respectiva entrega final do projeto.
Semana 09: Elaboração de pesquisa para o desenvolvimento do tema.
Semana 10: Suspensão de Atividades.
Semana 11: Live sobre as questões metodológicas de referências e citações bibliográficas e conceituais utilizadas no projeto (REFERÊNCIAS).
Semana 12: Live sobre as orientações de apresentação dos projetos.
Semana 13: Suspensão de Atividades.
Semana 14: Live sobre o preenchimento do relatório de extensão para entrega via sistema.
Semana 15: Revisão final dos Projetos Integrados.
Semana 16: Entrega final dos Projetos em PDF.
Semana 17: Apresentações dos Projetos Integrados.

a. Aspectos positivos: Com o desenvolvimento desta atividade de extensão, foi possível adquirir diversos novos conhecimentos teóricos e técnicos nos campos de estudo de Programação Orientada a Objeto, Lógica de Programação, Modelagem de Dados e Gestão Financeira. Isso ocorreu através da aplicação prática desses conhecimentos no desenvolvimento da empresa e na implementação da solução escolhida pela equipe. Outro ponto positivo a ser destacado é o aprimoramento das habilidades em linguagens como Python, JavaScript, Programação Orientada a Objeto e SQL. Essas linguagens foram essenciais para o funcionamento da aplicação, permitindo a integração eficaz e a manipulação precisa das informações em tempo real. A capacidade de misturar essas linguagens foi crucial para garantir que o sistema mapeasse as informações de maneira precisa, sendo possível acessá-las por meio da aplicação Desktop.

b. Dificuldades encontradas: A principal dificuldade enfrentada durante todo o processo foi compreender claramente qual era o objetivo do projeto e desenvolver o conteúdo de forma eficaz. A atividade proposta demandava um certo nível de entendimento e conhecimento, o que tornou desafiador para alguns membros do grupo. Foi necessário dividir as tarefas com base no conhecimento de cada um,

mas mesmo assim houve certa dificuldade na organização, além de estabelecer datas para a entrega, demandando um nível de cobrança um pouco mais insistente. Conciliar o tempo para a realização das atividades relacionadas ao projeto também representou um desafio, visto que nem todos os integrantes do grupo tinham a mesma disponibilidade. Essas dificuldades foram superadas com esforço conjunto e uma abordagem flexível para garantir o progresso do projeto.

c. Resultados atingidos: O projeto foi concluído de maneira satisfatória e positiva, com o desenvolvimento bem-sucedido das principais melhorias que puderam ser implementadas, levando em consideração a adaptação de todos os alunos e colaboradores da Unifeob. Foi possível criar conteúdos informativos sobre 'Formação para a Vida' nas redes sociais, além de desenvolver o aplicativo com uma interface responsiva, de fácil acesso e visualização ampla das informações de mapeamento. O aplicativo exibe detalhes como data, hora, tipo de cartão, entre outras informações relevantes. Além disso, apresenta um gráfico que ilustra o número de pessoas por tipo de cartão registrado no mês. As expectativas e objetivos foram alcançados dentro do prazo de desenvolvimento dedicado, resultando em um sistema complexo de identificação com múltiplas funcionalidades e etapas de implementação, apoiado por uma base sólida e bem desenvolvida.

d. Sugestões / Outras observações: Incluir sugestões e/ou observações sobre o projeto (opcional).

3. EQUIPE DOS ESTUDANTES NO PROJETO

RA 23000780	NOME Matheus Farnetani Friedrich
RA 23000143	NOME Lucas Guimarães Castro Nunes
RA 23000786	NOME Leticia Ramiro Abreu
RA 23000710	NOME Fellipe Kozerski Silva
RA 23000679	NOME Carlos Gabriel dos Santos

Curso de Análise e Desenvolvimento de Sistemas e Ciência da Computação

Módulo de Modelagem e Desenvolvimento de Sistemas

Cronograma de Validação - Projeto Integrado

Unidade Estudo	Participação no Projeto	Validação
Gestão Financeira	Custo do Produto/Serviço	25/10/2023
Programação Orientada a Objeto	Codificação Métodos Privados e Públicos	03/10/2023
	Codificação Estruturada em POO	14/11/2023
Lógica de Programação	Prototipação da Interface do Projeto no Figma	25/08/2023
	Desenvolvimento do Front-end do Projeto	27/10/2023
	Desenvolvimento do Back-end do Projeto	17/11/2023
Modelagem de Dados	Apresentação do Modelo Conceitual do Banco de Dados	28/08/2023
	Apresentação do Modelo Lógico e Físico do Banco de Dados	25/09/2023
	Demonstrar a manipulação dos dados no banco, por meio de scripts.	16/10/2023
<u>Descrição do Projeto:</u> Desenvolver um sistema para consumir e processar dados em tempo real a partir de um ou vários dispositivos de Internet das Coisas (IoT).		