



**UNifeob**  
| ESCOLA DE NEGÓCIOS

2024

# PROJETO INTEGRADO



UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**CIÊNCIA DA COMPUTAÇÃO**

**PROJETO INTEGRADO**

SISTEMA DE GESTÃO E INTELIGÊNCIA DE NEGÓCIOS  
PARA ORGANIZAÇÕES SOCIAIS

**<Projeto Beija Flor>**

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2024

UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
CIÊNCIA DA COMPUTAÇÃO

**PROJETO INTEGRADO**  
SISTEMA DE GESTÃO E INTELIGÊNCIA DE NEGÓCIOS  
PARA ORGANIZAÇÕES SOCIAIS

**<Projeto Beija Flor>**

MÓDULO COMPUTAÇÃO EM NUVEM

Estrutura de Dados – Prof. Marcelo Ciacco Almeida

Linguagem e Técnicas de Programação – Prof. Nivaldo de Andrade

Tópicos Avançados de Banco de Dados – Prof. Max Streicher Vallim

Computação em Nuvem – Prof. Rodrigo Marudi de Oliveira

Projeto de Computação em Nuvem – Profª. Mariângela Martimbianco Santos

Estudantes:

Igor Garcia Nunes de Araujo RA : 23000093

Igor Jose Ananias Campos RA : 23000131

Joao Pedro Pizoli Carvalho RA : 23001145

Vinicius Felipe Tesser RA : 23000778

SÃO JOÃO DA BOA VISTA, SP  
NOVEMBRO 2024

# SUMÁRIO

1. INTRODUÇÃO	5
2. DESCRIÇÃO DA EMPRESA	6
3. PROJETO INTEGRADO	7
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS	7
3.1.1 MODELO LÓGICO	8
3.1.2 MODELO FÍSICO	12
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	15
3.2.1 APPLICATION PROGRAMMING INTERFACE ( API ) - BACK-END.	15
3.2.2 FRONT-END	16
3.3 COMPUTAÇÃO EM NUVEM	16
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	16
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	17
3.3.3 VANTAGENS DA CLOUD COMPUTING	18
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	20
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	22
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	23
3.3.7 GOOGLE CLOUD ou AWS	25
3.4 ESTRUTURA DE DADOS	26
3.4.1 LEVANTAMENTO DE REQUISITOS	27
3.4.2 VALIDAÇÃO DOS REQUISITOS	29
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	30
3.5.1 ENFRENTANDO ESTEREÓTIPOS	30
3.5.2 ESTUDANTES NA PRÁTICA	31
4. CONCLUSÃO	31
REFERÊNCIAS	33
ANEXOS	33

# 1. INTRODUÇÃO

Nesta seção do Projeto Integrado (PI), apresentamos os objetivos do projeto e o que almejamos alcançar com sua implementação. O módulo de "Computação em Nuvem" dá continuidade ao trabalho realizado no módulo anterior, que se concentrou na criação de um "Sistema de Gestão e Inteligência de Negócios para Organizações Sociais". O objetivo principal deste módulo é aprimorar esse sistema de gestão para atender de forma mais eficaz organizações sociais como ONGs, instituições de caridade e cooperativas, com ênfase na eficiência, escalabilidade e segurança dos dados.

Para atingir esses objetivos, o sistema foi projetado utilizando a infraestrutura de computação em nuvem da AWS, empregando o SQL Server como banco de dados e uma API desenvolvida com Node.js, Express e Knex. Essa arquitetura garante a segurança dos dados e alta disponibilidade da aplicação, além de oferecer um sistema escalável e acessível para a gestão das operações diárias dessas organizações.

O desenvolvimento deste projeto aplica os conhecimentos adquiridos nas disciplinas de Estrutura de Dados, Linguagens e Técnicas de Programação, Tópicos Avançados de Banco de Dados, Computação em Nuvem e Projeto de Computação em Nuvem. Ao final, este PI consolida esses conceitos de forma prática, oferecendo uma solução robusta e eficiente para as organizações sociais, contribuindo para uma gestão mais ágil e organizada e promovendo um impacto positivo na sociedade.

## **2. DESCRIÇÃO DA EMPRESA**

A Associação Educacional e Cultural Beija-Flor, conhecida como Projeto Beija-Flor, é uma Organização Não Governamental (ONG) registrada sob o CNPJ 12.345.678/0001-99. Está localizada na Rua Barão de Monte Santo, 1910, Jardim Rigobelo, Mococa, SP, e atua na área educacional e cultural, oferecendo educação musical gratuita para crianças e adolescentes de comunidades carentes. A principal missão da ONG é promover o desenvolvimento social e pessoal dos jovens atendidos, utilizando a música como ferramenta de inclusão e transformação social.

O Projeto Beija-Flor opera no terceiro setor, preenchendo uma lacuna na oferta de educação artística, que muitas vezes não é contemplada no sistema público de ensino. Entre os serviços oferecidos, destacam-se aulas de instrumentos como violão, piano, flauta e bateria. A ONG também organiza eventos culturais e apresentações públicas, permitindo que as crianças demonstrem seu aprendizado e se integrem à comunidade.

As informações para a realização deste PI foram obtidas por meio de reuniões com os gestores da ONG ao longo do semestre e complementadas por fontes bibliográficas, como artigos e publicações sobre o impacto social da educação musical. Todas as referências foram devidamente citadas de acordo com a Metodologia Científica, reforçando a importância e o embasamento do projeto.

### **3. PROJETO INTEGRADO**

Nesta etapa do PI serão apresentados os conteúdos que cada unidade de estudo utilizará para realizar o projeto, assim como a forma que serão aplicados na empresa escolhida para a realização do projeto.

#### **3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS**

Este tópico consiste no desenvolvimento de um sistema de banco de dados para uma Organização Não Governamental (ONG). Nesta etapa do projeto, foram desenvolvidos modelos de dados para estruturar o banco de dados que armazenará informações relacionadas ao sistema, para que possa gerenciar e armazenar a presença de alunos e professores nas aulas, verificar as futuras aulas, seus respectivos instrumentos e a quantidade de alunos cadastrados nas mesmas.

Começando com o modelo lógico onde podemos ter uma visão detalhada e formalizada das estruturas de dados, organizada para evidenciar as necessidades funcionais do sistema. E também o modelo físico que é a representação detalhada das estruturas de dados definidas para armazenamento em um sistema de gerenciamento de banco de dados (SGBD).

Para o desenvolvimento do projeto foi utilizado o HeidiSQL para a construção da aplicação e integrada com Amazon RDS (Relational Database Service), para que pudesse ser operado, escalado e configurado, com uso de stored procedures, que são um conjunto de instruções SQL predefinidas que são armazenadas e executadas diretamente no servidor de banco de dados. Triggers, que são um conjunto de instruções que são automaticamente executados em resposta a certos eventos em uma tabela ou em uma view, como inserções, atualizações ou exclusões de dados. Views, que é uma consulta armazenada que cria uma "tabela virtual" com dados derivados de uma ou mais tabelas reais, também utilizamos colunas personalizadas que são campos adicionais criados em uma tabela para armazenar informações específicas que não foram previstas inicialmente, e por último as funções, que são conjuntos de instruções que realizam tarefas específicas, como calcular um valor ou transformar dados de uma tabela.

Esses modelos foram utilizados visando automatizar o registro e validação dos alunos e professores, além de permitir relatórios detalhados sobre a assiduidade dos participantes.

### 3.1.1 MODELO LÓGICO

Como afirma Eliezio (2024), “Um modelo lógico deve definir quais tabelas o banco contém e, quais os nomes de cada uma das colunas das tabelas”. Com base nisso, no desenvolvimento do projeto desenvolvemos uma representação gráfica dos requisitos de informação que serão coletados por meio do aplicativo, utilizando o modelo lógico para estabelecer as bases do modelo físico do banco de dados.

Para exemplificar, imagine que você está inserindo suas informações pessoais em um site ou aplicativo, essas informações ficam armazenadas em algum banco de dados e serão utilizadas posteriormente, as informações que você cadastrar em nosso aplicativo, serão armazenadas em nosso banco de dados e corretamente divididas, como dito anteriormente, utilizamos o modelo lógico apenas para ter uma ideia.

Neste modelo lógico, utilizamos três tabelas principais:

1. **Usuários:** destinada ao cadastro de dados pessoais, como senha, nome de usuário e e-mail.
2. **Professores:** onde são armazenados o nome dos professores, informações sobre as próximas aulas, códigos únicos de identificação e o compartimento correspondente a cada professor.
3. **Alunos:** responsável por armazenar o nome dos alunos, as aulas em que estão matriculados e os instrumentos que utilizam.

O modelo lógico foi desenvolvido para dar uma visão clara e organizada da estrutura de dados necessária, servindo como base para a criação do modelo físico do banco de dados que será implementado.

Aquí está a tabela usuários:





Aqui está a tabela professores:

## Professores:

```
-- MySQL dump 10.13 Distrib 8.0.39, for Win64 (x86_64)
--
-- Host: localhost    Database: projetopi
-- -----
-- Server version    8.0.39

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `professores`
--

DROP TABLE IF EXISTS `professores`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `professores` (
  `id` int NOT NULL AUTO_INCREMENT,
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `areaAtuacao` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_professores_usuarios_idx` (`email`),
  CONSTRAINT `fk_professores_usuarios` FOREIGN KEY (`email`) REFERENCES `usuarios` (`email`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `professores`
--

LOCK TABLES `professores` WRITE;
/*!40000 ALTER TABLE `professores` DISABLE KEYS */;
INSERT INTO `professores` VALUES (1,'professorx@escola.com',NULL),(2,'professorxx@escola.com','Vidão');
/*!40000 ALTER TABLE `professores` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2024-09-19 22:37:25
```

Aqui está a tabela alunos:

## Alunos:

```
-- MySQL dump 10.13  Distrib 8.0.38, for Win64 (x86_64)
--
-- Host: localhost    Database: projetopi
-----
-- Server version    8.0.39

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `alunos`
--

DROP TABLE IF EXISTS `alunos`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `alunos` (
  `id` int NOT NULL AUTO_INCREMENT,
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `instrumento` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `matricula` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `dataNascimento` date DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_alunos_usuarios_idx` (`email`),
  CONSTRAINT `fk_alunos_usuarios` FOREIGN KEY (`email`) REFERENCES `usuarios` (`email`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `alunos`
--

LOCK TABLES `alunos` WRITE;
/*!40000 ALTER TABLE `alunos` DISABLE KEYS */;
/*!40000 ALTER TABLE `alunos` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

### 3.1.2 MODELO FÍSICO

De acordo com Erwin (2024), “Um modelo físico de dados introduz o contexto específico do banco de dados ausente em modelos conceituais e lógicos de dados. Ele representa as tabelas, colunas, tipos de dados, visualizações, restrições, índices e procedimentos dentro do banco de dados e/ou as informações comunicadas durante os processos de computador.” Com base nessa perspectiva, desenvolvemos o modelo físico para traduzir o modelo lógico em uma estrutura detalhada e implementável, que considera os requisitos técnicos e a otimização do desempenho no ambiente escolhido.

O modelo físico detalha com precisão a estrutura de armazenamento dos dados, definindo tabelas e colunas, selecionando tipos de dados adequados para cada campo e implementando restrições e índices. Esse nível de especificação possibilita manter a integridade dos dados, otimizar consultas para maior eficiência e garantir que as regras de negócio sejam aplicadas corretamente no sistema.

Com a estruturação adequada do modelo físico, foi possível "popular" o banco de dados com as informações essenciais para o funcionamento do sistema, garantindo que cada dado armazenado esteja devidamente categorizado e otimizado para acessos e modificações futuras, de maneira eficiente e segura.

Aqui estão alguns exemplos de objetos de banco de dados que foram utilizados para a realização do projeto:

Views que foram utilizadas:

**Views:**

```

-- View para listar todos os professores com dados de usuários
CREATE VIEW view_professores_usuarios AS
SELECT u.id AS usuario_id, u.nome AS usuario_nome, u.email, u.role, p.areaAtuacao
FROM usuarios u
JOIN professores p ON u.email = p.email;

-- View para listar todos os alunos com dados de usuários
CREATE VIEW view_alunos_usuarios AS
SELECT u.id AS usuario_id, u.nome AS usuario_nome, u.email, u.role, a.instrumento, a.matricula, a.dataNascimento
FROM usuarios u
JOIN alunos a ON u.email = a.email;

```

Triggers utilizados:

### Triggers:

```

-- Trigger para log de inserção em alunos
DELIMITER //
CREATE TRIGGER after_insert_alunos
AFTER INSERT ON alunos
FOR EACH ROW
BEGIN
    INSERT INTO logs (action, description, timestamp)
    VALUES ('INSERT', CONCAT('Novo aluno adicionado: ', NEW.email), NOW());
END //
DELIMITER ;

-- Trigger para auditoria de atualização de usuários
DELIMITER //
CREATE TRIGGER after_update_usuarios
AFTER UPDATE ON usuarios
FOR EACH ROW
BEGIN
    INSERT INTO logs (action, description, timestamp)
    VALUES ('UPDATE', CONCAT('Usuário atualizado: ', NEW.email), NOW());
END //
DELIMITER ;

```

## Stored Procedures:

```
-- Stored Procedure para adicionar um novo aluno
DELIMITER //
CREATE PROCEDURE AddAluno(IN alunoEmail VARCHAR(255), IN alunoInstrumento VARCHAR(255), IN alunoMatricula VARCHAR(50), IN alunoDataNascimento DATE)
BEGIN
    INSERT INTO alunos (email, instrumento, matricula, dataNascimento)
    VALUES (alunoEmail, alunoInstrumento, alunoMatricula, alunoDataNascimento);
END //
DELIMITER ;

-- Stored Procedure para adicionar um novo professor
DELIMITER //
CREATE PROCEDURE AddProfessor(IN professorEmail VARCHAR(255), IN professorAreaAtuacao VARCHAR(255))
BEGIN
    INSERT INTO professores (email, areaAtuacao)
    VALUES (professorEmail, professorAreaAtuacao);
END //
DELIMITER ;

-- Stored Procedure para atualizar dados do usuário
DELIMITER //
CREATE PROCEDURE UpdateUsuario(IN userId INT, IN novoNome VARCHAR(255), IN novoEmail VARCHAR(255), IN novaSenha VARCHAR(255), IN novaRole ENUM('admin', 'aluno', 'professor'))
BEGIN
    UPDATE usuarios
    SET nome = novoNome, email = novoEmail, password = novaSenha, role = novaRole
    WHERE id = userId;
END //
DELIMITER ;
```

## Indexes que foram utilizados:

### Indexes:

```
-- Índices para melhorar a busca em colunas de e-mail nas tabelas alunos e professores
CREATE INDEX idx_alunos_email ON alunos(email);
CREATE INDEX idx_professores_email ON professores(email);
```

## Constraints utilizadas:

### Constraints:

```
-- Restrições de unicidade para garantir que e-mails de alunos e professores sejam únicos
ALTER TABLE alunos ADD CONSTRAINT unique_aluno_email UNIQUE (email);
ALTER TABLE professores ADD CONSTRAINT unique_professor_email UNIQUE (email);
```

## **3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO**

Neste tópico, aplicamos técnicas de programação como orientação a objetos e programação funcional, que trouxeram modularidade e facilitaram a reutilização do código, permitindo um desenvolvimento mais estruturado e expansível. Para manter o código limpo e de fácil manutenção, seguimos práticas de clean code e utilizamos design patterns, o que garante clareza e facilita o trabalho em equipe.

O back-end foi desenvolvido em Node.js, utilizando as bibliotecas Express e Knex. O Express permitiu a criação e organização de rotas de forma eficiente, enquanto o Knex facilitou a integração com o banco de dados SQL, simplificando operações de manipulação e consulta de dados. Essa combinação de ferramentas trouxe escalabilidade e segurança ao sistema, além de otimizar o fluxo de desenvolvimento.

Para o front-end, utilizamos HTML e CSS puro, com foco em um design simples e responsivo. No CSS, aplicamos técnicas de responsividade, como media queries, garantindo que o sistema funcione bem em diferentes dispositivos, como desktops, tablets e smartphones. Essa abordagem oferece uma interface leve e acessível, mantendo o foco na funcionalidade e experiência do usuário.

### **3.2.1 APPLICATION PROGRAMMING INTERFACE ( API ) - BACK-END.**

A Interface de Programação de Aplicações (API) é um conjunto de padrões, ferramentas e protocolos que facilita a comunicação entre diferentes sistemas, permitindo que aplicações interajam umas com as outras. Esse recurso é fundamental para criar plataformas de software escaláveis, seguras e de fácil manutenção. Segundo Louzada, Carvalho e Laranja (2024), "de forma geral, é um conjunto de padrões, ferramentas e protocolos que permite a criação mais simplificada e segura de plataformas, pois permite a integração e a comunicação de softwares e seus componentes".

No desenvolvimento da API para o "Sistema de Gestão e Inteligência de Negócios para Organizações Sociais", aplicamos conceitos de arquitetura RESTful, que são essenciais para criar uma API eficiente, escalável e de fácil manutenção. Essa estrutura RESTful organizou os endpoints de forma clara, permitindo uma comunicação eficiente entre o front-end e o back-end.

A segurança dos dados foi uma prioridade, implementada através de autenticação e autorização com JSON Web Tokens (JWT). Esse recurso garante que apenas usuários autenticados possam acessar informações sensíveis, protegendo os dados e restringindo o uso de funcionalidades conforme o nível de acesso.

Para assegurar a funcionalidade da API, todas as rotas foram testadas no Postman, o que permitiu validar as respostas dos endpoints e identificar ajustes necessários. Seguimos boas práticas de desenvolvimento, mantendo o código modular para facilitar futuras expansões e adaptações, assegurando assim a evolução contínua e a robustez do sistema.

### **3.2.2 FRONT-END**

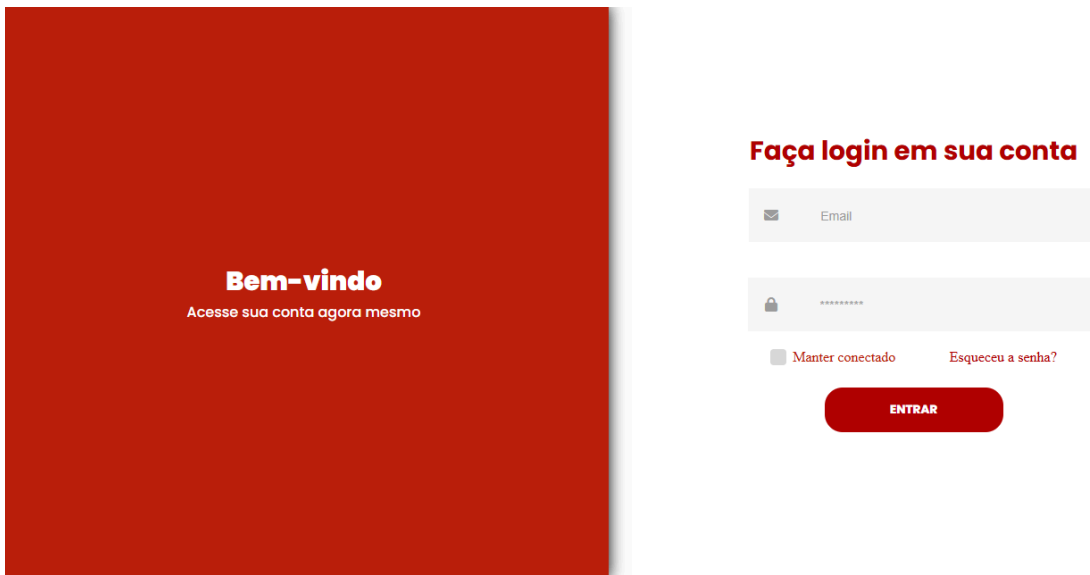
No desenvolvimento do front-end para o "Sistema de Gestão e Inteligência de Negócios para Organizações Sociais", aplicamos princípios de design responsivo para assegurar que o sistema seja acessível e funcional em diversos dispositivos, como computadores, tablets e smartphones. Utilizamos HTML e CSS puro, com técnicas de responsividade, como media queries, para garantir que a interface se ajuste automaticamente às diferentes resoluções de tela, proporcionando uma experiência uniforme para todos os usuários.

O design foi pensado com foco na experiência do usuário (UX) e na interface do usuário (UI), criando uma interface intuitiva e simples que facilita a navegação e o uso do sistema. Elementos de UI foram organizados de forma clara e objetiva, contribuindo para que os usuários finais – como administradores e colaboradores das ONGs – possam acessar informações e realizar operações com facilidade.

Embora não tenhamos utilizado frameworks modernos como Angular, a construção com HTML e CSS puro resultou em uma interface leve, eficiente e de fácil manutenção. A escolha de uma abordagem mais direta permitiu desenvolver uma interface funcional e prática, adequada às necessidades das organizações sociais envolvidas.



**Figura 1: Tela de login**

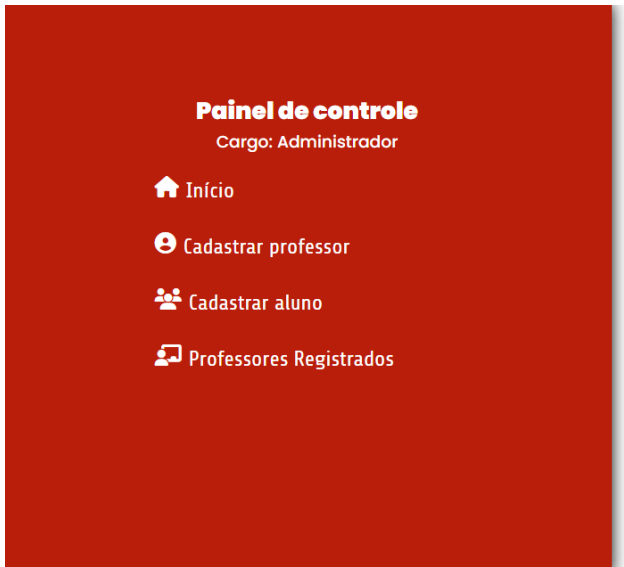


Fonte: Autores

**Figura 2: Validação de código de segurança**



**Figura 3: Tela de controle**



Fonte: Autores

**Figura 4: Tela de cadastro de professores**



Fonte: Autores

**Figura 5: Tela de cadastro de alunos**





Fonte: Autores

### Cadastro de aluno

Cadastre os alunos nas disciplinas

Nome do aluno

Email do aluno

Telefone do Aluno

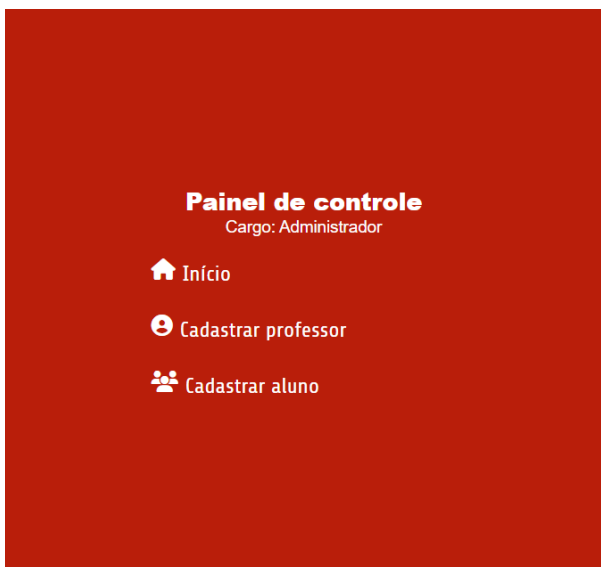
Instrumento

Matricula

dd/mm/aaaa

**CADASTRAR**

**Figura 6:** Professores cadastrados

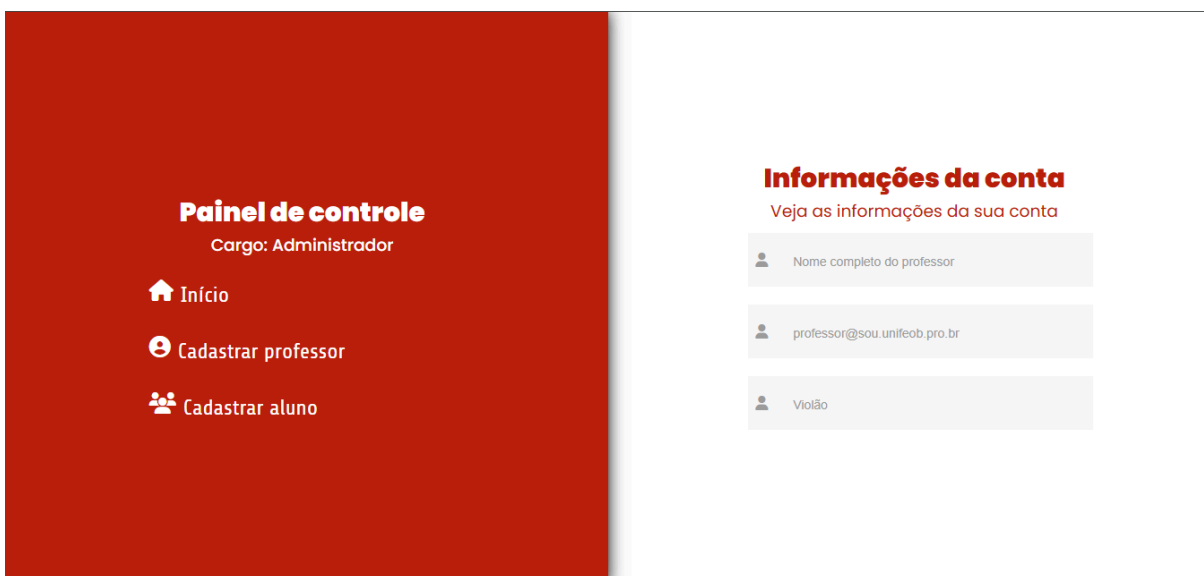


Fonte: Autores

### Professores Cadastrados

Aqui você pode visualizar, editar e excluir os professores registrados no sistema.

**Figura 7:** Informações da conta



Fonte: Autores

## 3.3 COMPUTAÇÃO EM NUVEM

### 3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

No desenvolvimento do projeto, foi criado um sistema de gestão que visa melhorar a administração, facilitar a comunicação com os alunos e organizar as aulas, proporcionando uma experiência mais eficiente e satisfatória para alunos e instrutores. Desde o início do desenvolvimento, visamos a utilização da computação em nuvem como uma solução estratégica, pois acreditamos que essa tecnologia pode potencializar ainda mais a eficácia do sistema.

Como afirma Armbrust (2009), “A computação em nuvem é um conjunto de serviços de rede, que proporciona escalabilidade, qualidade de serviço, infraestrutura barata de computação sob demanda, que pode ser acessado de uma forma simples.” Essa abordagem é fundamental para o sistema, pois permite que a infraestrutura se adapte rapidamente à demanda, garantindo que alunos e instrutores tenham acesso fácil e seguro a todos os recursos.

Além de diminuir os custos operacionais, a computação em nuvem permite uma melhor alocação dos recursos financeiros, tornando a gestão administrativa mais eficiente. A

nuvem é uma realidade atual e ajuda a atingir objetivos como a melhoria da eficiência operacional, permitindo que a organização acesse recursos de computação de forma rápida e flexível. Isso possibilita que, em períodos de grande demanda, como durante matrículas ou eventos especiais, a capacidade da empresa seja ampliada sem a necessidade de investimentos iniciais, como seria o caso de um servidor físico, que, por sua vez, necessitaria de recursos suficientes para funcionar adequadamente.

Dessa forma, a computação em nuvem não apenas otimiza a gestão administrativa, como também enriquece a experiência educacional, atendendo de forma mais eficiente às necessidades de alunos e instrutores.

### **3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO**

Como afirma Armbrust (2009), “A computação em nuvem é um conjunto de serviços de rede, que proporciona escalabilidade, qualidade de serviço, infraestrutura barata de computação sob demanda, que pode ser acessado de uma forma simples.” Dessa forma, podemos compreender que a computação em nuvem oferece uma série de benefícios significativos. Neste projeto, vamos analisar como esses benefícios serão aplicados para otimizar a gestão e a experiência educacional. Entre os exemplos de uso relevantes no projeto, destacam-se:

1. **Maior Flexibilidade:** Permite que a empresa se adapte rapidamente às mudanças no mercado e nas necessidades dos usuários, ajustando os recursos de forma dinâmica, por tanto quando o sistema estiver necessitando de mais recursos é facilmente resolvido.
2. **Redução de Custos Operacionais:** Elimina a necessidade de investimentos em hardware físico e manutenção constante, resultando em uma significativa redução de custos. Além disso, a empresa pagará apenas pelos recursos utilizados, e quando necessário como dito anteriormente pagará pelos recursos adicionais sem a necessidade de já ter esses recursos prontos como seria no caso de um servidor físico.
3. **Aumento da Segurança:** Um dos grandes casos de uso da computação em nuvem para a empresa é a segurança que ela proporciona. Os dados de professores e alunos, como notas e informações pessoais, são armazenados na nuvem, garantindo que permaneçam protegidos mesmo em caso de falhas no computador do usuário.

Assim, independentemente de problemas técnicos, as informações continuam seguras e acessíveis.

A Partir de todos os pilares citados acima, concluímos que a computação em nuvem apresenta uma série de aplicações e benefícios que podem transformar a gestão e a experiência educacional na empresa. A flexibilidade, a redução de custos e a segurança oferecidas por essa tecnologia tornam-na uma escolha estratégica para atender às demandas atuais e futuras. A adoção de soluções em nuvem permitirá que a empresa não apenas otimize seus processos, mas também melhore a experiência de aprendizado dos seus usuários.

### 3.3.3 VANTAGENS DA CLOUD COMPUTING

A computação em nuvem oferece diversas vantagens específicas para as empresas, especialmente para aquelas que buscam otimizar seus recursos, reduzir custos e melhorar a eficiência operacional. Durante o desenvolvimento deste projeto, foi realizado um estudo detalhado com o objetivo de identificar as melhores soluções que a tecnologia de nuvem pode oferecer à organização. Esse estudo considerou as necessidades atuais da empresa.

Durante o estudo realizado ficou claro que a redução de custos seria uma das principais vantagens da computação em nuvem. Além de também oferecer vantagens como elasticidade, que permite ajustar a capacidade do sistema conforme a necessidade, pagamento conforme o uso, o que significa que a empresa só paga pelos recursos efetivamente utilizados, alta disponibilidade, garantindo que o sistema esteja sempre disponível, sem interrupções; e segurança melhorada, que protege os dados e informações de forma mais eficiente, juntos, esses benefícios auxiliam a empresa a controlar melhor seus custos, otimizar recursos e oferecer um serviço de qualidade com maior segurança e flexibilidade.

A partir do estudo detalhado realizado, foi possível estimar os gastos mensais com os serviços essenciais oferecidos pela Amazon Web Services (AWS), totalizando aproximadamente 28,77 dólares por mês, o que resulta em um gasto anual de 345,24 dólares. Esse valor cobre os seguintes serviços:

1. **Amazon EC2:** Custo mensal de 4,24 dólares para uma instância do tipo t2.nano, com utilização de 9 horas por dia. Esse serviço oferece a flexibilidade necessária para o funcionamento do sistema de gestão de aulas, permitindo ajustar os recursos conforme a demanda.

2. **Amazon S3:** Custo mensal de 0,78 dólares para armazenar até 6 GB de dados, incluindo operações de PUT, COPY e GET. O Amazon S3 é utilizado para o armazenamento de dados essenciais do sistema.
3. **Amazon SNS:** Custo mensal de 0,95 dólares para enviar notificações e alertas, facilitando a comunicação entre professores, alunos e administradores. Esse serviço é crucial para a gestão e o envio de mensagens em tempo real.
4. **Amazon RDS for MySQL:** Custo mensal de 22,80 dólares para o serviço de banco de dados MySQL, utilizado para armazenar informações críticas, como dados de alunos, professores e horários das aulas.

Esse modelo de pagamento conforme o uso, juntamente com a escalabilidade da nuvem, permite que a empresa evite grandes investimentos iniciais em infraestrutura física e pague apenas pelos recursos efetivamente consumidos, exemplifica as vantagens que a computação em nuvem oferece à empresa.

Estudo que foi mencionado no texto:

Estimar URL: <https://calculator.aws/#/estimate?id=352007008eb854da93de25e7059225e0a8cd7090>

Resumo da estimativa		
Custo inicial	Custo mensal	Custo total de 12 months
0,00 USD	28,77 USD	345,24 USD
		Inclui um custo inicial

#### Estimativa detalhada

Nome	Grupo	Região	Custo inicial	Custo mensal
Amazon EC2	Nenhum grupo aplicado	South America (Sao Paulo)	0,00 USD	4,24 USD

Status: -

Descrição:

**Resumo da configuração:** Locação (Instâncias compartilhadas), Sistema operacional (Linux), Carga de trabalho (Consistent, Número de instâncias: 1), Instância do EC2 avançada (t2.nano), Pricing strategy (On-Demand Utilization: 9 Hours/Day), Habilitar monitoramento (desabilitada), EBS Quantidade de armazenamento (20 GB), DT Entrada: Internet (100 TB por mês), DT Saída: Not selected (0 TB por mês), DT Intranregião: (0 TB por mês)

Amazon Simple Storage Service (S3)	Nenhum grupo aplicado	South America (Sao Paulo)	0,00 USD	0,78 USD
------------------------------------	-----------------------	---------------------------	----------	----------

Status: -

Descrição:

**Resumo da configuração:** Armazenamento S3 Standard (6 GB por mês), Solicitações PUT, COPY, POST, LIST para S3 Standard (100), GET, SELECT e todas as outras solicitações do S3 Standard (100), Dados retornados pelo S3 Select (100 GB por mês), Dados verificados pelo S3 Select (100 GB por mês) DT Entrada: Internet (100 TB por mês), DT Saída: Amazon CloudFront (100 TB por mês)

### 3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

Com a base teórica adquirida nas aulas, foi possível concluir que o modelo de serviço mais adequado para o projeto seria o Software as a Service (SaaS). De acordo com Bastos (2023), “Nesse tipo de cloud computing, o provedor de serviço concede acesso a softwares que estão na nuvem, por meio de uma interface de navegador ou de programa.” Isso significa que o sistema de gestão será acessado pelos usuários (professores, alunos e administradores)



pela internet, sem a necessidade de instalação de programas adicionais, facilitando o uso e reduzindo custos com manutenção.

Outro ponto essencial para o bom funcionamento do sistema é o balanceamento de carga, que é crucial para garantir o desempenho, a disponibilidade e a eficiência da aplicação. O balanceamento de carga distribui as solicitações dos usuários de forma equilibrada entre diferentes servidores, evitando sobrecarga de um único servidor, o que poderia causar lentidão ou falhas no sistema. Além disso, esse processo assegura alta disponibilidade, redirecionando automaticamente o tráfego para servidores que estão funcionando em caso de falha de algum servidor, minimizando o tempo de inatividade. O balanceamento de carga também contribui para a redução de custos, uma vez que os recursos computacionais serão utilizados de maneira mais eficiente, pagando-se apenas pelo que for consumido.

Diversos componentes da arquitetura em nuvem foram analisados e selecionados para garantir o bom funcionamento do sistema. A seguir, explico como esses componentes trabalham de forma integrada para atender às necessidades da ONG:

1. **Computação – Amazon EC2:** Este serviço será utilizado para hospedar o sistema de gestão na nuvem. O EC2 oferece instâncias de servidores virtuais que podem ser configuradas e escaladas conforme a demanda de uso, ou seja, se o número de usuários aumentar, mais servidores poderão ser adicionados automaticamente para atender a essa demanda.
2. **Armazenamento de Dados – Amazon S3:** O S3 será responsável por armazenar dados como materiais didáticos, registros de alunos, notas e históricos. Esse serviço de armazenamento permite que a ONG tenha acesso a dados de forma rápida e segura, sem a necessidade de manter servidores físicos.
3. **Banco de Dados – Amazon RDS:** O RDS será usado para armazenar dados estruturados do sistema, como informações sobre alunos, professores, horários de aulas e registros de presença. Esse serviço facilita a administração do banco de dados, automatizando tarefas como backups e atualizações.
4. **Comunicação e Notificações – Amazon SNS:** O SNS será utilizado para enviar notificações em tempo real aos usuários do sistema, como avisos sobre novas aulas, mudanças de horário, faltas ou atualizações. Essa ferramenta garante que a comunicação entre professores, alunos e administradores seja eficiente e imediata.
5. **Balanceamento de Carga – Elastic Load Balancer (ELB):** O ELB vai distribuir o tráfego de usuários entre as instâncias do EC2, assegurando que nenhum

servidor fique sobrecarregado e garantindo que o sistema esteja sempre disponível, mesmo em caso de falhas em algum servidor.

A arquitetura em nuvem trabalha de forma integrada para assegurar que o sistema de gestão seja eficiente, escalável e seguro, atendendo às necessidades específicas da organização e dos usuários do sistema.

### 3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)

Foi realizado um estudo detalhado sobre os principais provedores de serviços em nuvem, com foco na análise de custo, desempenho e adequação às necessidades da organização. No estudo realizado, a Amazon Web Services (AWS) foi escolhida como a solução de nuvem mais adequada para a organização, devido aos seus recursos alinhados com os objetivos do projeto. A seguir, estão os principais benefícios oferecidos pela AWS:

1. **Maior Flexibilidade:** A AWS permite que a organização se ajuste rapidamente às mudanças na demanda de recursos, utilizando serviços como Amazon EC2. Isso garante que o sistema de gestão de aulas possa ser escalado facilmente, sem a necessidade de grandes investimentos iniciais, como ocorreria em servidores físicos.
2. **Redução de custos operacionais:** A AWS elimina a necessidade de investimentos em infraestrutura física, com seu modelo **pay-as-you-go** (pague conforme o uso). A organização paga apenas pelos recursos que utiliza e pode expandir a capacidade de forma rápida, conforme necessário, o que resulta em uma gestão de custos mais eficiente.
3. **Aumento da Segurança:** Com soluções como Amazon S3 e Amazon RDS, a AWS garante que os dados sensíveis, como informações de alunos e professores, sejam armazenados de forma segura e protegida contra falhas. A plataforma oferece criptografia e alta disponibilidade, garantindo a continuidade e a integridade dos dados.

Outro fator que foi de extrema importância é a confiabilidade da Amazon Web Services (AWS), que oferece uma infraestrutura robusta, com múltiplas zonas de disponibilidade e regiões geográficas distribuídas globalmente, o que garante alta redundância, que seria à duplicação de componentes críticos, como servidores, sistemas ou recursos de rede. Além disso, a AWS também oferece um suporte técnico especializado,

disponível 24/7, por meio de diversos canais, como chat, telefone e e-mail. O suporte pode ser personalizado conforme a necessidade do cliente.

Além de tudo que já foi citado a segurança e a conformidade foram aspectos essenciais na hora da escolha, A Amazon Web Services (AWS) oferece uma infraestrutura projetada para atender às mais rigorosas normas de segurança e regulamentações globais, garantindo que os dados armazenados e processados na nuvem estejam sempre protegidos.

### **3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING**

Com a base teórica adquirida nas aulas, foi possível concluir que o modelo de serviço mais adequado para o projeto seria o Software as a Service (SaaS). De acordo com Bastos (2023), “Nesse tipo de cloud computing, o provedor de serviço concede acesso a softwares que estão na nuvem, por meio de uma interface de navegador ou de programa.” Isso significa que o sistema de gestão será acessado pelos usuários pela internet, sem a necessidade de instalação de programas adicionais, facilitando o uso e reduzindo custos com manutenção. Nesse modelo, o software é hospedado na nuvem, e o usuário apenas utiliza o aplicativo sem se preocupar com sua instalação ou manutenção.

O balanceamento de carga é uma prática essencial para garantir o desempenho, a disponibilidade e a eficiência de sistemas hospedados na nuvem. Ele funciona distribuindo de maneira equilibrada as solicitações dos usuários entre múltiplos servidores. Isso evita que um único servidor seja sobrecarregado e, conseqüentemente, que o sistema sofra lentidão ou falhas. Quando um servidor falha, o balanceador de carga redireciona automaticamente o tráfego para servidores em funcionamento, garantindo que o serviço permaneça disponível e sem interrupções. Além disso, o balanceamento de carga contribui para a redução de custos operacionais, pois permite a utilização eficiente dos recursos da nuvem, ajustando a quantidade de servidores conforme a demanda, e pagando apenas pelos recursos utilizados. Desempenho se refere à rapidez e eficiência com que o sistema responde às solicitações dos usuários. Disponibilidade significa garantir que o sistema esteja sempre acessível, sem interrupções. Eficiência é a utilização otimizada dos recursos, garantindo que o sistema funcione sem desperdiçar recursos computacionais.

A anatomia de cloud computing é a estrutura que compõe a entrega dos serviços na nuvem. A arquitetura foi elaborada com base em diversos componentes que atuam em conjunto para assegurar a segurança, a escalabilidade e o desempenho do sistema. A seguir, são descritos os principais componentes e suas funções.

- **Amazon EC2:** Este serviço será utilizado para hospedar o sistema de gestão na nuvem. O EC2 oferece servidores virtuais configuráveis, que podem ser escalados automaticamente conforme o número de usuários ou a carga de trabalho, permitindo que a infraestrutura se ajuste à demanda. Servidores virtuais são computadores que não estão fisicamente presentes no local, mas são criados e gerenciados através da nuvem, com a capacidade de expandir ou reduzir recursos conforme necessário.
- **Amazon S3 (Simple Storage Service):** Serviço de armazenamento de dados na nuvem. Ele será utilizado para armazenar materiais didáticos, registros de alunos, notas e outros arquivos essenciais. O S3 oferece alta disponibilidade e segurança para o armazenamento de dados. Armazenamento de dados refere-se à capacidade de guardar informações de forma digital e acessível, e disponibilidade significa que esses dados estarão sempre acessíveis quando necessário.
- **Amazon RDS (Relational Database Service):** O RDS será usado para gerenciar o banco de dados, armazenando informações estruturadas como dados de alunos, professores e registros de aula. Ele oferece automação de backups e atualizações, facilitando a administração do banco de dados. Banco de dados é um sistema organizado para armazenar e acessar informações, enquanto backups são cópias de segurança dos dados para garantir que não sejam perdidos.
- **Amazon SNS (Simple Notification Service):** Este serviço será utilizado para enviar notificações em tempo real, como atualizações sobre horários de aulas, ausências ou eventos importantes. O SNS garante que todos os usuários, como professores e alunos, recebam as informações de forma imediata. Notificações em tempo real são mensagens instantâneas enviadas para informar os usuários sobre mudanças ou atualizações importantes no sistema.
- **Elastic Load Balancer (ELB):** O ELB será utilizado para distribuir o tráfego de usuários entre os servidores EC2, garantindo que nenhum servidor fique sobrecarregado, otimizando a performance do sistema e garantindo alta disponibilidade. O tráfego de usuários é o fluxo de dados entre os servidores e os dispositivos dos usuários, e o ELB distribui essas solicitações de forma equilibrada para evitar que um único servidor sofra excesso de trabalho.

Existem diversos paradigmas tecnológicos que sustentam a computação em nuvem, garantindo sua flexibilidade, escalabilidade e segurança. Paradigmas tecnológicos são modelos ou abordagens que fundamentam as tecnologias de nuvem. Entre os principais paradigmas, destacam-se:

- **Escalabilidade sob demanda:** A capacidade de aumentar ou diminuir os recursos de forma dinâmica, de acordo com a necessidade do sistema. Quando o número de usuários cresce, a nuvem permite que mais recursos sejam alocados automaticamente, sem a necessidade de uma intervenção manual. Escalabilidade é a habilidade de um sistema de aumentar ou diminuir seus recursos (como armazenamento ou processamento) conforme a demanda, sem afetar sua performance.
- **Modelo de pagamento por uso:** Na nuvem, o modelo de cobrança é baseado no uso real dos recursos, o que permite um controle mais eficiente dos custos. Isso significa que a organização paga apenas pelos recursos que utilizar, sem a necessidade de investir em infraestrutura física de servidores. Esse modelo é também chamado de pay-per-use (pague pelo que usar), sendo uma vantagem em relação à infraestrutura tradicional, onde é necessário fazer um investimento fixo.

Esses paradigmas são fundamentais para que a computação em nuvem seja uma solução econômica, eficiente e escalável, atendendo às necessidades da organização e oferecendo uma infraestrutura flexível para o sistema de gestão.

### 3.3.7 GOOGLE CLOUD ou AWS

Como afirma Armbrust (2009), “A computação em nuvem é um conjunto de serviços de rede, que proporciona escalabilidade, qualidade de serviço e infraestrutura barata de computação sob demanda, que pode ser acessado de uma forma simples.” Essa definição de computação em nuvem é crucial para o desenvolvimento do sistema proposto, pois reflete as vantagens que a utilização da nuvem oferece, como a flexibilidade e a capacidade de ajustar os recursos de acordo com a demanda.

Para atender às necessidades de escalabilidade e alta disponibilidade do sistema, foi proposta a utilização de serviços de computação em nuvem, com foco nas plataformas Google Cloud e AWS (Amazon Web Services), que são duas das principais opções disponíveis no mercado. Contudo, levando em consideração as características específicas do projeto e seus requisitos, optou-se por uma solução baseada no Google Cloud. Esta escolha visa garantir que o sistema seja eficiente, flexível e acessível para os usuários, independentemente do crescimento ou variação na demanda de serviços. O Google Cloud oferece uma série de funcionalidades essenciais para o sucesso do projeto, como armazenamento de dados,

processamento em tempo real e gerenciamento de banco de dados, que serão exploradas para assegurar um desempenho ideal e a continuidade dos serviços.

Ao aplicar o conceito de computação em nuvem neste projeto, diferentes cenários de implantação utilizando o Google Cloud foram considerados para otimizar os recursos conforme as necessidades específicas do sistema. A seguir, são apresentados alguns exemplos de como os recursos dessa plataforma serão aplicados para garantir a eficiência e a segurança do sistema:

1. **Armazenamento de Dados:** A plataforma Google Cloud Storage oferece uma solução de armazenamento seguro, escalável e de fácil acesso, que permitirá armazenar dados essenciais, como informações sobre alunos, professores, notas e frequência. Além disso, o uso de serviços de banco de dados em nuvem, como o Google Cloud SQL, assegura que esses dados sejam facilmente gerenciados, acessados e consultados conforme a necessidade.
2. **Processamento de Dados:** O serviço Google Compute Engine permitirá a criação de máquinas virtuais para processamento em tempo real, como a geração de relatórios de notas ou o acompanhamento da frequência dos alunos. Esse processamento pode ser ajustado automaticamente, conforme a demanda, otimizando o desempenho e garantindo que o sistema funcione de forma eficiente, mesmo em picos de acesso.
3. **Segurança e Controle de Acesso:** A plataforma Google Identity and Access Management (IAM) oferece ferramentas para gerenciar quem pode acessar os diferentes recursos dentro do Google Cloud. Isso é crucial para garantir a segurança dos dados sensíveis, como informações financeiras ou pessoais de alunos e professores, permitindo que apenas usuários autorizados tenham acesso a essas informações.

Todos esses processos trabalham de forma integrada para assegurar que o sistema de gestão seja eficiente, escalável e seguro, atendendo às necessidades específicas da organização e dos usuários do sistema.

### **3.4 ESTRUTURA DE DADOS**

No contexto do sistema de gestão desenvolvido para a ONG (Organização Não Governamental), é crucial fazer um levantamento de requisitos que servirá de base para a elaboração do projeto. Esse levantamento procura identificar as funcionalidades e as necessidades específicas do sistema, como o cadastro de alunos, professores, o agendamento de aulas e o controle de faltas. A partir dos requisitos, serão selecionadas as estruturas de dados mais adequadas para organizar e administrar as informações de forma eficiente. As escolhas das estruturas de dados, como listas, pilhas, filas, árvores e tabelas de dispersão (ou hash tables), serão feitas com base nas operações que o sistema precisa realizar e na linguagem de programação utilizada. Esse processo é fundamental para garantir que o sistema seja rápido, eficiente e atenda aos requisitos de desempenho exigidos pela ONG.

### **3.4.1 LEVANTAMENTO DE REQUISITOS**

Segundo Amoasei (2023), "podemos afirmar que um programa é composto de algoritmos e estruturas de dados, que juntos fazem com que o programa funcione como deve". A partir dessa definição, é possível compreender que a escolha correta das estruturas de dados foi um passo fundamental para garantir o funcionamento eficiente do sistema de gestão desenvolvido para a ONG (Organização Não Governamental). No contexto desse projeto, as estruturas de dados não só determinam como os dados são armazenados, mas também têm impacto direto no desempenho do sistema, como o tempo de resposta e o uso de memória, aspectos essenciais para o seu bom funcionamento.

A seleção das estruturas de dados para o sistema foi baseada no tipo de dado a ser manipulado, na frequência das operações realizadas, como inserção, remoção e busca, e nos requisitos de desempenho, como o tempo de resposta dessas operações e o uso de memória. Considerando que o sistema lida com grandes volumes de dados e exige alto desempenho, foi essencial escolher as opções mais eficientes em termos de complexidade algorítmica, conceito que se refere ao tempo necessário para a execução de uma operação, variando conforme o tamanho dos dados envolvidos. Assim, as decisões tomadas quanto às estruturas de dados asseguraram que as operações fossem realizadas de maneira eficiente, atendendo aos requisitos do sistema de forma otimizada.

As estruturas de dados escolhidas para o sistema incluem listas, pilhas, filas, árvores, grafos e tabelas de dispersão (ou hash tables). A seleção de cada uma dessas opções foi fundamentada nas características específicas dos dados e nas operações mais frequentes

dentro dos módulos do sistema, como o cadastro de alunos, o controle de faltas, o agendamento de aulas e o gerenciamento de turmas. A seguir, são apresentadas as justificativas para o uso de cada uma dessas estruturas:

1. **Listas:** Foi utilizada as listas para armazenar dados de forma sequencial, sendo especialmente úteis nos módulos de cadastro de alunos e professores. Como o número de registros pode variar com o tempo e as operações de inserção e remoção precisam ser feitas de maneira flexível, as listas são adequadas para lidar com esses requisitos. Elas proporcionam um acesso eficiente aos dados, permitindo a adição, remoção ou modificação de registros de maneira simples e eficaz.
2. **Pilha (Stack):** A pilha foi utilizada para gerenciar operações que dependem da ordem de execução, como o controle de alterações em registros ou o gerenciamento de ações temporárias. No modelo de pilha, a última ação realizada é a primeira a ser desfeita, o que facilita a reversão de ações no sistema, como alterações em cadastros de alunos ou ajustes no agendamento de aulas.
3. **Fila (Queue):** A fila foi escolhida para os módulos de agendamento de aulas e controle de faltas, uma vez que a estrutura FIFO (First In, First Out, ou "primeiro a entrar, primeiro a sair") é adequada para processos em que a ordem de chegada deve ser respeitada. No agendamento de aulas, isso garante que os alunos que se inscrevem primeiro sejam os primeiros a ser alocados nas turmas. No controle de faltas, a fila assegura que as ausências sejam registradas na ordem em que ocorreram.
4. **Árvores:** As árvores foram adotadas para representar a hierarquia de turmas e o calendário de aulas. As árvores são estruturas ideais para organizar dados de maneira hierárquica, o que facilita a busca por informações, como o horário de uma aula ou a organização de turmas. Além disso, as árvores permitem uma busca eficiente e estruturada, garantindo tempos reduzidos para encontrar informações dentro de uma estrutura organizada.
5. **Grafos:** Os grafos foram utilizados para modelar as relações complexas entre alunos, professores e aulas. Em um grafo, cada elemento é representado por um nó, e as conexões entre eles são feitas por arestas. Essa estrutura é ideal para mapear e analisar interações entre diferentes entidades do sistema, como o relacionamento entre alunos e as aulas em que estão matriculados. Os grafos possibilitam uma representação visual e eficiente dessas interações, facilitando o gerenciamento e a análise das relações no sistema.



**6. Tabelas de Dispersão (Hash Tables):** As tabelas de dispersão (ou hash tables) foram empregadas para garantir consultas rápidas e eficientes no sistema. Ao utilizar uma função de hash, uma chave única é mapeada para uma posição específica dentro da tabela, permitindo que operações de busca, inserção e remoção sejam realizadas de maneira extremamente rápida (tempo constante,  $O(1)$ ). Esse tipo de estrutura é especialmente útil no cadastro de dados, onde a busca rápida por registros é crucial para a eficiência do sistema.

A realização de uma escolha cuidadosa das estruturas de dados para o sistema de gestão da ONG foi fundamental para garantir a eficiência e o bom desempenho do sistema. Ao optar por listas, pilhas, filas, árvores, grafos e tabelas de dispersão, foi possível atender às necessidades específicas de cada módulo, como o cadastro de alunos e professores, o agendamento de aulas e o controle de faltas. Essas escolhas permitem que o sistema seja rápido, escalável e fácil de manter, além de possibilitar sua expansão conforme a ONG cresce e suas demandas se tornam mais complexas.

### **3.4.2 VALIDAÇÃO DOS REQUISITOS**

Para o sistema de gestão desenvolvido para a ONG (Organização Não Governamental), a validação da integração das estruturas de dados com a infraestrutura de computação em nuvem é um passo crucial. O objetivo é garantir que o armazenamento e o gerenciamento de grandes volumes de dados sejam realizados de forma eficiente, considerando as características dos ambientes de nuvem. A computação em nuvem permite que os dados sejam armazenados em servidores remotos e acessados de qualquer lugar, o que traz benefícios como maior escalabilidade e flexibilidade. Isso significa que o sistema pode crescer conforme a ONG aumenta sua base de alunos, professores e aulas, sem comprometer o desempenho.

Ao escolher as estruturas de dados, como listas, pilhas, filas, árvores e tabelas de dispersão, foi necessário considerar como essas estruturas funcionam em um ambiente de nuvem, onde os dados são distribuídos entre múltiplos servidores. Por exemplo, as tabelas de dispersão são eficientes para buscar dados rapidamente e, quando usadas em nuvem, ajudam a distribuir essas consultas de forma eficaz entre diferentes servidores, evitando sobrecarga em um único ponto. Da mesma forma, a escolha de árvores e filas no sistema de gestão garante

que as operações, como o controle de faltas ou o agendamento de aulas, sejam processadas de maneira ágil, mesmo quando o volume de dados cresce.

Outro ponto importante é garantir que as soluções adotadas atendam às necessidades de integridade dos dados, disponibilidade e eficiência. A integridade dos dados se refere à garantia de que os dados armazenados no sistema sejam precisos e não sejam corrompidos, mesmo em ambientes com grande tráfego. A disponibilidade garante que os dados estejam acessíveis a qualquer momento, enquanto a eficiência assegura que as operações de leitura e escrita, como quando um aluno se cadastra ou uma falta é registrada, ocorram rapidamente. No ambiente de nuvem, esses aspectos são alcançados por meio da elasticidade, que permite que a infraestrutura se ajuste automaticamente à demanda, aumentando ou diminuindo recursos conforme necessário, e da distribuição de carga, que distribui o processamento das operações entre diferentes servidores para evitar que algum servidor fique sobrecarregado.

Portanto, a integração das estruturas de dados com a computação em nuvem foi cuidadosamente planejada para garantir que o sistema de gestão da ONG seja escalável, eficiente e seguro. Com isso, o sistema estará preparado para lidar com volumes crescentes de dados e com as demandas de operação da ONG, sem comprometer a performance e a disponibilidade das informações.

### **3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS**

A Formação para a Vida é um dos eixos do Projeto Pedagógico de Formação por Competências da UNIFEQB.

Esta parte do projeto está diretamente relacionada com a extensão universitária, ou seja, o objetivo é que seja aplicável e que tenha real utilidade para a sociedade, de um modo geral.

#### **3.5.1 ENFRENTANDO ESTEREÓTIPOS**

**Tópico 1: Estereótipo e convívio social:** Este tópico explora como estereótipos afetam o ambiente universitário e nosso convívio social. Os estereótipos formam-se através de generalizações sobre grupos, o que, muitas vezes, leva a preconceitos e limita a aceitação da

diversidade. Exemplo prático: no ambiente universitário, a tendência de rotular estudantes com base na área de estudo, como associar um jaleco à área de saúde, reforça estereótipos que influenciam a visão e tratamento entre grupos.

**Tópico 2: Estereótipo e representação:** Explora como a sociedade e a mídia constroem imagens idealizadas, especialmente ligadas a gênero e aparência. Exemplo prático: a idealização do corpo perfeito nas mídias leva muitos jovens a buscar padrões de beleza que não refletem a realidade, causando problemas como baixa autoestima e transtornos alimentares.

**Tópico 3: Troco likes: a idealização da vida na internet:** Discute a vida idealizada nas redes sociais e o impacto dos influenciadores digitais na percepção de sucesso e felicidade. Exemplo prático: a busca por curtidas e seguidores leva as pessoas a mostrar apenas os aspectos mais atraentes de suas vidas, criando uma pressão social para aparentar perfeição, o que pode gerar ansiedade e uma visão distorcida de sucesso.

**Tópico 4: Convivendo com a diferença:** Aborda a importância de respeitar e aceitar as diferenças culturais, étnicas e geracionais. Enfrentar estereótipos étnicos e regionais no dia a dia é essencial para promover uma convivência mais inclusiva. Exemplo prático: em um ambiente de trabalho, o respeito pelas diferentes gerações e suas características contribui para um ambiente de colaboração e aprendizado mútuo, onde cada pessoa traz suas experiências únicas para o grupo.

### 3.5.2 ESTUDANTES NA PRÁTICA

A equipe produziu um podcast que aborda os estereótipos e percepções equivocadas frequentemente associadas às pessoas do interior do Brasil. A discussão explora como termos como “caipira” e “agroboy” são utilizados na mídia, muitas vezes de forma simplista, para representar uma realidade que não reflete a diversidade e complexidade das pessoas que vivem no campo. O podcast destaca como a mídia reforça esses estereótipos e como essas generalizações impactam a forma como os próprios moradores do interior se veem. Além disso, a equipe enfatiza a importância de reconhecer a diversidade cultural do interior e superar as visões limitadas que se têm das pessoas que lá vivem. O material pode ser acessado por meio do link abaixo para verificação e avaliação:

[https://youtu.be/iZWHD5nw9To?si=k1Gq4cN\\_NDddPZ1K](https://youtu.be/iZWHD5nw9To?si=k1Gq4cN_NDddPZ1K)

## 4. CONCLUSÃO

Neste Projeto Integrado (PI), desenvolvemos um “Sistema de Gestão e Inteligência de Negócios para Organizações Sociais” com o objetivo de aprimorar a eficiência operacional, escalabilidade e segurança dos dados para organizações como ONGs, instituições de caridade e cooperativas. Este projeto permitiu a aplicação prática dos conhecimentos adquiridos nas disciplinas de Estrutura de Dados, Linguagens e Técnicas de Programação, Tópicos Avançados de Banco de Dados, Computação em Nuvem e Projeto de Computação em Nuvem.

Ao longo do desenvolvimento, abordamos a definição dos requisitos do sistema, a modelagem utilizando UML e a implementação de uma infraestrutura robusta na nuvem da AWS. Utilizamos o Amazon RDS para o gerenciamento do banco de dados SQL Server, o que nos proporcionou um serviço gerenciado com alta disponibilidade, escalabilidade e segurança integrada. Além disso, desenvolvemos uma API com Node.js, Express e Knex, garantindo uma comunicação eficiente entre o front-end e o banco de dados.

Entre as principais dificuldades encontradas, destacamos a configuração adequada dos serviços em nuvem da AWS, em especial o Amazon RDS, para atender às necessidades específicas do projeto. A integração das diversas tecnologias utilizadas também foi desafiadora, exigindo um aprofundamento técnico e coordenação entre os membros da equipe. Além disso, assegurar a segurança e a confidencialidade dos dados em um ambiente de computação em nuvem exigiu a implementação rigorosa de boas práticas de segurança da informação.

Apesar dos desafios, conseguimos superar as barreiras por meio de pesquisa contínua, colaboração em equipe e aplicação de metodologias ágeis. O resultado foi um sistema funcional e eficiente, capaz de contribuir significativamente para a gestão das organizações sociais, facilitando a tomada de decisões estratégicas e ampliando o impacto positivo na sociedade.

Realizamos uma revisão completa dos textos, corrigindo erros de digitação e ortografia, e efetuamos uma última conferência na formatação para garantir a clareza e qualidade do documento. Também atualizamos o sumário para refletir todas as alterações realizadas.

## REFERÊNCIAS

MELODY RODRIGUES. **Computação em Nuvem: Estudo de Viabilidade**. *Disponível em:* [https://repositorio.utfpr.edu.br/jspui/bitstream/1/20004/2/CT\\_TELEINFO\\_XIX\\_2011\\_15.pdf](https://repositorio.utfpr.edu.br/jspui/bitstream/1/20004/2/CT_TELEINFO_XIX_2011_15.pdf). Acesso em: 04 nov. 2024, às 02:24.

Athena Bastos. **O que é e quais as vantagens do cloud computing (computação em nuvem)?:**

*Disponível em:*

[https://www.alura.com.br/empresas/artigos/cloud-computing?srsltid=AfmBOoqWH\\_tA5Ch-wswhoznRBjanGl\\_4CbJBP8K9wr4CnOBzygcegJ8](https://www.alura.com.br/empresas/artigos/cloud-computing?srsltid=AfmBOoqWH_tA5Ch-wswhoznRBjanGl_4CbJBP8K9wr4CnOBzygcegJ8). Acesso em: 08 nov. 2024, às 02:32.

Msc. Eliezio Soares. **PROGRAMAÇÃO COM ACESSO A BANCO DE DADOS**.

*Disponível*

*em:* <https://docentes.ifrn.edu.br/elieziosoares/disciplinas/programacao-com-acesso-a-banco-de-dados/10-modelo-logico>. Acesso em: 11 nov. 2024, às 14:30.

ERWIN. **Modelagem física de dados, esse é o terceiro**. *Disponível em:*

<https://www.erwin.com/br-pt/solutions/data-modeling/physical.aspx>. Acesso em: 11 nov. 2024, às 14:49.

Juliana Amoasei. **Estruturas de dados: uma introdução**. *Disponível em:*

<https://www.alura.com.br/artigos/estruturas-de-dados-introducao?srsltid=AfmBOorA1KnXk5m72xflFBTL4vMKkcpzhbiVgzAaPSOexTG0ghOCkcls#lista-ligada>. Acesso em: 24 nov. 2024, às 03:53.

LOUZADA, Vinícius; CARVALHO, Caroline; LARANJA, Emerson. **API: o que é, para quê serve e qual é a sua importância**. Disponível em:

<https://www.alura.com.br/artigos/api#:~:text=Em%20portugu%C3%AAs%2C%20significa%20Interface%20de,de%20softwares%20e%20seus%20componentes>. Acesso em: 24 nov. 2024, às 18:54.

## ANEXOS

Essa parte está reservada para os anexos, caso houver, como figuras, organogramas, fotos etc.

O estudante também deve enviar pela Intranet o Relatório Final das Atividades de Extensão, conforme modelo a seguir.





## RELATÓRIO FINAL DAS ATIVIDADES DE EXTENSÃO

### 1. IDENTIDADE DA ATIVIDADE

**RELATÓRIO:** Sistema de gestão e inteligência de negócios para organizações sociais

**CURSO:** Ciências da Computação

**MÓDULO:** Computação em Nuvem

**PROFESSOR RESPONSÁVEL:** Mariângela Martimbianco Santos

**ESTUDANTE:** Vinicius Felipe Tesser

**PERÍODO DE REALIZAÇÃO:** 08/2024 a 12/2024

### 2. DESENVOLVIMENTO

**Contextualização:** O objetivo deste projeto é prototipar um sistema de gestão personalizado para organizações sociais, incluindo Organizações Não Governamentais (ONGs), organizações de caridade e cooperativas, com um foco robusto em Business Intelligence (BI) para suportar a tomada de decisões estratégicas. A empresa beneficiada pelo projeto foi: ONG Projeto Beija Flor.

**Desafio:** Durante o desenvolvimento deste projeto, enfrentamos duas dificuldades principais. A primeira foi na construção e implantação da API na nuvem, que exigiu mais tempo do que o previsto devido a problemas com configurações e autenticação. A segunda dificuldade ocorreu ao utilizar o Angular para o frontend, que apresentou complexidade maior do que imaginávamos. Após algumas tentativas, optamos por substituir o Angular por outra tecnologia mais simples e adequada ao projeto. Apesar dos desafios, essas experiências proporcionaram aprendizados importantes e nos permitiram alcançar os resultados desejados.

**Cronograma das Ações:** Os encontros aconteceram de sexta-feira, nos seguintes dias: 09 de agosto de 2024; 16 de agosto de 2024; 23 de agosto de 2024; 30 de agosto de 2024; 06 de setembro de 2024; 13 de setembro de 2024; 20 de setembro de 2024; 27 de setembro de 2024; 04 de outubro de 2024; 11 de outubro de 2024; 18 de outubro de 2024; 25 de outubro de 2024; 01 de novembro de 2024; 08 de novembro de 2024; 15 de novembro de 2024; 22 de novembro de 2024; 29 de novembro de 2024; 06 de dezembro de 2024; 13 de dezembro de 2024; 20 de dezembro de 2024.

**Síntese das Ações:** Live de apresentação do cronograma dos Projetos Integrados; Organização e definição das equipes junto aos professores orientadores; Definição e apresentação das informações sobre a empresa a ser utilizada no projeto; Instruções sobre as questões metodológicas de referências, citações bibliográficas e formatação utilizadas no projeto de acordo com as Normas da ABNT; Período destinado ao desenvolvimento do projeto; Instruções sobre a atividade relacionada ao eixo de Formação para a Vida; Entrega da Formação para a vida (1ª entrega). Tópico 3.5 do Projeto Integrado; Entrega parcial para verificação da formatação do trabalho seguindo as normas da ABNT. (2ª entrega); Preenchimento do formulário sobre trabalho em equipe; Instruções sobre a elaboração e critérios de avaliação das apresentações dos projetos; Live sobre o preenchimento do relatório de extensão para entrega via sistema e início das entregas dos Projetos em PDF pelo Classroom; Entregar parte escrita completa com citações, referências e textos nas normas da ABNT; Preenchimento do segundo formulário sobre trabalho em equipe; Apresentações dos Projetos Integrados e período de retificações dos projetos reprovados.

**a. Aspectos positivos:** O projeto nos proporcionou aprendizados relevantes sobre computação em nuvem. Dominamos conceitos fundamentais dessa área, como a implementação e a gestão de recursos em nuvem, além de configurar ambientes escaláveis. Aperfeiçoamos nossas habilidades no uso de ferramentas e plataformas de nuvem para otimizar o desempenho e a segurança dos sistemas. Além disso, a necessidade de compreender as exigências do usuário e adaptar as soluções para a nuvem aprimorou nossa comunicação e colaboração em equipe.

**b. Dificuldades encontradas:** Foram encontrados alguns desafios durante o desenvolvimento, especialmente no contexto de computação em nuvem. Houve um desequilíbrio no trabalho, com algumas tarefas demandando mais tempo e esforço do que outras. Além disso, foi bastante desafiador configurar e otimizar a infraestrutura na nuvem, garantindo desempenho e escalabilidade adequados. Criar soluções eficientes e seguras, considerando as especificidades da nuvem, exigiu bastante esforço e adaptação, além da necessidade de um constante alinhamento com as melhores práticas para garantir o sucesso do projeto.

**c. Resultados atingidos:** O projeto resultou na criação de um software de gestão personalizado para atender às necessidades específicas da ONG Projeto Beija Flor. O software é benéfico para a ONG, pois oferece uma solução eficiente para o gerenciamento de suas operações, melhorando a organização e o acompanhamento das atividades. Além disso, ele proporciona uma base sólida que pode ser aprimorada e expandida conforme a demanda da ONG, facilitando o processo de tomada de decisões e a administração dos recursos de forma mais eficiente.

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

### 3. EQUIPE DOS ESTUDANTES NO PROJETO

RA : 23000093	NOME : Igor Garcia Nunes de Araujo
RA : 23000131	NOME : Igor Jose Ananias Campos
RA : 23001145	NOME : Joao Pedro Pizoli Carvalho
RA : 23000778	NOME : Vinicius Felipe Tesser