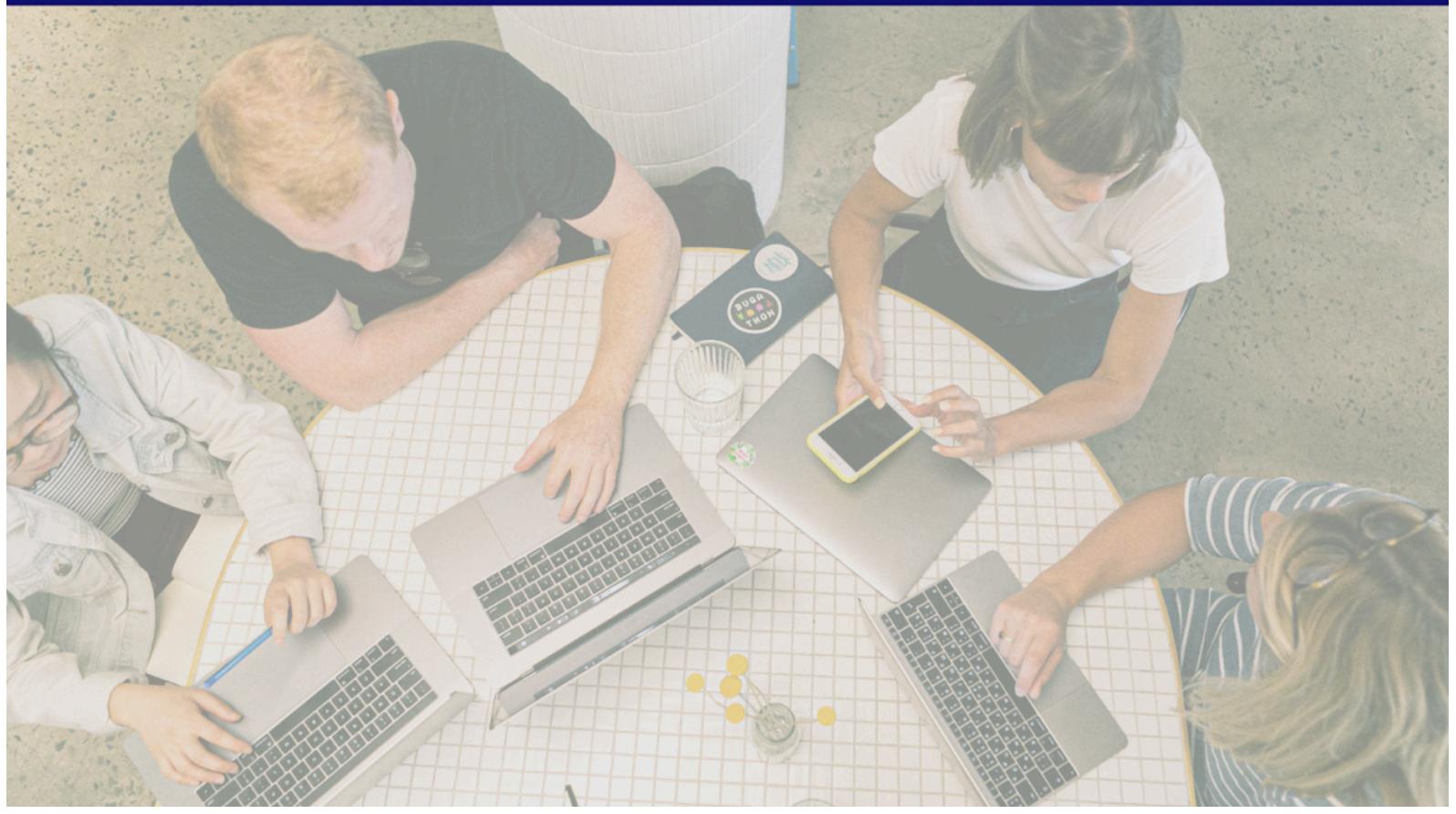




**UNifeob**  
| ESCOLA DE NEGÓCIOS

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

**NÚCLEO SOCIAL JOÃO DE BARRO**

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

**UNIFEOB**

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 – Prof<sup>a</sup>. Mariângela Martimbianco Santos

Estudantes:

Angelo Marcos Rodrigues Ridolfi Carvalho, 23001021

Caio Henrique Sibrão da Cruz, 23000666

Gabriel Marcos Freire, 23000669

Lanna Gabriela Gregghi Amaro, 23001112

Victor Venturini Neto, 23001159

SÃO JOÃO DA BOA VISTA, SP  
NOVEMBRO 2024

# SUMÁRIO

|   |    |
|---|----|
| 1. INTRODUÇÃO   | 4  |
| 2. DESCRIÇÃO DA EMPRESA   | 5  |
| 3. PROJETO INTEGRADO  | 6  |
| 3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS                         | 7  |
| 3.1.1 MODELO LÓGICO   | 8  |
| 3.1.2 MODELO FÍSICO   | 11 |
| 3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO                         | 15 |
| 3.2.1 APPLICATION PROGRAMMING INTERFACE ( API ) - BACK-END.     | 18 |
| 3.2.2 FRONT-END   | 19 |
| 3.3 COMPUTAÇÃO EM NUVEM   | 22 |
| 3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING                   | 22 |
| 3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO | 22 |
| 3.3.3 VANTAGENS DA CLOUD COMPUTING                              | 23 |
| 3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING                        | 25 |
| 3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)        | 27 |
| 3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING                        | 31 |
| 3.3.7 GOOGLE CLOUD ou AWS                                       | 36 |
| 3.4 ESTRUTURA DE DADOS  | 41 |
| 3.4.1 LEVANTAMENTO DE REQUISITOS                                | 43 |
| 3.4.2 VALIDAÇÃO DOS REQUISITOS                                  | 44 |
| 3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS  | 46 |
| 3.5.1 ENFRENTANDO ESTEREÓTIPOS                                  | 46 |
| 3.5.2 ESTUDANTES NA PRÁTICA                                     | 49 |
| 4. CONCLUSÃO  | 50 |
| REFERÊNCIAS   | 51 |
| ANEXOS  | 54 |

# 1. INTRODUÇÃO

Nesse projeto foi visada a continuidade do desenvolvimento do sistema personalizado para o Núcleo Social João de Barro, esse projeto se concentra em integrar o uso de disciplinas fundamentais, como estrutura de dados, linguagem e técnicas de programação, tópicos avançados de banco de dados e computação em nuvem com a documentação de engenharia de software e seus requisitos. Através dessas disciplinas, o sistema será projetado e otimizado para garantir eficiência e eficácia no gerenciamento das atividades noturnas da organização, que desempenha um papel crucial na comunidade ao oferecer apoio e recursos para pessoas em situação de vulnerabilidade.

Esse projeto buscará aprimorar o registro e monitoramento dos participantes, oferecendo um sistema robusto de chamada que controle com precisão o número de acolhidos e permita a inclusão de observações detalhadas sobre o atendimento psicopedagógico.

A implementação em um ambiente de nuvem trará vantagens significativas, como escalabilidade, segurança e acessibilidade, permitindo que o sistema cresça junto com as necessidades da OSC João de Barro. Ao alavancar esses recursos tecnológicos, o projeto visa fortalecer ainda mais o impacto positivo da organização na comunidade que serve, assegurando que as operações noturnas sejam geridas de maneira eficaz e contínua.

## **2. DESCRIÇÃO DA EMPRESA**

A Organização da Sociedade Civil (OSC) Núcleo Social João de Barro, registrada sob o CNPJ 04.893.486/0001-95, desempenha um papel ativo na comunidade de Santa Cruz das Palmeiras por mais de 22 anos. Seu principal propósito é o atendimento de crianças durante o contraturno escolar, oferecendo uma variedade de atividades e atendimento psicopedagógico.

Recentemente, a OSC inaugurou um novo programa voltado para famílias em situação de vulnerabilidade durante o período noturno. Esse programa visa oferecer apoio às famílias carentes, proporcionando atividades enriquecedoras e acesso a serviços de atendimento psicopedagógico.

A iniciativa do novo programa demonstra o contínuo compromisso do Núcleo Social João de Barro em adaptar-se às necessidades emergentes da comunidade e em ampliar seu alcance. Tal expansão reflete não apenas a evolução da organização ao longo do tempo, mas também sua dedicação em promover o bem-estar e o desenvolvimento integral dos indivíduos e famílias atendidas.

### 3. PROJETO INTEGRADO

O Núcleo Social João de Barro é uma Organização da Sociedade Civil (OSC) que atua de forma dedicada no apoio a pessoas em situação de vulnerabilidade social há mais de 20 anos, oferecendo serviços e recursos essenciais tanto no período diurno quanto noturno. Para potencializar ainda mais suas atividades noturnas, surge a necessidade de um sistema personalizado, capaz de gerenciar com eficiência as diversas demandas e processos envolvidos na gestão da OSC.

O sistema a ser desenvolvido será uma aplicação web, estruturada com base em conceitos sólidos de Engenharia de Software e utilizando tecnologias avançadas. A escolha do TypeScript como linguagem de desenvolvimento reflete a necessidade de um código mais robusto e escalável, beneficiando a implementação de estruturas de dados eficientes que suportem o gerenciamento de grandes volumes de informações. Em termos de infraestrutura, o sistema será hospedado em uma plataforma de computação em nuvem, com opções entre Google Cloud ou AWS, garantindo alta disponibilidade, segurança e a capacidade de escalabilidade conforme o crescimento das operações do Projeto João de Barro.

No que se refere ao banco de dados, a utilização de MySQL permitirá a criação de um ambiente de armazenamento de dados otimizado, com suporte a transações complexas e a capacidade de realizar consultas eficientes.

Para o backend do sistema, será utilizado TypeScript com Knex e Node.js, que oferece um ambiente eficiente para o desenvolvimento de APIs e a gestão de interações com o banco de dados. Já para o front-end, o Angular será a ferramenta escolhida, proporcionando uma interface de usuário dinâmica e responsiva, capaz de melhorar a experiência dos usuários ao interagir com o sistema.

Cada uma dessas disciplinas desempenha um papel crucial no sucesso do projeto. Estruturas de Dados fornecem a base para o armazenamento e acesso rápido às informações, enquanto a Computação em Nuvem assegura que o sistema esteja sempre disponível e seguro. O MySQL nos Tópicos Avançados de Banco de Dados garante um banco de dados robusto, enquanto a combinação de Node.js e Angular em Linguagem e Técnicas de Programação facilita o desenvolvimento de um sistema web que seja tanto funcional quanto intuitivo. Juntos, esses componentes formarão um sistema integrado, eficiente e preparado para atender às necessidades crescentes do Núcleo Social João de Barro.

### 3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

O projeto se concentra em organizar e otimizar as atividades e informações relacionadas às crianças e suas famílias, exigindo um armazenamento de dados que suporte transações complexas e consultas eficientes, refletindo a dinâmica das operações da organização.

Considerando a complexidade dos dados a serem geridos, a utilização de um banco de dados relacional se apresenta como a solução mais adequada. O MySQL foi escolhido como sistema de gerenciamento de banco de dados (SGBD) para este projeto, oferecendo uma estrutura robusta para a criação de tabelas, definição de relacionamentos e implementação de regras de integridade que asseguram a consistência das informações. Essa abordagem permite o armazenamento eficiente de dados relacionados a atividades, aprovações, beneficiários e usuários, facilitando o acesso e a manipulação de informações críticas.

Os bancos de dados relacionais foram desenvolvidos com a finalidade de dissociar o armazenamento físico dos dados de sua representação conceitual, oferecendo uma base matemática para os bancos de dados. Este modelo também introduziu linguagens de consulta de alto nível, que possibilitaram uma alternativa às interfaces tradicionais de programação, facilitando a escrita de novas consultas. Inicialmente, os sistemas relacionais foram criados para atender às mesmas demandas dos sistemas anteriores, mas sua flexibilidade para o desenvolvimento ágil de novas consultas e a reorganização do banco de dados diante de alterações nos requisitos foram decisivas. Embora os primeiros sistemas relacionais experimentais tenham surgido no final dos anos 70, os sistemas de gerenciamento de banco de dados relacional (SGBDR) introduzidos no início dos anos 80 apresentavam desempenho lento, uma vez que não utilizavam ponteiros para o armazenamento físico ou registros de localização para acessar os dados relacionados. O aprimoramento nas técnicas de armazenamento, indexação e no processamento de consultas melhorou significativamente seu desempenho, tornando os bancos de dados relacionais os sistemas dominantes em aplicações tradicionais de banco de dados, hoje presentes em diversos computadores, desde pessoais até grandes servidores (ELMASRI; NAVATHE, 2005).

Ademais, a implementação do banco de dados em um ambiente de nuvem possibilitará benefícios significativos, como escalabilidade e segurança. Ao optar pelo provedor de nuvem AWS, o projeto assegura que o sistema esteja sempre disponível, permitindo a expansão das operações conforme necessário, sem comprometer a performance. Essa flexibilidade é

essencial para atender às demandas em constante mudança da OSC João de Barro, especialmente na gestão de atividades noturnas e no controle de presença dos beneficiários.

Compreendendo as necessidades específicas do projeto, a equipe buscará uma solução de banco de dados que não apenas suporte as funcionalidades desejadas, mas que também proporcione uma base sólida para o crescimento futuro. A integração de conceitos de banco de dados relacional com a infraestrutura em nuvem garantirá que a organização possa continuar a oferecer serviços de qualidade às crianças e suas famílias, ampliando seu impacto positivo na comunidade.

### **3.1.1 MODELO LÓGICO**

O Projeto Lógico é uma etapa crucial no desenvolvimento do sistema de gestão para o Núcleo Social João de Barro, pois permite mapear os modelos de entidade-relacionamento necessários para a estrutura do banco de dados. Nesta fase, foi trabalhado a criação de modelos internos que descrevem detalhadamente como as informações serão armazenadas e interligadas, garantindo que o banco de dados atenda às necessidades específicas da organização.

Durante a elaboração do projeto lógico, foram definidas as tabelas, seus relacionamentos e as regras de integridade que irão governar os dados. Essa definição cuidadosa é essencial para manter a integridade e o desempenho do sistema. Por exemplo, serão criadas tabelas para os beneficiários, usuários, presenças, aprovações de atividades e atividades em si. Cada tabela terá metadados específicos, como tipos de dados, tamanhos e obrigatoriedade de preenchimento das colunas, permitindo uma estrutura organizada e acessível.

Além disso, a documentação dos metadados e a criação de visões simplificam a manutenção e o uso do banco de dados, oferecendo aos usuários uma maneira intuitiva de acessar as informações. Isso é especialmente importante para a ONG, que precisa controlar a presença dos alunos e gerenciar as atividades oferecidas, assegurando que todos os registros sejam precisos e facilmente recuperáveis.

O modelo lógico detalha a estrutura das tabelas e seus relacionamentos, atendendo às necessidades específicas da organização. Abaixo estão as principais tabelas e suas descrições:

1. usuario: contém informações sobre os usuários do sistema, suas credenciais de acesso e o papel desempenhado na organização.

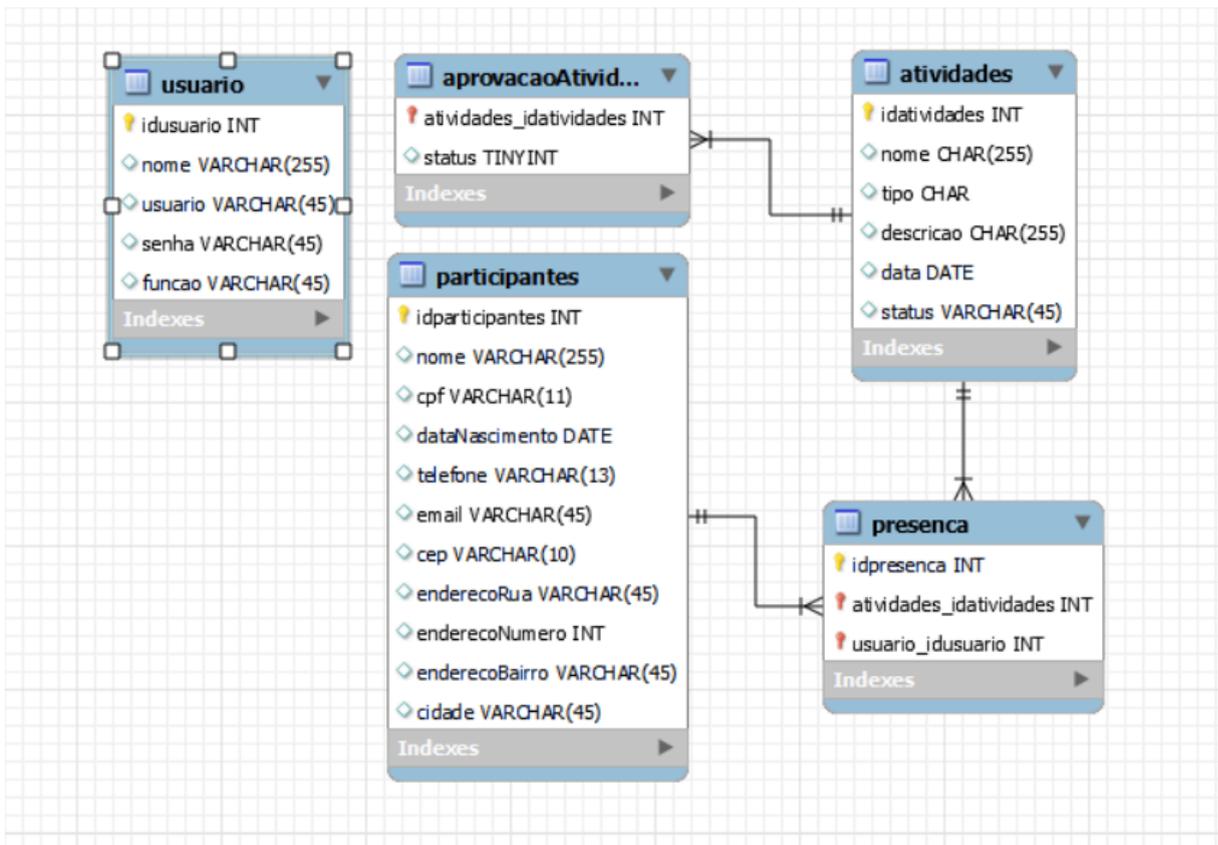
- Campos:
  - idusuario (INT): identificador único do usuário.
  - nome (VARCHAR 255): nome completo do usuário.
  - usuario (VARCHAR 45): nome de usuário para login.
  - senha (VARCHAR 45): senha do usuário, armazenada de forma segura.
  - funcao (VARCHAR 45): papel do usuário no sistema (por exemplo, direção, professor, coordenação).
- 2. atividades: armazena as informações sobre as atividades realizadas pelo Núcleo Social, como descrição e status de aprovação.
  - Campos:
    - idatividades (INT): identificador único da atividade.
    - nome (VARCHAR 255): nome da atividade.
    - tipo (CHAR): tipo da atividade (ex.: educacional, recreativa).
    - descricao (CHAR 255): breve descrição da atividade.
    - data (DATE): data da atividade.
    - status (VARCHAR 45): status atual da atividade (ex.: agendada, concluída, cancelada).
- 3. aprovacaoAtividades: controla o status de aprovação das atividades, ligando a atividade ao seu status de aprovação.
  - Campos:
    - atividades\_idatividades (INT): chave estrangeira referenciando o identificador da atividade em atividades.
    - status (TINYINT): indica o status de aprovação (0 para não aprovada e 1 para aprovada).
- 4. participantes: guarda informações dos beneficiários da organização, incluindo dados de contato e endereço.
  - Campos:
    - idparticipantes (INT): identificador único do participante.
    - nome (VARCHAR 255): nome completo do participante.
    - cpf (VARCHAR 11): CPF do participante.
    - dataNascimento (DATE): data de nascimento do participante.
    - telefone (VARCHAR 13): número de telefone.
    - email (VARCHAR 45): e-mail de contato.
    - cep (VARCHAR 10): CEP do endereço.

- enderecoRua (VARCHAR 45): rua do endereço.
  - enderecoNumero (INT): número do endereço.
  - enderecoBairro (VARCHAR 45): bairro do endereço.
  - cidade (VARCHAR 45): cidade de residência.
5. presença: registra a presença dos participantes nas atividades, associando cada registro a um usuário e a uma atividade específica.
- Campos:
    - idpresenca (INT): identificador único da presença.
    - atividades\_idatividades (INT): chave estrangeira referenciando a atividade.
    - usuario\_idusuario (INT): chave estrangeira referenciando o usuário responsável pelo registro.

#### Relacionamentos entre as Tabelas

- usuario - presenca: o relacionamento entre essas tabelas permite identificar qual usuário registrou a presença dos participantes.
- atividades - aprovacaoAtividades: define o status de aprovação de uma atividade específica, permitindo que apenas atividades aprovadas sejam realizadas.
- participantes - presenca: possibilita o controle de frequência dos participantes nas atividades organizadas pela OSC.

O resultado do Projeto Lógico será um esquema de banco de dados que se assemelha ao modelo conceitual, mas com muito mais detalhes técnicos. Esse esquema incluirá informações sobre como as tabelas se relacionam, como no caso do relacionamento entre usuário e presença, beneficiário e usuário, usuário e aprovação de atividades, e atividades com presenças. Dessa forma, a equipe garantirá que o banco de dados não apenas suporte às funcionalidades requeridas pelo sistema, mas também ofereça uma base sólida e escalável para o gerenciamento das operações do Núcleo Social João de Barro. Com um projeto lógico bem definido, a implementação futura se torna mais fluida e eficiente, minimizando potenciais problemas e maximizando a eficácia do sistema.



Modelo lógico do banco de dados utilizando *MySQL*.

### 3.1.2 MODELO FÍSICO

O Projeto Físico representa a etapa final do desenvolvimento do banco de dados, onde são definidos os detalhes técnicos necessários para sua implementação efetiva. Nesta fase, a equipe se concentra em como os dados serão armazenados e organizados, além de elaborar os scripts que criarão os objetos do banco de dados, como tabelas, visões, colunas, funções, triggers e stored procedures. Essa etapa está intrinsecamente ligada ao Sistema de Gerenciamento de Banco de Dados (SGBD) que será adotado, como o *MySQL*, que servirá como a base para as operações de armazenamento e recuperação de dados.

Para atender às necessidades do Núcleo Social João de Barro, as tabelas do banco de dados são projetadas para armazenar informações cruciais sobre atividades, aprovações, beneficiários e usuários. Cada tabela terá colunas específicas que determinam o tipo de dados que serão guardados, como o nome das atividades, dados pessoais dos beneficiários e informações pertinentes aos usuários do sistema. A implementação dessas tabelas será realizada através de scripts que incluem instruções para a criação de todos os elementos

necessários, garantindo que a estrutura do banco de dados esteja alinhada com o modelo lógico previamente definido.

Os triggers desempenham um papel importante na manutenção da integridade dos dados, garantindo que, ao aprovar ou remover uma atividade, o status correspondente na tabela de atividades seja atualizado automaticamente. Essa automação minimiza a necessidade de intervenções manuais, ajudando a manter as informações consistentes e confiáveis.

A *trigger* `after_insert_aprovacaoatividades` foi criada para acionar a stored procedure `alterar_status_aprovacao` automaticamente, sempre que um novo registro for inserido na tabela `aprovacaoatividades`. Dessa forma, a atualização do status da atividade é feita de maneira imediata e consistente, eliminando a necessidade de processos manuais. Abaixo está o código completo da trigger:

```
DELIMITER $$
USE `bancopjb` $$
CREATE DEFINER=`root`@`localhost` TRIGGER `bancopjb`.`after_insert_aprovacaoatividades`
AFTER INSERT ON `bancopjb`.`aprovacaoatividades`
FOR EACH ROW
BEGIN
    CALL alterar_status_aprovacao(NEW.atividades_idatividades);
END$$
DELIMITER ;
```

Codificação da Trigger em MySQL

O comportamento da trigger é definido para ser executado após (*AFTER*) a inserção de um novo registro na tabela `aprovacaoatividades`. Ela utiliza o valor da coluna `atividades_idatividades` do registro recém-inserido (*NEW*) como parâmetro ao chamar a stored procedure. Isso assegura que toda nova aprovação registrada automaticamente atualize o status correspondente na tabela de atividades.

As stored procedures também são uma parte fundamental deste projeto, sendo utilizadas para realizar operações controladas, como a inserção de dados nas tabelas e a atualização de status de atividades e aprovações. Essa centralização da lógica de negócios não apenas torna o sistema mais coeso, mas também simplifica a manutenção, permitindo que alterações possam ser feitas de maneira mais eficiente.

No sistema de gestão, a *stored procedure* `alterar_status_aprovacao` desempenha o papel de automatizar a atualização do status das atividades no banco de dados. Seu principal

objetivo é alterar o campo status de uma atividade específica para o valor 1, que simboliza que a atividade foi aprovada. A procedure recebe um parâmetro chamado p\_id, que representa o identificador da atividade. Abaixo está o código completo da stored procedure:

```
DELIMITER $$
USE `bancopjb` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `alterar_status_aprovacao`(p_id INT)
BEGIN
    UPDATE atividades SET atividades.status = 1 WHERE idatividades = p_id;
END $$
DELIMITER ;
```

Codificação da Stored Procedure em MySQL

Este código é encapsulado dentro de uma estrutura que garante que a lógica possa ser reutilizada em diferentes momentos do sistema, reduzindo redundâncias. Quando invocada, a procedure executa um comando UPDATE na tabela atividades, localizando o registro correspondente ao ID fornecido e alterando seu status.

Outro aspecto crucial abordado nesta fase é a definição das permissões de acesso dos usuários. O sistema é projetado para assegurar que o cargo de direção tenha acesso total a todas as funcionalidades do banco de dados, incluindo a aceitação e exclusão de atividades. Por outro lado, os professores e membros da coordenação têm acesso a quase todas as funções, exceto à capacidade de aprovar ou negar atividades. Essa estrutura de permissões é vital para garantir que apenas usuários autorizados possam acessar ou modificar informações sensíveis, aumentando assim a segurança do sistema e protegendo dados críticos relacionados às operações do Núcleo Social João de Barro.

O schema bancopjb foi projetado para organizar os dados do sistema de gestão, contemplando tabelas essenciais como atividades, aprovacaoatividades, participantes, presenca e usuario. Esses elementos formam a base para a gestão de atividades, aprovação, controle de presença e informações dos usuários.

```
CREATE SCHEMA IF NOT EXISTS `bancopjb` DEFAULT CHARACTER SET utf8mb3 ;
USE `bancopjb` ;

CREATE TABLE IF NOT EXISTS `bancopjb`.`atividades` (
  `idatividades` INT NOT NULL AUTO_INCREMENT,
  `nome` CHAR(255) NULL DEFAULT NULL,
  `tipo` CHAR(45) NULL DEFAULT NULL,
  `descricao` CHAR(255) NULL DEFAULT NULL,
  `data` DATE NULL DEFAULT NULL,
  `status` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idatividades`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb3;
```

Codificação em MySQL da criação do Schema

A tabela “atividades” armazena as informações principais de cada atividade, como nome, tipo, descrição, data e status. Esse status é atualizado automaticamente através de triggers e *stored procedures* para refletir o progresso da atividade no sistema.

Com o Projeto Físico bem definido, a implementação do banco de dados será mais fluida, permitindo que a equipe garanta a eficácia e a eficiência do sistema na gestão das informações e no suporte às atividades da organização.

## 3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

Como mencionado anteriormente, o sistema de gestão que foi desenvolvido para a organização social João de Barro, utiliza TypeScript com Node.js. A API oferece funcionalidades diferentes de acordo com o perfil do usuário (administrador ou coordenação/professor). Enquanto o administrador tem acesso a todas as funcionalidades do sistema, os coordenadores e professores possuem permissões restritas, com acesso a recursos como a listas de presença, cadastro/atualização de participantes, submissão de atividades para aprovação, e visualização do calendário. Somente o administrador pode aprovar atividades, que são automaticamente adicionadas ao calendário.

A abordagem OOP (Object-Oriented Programming) oferece uma estrutura modular e escalável, com ênfase na criação de entidades independentes e reutilizáveis. Ao modelar usuários (admin, coordenadores, professores) como classes com diferentes permissões, o sistema assegura melhor controle de acesso e gerenciamento de funcionalidades. A herança pode ser usada para compartilhar comportamentos comuns entre diferentes perfis de usuários, enquanto o polimorfismo permite a personalização de funções específicas.

Com a OOP, é possível criar classes reutilizáveis para manipulação de usuários, gerenciamento de atividades e controle de acesso. Isso facilita a manutenção e escalabilidade do sistema.

A programação funcional, amplamente suportada pelo JavaScript e TypeScript, oferece uma maneira eficiente de lidar com a imutabilidade e a composição de funções puras. Ao trabalhar com dados (como usuários, atividades, presenças) de maneira funcional, você evita efeitos colaterais indesejados, tornando o sistema mais previsível e fácil de depurar.

Falando ainda sobre programação funcional, ela facilita a manipulação de dados de forma declarativa e legível, especialmente útil em casos de processamento em lote de dados ou filtros aplicados a grandes conjuntos de usuários ou atividades.

Utilizando Node.js, a aplicação aproveita o modelo de programação assíncrona e não bloqueante. Isso permite lidar com múltiplas requisições simultâneas sem que o sistema sofra gargalos de desempenho. A concorrência é especialmente importante no processamento de listas de presença ou aprovação de múltiplas atividades de forma paralela.

Os sistemas para web desenvolvidos nativamente sobre a plataforma .NET, Java, PHP, Ruby ou Python possuem uma característica em comum: eles paralisam um processamento

enquanto utilizam um I/O (Input/Output) no servidor. Essa paralisação é conhecida como modelo bloqueador (*Blocking-Thread*).

Baseado neste problema, no final de 2009, Ryan Dahl criou o node.js, com a ajuda de 14 outros colaboradores. O node.js possui um modelo inovador, sendo sua arquitetura totalmente non-blocking-thread. Esse tipo de arquitetura resulta em uma boa performance com relação ao consumo de memória, já que usa ao máximo e de forma eficiente o poder de processamento dos servidores, principalmente em sistemas que produzem uma alta carga de processamento. A utilização de Promises e async/await possibilita um fluxo de controle mais fácil de ler, enquanto a performance do sistema é otimizada ao evitar bloqueios desnecessários de I/O.

É claro que, organizar o código em módulos, como controladores de usuários, atividades e presenças, facilita o desenvolvimento colaborativo e a manutenção futura. Cada módulo é responsável por uma parte específica do sistema, seguindo o princípio de responsabilidade única (SRP).

Componentes como middlewares para autenticação, classes de usuários e funções para validação de dados são exemplos de trechos de código reutilizáveis, que podem ser aplicados em diferentes partes da aplicação. A programação concorrente, somada à modularidade e à reutilização de componentes, contribui para um sistema mais rápido e eficiente, principalmente quando utilizado em larga escala.

Outro aspecto fundamental para a sustentabilidade a longo prazo do sistema é a clareza no código. Boas práticas de clean code, como nomes descritivos, funções curtas e responsabilidades bem definidas, ajudam a manter o código legível e fácil de entender para novos desenvolvedores que venham a trabalhar no projeto futuramente.

Num sistema como esse, também é muito importante o uso de design patterns que são soluções reutilizáveis para problemas comuns encontrados durante o desenvolvimento de software. Eles não são pedaços de código prontos, mas sim guias para estruturar o código de forma mais eficiente e organizada. A aplicação de design patterns melhora a modularidade, manutenção e flexibilidade de sistemas.

Aqui são alguns design patterns que utilizamos na criação do sistema:

- **Factory Pattern:** Pode ser usado para a criação de diferentes tipos de usuários (admin, coordenadores, professores) de forma flexível, facilitando a extensão do sistema.

- **Repository Pattern:** Facilita o acesso e manipulação dos dados, abstraindo a lógica de consultas ao banco de dados, melhorando a manutenção e promovendo a reutilização de código.

Podemos também, discutir brevemente algumas das ferramentas e frameworks utilizados:

- **Node.js:** O ambiente de execução assíncrono do Node.js, aliado à capacidade de lidar com múltiplas requisições simultaneamente, o torna ideal para o desenvolvimento de sistemas web escaláveis.
- **TypeScript:** A tipagem estática do TypeScript aumenta a robustez e a segurança do código, ajudando a identificar erros durante o desenvolvimento.
- **Express.js:** Este framework web minimalista simplifica a criação de rotas e a gestão de middlewares, acelerando o desenvolvimento da aplicação.
- **Bibliotecas para Autenticação e Segurança:** O uso de bibliotecas como `'jsonwebtoken'` para JWT, `'bcrypt'` para hashing de senhas, e middlewares de autenticação, garantem a segurança das operações de login e controle de acesso.

O Angular foi utilizado para o desenvolvimento do front-end do sistema, proporcionando uma interface dinâmica e reativa que facilita a interação dos usuários com as funcionalidades disponíveis. Angular, como um framework front-end baseado em TypeScript, oferece uma estrutura poderosa para o desenvolvimento de aplicações SPA (Single Page Applications). Suas funcionalidades, como a injeção de dependências e a modularidade, alinham-se perfeitamente com os princípios de Clean Code e padrões de design, assegurando que o front-end seja escalável e fácil de manter.

O uso de componentes reutilizáveis no Angular permite uma integração eficiente com a API, garantindo uma experiência de usuário fluida e responsiva. Além disso, o Angular oferece suporte nativo para serviços e módulos, facilitando a comunicação com o back-end baseado em Node.js e promovendo uma arquitetura bem organizada e modular.

Por fim, a combinação de técnicas avançadas de programação, como OOP e FP, com práticas de clean code e padrões de design, assegura que o sistema de gestão de pessoas seja robusto, escalável e sustentável a longo prazo. O uso de Node.js e TypeScript, juntamente com frameworks e bibliotecas eficientes, contribui para o rápido desenvolvimento e manutenção do sistema, garantindo sua eficácia e segurança.

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

O backend do "Sistema de Gestão e Inteligência de Negócios para Organizações Sociais" foi desenvolvido com uma arquitetura baseada em API RESTful, utilizando Node.js e TypeScript. Essa abordagem fundamenta-se na filosofia REST, projetada para criar serviços escaláveis e de fácil manutenção, permitindo uma interação estruturada e harmoniosa entre os componentes do sistema.

REST, como definido por Roy Fielding, baseia-se em uma abordagem arquitetural restritiva, que parte de um sistema sem restrições e, progressivamente, adiciona restrições aos componentes individuais até que eles se integrem de forma coesa. Diferentemente de um design emergente do zero, REST considera um conjunto completo de necessidades do sistema, impondo restrições a cada elemento da arquitetura para que todos atuem de forma harmoniosa, promovendo a robustez e a modularidade do sistema. Essa organização permite que as APIs RESTful implementem essas restrições, de modo a garantir que a estrutura do backend do sistema se mantenha coerente com as necessidades da organização e os princípios fundamentais da arquitetura web.

A API foi organizada de maneira modular, com a pasta principal ``src`` contendo subpastas para cada camada funcional. Dentro de ``controllers``, foram centralizadas as operações de controle, responsáveis por coordenar a comunicação entre a camada de rotas e a lógica de negócio. Na pasta ``data``, encontra-se a configuração de acesso e manipulação do banco de dados, essencial para integrar a API com a estrutura de dados distribuída. A implementação de middlewares na pasta ``middlewares`` abrange as restrições de segurança e a validação de acesso, alinhando-se com as práticas REST para garantir que cada usuário final tenha acesso seguro e autenticado aos dados, assegurando a integridade e confidencialidade dos registros.

Na camada de dados, a pasta ``models`` define as entidades essenciais para a manipulação de dados no banco, em consonância com as restrições de integridade e consistência de dados da arquitetura REST. A pasta ``routers`` organiza as rotas da API, estruturadas conforme o estilo RESTful, facilitando operações de CRUD (Create, Read, Update, Delete) de forma intuitiva e segura. Esse design visa garantir que o sistema mantenha uma estrutura robusta e escalável, com foco em segurança e conformidade com o modelo REST, que inclui práticas de autenticação e autorização para a proteção dos dados sensíveis das organizações sociais.

A integração com a nuvem, por meio da AWS, possibilita uma escala distribuída e resiliência para o sistema, com suporte a grandes volumes de dados e cargas variáveis,

aproveitando a capacidade de armazenamento de dados em servidores distintos de forma equilibrada e eficiente. A elasticidade na AWS permite que o sistema se ajuste automaticamente à demanda, assegurando a leitura e escrita de dados de forma paralela e não bloqueante. Com isso, a API é capaz de atender às operações de alta demanda de forma otimizada, alinhando-se às práticas RESTful, que recomendam uma estrutura de respostas eficientes e que mantenham a consistência e a integridade dos dados ao longo de toda a aplicação.

Além disso, o projeto contém arquivos de configuração fora da pasta `src` que otimizam o gerenciamento e a segurança do sistema. O uso de variáveis de ambiente em `env.txt` permite configurar credenciais sensíveis de forma segura, enquanto o `nodemon.json` facilita o desenvolvimento contínuo, reiniciando automaticamente o servidor a cada alteração no código. Os arquivos `package-lock.json` e `package.json` mantêm as dependências do projeto, enquanto `tsconfig.json` define a configuração TypeScript, convertendo o código para execução no Node.js. O arquivo principal `server.ts` realiza a inicialização da aplicação, assegurando a arquitetura RESTful para que a API esteja pronta para operar de forma segura e escalável.

Essas práticas, alinhadas aos princípios REST, garantem que o backend do sistema seja projetado para evoluir conforme as demandas de crescimento das organizações sociais. A estrutura modular, aliada à filosofia de design REST, oferece um sistema altamente seguro, eficiente e preparado para atender as operações distribuídas de forma escalável e consistente, conforme os requisitos modernos de uma aplicação web profissional.

### **3.2.2 FRONT-END**

O desenvolvimento do front-end do sistema foi realizado com Angular, um framework de código aberto desenvolvido pelo Google para a criação de aplicativos dinâmicos e interativos da web. Ele se baseia em TypeScript, uma linguagem superset do JavaScript, e utiliza uma abordagem orientada a componentes para a construção de interfaces. O Angular oferece uma estrutura sólida para desenvolvimento, incluindo suporte a gerenciamento de estado, roteamento e validação de formulários.

A aplicação Angular funciona como uma interface que permite ao usuário final interagir diretamente com o sistema, facilitando o acesso às funcionalidades e dados da plataforma. A escolha do Angular, com sua arquitetura baseada em componentes, possibilitou a criação de uma interface dinâmica e adaptada às necessidades da organização social,

respeitando os princípios de design responsivo e garantindo a acessibilidade em diferentes dispositivos, como computadores e smartphones.

O código front-end conta com uma organização modular, onde o diretório ``app`` concentra o núcleo da aplicação e suas rotas principais. A estrutura de arquivos inclui os elementos principais da configuração Angular, como o ``app-routing.module.ts`` que organiza as rotas para as diversas telas da aplicação. Dentro da pasta ``componentes``, foram desenvolvidos módulos específicos para cada funcionalidade do sistema, incluindo as telas de navegação (``tela1`` a ``tela8``), formulários de entrada (``primary-input``), recuperação de senha (``forget-pass``) e outros componentes de interface, todos com arquivos separados para estilos (``css``), estrutura HTML (``html``) e lógica (``ts``). Cada um desses módulos é encapsulado e independente, o que facilita a manutenção e expansão do sistema, promovendo uma arquitetura limpa e organizada.

A aplicação interage com o backend através de uma série de serviços implementados em arquivos ``service.ts`` dentro de cada componente. Esses serviços são responsáveis por enviar requisições HTTP para a API RESTful do backend, desenvolvido em Node.js e hospedado na AWS. A comunicação entre o front-end e o back-end segue o modelo de arquitetura REST, onde endpoints específicos no backend são acessados para operações de CRUD (Create, Read, Update e Delete) de dados. Esse mecanismo garante que as informações armazenadas no sistema — como dados de usuários, atividades e relatórios gerenciais — possam ser acessadas, manipuladas e exibidas no front-end de maneira eficiente e segura.

Para reforçar a experiência do usuário (UX) e a interface do usuário (UI), o front-end incorpora práticas de design responsivo. Dessa maneira, os visuais auxiliam os usuários a monitorar e gerenciar dados críticos da organização social, permitindo decisões informadas e baseadas em dados atualizados.

Além disso, o sistema Angular foi configurado para facilitar o desenvolvimento colaborativo e o controle de versões, utilizando arquivos como ``editorconfig`` para padronizar a formatação do código, ``gitignore`` para gerenciamento de versões, ``tsconfig.json`` para configurar o compilador TypeScript, e ``angular.json`` para a configuração do projeto. Para otimizar a performance e garantir a consistência visual, arquivos como ``styles.css`` centralizam os estilos globais, enquanto o diretório ``assets/imagens`` armazena os elementos visuais necessários para a interface.

Por fim, a estrutura do projeto foi configurada para atender aos requisitos de segurança e acessibilidade, com práticas de teste automatizado implementadas em arquivos ``spec.ts``

para garantir a estabilidade dos componentes e sua conformidade com as melhores práticas de usabilidade e acessibilidade. Isso assegura que o sistema mantenha sua integridade e alta performance, além de ser inclusivo para todas as pessoas que vão utilizá-lo. Em resumo, o front-end feito para a organização social João de Barro é uma interface robusta, projetada para ser intuitiva, escalável e eficiente, oferecendo suporte completo ao backend e promovendo uma experiência de uso otimizada para os usuários finais.

```
private meses: string[] = [
  'Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio', 'Junho',
  'Julho', 'Agosto', 'Setembro', 'Outubro', 'Novembro', 'Dezembro'
];

private mesAnterior!: string;

constructor(private http: HttpClient) {}

ngOnInit(): void {
  this.mes = 'Novembro';
  this.ano = 2024;
  this.carregarCalendario();
  this.buscarAtividadesAprovadas();
}

carregarCalendario() {
  const numeroMes = this.getNumeroMes(this.mes);
  const diasNoMes = new Date(this.ano, numeroMes + 1, 0).getDate();
  this.dias = Array.from({ length: diasNoMes }, (_, i) => {
    const dia = i + 1;
    return { dia, comAtividade: false };
  });
}
```

Codificação do calendário

## **3.3 COMPUTAÇÃO EM NUVEM**

### **3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING**

O uso da computação em nuvem no sistema de gestão do Núcleo Social João de Barro viabiliza vários objetivos estratégicos da organização. De início, os serviços de nuvem aumentam em geral a eficiência operacional, fornecendo uma infraestrutura flexível e automação de serviços. Em comparação com sistemas comuns e tradicionais baseados em servidores locais limitados à capacidade ou a tempo e custo, os servidores em nuvem permitem ao usuário ajustar a capacidade de processamento e armazenamento conforme necessário. Para o Núcleo Social João de Barro e o projeto Avante Corpo e Mente, isso é particularmente importante, pois a organização opera em grande número de atividades e beneficiários, exigindo um sistema rápidos e operações funcionais.

De acordo com Y Wei e MB Blake (2010):

Cinco décadas atrás, em 1961, o pioneiro da computação John McCarthy previu que 'a computação pode um dia ser organizada como uma utilidade pública.' A computação em nuvem é essa realização, pois o paradigma facilita a entrega de computação sob demanda, assim como outras utilidades públicas, como eletricidade e gás.

Em relação à economia, a computação em nuvem oferece um modelo econômico de 'pay-as-you-go' que envolve o pagamento apenas dos recursos consumidos pela organização. Isso elimina custos de hardware, manutenção de servidores físicos e economiza custos com energia e espaço físico. Como a organização Núcleo Social João de Barro trabalha com recursos limitados, a economia se traduz em fundos que podem ser liberados para outras áreas, como a prestação direta de serviços aos beneficiários.

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

A computação em nuvem possibilita várias oportunidades concretas para uma empresa que busca operações otimizadas e o desenvolvimento de soluções tecnológicas escaláveis e seguras. Em relação ao Núcleo Social João de Barro, esta tecnologia pode ser criticamente aplicada para melhorar o sistema de gerenciamento das atividades noturnas, uma vez que permite a integração rápida e eficiente de dados sobre os participantes. O primeiro caso de uso é o armazenamento centralizado e seguro das informações com um rápido acesso a elas

independentemente da localização física dos usuários autorizados. Isso é essencial para uma organização noturna com dados altamente confidenciais, como as folhas de atendimento dos acolhidos.

Outro fator que a computação em nuvem garante é o da escalabilidade. Com os avanços das atividades desenvolvidas pelo Núcleo Social João de Barro, é perceptível que será necessário, ao longo do tempo, um sistema diferente para lidar com um número elevado de beneficiários, além de operações mais complexas. O ambiente em nuvem comunica escalabilidade do mesmo, podendo ampliar-se à capacidade de data center e processamento sem a necessidade de infraestrutura e capital adicional.

De acordo com AL-ROUSAN (2015):

O maior impacto potencial da computação em nuvem no Desenvolvimento Global de Software (GSD) pode ser a eliminação de custos de desenvolvimento e economia de recursos. Isso porque a computação em nuvem é principalmente baseada na virtualização de recursos de hardware e software. Devido às técnicas de virtualização, os recursos físicos podem ser conectados dinamicamente a diferentes aplicações que rodam em diversos sistemas operacionais. Como resultado, os recursos físicos podem ser compartilhados entre todos os usuários e há uma gestão eficiente dos recursos, o que pode proporcionar um uso mais elevado dos recursos e escalabilidade sob demanda.

Outro benefício é a agilidade no desenvolvimento e na implementação das soluções. A nuvem possibilita que a equipe de desenvolvedores implemente novas funcionalidades, sem a interrupção dos serviços que o Núcleo Social João de Barro. Esse é um ponto importante para garantir flexibilidade e adequação às demandas da organização.

Por fim, a computação em nuvem contribui com a redução de custos, elimina a necessidade de manutenção, evita a atualização de servidores locais e reduz o custo de energia e equipamentos locais. Ademais, o uso do modelo de pagamento conforme o uso (“pay-as-you-go”) assegura que a organização pague apenas pelos recursos que irão utilizar. Desta maneira, a computação em nuvem se alinha aos objetivos do projeto, concedendo soluções que aumentam a eficiência e impacto do Núcleo Social João de Barro na sociedade.

### **3.3.3 VANTAGENS DA CLOUD COMPUTING**

A computação em nuvem oferece um número de vantagens que atendem diretamente às necessidades e à realidade atual do Núcleo Social João de Barro, levando a maior eficiência e eficácia ao gerenciar seus recursos tecnológicos. Em particular, a elasticidade dos recursos é um fator crítico, já que “se a demanda aumentar, os recursos podem ser ajustados automaticamente”. Isso é vital para a organização, pois suas atividades dependem do número

de pessoas que participam e residentes em diferentes momentos. Com a ajuda dos recursos da nuvem, o sistema ajustará o volume de processamento e armazenamento necessários, permitindo que o Núcleo use o sistema necessário sem limitar a capacidade operacional seja muito ou pouco necessário.

De acordo com Apostu e outros autores, a computação em nuvem oferece diversos benefícios para organizações e empresas de diferentes portes. Esse modelo é considerado uma das melhores opções quando se trata de custo de uso, manutenção e manutenção/atualização de sistemas. Enquanto softwares tradicionais de desktop exigem altos investimentos, a *cloud computing* apresenta taxas menores de custo, reduzindo os gastos com as áreas de TI no ramo empresarial. Além disso, existem opções flexíveis de pagamento, como o único ou conforme o uso, o que torna o modelo de computação em nuvem ainda mais atrativo e acessível. Outro benefício interessante é a grande capacidade de armazenamento de dados em nuvem, o que facilita backups e recuperação de informações comparado com modelos tradicionais. A integração de software também é realizada automaticamente, o que elimina a necessidade de ajustes manuais. A computação em nuvem permite o acesso remoto a informação de qualquer local com internet, superando barreiras geográficas e de fuso horário. Além disso, a implantação rápida desse sistema possibilita que empresas se tornem funcionais em um tempo muito curto em comparação com métodos tradicionais, de acordo com as necessidades tecnológicas específicas de cada negócio. A escalabilidade dos serviços também é um diferencial permitindo que as empresas ajustem seus recursos conforme demanda. Finalmente, a nuvem viabiliza novas classes de aplicações e entrega de serviços interativos (APOSTU et al., 2013).

Outro aspecto relevante é o modelo de pagamento pay-as-you-go, que tem um impacto financeiro incrível para empresas e organizações. O Núcleo Social João de Barro pagará apenas pelos recursos utilizados, evitando possíveis gastos com infraestrutura ociosa. Pode, assim, gerir o orçamento de forma mais eficaz, o que é essencial, uma vez que é uma organização social e tem financiamento limitado e quer garantir que o valor do dinheiro dos contribuintes esteja otimizado em termos de gastos com tecnologia. Esta abordagem é muito mais econômica na prática, uma vez que não terão de adquirir servidores empresariais, mantê-los e executá-los, já que a energia, o ar condicionado e o suporte físicos não serão necessários.

Outra vantagem importante que a computação em nuvem oferece é a alta disponibilidade, o que é crucial para um negócio que precisa de acesso constante ao seu banco de dados e ao sistema de gestão. Na Nuvem, as soluções estão sempre disponíveis, mesmo em

caso de falha local, interrupção ou falta de energia. Além disso, a computação em nuvem fornece segurança aprimorada; é equipada com a mais recente criptografia, backups automáticos e monitoramento completo para proteger o Núcleo Social João de Barro dos dados dos participantes de serem roubados. Isso é extremamente útil porque o Núcleo Social João de Barro lida com informações confidenciais, como informações sensíveis dos participantes.

No aspecto econômico, a migração para a nuvem reduz os custos, uma vez que elimina a necessidade de aquisição de hardware e de licenças de software complexas. Além disso, estudos econômicos sobre a implementação de cloud computing demonstram uma otimização de recursos ao longo do tempo, com redução de despesas de manutenção e suporte. Para o Núcleo, isso se traduz em uma economia significativa de capital, permitindo que os fundos economizados sejam direcionados para outras áreas críticas da organização, como o atendimento direto às pessoas em situação de vulnerabilidade.

Portanto, a adoção da computação em nuvem não apenas otimiza a gestão tecnológica do Núcleo Social João de Barro, como também proporciona uma economia substancial, oferecendo flexibilidade, segurança e eficiência em um contexto de recursos limitados.

### **3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING**

O sistema criado para o projeto do Núcleo Social João de Barro será implementado usando uma combinação de modelos de *cloud computing*, especificamente *Software as a Service* (SaaS), *Platform as a Service* (PaaS) e *Infrastructure as a Service* (IaaS). Cada um desses modelos terá seu papel na otimização das operações da organização social, fornecendo soluções tecnológicas de nuvem que garantem escalabilidade, flexibilidade e eficiência no gerenciamento das atividades oferecidas pelo Núcleo Social João de Barro.

O modelo SaaS será empregado para a disponibilização de ferramentas de gestão dos beneficiários, oferecendo uma interface intuitiva para os funcionários e administração utilizarem o sistema. Isso possibilita que a equipe de profissionais tenha acesso imediato ao sistema, sem a necessidade de instalação de software local, permitindo a centralização de dados e o uso de funcionalidades em qualquer dispositivo com conexão à internet. Além disso, o SaaS garante que as manutenções e atualizações necessárias sejam feitas automaticamente, sem interrupção do sistema.

Já o PaaS é comumente usado no desenvolvimento e na implantação de funcionalidades personalizadas de acordo com as demandas do projeto. Haverá a

possibilidade futura da plataforma fornecer um ambiente de desenvolvimento simples, permitindo que a equipe de desenvolvedores trabalhem em novas features do projeto com fácil integração, como o controle de presença de beneficiários.

O modelo IaaS será fundamental para fornecer a infraestrutura necessária ao armazenamento de dados e ao processamento computacional. Com o IaaS, o Núcleo poderá provisionar e gerenciar servidores virtuais conforme a necessidade, ajustando o poder de processamento e a capacidade de armazenamento em tempo real, sem a necessidade de adquirir ou manter hardware físico. Isso oferece maior flexibilidade, permitindo que a infraestrutura cresça conforme o aumento das demandas da organização.

O modelo IaaS será fundamental, pois entrega uma infraestrutura necessária ao armazenamento de dados e ao processamento computacional com funções de cloud. Com o IaaS, o Núcleo Social João de Barro e o projeto Avante poderão gerenciar e utilizar servidores virtuais conforme a necessidade, ajustando o processamento e a capacidade de armazenamento em tempo real, sem a necessidade de adquirir ou manter hardware físico. Isso oferece maior flexibilidade e com isso, a infraestrutura irá crescer conforme as demandas da organização.

Uma função essencial para a implementação correta e funcional do sistema em nuvem será a aplicação de um balanceamento de carga, função que desempenha um papel importante na manutenção do desempenho, disponibilidade e eficiência das aplicações utilizadas no sistema criado para o projeto em questão. O balanceador de carga distribui o tráfego de rede e as solicitações dos usuários de maneira equilibrada entre os servidores virtuais, evitando sobrecargas e garantindo que os serviços permaneçam operacionais mesmo em momentos de alta demanda. Esse mecanismo é importante para garantir a confiabilidade do sistema, especialmente considerando a necessidade de acesso constante a dados críticos dos beneficiários do Núcleo Social João de Barro e do Avante.

Quando se trata de arquitetura de cloud computing, os principais pontos que serão relevantes para o projeto incluem servidores virtuais (*VMs*), redes virtuais, *storage* em nuvem, banco de dados escalável e serviços de monitoramento. Esses elementos trabalham integradamente para garantir a eficiência e a segurança do sistema. As *VMs* serão responsáveis por hospedar a aplicação e processar as requisições dos usuários. As redes virtuais irão conectar os servidores, promovendo uma comunicação segura e rápida entre diferentes componentes do sistema. O *storage* em nuvem proporcionará o armazenamento seguro e escalável dos dados, enquanto o banco de dados garantirá que o sistema possa crescer conforme o aumento no volume de informações.

É importante também que haja monitoramento contínuo da infraestrutura, pois é outro componente essencial que será utilizado para detectar e resolver possíveis falhas antes que afetem a operação real do Núcleo. A interação entre os elementos destacados trabalha para garantir que a infraestrutura de nuvem funcione integradamente, atendendo as demandas do sistema e segurança do projeto, e, ao mesmo tempo, garantindo a eficiência do sistema.

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

A migração para um sistema em nuvem é uma decisão estratégica que busca otimizar e economizar a infraestrutura de TI, reduzir custos operacionais e garantir escalabilidade para atender as demandas de serviços. No contexto do Núcleo Social João de Barro, essa migração é importante para aguentar o sistema de gerenciamento de atividades e beneficiários nas atividades. A escolha de um provedor de nuvem correto impactará diretamente no sucesso do projeto, sendo importante considerar e escolher uma plataforma que atenda corretamente às necessidades e expectativas da organização. Com o objetivo de atender as demandas do Núcleo Social João de Barro e o projeto Avante, a escolha foi a Amazon Web Services (AWS) como provedor de nuvem, em detrimento do Google Cloud.

De acordo com Mufti, Mittal e Gupta a Amazon Web Services (AWS) é uma das plataformas de computação em nuvem mais antigas e amplamente conhecidas, lançada em 2006. A AWS oferece uma variedade de serviços, incluindo armazenamento em nuvem, banco de dados, análises, redes, Internet das Coisas, computação móvel e serviços empresariais. Esses serviços permitem que as organizações cresçam rapidamente, reduzam custos e ampliem seus negócios de forma eficiente. Atualmente, a AWS opera em 63 zonas de disponibilidade ao redor do mundo. Já a Google Cloud Platform (GCP), lançada em 2011, também oferece uma ampla gama de serviços, como armazenamento, Big Data, inteligência artificial na nuvem, rede, segurança, ferramentas de desenvolvimento e gestão, além de Internet das Coisas. A GCP opera em 21 zonas de disponibilidade globalmente (MUFTI; MITTAL; GUPTA, 2021).

Para justificar a decisão, foi feito um levantamento e alinhamento com as necessidades da organização, levando em consideração os seguintes fatores:

- Carga de trabalho: O sistema do Núcleo deve lidar com picos de demanda em momentos críticos, como nos horários de chegada e saída de participantes, além de manter a estabilidade durante períodos de menor uso.

- Orçamento: Sendo uma organização sem fins lucrativos, o Núcleo dispõe de um orçamento limitado, demandando uma solução de custo-benefício otimizado.
- Experiência técnica: A equipe técnica responsável pelo desenvolvimento do sistema possui maior experiência e familiaridade com as tecnologias da AWS, o que facilita a implementação e o suporte contínuo da solução.
- Integração: O Núcleo já utiliza alguns serviços da AWS, o que simplifica a integração com o novo sistema de gerenciamento, reduzindo o tempo e os custos de implementação.
- Segurança: A organização lida com dados sensíveis dos participantes e acolhidos, como históricos psicopedagógicos e registros de atendimento. Isso exige altos padrões de segurança para garantir a proteção dessas informações.
- Conformidade: O Núcleo precisa aderir a normas de conformidade, como a Lei Geral de Proteção de Dados (LGPD), e seguir as melhores práticas para a proteção de dados pessoais.

Com base na avaliação dos serviços e requisitos do Núcleo, a AWS foi escolhida como o provedor de nuvem mais adequado para o projeto. As principais razões que sustentam essa escolha incluem:

- Integração simplificada: A presença de outros serviços AWS na organização facilita a integração com o novo sistema de gerenciamento, reduzindo a complexidade da implementação e os custos associados.
- Flexibilidade e amplitude de serviços: A AWS oferece uma vasta gama de serviços, incluindo Amazon EC2, S3, RDS e Lambda, que podem ser adaptados às necessidades específicas do projeto, permitindo maior personalização e eficiência no uso dos recursos.
- Suporte e comunidade ativa: A AWS conta com uma comunidade ampla e ativa, oferecendo suporte técnico contínuo e uma vasta gama de documentação, o que proporciona segurança e agilidade na resolução de problemas.

A segurança é um fator essencial para o projeto do Núcleo Social João de Barro, considerando a sensibilidade dos dados armazenados e processados. A AWS oferece um conjunto robusto de ferramentas de segurança, como criptografia de dados em trânsito e em repouso, controle de acesso granular via AWS Identity and Access Management (IAM), e

monitoramento constante através do AWS CloudWatch e AWS CloudTrail. Além disso, a AWS atende a diversas normas internacionais de conformidade, como ISO 27001, SOC 1, 2 e 3, além de estar em conformidade com a GDPR e a LGPD, o que assegura que os dados dos acolhidos e participantes sejam protegidos de acordo com as melhores práticas do setor.

Para a hospedagem da API e banco de dados do Núcleo Social João de Barro, comparamos os custos e serviços oferecidos por dois provedores de computação em nuvem: Amazon Web Services (AWS) e Google Cloud Platform (GCP). Abaixo, detalhamos as diferenças em termos de custo, serviços e funcionalidades.

## 1. Amazon Web Services (AWS)

Serviços Utilizados:

- Amazon EC2 (Elastic Compute Cloud): Utilizado para rodar a API.
- Amazon RDS (Relational Database Service): Usado para o banco de dados MySQL.
- Amazon S3 (Simple Storage Service): Armazenamento para hospedagem estática da aplicação Angular.

Resumo de Custos na AWS:

- Amazon EC2 (API):
  - Custo mensal estimado: \$74,75 USD
  - Custo anual estimado: \$897,00 USD
- Amazon RDS (Banco de Dados):
  - Custo mensal estimado: \$54,94 USD
  - Custo anual estimado: \$659,28 USD
- Amazon S3 (Hospedagem Angular e Armazenamento de Notificações):
  - Custo mensal estimado: \$1,11 USD
  - Custo anual estimado: \$13,32 USD
- Movimentação de dados para fora da AWS: Potencial custo adicional dependendo do tráfego de saída.
- Total de Custos na AWS:
  - Custo mensal total estimado: \$130,80 USD
  - Custo anual total estimado: \$1.569,60 USD

A AWS oferece uma infraestrutura escalável, com custo inicial zero para configuração. Contudo, o custo total depende do uso de instâncias EC2, RDS para MySQL e armazenamento S3, o que pode aumentar conforme a demanda de tráfego e armazenamento.

## 2. Google Cloud Platform (GCP)

Serviços Utilizados:

- Cloud Run: Plataforma para rodar a API.
- Cloud SQL (MySQL): Banco de dados MySQL com instância gerenciada.
- Região: Configurado para a região São Paulo (southamerica-east1), o que pode reduzir latência para usuários brasileiros.

Resumo de Custos na GCP:

- Cloud Run (API):
  - CPU (2 vCPUs) e Memória (5 GiB): R\$ 39,52 por mês, cobrindo até 10 milhões de requisições mensais com alocação de CPU durante o processamento das requisições.
- Cloud SQL MySQL (Banco de Dados):
  - Instância db-standard-2 (2 vCPUs, 7.5 GiB RAM), uso de 160 horas: R\$ 201,05
  - Armazenamento de 100 GiB (SSD): R\$ 34,64
  - Custo mensal total para banco de dados: R\$ 235,69
- Total de Custos na GCP:
  - Cloud Run + Cloud SQL: R\$ 275,21 por mês (com base em 160 horas para a instância do banco de dados)

A GCP oferece uma alternativa econômica, especialmente para cenários onde a carga de trabalho varia e não requer instâncias permanentes. O Cloud Run permite escalabilidade automática, otimizando custos para APIs que rodam apenas quando há requisições. No entanto, a carga de 160 horas na instância Cloud SQL é ideal para situações de uso intermitente, mas pode se tornar mais onerosa em cargas contínuas.

| Provedor | Serviço de API | Serviço de Banco de Dados | Custo Mensal Total Estimado | Custo Anual Total Estimado |
|----------|----------------|---------------------------|-----------------------------|----------------------------|
| AWS      | Amazon EC2     | Amazon RDS                | \$130,80 USD                | \$1.569,60 USD             |
| GCP      | Cloud Run      | Cloud SQL (MySQL)         | R\$ 275,21                  | N/A (custo mensal varia)   |

Tabela de comparação de custos entre Google Cloud e AWS

Após uma análise conjunta das necessidades específicas do Núcleo Social João de Barro, do projeto Avante, experiência dos desenvolvedores do projeto e dos serviços oferecidos pela AWS e pelo Google Cloud, concluiu-se que a AWS é a opção mais adequada para a implementação do sistema de gestão da organização.

Embora no momento atual do projeto iremos apenas utilizar o EC2 ou Cloud Run, contabilizamos os custos de S3, RDS e Cloud SQL para auxiliar na visibilidade de uma escalada futura do projeto.

A AWS oferece a escalabilidade, flexibilidade e segurança necessárias para o sucesso do projeto, além de atender aos requisitos orçamentários e operacionais da organização. A combinação de integração simplificada, suporte técnico e conformidade com normas de segurança faz da AWS a escolha mais eficiente e segura para o projeto.

### 3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

O projeto de sistema de gestão para o Núcleo Social João de Barro utiliza uma abordagem em conjunto da computação em nuvem, abrangendo os modelos de Software as a Service (SaaS), Platform as a Service (PaaS) e Infrastructure as a Service (IaaS). Cada um desses modelos desempenha um papel crucial na otimização das operações da organização, permitindo escalabilidade, flexibilidade e eficiência no gerenciamento das atividades.

O modelo SaaS será aplicado para a disponibilização de ferramentas de gestão dos beneficiários, oferecendo uma interface intuitiva para a equipe de funcionários e administração. Esse modelo elimina a necessidade de instalação local de software, permitindo que os usuários acessem o sistema de qualquer dispositivo com conexão à internet. A centralização de dados facilitada pelo SaaS garante que as informações sejam atualizadas automaticamente, assegurando que a equipe tenha acesso às funcionalidades mais recentes sem interrupções. Um exemplo prático é o uso de um sistema de gerenciamento de atividades que permite a visualização e atualização em tempo real dos dados dos beneficiários.

O PaaS, por sua vez, será empregado no desenvolvimento e na implantação de funcionalidades personalizadas para o projeto. Esse modelo possibilita que os desenvolvedores trabalhem em novas *features*, como o controle de presença dos beneficiários, em um ambiente de desenvolvimento simplificado e integrado. A capacidade de criar e testar novas funcionalidades sem a preocupação com a infraestrutura subjacente é um aspecto distintivo do PaaS, proporcionando agilidade e inovação contínua no sistema.

No que diz respeito ao IaaS, este modelo será fundamental para o fornecimento da infraestrutura necessária ao armazenamento de dados e ao processamento computacional. O Núcleo Social João de Barro poderá provisionar e gerenciar servidores virtuais conforme a demanda, ajustando a capacidade de processamento e armazenamento em tempo real. Isso permite uma grande flexibilidade, uma vez que a infraestrutura pode crescer conforme as necessidades da organização, sem a necessidade de investimentos em hardware físico. Por exemplo, durante períodos de alta demanda, como eventos especiais, a capacidade do sistema pode ser facilmente ampliada.

Quanto ao Balanceamento de Carga, ele é um componente crítico na arquitetura de sistemas em nuvem, desempenhando um papel vital na distribuição equitativa do tráfego de rede e na alocação eficiente de recursos computacionais. Através do balanceamento de carga, as solicitações dos usuários são distribuídas entre múltiplas instâncias de servidores, evitando sobrecargas em um único recurso e garantindo que o sistema permaneça responsivo e disponível.

Esse processo não apenas melhora a performance do sistema, mas também aumenta a resiliência. Por exemplo, se uma instância de servidor falhar, o balanceador de carga redireciona automaticamente o tráfego para outras instâncias disponíveis, minimizando o tempo de inatividade. Assim, o balanceamento de carga contribui para a alta disponibilidade e continuidade do serviço, características essenciais em um sistema que lida com um grande volume de atividades e beneficiários, como o projeto do Núcleo Social João de Barro.

Já a anatomia da computação em nuvem é composta por diversas camadas e componentes que trabalham em conjunto para oferecer serviços escaláveis e confiáveis. Entre os principais aspectos, destacam-se:

1. Front-end: Refere-se à interface de usuário e às aplicações que interagem diretamente com os usuários finais. No contexto do projeto, a interface para a gestão de beneficiários é um exemplo de front-end que proporciona acesso amigável ao sistema.
2. Back-end: Compreende os servidores, bancos de dados e serviços que suportam a aplicação. É a parte da infraestrutura onde os dados são armazenados, processados e

gerenciados. Para o Núcleo, isso inclui o Amazon EC2, RDS e S3, que formam a espinha dorsal do sistema.

3. Rede: A comunicação entre o front-end e o back-end ocorre através da rede, que permite a troca de informações e garante a conectividade necessária para o funcionamento do sistema.
4. Gerenciamento e segurança: Inclui ferramentas e práticas que asseguram a proteção dos dados e a integridade do sistema. A segurança em cloud computing é multifacetada, abrangendo criptografia, autenticação e controle de acesso, fundamentais para a proteção das informações dos beneficiários.

No contexto da implementação na AWS, os serviços escolhidos e seus custos estimados são os seguintes:

- Amazon EC2 (Elastic Compute Cloud): Este serviço fornece capacidade de computação escalável na nuvem, permitindo a configuração e execução de instâncias (máquinas virtuais) para hospedar a API e o front-end da aplicação em Angular. Para o projeto, optou-se pelas instâncias do tipo "t2.micro" ou "t3.micro", que são adequadas para uso inicial e elegíveis ao nível gratuito. Os custos estimados são:
  - Custo mensal: 74,75 USD
  - Custo anual (12 meses): 897,00 USD

| Resumo da estimativa |              |                          |
|----------------------|--------------|--------------------------|
| Custo inicial        | Custo mensal | Custo total de 12 months |
| 0,00 USD             | 74,75 USD    | 897,00 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      | 74,75 USD    |

Status: -

Descrição:

**Resumo da configuração:** Locação (Instâncias compartilhadas), Sistema operacional (Windows Server), Carga de trabalho (Consistent, Número de instâncias: 1), Instância do EC2 avançada (t2.large), Pricing strategy (Compute Savings Plans 3yr No Upfront), Habilitar monitoramento (desabilitada), DT Entrada: Not selected (0 TB por mês), DT Saída: Not selected (0 TB por mês), DT Intranregião: (0 TB por mês)

Estimativa de custo de Amazon EC2 no Cloud AWS

- Amazon RDS (Relational Database Service) com MySQL: O Amazon RDS facilita a configuração e operação de um banco de dados relacional na nuvem. Para este projeto, escolheu-se o MySQL, armazenando informações críticas de usuários e dados operacionais. A instância "db.t2.micro" é adequada para ambiente de desenvolvimento, com um armazenamento de 20 GB em SSD. Os custos são:
  - Custo mensal: 54,94 USD
  - Custo anual (12 meses): 659,28 USD

| Resumo da estimativa |              |                          |
|----------------------|--------------|--------------------------|
| Custo inicial        | Custo mensal | Custo total de 12 months |
| 0,00 USD             | 54,94 USD    | 659,28 USD               |
|                      |              | Inclui um custo inicial  |

#### Estimativa detalhada

| Nome   | Grupo                 | Região                    | Custo inicial | Custo mensal |
|--|-----------------------|---------------------------|---------------|--------------|
| Amazon RDS Custom for SQL Server   | Nenhum grupo aplicado | South America (Sao Paulo) | 0,00 USD      | 54,94 USD    |
| <b>Status:</b> -   |                       |                           |               |              |
| <b>Descrição:</b>  |                       |                           |               |              |
| <b>Resumo da configuração:</b> Armazenamento para cada instância do RDS Custom para SQL Server (SSD de uso geral (gp2)), Quantidade de armazenamento (1 TB), Tipo de instância (db.m5.large), Número de instâncias do RDS Custom para SQL Server (1), Utilização (somente sob demanda) (5 Hours/Day), Database Edition (Developer), Opção de implantação (Single-AZ), Licença (Customer-provided), Modelo de preço (OnDemand), Armazenamento de backup adicional (10 GB) |                       |                           |               |              |

#### Estimativa de custo de Amazon RDS no Cloud AWS

- Amazon S3 (Simple Storage Service): O Amazon S3 é uma solução de armazenamento de objetos que permite armazenar e recuperar grandes volumes de dados. No projeto, será utilizado para armazenar arquivos estáticos da aplicação. O custo estimado para o uso do S3, com um bucket configurado para até 5 GB de dados, é:
  - Custo mensal: 1,11 USD
  - Custo anual (12 meses): 13,32 USD

| Resumo da estimativa |              |                          |
|----------------------|--------------|--------------------------|
| Custo inicial        | Custo mensal | Custo total de 12 months |
| 0,00 USD             | 1,11 USD     | 13,32 USD                |
|                      |              | Inclui um custo inicial  |

#### Estimativa detalhada

| Nome   | Grupo                 | Região                    | Custo inicial | Custo mensal |
|--|-----------------------|---------------------------|---------------|--------------|
| Amazon Simple Storage Service (S3)   | Nenhum grupo aplicado | South America (Sao Paulo) | 0,00 USD      | 1,11 USD     |
| <b>Status:</b> -   |                       |                           |               |              |
| <b>Descrição:</b>  |                       |                           |               |              |
| <b>Resumo da configuração:</b> Armazenamento S3 Standard (5 GB por mês), Solicitações PUT, COPY, POST, LIST para S3 Standard (10000), GET, SELECT e todas as outras solicitações do S3 Standard (100000), Dados retornados pelo S3 Select (5 GB por mês), Dados verificados pelo S3 Select (5 GB por mês) DT Entrada: Internet (5 GB por mês), DT Saída: Internet (5 GB por mês) |                       |                           |               |              |

## Estimativa de custo de Amazon S3 no Cloud AWS

Esses custos refletem uma solução escalável e econômica, permitindo que o Núcleo Social João de Barro gerencie suas atividades de forma eficiente.

Entretanto, a computação em nuvem é sustentada por uma série de paradigmas tecnológicos que permitem sua eficácia e eficiência. Entre eles, destacam-se:

1. **Virtualização:** Essa tecnologia é a base da computação em nuvem, permitindo a criação de múltiplas instâncias virtuais em um único hardware físico. Isso maximiza a utilização dos recursos e proporciona uma infraestrutura flexível.
2. **Automação:** A automação de processos é essencial para gerenciar a infraestrutura em nuvem de forma eficiente. Ferramentas de automação permitem que tarefas repetitivas sejam realizadas sem intervenção manual, reduzindo o risco de erros e aumentando a agilidade na resposta a demandas do sistema.
3. **Escalabilidade:** Este paradigma é fundamental para a nuvem, permitindo que a capacidade de processamento e armazenamento seja ajustada de acordo com as necessidades do projeto. A escalabilidade pode ser vertical (aumento da capacidade de um único servidor) ou horizontal (adição de novos servidores), proporcionando flexibilidade conforme o crescimento do Núcleo.
4. **Multitenancy:** A arquitetura de multitenancy permite que múltiplos usuários ou organizações compartilhem a mesma infraestrutura, mantendo a segurança e a

privacidade dos dados. Esse modelo é especialmente vantajoso para organizações como o Núcleo Social João de Barro, que busca otimizar custos sem comprometer a segurança das informações dos beneficiários.

Portanto, o desenvolvimento em cloud computing para o Núcleo Social João de Barro não apenas se alinha às melhores práticas de tecnologia, mas também garante que a organização esteja equipada para atender às demandas de um ambiente em constante mudança, promovendo eficiência operacional e melhorando o impacto social de suas atividades.

Com a crescente necessidade de escalabilidade e flexibilidade nos sistemas de gerenciamento de dados, a computação em nuvem surge como uma solução eficaz para a implantação e manutenção de aplicações robustas. Este estudo define um cenário de implantação para o sistema em desenvolvimento, selecionando a Amazon Web Services (AWS) como provedor de nuvem, utilizando recursos de Infraestrutura como Serviço (IaaS) e Plataforma como Serviço (PaaS). A proposta inclui o uso de máquinas virtuais para hospedagem da aplicação, bancos de dados gerenciados, balanceadores de carga e serviços de segurança.

Optou-se pela Amazon Web Services (AWS) como provedor de nuvem para o sistema devido à sua ampla gama de serviços, alta escalabilidade e integração facilitada entre os diversos componentes da infraestrutura. A AWS oferece uma arquitetura flexível, com suporte global e serviços regionais estratégicos que permitem o ajuste dinâmico dos recursos, essenciais para garantir a continuidade e eficiência da aplicação em situações de alta demanda.

### **3.3.7 GOOGLE CLOUD ou AWS**

Com a crescente necessidade de escalabilidade e flexibilidade nos sistemas de gerenciamento de dados, a computação em nuvem surge como uma solução eficaz para a implantação e manutenção de aplicações robustas. Este estudo define um cenário de implantação para o sistema em desenvolvimento, selecionando a Amazon Web Services (AWS) como provedor de nuvem, utilizando recursos de Infraestrutura como Serviço (IaaS) e Plataforma como Serviço (PaaS). A proposta inclui o uso de máquinas virtuais para hospedagem da aplicação, bancos de dados gerenciados, balanceadores de carga e serviços de segurança.

Optou-se pela Amazon Web Services (AWS) como provedor de nuvem para o sistema devido à sua ampla gama de serviços, alta escalabilidade e integração facilitada entre os diversos componentes da infraestrutura. A AWS oferece uma arquitetura flexível, com suporte global e serviços regionais estratégicos que permitem o ajuste dinâmico dos recursos, essenciais para garantir a continuidade e eficiência da aplicação em situações de alta demanda.

A configuração da AWS para este projeto incluirá os seguintes serviços principais:

- **Amazon EC2 (Elastic Compute Cloud):** Este serviço de IaaS será utilizado para a hospedagem da API. O Amazon EC2 permite a criação de máquinas virtuais com sistemas operacionais configuráveis, como Windows ou Linux, garantindo flexibilidade de acordo com os requisitos de desenvolvimento e operação. O uso de instâncias baseadas em Linux, devido ao seu menor custo e à independência de aplicações específicas do Windows, reduz os custos operacionais mensais e anuais.
- **Amazon RDS (Relational Database Service):** Para o armazenamento e gerenciamento do banco de dados, o Amazon RDS com MySQL foi escolhido. Este serviço gerenciado garante backup automático, alta disponibilidade e permite escalabilidade vertical para atender ao aumento de dados sem impactar no desempenho da aplicação.
- **Amazon S3 (Simple Storage Service):** O armazenamento de arquivos estáticos, como arquivos de front-end da aplicação Angular e dados de usuários, será realizado no Amazon S3. Este serviço PaaS oferece alta durabilidade, disponibilidade global e permite escalabilidade automática com um custo acessível. Além disso, o Amazon S3 facilita o armazenamento de arquivos em uma estrutura hierárquica, suportando o crescimento do projeto.
- **Amazon CloudWatch:** Este serviço de monitoramento fornecerá visibilidade sobre o desempenho da aplicação e dos recursos em uso. O CloudWatch permite a criação de alarmes para detectar possíveis problemas e tomar ações automáticas, aumentando a confiabilidade do sistema.

A escolha do provedor AWS e dos serviços mencionados deve-se aos seguintes fatores:

- **Escalabilidade:** A AWS permite a alocação e o gerenciamento dinâmico de recursos, possibilitando o aumento da capacidade de servidores e bancos de dados conforme o crescimento da demanda. A integração entre o Amazon EC2, RDS e S3 facilita a expansão do sistema de forma escalável e eficiente.

- **Segurança:** A AWS fornece uma infraestrutura de segurança robusta, com criptografia de dados em trânsito e em repouso, além de uma arquitetura de rede segura com controles de acesso granulares. A implementação de grupos de segurança e VPC (Virtual Private Cloud) garante a proteção dos dados e a integridade dos usuários do sistema.
- **Custos:** A análise dos custos estimados para a hospedagem do sistema na AWS demonstra a viabilidade econômica da escolha. Abaixo, estão os custos detalhados para os serviços selecionados:
  - Amazon EC2:
    - Custo mensal: USD 74,75
    - Custo anual: USD 897,00
  - Amazon RDS:
    - Custo mensal: USD 54,94
    - Custo anual: USD 659,28
  - Amazon S3:
    - Custo mensal: USD 1,11 (para armazenamento de até 5 GB)
    - Custo anual: USD 13,32

Total Estimado:

- Custo mensal total: USD 130,80
- Custo anual total: USD 1.569,60

Esse orçamento foi baseado na escolha do sistema operacional Linux para as instâncias de EC2, devido ao seu menor custo em comparação com Windows. A escolha por Linux é justificada pela ausência de dependência de aplicativos específicos do sistema Windows, o que torna o Linux uma opção mais econômica e igualmente eficaz para o sistema em questão.

A região escolhida para hospedar a aplicação foi a de São Paulo (*southamerica-east1*). A escolha desta região visa reduzir a latência de acesso aos usuários locais, uma vez que o público-alvo do sistema está concentrado na América do Sul. Essa localização também proporciona menor custo em relação a regiões mais distantes, além de estar em conformidade com os requisitos de proteção de dados locais.

A escolha da AWS como provedor de nuvem para a implantação do sistema baseia-se em fatores de escalabilidade, segurança e custo-benefício. A arquitetura projetada, com uso de Amazon EC2, RDS e S3, proporciona uma solução confiável e econômica, permitindo que o sistema cresça de acordo com as necessidades da organização. A escolha pela região de São

Paulo e o uso do sistema operacional Linux contribuem para a redução de custos e a melhoria do desempenho local.

O objetivo deste plano é orientar a configuração e implantação de um sistema de gestão hospedado na AWS Academy, utilizando os créditos educacionais disponíveis. A implantação inclui uma API em Angular, banco de dados MySQL, e serviços complementares para garantir a funcionalidade e escalabilidade do sistema.

Para hospedar o sistema de gestão, utilizaremos três principais serviços da AWS: EC2, RDS e S3. Abaixo estão descritos os serviços escolhidos, suas funcionalidades no contexto do projeto, e a estimativa de custos de cada um:

#### Amazon EC2 (Elastic Compute Cloud):

- Descrição: O Amazon EC2 fornece capacidade de computação escalável na nuvem, permitindo que configuremos e executemos instâncias (máquinas virtuais) para hospedar nossa API e o front-end da aplicação em Angular. É uma escolha ideal para o processamento de requisições e respostas que nossa API precisa lidar, pois permite fácil configuração e gerenciamento de recursos de processamento.
- Configuração para o Projeto:
  - Tipo de instância: "t2.micro" ou "t3.micro", ideais para uso inicial e elegíveis ao nível gratuito em contas novas.
  - Região: Configuramos a instância na região mais próxima do público-alvo, como "us-east-1" (América do Norte) ou "southamerica-east1" (América do Sul), para otimizar a latência.
- Custo Estimado:
  - Custo inicial: 0,00 USD
  - Custo mensal: 74,75 USD
  - Custo anual (12 meses): 897,00 USD
- Amazon RDS (Relational Database Service) com MySQL:
  - Descrição: O Amazon RDS facilita a configuração, operação e escalabilidade de um banco de dados relacional na nuvem. Para este projeto, o banco de dados MySQL gerenciado pela RDS armazenará informações críticas de usuários e dados operacionais, com alta disponibilidade e segurança. O RDS permite o gerenciamento simplificado do banco de dados e integração direta com a instância EC2.
  - Configuração para o Projeto:

- Tipo de banco de dados: MySQL
- Tipo de instância: "db.t2.micro", adequado para ambiente de desenvolvimento e no nível gratuito.
- Armazenamento: 20 GB de armazenamento SSD.
- Custo Estimado:
  - Custo inicial: 0,00 USD
  - Custo mensal: 54,94 USD
  - Custo anual (12 meses): 659,28 USD
- Amazon S3 (Simple Storage Service):
  - Descrição: O Amazon S3 é uma solução de armazenamento de objetos que permite armazenar e recuperar grandes volumes de dados com alta durabilidade e disponibilidade. No nosso projeto, o S3 será utilizado para armazenar arquivos estáticos da aplicação, como imagens, documentos e outros arquivos de usuário, que podem ser acessados tanto pela API quanto pelos usuários finais.
  - Configuração para o Projeto:
    - Criar um bucket com política de acesso configurada para armazenar até 5 GB de dados, suficiente para o armazenamento básico do projeto.
  - Custo Estimado:
    - Custo inicial: 0,00 USD
    - Custo mensal: 1,11 USD
    - Custo anual (12 meses): 13,32 USD

Com esses serviços configurados, a arquitetura inicial do sistema estará preparada para hospedar a aplicação e gerenciar os dados com escalabilidade e segurança, dentro do orçamento disponível para o projeto.

Em conclusão, a escolha de uma plataforma de computação em nuvem para o projeto de gestão do Núcleo Social João de Barro foi baseada na análise das necessidades específicas de escalabilidade, segurança e custo. A AWS se mostrou uma opção sólida, com uma ampla gama de serviços maduros e bem integrados, como EC2, RDS e S3, além de uma estimativa de custo mensal viável para o orçamento disponível. Contudo, alternativas como Google Cloud e Azure também foram consideradas, e suas ofertas de serviços e preços devem ser monitoradas, já que o cenário de provedores de nuvem está em constante evolução.

### 3.4 ESTRUTURA DE DADOS

Em computação, normalmente utilizamos os dados de forma conjunta. A forma como estes dados serão agregados e organizados depende muito de como serão utilizados e processados, levando-se em consideração, por exemplo, a eficiência para buscas, o volume dos dados trabalhados, a complexidade da implementação e a forma como os dados se relacionam. Estas diversas formas de organização são as chamadas estruturas de dados.

Independentemente do problema computacional que os profissionais de dados enfrentam, eles terão que lidar com dados em algum momento para resolver esses problemas; caso os dados da base estejam desestruturados, é bem provável que eles demorem para solucionar esses impasses, e que muitos de seus processos internos, projetos e atividades operacionais sejam afetados, pensando nisso, no desenvolvimento do sistema de gestão para o Projeto João de Barro, foi essencial que fossem adotadas abordagens de estruturas de dados que garantissem a eficiência, a sustentabilidade e o desempenho da aplicação, considerando as tecnologias utilizadas; — TypeScript e Node.js no backend e Angular no frontend.

Dessa forma, os requisitos funcionais e não funcionais do sistema precisaram ser analisados e implementados cuidadosamente, não apenas quanto às operações de cada módulo, mas também quanto às estruturas de dados que suportem o armazenamento e o acesso de maneira otimizada.

Os requisitos funcionais, que descrevem as funções e características essenciais do sistema, demandam o uso de estruturas que permitam a eficiência nas operações de aprovação de atividades, gestão de alunos, registro de presença e visualização de calendário. A aprovação de atividades, por exemplo, requer a criação de uma estrutura que possibilite a alteração de status (como aprovado ou pendente) de forma rápida e segura, com filas para armazenamento temporário de atividades submetidas para revisão. Neste cenário, a organização dos dados em uma modelagem relacional, onde tabelas específicas para atividades e usuários (administradores e coordenação/professores) registram as associações entre os dados, viabiliza um fluxo de aprovação que favorece a rastreabilidade e auditoria das ações realizadas por cada perfil de usuário.

A funcionalidade de gestão de alunos, por sua vez, necessita de uma estrutura que permita adicionar e atualizar dados pessoais dos participantes, garantindo que as informações permaneçam precisas e acessíveis em uma estrutura relacional que utiliza identificadores únicos. Essa abordagem permite a manipulação de dados por meio de buscas e edições otimizadas, assegurando o uso de índices para acelerar consultas e modificar registros de

forma mais ágil e eficiente. Já para o registro de presença, uma estrutura relacional de armazenamento de dados em uma tabela de presença possibilita o monitoramento da assiduidade dos alunos com precisão, incluindo atributos que refletem a presença ou ausência em cada atividade específica, garantindo a consistência dos registros e a possibilidade de processamento em lote, para evitar sobrecarga do banco de dados em operações de atualização.

O requisito de visualização do calendário exige que as atividades sejam organizadas de maneira a otimizar a busca por datas, viabilizando buscas rápidas e eficientes e facilitando o uso do calendário no front-end desenvolvido em Angular. Assim, a tabela de atividades, devidamente indexada por data, permite uma integração eficiente com bibliotecas de visualização de calendário que exibem as atividades de forma clara, atendendo à necessidade de planejamento e organização.

Os requisitos não funcionais, que englobam aspectos como desempenho, conectividade e usabilidade, também guiam a definição de estruturas de dados e práticas de implementação que mantêm o sistema eficiente e acessível. Em termos de desempenho, garantir que o sistema opere bem mesmo em computadores com configurações limitadas exige que as operações sejam otimizadas para minimizar o uso de processamento. A criação de índices no banco de dados e o uso de caching para consultas frequentes asseguram que o sistema se mantenha rápido e responsivo.

Em relação à conectividade e à usabilidade, é fundamental que o sistema funcione de forma eficiente com uma conexão de internet de velocidade média, garantindo que todas as operações sejam executadas sem falhas ou latência, promovendo uma experiência de uso estável e contínua. A transferência de dados entre o backend e o frontend é otimizada por meio de JSON compactado, reduzindo o tempo de carregamento de informações em Angular, que utiliza objetos e arrays para a gestão temporária de dados. Esse método simplifica a manipulação dos dados em dispositivos de baixo desempenho, facilitando a navegação e interação dos usuários.

A escalabilidade e flexibilidade do sistema são essenciais para acomodar um crescimento no número de participantes e atividades. Para tal, o sistema deve ser projetado de modo a suportar uma expansão gradual dos dados. A partição de tabelas para grandes volumes de informações, como históricos de presença e atividades, é uma abordagem que distribui os dados de forma gerenciável, mantendo a eficiência do sistema. A arquitetura em APIs RESTful promove a modularidade e o particionamento do sistema em serviços menores

e mais específicos, facilitando a manutenção e permitindo que cada módulo seja aprimorado de acordo com as demandas de uso.

Além disso, a segurança de dados é tratada por meio de uma estrutura robusta que assegura a privacidade e a proteção das informações dos usuários. Dados sensíveis, como senhas, são armazenados com criptografia e hashing, utilizando bibliotecas de segurança como bcrypt para o gerenciamento seguro das credenciais de acesso. Um controle de acesso baseado em função (Role-Based Access Control) também é implementado, permitindo que usuários autenticados tenham permissões restritas de acordo com seu perfil, o que evita o acesso indevido a funcionalidades sensíveis.

A escolha de Angular no front-end e Node.js e TypeScript no back-end facilita o desenvolvimento modular e a reutilização de componentes essenciais. Em Angular, os dados temporários são armazenados em serviços compartilhados, evitando o uso excessivo de requisições ao servidor e otimizando o fluxo de informações. No backend, o modelo de execução assíncrono e não bloqueante do Node.js permite o gerenciamento eficiente de múltiplas requisições simultâneas, ideal para o registro em tempo real de presenças e atividades. A tipagem estática do TypeScript contribui para a segurança e previsibilidade do código, reduzindo a incidência de erros de tipo e facilitando a manutenção do sistema.

Em suma, a implementação de um sistema de gestão para o Projeto João de Barro exige um planejamento criterioso na definição das estruturas de dados, visando atender a todos os requisitos funcionais e não funcionais propostos. A adoção de boas práticas de desenvolvimento e de arquitetura em Angular, TypeScript e Node.js favorece a modularidade, o desempenho e a segurança, permitindo que o sistema seja expansível e adaptável a futuras necessidades e evoluções do projeto.

### **3.4.1 LEVANTAMENTO DE REQUISITOS**

Diante das características do sistema; estruturas dinâmicas como listas e filas são altamente adequadas para representar informações sobre atividades e presença dos alunos, que são frequentemente atualizadas e consultadas. Para listas de participantes ou atividades, uma estrutura de lista encadeada pode ser mais apropriada devido à sua flexibilidade na inserção e remoção de dados em tempo de execução, proporcionando simplicidade ao lidar com um volume variável de dados. No caso da gestão de permissões de acesso (para administradores, coordenadores e professores), a utilização de árvores ou grafos permite organizar a hierarquia de permissões de maneira eficiente, com um sistema de buscas e

verificações que permite a cada tipo de usuário acessar apenas os módulos permitidos de acordo com seu perfil.

### **3.4.2 VALIDAÇÃO DOS REQUISITOS**

Outro aspecto fundamental para o Projeto João de Barro é a otimização das estruturas de dados para suporte à gestão na nuvem, especificamente utilizando a infraestrutura da Amazon Web Services (AWS). Em um ambiente de computação em nuvem como o da AWS, é crucial que a arquitetura do sistema seja escalável e responsiva, especialmente para operações de alta demanda, como o gerenciamento de grandes volumes de dados distribuídos. A integração com a AWS envolve a escolha de estruturas de dados que suportem uma elasticidade eficiente, ou seja, que permitam ao sistema escalar automaticamente, distribuindo a carga de dados de forma equilibrada conforme o número de usuários e dados cresce.

A AWS oferece diversos serviços que facilitam essa escalabilidade, como o Amazon S3 para armazenamento de dados e o Amazon RDS (Relational Database Service) para gerenciamento de bancos de dados. Essas ferramentas possibilitam a distribuição e o balanceamento da carga de dados, além de oferecerem redundância e recuperação em caso de falhas. Estruturas como filas de dados são essenciais para garantir que a comunicação entre os componentes da aplicação seja eficiente e resiliente, mantendo a integridade dos dados.

O sistema deve ser capaz de responder a cargas variáveis, permitindo a leitura e escrita de dados de maneira paralela e não bloqueante, com a garantia da consistência e integridade dos dados por meio de uma estratégia de distribuição de carga balanceada. A arquitetura da AWS também permite distribuir o tráfego de rede entre múltiplos servidores, otimizando a performance e a disponibilidade.

Além disso, para garantir a integridade e a disponibilidade dos dados no sistema, será essencial adotar práticas que sustentem a confiabilidade das operações de CRUD (Create, Read, Update, Delete) em tempo real.

A arquitetura de dados para o Projeto João de Barro mantém-se em um alto nível de coesão e separação de responsabilidades, priorizando a simplicidade de manutenção e a clareza na organização dos dados. Utilizando um backend em Node.js com TypeScript e um frontend em Angular, a comunicação entre o cliente e o servidor deve ser otimizada para manipular as estruturas de dados de forma eficiente. As bibliotecas do Angular, como aquelas para manipulação de formulários e gerenciamento de estado, permitirão sincronizar dados rapidamente, sem sobrecarregar o backend. Isso também permite que o sistema aproveite a

escalabilidade proporcionada pela AWS, enquanto mantém a modularidade e flexibilidade para expansões futuras.

Ao integrar a infraestrutura da AWS com a arquitetura de dados e as melhores práticas de programação, como clean code e design modular, o sistema do Projeto João de Barro será capaz de atender aos requisitos de performance e escalabilidade, garantindo que ele possa se expandir e evoluir conforme as necessidades da organização cresçam.

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

A formação para a vida é um dos pilares do modelo do Projeto Pedagógico de Formação por Competências da UNIFEQB. A extensão universitária, por meio da qual o conhecimento deve, por definição, ser aplicável, é o eixo central do projeto mencionado. Assim, a diversidade cultural e a luta contra os estereótipos são questões-chaves da formação do estudante. Ao abordar o tema da diversidade cultural, o projeto busca mobilizar e sensibilizar o público para a valorização e respeito entre as culturas, bem como as diferenças culturais da sociedade moderna.

Com isso, o projeto propõe que o confronto aos preconceitos certamente conduzirá a uma coexistência justa e igualitária. Como resultado, a Formação para a Vida é capaz de preparar profissionais não apenas competentes, mas também conscientes de seu papel social e que possam contribuir para o desenvolvimento de uma sociedade igualitária e amplamente plural.

### **3.5.1 ENFRENTANDO ESTEREÓTIPOS**

- **Tópico 1: Estereótipo e convívio social**

O conceito de estereótipo é um fator que influencia diretamente a dinâmica da sociedade atual, já que é um tema pertinente e presente no dia a dia. Entretanto, os estereótipos são apenas construções sociais que resumem a complexidade das identidades individuais e coletivas em padrões superficiais - e muitas vezes, negativos. Desta maneira, os estereótipos são condutas pré-estabelecidas que impactam a convivência interpessoal no dia a dia, tanto em cenários acadêmicos quanto profissionais e as relações sociais.

Walter Lippmann, em sua obra "Public Opinion" de 1922, nos mostra o conceito de "estereótipo" nas ciências sociais, descrevendo-os como "imagens em nossas cabeças" que sintetizam a forma como as pessoas pensam sobre os grupos sociais. Segundo Lippmann, as pessoas dependem dessas imagens simples ao formar e expressar opiniões sobre os outros, o que gera mal-entendidos e conflitos sociais, uma vez que os estereótipos distorcem a realidade com preconceitos.

Em contexto universitário, os estereótipos podem, inclusive, moldar maneiras de atuar e de se comportar. No final das contas, a universidade é um espaço diverso, e seus ambientes

são locais onde estereótipos podem ser desafiados e reforçados. No entanto, a simples presença destes códigos sociais pode influenciar grupos homogêneos e excluir indivíduos que fujam dos padrões, impossibilitando a inclusão e aprendizado.

Estimular a convivência baseada em respeito e igualdade propõe a desconstrução de preconceitos estabelecidos, de modo a almejar a complexidade das identidades para lá das caixas e estereótipos. Assim, o reconhecimento e o enfrentamento contra os estereótipos são ferramentas fundamentais para a construção de um espaço social mais inclusivo e equitativo.

- **Tópico 2:** Estereótipo e representação

O tópico “Estereótipo e Representação” discute de que forma os estereótipos são transmitidos e reforçados por representações sociais e culturais e atingem a percepção pública e individual. O estereótipo é uma definição simplificada que descreve um indivíduo ou um grupo de acordo com características facilmente identificáveis, o que pode simplificar um para um grau insustentável e não reconhecer a diversidade e complexidade de identidades reais.

As representações são difundidas por meio de redes sociais, publicidade, filmes e literatura, entre outros, e afetam o modo como os outros percebem grupos específicos de seres humanos na vida real. A representação estereotipada, ao descrever grupos com ideias preconceituosas e falsas, serve para perpetuar e justificar estereótipos e tratá-los como verdadeiros. Por exemplo, as representações de determinados grupos em contextos negativos ou exagerados podem consolidar preconceitos e reforçar desigualdades, afetando a maneira como esses grupos são percebidos e tratados no cotidiano.

Essas visões não se desenvolvem sozinhas, elas formam os comportamentos individuais e as políticas institucionais caso a caso. Para este efeito, é necessário analisar a fundo as representações estereotipadas e oferecer mais variedade e realismo às representações públicas. Uma imagem mais equilibrada para representação pode facilitar a erradicação dos preconceitos. Assim, acredita-se que uma nova cultura seja essencial para uma maior equidade e respeito.

- **Tópico 3:** Troco likes: a idealização da vida na internet

O tema “Troco Likes: A Idealização da Vida na Internet” investiga até que ponto a busca por validação e pertencimento nas redes sociais tem influenciado a percepção e a representação da vida diária. A era digital, que foi caracterizada pela ascensão das redes sociais, criou um ambiente no qual a aparência e a popularidade online frequentemente

superam a autenticidade e a realidade. Portanto, a prática do Troco Likes, significando a confirmação obtida por meio de curtidas, compartilhamentos e comentários, está levando à idealização da vida das outras pessoas e criando uma realidade virtual frequentemente distorcida.

A idealização da vida na internet levou a um retrato frequentemente irreal da experiência de aceitação, promovendo um padrão de comparação que se traduz em problemas de autoconfiança, autoimagem e autoestima. Além do mais, essa dinâmica incentiva não apenas uma competição por destaque e reconhecimento, mas também uma cultura de estereótipos e expectativas irreais em relação ao sucesso, felicidade e estilo de vida. A exposição contínua a essas visões idealizadas levará a sentimentos de inadequação e descontentamento entre os usuários, que poderão medir sua realidade contra as versões encenadas e editadas dos outros. Como tal, a conscientização sobre essas representações digitais precisa ser promovida para encorajar uma abordagem mais realista e equilibrada ao uso das redes sociais.

- **Tópico 4: Convivendo com a diferença**

O tema "Convivendo com a Diferença" explora a importância da tolerância e da cooperação entre indivíduos de diferentes origens, gerações e experiências. Em um mundo caracterizado pela diversidade crescente, seja no contexto acadêmico, profissional ou social, a habilidade de interagir e respeitar as diferenças é crucial para a construção de um ambiente harmonioso e produtivo.

A convivência com a diferença requer uma abordagem que valorize as experiências individuais e os múltiplos pontos de vista que cada pessoa traz. Isso implica em reconhecer e aceitar as particularidades culturais, geracionais e sociais que compõem o tecido social. A integração de diferentes perspectivas pode enriquecer o ambiente, promover o aprendizado mútuo e fortalecer as relações interpessoais.

No contexto acadêmico e profissional, a diversidade pode ser uma fonte de inovação e criatividade, desde que haja um esforço consciente para superar preconceitos e estereótipos. A prática da empatia e do respeito é fundamental para que as interações sejam construtivas e inclusivas. O reconhecimento da importância de cada indivíduo e de suas contribuições únicas é essencial para a convivência harmoniosa e para a construção de uma sociedade mais justa e equitativa.

### 3.5.2 ESTUDANTES NA PRÁTICA

O podcast criado e desenvolvido pela equipe aborda a questão do estereótipo de pessoas que vivem no interior do Brasil, frequentemente chamadas de "caipiras". O nosso objetivo foi gerar uma reflexão sobre como os estereótipos são formados em nossa sociedade, especialmente pela mídia, como novelas e filmes. Durante o podcast, exploramos como as regiões do interior do Brasil são ricas culturalmente e como a generalização cultural simplifica a identidade das pessoas afetadas.

Ao longo do podcast, discutimos a origem dos estereótipos, seu impacto na vida dos indivíduos, e conceitos contemporâneos como o "agrobóy", que exemplificam a evolução dos estigmas associados ao interior. Buscamos desmistificar a ideia de que o interior é homogêneo e destacamos a ideia de uma cultura plural e rica em atividades econômicas e costumes.

O podcast foi desenvolvido de forma colaborativa, com cada membro da equipe contribuindo para um tópico específico, garantindo uma abordagem ampla e detalhada. O episódio completo está disponível no link abaixo para ser verificado e avaliado.

Link do podcast: <https://youtu.be/c7ROalYhVnE>

## 4. CONCLUSÃO

A elaboração do Projeto Integrador (PI) para o Núcleo Social João de Barro proporcionou uma experiência significativa na aplicação de conhecimentos acadêmicos em um contexto prático, culminando no desenvolvimento de um sistema de gestão robusto e eficiente. O projeto abordou diversas áreas, como Estruturas de Dados, Linguagem e Técnicas de Programação, Tópicos Avançados de Banco de Dados e Computação em Nuvem, cada uma contribuindo de forma crucial para a construção de uma solução que atenda às necessidades operacionais da ONG.

Um dos principais pontos abordados foi a modelagem do banco de dados, que foi realizada em duas etapas: o Projeto Lógico, onde se definiu a estrutura e os relacionamentos das entidades, e o Projeto Físico, que detalhou a implementação técnica e os scripts necessários. Essa abordagem estruturada garantiu a integridade e a eficiência do sistema, permitindo um gerenciamento eficaz da presença dos alunos e das atividades desenvolvidas pelo Núcleo.

Durante o desenvolvimento, enfrentamos diversas dificuldades, como a integração entre as diferentes tecnologias escolhidas e a adaptação do sistema às demandas específicas da ONG. A escolha de um ambiente de computação em nuvem para a hospedagem do sistema também apresentou desafios, exigindo uma análise cuidadosa das opções disponíveis, considerando custo, escalabilidade e segurança. No entanto, essas dificuldades foram superadas por meio de pesquisa, colaboração entre os membros da equipe e a aplicação de boas práticas de programação e desenvolvimento.

Além disso, a experiência adquirida durante a realização deste projeto foi fundamental para o crescimento profissional dos envolvidos. A aplicação prática dos conceitos teóricos ensinados nas disciplinas permitiu um entendimento mais profundo das tecnologias utilizadas, além de desenvolver habilidades em trabalho em equipe, resolução de problemas e gerenciamento de projetos.

Em suma, o Projeto Integrador para o Núcleo Social João de Barro não apenas atendeu às expectativas iniciais, mas também proporcionou um aprendizado significativo e uma contribuição valiosa para a comunidade atendida. Com a implementação deste sistema, espera-se que o Núcleo consiga gerenciar suas atividades de forma mais eficaz, beneficiando todos os alunos e facilitando o trabalho dos colaboradores envolvidos. A experiência adquirida ao longo deste processo certamente servirá como base sólida para futuros desafios e empreendimentos na área de tecnologia e gestão social.

## REFERÊNCIAS

ADAPALA, K. Raghu. **REST vs RESTful API: What's the difference?**

TheServerSide, 15 jul. 2024. Disponível em:

<https://www.theserverside.com/video/REST-vs-RESTful-API-Whats-the-difference#:~:text=T%20key%20difference%20between%20the,properly%20implements%20the%20REST%20philosophy>. Acesso em: 08 nov. 2024.

AL-ROUSAN, Thamer. **Cloud Computing for Global Software Development: Opportunities and Challenges**. International Journal of Cloud Applications and Computing, v. 5, n. 1, p. 19-30, jan./mar. 2015. Disponível em:  
[https://www.researchgate.net/profile/Thamer-Al-Rousan/publication/281377281\\_Cloud\\_Computing\\_for\\_Global\\_Software\\_Development\\_Opportunities\\_and\\_Challenges/links/5f33de5ea6fdccc43c24fc3/Cloud-Computing-for-Global-Software-Development-Opportunities-and-Challenges.pdf](https://www.researchgate.net/profile/Thamer-Al-Rousan/publication/281377281_Cloud_Computing_for_Global_Software_Development_Opportunities_and_Challenges/links/5f33de5ea6fdccc43c24fc3/Cloud-Computing-for-Global-Software-Development-Opportunities-and-Challenges.pdf). Acesso em: 12 set. 2024.

AMOASEI, Juliana. **Estrutura de dados: uma introdução**. Alura, 18 set. 2023.

Disponível em:

<https://www.alura.com.br/artigos/estruturas-de-dados-introducao?srsIid=AfmBOoqi7FyqMW SWNgKIaHdQcz9aVjihbfZ7XTaJOXBghvGdjQSSuE-Z#o-que-sao-estruturas-de-dados>.

Acesso em: 05 out. 2024.

APOSTU, Anca; PUICAN, Florina; ULARU, Geanina; SUCIU, George; TODORAN, Gyorgy. **Study on advantages and disadvantages of Cloud Computing - The advantages of Telemetry Applications in the Cloud**. Economic Informatics and Cybernetics

Department, Academy of Economic Studies, 2013. Disponível em:

[https://d1wqtxts1xzle7.cloudfront.net/53325989/cloudcomputing-pros\\_cons-libre.pdf?1496127398=&response-content-disposition=inline%3B+filename%3DStudy\\_on\\_advantages\\_and\\_disadvantages\\_of.pdf&Expires=1729638837&Signature=Q2g-2NERVtXnIDVYRWwePWctL1DLE8KrRtQ7vZ1MfgMr8~cqQRsiC9so2Xi~6P2zv1nzEJPsN4CX6V0I8uWjdSHiamxBM6Wt-FledVTO5bcYW5O7JbmZD4e8z~m8WhGxWMz2dIS~i6GTBySTOcNp8vRGjiKc2qlJzmdrvbFIUwsFLd4-vivDeA554rz~eVbx2yXbWHgGPsspyQwdkqRf~VVL7Py4o8XckUMKQymxezCR4dpPjwZ4rhdcTOVj-IEGU0OGIOvDktkgqX1ud7durgABrQVcygFIE~ZTZWmW9](https://d1wqtxts1xzle7.cloudfront.net/53325989/cloudcomputing-pros_cons-libre.pdf?1496127398=&response-content-disposition=inline%3B+filename%3DStudy_on_advantages_and_disadvantages_of.pdf&Expires=1729638837&Signature=Q2g-2NERVtXnIDVYRWwePWctL1DLE8KrRtQ7vZ1MfgMr8~cqQRsiC9so2Xi~6P2zv1nzEJPsN4CX6V0I8uWjdSHiamxBM6Wt-FledVTO5bcYW5O7JbmZD4e8z~m8WhGxWMz2dIS~i6GTBySTOcNp8vRGjiKc2qlJzmdrvbFIUwsFLd4-vivDeA554rz~eVbx2yXbWHgGPsspyQwdkqRf~VVL7Py4o8XckUMKQymxezCR4dpPjwZ4rhdcTOVj-IEGU0OGIOvDktkgqX1ud7durgABrQVcygFIE~ZTZWmW9)

[1TcLIYDFv21w7rfYIFEheRPUnGWu73knAVv9wyzpzUCDA\\_\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://www.alura.com.br/artigos/angular-js?srsltid=AfmBOopa1ie4ybx7Xke12u6a3NU55JKxbCLDqR7HWL1jXQrmNmCyxD-f). Acesso em: 16 set. 2024.

BATISTA, Nyanne. **Angular: o que é, para que serve e um Guia para iniciar no framework JavaScript** Alura, 16 abr. 2024. Disponível em: <https://www.alura.com.br/artigos/angular-js?srsltid=AfmBOopa1ie4ybx7Xke12u6a3NU55JKxbCLDqR7HWL1jXQrmNmCyxD-f>. Acesso em: 08 nov. 2024.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. Tradução de Marília Guimarães Pinheiro et al. São Paulo: Pearson Education do Brasil Ltda, 2005.

**Estruturas de dados: Quais são as mais relevantes?** Engineering do Brasil, 26 dez. 2023. Disponível em: <https://blog.engdb.com.br/estruturas-de-dados/>. Acesso em: 06 out. 2024.

LIPPMANN, Walter. **Public Opinion. With a new introduction by Michael Curtis**. New Brunswick: Transaction Publishers, 1998.

MUFTI, Tabish; MITTAL, Pooja; GUPTA, Bulbul. **A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services**. International Journal of Research in Engineering and Applied Sciences, 2021. Disponível em: <https://arxiv.org/pdf/1903.03219>. Acesso em: 16 set. 2024.

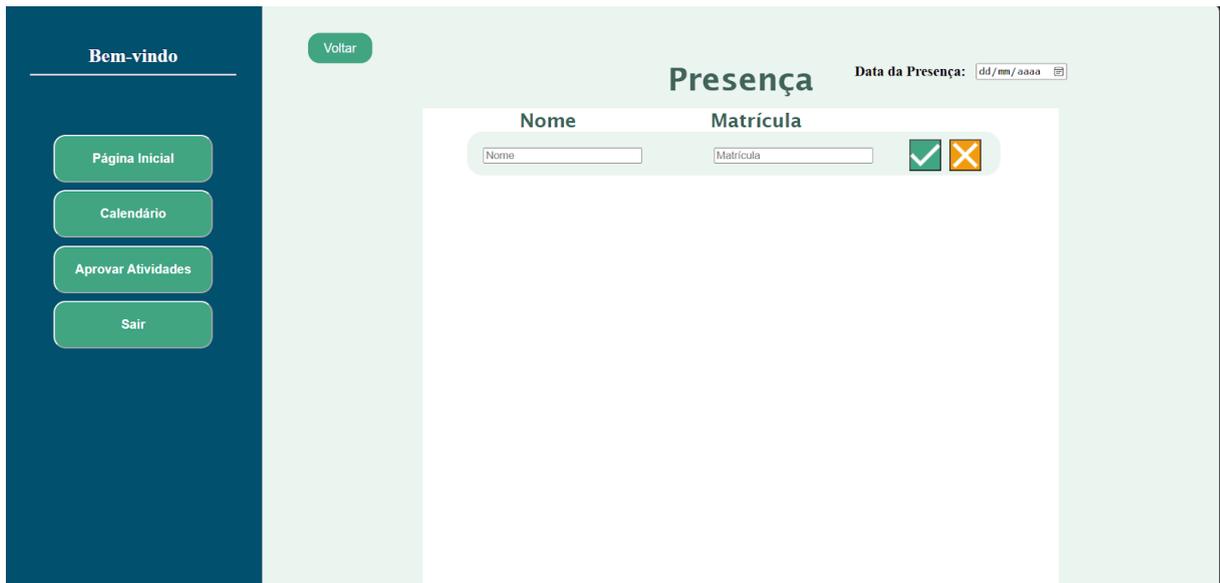
PATNI, Sanjay. **Pro RESTful APIs: Design, Build and Integrate with REST, JSON, XML and JAX-RS**. 1. ed. Apress, 2017.

PEREIRA, C. **Node.js: Aplicações web real-time com Node.js**. Casa do Código, 16 de abril de 2014.

WEI, Yi; BLAKE, M. Brian. **Service-Oriented Computing and Cloud Computing: Challenges and Opportunities**. IEEE Internet Computing, v. 14, n. 6, p. 72-75, 2010. Disponível em: [https://www.researchgate.net/profile/Yi-Wei-22/publication/224187646\\_Blake\\_MB\\_Service-](https://www.researchgate.net/profile/Yi-Wei-22/publication/224187646_Blake_MB_Service-)

[oriented\\_Computing\\_and\\_Cloud\\_Computing\\_Challenges\\_and\\_Opportunities\\_IEEE\\_Internet\\_Computing\\_146\\_72-75/links/0deec53909d346dc6a000000/Blake-MB-Service-oriented-Computing-and-Cloud-Computing-Challenges-and-Opportunities-IEEE-Internet-Computing-146-72-75.pdf](https://doi.org/10.1109/IC146.2024.106253909d346dc6a000000/Blake-MB-Service-oriented-Computing-and-Cloud-Computing-Challenges-and-Opportunities-IEEE-Internet-Computing-146-72-75.pdf). Acesso em 06 set. 2024.

# ANEXOS



Protótipo da tela de presença de alunos para as aulas.



Protótipo da tela de criação de atividades

```
DELIMITER $$
USE `bancopjb` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `alterar_status_aprovacao`(p_id INT)
BEGIN
    UPDATE atividades SET atividades.status = 1 WHERE idatividades = p_id;
END $$

DELIMITER ;
USE `bancopjb`;

DELIMITER $$
USE `bancopjb` $$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `bancopjb`.`after_insert_aprovacaoatividades`
AFTER INSERT ON `bancopjb`.`aprovacaoatividades`
FOR EACH ROW
BEGIN
CALL
alterar_status_aprovacao(NEW.atividades_idatividades);
END $$
```

Codificação de Stored Procedure e Trigger em MySQL

### Resumo da estimativa

| Custo inicial | Custo mensal | Custo total de 12 months |
|---------------|--------------|--------------------------|
| 0,00 USD      | 74,75 USD    | 897,00 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      | 74,75 USD    |

Status: -

Descrição:

**Resumo da configuração:** Locação (Instâncias compartilhadas), Sistema operacional (Windows Server), Carga de trabalho (Consistent, Número de instâncias: 1), Instância do EC2 avançada (t2.large), Pricing strategy (Compute Savings Plans 3yr No Upfront), Habilitar monitoramento (desabilitada), DT Entrada: Not selected (0 TB por mês), DT Saída: Not selected (0 TB por mês), DT Intra-região: (0 TB por mês)

Estimativa de Custo de Instância EC2 da AWS Cloud

The screenshot shows the Google Cloud Run configuration page. At the top, the service name is 'Cloud Run' with an information icon, and the cost is displayed as 'R\$ 39,52 / month'. Below this, the 'Cloud Run configuration' section is shown with a toggle for 'Advanced settings' which is turned on. The 'Region' is set to 'southamerica-east1 (São Paulo) - Tier 2'. Under the 'Resource Type' section, there are two buttons: 'Service' (which is selected) and 'Job'.

Estimativa de Custo de Cloud Run da Google Cloud