



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**

APAE

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
APAE

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^a. Mariângela Martimbianco Santos

Estudantes:

Arthur Fabris Peluque, RA 23000433

Luis Eduardo Leme Silva, RA 23000607

Matheus Farnetani Friedrich, RA 23000780

Renan Ruy Rodrigues Reis, RA 23000704

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2024

SUMÁRIO

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	4
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS	5
3.1.1 MODELO LÓGICO	5
3.1.2 MODELO FÍSICO	9
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	14
3.2.1 APPLICATION PROGRAMMING INTERFACE (API) - BACK-END.	15
3.2.2 FRONT-END	17
3.3 COMPUTAÇÃO EM NUVEM	18
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	19
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	19
3.3.3 VANTAGENS DA CLOUD COMPUTING	21
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	23
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	26
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	27
3.3.7 GOOGLE CLOUD ou AWS	29
3.4 ESTRUTURA DE DADOS	33
3.4.1 LEVANTAMENTO DE REQUISITOS	33
3.4.2 VALIDAÇÃO DOS REQUISITOS	36
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	40
3.5.1 ENFRENTANDO ESTEREÓTIPOS	40
3.5.2 ESTUDANTES NA PRÁTICA	42
4. CONCLUSÃO	45
REFERÊNCIAS	47

1. INTRODUÇÃO

Nosso projeto em conjunto com a APAE, consiste em desenvolver um sistema de gerenciamento de estoque voltado para alimentos. Nosso objetivo é conseguir prover um software simples e eficaz, capaz de facilitar o controle interno de logística da instituição, com o intuito de manter um controle da validade das doações alimentícias e facilitar os processos de estoque dentro da empresa.

2. DESCRIÇÃO DA EMPRESA

A empresa para qual o serviço está sendo prestado é a Associação de Pais e Amigos dos Excepcionais(APAE), CNPJ 44.832.426.0001/87, com limite territorial circunscrito ao município de São João da Boa Vista, Estado de São Paulo, é uma associação civil, beneficente, sendo sua sede na Vila Estrela, na cidade de São João da Boa Vista, Estado de São Paulo.

Suas atividades sociais beneficentes têm a missão de educar, prestar atendimento na área de saúde e tendo como missão, promover e articular ações de defesa de direitos das pessoas com deficiência e representar o movimento perante os organismos nacionais e internacionais para a melhoria da qualidade dos serviços prestados pelas Apaes, na perspectiva da inclusão social de seus usuários. Suas estruturas são preparadas e capacitadas com profissionais treinados para conseguir fornecer aos seus alunos, todos os cuidados e sanar as necessidades que precisam. Parcerias com empresas da cidade e regionais viabiliza a ação de doações de equipamentos, alimentos e entre outras formas de arrecadação

3. PROJETO INTEGRADO

3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

Segundo C.J Date, o banco de dados, por si só, pode ser considerado como o equivalente eletrônico de um armário de arquivamento; ou seja, ele é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados.

Para o sistema de controle de estoque da APAE, onde são necessárias operações como cadastro de produtos, movimentações de estoque, controle de validade e geração de relatórios, o banco de dados relacional parece ser a melhor solução, pois os dados envolvidos têm uma estrutura bem definida e a integridade dos dados é um requisito importante para garantir o correto gerenciamento do estoque.

Com base nesse contexto, a solução ideal seria utilizar um banco de dados relacional em nuvem, já que ele oferece alta disponibilidade, escalabilidade, e segurança, atendendo aos requisitos do sistema de forma eficiente.

Banco de Dados Relacional

Um banco de dados relacional é a escolha ideal para sistemas que lidam com dados estruturados, com relações bem definidas entre diferentes entidades. A estrutura de dados em um banco relacional é organizada em tabelas, que podem ser relacionadas entre si através de chaves primárias e chaves estrangeiras.

Com isso, a APAE poderá armazenar e manipular dados como produtos, movimentações de estoque, usuários, categorias de produtos, relatórios, entre outros, de maneira eficiente e organizada.

3.1.1 MODELO LÓGICO

O modelo lógico é uma representação de dados que descreve como os dados serão organizados em um sistema de gerenciamento de banco de dados (SGBD). Ele traduz o modelo conceitual em uma estrutura compatível com um tipo específico de banco de dados, definindo tabelas, colunas, tipos de dados, restrições, relacionamentos e regras de integridade, mas sem se preocupar com detalhes físicos de armazenamento.

Essa etapa é fundamental para garantir que o banco de dados seja eficiente, consistente e atenda aos requisitos do sistema.

Abaixo está algumas tabelas do projeto de armazenamento de estoque e suas descrições:

Tabela: estoque_entrada

- Descrição: Registra as movimentações de entrada de insumos no estoque, permitindo o rastreamento de novas aquisições ou recebimentos.

- **Atributos:**
 - id (INT, chave primária, auto incremento): Identificador único para cada entrada.
 - id_insumo (INT, chave estrangeira): Referência ao insumo que está entrando no estoque.
 - quantidade (DECIMAL): Quantidade do insumo recebida.
 - data_entrada (DATETIME): Data em que o insumo foi adicionado ao estoque.
 - id_usuario (INT, chave estrangeira): Referência ao usuário que registrou a entrada.
 - origem (VARCHAR): Informação sobre a origem do insumo (ex.: compra, doação).

- **Relacionamentos:**
 - Relaciona-se com insumos para identificar o item recebido.
 - Relaciona-se com usuarios para rastrear quem registrou a entrada.

2. Tabela: estoque_saida

- Descrição: Registra as movimentações de saída de insumos, como distribuições, vendas ou perdas.

- **Atributos:**
 - id (INT, chave primária, auto incremento): Identificador único para cada saída.
 - id_insumo (INT, chave estrangeira): Referência ao insumo que está saindo do estoque.
 - quantidade (DECIMAL): Quantidade do insumo retirada.
 - data_saida (DATETIME): Data em que o insumo foi retirado.
 - id_usuario (INT, chave estrangeira): Referência ao usuário que registrou a saída.
 - destino (VARCHAR): Informação sobre o destino do insumo (ex.: cliente, doação).

- **Relacionamentos:**
 - Relaciona-se com insumos para identificar o item retirado.
 - Relaciona-se com usuarios para rastrear quem registrou a saída.

3. Tabela: insumos

- Descrição: Armazena informações detalhadas sobre os insumos disponíveis no sistema, como nome, descrição e unidade de medida.

- **Atributos:**
 - id (INT, chave primária): Identificador único do insumo.
 - nome (VARCHAR): Nome do insumo.
 - descricao (TEXT): Detalhes adicionais ou especificações do insumo.
 - id_unidade_medida (INT, chave estrangeira): Referência à tabela unidades_medida, definindo como o insumo é quantificado (ex.: kg, litros).

- **Relacionamentos:**

- Relaciona-se com unidades_medida para especificar a unidade usada para medir o insumo.
- É referenciado por estoque_entrada e estoque_saida para registrar movimentações do insumo.

4. Tabela: doacoes

- Descrição: Registra doações recebidas, detalhando informações sobre o que foi doado, quem doou e como os itens foram tratados no estoque.

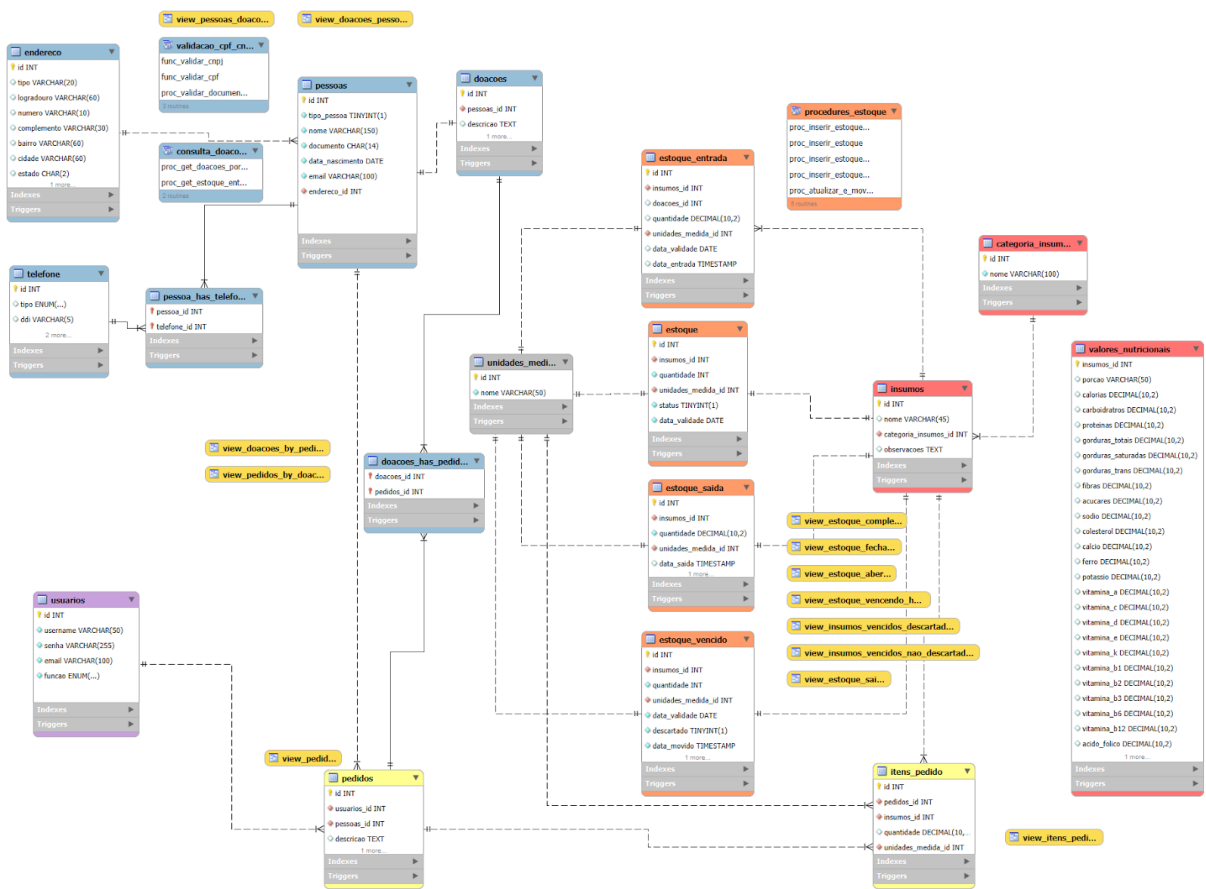
- **Atributos:**

- id (INT, chave primária, auto incremento): Identificador único da doação.
- id_insumo (INT, chave estrangeira): Referência ao insumo que foi doado.
- quantidade (DECIMAL): Quantidade do insumo doada.
- data_doacao (DATETIME): Data em que a doação foi registrada.
- id_usuario (INT, chave estrangeira): Referência ao usuário que registrou a doação.
- doador (VARCHAR): Nome ou identificação do doador.

- **Relacionamentos:**

- Relaciona-se com insumos para identificar o item doado.
- Relaciona-se com usuarios para rastrear quem registrou a doação.

Figura 1



Fonte: Autores

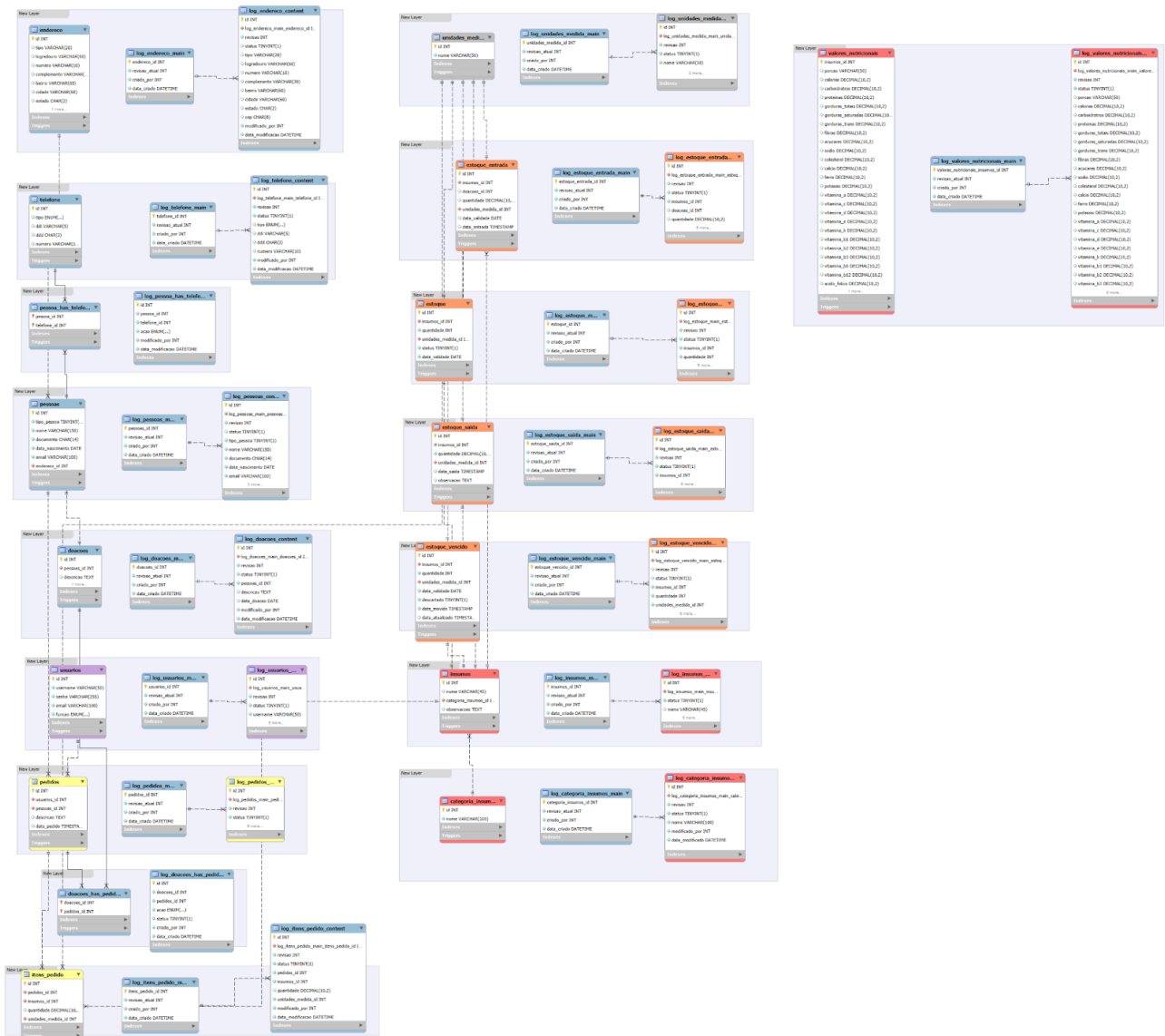
3.1.2 MODELO FÍSICO

O modelo físico de banco de dados é a etapa que define como os dados serão armazenados fisicamente em um sistema de gerenciamento de banco de dados (SGBD). Ele

traduz o modelo lógico em detalhes técnicos, como esquemas de tabelas, índices, organização de arquivos, particionamento de dados e estratégias de desempenho. Nesta etapa, são consideradas limitações específicas do SGBD escolhido, otimizando o armazenamento e o acesso aos dados.

Por exemplo, enquanto o modelo lógico define as tabelas e relacionamentos, o modelo físico especifica o uso de índices, métodos de particionamento e como os dados serão distribuídos em discos. O foco principal é garantir eficiência no armazenamento e na recuperação de informações.

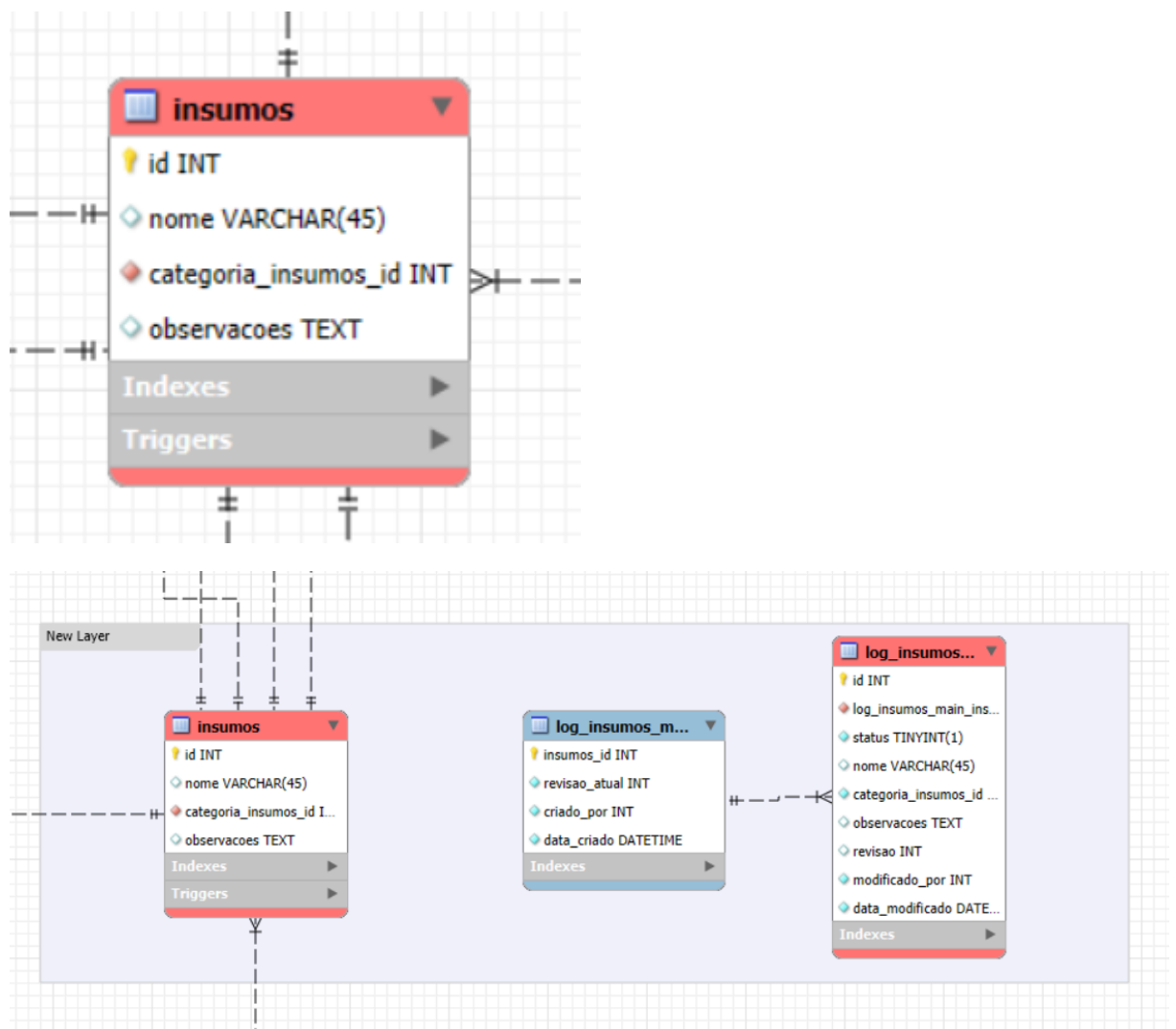
Figura 2: Modelo Físico



Fonte : Autores

Abaixo temos exemplos dos triggers nos insumos

Figura 3 e 4: Insumos



Fonte: Autores

1. Trigger: Prevenir Exclusão de Insumos Vinculados

Nome: prevenir_exclusao_insumos_vinculados

Descrição: Impede que um insumo seja excluído da tabela insumos se ele estiver associado a registros em outras tabelas, como estoque_entrada, estoque_saida ou doacoes.

Implementação:

```
DELIMITER //
CREATE TRIGGER prevenir_exclusao_insumos_vinculados
BEFORE DELETE ON insumos
FOR EACH ROW
BEGIN
    -- Verificar se o insumo está vinculado a entradas no estoque
    IF EXISTS (SELECT 1 FROM estoque_entrada WHERE id_insumo = OLD.id)
THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é possível excluir o insumo, pois ele está
associado a entradas no estoque.';
    END IF;

    -- Verificar se o insumo está vinculado a saídas no estoque
    IF EXISTS (SELECT 1 FROM estoque_saida WHERE id_insumo = OLD.id)
THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é possível excluir o insumo, pois ele está
associado a saídas no estoque.';
    END IF;

    -- Verificar se o insumo está vinculado a doações
    IF EXISTS (SELECT 1 FROM doacoes WHERE id_insumo = OLD.id) THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é possível excluir o insumo, pois ele está
associado a doações.';
    END IF;
END //
DELIMITER ;
```

2.Trigger: Atualizar Quantidade Total no Estoque

Nome: atualizar_quantidade_estoque_insumos

Descrição: Atualiza automaticamente a quantidade total de um insumo na tabela estoque quando o registro de um insumo é atualizado na tabela insumos (ex.: mudança de unidade de medida ou ajustes de dados que impactam a contagem do insumo).

Implementação:

```
DELIMITER //
CREATE TRIGGER atualizar_quantidade_estoque_insumos
AFTER UPDATE ON insumos
FOR EACH ROW
BEGIN
    -- Atualizar a quantidade total de estoque se necessário
    IF OLD.id_unidade_medida != NEW.id_unidade_medida THEN
        UPDATE estoque
            SET quantidade = quantidade * (SELECT fator_conversao FROM
unidades_medida WHERE id = NEW.id_unidade_medida)
            WHERE id_insumo = NEW.id;
    END IF;
END //
DELIMITER ;
```

Objetivo das Triggers:

Prevenir Exclusão de Insumos Vinculados: Garante que não haja inconsistências no sistema ao tentar excluir insumos que ainda possuem registros associados.

Atualizar Quantidade de Estoque: Ajusta automaticamente os dados de estoque caso alguma modificação nos insumos afete suas unidades de medida ou características relevantes.

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

Segundo o autor Sebesta, as linguagens de programação são ferramentas essenciais para a criação de software. Elas permitem que programadores descrevam algoritmos de forma compreensível para máquinas, transformando ideias abstratas em instruções executáveis. Cada linguagem possui uma sintaxe (estrutura de escrita) e uma semântica (significado), facilitando o desenvolvimento e a manutenção de sistemas. No início, a programação era feita em linguagens de baixo nível, como Assembly, exigindo conhecimento detalhado da arquitetura do computador. No entanto, com o avanço da tecnologia, surgiram linguagens de alto nível como Fortran, Lisp e COBOL, que trouxeram maior abstração e facilidade de uso. Essas linguagens permitiram o desenvolvimento de sistemas mais complexos e abrangentes, com aplicação em diversos paradigmas, como o estruturado e o orientado a objetos.

Essa evolução das linguagens reflete o esforço contínuo em tornar a programação mais acessível e eficiente, possibilitando que áreas diversas, como inteligência artificial, ciência de dados e sistemas embarcados, sejam exploradas.

A Programação Orientadas Objetos (POO) é baseada na ideia de "objetos", que são instâncias de classes. Cada objeto contém dados (atributos) e comportamentos (métodos). A POO organiza o código em entidades que representam objetos do mundo real ou abstrações, promovendo o encapsulamento, herança e polimorfismo.

Com a POO conseguimos fazer a divisão do código em classes, que podem ser vistas como módulos independentes. Isso facilita o isolamento de funcionalidades e a manutenção, e através da herança, é possível reutilizar código de classes base em classes derivadas. Além disso, com o uso de bibliotecas e frameworks orientados a objetos, a reutilização se torna mais fácil. Embora a POO não seja diretamente focada em desempenho, o encapsulamento e a abstração de detalhes internos permitem melhorias no design do sistema, o que pode otimizar processos complexos.

A programação funcional foca em funções como blocos de construção fundamentais do software. Essas funções são puras, ou seja, não têm efeitos colaterais e sempre retornam o mesmo valor para os mesmos argumentos. O conceito de imutabilidade e a ausência de estado mutável são centrais na programação funcional. Funções puras são isoladas do resto do sistema, o que torna fácil sua compreensão e manutenção. Cada função faz uma única tarefa de forma previsível. A composição de funções é uma técnica poderosa em programação

funcional. Funções menores podem ser reutilizadas para formar funções mais complexas. Além disso, a ausência de estado torna mais simples reutilizar funções em diferentes partes do código. Embora as linguagens funcionais possam sofrer inicialmente em termos de desempenho devido à imutabilidade, otimizações como avaliação preguiçosa e paralelização automática podem compensar. A programação funcional também facilita a programação concorrente, já que não há estados mutáveis para gerenciar.

A programação concorrente envolve a execução de múltiplas tarefas ao mesmo tempo, seja em múltiplos threads, processos ou em diferentes núcleos de processamento. O objetivo é permitir que programas façam mais de uma coisa simultaneamente, aumentando a eficiência em sistemas com múltiplos núcleos ou em operações de I/O intensivas.

A concorrência exige a divisão do programa em tarefas ou threads que podem rodar de forma independente. Isso força a decomposição do sistema em partes menores e mais modulares. Componentes que são projetados para serem executados concorrentemente (como threads ou co-rotinas) podem ser reutilizados em diferentes contextos. Por exemplo, um worker que processa filas pode ser usado em várias partes do sistema. A principal vantagem da programação concorrente é o aumento de desempenho. A capacidade de rodar múltiplas tarefas ao mesmo tempo pode reduzir o tempo de execução de programas que lidam com operações de I/O intensivas ou que possam ser paralelizados.

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

A autenticação e a autorização desempenham papéis cruciais na segurança de APIs, especialmente em organizações sociais que lidam com dados sensíveis. Vamos abordar a importância de cada um desses aspectos:

Autenticação

A autenticação é o processo de verificar a identidade de um usuário ou sistema que está tentando acessar a API. Em outras palavras, a autenticação confirma se o usuário ou serviço é realmente quem diz ser.

Importância da autenticação:

- Proteção contra acesso não autorizado: Sem autenticação adequada, qualquer pessoa ou sistema poderia tentar acessar a API, expondo dados sensíveis.
- Garantia de identidade: Em uma organização social, como em qualquer outra, garantir que apenas usuários ou serviços legítimos tenham acesso é essencial para evitar violações de dados.
- Controle de acesso a recursos: Autenticação assegura que apenas usuários autenticados possam interagir com os recursos oferecidos pela API.
- Segurança adicional com MFA (Autenticação Multifator): A adoção de MFA aumenta a segurança, exigindo mais de uma forma de verificação de identidade, protegendo contra ataques como roubo de senhas.

Autorização

Autorização é o processo que determina o que um usuário autenticado pode ou não fazer dentro do sistema. Enquanto a autenticação confirma quem é o usuário, a autorização controla seus níveis de acesso e permissões.

Importância da autorização:

- Controle granular de permissões: Uma organização social pode ter diferentes perfis de usuários (administradores, voluntários, parceiros), cada um com diferentes níveis de acesso aos dados. A autorização assegura que cada usuário acesse apenas os dados e recursos que lhe são permitidos.
- Proteção contra vazamento de dados: Sem uma autorização adequada, mesmo usuários autenticados podem acessar informações sensíveis que não deveriam. Isso pode causar vazamento de dados confidenciais, como informações pessoais de beneficiários.
- Prevenção de abusos: Um sistema de autorização robusto previne abusos de poder, como um voluntário ter permissões de administrador, ou um parceiro acessar dados internos da organização.
- Aplicação do princípio do menor privilégio: A autorização deve garantir que os usuários tenham apenas o mínimo necessário de acesso para desempenhar suas funções, o que reduz a superfície de ataque em caso de comprometimento de contas.

Conjunção de autenticação e autorização

A autenticação e a autorização, juntas, formam uma camada essencial de segurança nas APIs de organizações sociais. Elas garantem:

- **Confidencialidade:** Somente pessoas autorizadas podem acessar os dados sensíveis.
- **Integridade:** Usuários não autorizados são impedidos de alterar dados críticos.
- **Auditoria e rastreamento:** A partir de sistemas de autenticação e autorização bem implementados, é possível rastrear quem acessou ou modificou informações, o que é importante em auditorias de segurança.

Boas práticas de autenticação e autorização em APIs:

- **Uso de tokens (JWT, OAuth2):** Para autenticação e autorização, utilizando padrões seguros e amplamente aceitos.
- **Criptografia:** Garantir que as credenciais e tokens de autenticação sejam sempre transmitidos de forma criptografada (usando HTTPS).
- **Rotação de chaves e tokens:** Evitar o uso de credenciais estáticas ou de longo prazo, rotacionando regularmente tokens de acesso e chaves.
- **Controle de sessão:** Implementar limites de sessão, para expirar acessos inativos, minimizando o impacto em caso de credenciais comprometidas.

Assim, autenticação e autorização são indispensáveis para proteger a integridade, confidencialidade e disponibilidade dos dados sensíveis em APIs de organizações sociais.

3.2.2 FRONT-END

O desenvolvimento da interface front-end deste projeto engloba uma série de conceitos importantes para criar uma aplicação completa, eficiente e acessível. Foi essencial aplicar técnicas de design responsivo, que permitiram à interface se ajustar automaticamente a diferentes tamanhos de tela, sendo nosso uso voltado aos desktops, visto que o uso será exclusivo interno em máquinas locais. Isso assegura uma experiência de uso uniforme nos dispositivos, facilitando o acesso e garantindo a melhor experiência para os usuários. Junto a um design focado em UX/UI, buscamos criar uma interação agradável e intuitiva, facilitando

a navegação e a execução de tarefas. Uma interface bem planejada, que seja clara e visualmente atraente, ajudando a reduzir dificuldades e visando melhor a satisfação do usuário, tornando o uso do sistema mais eficiente e agradável.

A inclusão de painéis de controle nos permitiu apresentar dados de forma visual e em tempo real, oferecendo informações relevantes que ajudam na análise e na tomada de decisões. Esses elementos tornam o sistema uma ferramenta poderosa para aumentar a efetividade das ações da organização.

O uso de frameworks no desenvolvimento web, como o Angular, que foi o escolhido para desenvolvermos nossa aplicação, nos forneceram uma base sólida para a criação de interfaces dinâmicas e responsivas, permitindo o desenvolvimento de componentes reutilizáveis e facilitando a manutenção do código. Além disso, eles melhoram a performance da aplicação, proporcionando uma experiência de usuário mais rápida e fluida.

É imprescindível a realização de testes de usabilidade e acessibilidade. Esses testes permitem identificar e corrigir possíveis dificuldades que diferentes grupos de usuários possam enfrentar, garantindo que a aplicação seja acessível para pessoas com diferentes habilidades e necessidades. Assim, é possível assegurar que o sistema atenda a organização social que utilizará o sistema.

3.3 COMPUTAÇÃO EM NUVEM

A computação em nuvem é uma abordagem revolucionária na tecnologia da informação que permite o acesso sob demanda a recursos computacionais, como armazenamento, servidores e aplicativos, via internet. Este modelo elimina a necessidade de aquisição e manutenção de infraestrutura física local, substituindo-a pelo consumo de serviços pagos conforme o uso, muitas vezes escaláveis de forma automática.

De acordo com Santos, esse paradigma tem sido amplamente adotado por organizações devido à flexibilidade, redução de custos e capacidade de oferecer soluções acessíveis em qualquer momento e lugar. A obra explora aspectos como os modelos de serviço (IaaS, PaaS, SaaS), desafios de migração, regulamentações e as vantagens estratégicas da nuvem no mercado corporativo

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

Utilizar a computação em nuvem permite a automação de processos e centraliza o gerenciamento de dados do estoque, facilitando o controle e atualização dos registros em tempo real. Isso significa que a equipe pode acessar e atualizar os dados de estoque a partir de qualquer dispositivo com acesso à internet, o que reduz o tempo gasto em tarefas manuais e minimiza o risco de erros humanos.

Também nos permite aumentar ou diminuir a capacidade de armazenamento e processamento conforme as necessidades do estoque, sem custos ou configurações adicionais complexas. Caso a APAE amplie suas operações ou precise armazenar mais dados, a nuvem facilita essa expansão sem atrasos, atendendo a mudanças de demanda de forma ágil e eficiente.

Uma infraestrutura de nuvem confiável oferece backup automático e recursos de recuperação de dados em caso de falhas ou incidentes, garantindo a segurança das informações.

A APAE pode manter a segurança dos dados do sistema de estoque, evitando perda de informações e garantindo a continuidade das operações, mesmo em casos de imprevistos.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

A implementação da nuvem é uma estratégia sólida para a APAE otimizar seus recursos, adaptando-se a novas demandas de forma sustentável e segura. Essas vantagens podem tornar o sistema de estoque mais robusto, permitindo que a instituição concentre seus esforços e recursos na missão de apoiar a comunidade que atende.

Armazenar Dados de Estoque de Forma Segura e Centralizada

- Implementar um sistema de armazenamento em nuvem para centralizar todos os dados relacionados ao estoque (produtos, quantidades, movimentações, datas de validade, etc.) em um único local seguro e de fácil acesso.

Garantir a Escalabilidade do Armazenamento de Dados

- Utilizar a capacidade de escalabilidade da computação em nuvem para expandir o armazenamento de dados conforme a necessidade, sem a necessidade de grandes investimentos em infraestrutura física.

Reduzir Custos com Infraestrutura e Manutenção de Dados

- Substituir a infraestrutura local de armazenamento (servidores, backups físicos) por soluções de armazenamento em nuvem, o que reduzirá os custos com hardware, energia e manutenção de equipamentos físicos.

Melhorar a Segurança e Recuperação de Dados

- Implementar soluções de segurança em nuvem, como criptografia de dados, autenticação multifatorial e backups automáticos, para proteger as informações de estoque contra perda, danos ou acessos não autorizados.

Facilitar o Acesso Remoto e a Colaboração

- Permitir o acesso remoto ao sistema de estoque por meio de dispositivos conectados à internet, promovendo a colaboração entre diferentes equipes e departamentos da APAE (por exemplo, administrativos, de logística, financeiro e operacional).

Garantir a Alta Disponibilidade e Continuidade do Serviço

- Garantir que os dados de estoque estejam sempre disponíveis e acessíveis, utilizando a infraestrutura da nuvem para garantir alta disponibilidade e continuidade do serviço, mesmo em caso de falhas técnicas.

Facilitar a Implementação de Relatórios e Análises de Dados

- Utilizar as ferramentas de análise de dados e relatórios disponíveis na plataforma de nuvem para gerar insights sobre o estoque, como tendências de consumo, necessidades de reposição, e datas de validade de produtos.

Proporcionar Flexibilidade e Agilidade na Gestão de Estoque

- Implementar um sistema ágil e flexível de gestão de estoque em nuvem, que permita à APAE adaptar rapidamente suas operações de acordo com as necessidades do momento, como a entrada de grandes doações ou ajustes na distribuição de materiais.

Integrar o Sistema de Estoque com Outros Sistemas de Gestão

- Facilitar a integração do sistema de estoque em nuvem com outros sistemas utilizados pela APAE, como sistemas financeiros, de gestão de voluntários ou de comunicação com doadores.

Melhorar a Experiência do Usuário com Interface Intuitiva

- Desenvolver uma interface de usuário simples e intuitiva para o sistema de gestão de estoque em nuvem, permitindo que os funcionários e voluntários da APAE possam facilmente cadastrar produtos, consultar dados e gerar relatórios, mesmo sem conhecimento técnico avançado.

3.3.3 VANTAGENS DA CLOUD COMPUTING

Melhorar a Eficiência Operacional

Utilizar a computação em nuvem permite a automação de processos e centraliza o gerenciamento de dados do estoque, facilitando o controle e atualização dos registros em tempo real.

Isso significa que a equipe pode acessar e atualizar os dados de estoque a partir de qualquer dispositivo com acesso à internet, o que reduz o tempo gasto em tarefas manuais e minimiza o risco de erros humanos.

Reduzir Custos Operacionais

A nuvem elimina a necessidade de investimento em infraestrutura física, como servidores locais e equipamentos de armazenamento de dados. Sólida para a APAE otimizar seus recursos, adaptando-se a novas demandas de forma sustentável e segura. Essas vantagens

podem tornar o sistema de estoque mais robusto, permitindo que a instituição concentre seus esforços e recursos na missão de apoiar a comunidade que atende.

Como a nuvem funciona em modelo de assinatura ou pagamento por uso, a APAE poderá reduzir gastos com manutenção e atualização de hardware, pagando apenas pelo espaço e serviços que realmente utiliza.

Caso a APAE amplie suas operações ou precise armazenar mais dados, a nuvem facilita essa expansão sem atrasos, atendendo a mudanças de demanda de forma ágil e eficiente.

Proteger Dados Sensíveis com Backup e Recuperação

Uma infraestrutura de nuvem confiável oferece backup automático e recursos de recuperação de dados em caso de falhas ou incidentes, garantindo a segurança das informações.

A APAE pode manter a segurança dos dados do sistema de estoque, evitando perda de informações e garantindo a continuidade das operações, mesmo em casos de imprevistos.

Promover a Colaboração e Transparência

A nuvem facilita o acesso simultâneo ao sistema de estoque por várias pessoas, o que ajuda na colaboração e coordenação entre os diferentes funcionários e voluntários.

Com o sistema na nuvem, toda a equipe pode verificar e gerenciar o estoque de forma transparente, promovendo uma melhor organização e controle das necessidades de insumos.

Aumentar a Escalabilidade

Elasticidade é uma das maiores vantagens da computação em nuvem. Ela permite que a infraestrutura de TI se ajuste automaticamente às necessidades da empresa, ou seja, a capacidade de recursos (como armazenamento e processamento de dados) pode ser aumentada ou diminuída de acordo com a demanda. Aplicado na APAE, se o volume de dados de estoque crescer, eles podem aumentar rapidamente o armazenamento na nuvem sem

a necessidade de adquirir hardware físico. Caso o volume diminua ou a organização tenha períodos com menos dados, pode reduzir os recursos de forma igualmente simples. precise de mais capacidade de backup por algum motivo específico (como um grande aumento de dados durante inventários), a nuvem permitirá que ela ajuste os recursos rapidamente, sem interrupção ou necessidade de investimento prévio.

O modelo de pagamento conforme o uso significa que a APAE pagará apenas pelos serviços que consumir, armazenamento de dados utilizado, sem a necessidade de investir em servidores ou outros equipamentos caros. Isso é especialmente importante para organizações sem fins lucrativos, que devem otimizar seus orçamentos. Embora o pagamento seja baseado no uso, a AWS oferece ferramentas de monitoramento e alertas de custo, permitindo que a APAE acompanhe em tempo real os gastos com backup e se ajuste conforme necessário.

A alta disponibilidade é um conceito central na computação em nuvem, que garante que os serviços e dados estarão sempre acessíveis, mesmo em casos de falhas de hardware, perda de rede ou outros imprevistos. Provedores como AWS operam com múltiplos centros de dados distribuídos globalmente, o que assegura redundância e continuidade dos serviços. Com múltiplas cópias dos dados espalhadas por diferentes locais e sistemas de backup automáticos, a segurança dos dados é significativamente aumentada. Em caso de falha em uma região, os dados ainda estarão acessíveis a partir de outra, garantindo que a APAE não perca informações vitais para o funcionamento da instituição.

Em geral, todas essas vantagens não apenas ajudam a APAE a economizar dinheiro, mas também melhoram a segurança dos dados de estoque, permitindo que a instituição possa operar de maneira mais eficiente e robusta. Com isso se resulta em uma operação mais ágil, segura e com um custo mais baixo, fatores extremamente importantes para uma organização sem fins lucrativos.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

Para implementar um projeto de armazenamento em nuvem que será utilizado apenas para fazer backup de dados de estoque, é importante considerar o modelo de computação em nuvem adequado, com base nas suas necessidades e no tipo de serviço desejado que no caso do projeto precisamos de um serviço que envolve salvar backups de forma eficiente e econômica.

Infrastructure as a Service (IaaS)

- **O que é?** IaaS fornece infraestrutura de TI, como máquinas virtuais, armazenamento, redes e servidores na nuvem. Você tem controle total sobre a infraestrutura e gerencia tudo, desde o sistema operacional até as configurações de rede.
- **Como se aplica?** Com IaaS, você teria um servidor virtual (VM) em um provedor de nuvem (como AWS EC2, Google Compute Engine ou Azure Virtual Machines). Se configura o sistema para gerenciar os backups localmente e, depois, programaria a transferência desses arquivos de backup para o armazenamento em nuvem (como S3 ou Blob Storage).
- **Vantagens:**
 - Controle total sobre o ambiente.
 - Capacidade de configurar o sistema de backup com flexibilidade.
 - Bom para projetos que precisam de personalização avançada.
- **Desvantagens:**
 - Requer mais esforço para configurar e manter.
 - Custo geralmente mais alto, porque você precisa provisionar e manter a infraestrutura.

Conclusão: IaaS não seria a melhor escolha aqui, pois exige mais gerenciamento e manutenção para um simples projeto de backup, além de ter custos operacionais maiores.

Platform as a Service (PaaS)

- **O que é?:** PaaS fornece uma plataforma gerenciada onde você pode executar e gerenciar suas aplicações sem se preocupar com a infraestrutura subjacente. Exemplos incluem bancos de dados gerenciados (como AWS RDS, Azure SQL, ou Google Cloud SQL) ou plataformas para hospedagem de aplicações.
- **Como se aplica?:** Em um modelo PaaS, você poderia usar um banco de dados gerenciado na nuvem para o armazenamento dos dados de estoque. Além disso, você poderia usar um serviço de backup gerenciado para armazenar os backups automaticamente na nuvem.
- **Vantagens:**

- Reduz a carga de gerenciamento de infraestrutura.
- Oferece autoSoftware as a Service (SaaS)
- **O que é?** SaaS oferece software e serviços prontos para uso via nuvem, sem a necessidade de gerenciar infraestrutura ou plataformas. Exemplos incluem Dropbox, Google Drive, ou Backblaze.
- **Como se aplica?** Neste caso, você usaria um serviço de backup em nuvem com alta disponibilidade.
 - Simplicidade de uso, com escalabilidade e segurança embutidas.
- **Desvantagens:**
 - Menos flexibilidade para personalizações complexas.
 - Custos podem ser mais altos se o volume de dados crescer rapidamente.

Conclusão: PaaS pode ser interessante para o caso de manter o banco de dados diretamente na nuvem com backups automatizados, mas como seu foco é manter os dados localmente e apenas fazer backup para a nuvem duas vezes por semana, PaaS pode ser uma solução mais sofisticada do que o necessário.

Software as a Service (SaaS)

- **O que é?:** SaaS oferece software e serviços prontos para uso via nuvem, sem a necessidade de gerenciar infraestrutura ou plataformas. Exemplos incluem Dropbox, Google Drive, ou Backblaze.
- **Como se aplica?:** Neste caso, você usaria um serviço de armazenamento em nuvem para realizar o backup. Serviços como Backblaze B2, Google Cloud Storage, AWS S3, ou Wasabi seriam configurados como destinos para os seus backups. Você geraria o backup localmente e o enviaria para o serviço escolhido.
- **Vantagens:**
 - Menos esforço de configuração e manutenção.
 - Custos baixos para armazenamento e backup.
 - Fácil de usar, com integração simples.
- **Desvantagens:**
 - Menor controle sobre a infraestrutura.
 - Menos flexibilidade em personalizações avançadas.

Conclusão: SaaS é a melhor opção para o projeto, já que precisamos de um serviço para armazenar backups do estoque e se vê desnecessário o gerenciamento de uma infraestrutura complexa. Podendo agendar e automatizar os backups diretamente para um serviço de armazenamento em nuvem.

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

A escolha do provedor de nuvem com o custo mais baixo depende de vários fatores, como o volume de dados que será armazenado, a frequência de acessos, e as regiões de armazenamento. Pelo fato do projeto só ter a necessidade de armazenar o backup e ter poucos acessos fizemos uma comparação dos principais serviços em nuvem com base no custo de armazenamento e transferência de dados sendo eles:

Amazon Web Services (AWS) - S3

- Custo de Armazenamento:
 - Standard: US\$ 0,023 por GB/mês.
 - Infrequent Access: US\$ 0,0125 por GB/mês (adequado para backups que você acessa raramente).
 - Glacier (Arquivo): US\$ 0,004 por GB/mês (para backups de longo prazo).
- Transferência de Dados:
 - Primeiro GB gratuito por mês.
 - Depois: US\$ 0,09 por GB para transferências de saída além do limite gratuito.
- Vantagem: O AWS S3 oferece diversas camadas de armazenamento com flexibilidade para ajustar ao seu uso.

Google Cloud - Cloud Storage

- Custo de Armazenamento:
 - Standard: US\$ 0,020 por GB/mês.
 - Nearline (30 dias de armazenamento mínimo): US\$ 0,010 por GB/mês.
 - Coldline (90 dias de armazenamento mínimo): US\$ 0,007 por GB/mês.

- Archive (365 dias de armazenamento mínimo): US\$ 0,0027 por GB/mês (ideal para arquivos raramente acessados).
- Transferência de Dados:
 - Primeiro GB gratuito por mês.
 - Depois: US\$ 0,12 por GB de saída.
- Vantagem: Google Cloud tem custos ligeiramente menores que AWS para camadas de armazenamento frio.

Para flexibilidade: AWS S3 e Google Cloud têm muitas opções de camadas de armazenamento, mas podem ser mais caros para grandes volumes de dados ou acessos frequentes.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

O uso da nuvem para armazenamento de backup nos permite guardar cópias de segurança de dados críticos com alta disponibilidade e confiabilidade.

- Exemplo: Utilizar serviços como Amazon S3, Google Cloud Storage, ou Azure Blob Storage para armazenar backups de bancos de dados, registros de insumos e movimentações.
- Motivo: Esses serviços oferecem escalabilidade automática, replicação geográfica e custo eficiente para armazenar grandes volumes de dados.

Estratégia de Backup:

- Incremental: Apenas os dados alterados desde o último backup são enviados para a nuvem, reduzindo custos e tempo de transferência.
- Agendamento: Realizar backups em horários pré definidos, como duas vezes por semana (conforme definido anteriormente).
- Política de Retenção: Manter versões anteriores dos backups por um período definido, garantindo recuperação de dados em caso de necessidade.

Importância e Funcionamento do Balanceamento de Carga no Backup

Embora o foco principal do balanceamento de carga seja em aplicações de alto tráfego, ele também pode ser relevante no contexto de backups:

- **Importância no Backup:**
 - Distribuir as requisições de upload ou recuperação de dados entre múltiplos servidores ou pontos de acesso na nuvem.
 - Garantir que a operação de backup não sobrecarregue o servidor principal ou cause lentidão para os usuários.

Exemplo de Funcionamento:

- Ao enviar grandes volumes de dados para o armazenamento na nuvem, o serviço pode distribuir as requisições por múltiplos data centers ou pontos de upload, garantindo maior eficiência no processo.

Anatomia do Armazenamento em Nuvem para Backup

- **Camada Física:**
 - Os dados de backup são armazenados em data centers distribuídos globalmente.
 - A replicação geográfica garante redundância e resiliência contra falhas regionais.
- **Camada de Gerenciamento:**
 - Os provedores de nuvem oferecem ferramentas para monitorar o status dos backups e gerenciar políticas de retenção e acessibilidade.
- **Camada de Serviço:**
 - Serviços de armazenamento como Amazon S3 ou Google Cloud Storage integram diretamente APIs para envio e recuperação de backups.

Paradigmas Tecnológicos no Backup na Nuvem

- **Elasticidade:**
 - A capacidade de escalar o armazenamento conforme o volume de dados aumenta garante que não há limite rígido para o crescimento.
- **Replicação Automática:**
 - Backups são replicados automaticamente entre diferentes data centers, garantindo alta disponibilidade.
- **Pagamentos Baseados no Uso:**
 - O modelo "pay-as-you-go" permite pagar apenas pelo armazenamento realmente utilizado.
- **Segurança e Criptografia:**
 - Dados de backup são criptografados em trânsito (durante a transferência) e em repouso (armazenados na nuvem).

Ao utilizar a nuvem para armazenamento de backups, o foco deve estar na confiabilidade, segurança e custo-benefício. Serviços de armazenamento em nuvem como Amazon S3, Google Cloud Storage, ou Azure Blob Storage oferecem as ferramentas necessárias para gerenciar backups com eficiência, escalabilidade e segurança, alinhando-se às necessidades do projeto.

3.3.7 GOOGLE CLOUD ou AWS

O principal objetivo do uso da nuvem neste projeto é implementar uma solução confiável e escalável para armazenamento de backup dos dados do sistema de gerenciamento de estoque. Isso garante proteção contra perda de dados, alta disponibilidade, e facilidade de recuperação em situações de falha ou necessidade de auditoria.

Proposta de Implantação e Aplicação do Projeto de Cloud Computing

Objetivo do Projeto na Nuvem

O principal objetivo do uso da nuvem neste projeto é implementar uma solução confiável e escalável para armazenamento de backup dos dados do sistema de gerenciamento de estoque. Isso garante proteção contra perda de dados, alta disponibilidade, e facilidade de recuperação em situações de falha ou necessidade de auditoria.

Etapas da Implantação

1. Escolha do Provedor de Nuvem:

- Selecionar um provedor que atenda aos requisitos de segurança, custo-benefício, e facilidade de integração.
- **Opções recomendadas:**
 - **Amazon Web Services (AWS):** Amazon S3 para backup com políticas de ciclo de vida para retenção.
 - **Google Cloud:** Google Cloud Storage com suporte a arquivamento e recuperação eficiente.
 - **Microsoft Azure:** Azure Blob Storage para gerenciamento de backups.

2. Definição da Arquitetura:

- **Backup Incremental:** Somente os dados alterados são enviados após o backup inicial, otimizando uso de recursos e tempo.
- **Automação de Backups:**
 - Implementar agendamentos para backups automáticos (duas vezes por semana) utilizando ferramentas como AWS Backup, Google Cloud Scheduler ou scripts personalizados.
- **Política de Retenção:**
 - Manter backups recentes (últimos 7 a 30 dias) e mover backups antigos para armazenamento de custo reduzido (cold storage).
- **Recuperação de Dados:**
 - Configurar processos automatizados para restaurar os dados do backup em caso de necessidade.

3. Configuração de Segurança:

- **Criptografia:**
 - Criptografar os dados antes do envio para a nuvem (em trânsito) e armazená-los de forma criptografada (em repouso).
 - **Controle de Acesso:**
 - Implementar políticas de acesso baseadas em funções (RBAC) para restringir quem pode visualizar, criar ou restaurar backups.
 - **Logs e Auditoria:**
 - Habilitar logs de acesso e atividade para monitorar e auditar o uso do sistema.
4. **Testes e Validação:**
- Realizar testes periódicos de backup e restauração para validar a integridade dos dados e o funcionamento do sistema.
 - Simular cenários de falha para garantir que o sistema de backup suporta recuperação eficiente.
5. **Treinamento e Documentação:**
- Treinar a equipe para utilizar o sistema de backup na nuvem e garantir que todos compreendam as políticas e processos de recuperação.
 - Documentar os procedimentos para futuras referências e auditorias.

Funcionalidades e Gerenciamento da Nuvem

Principais Funcionalidades

1. **Armazenamento Escalável:**
 - Adaptação automática ao crescimento dos dados, sem necessidade de intervenções manuais.
2. **Gestão de Ciclo de Vida:**
 - Mover automaticamente backups antigos para camadas de armazenamento mais baratas (como Amazon Glacier ou Google Coldline).
3. **Monitoramento e Alertas:**
 - Alertas sobre falhas no backup ou superação de limites de armazenamento.
4. **Replicação Geográfica:**
 - Garantir que os backups estejam disponíveis em múltiplas regiões para proteção contra desastres.
5. **Integração com Sistemas Locais:**

- APIs e ferramentas para integração com o sistema local de gerenciamento de estoque.

Gerenciamento da Nuvem

1. Políticas de Acesso e Segurança:

- Implementar autenticação multifator (MFA) e políticas de acesso com base em permissões granulares.

2. Monitoramento de Custos:

- Utilizar ferramentas nativas como AWS Cost Explorer ou Google Billing para acompanhar gastos e otimizar custos.

3. Escalabilidade Automatizada:

- Configurar auto-escalabilidade para ajustar a capacidade de armazenamento automaticamente, eliminando a necessidade de alocação manual.

4. Resiliência e Disponibilidade:

- Implementar políticas de replicação e alta disponibilidade para garantir que os backups estejam acessíveis mesmo em cenários de falha.

A proposta de implantação do projeto de cloud computing foca na integração do sistema de gerenciamento de estoque com soluções robustas de armazenamento na nuvem. O uso de serviços como **Amazon S3**, **Google Cloud Storage**, ou **Azure Blob Storage** garante:

- **Segurança:** Criptografia e políticas de acesso confiáveis.
- **Escalabilidade:** Adaptação automática à demanda.
- **Alta Disponibilidade:** Replicação geográfica e redundância de dados.

O gerenciamento eficiente da nuvem com monitoramento de custos, auditorias e automação de processos permitirá que o sistema funcione com confiabilidade e otimização de recursos.

3.4 ESTRUTURA DE DADOS

Segundo o autor Tenebaum, Estruturas de dados são formas de organizar e armazenar informações em um computador para que possam ser acessadas e modificadas de maneira eficiente. Elas incluem elementos básicos como vetores e listas encadeadas, bem como estruturas mais complexas, como pilhas, filas, árvores e grafos. A escolha da estrutura de dados adequada pode afetar diretamente a eficiência de um algoritmo, influenciando o tempo de execução e o uso de memória. Além disso, o entendimento de estruturas de dados é essencial para resolver problemas computacionais, desde tarefas simples como a ordenação de números até questões mais avançadas como o design de sistemas distribuídos. Um dos objetivos principais ao estudar estruturas de dados é compreender as propriedades e as operações associadas a cada tipo, bem como os cenários em que cada um é mais adequado.

3.4.1 LEVANTAMENTO DE REQUISITOS

Com base nos requisitos levantados e na natureza do sistema (gerenciamento de estoque, doações, entradas e saídas de insumos), a escolha da estrutura de dados adequada será guiada por critérios como eficiência, complexidade e adequação ao contexto. Aqui está a análise para cada módulo do sistema:

1. Módulo: Gerenciamento de Usuários

- **Operações Principais:**
 - Inserção de novos usuários.
 - Validação de credenciais durante o login.
 - Busca de informações de usuários.

- **Estrutura Recomendada: Tabela de Dispersão (Hash Table).**
 - **Motivo:** Proporciona eficiência na busca e validação de credenciais (tempo médio de $O(1)$).
 - **Implementação:** O nome de usuário pode ser usado como chave, com valores associados a informações como senha criptografada e permissões.

2. Módulo: Controle de Insumos

- **Operações Principais:**
 - Inserção e remoção de registros de insumos.
 - Atualização de dados (nome, descrição, unidade de medida).
 - Busca frequente por insumos para entradas, saídas e doações.
- **Estrutura Recomendada: Árvore Binária de Busca Balanceada (ex.: AVL Tree ou Red-Black Tree).**
 - **Motivo:** Suporta buscas eficientes ($O(\log n)$) e mantém os dados ordenados, facilitando operações como listagem em ordem alfabética.
 - **Alternativa:** Uma tabela hash também pode ser eficiente para buscas exatas, mas não facilita operações que dependam de ordenação.

3. Módulo: Estoque (Entradas e Saídas)

- **Operações Principais:**
 - Inserção de novos registros (entrada ou saída).
 - Busca por movimentações recentes.
 - Atualização de estoque com base nas movimentações.
- **Estrutura Recomendada:**
 - **Fila** para registrar as movimentações em ordem cronológica (entrada e saída).
 - **Motivo:** Garante que o processamento das movimentações ocorra em ordem de chegada ($O(1)$) para inserção e remoção).
 - **Árvore de Segmentos** ou **Tabelas Hash** para rastrear o saldo atual de cada insumo.
 - **Motivo:** Ambas permitem atualizações rápidas e suporte a buscas eficientes ($O(\log n)$) para árvores e $O(1)$ em hash tables).
 - **Detalhe:** Escolher entre as duas dependerá da frequência de operações de agregação (como totais por categoria).

4. Módulo: Doações

- **Operações Principais:**

- Inserção de novas doações.
- Busca por doações com base em doadores ou datas.
- Listagem de doações por insumo.

- **Estrutura Recomendada:**

- **Lista Duplamente Encadeada** para registrar as doações.
 - **Motivo:** Suporta inserções rápidas ($O(1)O(1)O(1)$) e navegação bidirecional para listar doações mais recentes ou anteriores.
- **Árvore Binária de Busca** para indexar por doador ou insumo.
 - **Motivo:** Facilita buscas rápidas ($O(\log n)O(\log n)O(\log n)$) e operações de ordenação.

Considerações sobre Desempenho

Frequência de Operações:

- Se as buscas forem muito frequentes e exigirem ordenação, estruturas baseadas em árvores são mais apropriadas.
- Para buscas exatas, tabelas de dispersão são mais eficientes.

2. Memória:

- Estruturas como árvores ocupam mais memória devido a ponteiros adicionais, mas oferecem maior flexibilidade.
- Tabelas de dispersão podem consumir mais memória em caso de alto fator de carga.

3. Tempo de Resposta:

- Operações críticas, como validação de credenciais e consulta de estoque, devem priorizar estruturas que oferecem tempo constante ($O(1)O(1)O(1)$) ou logarítmico ($O(\log n)O(\log n)O(\log n)$).

Cada módulo do sistema pode ser estruturado com a combinação de estruturas de dados que melhor atende aos seus requisitos específicos de desempenho e funcionalidade. Durante a implementação, ajustes podem ser feitos para otimizar ainda mais o tempo de resposta e o uso de recursos, garantindo que o sistema seja escalável e eficiente.

3.4.2 VALIDAÇÃO DOS REQUISITOS

Para garantir que o sistema de gerenciamento de estoque, com foco em insumos, entradas e saídas, doações, e usuários, seja eficiente e escalável em um ambiente de nuvem, é crucial considerar tanto as características das estruturas de dados quanto a forma como elas interagem com as tecnologias de nuvem. A nuvem oferece escalabilidade, elasticidade e distribuição de carga, mas a eficiência e integridade dos dados dependem de como as estruturas de dados são escolhidas e implementadas.

Aqui estão os pontos principais a serem considerados ao validar a integração das estruturas de dados com o sistema de computação em nuvem:

Otimização do Armazenamento

A nuvem oferece escalabilidade dinâmica e flexível, o que significa que as soluções de armazenamento precisam ser eficientes tanto no gerenciamento de grandes volumes de dados quanto na redução dos custos associados ao armazenamento. A escolha das estruturas de dados deve se alinhar com essas necessidades:

- **Banco de Dados NoSQL (Documentos ou Key-Value Stores):**
 - Uso: Para armazenar dados que podem ter estrutura flexível, como registros de insumos, entradas e saídas de estoque, e doações.
 - Exemplo: MongoDB, Amazon DynamoDB, ou Google Firestore.
 - Motivo: Estruturas NoSQL podem ser ajustadas conforme a demanda, permitindo escalabilidade horizontal e otimização de leitura/escrita, especialmente quando se lidam com grandes volumes de dados não estruturados ou semiestruturados.

- Benefício: Maior flexibilidade para armazenar dados com diferentes formatos e tipos, além de ser ideal para escalabilidade horizontal em nuvem.

- **Banco de Dados Relacional (RDBMS):**
 - Uso: Caso o sistema necessite de integridade transacional, como nas movimentações de estoque (entradas/saídas), um banco de dados relacional (ex.: MySQL, PostgreSQL) com alta disponibilidade pode ser implementado.
 - Exemplo: Amazon RDS (PostgreSQL, MySQL).
 - Motivo: RDBMS oferece consistência e integridade de dados com suporte para ACID (Atomicidade, Consistência, Isolamento, Durabilidade), sendo adequado para sistemas que precisam garantir que todas as operações sejam realizadas corretamente.
 - Benefício: Permite consultas SQL poderosas, com integridade referencial e suporte a transações complexas, garantindo dados consistentes.

- **Armazenamento em Blobs ou Arquivos:**
 - Uso: Para dados grandes ou arquivos como relatórios e backups de dados.
 - Exemplo: Amazon S3, Google Cloud Storage.
 - Motivo: Armazenamento de dados não estruturados em nuvem, com capacidade de escalar de forma eficiente à medida que o volume de dados cresce.
 - Benefício: Oferece baixo custo por GB e alta disponibilidade, ideal para backups regulares e dados pesados.

Gerenciamento de Dados Distribuídos

Em ambientes de nuvem, a distribuição de dados é essencial para garantir alta disponibilidade e a distribuição de carga. As seguintes considerações são necessárias para lidar com dados distribuídos:

- **Particionamento de Dados:**
 - Uso: Dividir grandes volumes de dados em partes menores, chamadas de "shards" ou "partições", distribuídas por diferentes servidores ou instâncias.

- Exemplo: Amazon DynamoDB, que realiza o particionamento automático de dados.
 - Motivo: Permite que as operações de leitura e escrita sejam distribuídas entre múltiplas instâncias, reduzindo a sobrecarga e melhorando a escalabilidade.
 - Benefício: A carga de trabalho pode ser balanceada entre várias instâncias de forma eficiente, garantindo maior performance durante picos de demanda.
- **Replicação de Dados:**
 - Uso: Criar cópias de dados em diferentes zonas de disponibilidade para garantir alta disponibilidade e recuperação em caso de falha.
 - Exemplo: Amazon RDS, Google Cloud Spanner.
 - Motivo: A replicação de dados garante que os dados estejam sempre disponíveis, mesmo em caso de falhas de infraestrutura, o que é essencial para sistemas críticos.
 - Benefício: Redundância de dados com a garantia de consistência e alta disponibilidade, mesmo em cenários de falha ou de picos de tráfego.
- **Armazenamento de Dados em Cache:**
 - Uso: Usar sistemas de cache para melhorar a performance de leitura, reduzindo a latência ao acessar dados frequentemente consultados.
 - Exemplo: Redis, Amazon ElastiCache.
 - Motivo: Ao armazenar dados mais acessados em cache, a leitura pode ser feita de forma mais rápida, sem sobrecarregar o banco de dados.
 - Benefício: Acelera o tempo de resposta, especialmente em operações de leitura frequentes, como consultas de estoque ou de registros de doações.

Garantir Integridade dos Dados

A integridade dos dados é essencial para garantir que o sistema funcione corretamente, especialmente em um ambiente distribuído na nuvem:

- **Transações Distribuídas:**
 - Uso: Em sistemas distribuídos, transações podem ser necessárias para garantir que múltiplas operações sejam completadas com sucesso, mantendo a consistência.

- Exemplo: Google Cloud Spanner, Amazon Aurora (com suporte a transações distribuídas).
- Motivo: Em um ambiente de nuvem com múltiplas instâncias e servidores, garantir que operações críticas (como entrada e saída de estoque) sejam realizadas de forma atômica é vital para a consistência.
- Benefício: As transações distribuídas garantem que, em caso de falha, o sistema possa reverter para um estado consistente sem perda de dados.

- **Mecanismos de Consistência Eventual:**

- Uso: Para garantir alta disponibilidade, algumas soluções de nuvem adotam consistência eventual em vez de consistência forte.
- Exemplo: Amazon DynamoDB, Google Cloud Bigtable.
- Motivo: Em sistemas de alta disponibilidade e escalabilidade, pode-se optar por consistência eventual, onde os dados podem ser temporariamente inconsistentes, mas eventualmente convergem para o estado correto.
- Benefício: Permite escalabilidade horizontal sem sacrificar a disponibilidade, ideal para sistemas que não exigem uma consistência imediata.

Escalabilidade e Elasticidade

A elasticidade é uma característica essencial da nuvem, permitindo que o sistema se ajuste automaticamente à carga de trabalho, aumentando ou diminuindo os recursos conforme necessário:

- **Auto-escalabilidade:**

- Uso: Ajustar automaticamente o número de instâncias de banco de dados ou servidores conforme a demanda (por exemplo, Amazon EC2 Auto Scaling).
- Motivo: Garante que o sistema possa lidar com picos de demanda sem perder desempenho, fornecendo mais recursos quando necessário.
- Benefício: Economiza recursos, pois só utiliza mais instâncias quando necessário, mantendo o custo sob controle.

- **Balanceamento de Carga:**

- Uso: Usar balanceadores de carga para distribuir as requisições de usuários entre múltiplas instâncias de servidores.
- Exemplo: Elastic Load Balancer (AWS).
- Motivo: Facilita a distribuição de tráfego entre servidores, evitando sobrecarga em uma única instância e garantindo desempenho consistente.

Para otimizar o armazenamento e garantir que o sistema gere grandes volumes de dados de maneira eficiente em nuvem, a escolha das estruturas de dados deve ser feita com foco em escalabilidade, elasticidade, consistência e eficiência de leitura/escrita. Além disso, a integração com soluções de nuvem deve considerar:

- Elasticidade para adaptar-se automaticamente às cargas variáveis.
- Alta disponibilidade por meio de replicação e particionamento de dados.
- Integridade de dados com transações distribuídas e consistência eventual, quando necessário.

Essas práticas garantirão um desempenho adequado, mesmo em cenários de alta demanda e grandes volumes de dados, atendendo aos requisitos do sistema e garantindo que ele seja eficiente e resiliente em ambientes de nuvem.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS

3.5.1 ENFRENTANDO ESTEREÓTIPOS

Estereótipo e Convívio Social

Os estereótipos são generalizações que, muitas vezes, influenciam como percebemos as pessoas ao nosso redor, afetando diretamente as interações sociais. Eles podem gerar preconceitos que impactam o convívio de maneira negativa, criando barreiras entre grupos. Um exemplo prático é o julgamento precipitado de que alguém de uma determinada classe social ou região do país tem um determinado comportamento ou forma de pensar. Esses julgamentos criam ambientes excludentes, levando a distanciamentos desnecessários.

Estereótipo e Representação

A maneira como diferentes grupos são representados na mídia, na publicidade e no entretenimento é muitas vezes estereotipada, reforçando imagens distorcidas. Isso molda a percepção coletiva e contribui para a manutenção desses estereótipos. Por exemplo, a representação de mulheres em posições subalternas ou associadas a papéis de cuidado perpetua a ideia de que elas são menos adequadas para cargos de liderança.

Troco Likes: A Idealização da Vida na Internet

As redes sociais promovem uma cultura de perfeição e idealização da vida, onde as pessoas compartilham apenas os melhores momentos, criando expectativas irreais. Isso pode alimentar estereótipos de sucesso e felicidade, enquanto as dificuldades são ocultadas. Essa idealização contribui para a criação de padrões inalcançáveis, o que gera frustração e baixa autoestima.

Convivendo com a Diferença

Convivência em ambientes diversos, onde pessoas de diferentes origens, crenças e formas de viver se encontram, desafia os estereótipos. Aceitar a diferença exige empatia, entendimento e uma disposição para desconstruir preconceitos. As escolas, por exemplo, são ambientes fundamentais para ensinar a convivência com a diversidade, ajudando a formar indivíduos mais abertos e respeitosos.

3.5.2 ESTUDANTES NA PRÁTICA

Particularidades Regionais e os Estereótipos Culturais

O Brasil é um país continental, com uma rica diversidade cultural. As diferentes regiões possuem costumes, sotaques e paisagens únicas. No entanto, essa variedade muitas vezes é simplificada pela criação de estereótipos, generalizações que podem distorcer a realidade e perpetuar preconceitos.



A Formação de Estereótipos

- 1 Simplificação da Realidade**
Estereótipos reduzem a complexidade das culturas regionais a características superficiais, ignorando nuances e individualidades.
- 2 Generalizações Prejudiciais**
Esses rótulos podem levar à discriminação, prejudicando a imagem e oportunidades de indivíduos e comunidades.
- 3 Influência da Mídia**
Representações estereotipadas na mídia, como filmes e novelas, contribuem para a perpetuação de ideias preconcebidas.
- 4 Falta de Conhecimento**
A falta de contato direto com diferentes regiões e culturas contribui para a formação de estereótipos, alimentados pela ignorância.



A Relação Interior e Campo

Percepção Comum

O interior do Brasil é frequentemente associado à vida rural, com atividades agrícolas e uma população simples e tradicional.

A Realidade

O interior é uma região diversificada, com cidades e vilas que possuem uma variedade de atividades, profissões e estilos de vida.

Impacto da Mídia

A mídia frequentemente retrata o interior como um lugar isolado, com pessoas "caipiras" e costumes atrasados, reforçando estereótipos.



Impactos dos Estereótipos

Distorção da Imagem

Generalizações reduzem a riqueza cultural das regiões a um único estereótipo, desvalorizando a diversidade.

Preconceitos

A crença em estereótipos pode levar à discriminação, limitando oportunidades e criando barreiras para o desenvolvimento.

Made with Gamma



Conclusão e Reflexão

1

Desconstruindo Estereótipos

É fundamental buscar conhecer a realidade das diferentes regiões do Brasil, desconstruindo preconceitos e valorizando a riqueza cultural de cada local.

2

Promovendo a Empatia

Compreender as particularidades regionais e respeitar as diferenças é essencial para construir uma sociedade mais justa e inclusiva.

3

Celebrando a Diversidade

A riqueza cultural do Brasil reside na sua diversidade regional. Valorizar as particularidades de cada local é celebrar a nossa identidade nacional.

Made with Gamma

4. CONCLUSÃO

O projeto do sistema de estoque foi desenvolvido com foco em atender às necessidades específicas da APAE, priorizando a organização, controle e segurança dos dados, aliado a um uso eficiente de recursos tecnológicos. A solução integra um banco de dados robusto e funcional com armazenamento em nuvem para garantir a preservação e a integridade das informações, mesmo em situações adversas.

Banco de Dados

O banco de dados foi estruturado para atender de maneira eficiente as operações de estoque, como:

- **Controle de Insumos:** Registro detalhado dos itens armazenados, entradas e saídas, e controle de validade.
- **Rastreamento de Doações:** Monitoramento de contribuições recebidas, associando-as a registros claros e confiáveis.
- **Automação e Consistência:** Uso de triggers e procedures para automatizar validações e manter a consistência dos dados.

Esse modelo foi planejado para reduzir erros manuais e proporcionar uma experiência confiável e intuitiva aos usuários do sistema, garantindo que as operações de rotina sejam realizadas de forma eficiente.

Uso da Nuvem

A integração com a nuvem foi projetada para atender especificamente às necessidades de backup e recuperação de dados, com ênfase em manter custos baixos e eficiência. Com base nos requisitos da APAE, a estratégia de armazenamento prioriza:

1. **Backup Incremental e Agendado:**
 - Backups realizados duas vezes por semana apenas para dados alterados, otimizando o uso de banda e armazenamento.
2. **Armazenamento Econômico:**
 - Utilização de camadas de baixo custo como Amazon Glacier ou Google Coldline para backups de longo prazo.

3. **Política de Retenção:**

- Retenção de versões recentes dos dados, permitindo recuperação rápida, enquanto versões mais antigas podem ser movidas para armazenamento de arquivamento mais barato.

Essas medidas garantem a segurança dos dados com o menor impacto financeiro, considerando as limitações orçamentárias da APAE.

Viabilidade Econômica

O projeto foi desenhado para manter o equilíbrio entre funcionalidade e custo:

- A infraestrutura local é utilizada para o funcionamento diário do sistema, limitando o uso da nuvem ao armazenamento de backup.
- O modelo de "pague pelo uso" da nuvem permite custos proporcionais à quantidade de dados armazenados, sem gastos fixos excessivos.

Conclusão Geral

Com uma estrutura de banco de dados eficiente e uma integração econômica com a nuvem, o sistema de estoque da APAE oferece:

- **Confiabilidade:** Dados organizados e automatizados para evitar erros.
- **Segurança:** Backups regulares e protegidos na nuvem.
- **Custo Acessível:** Uso otimizado de recursos para se alinhar ao orçamento limitado da instituição.

Esse projeto não apenas atende às demandas técnicas, mas também reflete um compromisso com a missão social da APAE, proporcionando ferramentas acessíveis e sustentáveis para melhorar a gestão e maximizar os benefícios às pessoas atendidas pela instituição.

REFERÊNCIAS

DATE, C. J. *Introdução a Sistemas de Bancos de Dados*. 8. ed. São Paulo: Pearson, 2004.

SANTOS, Tiago. *Fundamentos da Computação em Nuvem*. São Paulo: Senac São Paulo, 2020.

SEBESTA, Robert W. *Conceitos de Linguagens de Programação*. 12. ed. São Paulo: Pearson, 2018.

TENENBAUM, Aaron M.; LANGSAM, Yedidyah; AUGENSTEIN, Moshe J. *Estruturas de Dados Usando C*. 2. ed. São Paulo: Pearson Makron Books, 1995.