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

<IDEAL COMPANY CONTABILIDADE>

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

IDEAL COMPANY CONTABILIDADE

MÓDULO COMPUTAÇÃO EM NUVEM

Estrutura de Dados – Prof. Marcelo Ciacco Almeida

Linguagem e Técnicas de Programação – Prof. Nivaldo de Andrade

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

Computação em Nuvem – Prof. Rodrigo Marudi de Oliveira

Projeto de Computação em Nuvem – Profª. Mariângela Martimbianco Santos

Estudantes:

Caic Sant Anna, RA 23000417
Lucas Guimarães Castro Nunes, RA 23000143
Gesiel dos Santos Guirra, RA 23000644
Vitor Alexandre Rocetti Rinke, RA 23000081

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2024

SUMÁRIO

1. INTRODUÇÃO

O principal objetivo deste projeto foi desenvolver uma API eficiente e escalável para a Ideal Company, uma empresa de contabilidade que presta serviços a um grande número de clientes, gerenciando dados financeiros essenciais para suas operações diárias. O sistema visa otimizar a gestão de clientes, faturas e receitas, além de integrar funcionalidades de automação por meio de triggers e stories procedentes. Essa API permitirá à empresa melhorar a precisão e a agilidade dos seus processos contábeis, ao mesmo tempo em que oferece uma base sólida para futuras expansões e integrações com outros sistemas de gerenciamento financeiro.

Nossa abordagem envolve a construção de uma infraestrutura tecnológica moderna, que utiliza o banco de dados MySQL, com a possibilidade de migrar para uma arquitetura de computação em nuvem, visando garantir a escalabilidade e a alta disponibilidade dos dados. Ao adotar essa solução, a Ideal Company poderá realizar a gestão centralizada das informações de seus clientes e operações financeiras de maneira mais eficiente e segura, evitando erros manuais e garantindo a conformidade com os padrões de segurança de dados.

Além disso, este projeto também tem como objetivo explorar conceitos avançados de gerenciamento de banco de dados, como triggers automáticas para controle de faturas e procedimentos armazenados para validação de dados, aplicando técnicas de segurança e compliance para garantir a integridade das informações. A solução proposta permitirá que a empresa reduza custos operacionais, aumente sua capacidade de lidar com um número crescente de clientes e se posicione como uma referência no uso de tecnologia no setor contábil.

Por meio desta iniciativa, almejamos não apenas implementar uma solução inovadora para os desafios operacionais da Ideal Company, mas também fornecer à equipe da empresa uma ferramenta útil que facilite a tomada de decisões estratégicas com base em dados precisos e acessíveis. Ao final do projeto, estaremos entregando uma API capaz de transformar o gerenciamento contábil da empresa, com a flexibilidade de se adaptar a novas demandas e a capacidade de escalar conforme a Ideal Company cresce.

2. DESCRIÇÃO DA EMPRESA

A Ideal Company é uma empresa de contabilidade que tem o propósito de oferecer serviços voltados para a gestão financeira e tributária de empresas de diversos segmentos. A empresa tem como foco fornecer soluções contábeis, desde o processamento de folhas de pagamento, controle de tributos, até a emissão de faturas e organização do fluxo financeiro dos clientes.

Razão Social: Ideal Company Contabilidade
CNPJ: 13.205.035/0002-90
Endereço: Avenida Doutor Durval Nicolau, 2140, Sala 07, Riviera de São João, São João da Boa Vista - SP, CEP 13874-788
Atividade Econômica: Serviços de contabilidade, auditoria e consultoria tributária.

O mercado em que a Ideal Company atua é altamente competitivo, sendo composto por pequenas, médias e grandes empresas que necessitam de serviços especializados para manter suas finanças organizadas e em conformidade com as exigências fiscais. Seus principais produtos e serviços incluem:

- **Gestão Contábil e Fiscal**: Processamento de livros contábeis, escrituração fiscal e entrega de obrigações acessórias.
- **Consultoria Tributária**: Análise e otimização de carga tributária de acordo com as legislações vigentes.
- **Folha de Pagamento e Obrigações Trabalhistas**: Processamento de salários, benefícios e encargos sociais.
- **Emissão de Faturas e Controle de Pagamentos**: Auxílio na emissão de faturas e monitoramento de pagamentos pendentes.

3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

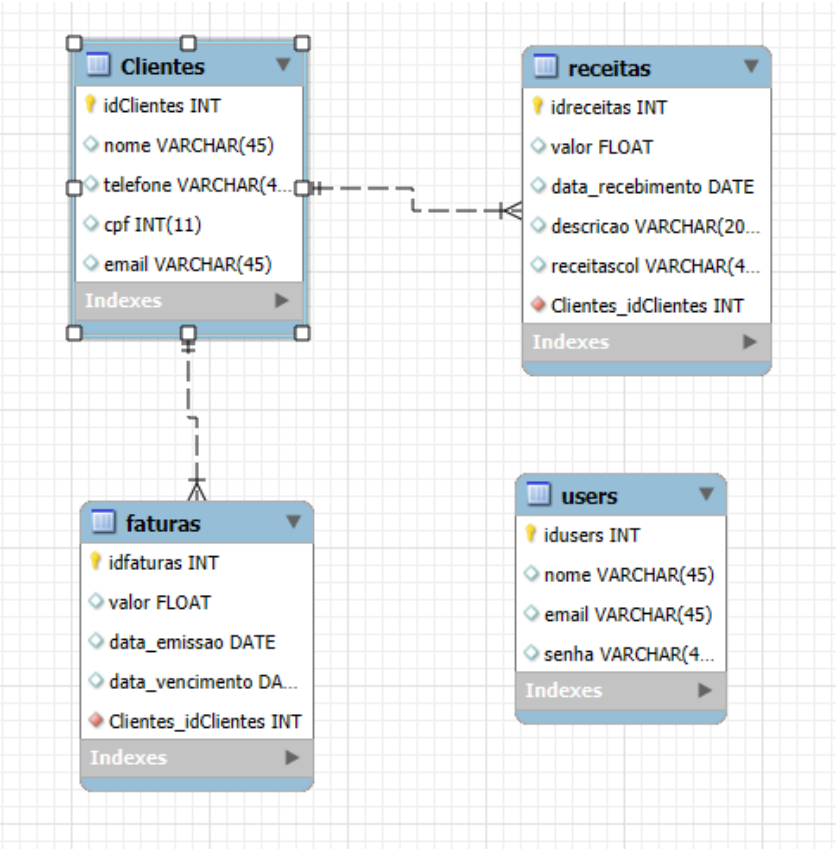
Durante o desenvolvimento do projeto da Ideal Company o objetivo principal foi criar uma solução de banco de dados eficiente que pudesse atender às necessidades da empresa, integrando suas operações financeiras e administrativas. A empresa presta serviços de contabilidade para diversos clientes, e a gestão precisa de um controle rigoroso das informações relacionadas a clientes, faturas e receitas.

Como parte da equipe de desenvolvimento, a solução que propusemos foi baseada em um banco de dados relacional, utilizando MySQL como a tecnologia principal. A escolha do banco de dados relacional se deu pela necessidade de lidar com grandes volumes de dados estruturados, garantir consistência nas transações, e utilizar mecanismos que facilitam a integridade referencial, algo importante em um cenário contábil. A Ideal Company lida com informações financeiras críticas, como o controle de pagamentos e a geração de relatórios, o que torna a consistência e integridade dos dados algo importante.

Embora o banco de dados local tenha sido a escolha inicial, também consideramos a possibilidade de migrar o sistema para uma solução em nuvem, utilizando plataformas como Amazon RDS, Google Cloud SQL ou Azure SQL Database. Isso proporciona maior escalabilidade e segurança, com backups automáticos e redundância de dados, o que reduziria os riscos operacionais da empresa. Além disso, a arquitetura em nuvem facilitaria o acesso remoto e a integração com outras ferramentas e sistemas utilizados pela empresa.

3.1.1 MODELO LÓGICO

O Projeto Lógico do banco de dados da Ideal traz as principais necessidades da empresa, através desses problemas esquematizamos um planejamento através do mapeamento lógico do workbench e de rascunhos no papel, permitindo com que a equipe organizasse melhor nossas ideias e melhorando o resultado a longo prazo permitindo uma gestão mais eficiente dos dados de clientes, faturas, receitas e usuários do sistema. Esta fase lógica definiu as entidades gerais do negócio, ao modelo físico, detalhando a implementação de tabelas, relacionamentos e regras de integridade que garantiram a consistência dos dados.



Fonte: Autores

Para atender às necessidades da Ideal, identificamos quatro tabelas principais:

- Clientes: Armazena os dados pessoais e de contato dos clientes da empresa.
- Faturas: Registra todas as faturas emitidas para os clientes, incluindo informações de valor, data de vencimento e status.
- Receitas: Mantém o registro de todas as receitas financeiras da empresa.
- Users: Gerencia os dados dos usuários autorizados a acessar o sistema, focando na segurança e controle de acesso.

No Projeto Lógico, também definimos relacionamentos que refletem as interações entre as entidades. Esses relacionamentos são implementados através de chaves estrangeiras:

- Clientes - Faturas: Um cliente pode ter várias faturas, mas cada fatura está associada a apenas um cliente. Esse relacionamento é representado pela coluna "id_cliente" em Faturas, que é uma chave estrangeira referenciando "id_cliente" em Clientes.

Esse mapeamento de chaves estrangeiras assegura que uma fatura só existe se houver um cliente associado, preservando a integridade referencial entre os dados.

As regras de integridade foram implementadas para garantir que o banco de dados seja confiável e consistente. As principais restrições aplicadas incluem:

- Chaves Primárias em cada tabela (por exemplo, "id_cliente" em Clientes), o que assegura que cada registro seja único e identificável.
- Chaves Estrangeiras para manter a integridade dos relacionamentos, como "id_cliente" em Faturas, que deve corresponder a um "id_cliente" existente em Clientes.
- Unicidade em colunas como "email" e "username" na tabela Users, evitando duplicação de registros de usuários.

Para facilitar o acesso a informações frequentes, o projeto inclui a criação de views para consultas recorrentes. Exemplo:

- Uma view que combina informações de Clientes e Faturas, facilitando a visualização de faturas pendentes e seu cliente associado sem a necessidade de consultas complexas em várias tabelas.

Para automatizar processos e simplificar operações, foram planejados procedures e triggers:

- Procedures para inserções, atualizações e exclusões nas tabelas principais, evitando erros manuais e permitindo o uso de lógica de negócios no banco de dados.
- Triggers podem ser criados para ações automáticas, como atualizar o status de uma fatura ao registrar o pagamento, ou bloquear exclusões de clientes que ainda tenham faturas pendentes.

Com o Projeto Lógico concluído, tivemos uma visão detalhada de como o banco de dados da Ideal Company deveria ser estruturado para garantir eficiência, integridade e suporte às necessidades do negócio. Esse esquema serviu como guia para a implementação do banco de dados no MySQL Workbench, garantindo que o modelo físico esteja alinhado com os requisitos lógicos definidos aqui.

3.1.2 MODELO FÍSICO

O Projeto Físico representa a etapa final de implementação do sistema de armazenamento de dados da Ideal. Nessa fase, foram definidos todos os aspectos técnicos e operacionais do banco de dados que sustentam as operações de contabilidade e gestão financeira da empresa..

A estrutura do banco foi desenhada para atender às necessidades da Ideal Company, com foco nas seguintes tabelas principais:

1. **Clientes**: Armazena dados dos clientes da empresa, incluindo informações pessoais como nome, e-mail, telefone e CPF. Cada cliente possui um identificador único que serve como referência para outras tabelas, como Faturas e Receitas.
2. **Faturas**: Responsável por armazenar as informações sobre as faturas emitidas para os clientes. Esta tabela contém colunas como valor da fatura, data de emissão, data de vencimento e o status de pagamento (Pendente, Pago, Cancelado). O relacionamento com a tabela Clientes é estabelecido por uma chave estrangeira que associa cada fatura ao cliente correspondente.
3. **Receitas**: Armazena os pagamentos recebidos pela empresa. Cada receita está vinculada a uma fatura por meio da coluna cliente_id, garantindo que o valor pago seja associado à fatura correta.

A criação das tabelas, chaves estrangeiras, e triggers foi implementada através de scripts SQL, que automatizam o processo de construção do banco de dados. Abaixo estão os principais objetos e s funcionalidade no sistema:

- **Tabelas**: Definidas com suas colunas, tipos de dados e chaves primárias. As tabelas possuem chaves estrangeiras para garantir a integridade referencial entre clientes, faturas e receitas.
- **Triggers**: Como parte do projeto físico, duas triggers principais foram implementadas:
    - AtualizaStatusFatura: Atualiza o status de uma fatura para "Pago" automaticamente sempre que um pagamento é registrado na tabela Receitas.
    - VerificaCPFUnico: Assegura que o CPF do cliente seja único, impedindo a inserção de clientes com CPF duplicado.
- **Stored Procedures**: As stored procedures foram criadas para facilitar a inserção, atualização e exclusão de dados nas tabelas, além de gerar relatórios financeiros. Exemplos incluem a AddCliente, Add Fatura e Relatório ReceitasPorCliente, que automatizam as operações principais de gerenciamento dos dados da empresa.

O controle de acesso aos dados é essencial para garantir a segurança e a confidencialidade das informações sensíveis da Ideal Company. No projeto físico, foram atribuídos níveis de permissão diferentes para os usuários que interagem com o sistema. Administradores têm acesso completo, enquanto usuários com funções específicas podem ser limitados a realizar apenas consultas ou determinadas inserções.

A definição do Projeto Físico foi baseada em uma abordagem estruturada e detalhada, focada em garantir a integridade, a performance e a segurança do banco de dados. A implementação eficiente dos objetos no banco de dados da Ideal é a chave para o sucesso das operações contábeis, permitindo que as tarefas diárias sejam realizadas de maneira automática e precisa.

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

No projeto foi utilizado conceitos de programação orientada a objetos pois o mesmo facilita a manutenção e escalabilidade do código, práticas de clean code como convenções assim também facilitando a manutenção do código segundo o site balta.io “Se você começou agora em um projeto ou acabaram de definir suas convenções, siga-as! Se utilizam por exemplo constantes em maiúsculo, enumeradores com E como prefixo, não importa! Siga sempre os padrões do projeto.”, typescript foi utilizado apesar de ser menos flexível que javascript a longo prazo os prós superam os contras.

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

A API segue os conceitos RESTful assim facilitando a escalabilidade e a manutenção da API como diz o site amazon web server “Os serviços da Web RESTful permitem a separação total do cliente do servidor. Eles simplificam e desacoplam vários componentes do servidor para que cada parte possa evoluir independentemente.”, a mesma possui sistema de autenticação e autorização o que garante a segurança da API e concede acesso a certas funcionalidades apenas aos autorizados.

3.2.2 FRONT-END

Focamos na criação de uma interface de usuário que fosse intuitiva e fácil de navegar. Fizemos uso do Materialize que contribuiu para um design moderno além agilizar o processo de estilização e ajudar o usuário a interagir melhor com o sistema. A experiência do usuário foi pensada para ser simples e eficiente, com uma navegação clara e funcionalidades organizadas em menus e opções acessíveis.

Após a autenticação de login, o usuário é direcionado para o componente "Home", onde pode acessar várias funcionalidades. O layout é dividido em duas seções principais:

Menu Lateral (Esquerdo): Onde o usuário pode selecionar opções como visualizar faturas, receitas, clientes, configurações e ajuda.

Área Principal (Direita): Aqui, é exibido um componente correspondente à opção selecionada, como visualização e manipulação de faturas, receitas e clientes. A interação com esses componentes permite a realização de operações de CRUD,

O consumo da API da Ideal Company foi realizado por meio do uso do HttpClientModule, que importamos no módulo principal para habilitar as requisições HTTP em toda a aplicação. Para cada uma das operações de CRUD, criamos métodos específicos que interagem diretamente com os endpoints da API.

Por exemplo:

- **GET**: para listar todos os clientes ou faturas, enviamos uma requisição GET ao endpoint específico e exibimos os dados retornados na área principal da tela.

- **POST**: ao cadastrar um novo cliente ou fatura, enviamos uma requisição POST com os dados inseridos pelo usuário nos campos de formulário.
- **PUT**: para editar registros existentes, utilizamos o método PUT, enviando o ID do registro e os dados atualizados.
- **DELETE**: ao excluir um cliente ou uma fatura, enviamos uma requisição DELETE, removendo o registro do banco de dados.

Para otimizar essas requisições e evitar redundâncias, criamos um service dedicado para cada entidade, como ClienteService e FaturaService. Esses serviços encapsulam toda a lógica de comunicação com a API e fornecem métodos reutilizáveis, facilitando a manutenção.

Além do HttpClientModule, fizemos uso dos seguintes módulos e dependências:

- **FormsModule** e **ReactiveFormsModule**: permitiram a criação e manipulação de formulários de entrada, essenciais para o CRUD de clientes e faturas.

- **Materialize CSS**: auxiliou na estilização dos formulários, botões e tabelas, garantindo que a interface ficasse visualmente agradável e consistente com o design da aplicação.

- **CommonModule**: importado em cada componente standalone para acesso a diretivas comuns como ngIf e ngFor.

Além das funcionalidades de CRUD, implementamos alguns detalhes que melhoram a experiência do usuário:

- **Validação de Formulários**: incluímos validações dinâmicas, onde campos obrigatórios e formatos de entrada são verificados em tempo real, evitando que dados incorretos sejam enviados à API.
- **Feedback de Usuário**: para cada operação bem-sucedida (como salvar ou excluir um registro), exibimos mensagens de confirmação. Em caso de erro, mensagens apropriadas orientam o usuário sobre o problema.

Também optamos por componentes standalone para cada página de CRUD (ex.: ClienteComponent e FaturaComponent). A vantagem dessa abordagem foi a independência entre os componentes, permitindo que cada um importasse apenas os módulos e serviços necessários para sua funcionalidade específica. Essa estrutura modular reduz a complexidade do código, melhorando o desempenho da aplicação e a clareza do código.

Com uma abordagem modular e standalone, conseguimos desenvolver uma aplicação Angular eficiente e de fácil manutenção, garantindo uma experiência fluida para o usuário. A integração com a API da Ideal permitiu que o sistema fosse funcional e que cada tela de CRUD fosse intuitiva e direta, auxiliando o usuário nas tarefas diárias de gestão de clientes e faturas.

Utilizamos boas práticas de desenvolvimento com Angular, abordagens RESTful para o consumo da API e uma estrutura de front-end baseada na modularidade e reutilização.

3.3 COMPUTAÇÃO EM NUVEM

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

Com a utilização da nuvem a empresa poderá usufruir dos benefícios do projeto sem ter que arcar com os custos de uma infraestrutura própria como diz o site SYDLE “A tecnologia em nuvem é a opção que o usuário tem para armazenar arquivos, servidores e softwares em um ambiente virtual”, assim reduzindo custos e aumentando a eficiência.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

A computação em nuvem traz ao projeto uma maior flexibilidade e eficiência pois não necessita que tenha um computador com um grande poder computacional para que o projeto funcione e a própria nuvem cuida da manutenção segundo o site Google Cloud “A nuvem oferece mais flexibilidade e confiabilidade, desempenho e eficiência melhorados e ajuda a reduzir os custos de TI”.

3.3.3 VANTAGENS DA CLOUD COMPUTING

Com a computação em nuvem possui-se uma alta disponibilidade que costuma ser um problema recorrente de projetos menores e também possui-se uma boa preservação de dados como diz o site Google Cloud “Os provedores de nuvem oferecem recursos de backup e recuperação de desastres. Armazenar dados na nuvem em vez de localmente pode ajudar a evitar a perda de dados em caso de emergência”.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

O projeto utiliza os serviços de Iaas como o EC2 que com o uso do serviço não será necessário ter uma infraestrutura prévia para que o projeto funciona como diz o site Digital Innovation One (DIO) “O uso do Amazon EC2 elimina sua necessidade de investir antecipadamente em hardware”, o projeto também utiliza os serviço S3 para que é totalmente compatível com o EC2 e o serviço amazon simple notification service que permite enviar mensagens por sms.

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

O provedor escolhido é a AWS porque já está no mercado há bastante tempo e é confiável como diz o site Amazon Web Services “Com a AWS, você usufrui da infraestrutura de computação global escalável, confiável e segura”, flexível e por familiaridade com os serviços.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

O projeto funciona em uma instância do serviço EC2 da AWS que possui balanceamento de carga, caso a instância atinja seu limite o balanceamento de carga distribui o tráfego entre as instâncias são ligadas quando necessário segundo o site cloudflare “O balanceamento de carga distribui o tráfego entre vários servidores para reduzir a latência e melhorar a disponibilidade e a confiabilidade do servidor”.

3.3.7 GOOGLE CLOUD ou AWS

O projeto utiliza as funcionalidades do serviço de cloud da AWS como gerenciar instâncias, o banco de dados. O motivo de o provedor escolhido foi por sua confiabilidade e é fácil de usar, segundo o site Amazon Web Services “A AWS é projetada para permitir que provedores de aplicativos, ISVs e fornecedores hospedem seus aplicativos com rapidez e segurança”, não houve necessidade de usar os serviços do Google Cloud.

3.4 ESTRUTURA DE DADOS

3.4.1 LEVANTAMENTO DE REQUISITOS

O projeto utiliza estrutura de lista no banco de dados e o mesmo possui uma baixa frequência de inserção, remoção, atualização e busca de dados o que faz que o requisito do sistema seja baixo, na API também se utiliza estrutura de lista, consumo de memória é baixo e o tempo de resposta também é baixo que é importante para que o usuário tenha uma boa experiência, segundo o site phoenixNap global IT services “O tempo de resposta é crucial porque impacta diretamente a experiência do usuário, o desempenho do sistema e a eficiência operacional”.

3.4.2 VALIDAÇÃO DOS REQUISITOS

Como parte do desenvolvimento do projeto Ideal Company, realizamos a migração do ambiente de desenvolvimento local para a nuvem da Amazon Web Services. foco da implementação na AWS está na criação de uma infraestrutura escalável, com capacidade de distribuir a carga de maneira eficiente e garantir a integridade dos dados.

- **Serviço Utilizado**: Amazon EC2 foi escolhido para hospedar a API, utilizando instâncias configuradas para escalar de acordo com a demanda de tráfego.

- **Configuração da Escalabilidade**: Para assegurar que a API possa lidar com volumes variáveis de requisições, o Auto Scaling da AWS foi configurado. Esse recurso ajusta

automaticamente a quantidade de instâncias EC2 em execução, aumentando-as em momentos de alta demanda e reduzindo-as quando o tráfego diminui.

- **Escalabilidade e Disponibilidade**: O banco de dados foi migrado para o Amazon RDS, que oferece suporte a réplicas de leitura e failover automático. Assim, o RDS pode lidar com operações de leitura intensiva sem comprometer o desempenho, além de garantir alta disponibilidade.

O banco de dados foi ajustado para incluir índices otimizados nas tabelas de maior acesso, como Clientes e Faturas, acelerando consultas frequentes e melhorando o tempo de resposta em alta demanda.

3.5. CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS

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

3.5.1 ENFRENTANDO ESTEREÓTIPOS

Neste tema , aprendemos que os estereótipos afetam profundamente a maneira como nos relacionamos com as pessoas e interpretamos o mundo ao nosso redor. Ao sintetizar os quatro tópicos, ficou claro para nós como esses conceitos estão presentes em diversas situações do dia a dia.

1. **Estereótipo e Convívio Social**: Os estereótipos são generalizações que impactam negativamente o convívio entre grupos. Por exemplo, no ambiente de trabalho, é comum ver pessoas assumindo que mulheres são menos capazes em certas áreas como tecnologia, o que cria barreiras desnecessárias e prejudica o relacionamento profissional. Esses estereótipos não só limitam as oportunidades, mas também reforçam preconceitos que dificultam a convivência social.

2. **Estereótipo e Representação**: A mídia é uma das principais fontes de construção de estereótipos. Muitas vezes, grupos sociais são representados de forma distorcida em filmes e novelas. Um exemplo que me veio à mente são as novelas que retratam moradores do interior como "caipiras" ou simplórios, ignorando a complexidade e a modernidade das vidas dessas pessoas. Essa visão equivocada influencia a forma como enxergamos quem não conhecemos profundamente, criando uma imagem estereotipada dessas pessoas.

3. **Troco Likes: A Idealização da Vida na Internet**: Refletir sobre a idealização da vida nas redes sociais nos fez perceber o quanto essa cultura de "likes" contribui para reforçar estereótipos de sucesso e felicidade. Vemos influenciadores que, muitas vezes, só mostram uma versão perfeita de suas vidas, o que cria expectativas irreais sobre como devemos viver. Isso acaba gerando uma pressão enorme, especialmente entre os jovens, que sentem a necessidade de corresponder a esses padrões inatingíveis.

4. **Convivendo com a Diferença**: Aprendemos também sobre a importância de respeitar as diferenças culturais, sociais e individuais. Um exemplo prático foi a experiência de conviver com pessoas de diferentes regiões do Brasil na faculdade. Ao ouvir suas histórias, percebemos como cada cultura tem seu valor e que, ao invés de reforçar estereótipos, devemos celebrar essas diferenças. Isso nos fez pensar em como atitudes simples, como evitar piadas sobre a origem das pessoas, podem ajudar a construir um ambiente mais inclusivo e respeitoso.

Em resumo, essa reflexão nos mostrou como os estereótipos afetam nosso cotidiano de diversas maneiras. Além disso, entendi que, ao reconhecer esses preconceitos e agir de maneira mais empática, podemos contribuir para um convívio social mais saudável e justo.

3.5.2 ESTUDANTES NA PRÁTICA

fonte:autoral

Neste banner, quisemos explorar a ideia de que a identidade das pessoas que vivem no interior vai muito além dos estereótipos. Muitas vezes, quem mora nessas regiões é visto de forma simplista, sendo rotulado como 'caipira', o que não reflete a complexidade cultural e a diversidade que realmente existe. A frase 'Caipira ou Cidadão?' foi escolhida para provocar uma reflexão sobre como as pessoas do interior são vistas e como, na verdade, elas desempenham papeis diversos na sociedade, seja no campo ou nas cidades.

A imagem das duas crianças caipiras serve para lembrar que, mesmo nas regiões rurais, existe uma riqueza de tradições e uma forma de vida que deve ser respeitada e valorizada. Queremos mostrar que essas crianças podem ser parte de uma cultura que é moderna, conectada, e, ao mesmo tempo, profundamente enraizada em suas tradições. Nosso objetivo com o banner é quebrar preconceitos e incentivar uma reflexão mais ampla sobre as diferenças culturais no Brasil, desafiando as visões superficiais que muitas vezes são impostas

4. CONCLUSÃO

Ao longo deste projeto integrado, nossa equipe documentou e implementou o sistema de gerenciamento de dados para a Ideal Company, utilizando uma combinação de ferramentas e tecnologias que englobam front-end, back-end e banco de dados. Esse processo nos permitiu aplicar conceitos fundamentais de Estruturas de Dados, Modelagem de Banco de Dados, lógica de programação com TypeScript, e a criação de uma interface de usuário com Angular.

O projeto começou com a definição da estrutura do banco de dados, onde foi necessário projetar tabelas para Clientes, Faturas, Receitas e Usuários. Durante essa fase, encontramos desafios em equilibrar a simplicidade do modelo com a necessidade de funcionalidades complexas, como controle de faturas e acesso seguro de usuários. Decidimos por uma modelagem relacional no MySQL, o que nos deu a vantagem de consistência nos dados e uso de consultas SQL. A criação de procedures e triggers para manipulação de pagamentos e validação de dados adicionou uma camada de automação útil, mas exigiu atenção extra para entender o impacto de cada operação nas diferentes tabelas, principalmente em casos de exclusão e atualização de registros.

Na implementação da API em Node.js, optamos por usar o TypeScript para melhorar a tipagem de dados, o que facilitou o trabalho em equipe e a manutenção do código. Essa escolha ajudou a reduzir erros durante o desenvolvimento, especialmente quando os dados trafegavam entre o front-end e o back-end. A integração com o MySQL foi feita utilizando Sequelize, um ORM que simplificou as operações no banco de dados, permitindo maior foco na lógica da aplicação e nas validações.

Entre as dificuldades encontradas, destacamos a necessidade de normalização dos dados e a configuração de diferentes ambientes de desenvolvimento para garantir que os dados fossem manipulados corretamente entre as tabelas, principalmente quando novas regras de negócios surgiam. A criação de rotas RESTful foi relativamente tranquila, mas algumas operações mais complexas, como a geração de relatórios e a consulta de faturas, exigiram ló

O uso de Angular no front-end foi um ponto de aprendizado significativo. A escolha dessa tecnologia permitiu-nos criar uma interface dinâmica e modular, mas também apresentou desafios, especialmente para aqueles na equipe com pouca experiência em desenvolvimento front-end. A estrutura modular do Angular ajudou na organização do

projeto, com cada componente representando uma funcionalidade específica, como a listagem de clientes e o gerenciamento de faturas.

Uma das principais dificuldades foi compreender o fluxo de dados no Angular e a comunicação entre componentes pai e filho. A configuração de **Services** para gerenciar chamadas à API exigiu um entendimento cuidadoso dos ciclos de vida dos componentes e da estrutura de observações do Angular. Apesar dos desafios, o Angular ofereceu uma boa base para futuras expansões, e a flexibilidade dos componentes será útil se quisermos incorporar novas funcionalidades.

A integração entre as diferentes camadas – banco de dados, back-end e front-end – foi uma fase desafiadora, especialmente ao testar o fluxo completo de dados. O uso de ferramentas como Postman facilitou os testes das rotas da API, permitindo simular as requisições antes da implementação no front-end. A verificação de erros e validação dos dados foi uma das facilidades mais notáveis do TypeScript, que permitiu capturar problemas durante a compilação e não apenas em tempo de execução.

As principais dificuldades foram encontradas na integração de tecnologias diferentes e na configuração do ambiente de desenvolvimento. Adicionalmente, compreender as práticas recomendadas de cada tecnologia levou tempo, mas ao final proporcionou uma base sólida. Como facilidades, destacamos o uso do ORM Sequelize, que simplificou as consultas no banco de dados, e o TypeScript, que trouxe clareza e segurança ao código. No front-end, o Angular ofereceu uma estrutura poderosa e facilitou o desenvolvimento de uma interface escalável.

Esse projeto nos permitiu trabalhar como uma equipe de desenvolvimento completa, enfrentando desafios reais do ciclo de vida de uma aplicação. A experiência de desenvolver uma solução integrada para a Ideal Company, aplicando Estruturas de Dados, técnicas de modelagem, e construção de APIs e interface de usuário, trouxe um aprendizado valioso e uma compreensão mais aprofundada de como diferentes tecnologias se complementam em um sistema completo.

REFERÊNCIAS

W3Schools - MySQL Tutorial. Disponível em: https://www.w3schools.com/mysql/. Acesso em 21 de outubro de 2024.

MDN Web Docs - JavaScript Guide. Disponível em: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide. Acesso em 21 de outubro de 2024.

Angular - Tour of Heroes Tutorial. Disponível em: https://angular.io/tutorial. Acesso em 21 de outubro de 2024.

freeCodeCamp.org - Node.js & Express Full Course for Beginners. Disponível em: https://www.youtube.com/watch?v=Oe421EPjeBE. Acesso em 21 de outubro de 2024.

TypeScript Documentation - Handbook. Disponível em: https://www.typescriptlang.org/docs/handbook/intro.html. Acesso em 21 de outubro de 2024.

Traversy Media - MySQL Crash Course. Disponível em: https://www.youtube.com/watch?v=9ylj9NR0Lcg. Acesso em 21 de outubro de 2024.

Codecademy - Learn JavaScript. Disponível em: https://www.codecademy.com/learn/introduction-to-javascript. Acesso em 21 de outubro de 2024.

Node.js Documentation. Disponível em: https://nodejs.org/en/docs/. Acesso em 21 de outubro de 2024.

Angular Documentation. Disponível em: https://angular.io/docs. Acesso em 21 de outubro de 2024.

Traversy Media - TypeScript Crash Course. Disponível em: https://www.youtube.com/watch?v=BCg4U1FzODs.
Acesso em 21 de outubro de 2024.

Amazon Web Services - O que é uma API RESTful? Disponível em: https://aws.amazon.com/pt/what-is/restful-api/.
Acesso em 21 de outubro de 2024.

Amazon Web Services - Benefícios do Hospedagem de Aplicações. Disponível em: https://aws.amazon.com/pt/application-hosting/benefits/.
Acesso em 5 de novembro de 2024.

Balta.io - Clean Code. Disponível em: https://balta.io/artigos/clean-code.
Acesso em 21 de outubro de 2024.

Cloudflare - Load Balancing. Disponível em: https://www.cloudflare.com/pt-br/learning/performance/cloud-load-balancing-lbaas/.
Acesso em 6 de novembro de 2024.

Digital Innovation One (DIO) - O que é Amazon EC2?. Disponível em: https://www.dio.me/articles/o-que-e-amazon-ec2.
Acesso em 6 de novembro de 2024.

phoenixNap global IT services - O que é tempo de resposta?. Disponível em: https://phoenixnap.pt/gloss%C3%A1rio/tempo-de-resposta acesso em 8 de novembro de 2024.

Google Cloud - Vantagens da Computação em Nuvem. Disponível em: https://cloud.google.com/learn/advantages-of-cloud-computing?hl=pt-BR.
Acesso em 5 de novembro de 2024.

SYDLE - Vantagens da Computação em Nuvem. Disponível em: https://www.sydle.com/br/blog/vantagens-computacao-em-nuvem-6155bf940e94dd1b0f13489e.
Acesso em 2 de outubro de 2024.