UNIFEOB
ESCOLA DE NEGÓCIOS

2023

PROJETO INTEGRADO

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

DESENVOLVIMENTO DE SIMULADORES ECONÔMICOS PARA LEIGOS

<UNIFEOB>

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2023

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

DESENVOLVIMENTO DE SIMULADORES ECONÔMICOS PARA LEIGOS

PRECISO

MÓDULO COMPUTAÇÃO EM NUVEM

Estrutura de Dados – Prof. Mauro Glória

Linguagem e Técnicas de Programação – Prof. Marcelo Ciacco de Almeida

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:

Arthur Tavares de Paula Valim, RA 22000357
Christian Mello Teio, RA 22000730
Geovana Neuberger Sorg, RA 22001825
João Vitor da Silva, RA 22000871
Leonardo Santello Garino, RA 22000048
Sophia Vilela Lopes, RA 22000211
Thainari Gabriele Gonçalves, RA 22000445

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2023

SUMÁRIO

1. INTRODUÇÃO

O desenvolvimento deste projeto, ocorreu a partir da premissa de ajudar e colaborar para o conhecimento básico em finanças e cálculos primários sobre juros e investimentos. Todo pensamento na direção de auxiliar de forma primária, iniciantes que precisam de suporte para começar sua vida financeira.

Através dos dizeres acima, o intuito é direcionar e orientar, tendo em vista a real necessidade desse aprendizado, pois, entender e saber gerir seu próprio dinheiro é de suma importância para um vida financeira saudável. Quanto antes adquirir este conhecimento melhor.

O site foi construído por meio dos conhecimentos adquiridos principalmente no quarto módulo dos cursos de Ciência da Computação e Análise e Desenvolvimento de Sistemas, como Banco de dados em MYSQL, hospedagem em nuvem, linguagens de programação JavaScript, Node e PHP, além de boas práticas de programação e estrutura de dados. Todos estes assuntos, serão explanados de forma mais ampla no decorrer deste documento.

Toda estrutura foi idealizada para ser transmitida de maneira didática e de fácil compreensão, sem exigir nenhum conhecimento prévio. Seu conceito pode ser resumido em: A matemática é precisa, ou seja, não há um meio termo para esta questão, somente certo ou errado, de mesmo modo, a matemática é precisa na vida de todos. Por esta razão, colocou-se o nome de *Preciso* no site.

2. DESCRIÇÃO DA EMPRESA

A empresa alvo do projeto tem razão social com a Fundação de Ensino Octávio Bastos, mais conhecida como UNIFEOB, portadora do CNPJ:59.764.555/0001-52 de São João da Boa Vista — SP localizada na AV. Doutor Octávio da Silva Bastos, 2439 — Jardim Nova São João, Campus Mantiqueira 13.874-149.

Aberta em 23/08/1968 a Fundação de Ensino Octávio Bastos é uma instituição de natureza jurídica de fundação privada, sem fins lucrativos, a primeira universidade de São João da Boa Vista deste tipo.

Tem como atividade principal a Educação superior graduação. Suas atividades secundárias são: educação superior, pós-graduação e extensão, educação profissional de nível técnico, educação profissional de nível tecnológico, pesquisa e desenvolvimento experimental em ciências físicas e naturais, pesquisa e desenvolvimento experimental em ciências sociais/humanas e atividades veterinárias.

3. PROJETO INTEGRADO

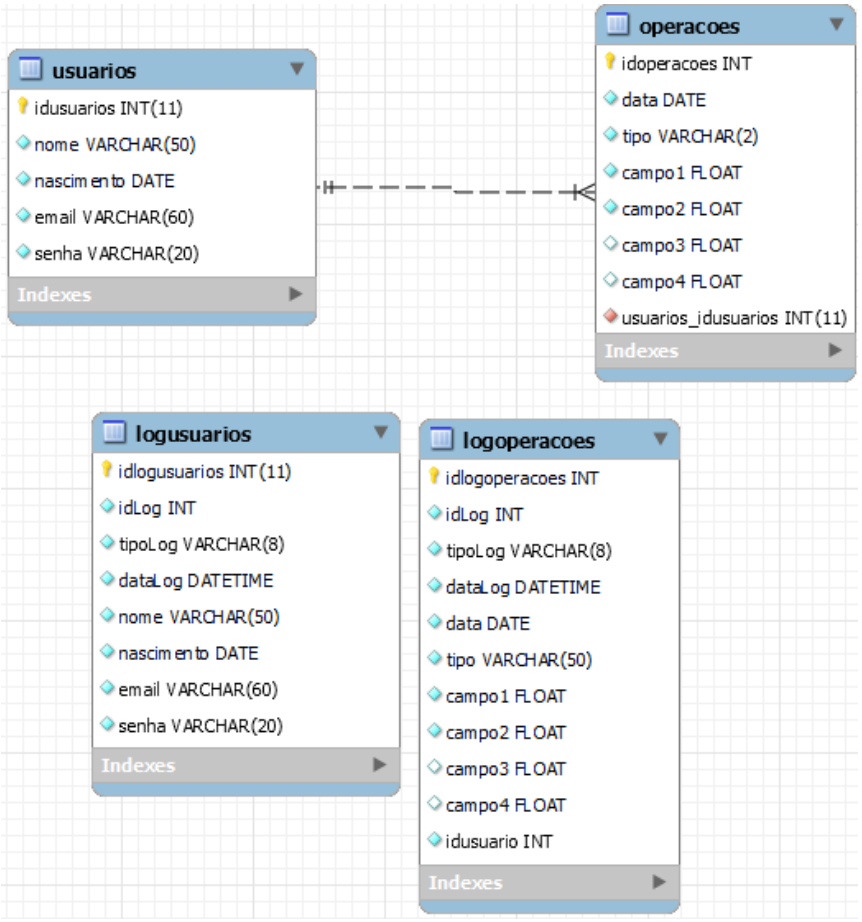
O Projeto Integrado é uma prática pedagógica de interdisciplinaridade que conecta os assuntos abordados em sala de aula durante nosso curso, facilitando a comunicação entre a teoria aprendida em sala de aula e nossa prática profissional. Portanto, diante deste projeto proposto, tentamos atender demandas e interesses da comunidade e do mercado local em uma dinâmica colaborativa. Constituindo-se, pois, em uma proposta de ensino interdisciplinar. A realização é uma exigência disciplinar e propõe o nosso desenvolvimento juntamente com o trabalho em equipe.

3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

O banco de dados é uma parte crucial dentro de um projeto, nele é guardado todas as informações pertinentes de um sistema. Através do banco de dados é possível modificar informações, ler, visualizar e se necessário deletar. Singularmente optamos em fazê-lo com a linguagem de programação MySQL, por meio da UI WorkBench, pois, entendemos que o modelo relacional contido nestas ferramentas são mais abrangentes. Contudo, utilizamos as quatro operações básicas do CRUD para interagir com dados, elas foram implantadas através dos stored procedures.

3.1.1 MODELO LÓGICO

O modelo lógico é uma parte importante tanto da documentação como do processo de criação de um banco de dados, através dele podemos visualizar de forma fácil as entidades e as suas relações, bem como as tabelas, regras e as características das colunas e métodos. O modelo lógico é um passo vital na elaboração de um banco de dados e a sua manutenção. Nosso banco contém duas tabelas principais (usuários e operações) e duas de suporte, que são os logs das principais. As tabelas principais interagem com um relacionamento de um para muitos, onde um usuários pode ter várias operações, e as tabelas de log são atualizadas através dos triggers. Abaixo está o modelo lógico do nosso banco:



3.1.2 MODELO FÍSICO

Os modelos físicos são a forma mais completa de um banco de dados, nele contém todas as informações de um modelo lógico. Seus diferenciais são os detalhes técnicos, pois ele especifica cada especificidade que conterá no banco final, entre esses detalhes temos, chave primária, chave estrangeira, stored procedures, triggers, funções e visões.

- Stored Procedures:

Nós criamos stored procedures que são responsáveis por fazer o (CRUD) Create, Read, Update e Delete nas tabelas de usuários e operação do banco. Pode-se visualizar esses procedures nos anexos – 1.

- Function:

A Function do banco é responsável por fazer as operações matemáticas pertinentes ao projeto, ela recebe qual operação será realizada, e para cada qual, executa o cálculo pertinente, sendo eles: juros simples, juros compostos, investimentos e porcentagem. Pode-se visualizar essa function no anexo – 2.

- Triggers:

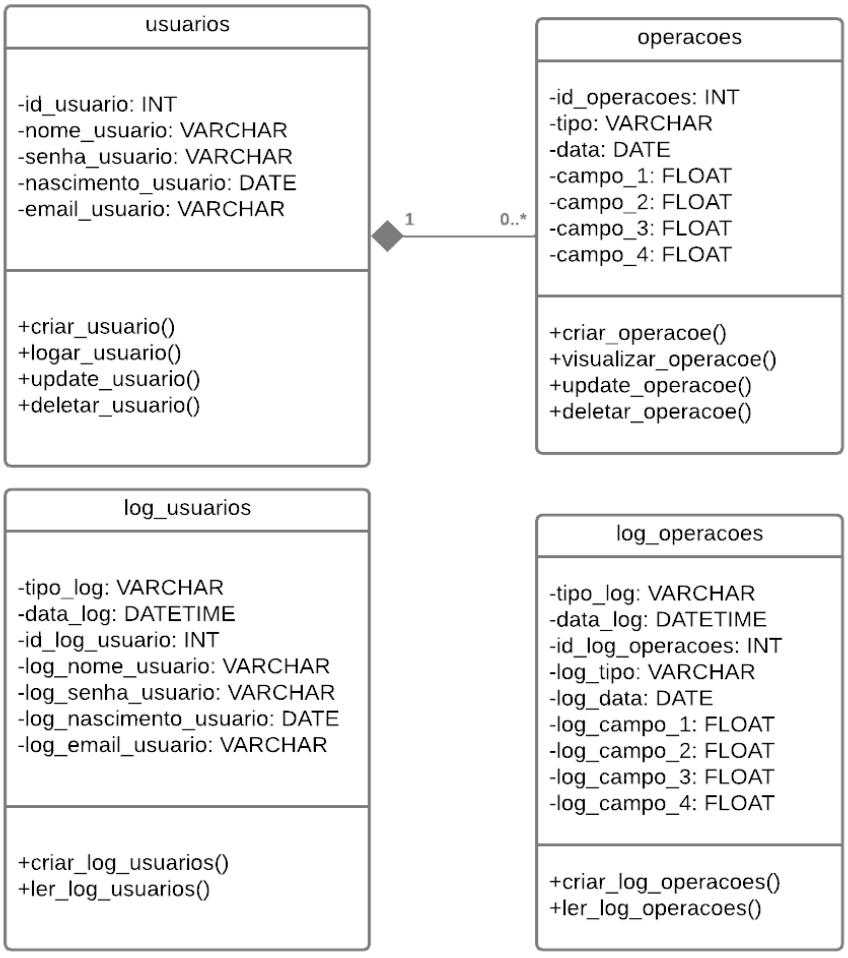
As triggers são responsáveis por guardar todas as mudanças feitas nas tabelas principais (usuários e operações). Ele guarda todos os dados e a data de mudança nas tabelas de logs respectivas. Pode-se visualizar esses triggers nos anexos – 3.

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

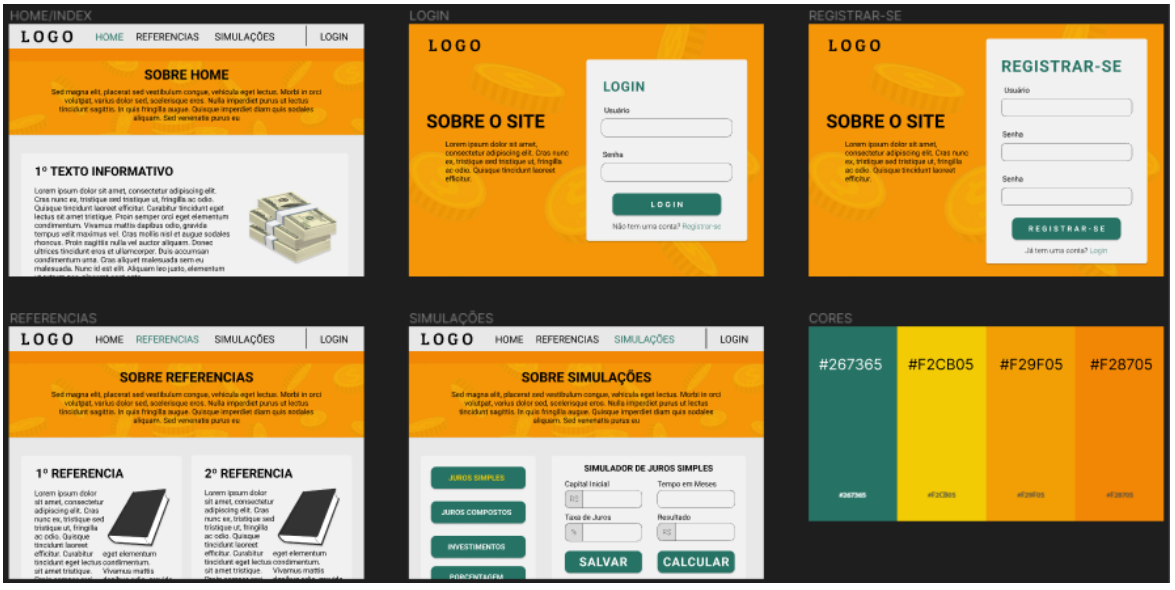
A engenharia de um software é um abordagem que visa criar sistemas de alta qualidade e fácil manutenção, para atingir esse objetivo não basta criar o software, deve-se documenta-lo para melhorar seu desenvolvimento e sua manutenção, essa documentação deve ser capaz de representar o sistema em um nível de alta abstração, para isso diagramas e protótipos foram criados.

3.2.1 PROTOTIPAÇÃO

A prototipação é uma etapa onde a partir dos requisitos e escopo um modelo visual não funcional do projeto é criado, o protótipo será usado como base para a criação do produto final, por isso é importante que todas as partes interessadas estejam satisfeitas com o protótipo. Um bom protótipo é importante para alinhar tanto a equipe no conceito visual e técnico do projeto, desta forma vários integrantes podem trabalhar separadamente para criar o produto.



(Diagrama de classes feito para ilustrar funcionamento do backend)



(Protótipo visual feito na plataforma Figma)

https://www.figma.com/file/Bz4cZAbiEbt4cG6cUk9NWA/Projeto-PI?type=design&node-id=8%3A2&mode=design&t=McNrfycdR6O6vM72-1

3.2.2 FRONT-END

O desenvolvimento do front-end se refere a construção da interface gráfica do usuário (GUI), nela elementos visuais como textos, botões, imagens e etc são criados, caso contrário o usuário teria que interagir com o programa através de linhas de comando.

Em nosso projeto iniciamos essa etapa utilizando a linguagem de marcação HTML5, a linguagem de estilo CSS, e a linguagem de programação JavaScript, porém mais tarde no desenvolvimento foi decidido transicionar para o framework Next.js, pois com ele poderia-se trabalhar com o paradigma de orientação a objeto, além de ele permitir criar sistemas reativos. A transição foi fácil pois o Next.js tem suporte para CSS, a parte do HTML5 teve que ter poucas alterações, no entanto o JavaScript foi reescrito utilizando as ferramentas no framework, fazendo esta parte muito mais sucinta e eficiente.

3.2.3 BACK-END

O desenvolvimento do back-end se refere a construção das ligações entre o front-end e o banco de dados, internet, outro front-end e etc, ele é responsável por validar o login de um usuário, por trazer e guardar os dados requisitados, sem ele sistemas seriam interfaces gráficas estáticas.

Em nosso projeto criamos o back-end através de uma API, para isso utilizamos a linguagem de programação JavaScript lado servidor, para facilitar esse processo usamos o ambiente de execução Node.js, ele permite trabalhar o JavaScript sem utilizar um navegador web, e em conjunto usamos o Express.js, e para se conectar com o banco de dados usamos o mapeamento de objeto-relacional (ORM) Sequelize.

3.2.4 CRUD

O sistema desenvolvido contém as quatro operações básicas de armazenamento persistente, create, read, update e delete (CRUD), isso significa que o sistema é capaz de criar, ler, alterar e apagar dados. Estas são funções básicas que o CRUD tem em nosso sistema:

- Create: Cria novos registros de usuários, salva os dados das simulações feitas.

- Read: Valida a existência de um usuário, traz as simulações salvas por um usuário.
- Update: Modificar um registro de uma simulação salva.
- Delete: Exclui um usuário, exclui um registro de uma simulação (O registro não é realmente deletado, ele ficará salvo, porém o usuário não terá mais acesso).

3.3 COMPUTAÇÃO EM NUVEM

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

A computação em nuvem é uma tecnologia que oferece uma série de benefícios para as empresas, incluindo melhorar a eficiência operacional, reduzir custos e aumentar a escalabilidade. Aqui estão algumas maneiras pelas quais a computação em nuvem pode ajudar a alcançar esses objetivos:

A nuvem permite escalar recursos para cima ou para baixo de acordo com as necessidades. Isso é particularmente valioso para empresas que experimentam flutuações sazonais ou crescimento imprevisível, provisionar novos recursos na nuvem é geralmente muito mais rápido do que adquirir e configurar hardware físico. Isso permite que a empresa responda rapidamente às mudanças nas demandas do mercado.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

São reduzidos os gastos com hardwares, licenças e atualizações de softwares e também com a manutenção de grandes equipes para o suporte dos serviços de tecnologia.

No serviço na Nuvem, os dados ficam disponíveis e acessíveis por meio da internet. Isso facilita muito o trabalho de todas as áreas da empresa e ainda traz mais flexibilidade para o dia a dia dos colaboradores, que podem trabalhar no modo remoto, em home office ou durante viagens, sem prejuízo para a qualidade da interação e da realização dos serviços.

Ao escolher uma plataforma de Cloud para execução de seus sistemas e armazenamento de dados, a empresa ganha também a possibilidade de fácil escalabilidade, ou seja, pode ampliar suas operações na Nuvem conforme a necessidade de crescimento do seu negócio.

Os provedores de serviços em Nuvem oferecem uma segurança muito mais contínua e eficiente do que a segurança que seria feita eventualmente por ferramentas e equipe interna de TI.

Redução de perdas, que podem ocorrer por causa dos mais diferentes fatores. Podem acontecer desastres naturais como enchentes ou vendavais, incêndios ou roubos, que podem impossibilitar o acesso às informações por até um longo período.

3.3.3 VANTAGENS DA CLOUD COMPUTING

O cloud computing traz, principalmente, benefícios financeiros. Como todo o processo é feito através da rede, não é necessário fazer um investimento muito pesado. Não é preciso ter um hardware poderoso e ter uma infraestrutura para mantê-lo. O Cloud Computing dispensa o uso de data centers, servidores, além dos custos de resfriamento e da energia, necessários para manter esses equipamentos em uso.

A escalabilidade diz respeito à capacidade de um sistema se expandir sem perder o desempenho. Simplificando, se um sistema de computação em nuvem (redes, armazenamento, servidores, aplicativos e serviços) puder responder rapidamente para atender a novas demandas em tamanho ou volume, ele é escalável.

O acesso fácil às informações também é uma das principais vantagens da tecnologia cloud computing. Isso acontece porque a nuvem é um servidor remoto, que funciona por meio da internet. Ou seja, o acesso aos dados armazenados podem ser feitos de qualquer lugar e sempre que o usuário precisar.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

Explicar como o modelo de aplicação em cloud computing será implementado no projeto da empresa, levando em consideração aspectos como Software as a Service (SaaS), Platform as a Service (PaaS) ou Infrastructure as a Service (IaaS).

Destacar a importância do balanceamento de carga em cloud computing para garantir o desempenho, a disponibilidade e a eficiência das aplicações na nuvem.

Descrever os principais componentes e elementos da arquitetura de cloud computing que serão relevantes para o projeto da empresa.

Explorar como esses elementos se relacionam entre si e como contribuem para o funcionamento da infraestrutura em nuvem, considerando as necessidades e requisitos específicos da empresa.

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

A AWS oferece uma ampla gama de serviços em nuvem, incluindo computação, armazenamento, banco de dados, machine learning, análise de dados, Internet das Coisas (IoT), segurança e muito mais.

Essa diversidade permite que as empresas escolham e integrem facilmente os serviços necessários para atender às suas demandas específicas, além disso tem uma grande comunidade de usuários e um ecossistema robusto de parceiros e desenvolvedores. Isso significa que há uma ampla gama de recursos, tutoriais, e soluções prontas disponíveis para os usuários, algo que não vemos no cloud provido pela google.

Outro ponto importante é que a AWS oferece integrações fáceis com outras tecnologias e serviços populares. Isso facilita a criação de soluções abrangentes que atendam a várias necessidades.

Segurança, os data centers da AWS são projetados com múltiplas camadas de segurança física, incluindo controles biométricos, vigilância por vídeo, equipe de segurança dedicada e acesso restrito, mas claro que além disso tem a segurança digital,

que a AWS também oferece recursos abrangentes de criptografia para proteger dados em trânsito e em repouso.

Isso inclui a capacidade de criptografar dados armazenados em serviços como Amazon S3 e Amazon RDS, bem como a opção de usar SSL/TLS para comunicação segura.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

O modelo de aplicação escolhido pelo grupo foi o PaaS (Platform as a Service), o motivo principal foi pelo fato deste modelo fornecer uma plataforma completa para os desenvolvedores construírem, hospedarem e escalarem o nosso aplicativo sem se preocupar com a infraestrutura subjacente. Ferramentas de desenvolvimento, serviços de banco de dados e ambientes de execução.

O balanceamento de carga em cloud computing é crucial para otimizar o desempenho, melhorar a confiabilidade e garantir a escalabilidade. Ele irá distribuir o tráfego entre as várias instâncias de servidores para evitar sobrecarga e garantir que cada servidor esteja operando eficientemente. Isso é especialmente importante no nosso ambiente de nuvem, onde a demanda pode variar rapidamente.

A elasticidade é a capacidade de aumentar ou diminuir dinamicamente os recursos conforme a demanda, garantindo eficiência e otimização de custos, algo que para uma nuvem é de suma importância, outro fator de suma importância é a acessibilidade da aplicação em nuvem, recursos na nuvem são acessados por meio da Internet, proporcionando acesso remoto e flexibilidade de localização.

Com o surgimento de novas tecnologias novas áreas de estudo da ciência da computação surgem, e com elas novos paradigmas tecnológicos, estes que são padrões de como certas tecnologias são usada, um desses paradigmas é a computação em nuvem, porém paradigmas tecnológicos podem ser tão complexos que para sustentá los são criados paradigmas subjacentes, e na computação em nuvem temos os seguintes:

- Computação de alto desempenho (HPC): Este modelo diz a respeito de usar vários processadores, discos e memórias para criar uma única unidade de processamento, os exemplos mais comuns são os clusters, que são computadores ligados entre si trabalhando em uma tarefa comum.

- Computação paralela: Este método oriundo do HPC visa usar várias unidades de processamento de dados para trabalhar em uma única tarefa ao mesmo

tempo, para isso problemas são divididos em partes menores, desta forma vários processos podem ser resolvidos ao mesmo tempo.

- Computação distribuída: Neste modelo vários computadores que estão conectados entre si em uma mesma rede, trabalham como um único sistema, desta forma vários processos podem ser executados ao mesmo tempo.

- Sistemas de clusterização: Neste sistema vários computadores são conectados em uma infraestrutura de rede dedicada, além disso existe um software responsável pela comunicação entre os processos existentes, desta forma todas as máquinas podem trabalhar em um único processo juntas.

- Computação em grade: Neste modelo computadores são conectados em uma rede, onde cada componente terá diferentes localizações geográficas, criando uma única unidade de computação, esta que trabalha para resolver tarefas em conjunto.

- Redes de computadores: Essa arquitetura de redes é o ponto principal da computação em nuvem, afinal ela representa a rede mundial de computadores, conhecida como internet, esta que permite um usuário que pode estar conectado a partir de qualquer lugar se comunicar com servidores que estão em outra posição geográfica, e isso é o conceito de computação em nuvem.

3.3.7 GOOGLE CLOUD ou AWS

A AWS é uma plataforma de computação em nuvem oferecida pela Amazon. Um dos grandes pontos positivos dela é sua escalabilidade, confiabilidade e variedade de serviços, a AWS permite que empresas e desenvolvedores hospedem aplicativos, armazenem dados e implementem soluções inovadoras de maneira eficiente.

Ela é uma solução de serviços tendo uma extensa quantidade de serviços, incluindo computação, armazenamento e análise de dados, a AWS desempenha um papel crucial na transformação digital de organizações em todo o mundo.Como por exemplo, sua infraestrutura global e recursos flexíveis facilitam o desenvolvimento e a execução de aplicações, independentemente do tamanho ou complexidade.

3.4 ESTRUTURA DE DADOS

Refere-se à organização, armazenamento e manipulação de informações de maneira eficiente, proporcionando meios para organizar e armazenar dados de modo que possam ser facilmente acessados e utilizados. As estruturas de dados são fundamentais para o desenvolvimento de algoritmos eficientes, pois influenciam diretamente no desempenho e na eficiência das operações realizadas sobre os dados.

Logo, existem diversas estruturas de dados, cada uma projetada para atender a diferentes requisitos e cenários de aplicação. Exemplos comuns incluem arrays, listas ligadas, pilhas, filas, árvores e grafos.

Portanto, a escolha adequada da estrutura de dados depende das operações que serão realizadas com os dados e das características específicas do problema a ser resolvido. O estudo das estruturas de dados é essencial para programadores e cientistas da computação, pois contribui para o desenvolvimento de soluções eficientes e otimizadas.

3.4.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é a descrição em tópicos de um software. Ele é dividido em duas partes, os requisitos funcionais, e os não funcionais, sendo os funcionais as funções que o sistema será capaz de tomar, já os não funcionais são características mais técnicas, como manutenção, uso, desempenho, custo e interface. O levantamento de requisitos de um software é imprescindível para um projeto, pois é ele quem estabelece as características primas do software, desta forma alinhando a equipe de desenvolvimento e o cliente. Abaixo temos os requisitos funcionais e não funcionais do software que desenvolvemos:

Requisitos funcionais:

- RF1: O sistema deve ter um sistema de registro e login;
- RF2: O sistema deve calcular juros simples e compostos, investimentos e porcentagem;
- RF3: O sistema deve permitir ao usuário salvar cálculos feitos, guardando os números usados para o cálculo e o resultado;
- RF4: O sistema deve conter textos informativos sobre finanças domésticas, bem como referências qualificadas.

Requisitos não funcionais:

- RNF 1: O sistema deve ser desenvolvido em plataforma web;
- RNF 2: O banco de dados deve ser o MySQL;
- RNF 3: O sistema deve ser desenvolvido em Node.js na parte do server-side e Next.js na parte do client-side;
- RNF 4: O sistema deve ser desenvolvido baseando-se no paradigma da programação orientada a objetos.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS

O conteúdo de formação para vida ensina a união entre pessoas, seja em um grupo de amigos, tarefa ou trabalho daqueles que querem o mesmo objetivo, dissolvendo estereótipos entre comunidades.

Ao trabalhar esse tema vemos que mesmo hoje em dia, a sociedade não está livre de preconceitos ou rejeições por razões fulas, claro que ao se trabalhar em equipe é preciso escolher uma pessoa no qual tenha-se compatibilidade, porém sempre carregando consigo respeito e valores embasados.

3.5.1 ENFRENTANDO ESTEREÓTIPOS

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

No convívio social, o estereótipo exerce uma forte influência na vida de diversas pessoas, ao moldar seu comportamento e percepção diante outros grupos sociais, categorizando pessoas baseado em características superficiais. Isto acaba limitando a compreensão e dificultando a convivência de uma sociedade.

- **Tópico 2:** Estereótipo e representação;

Mídias e narrativas que perpetuam estereótipos podem reforçar visões distorcidas e prejudiciais da realidade, como por exemplo dizer que todos que moram no Brasil sabem jogar futebol, portanto buscar representações concisas e diversificadas auxilia em uma compreensão mais justa ao abordar esse tema.

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

O mundo nas redes sociais é onde normalmente todos compartilham uma vida de glamurosa e divertida a todo momento, buscando likes e validação de outros usuários que por sua vez, procuram espelhar-se em sonhos perfeitos de felicidade, sucessos e ambições a todo custo, mas na realidade, criam uma realidade distorcida, por isso é crucial saber as discrepâncias vida real e virtual.

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

"É em mais fácil desintegrar um átomo do que um preconceito." - Albert Einstein. Saber conviver em sociedade é um desafio e uma oportunidade para todo ser humano, enriquecer pessoalmente, além de proporcionar experiências diversas, aumenta o conhecimento sobre tópicos que jamais se questionou, como um médico sabe as dificuldades de um pedreiro ao construir uma casa?. Apenas deixando as diferenças de lado, com diálogo aberto tem-se oportunidade de aprender.

3.5.2 ESTUDANTES NA PRÁTICA

FILHOS DA MESMA TERRA

O Brasil é um pais de incontáveis costumes, diversidades culturais e variações linguísticas. Cada pedacinho do seu chão é regado por boas histórias e tradições.

A pluralidade brasileira seguidamente é atacada por falta de conhecimento e vivencia. As pessoas do interior, do campo, fazendas, roças ou sítios, são constantemente de forma infundada menosprezadas e diminuídas perante a uma sociedade urbanizada, com crença de superioridade.

É válido ressaltar que, também é do campo, das fazendas, roças e sitios, que toda alimentação chega aos grandes centros e cidades, sejam da agricultura ou de origem animal.

O Brasil é altamente privilegiado com sua terra boa e fértil, com sua água e muito mais com sua gente simples e trabalhadora, com mãos muitas vezes calejadas, e seus rostos queimados de sol, trabalham muito para levar o alimento para todos aqueles que somente conhecem o asfalto e o cimento.

As diferenças nos ensinam constantemente como se tornar um ser humano melhor, cabe a cada um, decidir se aproveita a oportunidade ou se deixa passar.

"PRECONCEITO É OPINIÃO SEM CONHECIMENTO" - VOLTAIRE.

4. CONCLUSÃO

Diante desse projeto, tivemos como objetivo ajudar e colaborar com o conhecimento básico de finanças e cálculos primários sobre juros juntamente com investimentos na vida financeira das pessoas. Sendo assim, a importância da educação financeira é dar condições para uma pessoa decidir melhor o que fazer com seu dinheiro.

Ademais, ter o poder de saber investir, organizar e saber quais atitudes tomar com suas economias, é um sentimento de paz. Sendo assim, isso vai além do simples fato de economizar, também diz respeito à consciência das oportunidades e riscos envolvendo esse tema, pois ninguém sabe quais eventos inesperados podem acontecer. Por isso, aprender a lidar com o dinheiro o quanto antes é essencial. E com nosso projeto podemos auxiliar isso na vida do povo.

Portanto, para nossa equipe foi de grande importância o desenvolvimento deste projeto, pois tivemos a oportunidade de aplicar os conhecimentos adquiridos ao decorrer do curso, trazendo grande crescimento educacional e profissional. Por fim, a realização deste projeto foi um aprendizado, não apenas para conhecimentos voltados a área da tecnologia, também tivemos o aprendizado do lado financeiro diante de todas as pesquisas feitas para esse projeto. Uma de nossas maiores dificuldades foi o tempo, como iríamos conciliá-lo e administrá-lo. Consoante, o efeito resultante foi satisfatório, aprendemos, trabalhamos e adquirimos conhecimentos, mas primeiro, concluímos de maneira satisfatória, de modo que tenhamos orgulho de nosso trajeto.

REFERÊNCIAS

AWS AMAZON (2023). O que é virtualização? Disponível em: https://aws.amazon.com/pt/what-is/virtualization/. Acesso em: 9 de maio de 2023.

Cloudflare (2023). O que é plataforma como serviço (PaaS)? . Disponível em: https://www.cloudflare.com/pt-br/learning/serverless/glossary/platform-as-a-service-paas/. Acesso em: 6 de maio de 2023.

Google Cloud (11 de Maio de 2023). Load balancing and scaling. Disponível em: https://cloud.google.com/compute/docs/load-balancing-and-autoscaling?hl=en. Acesso em: 4 de maio de 2023.

Oracle (2023).O que é um banco de dados em nuvem? Disponível em: https://www.oracle.com/br/database/what-is-a-cloud-database/. Acesso em: 5 de maio de 2023.

Vitaliy Ilyukha (2023).Choose the Best Cloud Platform: AWS vs Azure vs Google Cloud Disponível em: https://jelvix.com/blog/aws-vs-google-cloud-vs-azure. Acesso em: 12 de maio de 2023.

Longbing Cao (2015) Metasynthetic Computing and Engineering of Complex Systems Disponível em: https://link.springer.com/book/10.1007/978-1-4471-6551-4.

React (2023). React Reference. Disponível em: https://react.dev/reference/react

PENSADOR. Albert Einstein. "É mais fácil desintegrar um átomo do que um preconceito." Disponível em: https://www.pensador.com/frase/MzYxMQ/. Acesso em: 21 nov. 2023.

ANEXOS

Name: nova_operacao

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `nova_operacao`(IN data0 DATE,IN tipo0 VARCHAR(2)
,IN campo10 FLOAT,IN campo20 FLOAT,IN campo30 FLOAT,IN campo40 FLOAT,IN usuario0 INT)
BEGIN
insert into
operacoes(data,tipo,campo1,campo2,campo3,campo4,usuarios_idusuarios)
values (data0,tipo0,campo10,campo20,campo30,campo40,usuario0);
END
```

(ANEXO 1.1 – Stored procedure que cria uma nova operação no banco)

Name: novo_usuario

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `novo_usuario`(IN nomeU VARCHAR(50),
IN nascimentoU DATE,IN emailU VARCHAR(60),IN senhaU VARCHAR(20))
BEGIN
insert into usuarios(nome,nascimento,email,senha) values (nomeU,nascimentoU,emailU,senhaU);
END
```

(ANEXO 1.2 – Stored procedure que cria um novo usuário no banco)

Name: deletar_operacao

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `deletar_operacao`(IN idoperacoesD INT)
BEGIN
delete from operacoes where idoperacoes = idoperacoesD;
END
```

(ANEXO 1.3 – Stored procedure que deleta uma operação no banco)

Name: deletar_usuario

The name of the routine is parsed automati
statement. The DDL is parsed automaticall

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `deletar_usuario`(IN idD INT)
BEGIN
delete from usuarios where idusuarios = idD;
END
```

(ANEXO 1.4 – Stored procedure que deleta um usuário no banco)

Name: update_operacao

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `update_operacao`(IN idoperacoesU INT,IN dataU VARCHAR(10),
IN tipoU VARCHAR(2),IN campo1U FLOAT,IN campo2U FLOAT,IN campo3U FLOAT,IN campo4U FLOAT)
BEGIN
update operacoes set
data = if (dataU="",data,dataU),
tipo = if (tipoU="",tipo,tipoU),
campo1 = if (campo1U="",campo1,campo1U),
campo2 = if (campo2U="",campo2,campo2U),
campo3 = if (campo3U="",campo3,campo3U),
campo4 = if (campo4U="",campo4,campo4U)
where idoperacoes = idoperacoesU;
END
```

(ANEXO 1.5 – Stored procedure que atualiza uma operação no banco)

Name: update_usuario

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `update_usuario`(IN idusuariosU INT,IN nomeU VARCHAR(50),
IN nascimentoU VARCHAR(10),IN emailU VARCHAR(60),IN senhaU VARCHAR(20))
BEGIN
update usuarios set
nome = if (nomeU="",nome,nomeU),
nascimento = if (nascimentoU="",nascimento,nascimentoU),
email = if (emailU="",email,emailU),
senha = if (senhaU="",senha,senhaU)
where idusuarios = idusuariosU;
END
```

(ANEXO 1.6 – Stored procedure que atualiza um usuário no banco)

Name: usuario_operacao

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `usuario_operacao`(IN idusuarioUO INT)
BEGIN
select data,tipo,campo1,campo2,campo3,campo4 from operacoes where usuarios_idusuarios = idusuarioUO;
END
```

(ANEXO 1.7 – Stored procedure que visualiza todas as operações de um usuário específico no banco)

Name: resultado

DDL:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `resultado`(idoperacoesF INT) RETURNS float
BEGIN
declare c1,c2,c3,c4,resultado FLOAT;
declare tipoF VARCHAR(45);

select campo1,campo2,campo3,campo4,tipo into c1,c2,c3,c4,tipoF
from operacoes where idoperacoes=idoperacoesF;

if tipoF="JS" then
    set resultado = (c1 * c2 * c3/100) + c1;
elseif tipoF="JC" then
    set resultado = c1*power((1 + c3/100),c2);
elseif tipoF="IN" then
    set resultado = c4 * ((power((1+c3/100),c2)-1)/c3/100);
elseif tipoF="PR" then
    set resultado = c1 * c2/100;
end if;
RETURN resultado;
END
```

(ANEXO 2 – A function que recebe o tipo do cálculo e executa-o)

Table Name: operacoes Schema: edfinanceiro

Charset/Collation: Default Charset / Default Collation Engine: InnoDB

Comments:

- BEFORE INSERT
- AFTER INSERT
  - operacoes_AFTER_INSERT
- BEFORE UPDATE
- AFTER UPDATE
  - operacoes_AFTER_UPDATE
- BEFORE DELETE
- AFTER DELETE
  - operacoes_AFTER_DELETE

```
CREATE DEFINER=`root`@`localhost` TRIGGER `operacoes_AFTER_INSERT` AFTER INSERT ON
`operacoes` FOR EACH ROW BEGIN
insert into logoperacoes(idLog,tipoLog,dataLog,data,tipo,campo1,campo2,campo3,
campo4,idusuario)
values (new.idoperacoes,"Criado",NOW(),new.data,new.tipo,new.campo1,new.campo2,
new.campo3,new.campo4,new.usuarios_idusuarios);
END
```

(ANEXO 3.1 – Trigger que guarda as modificações de entrada da tabela operações no operacoeslog)

Table Name: operacoes Schema: edfinanceiro

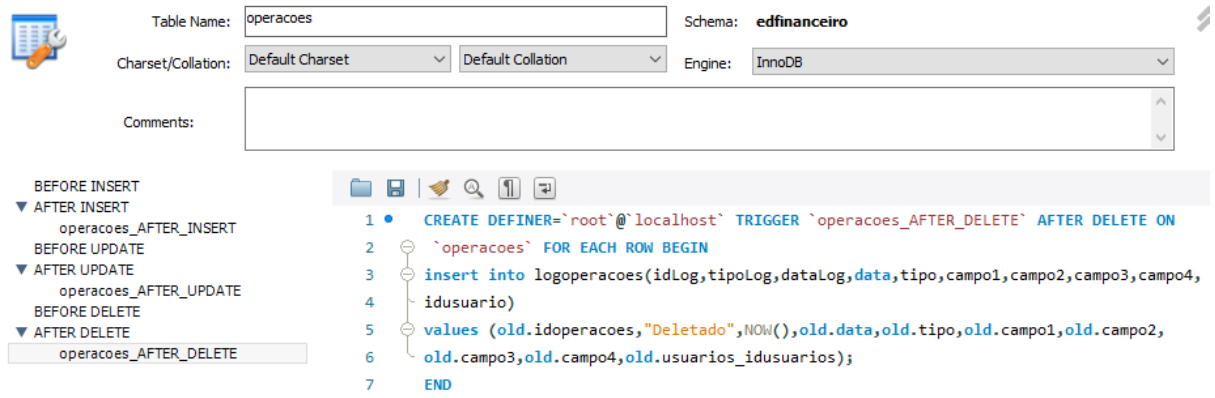
Charset/Collation: Default Charset / Default Collation Engine: InnoDB

Comments:

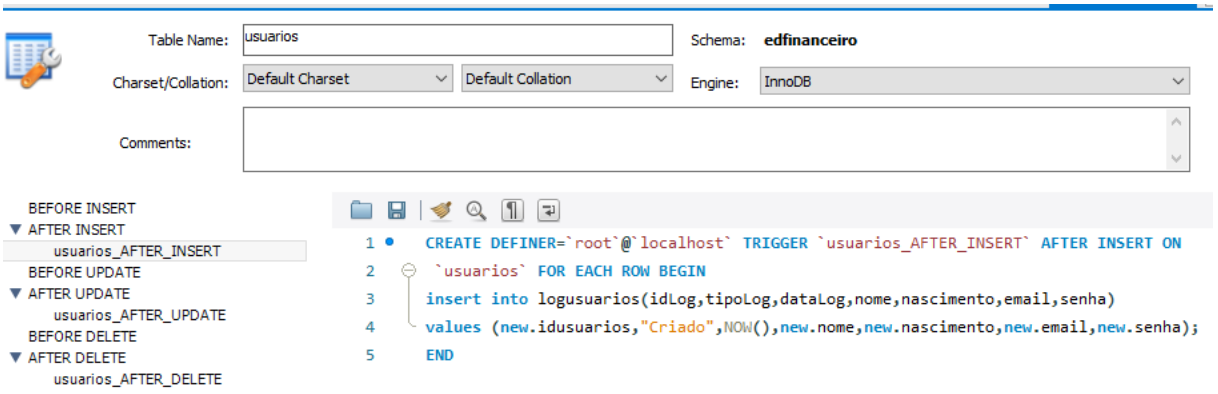
- BEFORE INSERT
- AFTER INSERT
  - operacoes_AFTER_INSERT
- BEFORE UPDATE
- AFTER UPDATE
  - operacoes_AFTER_UPDATE
- BEFORE DELETE
- AFTER DELETE
  - operacoes_AFTER_DELETE

```
CREATE DEFINER=`root`@`localhost` TRIGGER `operacoes_AFTER_UPDATE` AFTER UPDATE ON
`operacoes` FOR EACH ROW BEGIN
insert into logoperacoes(idLog,tipoLog,dataLog,data,tipo,campo1,campo2,
campo3,campo4,idusuario)
values (new.idoperacoes,"Update",NOW(),new.data,new.tipo,new.campo1,new.campo2,
new.campo3,new.campo4,new.usuarios_idusuarios);
END
```

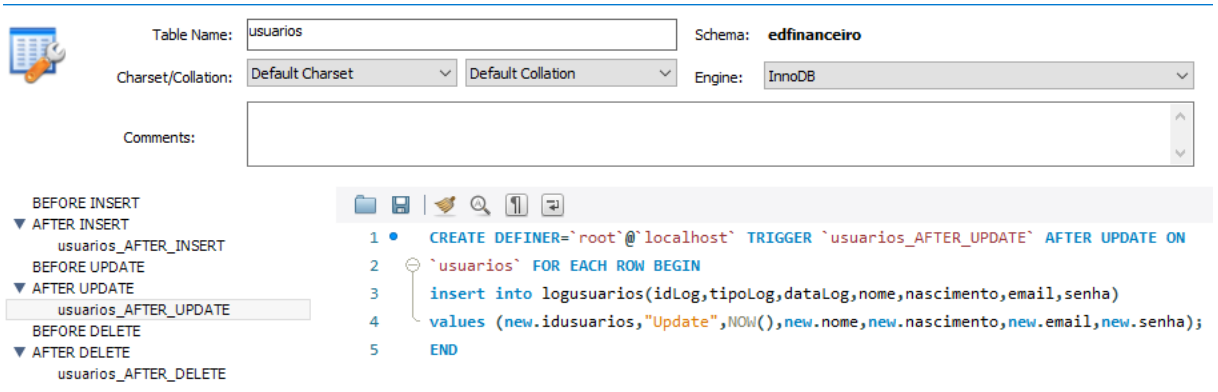
(ANEXO 3.2 – Trigger que guarda as modificações de atualização da tabela operações no operacoeslog)



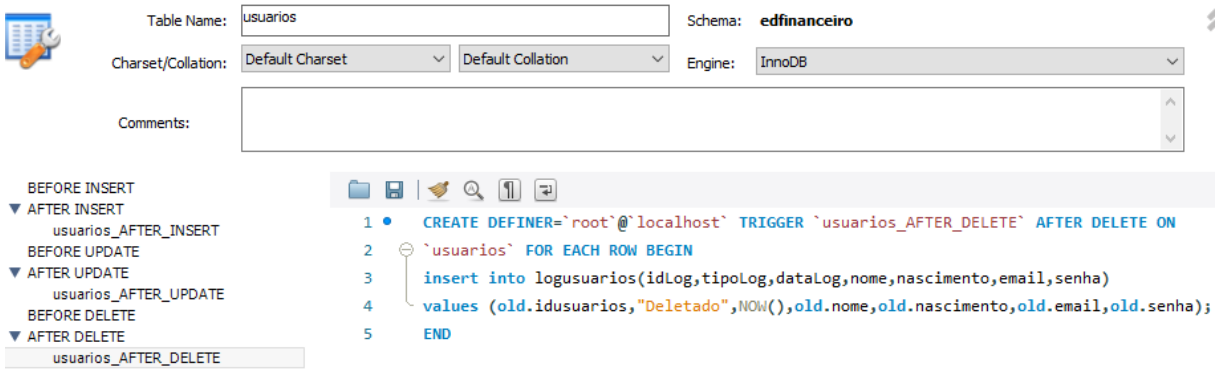
(ANEXO 3.3 – Trigger que guarda as modificações de deletar da tabela operações no operacoeslog)



(ANEXO 3.4 – Trigger que guarda as modificações de entrada da tabela usuarios no usuarioslog)



(ANEXO 3.5 – Trigger que guarda as modificações de atualização da tabela usuarios no usuarioslog)



(ANEXO 3.6 – Trigger que guarda as modificações de deletar da tabela usuarios no usuarioslog)