



**USP - UNIVERSIDADE DE SÃO PAULO**  
**INSTITUTO DE CIÊNCIAS MATEMÁTICAS E COMPUTAÇÃO (ICMC)**  
**DISCIPLINA BASE DE DADOS**  
**DOCENTE ELAINE PARROS MACHADO DE SOUSA**

**PROJETO: RECURSOS**  
**MÍNIMOS DE ACESSIBILIDADE**

**Bernardo Maia Coelho (12542481)**  
**Carlos Filipe de Castro Lemos (12542630)**  
**João Gabriel Sasseron Roberto Amorim (12542564)**  
**Pedro Guilherme dos Reis Teixeira (12542477)**  
**Pedro Henrique Vilela do Nascimento(12803492)**

**São Carlos**

**2023**

## Sumário

<b>1 - Descrição do Problema e dos Requisitos dos Dados.....</b>	<b>2</b>
1.1 - Restrições de Integridade.....	3
1.2 - Principais operações.....	4
<b>2 - Modelo Entidade de Relacionamento (MER).....</b>	<b>6</b>
2.1 - Diagrama.....	6
2.2 - Mudanças relacionadas a primeira entrega.....	6
<b>3 - Modelo Relacional.....</b>	<b>8</b>
3.1 - Diagrama.....	8
3.2 - Justificativas e anotações para o Modelo Relacional.....	8
3.2.1 - Método de especialização para a entidade Serviço.....	8
3.2.2 - Método de Especialização para a entidade Cliente.....	9
3.2.3 - Método de Especialização para a entidade Pessoa.....	9
3.2.4 - Utilização de ID sintético na relação Contribuição.....	10
3.2.5 - Utilização de tabelas para atributos multivalorados.....	10
3.2.6 - Restrição de Participação Total.....	10
3.2.7 - Relacionamento Ternário.....	10
3.3 - Mudanças relacionadas ao Modelo Relacional.....	11
<b>4 - Consultas.....</b>	<b>12</b>
4.1 - Informações sobre cada cliente e quanto lucro ele gerou de acordo com tipo de serviço.....	12
4.2 - Informações sobre todas as inform.....	12
4.3 - Todas as Informações sobre todos os comentários dados por cada testador de cada consultor em relação a um dado site requisitado por um dado cliente.....	13
4.4 - Quantidade de mudanças que cada desenvolvedor fez em projetos que iniciaram no mês de junho em 2023.....	14
4.5 - Seleciona os nomes de todos os consultores que contribuíram em todas as avaliações que o testador Endrick participou.....	15
<b>5 - Aplicação.....</b>	<b>16</b>
5.1 - Descrição Inicial.....	16
5.2 - Funcionalidades.....	16
5.2.1 - Inserção de dados.....	16
5.2.2 - Pesquisa de dados.....	17
5.2.3 - Remoção de dados.....	18
<b>6 - Conclusão.....</b>	<b>18</b>

# 1 - Descrição do Problema e dos Requisitos dos Dados

Segundo dados da Pesquisa Nacional de Saúde, produzida pelo IBGE (2019), cerca de 17,3 milhões de pessoas possuem algum tipo de deficiência. Além disso, com o aumento da expectativa de vida do brasileiro, associado com a diminuição das taxas de fecundidade, houve um natural envelhecimento da população no Brasil, redundando em 14% das mulheres e 12% dos homens com mais de 60 anos. De fato, essas pessoas representam uma pequena parcela do total de pessoas que enfrentam inúmeras dificuldades para ter acesso aos serviços básicos do dia a dia, especialmente no meio digital.

Nesse contexto, a empresa Recursos Mínimos encontrou um nicho de mercado para desenvolver produtos tecnológicos com empatia, de modo a criar ou ampliar a inclusão digital dos usuários e aproximá-los de empresas ou associações públicas ou privadas. Em termos mais específicos, os serviços da empresa visam **avaliar e adaptar websites para eliminar problemas de acessibilidade**, dentro dos padrões determinados pela *World Wide Web Consortium* (W3C) e pelo *Web Content Accessibility Guidelines* (WCAG). Além disso, mantém *online* um catálogo de suas avaliações sobre diferentes sites.

Em seguida, serão definidos os conceitos utilizados na descrição do funcionamento de tal empresa:

Primeiramente existe uma **pessoa**, um indivíduo o qual a empresa possui dados relevantes como CPE, nome e endereço. Nesse contexto, tal indivíduo é exclusivamente um **voluntário**, uma pessoa que contribui para uma avaliação profunda do website em conjunto com um ou mais consultores da empresa, agindo essencialmente como um avaliador sem viés; Ou um **funcionário**, um membro da empresa que é necessariamente um **consultor** e/ou um **desenvolvedor**.

O **cliente** é uma entidade que requisita os serviços da empresa, sendo identificado pelo seu e-mail. Um cliente sempre pertence a apenas uma dessas duas categorias: **Pessoa física** ou **pessoa jurídica**, sendo também armazenados os respectivos CPF e CNPJ de tais clientes. É importante notar que não é necessário que o cliente seja o proprietário legal do site para que possa requisitar sua avaliação, isto é, uma Associação de Cegos poderia requisitar a avaliação de acessibilidade de um dado site mesmo que não seja seu proprietário legalmente.

O **site** é um domínio digital onde um determinado serviço é realizado, este é identificado primeiramente pelo seu URL, mas secundariamente pelo seu nome e seu proprietário armazenados no banco de dados.

O **serviço** é um trabalho prestado pela empresa que foi solicitado por um cliente em relação a um ou mais sites. Ele é identificado pelo seu código de protocolo de serviço e é conhecido sua data de início, custo, preço e duração. O mesmo se encaixa exclusivamente em um de dois tipos:

**Adaptação**, onde **desenvolvedores** atualizam um site para que ele satisfaça certos critérios de acessibilidade; E **avaliação**, que recebe a contribuição de **consultores** e **voluntários** no seu desenvolvimento e gera um relatório de avaliação. Esse sendo identificado por notas em diferentes critérios de acessibilidade, comentários e, primariamente, uma data em junção ao código de protocolo de seu respectivo serviço.

A **mudança** é um registro de cada mudança realizada por um **desenvolvedor** sobre um **serviço** de **adaptação**. São armazenados a data e hora em que essas mudanças foram realizadas e também uma descrição geral do que foi alterado. Cada **mudança** pode ser identificada por sua data e hora e pelo CPF do desenvolvedor que a gerou.

Por último, existe o **usuário**, um indivíduo que criou uma conta online no site da empresa a fim de comentar e dar notas em relação aos sites avaliados. Ele é unicamente identificado pelo seu nome de usuário, mas outras informações também são armazenadas, como seu e-mail e o estado de atividade da conta, isto é, se ela está ou não suspensa por infringir o regulamento do site.

## 1.1 - Restrições de Integridade

- **Possível inconsistência do atributo “e-mail” de um cliente:**

Existe a possibilidade de o cliente trocar seu e-mail principal por outro após ele ter sido inserido no banco de dados, fazendo com que tal informação seja desatualizada. Durante o processo de requisição de serviço o cliente é informado de tal fato e que é necessário contactar a empresa para atualizar o seu e-mail, nessa situação a implementação é responsável por garantir que o novo dado em questão não seja nulo

- **Possível inconsistência de Domínio em relação às “Notas” em Contribui:**

Uma das operações possíveis no banco de dados é a relatórios de avaliações a partir de uma nota específica, porém as notas não são identificadas como chave primária da entidade e por consequência têm a possibilidade de serem nulas. É de responsabilidade do modelo relacional a garantia de que esse campo não esteja vazio.

- **Possível inconsistência de vazio em relação aos atributos CPF e CNPJ de uma Pessoa Física e uma Pessoa Jurídica, respectivamente:**

Por ambas entidades mencionadas serem especializações da entidade Cliente elas possuem e-mail como a sua chave primária, porém por razões de segurança e integridade de dados os atributos CPF e CNPJ não podem

ser nulos em qualquer situação. Deste modo, é responsabilidade do modelo relacional, da implementação do banco de dados ou da aplicação garantir que tais campos sejam sempre preenchidos com dados válidos.

## 1.2 - Principais operações

Considerando a abordagem anterior, as principais operações realizadas no sistema pelas entidades de maior importância serão:

- **Cliente:**
  - Cadastrar seu conjunto de dados;
  - Requisitar a deleção ou alteração de seus dados, possivelmente cancelando o serviço antes de ele estar completo.
  - Buscar pelo relatório gerado a partir da avaliação, se este foi gerado pelo serviço solicitado, filtrando pelo “URL” do respectivo site;
  - Buscar pelos atributos referentes ao serviço realizado pela empresa para o cliente, como quando ele foi iniciado, o quanto ele custou, os termos definidos, etc. Utilizando do “Código de protocolo de serviços” que o cliente recebe.
- **Usuário:**
  - Cadastrar, inserir, editar e deletar seu conjunto de dados;
  - Inserir, editar e deletar comentários e notas;
  - Buscar por comentários e notas, de um site e/ou aqueles feitos por outro usuário, filtrando pelo “URL” e/ou “Nome de Usuário”, respectivamente.
- **Voluntário:**
  - Inserir nova avaliação sobre um site em específico;
  - Buscar, editar e deletar seus próprios relatórios de avaliação.
- **Funcionários:**
  - Cadastrar, inserir, editar e deletar os dados de voluntários, usuários e clientes;
  - Inserir dados sobre o serviço a ser prestado, seguindo as especificações discutidas com o Cliente;
  - Buscar por Serviços prestados, podendo filtrar por “Código de protocolo”, “Data de início”, “Tipo” ou “Duração”;
  - Buscar por Relatório de Avaliação, podendo filtrar por “Data” ou “Notas”;
  - Buscar por Cliente, podendo filtrar por “e-mail”, “CPF” ou “CNPJ”.

- Buscar por Sites, podendo filtrar por “URL”;
- Buscar por Voluntário, podendo filtrar por “CPF”.

Vale salientar que Desenvolvedor e Consultor são especializações de funcionários, por causa disso, podem realizar todas as suas operações. Porém, estas instâncias possuem funcionalidades próprias que são as seguintes:

- **Desenvolvedor:**

- Inserir o site adaptado pela empresa com seu “URL” específico criado na base de dados;
- Editar e deletar sites na base de dados, podendo filtrar por “URL” desenvolvido pela empresa.

- **Consultor:**

- Inserir nova avaliação sobre um site em específico;
- Buscar, editar e deletar quaisquer relatórios de avaliação de seu serviço, podendo filtrar por “Código de protocolo”, “Data de início” ou “Duração”.

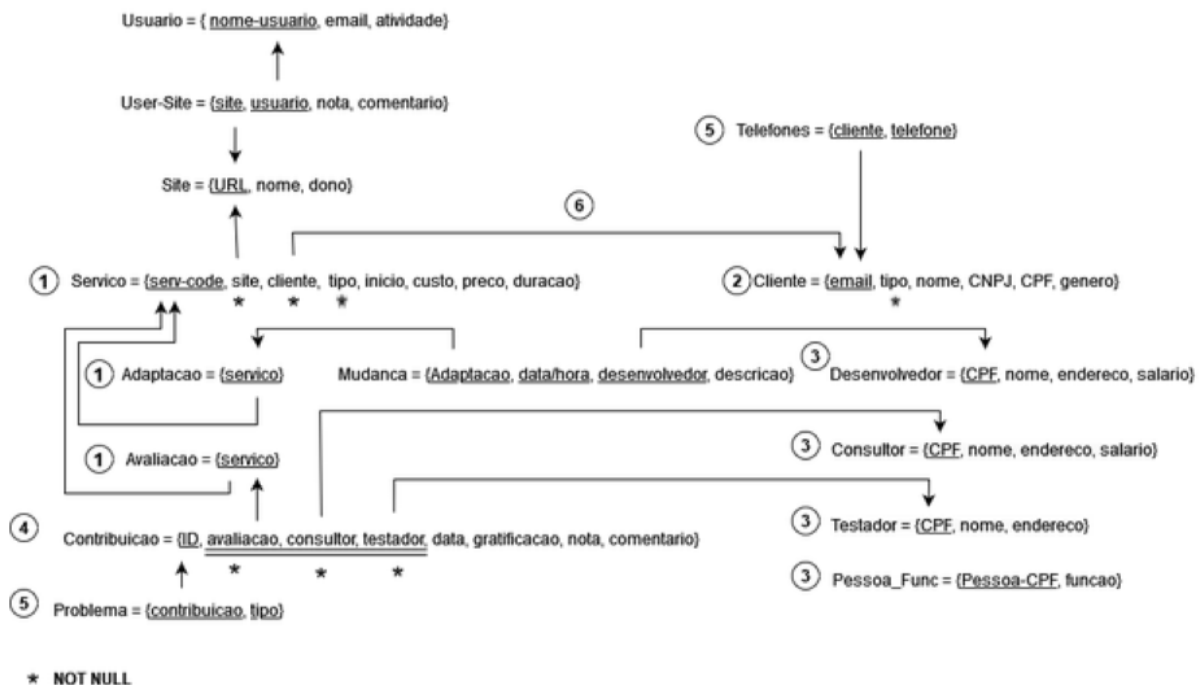


- O atributo “Salário” da generalização “Pessoa” passou a ser um atributo de sua especialização “Funcionário”;
- A especialização “Voluntário” foi renomeada para “Testador”;
- A entidade fraca “Relatório de Avaliação” foi removida;
- A relação “Contribui” agora é uma relação ternária N para N para N entre a entidade “Testador”, a entidade “Consultor” e a entidade “Avaliação”, respectivamente;
- A relação “Contribui” recebeu os atributos “Data/Hora”, “Gratificação”, “Nota”, “Comentário” e o atributo multivalorado “Problemas Encontrados”;
- A entidade “Cliente” recebeu o atributo multivalorado “Telefones”;
- A especialização “Pessoa Física” recebeu o atributo “Gênero”;
- A descrição dos requisitos de dados foram alteradas para refletir as mudanças no DER e um parágrafo foi adicionado para a agregação Mudança;
- A descrição das restrições de integridade relacionadas às notas da relação “Contribui” e aos atributos “CPF” e “CNPJ” das especializações “Pessoa Física” e “Pessoa Jurídica” receberam ligeiras alterações para refletirem as atualizações do DER e responsabilizar o modelo relacional com a garantia de integridade desses atributos;
- A descrição das principais operações foi alterada ligeiramente para refletir as atualizações do DER.



## 3 - Modelo Relacional

### 3.1 - Diagrama



### 3.2 - Justificativas e anotações para o Modelo Relacional

As justificativas para as decisões tomadas no modelo relacional serão apresentadas abaixo explicando a justificativa, assim como as vantagens e desvantagens.

#### 3.2.1 - Método de especialização para a entidade *Serviço*

- **Solução Adotada:** Foi utilizado um atributo *tipo* para a distinção das especializações *Avaliação* e *Adaptação*, tais tabelas sendo criadas a fim de manter as relações específicas de cada uma enquanto a tabela genérica *Serviço* armazena os atributos realmente utilizados.
- **Vantagens:** Permite uma manutenção da relação da entidade genérica serviço para *Cliente* e *Site*, simultaneamente aos relacionamentos específicos de cada especialização de uma forma simples facilitando ambos o desenvolvimento da aplicação e o processamento necessário. Além de mantém a consistência do bando de dados.
- **Desvantagens:** Não é possível garantir especialização total das especializações com este método. Assim como, não é possível manter a disjunção.

- **Solução Alternativa:** Criação de duas tabelas específicas (*Avaliação* e *Adaptação*) que possuiriam todos os atributos gerais, nesse caso a tabela genérica *Serviço* não existiria. Isso garantiria totalidade de especialização, porém faria com que a manutenção de relações entre entidades fosse mais complexa e portanto mais custosa.

### 3.2.2 - Método de Especialização para a entidade *Cliente*

- **Solução Adotada:** Na entidade *Cliente* foram armazenados todos os atributos especializados em conjunto com o atributo *tipo*, é esperado que a aplicação garanta que apenas os atributos relevantes ao tipo da especialização sejam armazenados.
- **Vantagens:** Esse método diminui o número de tabelas facilitando o processamento de pesquisas relacionadas a *Clientes* e a manutenção de suas relações.
- **Desvantagens:** Existência de NULLs nos atributos que não pertencem a especialização o que por consequência resulta em um uso maior do espaço de armazenamento. Ademais, não se garante a unicidade de CNPJ e CPF e não se garante que cada cliente tem ou CNPJ ou CPF não nulos (cabe à implementação do banco de dados ou à aplicação garantirem isso).
- **Solução Alternativa:** Como no ponto 3.2.1 uma alternativa seria a criação de tabelas referentes às especializações sem uma entidade genérica. Isso garantia um uso mais eficiente de espaço pela inexistência de NULLs nesse contexto, porém faria com que a manutenção de relações entre entidades fosse mais complexa e, portanto mais custosa em critério de processamento.

### 3.2.3 - Método de Especialização para a entidade *Pessoa*.

- **Solução Adotada:** Foram criadas tabelas das especializações *Desenvolvedor*, *Consultor* e *Testador*, que armazenam todos seus atributos genéricos em vez de uma tabela *Pessoa*. Além disso, foi criada uma tabela que *Pessoa\_Func* possibilita a pesquisa das funções de uma *Pessoa*, tais podendo ser: TE (*Testador*), DE(*Desenvolvedor*), CO (*Consultor*) e DC (Ambos *Desenvolvedor* e *Consultor*).
- **Vantagens:** Garante especialização total de todas as entidades, adicionalmente a tabela de pesquisa possibilita um método simples e eficiente de identificar as funções de cada indivíduo armazenado no sistema comparado a outros métodos que necessitam um maior poder de processamento. Além disso, possibilita que a consultado seja feita de mais eficiente, pois não há a necessidade de junções com a entidade genérica.
- **Desvantagens:** É possível que uma pessoa seja ambos desenvolvedor e consultor, o que causaria duplicação de seus dados em ambas as tabelas, isso adicionado a tabela de pesquisa faz com que um maior espaço de dados seja ocupado pela base de dados. Além disso, a disjunção não é garantida.

- **Solução Alternativa:** Poderia ser utilizado uma entidade genérica com as especializações, assim evitando a duplicação de dados e a necessidade de uma tabela de pesquisa. Porém, não seria possível garantir a totalidade de especializações ao mesmo tempo, em que a tabela genérica *Pessoa* não possuiria quaisquer relações próprias.

#### 3.2.4 - Utilização de ID sintético na relação *Contribuição*

- **Solução Adotada:** Na tabela *contribuição* foi criado um ID sintético, a fim de usá-lo como chave ao invés da chave composta tripla: *Avaliação*, *Testador*, *Consultor*.
- **Vantagens:** Não há a propagação da chave tripla para a tabela *Problema* que utiliza *contribuição* como chave estrangeira. Além da menor utilização de memória.
- **Desvantagens:** Há a criação do atributo extra ID sintético, inevitavelmente gastando uma maior quantidade de espaço, porém utiliza-se menos memória para a replicação da chave, assim não sendo necessário a duplicação de todos os valores da chave, apenas do ID.
- **Solução Alternativa:** A utilização direta da chave tripla na tabela *Problema*.

#### 3.2.5 - Utilização de tabelas para atributos multivalorados

- **Solução Adotada:** Foi criada uma tabela para cada atributo multivalorado, no caso *Telefones* (da entidade *Clientes*) e *Problema* (da entidade *Contribuição*).
- **Vantagens:** Aumenta o limite numérico para os atributos multivalorados em suas respectivas tabelas.
- **Desvantagens:** São criadas duas tabelas a mais, assim utilizando um maior espaço de armazenamento e criando a necessidade de mais *joins* para a análise de tais dados.
- **Solução Alternativa:** Determinar um número máximo fixo de tais atributos, sendo essas colunas em suas respectivas tabelas. Isso limitaria o número de atributos e possivelmente resultaria em NULLs em certas linhas das tabelas.

#### 3.2.6 - Restrição de Participação Total

No modelo relacional, não é possível garantir a participação total na relação entre cliente e serviço em decorrência da cardinalidade N para N. Cabe à aplicação ou à implementação do banco de dados garantir essa restrição.

#### 3.2.7 - Relacionamento Ternário

No modelo relacional, foi adicionado o relacionamento ternário, pois na relação *Avaliação* o *Voluntário* é sempre acompanhado por um consultor para a geração de

seus dados. Deste modo, reduzindo possíveis redundâncias de dados, assim como uma maior consistência para os dados sobre um site.

### 3.3 - Mudanças relacionadas ao Modelo Relacional

- Atribuição de NOT NULL para o atributo *tipo* da especialização nas relações *Serviço* e *Cliente*, no diagrama do modelo relacional.
- Complemento das vantagens e desvantagem do Método de especificação para a entidade *Serviço* (3.2.1). Adicionando a vantagem de garantir a consistência dos dados. Além da desvantagem de não garantir a disjunção na especialização.
- Complemento das vantagens e desvantagem do Método de especificação para a entidade *Pessoa* (3.2.3). Adicionando a vantagem em relação à consulta que é feita de forma mais eficiente. Além da desvantagem de não garantir a disjunção na especialização.
- Complemento das vantagens e desvantagem da Utilização de ID sintético na relação *Contribuição* (3.2.4). Adicionado a vantagem de que se utiliza um menor espaço de memória para referência da chave. Além da justificativa para a desvantagem.
- Alteração da vantagem em Utilização de tabelas para atributos multivalorados (3.2.5).
- Adição da justificativa da utilização do relacionamento ternário.

## 4 - Consultas

### 4.1 - Informações sobre cada cliente e quanto lucro ele gerou de acordo com tipo de serviço

```
SELECT
    NVL(S.TIPO, 'NONE'),
    C.NOME,
    C.EMAIL,
    NVL(SUM(S.PRECO - S.CUSTO), 0) AS LUCRO,
    COUNT(S.CUSTO) AS QTD_SERVICO
FROM CLIENTE C
    LEFT JOIN SERVICO S
        ON S.CLIENTE = C.EMAIL
GROUP BY C.NOME, C.EMAIL, S.TIPO
ORDER BY S.TIPO, LUCRO DESC;
```

Consulta quanto lucro a soma dos serviços prestados para cada cliente gerou e quantos serviços o cliente contratou, dividido por tipo de serviço prestado, ordenado de forma decrescente pela quantidade de lucro. Essa funcionalidade permite à empresa identificar quais clientes geram maior resultado para a empresa e, então, devem receber maior prioridade.

**OBS:** Clientes que não contrataram nenhum serviço também são selecionados, porém, são separados dos outros clientes para que a empresa possa identificar clientes que não contrataram nenhum serviço.

### 4.2 - Informações sobre todos os comentários selecionados de acordo com suas respectivas notas

```
SELECT DISTINCT
    U.NOME_USUARIO AS USUARIO,
    U.EMAIL,
    UST.NOTA,
    UST.COMENTARIO
FROM USUARIO U
    JOIN USER_SITE UST
        ON UST.USUARIO = U.NOME_USUARIO
    JOIN SITE ST
        ON UST.SITE = ST.URL
WHERE UST.NOTA > 3
    AND ST.NOME = 'InclusioTech'
    AND U.ATIVIDADE = 'S'
ORDER BY UST.NOTA DESC;
```

Consulta todos os comentários cuja nota é maior que um valor arbitrário escritos por usuários sobre um dado site. Essa consulta é essencial para a

ferramenta de filtragem de comentários que os usuários teriam acesso na página de review do site.

**OBS:** Nesse caso, comentários de usuários inativos (usuários cujo acesso às suas contas foi restrito pela moderação) são excluídos dos comentários que aparecem para outros usuários.

#### 4.3 - Todas as Informações sobre todos os comentários dados por cada testador de cada consultor em relação a um dado site requisitado por um dado cliente

```
SELECT
  CS.NOME AS CONSULTOR,
  TC.NOME AS TESTADOR,
  EXTRACT(YEAR FROM CT.DATA) AS ANO,
  CL.NOME AS CLIENTE,
  ST.NOME AS SITE,
  CT.NOTA,
  CT.COMENTARIO
FROM TESTADOR TC
  JOIN CONTRIBUICAO CT
    ON TC.CPF = CT.TESTADOR
  JOIN CONSULTOR CS
    ON CS.CPF = CT.CONCONSULTOR
  JOIN (
    SELECT S.SERV_COD, S.CLIENTE, S.SITE
    FROM AVALIACAO A
      JOIN SERVICOS S
        ON A.SERVICO = S.SERV_COD
    ) SA
    ON SA.SERV_COD = CT.AVALIACAO
  JOIN CLIENTE CL
    ON CL.EMAIL = SA.CLIENTE
  JOIN SITE ST
    ON ST.URL = SA.SITE
ORDER BY CONSULTOR, ANO DESC, CT.NOTA DESC;
```

Essa consulta é utilizada para documentar o trabalho feito por cada consultor em cada ano que coordenou e orientou os testadores. Com ela, se produz um relatório das notas e comentários dados pelos testadores sob a supervisão dos consultores em cada serviço em que participaram, listando também o site e cliente de tal serviço.

#### 4.4 - Quantidade de mudanças que cada desenvolvedor fez em projetos que iniciaram no mês de junho em 2023

```
SELECT
  D.CPF,
  D.NOME,
  COUNT(S.SERV_COD) AS NUMERO_MUDANCAS
FROM DESENVOLVEDOR D
  JOIN MUDANCA M
    ON D.CPF = M.DESENVOLVEDOR
  JOIN ADAPTACAO A
    ON M.ADAPTACAO = A.SERVICO
  LEFT OUTER JOIN SERVICOS S
    ON A.SERVICO = S.SERV_COD
WHERE EXTRACT(MONTH FROM S.INICIO) = '06'
  AND EXTRACT(YEAR FROM S.INICIO) = '2023'
GROUP BY D.CPF, D.NOME
ORDER BY NUMERO_MUDANCAS;
```

Consulta a quantidade de mudanças que cada desenvolvedor aplicou nos projetos que iniciaram no mês de junho de 2023. Além do valor de documentação, essa consulta também é utilizada para fins organizacionais e de acompanhamento. O mês e ano podem ser facilmente iterados para acompanhar o trabalho dos desenvolvedores ao longo dos projetos iniciados em cada mês.

4.5 - Seleciona os nomes de todos os consultores que contribuíram em todas as avaliações que o testador Endrick participou

```
SELECT
    CS.CPF,
    CS.NOME
FROM CONSULTOR CS
JOIN CONTRIBUICAO CT
ON CS.CPF = CT.CONSULTOR,
(
    SELECT C.AVALIACAO
    FROM CONTRIBUICAO C
    JOIN TESTADOR T
    ON C.TESTADOR = T.CPF
    WHERE T.NOME = 'Endrick'
) TABLE2
WHERE CT.AVALIACAO = TABLE2.AVALIACAO
GROUP BY CS.CPF, CS.NOME
HAVING COUNT(CT.AVALIACAO) = (
    SELECT COUNT(AVALIACAO)
    FROM (
        SELECT C.AVALIACAO
        FROM CONTRIBUICAO C
        JOIN TESTADOR T
        ON C.TESTADOR = T.CPF
        WHERE T.NOME = 'Endrick'
    )
);
```

Essa consulta foi elaborada para selecionar os nomes de todos os consultores que contribuíram em todas as avaliações em que um dado testador participou. Essa necessidade surge devido à importância de identificar os consultores que estiveram envolvidos em todas as avaliações realizadas por tal testador, indicando uma alta consistência e desempenho desses consultores nas tarefas avaliadas, além de avaliar a coordenação e comunicação entre o testador e os funcionários da empresa.



## 5 - Aplicação

### 5.1 - Descrição Inicial

A aplicação utilizou o SGBD relacional Oracle disponibilizado pela docente. Além disso, foi empregada a linguagem Python para a criação da aplicação. Para a execução aplicação foi enviado com o arquivo setup.md para mais informações dos requisitos necessários para a execução da aplicação.

### 5.2 - Funcionalidades

#### 5.2.1 - Inserção de dados

A função "*insert*" permite que o usuário adicione novas tuplas à tabela "site" ao fornecer os atributos desejados. Isso oferece a possibilidade de inserir um novo site de forma simples e rápida, ampliando a funcionalidade de adição de dados à tabela.

```
def insert(params : list[str], flags : dict[str, any]):
    print('CADASTRANDO O NOVO SITE...')

    if 3 > len(params[0]):
        raise Exception(
            '[ERRO] Numero de parametros insuficiente.'
            + ' Para inserir um novo site, é preciso passar, respectivamente,'
            + ' URL, NOME e DONO.'
        )

    db.trans(
        'INSERT INTO SITE(URL, NOME, DONO) VALUES (:url, :nome, :dono)',
        {
            'url': params[0],
            'nome': params[1],
            'dono': params[2],
        },
    ).commit()

    print('Cadastro finalizado com sucesso!')
```

### 5.2.2 - Pesquisa de dados

A função "search" permite ao usuário pesquisar tuplas em um conjunto de dados por atributo específico, facilitando a recuperação de informações relevantes. É uma forma eficiente de encontrar resultados direcionados, melhorando a busca de dados.

```
def search(params : list[str], flags : dict[str, any]):
    print('BUSCANDO...')

    val = params[0] if params else None

    sql = 'SELECT NOME, URL, DONO FROM SITE WHERE '

    where = ['URL = :val'] if val else []
    params = {'val': val} if val else {}

    w, d = where_flags(['DONO', 'NOME'], flags)
    where = where + w
    params.update(d)

    if global_state['debug']:
        print('Where value:')
        print(where)
        print('Params value:')
        print(params)

    if not where: raise Exception('[ERRO] Os dados providos são insuficientes.')

    sql = sql + ' AND '.join(where)

    if global_state['debug']:
        print('SQL value:')
        print(sql)

    result = db.query(sql, params)
    readable_results = [f'{i}: {v}' for i, v in enumerate(result)]

    print('RESULTADOS: ')
    print('#: (URL, NOME, DONO)')
    print('\n'.join(readable_results))
```

### 5.2.3 - Remoção de dados

A função "*delete*" possibilita ao usuário a exclusão de tuplas de um conjunto de dados por meio de um atributo específico, tornando a remoção de dados mais fácil e direcionada.

```
def delete(params: list[str], flags : dict[str, any]):
    print('DELETANDO...')

    val = params[0] if params else None

    sql = 'DELETE FROM SITE WHERE '

    where = ['URL = :val'] if val else []
    params = {'val': val} if val else {}

    w, d = where_flags(['DONO', 'NOME'], flags)
    where = where + w
    params.update(d)

    if global_state['debug']:
        print('Where value:')
        print(where)
        print('Params value:')
        print(params)

    if not where: raise Exception('[ERRO] Os dados providos são insuficientes.')

    sql = sql + ' AND '.join(where)

    if global_state['debug']:
        print('SQL value:')
        print(sql)

    db.trans(sql, params).commit()

    print('Cadastro deletado com sucesso!')
```

## 6 - Conclusão

O projeto teve como objetivo desenvolver um banco de dados sobre inclusão digital, seguindo os processos de criação introduzidos no curso de banco de dados. Dessa forma, pudemos aplicar os conhecimentos adquiridos em sala de aula e ter uma visão de como a aplicação funcionaria na prática, seguindo todo o processo, desde a elaboração da ideia até a aplicação final.

O primeiro passo do projeto foi o desenvolvimento da ideia relacionada ao tema do trabalho, por meio de um brainstorming entre os integrantes do grupo. Em seguida, realizamos a criação da 'Descrição do Problema e dos Requisitos dos Dados', que foi a primeira etapa do projeto. Durante essa fase, encontramos algumas dificuldades na atribuição de todos os requisitos das relações e atributos de uma entidade.

Com a ideia pronta e documentada, o segundo passo foi a criação do Modelo Entidade-Relacionamento (MER) com base no texto desenvolvido. Essa parte do projeto nos permitiu visualizar a estrutura inicial do banco de dados, estabelecendo os relacionamentos entre as entidades e decidindo quais atributos seriam armazenados.

Em seguida, o terceiro passo foi transformar o MER em um modelo Relacional, para obtermos uma visão exata de como será o nosso banco de dados e termos em mente as sequências lógicas que serão seguidas para a elaboração de pesquisas mais eficientes.

Após a conclusão da modelagem, o último passo do projeto foi o mapeamento do Modelo Relacional para o SQL, inserindo tuplas com atributos fictícios para realizar consultas e comprovar nosso conhecimento na matéria. Foram feitas cinco consultas de nível médio/difícil, conforme aprendido em aula, além do desenvolvimento de uma aplicação simples em Python para conectar um usuário ao banco de dados.

Em suma, o projeto nos aproximou da criação do banco de dados, proporcionando desafios e conhecimentos que apenas o processo prático pode fornecer. Demonstrou a importância de cada etapa do desenvolvimento para obter um resultado eficaz, deixando claro como um banco bem modelado, com baixo nível de inconsistência, favorece sua utilização e manuseio.