



Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

SSC0240 - Bases de Dados (2023.1)

Docente: Profa. Dra. Elaine Parros M. de Sousa

Estagiário PAE: André Moreira Souza

BRIDGE

ACESSO E TREINAMENTO TECNOLÓGICO PARA COMUNIDADES

| | |
|-----------------------------------|----------|
| Danielle Modesti | 12543544 |
| Eduardo Figueiredo Freire Andrade | 11232820 |
| Hélio Nogueira Cardoso | 10310227 |
| Jhonathan Oliveira Alves | 11838116 |
| Pedro Dias Batista | 10769809 |

São Carlos, 1 de julho de 2023

Sumário

| | | |
|----------|----------------------------------------------------------------|-----------|
| 1 | Introdução | 1 |
| 1.1 | Descrição do Problema | 1 |
| 2 | Modelo Entidade-Relacionamento (MER) | 3 |
| 2.1 | Levantamento de requisitos de dados | 3 |
| 2.2 | Principais funcionalidades | 5 |
| 2.3 | Diagrama Entidade-Relacionamento (DER) | 7 |
| 2.4 | Restrições de integridade | 9 |
| 2.5 | Análise dos Ciclos Presentes no MER | 10 |
| 2.6 | Mudanças Relacionadas à Primeira Entrega | 13 |
| 3 | Modelo Relacional | 14 |
| 3.1 | Diagrama Relacional | 14 |
| 3.2 | Discussão Sobre os Mapeamentos Propostos | 16 |
| 3.3 | Observações Complementares sobre o Modelo Relacional | 26 |
| 3.4 | Mudanças Relacionadas à Segunda Entrega | 29 |
| 4 | Implementação | 30 |
| 4.1 | <i>Script</i> de consultas | 31 |
| 4.2 | Aplicação | 36 |
| 4.2.1 | Descrição da aplicação | 36 |
| 4.2.2 | Requisitos de sistema | 36 |
| 4.2.3 | Instruções para utilização | 37 |
| 4.2.4 | Inserção realizada | 38 |
| 4.2.5 | Consulta realizada | 38 |
| 5 | Conclusão | 39 |

1 Introdução

Uma das principais características do mundo atual é a ubiquidade [CHA22] da computação, isto é, sua forte presença no cotidiano das pessoas, seja na forma de internet das coisas, com assistentes virtuais como *Alexa* e *Siri* se comunicando com usuários em suas residências, controlando lâmpadas e televisões inteligentes, seja nos celulares, dispositivos acessados por milhões de pessoas e que as acompanham no meio doméstico ou empresarial. Nesse contexto, a despeito da revolução causada pela tecnologia na realidade humana e de seus muitos benefícios para o aumento da produtividade e da facilidade da intercomunicação, nem todos puderam acompanhar esta evolução, e isso se deve tanto à falta de recursos financeiros para adquirir um dispositivo tecnológico como à falta de oportunidades para adquirir habilidades com estas tecnologias.

1.1 Descrição do Problema

Dessa maneira, ações de inclusão digital são importantes em vários setores sociais, como, por exemplo: idosos com dificuldades para utilizar seus celulares, os quais são atualizados constantemente com novas funcionalidades das quais não usufruem; professores acostumados a dar aulas no quadro de giz aprendendo a ensinar em sistemas *online*; e pessoas de comunidades carentes precisando utilizar uma tecnologia que não possuem em suas casas e que talvez não saibam manipular.

Levando-se em consideração a demanda destas últimas, **propõe-se um sistema, de nome "*Bridge*", que auxiliará na administração de uma iniciativa filantrópica de uma instituição. Esta atua dentro de áreas carentes com suas diversas unidades, promovendo inclusão digital em três frentes: acesso, treinamento e doações.**

Para oferecer acesso a tecnologias digitais, a instituição filantrópica disponibiliza, em suas unidades, laboratórios com diversos destes equipamentos para que os usuários possam utilizá-los no local, podendo inclusive pedir auxílio a monitores que ficam disponíveis nestes laboratórios.

Nas unidades são também oferecidos treinamentos. Exemplos de treinamentos seriam: prover habilidades na formatação de documentos de texto para a elaboração de currículo; utilização de planilhas para organização financeira pessoal e de negócios próprios; utilização eficiente de *tablets* e *smartphones* para o acompanhamento de cursos *online*; entre outros. Para a ministração dos treinamentos são necessários professores capacitados.

Por fim, as unidades fazem a administração de doações. São aceitas doações de equipamentos eletrônicos, tais como computadores, *tablets* e impressoras. Pessoas com necessidade de equipamentos como estes podem se cadastrar como potenciais receptores das doações. Os equipamentos doados passam por uma triagem, após a qual podem ser destinados da melhor forma, tanto aos assistidos que as receberão como para a manutenção e expansão das próprias unidades (se houver necessidade).

O sistema *Bridge* possui dois usuários-alvo: os administradores das unidades e os assistidos da iniciativa filantrópica. Os administradores precisam zelar pelo oferecimentos dos treinamentos, coordenar a triagem de doações, gerenciar o cadastro de equipamentos, laboratórios, salas, professores e pessoas assistidas. Os assistidos do programa podem utilizar o sistema para se inscrever em treinamentos oferecidos (em qualquer unidade do programa) e solicitar doações de equipamentos tecnológicos.

2 Modelo Entidade-Relacionamento (MER)

2.1 Levantamento de requisitos de dados

A iniciativa da instituição filantrópica providenciará **unidades**, locais de sua propriedade, para acesso e treinamento da população assistida em algumas tecnologias básicas. As **unidades** possuem **funcionários**, que podem ser exclusiva ou simultaneamente **administradores**, **professores**, **monitores**, ou ter, ainda, outro tipo de cargo para manutenção da **unidade**. Cada uma das **unidades** é composta por várias **salas**, que devem ser **salas de aula** ou **laboratórios**. Nas **salas de aula**, haverá **oferecimentos** de **treinamentos** ministrados por **professores** neles certificados. Os **treinamentos** são ofertados pensando na capacitação em tecnologias digitais de pessoas das comunidades em que as **unidades** da instituição estão presentes. Estas pessoas são os **assistidos** da iniciativa filantrópica. Cada **assistido** pode participar de **oferecimentos** de **treinamentos** e certificar-se neles, assim como utilizar os **laboratórios**. Nos **laboratórios**, estarão disponíveis **equipamentos** para que os **assistidos** possam usar, assim como **monitores** para auxiliar no manejo destes **equipamentos**. Cada **laboratório** deve obrigatoriamente possuir pelo menos um **monitor** responsável por ele em algum turno de algum dia da semana. Um **assistido** também pode ser o potencial receptor de doações ao cadastrar uma **solicitação de doação**, com seu nível de necessidade de **equipamento**, em uma lista de espera com critérios socioeconômicos para recepção de **equipamentos** cadastrados e disponíveis para doação.

Administradores das **unidades** são pessoas que organizam o funcionamento destas e são responsáveis pelo gerenciamento de todo o sistema *Bridge*; também são especialmente responsáveis pelo gerenciamento das doações de **equipamentos**, por suas **solicitações** e por seus destinos (elas podem ser direcionadas às **salas de aula** e **laboratórios** da própria **unidade**, caso ali sejam

necessárias, ou a algum **assistido** que cadastrou sua **solicitação de doação**, representando uma pessoa que necessita daquele **equipamento** e espera numa fila até que ele esteja disponível para ela, por ordem de prioridade no sistema).

Cada **treinamento** é cadastrado com seu nome e conteúdo programático. Um **oferecimento** de um **treinamento** ocorre sempre em uma mesma **sala de aula**. Para cada um deles devem ser armazenados as datas e horários de início e fim, os dias da semana em que ocorre, respectivos horários de início e fim das aulas nestes dias da semana e a quantidade máxima de alunos que podem se inscrever nele. Após a conclusão de um **oferecimento**, um **assistido** pode solicitar uma certificação no respectivo **treinamento**, para a qual deve-se guardar a data.

Para um **assistido** devem ser guardados nome, CPF, data de nascimento, telefone, profissão, renda mensal base e a sua senha de acesso (ele acessa o sistema com seu CPF e sua senha). Também se deve guardar seu *e-mail* e endereço.

De um **equipamento** deve-se guardar o número serial, seu tipo e descrição. Este **equipamento** cadastrado pode ser exclusivamente direcionado para compor uma **sala** ou para atender a uma **solicitação de doação**. Cada **solicitação de doação** é feita por apenas um **assistido** (que pode fazer várias **solicitações de doação** diferentes). Para uma **solicitação de doação** deve-se guardar a data e a hora em que foi realizada, o grau de necessidade autodeclarada em uma escala de 0 a 10, em que 10 representa necessidade máxima e 0 a mínima, o tipo de equipamento que está sendo solicitado e quaisquer detalhes adicionais. A prioridade da **solicitação** é calculada dinamicamente em relação ao grau de necessidade, renda mensal do assistido e o tempo decorrido desde sua realização. Quando um **equipamento** cadastrado atende

a uma **solicitação**, deve-se guardar a data e a hora da doação para fins de histórico.

Uma **sala** possui capacidade, número e andar, além de seu tipo, que pode ser uma **sala de aula** ou **laboratório**. Uma **sala de aula** pode comportar **oferecimentos** de **treinamentos**. Um **laboratório** pode ser monitorado por um ou mais **monitores**. Um **monitor** pode monitorar mais de uma sala, mas apenas uma em um mesmo turno e dia da semana.

Para cada **unidade**, deve-se armazenar CNPJ, telefones, a comunidade assistida por ela (identificada, por exemplo, por um "bairro-alvo") e seu endereço. Para cada **funcionário** que nela trabalha, devem ser armazenados: endereço, CPF, nome, e-mail e cargo. Um **administrador** possui ainda a sua senha que, juntamente de seu CPF, permite-lhe acessar o sistema com seus privilégios para gerenciar a **unidade**.

2.2 Principais funcionalidades

Os usuários-alvo da aplicação, que são os administradores das unidades da instituição filantrópica e os assistidos, possuem diferentes funcionalidades a sua disposição. Dentre elas, para cada um, tem-se:

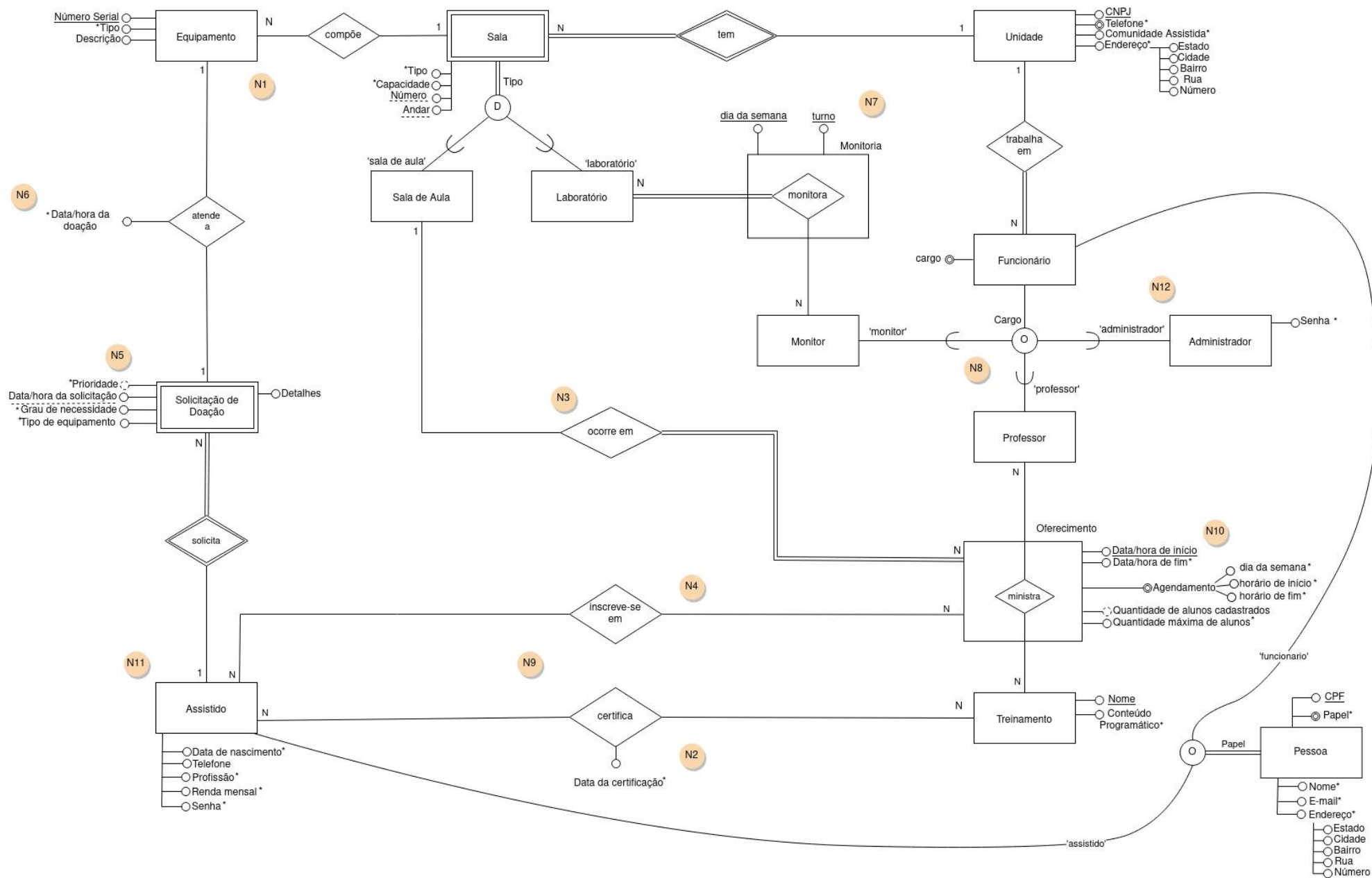
- Administradores de unidade:
 - Inserção (cadastro), alteração e remoção de assistidos no sistema;
 - Inserção, alteração e remoção de equipamentos disponíveis;
 - Inserção, alteração e remoção de unidades;
 - Inserção de salas, cadastrando-as como salas de aulas ou como laboratórios, sua alteração e remoção;

- Inserção, alteração e remoção de funcionários (professores, monitores, administradores, dentre outros);
 - Inserção, alteração e remoção de treinamentos;
 - Inserção, alteração e remoção de solicitações de doação;
 - Inserção, alteração e remoção de oferecimentos de treinamentos;
 - Inserção, alteração e remoção de monitorias;
 - Inserção, alteração e remoção do vínculo da doação de um equipamento a um assistido receptor potencial;
 - Busca por equipamentos disponíveis para doação de um determinado tipo;
 - Busca por solicitações de equipamentos de um determinado tipo em ordem de prioridade;
 - Busca por equipamentos que satisfazem alguma solicitação de doação, e a quais assistidos eles pertencerão;
 - Busca por oferecimentos ativos em um certo intervalo de datas em determinados dias da semana;
 - Busca por quantos alunos concluíram um oferecimento a partir de quantos estavam cadastrados inicialmente.
- Assistidos:
 - Acesso (com *login*) ao sistema dos equipamentos das salas, a fim de utilizá-los;

- Cadastro (pessoal) em um oferecimento de treinamento;
- Busca por oferecimentos de treinamentos que ainda não começaram;
- Busca por oferecimentos de treinamentos que não começaram e ainda há vagas (turma não lotada);
- Consulta aos treinamentos nos quais já está certificado em ordem cronológica de data de certificação;
- Consulta aos oferecimentos ativos nos quais está inscrito;
- Consulta às solicitações de doação que fez;
- Remoção de inscrição em oferecimentos de treinamentos que está atualmente cursando.

2.3 Diagrama Entidade-Relacionamento (DER)

Com base nos requisitos de dados e na modelagem conceitual do problema, foi elaborado um Diagrama Entidade-Relacionamento, o qual se encontra na página seguinte.



2.4 Restrições de integridade

O Diagrama Entidade-Relacionamento (DER) elaborado na seção 2.3 possui indicações de anotação referentes a restrições de integridade do modelo. Estas restrições estão indicadas a seguir com seus respectivos números identificados.

- **N1:** Um equipamento está apenas relacionado a uma sala (para compô-la) ou a uma solicitação de doação (para atendê-la);
- **N2:** Um assistido só pode ser certificado em um treinamento se já concluiu um oferecimento deste;
- **N3:** Uma mesma sala pode ter apenas um oferecimento ocorrendo nela em cada janela de horário;
- **N4:** Um assistido só pode se inscrever em um oferecimento cuja quantidade de alunos já cadastrados não tenha atingido a quantidade máxima de alunos para o respectivo treinamento ou a capacidade máxima da sala de aula em que ocorre;
- **N5:** A prioridade é calculada com base nos atributos renda mensal, grau de necessidade do equipamento e no tempo decorrido desde a solicitação;
- **N6:** Atributos com a indicação * são obrigatórios na aplicação;
- **N7:** Identificador de Monitoria é a composição de dia da semana, turno e CPF do monitor. Note-se que turno só pode assumir valores 'manhã', 'tarde' e 'noite';
- **N8:** Caso o funcionário tenha a especialização de professor e monitor, eles não podem ter um oferecimento e uma monitoria no mesmo horário;

- **N9:** Um assistido não pode se inscrever em dois oferecimentos que sejam no mesmo horário, nem em um oferecimento em cujo treinamento já seja certificado;
- **N10:** A identificação de oferecimento é feita pela composição do CPF do professor que o ministra e de sua data/horário de início. É importante que a data/horário de início corresponda à data e ao horário da primeira aula e que essa aula ocorra num dia da semana e horário que pertença a algum agendamento do oferecimento em questão. Outro ponto relevante é que a aplicação deve tratar para que oferecimentos diferentes que estejam ativos em um mesmo conjunto de datas não aconteçam na mesma sala em horários sobrepostos;
- **N11:** E-mail também é identificador de assistido;
- **N12:** E-mail também é identificador de funcionário.

2.5 Análise dos Ciclos Presentes no MER

- **Ciclo: Equipamento → Solicitação de Doação → Assistido → Treinamento → Oferecimento → Sala de Aula**

Há a presença de um ciclo de dependência entre as entidades Equipamento, Solicitação de Doação, Assistido, Treinamento, Oferecimento e Sala de Aula, considerando os relacionamentos “atende a”, “solicita”, “inscreve-se em”, “ocorre em” e “compõe”, pois um equipamento só pode ser doado (atender a uma solicitação) se ele não compõe uma sala, mas, na verdade, é destinado à doação. Além disso, uma vez que ele é doado, não pode mais compor salas da unidade. Este ciclo não pôde ser desfeito, pois há semânticas diferentes para os relacionamentos feitos pelo equipamento; se não tratado na aplicação, permite que um equipamento seja doado a

um assistido e possa compor uma sala da unidade, o que não faz sentido. Similarmente, permitiria que o equipamento que já compõe uma sala da unidade seja doado.

- **Ciclo: Assistido → Oferecimento → Treinamento**

Há a presença de um ciclo de dependência entre as entidades Assistido, Oferecimento e Treinamento, observando os relacionamentos “inscreve-se em” e “certifica”, visto que um assistido só pode se certificar após ter sido relacionado a um oferecimento daquele treinamento em questão e concluindo-o, tornando-se apto para ser certificado. Ele não pôde ser desfeito e, se não for tratado na aplicação, permitirá com que alunos que não cursaram o oferecimento se tornem certificados em um treinamento. Outro dependência neste ciclo é que um assistido não deve também poder se inscrever num oferecimento em que já está certificado, para que não chegue a consumir recursos da iniciativa sem maior necessidade.

- **Ciclo: Unidade → Monitor → Laboratório**

Há a presença de um ciclo de dependência, que não pôde ser removido, entre as entidades Unidade, Monitor e Laboratório, considerando os relacionamentos “trabalha em”, “monitora” e “tem”, pois o monitor só pode monitorar laboratórios pertencentes à unidade da qual ele é funcionário.

- **Ciclo: Unidade → Professor → Oferecimento → Sala de Aula**

Há a presença de um ciclo de dependência entre as entidades Unidade, Professor, Oferecimento e Sala de Aula, com os relacionamentos “trabalha em”, “ocorre em” e “tem”, com uma justificativa análoga à do ciclo anterior, apenas se tratando de outra especialização do funcionário: o professor de uma unidade precisa (depende de) oferecer um treinamento em uma sala de aula daquela unidade em específico.

- **Ciclo: Unidade → Professor → Treinamento → Assistido → Oferecimento → Sala de Aula**

Há a presença de um ciclo entre as entidades Unidade, Professor, Treinamento, Assistido, Oferecimento e Sala de Aula, envolvendo os relacionamentos "trabalha em", "ministra", "certifica", "inscreve-se em", "ocorre em" e "tem". Tal ciclo engloba dependências em ciclos já analisados (o segundo, Assistido → Oferecimento → Treinamento; e o quarto, Unidade → Professor → Oferecimento → Sala de Aula). Nesse sentido, ao tratá-los, este ciclo está automaticamente resolvido.

- **Ciclo: Unidade → Professor → Treinamento → Assistido → Solicitação de Doação → Equipamento → Sala**

Há a presença de um ciclo entre as entidades Unidade, Professor, Treinamento, Assistido, Solicitação de Doação, Equipamento e Sala, com os relacionamentos "trabalha em", "ministra", "certifica", "solicita", "atende a", "compõe" e "tem". Este ciclo é o mais externo e, ao se resolverem todos os outros indicados acima, com destaque ao primeiro ciclo (pela dependência envolvendo Equipamento, Solicitação de Doação, Assistido, Oferecimento e Sala de Aula), ele estará tratado.

2.6 Mudanças Relacionadas à Primeira Entrega

Algumas observações foram feitas pelo monitor da disciplina e exigiram pequenas modificações da seção de Modelo Entidade-Relacionamento (2).

- A primeira modificação foi na seção de restrições de integridade 2.4, mais especificamente na nota N4. Reformulou-se a nota para que o número de alunos inscritos em um oferecimento de treinamento não superasse, além do número máximo de alunos para o treinamento, também a capacidade máxima da sala em que ocorre.
- Outra modificação foi no próprio DER, na seção 2.3. A partir de agora, considera-se que um assistido possa, eventualmente, se tornar um funcionário de uma unidade. Para evitar inconsistências e redundâncias, decidiu-se criar uma entidade 'Pessoa' mais genérica que se especializa em um papel, que pode ser exclusiva ou simultaneamente 'Assistido' ou 'Funcionário'.

3 Modelo Relacional

Nesta seção, realiza-se o mapeamento do modelo conceitual desenvolvido para o modelo lógico-relacional. Para isso, em certos casos, é necessário tomar algumas decisões dentre as opções possíveis de mapeamento, discutindo vantagens e desvantagens.

3.1 Diagrama Relacional

O diagrama segue na próxima página, exclusivamente.



3.2 Discussão Sobre os Mapeamentos Propostos

Seguem ponderações acerca da tradução do modelo conceitual para o relacional, quando existentes (elas não são explicitadas quando o mapeamento se determina, geralmente, de uma única maneira, como ocorrem com entidades fracas e com atributos compostos, por exemplo). Cada justificativa está numerada com um número correspondente a uma indicação em círculo laranja no diagrama do modelo relacional presente na seção 3.1.

1. Atributo multivalorado 'Telefone' do conjunto de entidades Unidade

- **Solução adotada:** os telefones de 'Unidade' foram mapeados como atributos, ou seja, colunas desta tabela, optando-se por armazenar um máximo de apenas dois valores, representando dois telefones possíveis para cada uma das unidades. Fez-se isso por ser o cenário mais comum para a aplicação. Uma decisão complementar em relação a este ponto foi a de tornar pelo menos um telefone obrigatório para cada unidade registrada, devido à semântica da aplicação.
- **Vantagens:** em relação à solução alternativa, a consulta torna-se mais eficiente, pois não é necessário realizar uma junção. Em vez disso, recuperam-se os telefones diretamente da tabela Unidade.
- **Desvantagens:** o número de telefones que cada unidade no sistema pode ter está limitado a dois. Além disso, quando uma unidade não possuir algum dos telefones, será armazenado um valor nulo.
- **Alternativas:** poderia ser criada uma tabela à parte, Telefone, vinculando cada unidade a seus múltiplos telefones.

2. Especialização do conjunto de entidades 'Sala'

- **Solução adotada:** para a especialização disjunta e obrigatória de 'Sala' em 'Laboratório' e 'Sala de Aula', foram criadas tabelas tanto para o conjunto de entidades genéricas (CEG) 'Sala' quanto para os conjuntos de entidades específicas (CEEs) 'Laboratório' e 'Sala de Aula', visto que todas participam de relacionamentos separadamente, com sua própria importância no modelo.

Além disso, foi adicionado o atributo de critério (AtC) 'tipo' na tabela do CEG com restrição NOT NULL, a fim de garantir que cada instância de 'Sala' será especializada, ou seja, terá valor para seu tipo. Deve-se aplicar a cláusula CHECK para que o tal valor seja, obrigatoriamente, 'sala de aula' ou 'laboratório'.

- **Vantagens:** os dados estão semanticamente encapsulados, o que também ajuda no mapeamento dos relacionamentos entre os conjuntos de entidades, e as replicações de atributos do CEG são reduzidas. Além disso, as consultas podem ser realizadas com maior enfoque em cada conjunto de entidades.
- **Desvantagens:** apesar das restrições de NOT NULL, do AtC definido na CEG e do uso da cláusula CHECK, não é possível garantir, a nível de esquema, a especialização total e disjunta (obrigar todas as salas a serem, de forma mutuamente exclusiva, sala de aula ou laboratório). Isso deve, portanto, ser garantido na aplicação. Outro ponto negativo é a necessidade de junção (operação custosa) para recuperar informação de salas de aula ou laboratórios.
- **Alternativas:** apesar de haver diversos procedimentos de mapeamento para uma especialização, julgou-se que:

- manter apenas a tabela do CEG era inviável, devido à participação dos CEEs em outros relacionamentos e à importância deles para a aplicação;
- manter apenas as tabelas dos CEEs (alternativa mais provável) também não era uma boa decisão, pois, apesar de essa opção garantir participação total, as salas por si só já têm devida importância na aplicação e relacionamentos próprios, o que ficaria prejudicado.

3. Especialização do conjunto de entidades 'Pessoa'

- **Solução adotada:** optou-se por mapear esta especialização apenas nos CEEs 'Assistido' e 'Funcionário'. Nesse sentido, embora um assistido possa, eventualmente, tornar-se um funcionário, não se espera que tal caso seja comum na aplicação, o que justifica não ter sido feita uma tabela para o atributo de critério.
- **Vantagens:** diminuição do custo de junção, dada a inexistência da tabela para o CEG 'Pessoa'. Além disso, garante-se a especialização total de 'Pessoa' em nível de esquema (toda pessoa deve ser um assistido e/ou um funcionário).
- **Desvantagens:** replicação de dados quando um assistido também assumir o papel de funcionário para esses raros casos. Disto, decorrem redundâncias e possíveis inconsistências.
- **Alternativas:** apesar de haver diversos procedimentos de mapeamento para uma especialização, julgou-se que, além de não garantirem participação total:
 - manter apenas a tabela do CEG era inviável, pois, apesar de conter diversos atributos, esta não seria efetivamente utilizada no sistema, apresentando-se apenas para fins conceituais e semânticos;

- além do mesmo motivo do CEG 'Pessoa' não ter utilização no sistema, a alternativa de mapear tanto a tabela do CEG como a dos CEEs também seria desvantajosa pela necessidade de junções nas consultas aos CEEs. É desejável que este tipo de custo seja evitado na aplicação em questão devido à existência de uma especialização cascadeada de 'Pessoa', seguindo para 'Funcionário', que segue nos tipos de funcionário.
- há ainda a alternativa (mais provável) de manter as tabelas dos CEEs, adicionando-se, ainda, uma outra tabela, referente ao atributo de critério. Isso, apesar de aumentar a consistência da base, aumentaria também os custos de junção, mas é esperado que casos nos quais uma pessoa incorpora um assistido e um funcionário simultaneamente sejam raros.

4. Especialização do conjunto de entidades 'Funcionário'

- **Solução adotada:** para a especialização sobreposta e parcial de 'Funcionário' em 'Monitor', 'Professor' e 'Administrador', foram criadas tabelas tanto para o CEG 'Funcionário' quanto para os CEEs, visto que todas participam de relacionamentos separadamente e/ou têm uma importância singular no modelo. Além disso, o atributo de critério (AtC) 'cargo' foi mapeado em uma tabela separada, 'FuncionarioCargo' - decidiu-se que ele era imprescindível para a aplicação, na qual os cargos de um funcionário não são pré-conhecidos e definem relações importantes.
- **Vantagens:** os dados estão semanticamente encapsulados, o que também ajuda no mapeamento dos relacionamentos entre os conjuntos de entidades, e as replicações de atributos do CEG são reduzidas. Além disso: as consultas podem ser realizadas com maior enfoque em

cada conjunto de entidades, e não há necessidade de varrer todas as tabelas de 'Monitor', 'Professor' e 'Administrador' para conhecer os cargos de um determinado funcionário.

- **Desvantagens:** necessidade de junção para recuperar informação de monitores, professores e administradores.
- **Alternativas:** apesar de haver diversos procedimentos de mapeamento para uma especialização, julgou-se que:
 - manter apenas a tabela do CEG seria inviável, porque os CEEs fazem parte de relacionamentos separadamente;
 - manter apenas a tabela dos CEEs também seria inviável, porque a participação é parcial e o CEG possui relacionamentos próprios;
 - a outra opção mais provável seria a de manter tabelas para o CEG e para os CEEs, mas descartando a tabela referente ao AtC. Isso não seria vantajoso, pois não se conhece, a priori, os cargos assumidos por um funcionário, havendo necessidade de varrer todas as tabelas de CEEs pelo identificador de um dado funcionário para conhecer todos os seus cargos.

5. Conjunto de Relacionamentos 'Monitora' entre Laboratório e Monitor - N:N

- **Solução adotada:** o conjunto de relacionamentos (CR) 'monitora' foi mapeado por meio da tabela 'Monitoria', que representa a agregação.
- **Vantagens:** menor custo de junções que a solução alternativa.
- **Desvantagens:** a participação total dos laboratórios não pode ser garantida a nível de esquema (isso deve ser feito na aplicação).
- **Alternativas:** como o CR é N:N, uma nova tabela já deveria ser

criada. O que poderia ser feito, adicionalmente, seria uma outra nova tabela para representar apenas o CR 'monitora', independente da agregação 'Monitoria'. Isso causaria maior custo de junção de forma desnecessária, pois nem mesmo há atributos no CR 'ministra'.

6. Conjunto de Relacionamentos 'Ministra' entre Professor e Treinamento - N:N

- **Solução adotada:** o CR 'ministra' foi mapeado por meio da tabela 'Oferecimento', que representa a agregação.
- **Vantagens:** menor custo de junções que a solução alternativa.
- **Desvantagens:** não foram encontradas desvantagens nessa abordagem.
- **Alternativas:** análogo ao que foi apontado no ponto de justificativa 5; ou seja, uma tabela poderia representar apenas o CR 'ministra', enquanto outra representaria a agregação 'Oferecimento', causando custo desnecessário de junção.

7. Atributo derivado 'Quantidade de Alunos Cadastrados' da agregação Oferecimento

- **Solução adotada:** optou-se por armazenar a quantidade de alunos cadastrados em um oferecimento como atributo, ou seja, coluna, da tabela 'Oferecimento'.
- **Vantagens:** as potenciais operações sobre o atributo são simples (incrementos e decrementos a cada cadastro ou cancelamento de cadastro de aluno). Além disso, não é necessário buscar todos os alunos da tabela de 'Assistido' que se relacionam com o oferecimento em

questão a cada consulta.

- **Desvantagens:** não foram encontradas grandes desvantagens para esse mapeamento; há apenas um pequeno gasto de memória (armazenamento de dado do tipo inteiro).
- **Alternativas:** a outra opção seria simplesmente não armazenar o dado na base de dados, permitindo seu cálculo dinâmico na aplicação, a cada consulta. Isso teria um gasto muito grande de processamento em relação à solução escolhida.

8. Atributo multivalorado 'Agendamento' da agregação Oferecimento

- **Solução adotada:** optou-se por criar uma tabela à parte, Agendamento, vinculando cada oferecimento a seus múltiplos agendamentos.
- **Vantagens:** não serão armazenados valores nulos (o que poderia ocorrer com a solução alternativa) e há maior flexibilidade do número de agendamentos possíveis para um mesmo treinamento dentro de uma semana.
- **Desvantagens:** maior gasto de memória com a criação da tabela e aumento de junções para consultar os agendamentos de um oferecimento.
- **Alternativas:** a outra possibilidade seria armazenar uma determinada quantidade de agendamentos como colunas da tabela 'Oferecimento', mas isso limitaria a quantidade de oferecimentos referentes a um treinamento a cada semana, armazenando valores nulos caso não fossem preenchidos todos os agendamentos estipulados para o oferecimento.

9. Conjunto de Relacionamentos 'Atende a' entre Solicitação de Doação e Equipamento - 1:1

- **Solução adotada:** no CR 'atende a' entre 'Solicitação de Doação' e 'Equipamento', optou-se por colocar o 'Equipamento' como chave estrangeira e chave secundária na tabela de 'Solicitação de Doação'; também foi colocado o atributo do relacionamento na tabela de 'Solicitação de Doação'. Isto pois haverá muitos equipamentos que não irão atender a nenhuma solicitação, visto que muitos fazem parte de uma sala e não serão sequer considerados para doação. Por outro lado, pretende-se atender a todas as solicitações com algum equipamento em algum momento.
- **Vantagens:** a solução escolhida minimiza a quantidade de memória não utilizada, visto que diminui a quantidade de atributos nulos. Além disso, concentrar todas as informações sobre a doação na tabela de 'Solicitação de Doação' torna as consultas mais eficientes e economiza junções, uma vez que as consultas mais frequentes estarão focadas nessa tabela.
- **Desvantagens:** todas as solicitações que ainda não foram atendidas terão valores nulos armazenados.
- **Alternativas:** duas alternativas eram possíveis:
 - criar uma tabela à parte para o relacionamento, o que possibilitaria a diminuição de valores nulos armazenados. Entretanto, ao longo do tempo, muitas solicitações podem ser atendidas (e o banco em questão guarda histórico delas). Dessa forma, a participação não seria baixa em relação ao relacionamento e, para consultar a solicitações, seriam necessárias mais junções, não compensando o custo;

- colocar 'Solicitação de Doação' como chave estrangeira e secundária na tabela de 'Equipamento', o que não foi feito, pois muitos dos equipamentos não podem ser doados, visto que fazem parte de uma sala. Dessa maneira, a quantidade de atributos nulos ficaria muito grande (mais espaço inutilizado).

10. Conjunto de Relacionamentos 'Ocorre em' entre Oferecimento e Sala de Aula - 1:N

- **Solução adotada:** optou-se por mapear 'Sala de Aula' na tabela de 'Oferecimento', colocando seus identificadores como chave estrangeira NOT NULL pela participação total.
- **Vantagens:** a participação total é garantida, sem ser necessário criar uma tabela extra.
- **Desvantagens:** não foram identificadas desvantagens nesse mapeamento.
- **Alternativas:** não foram encontradas soluções alternativas para esse relacionamento de participação total, mas considerou-se necessário comentar a decisão.

11. Conjunto de Relacionamentos 'Compõe' entre Equipamento e Sala - 1:N

- **Solução adotada:** mapearam-se os identificadores de 'Sala' como chave estrangeira composta na tabela de 'Equipamento'.
- **Vantagens:** as consultas são mais eficientes, por não precisar de junções.
- **Desvantagens:** possibilidade de nulos quando o equipamento não

compõe uma sala.

- **Alternativas:** a outra possibilidade seria criar uma tabela a mais para o relacionamento, mas isso gastaria mais espaço e necessitaria de junções, definindo algo que só seria viável fazer se houvesse baixa participação no relacionamento, o que não é o caso.

12. Conjunto de Relacionamentos 'Trabalha em' entre Funcionário e Unidade - 1:N

- **Solução adotada:** optou-se por mapear 'Unidade' na tabela de 'Funcionário', colocando seus identificadores como chave estrangeira NOT NULL pela participação total.
- **Vantagens:** a participação total é garantida, sem ser necessário criar uma tabela extra.
- **Desvantagens:** não foram identificadas desvantagens nesse mapeamento.
- **Alternativas:** não foram encontradas soluções alternativas que mantivessem os mesmos resultados que a solução adotada, especialmente em relação à participação total.

13. Atributo Derivado 'Prioridade' do conjunto de entidades Solicitação de Doação

- **Solução adotada:** optou-se por não armazenar a prioridade de uma solicitação de doação como atributo em sua tabela, preferindo calculá-la sob demanda, visto que o atributo deve, na aplicação, ser atualizado diariamente. Além disso, o cálculo só será feito para solicitações ainda em aberto.

- **Vantagens:** sem gasto de memória para armazenar o atributo na base e valor sempre atualizado/consistente.
- **Desvantagens:** o cálculo deste atributo deve ser feito a cada consulta que o envolve, apesar de não ser tão computacionalmente caro.
- **Alternativas:** a outra possibilidade seria armazenar o atributo na tabela 'Solicitação de Doação', mas as múltiplas atualizações sobre ele tornam mais viável a abordagem utilizada.

3.3 Observações Complementares sobre o Modelo Relacional

Seguem comentários relevantes para a compreensão completa da solução apresentada e suas restrições:

- Reforça-se que o atributo 'Turno' de 'Monitoria', o qual faz parte da chave primária desta, só pode assumir os valores 'manhã', 'tarde' e 'noite'. Isto deixa claro que não se trata de um problema de data/hora, em que se teria de tratar intervalos. Para restringir esses valores, pode-se utilizar a cláusula CHECK.
- Ainda é preciso garantir separadamente que um equipamento deve estar apenas relacionado a uma sala, para compô-la, ou a uma solicitação de doação, para atendê-la. No mapeamento feito, isto significa que para se cadastrar um equipamento numa tupla da tabela 'SolicitaçãoDoação', é necessário que os campos deste equipamento referentes à sala ('Unidade', 'Andar' e 'Sala') sejam nulos. O contrário também deve ser checado: se um equipamento está cadastrado como parte de uma solicitação atendida, não deve ser permitido adicionar valores para os campos referentes à sala.

- O atributo de relacionamento 'DataHoraDoação' foi anotado como obrigatório no modelo conceitual, pois considerou-se importante que, ao se atender uma solicitação, registrasse-se a data desta doação. No entanto, como este atributo foi mapeado dentro da tabela de solicitação de doação, não se poderia colocar o atributo 'DataHoraDoação' como NOT NULL. Isto porque, quando da criação da tupla de solicitação, ela não será imediatamente atendida, não tendo, portanto, uma data de doação. Logo, é importante garantir que, ao se inserir um equipamento numa tupla de solicitação de doação, o que representa a doação em si, garanta-se que seja cadastrada também a data desta doação.
- Um 'Assistido' só pode ser certificado em um 'Treinamento' se já concluiu um 'Oferecimento' deste. No mapeamento realizado, isto representa que só se poderá cadastrar uma tupla para a tabela 'Certifica' se o correspondente 'Assistido' possuir registrado na tabela de 'InscreveSeEm' alguma tupla referente a um 'Oferecimento' do respectivo 'Treinamento' cuja 'DataHoraFim' já passou.
- Se um 'Assistido' já estiver vinculado a um 'Treinamento' na tabela de 'Certifica', ele não deve poder se inscrever na tabela 'InscreveSeEm' em um 'Oferecimento' deste mesmo 'Treinamento'.
- Um 'Assistido' não pode se inscrever na tabela de 'InscreveSeEm' em dois 'Oferecimentos' diferentes que ocorram em horários sobrepostos.
- A quantidade de alunos inscritos em um 'Oferecimento' é iniciada com 0 quando da sua criação. A cada inscrição neste 'Oferecimento', seu número é incrementado. A cada cancelamento de inscrição, este número é decrementado.
- Dois 'Oferecimentos' que ocorrem em algum intervalo de tempo em co-

mum não podem ocorrer em intervalos de horário sobrepostos no mesmo dia da semana, especialmente na mesma 'Sala de Aula', nem com o mesmo 'Professor' provendo tais 'Oferecimentos' em horários sobrepostos. Isto deve ser tratado para que não ocorra ao criar um 'Oferecimento'.

- Uma tupla só pode ser adicionada à tabela de 'InscreveSeEm', se o número de alunos inscritos no 'Oferecimento' correspondente não tiver atingido o número máximo de alunos do 'Treinamento' respectivo, nem a capacidade da 'Sala de Aula' em que ele ocorre.
- Caso um 'Funcionário' possua na tabela de 'CargoFuncionário' o atributo 'Cargo' como 'professor' e 'monitor' em tuplas diferentes (possue estes dois cargos simultaneamente), tem-se de garantir que ele não possa prover um 'Oferecimento' e dar uma 'Monitoria' em horários sobrepostos (esta checagem terá de ser feita entre turno e horários da devida forma).
- O atributo 'DataHoraInício' de 'Oferecimento' diz respeito à data/hora do início da primeira aula de todas as aulas de um oferecimento. A identificação de um oferecimento é feita pela composição deste atributo 'DataHoraInício' e do 'Professor' que ministrar este oferecimento. Isto ajuda a esclarecer os atributos da tabela de 'Agendamento'. Uma tupla de agendamento possui 'Professor' e esta 'DataHoraInício' como parte de sua chave primária, mas também como chave estrangeira para a tabela de 'Oferecimento'. O atributo 'HoraInício' de uma tupla de agendamento diz respeito ao horário de início daquele agendamento do respectivo oferecimento. Isto funciona junto dos atributos 'HoraFim' e 'DiaDaSemana'. Para exemplificar, pode ser que um determinado oferecimento cuja primeira aula começará na data/hora '07/08/2023 - 14:00', possua dois agendamentos semanais: um é às segundas-feiras, de 14:00 às 15:00, outro é de quartas-feiras, de 16:00 às 17:00. Veja que, neste exemplo, para total

consistência, o dia 07/08/2023 tem de ser uma segunda ou uma quarta-feira (no caso, uma segunda-feira), e o horário desta primeira aula tem de corresponder ao horário de início do agendamento deste dia da semana (no caso, 14:00).

3.4 Mudanças Relacionadas à Segunda Entrega

Algumas considerações sobre a seção foram feitas pelo monitor da disciplina e, por isso, dois pontos de justificativa foram modificados para maior completude da seção de Discussão sobre os Mapeamentos Propostos (3.2).

- A primeira modificação foi na discussão do mapeamento da especialização do CE 'Sala' (ponto de justificativa 2), mais especificamente no tópico das desvantagens. Complementou-se a discussão para indicar que, além da especialização total, a disjunção (exclusividade mútua) também não poderia ser garantida na estrutura (ou seja, não há como garantir que uma mesma sala não seja tanto uma sala de aula como um laboratório), apesar das restrições, da cláusula CHECK e do atributo de critério definido em 'Sala' para evitar que isso aconteça.
- Outra modificação foi na discussão do mapeamento da especialização do CE 'Funcionário' (ponto de justificativa 4), mais especificamente no tópico da solução adotada. A explicação da decisão foi corrigida (o AtC não foi colocado na mesma tabela do CEG, mas em uma tabela separada, 'FuncionárioCargo') para ser coerente com os demais tópicos de discussão. Por causa do equívoco neste tópico, o Diagrama Relacional da seção 3.1 também precisou corrigir esse mapeamento, que estava duplo (atributo de critério estava em tabela separada, o que era correto, mas também na tabela de 'Funcionário', como atributo 'Cargo'). Foi removido o atributo 'Cargo' do CE 'Funcionário' para corrigir isto.

4 Implementação

Para finalizar o trabalho, pede-se para desenvolver um protótipo simples do sistema para o usuário final, com funcionalidades relevantes para o contexto de inclusão digital (conceito que envolve garantia de acesso e habilidades para usar tecnologias digitais).

Para tanto, nesta seção, serão descritos os processos de implementação do sistema desenvolvido durante a disciplina, bem como a implantação da base de dados da plataforma, chamada de *Bridge*. Nesta etapa final de implementação, confeccionou-se o esquema da base de dados (todas as tabelas envolvidas, com seus tratamentos de consistência) e dados para alimentar essa base. Ademais, cinco consultas foram formuladas para teste e um protótipo inicial, integrado à base de dados, foi proposto pelo grupo.

O protótipo foi implementado na linguagem **Python** e, para a base de dados, utilizou-se o Sistema Gerenciador de Banco de Dados (SGBD) **PostgreSQL**. O repositório com o conteúdo produzido durante esta parte do projeto pode ser consultado no *GitHub*, em [Eduardofig/Bridge/](https://github.com/Eduardofig/Bridge/). Ainda assim, este terceiro envio do projeto inclui, além do presente relatório, os códigos referentes a ele (código-fonte da aplicação e *scripts* de esquema, dados e consultas SQL).

4.1 Script de consultas

1. Consultar todos os alunos que completaram todos os treinamentos oferecidos por um determinado professor (Danielle Modesti)

```
01 | SELECT A.CPF, A.NOME, A.EMAIL FROM
02 | ASSISTIDO A WHERE NOT EXISTS (
03 |     (
04 |         SELECT O.TREINAMENTO FROM
05 |         OFERECIMENTO O JOIN FUNCIONARIO F ON F.CPF = O.PROFESSOR
06 |         WHERE F.NOME = 'Danielle Modesti'
07 |     )
08 |
09 | EXCEPT
10 |
11 |     (
12 |         SELECT CE.TREINAMENTO FROM
13 |         CERTIFICA CE
14 |         WHERE CE.ASSISTIDO = A.CPF
15 |     )
16 | );
17 |
```

Nesta consulta, envolve-se o operador de divisão, realizando uma **divisão relacional**, conceito adquirido nos estudos de Álgebra Relacional (AR). Para isso, utiliza-se o operador de EXCEPT, que realiza subtração entre conjuntos. Na consulta acima, para que sejam identificados todos os alunos que completaram os treinamentos oferecidos por um determinado professor, deve-se, primeiramente, recuperar todos os oferecimentos de treinamento oferecidos por esse professor (no nosso exemplo, o professor tem o nome 'Danielle Modesti'), tratando-se de um tipo de consulta aninhada não correlacionada; posteriormente, seleciona-se os treinamentos nos quais cada um dos assistidos é certificado (esta é uma consulta

aninhada correlacionada, custosa, porém necessária).

Dessa maneira, um assistido será selecionado para compor a resposta da consulta caso o resultado da operação de divisão relacional seja conjunto vazio: isto acontece quando um assistido se certifica em, no mínimo, todos os cursos de um dado professor (podendo ter cursado mais). Caso ele não tenha cursado todos, não será escolhido para compor a resposta, que resulta nos CPFs e e-mails de cada assistido que passou na seleção (são identificadores de assistidos, então devem vir na resposta para fazer sentido), além de seus nomes para uma análise visual mais semântica.

2. Listar solicitações não atendidas, ordenadas por grau de prioridade, além da cidade dos assistidos que fizeram a solicitação

```
01 | SELECT SOL.ASSISTIDO, SOL.DATA_HORA_SOL, A.ESTADO, A.CIDADE, (  
    GRAU_NECESSIDADE * 1000 / EXTRACT(EPOCH FROM DATA_HORA_SOL)) AS  
    PRIORIDADE  
02 | FROM SOLICITACAO_DOACAO SOL  
03 | JOIN ASSISTIDO A  
04 | ON A.CPF = SOL.ASSISTIDO  
05 | WHERE EQUIPAMENTO IS NULL  
06 | ORDER BY PRIORIDADE DESC;  
07 |
```

Nesta consulta, calcula-se a prioridade com base no horário de solicitação da doação de um determinado equipamento e no grau de necessidade autodeclarado da solicitação. Este cálculo de prioridade foi feito para definir uma fórmula para a prioridade, mas na aplicação real ela seria calculada de uma forma mais completa. Para tratar com os horários de solicitações, foi utilizado o EXTRACT dentro de uma expressão - definiu-se uma coluna virtual de prioridade, visto que tal atributo modelado como derivado não foi armazenado no banco (optou-se por computá-lo dinamicamente);

adicionalmente, uma junção foi necessária para vincular o assistido à solicitação feita por ele, e a cláusula ORDER BY foi necessária para apresentar dados selecionados de forma ordenada e descendente para o usuário (da maior para a menor prioridade).

3. Selecionar quantidade de agendamentos atualmente existentes em uma determinada unidade, agrupando contagem por dia da semana

```
01 | SELECT
02 | CASE
03 |     WHEN A.DIA_SEMANA = 0 THEN 'Domingo'
04 |     WHEN A.DIA_SEMANA = 1 THEN 'Segunda'
05 |     WHEN A.DIA_SEMANA = 2 THEN 'Terca'
06 |     WHEN A.DIA_SEMANA = 3 THEN 'Quarta'
07 |     WHEN A.DIA_SEMANA = 4 THEN 'Quinta'
08 |     WHEN A.DIA_SEMANA = 5 THEN 'Sexta'
09 |     WHEN A.DIA_SEMANA = 6 THEN 'Sabado'
10 | END AS DIA_DA_SEMANA,
11 | COUNT(HORA_INICIO) AS QTD_AGENDAMENTOS FROM
12 | OFERECIMENTO O LEFT JOIN AGENDAMENTO A ON O.PROFESSOR = A.PROFESSOR
13 | AND O.DATA_HORA_INICIO = A.DATA_HORA_INICIO
14 | WHERE O.UNIDADE = '12345678000200'
15 | AND CURRENT_DATE BETWEEN O.DATA_HORA_INICIO AND O.DATA_HORA_FIM
16 | GROUP BY DAY_NAME;
17 |
```

Nesta consulta, utilizou-se o CASE para formatar o dia da semana, uma vez que, para o banco em questão, um dia da semana é armazenado com um número de 0 a 6, correspondendo respectivamente de domingo (0) a sábado (6). Foi utilizada a função de agregação COUNT para contar quantos agendamentos de oferecimentos de treinamentos estão ativos

na data corrente, por dia da semana, considerando a unidade cujo CNPJ é '12345678000200'. Na resposta para o usuário, é apresentado o dia da semana em formato mais semântico (como *string*), e uma segunda coluna para a quantidade de agendamentos naquela unidade para aquele dia da semana.

4. Consultar todos os assistidos que não fizeram nenhum treinamento de um conjunto de treinamentos especificados

```
01 | SELECT A.NOME FROM
02 | ASSISTIDO A WHERE
03 | A.CPF NOT IN
04 | (
05 |     SELECT ISE.ASSISTIDO FROM
06 |     INSCREVE_SE_EM ISE JOIN OFERECIMENTO O
07 |     ON O.PROFESSOR = ISE.PROFESSOR AND O.DATA_HORA_INICIO = ISE.
    INICIO_OFERECIMENTO
08 |     WHERE UPPER(O.TREINAMENTO) IN ('Treinamento de Excel Parte 1', '
    Treinamento de Excel Parte 2')
09 | );
10 |
```

Nesta seleção de dados, utiliza-se uma consulta aninhada não correlacionada para buscar os assistidos cadastrados que não participaram de nenhum oferecimento de 'Treinamento de Excel Parte 1' nem de 'Treinamento de Excel Parte 2'.

5. Recuperar todos os professores que oferecem pelo menos uma determinada quantidade de oferecimentos em uma determinada unidade; listar, também, a quantidade média de alunos nestes oferecimentos

```
01 | SELECT F.NOME, O.PROFESSOR, AVG(QTD_ALUNOS) AS MEDIA_DE_ALUNOS, COUNT
    | (O.TREINAMENTO) AS QTD_OFERECIMENTOS FROM OFERECIMENTO O RIGHT JOIN
    | FUNCIONARIO F ON F.CPF = O.PROFESSOR
02 | WHERE O.UNIDADE = '12345678000100'
03 | GROUP BY (F.NOME, O.PROFESSOR)
04 | HAVING COUNT(O.TREINAMENTO) >= 2;
05 |
```

Nesta consulta, além das funções de agregação COUNT e AVG, foi empregada também a cláusula HAVING, a fim de selecionar, após agrupamento por professor da unidade de CNPJ '12345678000100', quais deles já ofereceram pelo menos dois treinamentos.

4.2 Aplicação

4.2.1 Descrição da aplicação

Trata-se de um *shell* interativo, criado na linguagem *Python*, que possui funcionalidade para inserção de registro de unidade no banco e busca de uma unidade por CNPJ. A aplicação foi integrada ao banco de dados através da biblioteca *Psycopg2*, que oferece uma interface de comunicação entre o *Python* e o *PostgreSQL*.

4.2.2 Requisitos de sistema

A aplicação foi desenvolvida e testada em ambiente *Linux* — mais especificamente, *Ubuntu 22.04 jammy* e *Mint 21.1 vera*. Foi utilizada a versão 3.10.6 do *Python* para execução da aplicação, e versões 14.8 e 15.3 do *PostgreSQL* para o banco de dados. É importante também a instalação prévia da biblioteca '*psycopg2*' no *Python*, a qual foi utilizada em sua versão 2.9.6 para o desenvolvimento. Por fim, é necessário ter instalado o módulo '*python-dotenv*'.

4.2.3 Instruções para utilização

Tendo o banco de dados funcionando, com o esquema carregado (disponível no arquivo *esquema.sql* no repositório), assim como os dados (disponível no arquivo *dados.sql* no repositório), basta adicionar as informações de conexão no arquivo *.env*, que deve se encontrar no mesmo diretório do código *bridge.py* da aplicação (disponível no repositório na pasta *src*). Um exemplo de arquivo *.env* é o seguinte:

```
01 | DB_HOST=localhost
02 | DB_PORT=5432
03 | DB_NAME=bridge
04 | DB_USER=<your_user>
05 | DB_PASSWORD=<your_password>
```

Adicione endereço da conexão em *'DB_HOST'*, a porta da conexão com o banco de dados em *'DB_PORT'*, o nome do banco em *'DB_NAME'*, O usuário do banco em *'DB_USER'* e, por fim, a senha deste usuário em *'DB_PASSWORD'*.

Agora, pelo terminal, com tudo configurado, rode o comando *'python3 bridge.py'* no diretório em que o código da aplicação se encontra. Siga o passo a passo para utilização que o próprio sistema fornece.

4.2.4 Inserção realizada

A inserção que foi realizada é a de uma unidade, com suas informações de:

- CNPJ (o qual deve ser composto de 14 dígitos);
- Nome da comunidade que a comunidade auxilia;
- Estado brasileiro em que a comunidade se encontra (selecionar dentre opções válidas apresentadas);
- Cidade em que a unidade se encontra;
- Bairro da unidade;
- Rua da unidade;
- Número do endereço da unidade;
- Um telefone da unidade (obrigatório);
- Segundo telefone da unidade (opcional).

O código para esta inserção foi feita na rotina `'insert_unidade()'` do código `'bridge.py'` disponível na pastas `'src'` do depósito.

4.2.5 Consulta realizada

A busca implementada foi a de busca de unidade por CNPJ. Basta entrar um CNPJ válido que a aplicação irá buscar no banco de dados as informações desta unidade, caso ela exista, ou informar que a unidade não foi encontrada, caso ela não exista na base de dados.

O código para esta inserção foi feita na rotina `'buscar_unidade_por_cnpj(cnpj)'` do código `'bridge.py'` disponível na pastas `'src'` do depósito.

5 Conclusão

No que tange à primeira etapa do projeto, sobre escolha do tema voltado à inclusão digital, modelagem no MER e representação com o DER: o grupo rapidamente chegou em consenso sobre o *Bridge* ao incrementar ideias mais básicas; as reuniões para elaboração do DER foram numerosas e um pouco cansativas, pois, muitas vezes, os membros do grupo precisavam revisar notações, possibilidades de utilização e suas semânticas (principalmente em questão de cardinalidade, dependência existencial e identificador primário), mas, paulatinamente, o sistema se encaminhou e se tornou mais claro para nós; tivemos bastante receio de possíveis incongruências sérias que afetariam as etapas seguintes e, sob essa ótica, toda a solicitude do monitor foi essencial para validação do que foi produzido, além da teoria passada pela docente, sempre destacando erros comuns e particularidades inerentes a cada construtor sintático.

Conforme o nosso DER se expandiu, porém, surgiu a dificuldade de analisar ciclos - nesse ponto, acreditamos que as aulas poderiam ter sido melhor ilustradas, com mais discussões e exemplos (de identificação e de resolução), para um entendimento sólido.

Em segundo lugar, a parte dois do projeto teve uma teoria dada em sala de aula de forma mais rápida, o que, em alguns momentos, fez com que o grupo tivesse uma visão um pouco não familiar das notações e das interpretações do modelo relacional; essa dificuldade inicial pôde ser superada quando o grupo se tornou mais experiente na tradução entre diagrama entidade-relacionamento e diagrama relacional, via exercícios propostos e avaliações na disciplina. Ademais, o projeto só pôde ser melhor observado nas perspectivas de modelo relacional quando estudou-se especialização, pela alta dependência desta abstração de dados na modelagem do sistema *Bridge*. Neste ponto, iniciou-se uma etapa de muita discussão ao redor dos mapeamentos possíveis, mas o grupo

acredita ter chegado a decisões convenientes, observando as necessidades da aplicação. É válido ressaltar, finalmente, que os esclarecimentos do monitor da disciplina foram novamente essenciais para validação das discussões e solução de dúvidas.

Quanto à etapa 3 do projeto, em última instância, foi nesta etapa que o grupo foi capaz de compreender, na prática aplicada, os conhecimentos adquiridos durante a disciplina. Deste modo, notou-se que toda a construção e o desenvolvimento mais teórico do início e do meio do semestre foram extremamente necessários e úteis para a menor dificuldade na linguagem SQL. A única sugestão sobre essa parte mais prática seria a de ter mais tempo para as práticas e dúvidas a respeito dela, pois as consultas possíveis com SELECT são muito diversificadas e o tempo dedicado a aprendê-las não foi tão grande. De qualquer maneira, o grupo pôde, por meio do trabalho, familiarizar-se bastante com a própria semântica da aplicação, durante o processo de alimentação da base com dados e de consulta a eles; foi possível, também, aprender a comunicar o banco com APIs de aplicação, em um bom exemplo de integração de conceitos em computação.

Finalmente, o grupo agradece mais uma vez a disponibilidade do monitor em tirar todas as dúvidas a respeito do trabalho, tanto presencialmente como *online*. O projeto, para nós, foi uma forma excelente e proveitosa para um primeiro contato com Bases de Dados.

Referências

- [CHA22] CHARLEAUX, Lupa. O que é computação ubíqua? <https://canaltech.com.br/produtos/o-que-e-computacao-ubiqua-1406/>, 2022.
- [ELM03] ELMASRI, R. e NAVATHE, S. B. *Fundamentals of Database Systems*. Pearson, 2003.
- [SIL01] SILBERSCHATZ, A., KORTH, H. e SUDARSHAN, S. *Database System Concepts*. McGraw-Hill Education, 2001.