

UNIVERSIDADE DE SÃO PAULO
DEP. DE CIÊNCIAS DA COMPUTAÇÃO
SCC0640 - BASES DE DADOS

Elaine Parros Machado de Souza

AUXÍLIO A DEPENDENTES QUÍMICOS E MORADORES DE RUA
PROJETO DE BASE DE DADOS - SCC0640

Emanuel Lace Aranha, n° 11234224
Felipe de Alcantara Tomé, n° 11800970
Francisco Reis Nogueira, n° 11954374
Joao Augusto Fernandes Barbosa, n° 11953348

São Carlos

2022

Sumário

1	Introdução	2
2	Levantamento de requisitos	2
2.1	<i>Descrição da aplicação prática da base de dados</i>	2
2.2	<i>Principais funcionalidades</i>	4
2.3	<i>Análise de inconsistências</i>	5
3	Modelo Entidade-Relacionamento expandido	6
4	Mudanças realizadas em relação a primeira entrega	7
5	Modelo Relacional	9
5.1	<i>Imagem do Modelo Relacional</i>	9
5.2	<i>Restrições de integridade</i>	10
5.3	<i>Justificativas para a modelagem</i>	10
6	Mudanças realizadas em relação a segunda entrega	14
7	Implementação	15
7.1	<i>Script de Consultas</i>	16
7.2	<i>Aplicação</i>	18
7.2.1	Requisitos do Sistema	19
7.2.2	Funcionamento geral	19
8	Conclusão e considerações finais	23

1 Introdução

Uma das principais problemáticas encaradas dentro do contexto sociopolítico urbano, principalmente dentro de megalópoles, é a gritante desigualdade encontrada por entre as entranhas de suas ruas, ocasionando em uma quantidade gritante de cidadãos sem acesso ao básico para sua sobrevivência. A origem desta situação é, por muitas vezes, definida erroneamente como vinda da dependência química, entretanto os motivos são diversos mas, fato é, que em regiões específicas de grandes centros urbanos existe uma concentração enorme de pessoas sem moradia fixa que fazem uso de algum entorpecente com frequência. Este quadro força que a abordagem de acolhimento em abrigos e em assistências comunitárias tenham maior sensibilidade quanto a isso, necessitando a observação dos dados de evolução do auxiliado, dos recursos financeiros, dos voluntários e das campanhas de auxílio.

O seguinte projeto se trata de uma iniciativa dos moradores do centro de uma grande metrópole para reestruturação da região da forma mais humanizada possível, essencialmente teremos um **auxiliado**, o qual será recebido por este coletivo, classificado por auto-declaração e atendido da forma mais personalizada possível.

2 Levantamento de requisitos

2.1 Descrição da aplicação prática da base de dados

A entidade principal desta base será o **auxiliado**, representando o **morador de rua** e/ou **dependente químico** assistido, pois ele será o ponto focal do projeto, o qual possui todos os identificadores comuns de um cidadão brasileiro como **CPF**, desta forma o **funcionário** da ação social poderá registrar todas as informações pessoais e específicas através desta chave identificadora, e.g, algum tipo de formação ou **habilidade** profissional, **idade**, a qual será um atributo derivado de **Data de nascimento**, **sexo**, **nome** e, caso aplicável, **local** de moradia fixo. Isto é feito para que as **frentes** do projeto consigam se organizar a fim de encaminhar o **auxiliado** para os programas de **reestruturação profissional** em **empresas parceiras** e de **reestruturação social** em **clínicas** com sensibilidade geográfica, de faixa etária e de lotação das instituições pois elas possuem um **número máximo de pacientes e funcionários**, a fim de funcionar corretamente e promover uma reerguida confortável aos auxiliados. Ademais, a generalização feita através desta entidade faz com que ocorra uma simplificação quando é necessário atribuir os **auxiliados** a programas focalizados, referenciando apenas as entidades específicas **dependente químico** e **morador de rua**, possibilitando a classificação de ambos através

do atributo multi-valorado **tipo**.

As instalações físicas da ação com **local** fixo, chamadas de **centros comunitários**, são pontos organizacionais que se encarregam de administrar o próprio fluxo financeiro através do atributo **caixa**, representando todo o dinheiro atualmente em posse do **centro comunitário**. Todo centro tem o próprio **presidente**, representado pela relação **preside** entre **funcionário** e **centro comunitário**, seus funcionários participantes representados pela relação **participa**, **CNPJ** único e as frentes que são administradas por ele. De forma que, cada **centro comunitário** realiza seus próprios eventos beneficentes com **data**, **hora** e **local** específicos, além de campanhas de subsídio de produtos de uso pessoal através da **tesouraria**.

Os casos de dependência química são tratados através da internação não compulsória do auxiliado em **clínicas** parceiras que promovam programas de **reabilitação**, caso ela aconteça se faz interessante manter a **presença** do **auxiliado**, assim como a **data** de início do tratamento. Por questões organizacionais e de respaldo legal, toda **clínica** deve ter um **CNPJ** válido, um **nome** e ter em registro seu **número total de funcionários e pacientes**(atributos **NuFuncionarios** e **NuPacientes**), vale ressaltar que a **clínica** não recebe somente pacientes dos **centros comunitários**.

Todos os **auxiliados** tem a possibilidade de ingressar em um esquema de **emprego supervisionado** caso queiram, estes acordos são firmados entre os ingressantes e as **empresas parceiras** com a ajuda da frente de **reestruturação profissional**. Os auxiliados são contratados caso o numero máximo de funcionários (atributo **MaxFuncionarios**) das empresas parceiras seja diferente do número atual de funcionários (atributo **NuFuncionarios**) e conforme o interesse da empresa.

O projeto possui acordo com ONG's, que oferecem especificamente alojamento temporário para os **moradores de rua** assistidos pelo projeto, representado pelo relacionamento **aloja**. Além das relações específicas dos **terceiros** supracitadas, todas as especificações desta classe generalizada, incluindo qualquer **pessoa física** (funcionário ou não), podem realizar doações para os centros, podendo ser **monetárias** ou de **produto**, desde esta seja feita formalmente e possua uma **nota fiscal**.

Os **eventos** realizados pela ação serão para recrutamento de novos **funcionários**, conscientização dos moradores da região e para divulgação e captação de novos auxiliados. Os **nomes** dos **eventos**, o **local** de realização e a **data** os identificarão em uma chave composta , i.e, ao passo que podem existir dois eventos na mesma data com o mesmo nome, presume-se que eles não estão acontecendo no mesmo local (seriam o mesmo evento duplicado).

As entidades **doação**, **emprego supervisionado** e **reabilitação** foram modeladas

como agregações, pois são inerentes a relações específicas dentro do MER, possuindo atributos que facilitam na organização e categorização destas relações e a criação de tabelas referentes a elas serão de extremo proveito para o futuro da base de dados.

2.2 Principais funcionalidades

As funcionalidades são dependentes do tipo de usuário que está acessando o banco de dados e estão discriminadas abaixo para cada um deles.

- Empresa parceira
 1. Registra vagas de emprego supervisionado, as administra e modifica;
 2. Consulta a listagem de auxiliados, em específico os atributos habilidade e local;
 3. Registra e altera seu número máximo e atual de funcionários;
 4. Consulta a listagem de doações realizadas.
- Clínica
 1. Registra e administra o número máximo e atual de pacientes;
 2. Registra as possíveis vagas para a reabilitação e os programas de reabilitação;
 3. Atualiza a presença dos pacientes nos próprios programas de reabilitação;
 4. Consulta a listagem de dependentes, em específico as características relevantes para a definição da saúde deles, como a idade e seus vícios;
- Funcionário
 1. Registra terceiros e atualiza suas informações;
 2. Registra empresas parceiras e atualiza suas informações;
 3. Registra novos eventos, novas campanhas de divulgação para estes e os administra;
 4. Consulta a listagem de todas as instâncias das demais entidades;
 5. Registra novas doações;
 6. Registra e administra informações dos pontos físicos e administrativos de atendimento (centros comunitários)
 7. Realiza atribuição e administração das frentes de cada centro, além de listar os funcionários atribuídos em cada frente

- ONG
 1. Registro de vagas de alojamento;
 2. Consulta a listagem de auxiliados.

2.3 Análise de inconsistências

- **Ciclo Funcionário → Centro comunitário → Funcionário**

Através das relações **preside** e **participa** forma-se um ciclo entre **funcionário** e **centro comunitário**, a possível inconsistência neste caso se dá caso um **funcionário** venha a presidir um **centro comunitário** que ele não faz parte, o que não se encaixaria no funcionamento idealizado do projeto. Quanto ao ciclo, poderia ser substituído ao definir-se um atributo "função" dentro da relação de "participa". A nossa abordagem visou garantir um maior destaque ao presidente, caso em algum momento note-se ações/funções que apenas ele poderia exercer.

- **Possível inconsistência no número de funcionários por frente**

É necessário que pelo menos um **funcionário** participe de cada frente, entretanto é necessário dentro da aplicação real que o funcionário seja atribuído administrativamente a um **centro comunitário** pois as frentes, como conceito abstrato puramente organizacional, não possuem informações geográficas.

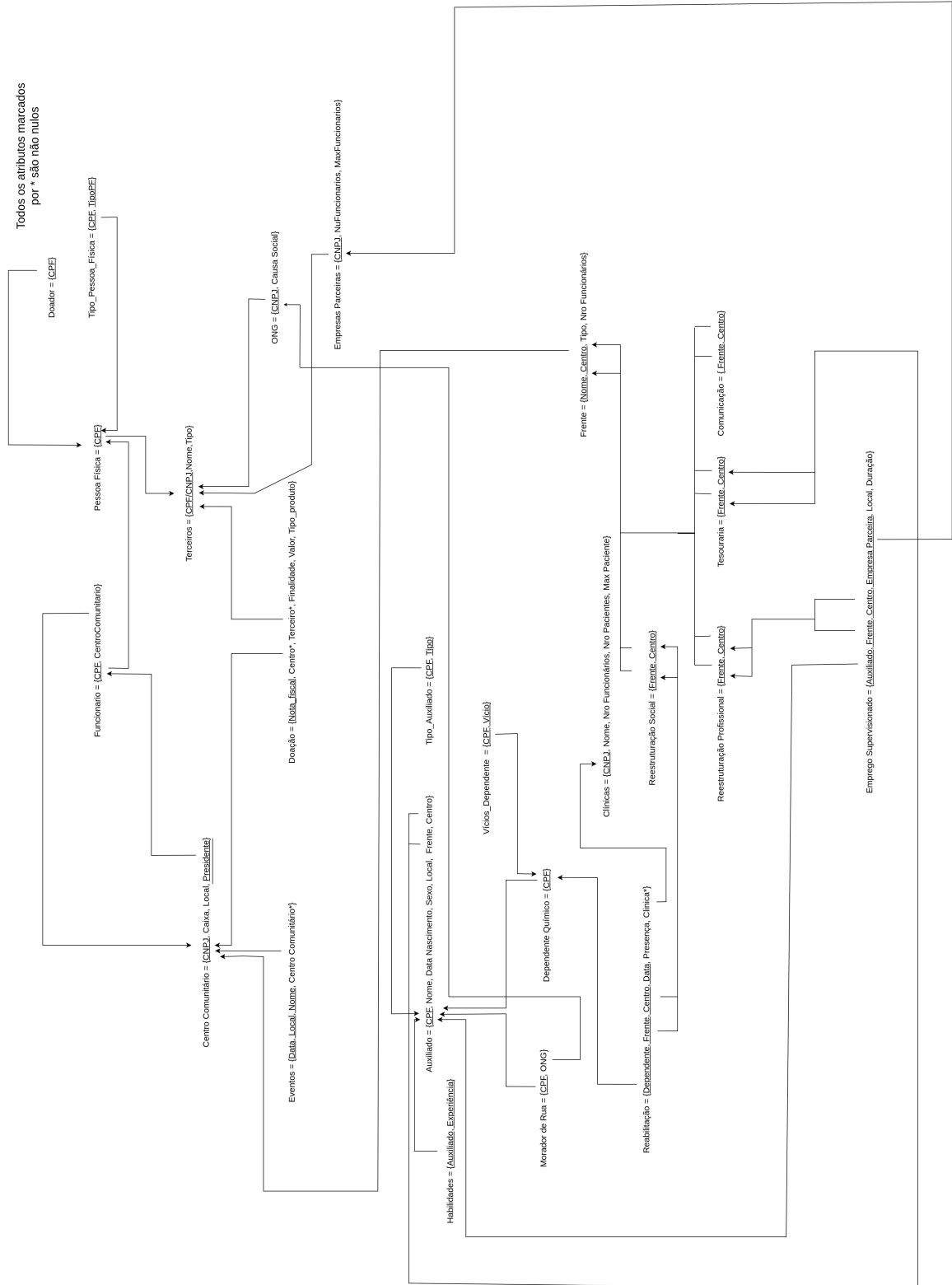
4 Mudanças realizadas em relação a primeira entrega

1. Agregações: A modelagem apresentada anteriormente tinha alguns problemas de apresentação. Boa parte das ligações referente às agregações não possuíam as conexões feitas de maneira correta - muitas vezes, a linha não se juntava ao losango interior. Assim, atualizamos o arquivo permitindo o melhor entendimento dessas entidades.
2. Auxiliado (Dependentes químicos ou moradores de rua): Foi realizada a correção da cardinalidade da entidade morador de rua com a relação de alojamento, uma vez que é mais semântico permitir que uma ONG aloje diversos moradores de rua.
3. Centro Comunitário: Foi corrigida a modelagem de presidente enquanto atributo. Neste caso, este evento foi recriado como uma relação entre centro comunitário e funcionário. Neste ponto, é importante destacar a criação de um ciclo. Aqui, faz-se indispensável que o presidente presida apenas um centro comunitário do qual faz parte.
4. Doação: A definição de doação torna-se mais clara caso seja modelada como múltiplos itens - como produtos ou monetária. Anteriormente, havia-se criado uma entidade fraca "Patrocínios". Contudo, essa nova abordagem atua de forma mais limpa e direta com os objetivos do trabalho. Com isso, modificamos também sua chave, que agora, ao invés de precisar depender das chaves de suas partes, poderá ser apenas a sua nota fiscal, a qual permite a presença tanto de valores monetários quanto de produtos.
5. Frentes de ação (Reabilitação, Reestruturação Profissional): A chave fraca de frente foi modelada como o seu "nome". Isso se deve ao analisar o próprio meio empresarial, onde diversas empresas possuem nomes diferentes para frentes de mesma atuação. Por exemplo, a gestão de recursos humanos em algumas é chamada de RH, em outras, "People". Contudo, elas são basicamente do mesmo tipo. Poderia-se tentar modelar sendo o próprio tipo como chave e atributo de especialização, mas isso geraria impactos negativos na modelagem relacional, tendo essa solução apresentada como uma boa maneira de contornar-se o problema. Foi pensado também a definição da chave parcial de frente como a sua assinatura eletrônica, porém, isso a faria ter um identificador único. Semanticamente, é mais correto deixar a frente como dependente do centro, por isso opta-se pela modelagem como entidade fraca.

6. Terceiros: A utilização de ID's não é recomendada para o MER. Neste caso, foi sugerida uma abordagem em que tanto CPF quanto CNPJ poderiam atuar como chave de Terceiros. Para isso, o valor informado seria tratado na aplicação, através de um RegEx - expressão regular. Nesta, haveriam regras de entrada para determinar o tipo da chave. Além disso, deveria também se ter uma análise à fim de garantir que o uso de CPF seja exclusivo de pessoas físicas, enquanto o uso de CNPJ seria apenas para empresas parceiras e ONG's.
7. Eventos: Anteriormente, a modelagem de um evento tendo apenas a chave como seu nome impedia que diversos centros possuísem eventos com o mesmo nome ou que um mesmo centro comunitário refizesse esse evento em outras edições. Agora, sua chave possui nome, data e local. Com isso, evita-se conflitos que poderiam surgir anteriormente e garante maior liberdade ao centro.
8. Descrição de entidades e relacionamentos: Foram expandidas e melhor explicadas as entidades e relacionamentos presentes no MER visando a coesão, a coerência e o maior detalhamento, a fim de facilitar a leitura e entendimento do trabalho.

5 Modelo Relacional

5.1 Imagem do Modelo Relacional



5.2 Restrições de integridade

1. Possível inconsistência nas especializações totais (Doação, auxiliado)

A modelagem utilizada não garante uma especialização total dessas partes, sendo necessária atenção especial no processo de implementação.

2. Possível inconsistência na chave de terceiro

Como supracitado, a chave necessita ser CPF para Pessoas físicas e CNPJ para ONGS/Empresas parceiras. Nesse ponto, o tratamento ficará a cargo da implementação, sendo sugerida a utilização de RegEx em parceria com regras de entrada para garantir a funcionalidade.

3. Ciclo Presidente → Centro comunitário

O entendimento correto desse ciclo demonstra que o Presidente de um centro comunitário deve ser um funcionário participante do mesmo.

5.3 Justificativas para a modelagem

A seguir estão listadas as justificativas tomadas para a criação do modelo relacional utilizando o modelo entidade relacionamento desenvolvido no tópico anterior.

1. Modelagem 1:1 (Preside)

Solução Adotada: Devido a participação total, a melhor alternativa encontrada foi referenciar, dentro de centro comunitário, um presidente através de uma chave estrangeira com a presença do indicador de not null.

Desvantagens: À priori, essa maneira é a ideal de representar a relação, não possuindo desvantagens evidentes.

Solução Alternativa: Poderia-se criar uma tabela à parte, com tupla de centro e funcionário. Sendo centro chave primária e presidente chave secundária não nula, o efeito final seria semelhante, mas haveria maior processamento.

2. Modelagem 1:N

Solução Adotada: Nestes casos, opta-se por modelar normalmente a entidade com cardinalidade "1" e inseri-la dentro da entidade com cardinalidade N. Em casos onde há participação total, essa inserção é acompanhada de um not null.

Vantagem: Em relacionamentos com participação total, essa solução garante que esse campo nunca será nulo na hora de inserção em tabela.

Desvantagem: Em situações como a relação de alojamento ou subsidia, onde não

há participação total, a entidade que engloba a outra poderá conter campos vazios. Por se tratar de pouca informação, não é um empecilho realmente grande.

Solução Alternativa: Poderia-se criar uma nova tabela referenciando por chave estrangeira cada uma das entidades. Contudo, trata-se de uma ação muito exagerada - devido a criação de tabela, indexação de valores - para reduzir um problema que à priori é muito pouco impactante (na parte relativa ao "Aloja"), por essa razão, opta-se pela abordagem realizada.

3. Mapeamento da agregação Reabilitação

Solução Adotada: Criação de uma tabela contendo as chaves da clínica, frente (que no caso, herda a também a chave de Centro, por ser entidade fraca) e do dependente químico.

Vantagem: Permite a modelagem e unicidade dos tratamentos de forma idealizada, permitindo manter um controle da presença.

Desvantagens: Como possui uma entidade fraca, o resultado final é muito sujo, por possuir muitas chaves estrangeiras.

Solução alternativa: A modelagem poderia possuir um ID, reduzindo esses custos. Contudo, a presença das chaves estrangeiras torna mais semântico o entendimento da agregação e não é um impacto realmente grande a ponto de necessitar de uma troca.

4. Mapeamento da agregação Emprego Supervisionado

Solução Adotada: Criação de uma tabela contendo as chaves da empresa, frente (que no caso, herda a também a chave de Centro, por ser entidade fraca) e do auxiliado.

Vantagem: Permite a modelagem e unicidade dos tratamentos de forma idealizada, permitindo manter um controle da atuação do auxiliado.

Desvantagens: Como possui uma entidade fraca, o resultado final é muito sujo, por possuir muitas chaves estrangeiras.

Solução alternativa: A modelagem poderia possuir um ID, reduzindo esses custos. Contudo, a presença das chaves estrangeiras torna mais semântico o entendimento da agregação e não é um impacto realmente grande a ponto de necessitar de uma troca.

5. Mapeamento da agregação Doação

Solução Adotada: Criação de uma tabela com chave primária sendo nota fiscal. Há também a referência tanto ao Centro como ao terceiro, mas à parte da chave

principal.

Vantagem: A utilização de uma nota fiscal aumenta a eficiência do sistema e, desta maneira, a agregação criada permite o controle de forma rápida e direcionada das doações recebidas pelo centro.

Desvantagens: Não há.

Solução Alternativa: O conjunto poderia ser identificado pelo doador, centro comunitário e data/hora, por exemplo. Contudo, a pesquisa ficaria mais lenta e não há realmente uma razão para realizar a troca de nota fiscal.

6. Mapeamento especialização Terceiros, Frente:

Solução Adotada: Criação de uma tabela geral, contendo a chave primária e atributos gerais. Criação de uma tabela específica, contendo os atributos específicos e, por fim, criação de uma tabela de controle, herdando a chave da entidade geral para tentar garantir seu disjoint.

Vantagem: Por conta das entidades específicas possuírem relações próprias, faz-se necessário sua modelagem. A criação de uma tabela de controle facilita a busca e torna o projeto mais eficiente

Desvantagens: As entidades específicas possuem poucos ou nenhum atributo próprio, tornando um pouco desperdício sua modelagem à parte. Todas elas são bem semelhantes em seus dados, podendo haver também uma repetição muito grande. Em alguns casos, com um erro por parte da inserção bem exagerado, poderia-se criar uma situação onde o Disjoint não fosse garantido.

Soluções Alternativas: A modelagem podia consistir de uma criação apenas das entidades específicas, sem a criação de uma geral. Contudo, haveria muita repetição de relacionamentos, uma vez que cada uma deveria replicar as relações da entidade geral retirada.

7. Mapeamento especialização Auxiliado:

Solução Adotada: Criação de uma tabela geral, contendo a chave primária e atributos gerais, bem como o atributo de especialização. Criação de uma tabela específica, contendo os atributos específicos.

Vantagem: Por conta das entidades específicas possuírem relações próprias, faz-se necessário sua modelagem. Evita a criação de uma tabela extra de controle

Desvantagens: As buscas podem se tornar um pouco lentas, por não se ter uma correlação direta entre Auxiliado e Tipo, como anteriormente usado na tabela de

controle. Em discussão com o grupo, concluímos que essa maneira era mais limpa e valeria a pena mesmo assim. Não garante também a especialização total.

Soluções Alternativas: A modelagem poderia seguir os padrões anteriores, utilizando uma tabela de controle. No fim, a indexação talvez não fizesse valer a pena, haja vista que essa consulta além de não ser tão utilizada, não fica lenta o suficiente na modelagem usada para ser necessário sua mudança.

8. Mapeamento da especialização Doação

Solução Adotada: Unificação de todos os atributos das entidades específicas na tabela única doação.

Vantagem: A utilização de menos tabelas faz com que o tempo de busca seja favorecido, pois não será necessário acessar duas tabelas distintas e é diminuído o tamanho em armazenamento da base de dados.

Desvantagens: Pode gerar muitos valores nulos caso várias doações de apenas um tipo, monetária ou de produto, sejam realizadas.

Solução Alternativa: Construir uma tabela para cada entidade específica, possuindo uma chave estrangeira como chave principal, a qual seria o número de nota fiscal da doação referente, além de uma tabela para categorizar os tipos, também tendo o número da nota fiscal como chave estrangeira e chave primária da tabela, que é composta com tipo.

9. Mapeamento da especialização Pessoa Física

Solução Adotada: Criação de uma tabela geral, contendo a chave primária e atributos gerais, bem como o atributo de especialização. Criação de uma tabela específica, contendo os atributos específicos.

Vantagem: Por conta das entidades específicas possuírem relações próprias, faz-se necessário sua modelagem. Evita a criação de uma tabela extra de controle, melhorando o tempo no quesito de indexação.

Desvantagens: As buscas podem se tornar um pouco lentas, por não se ter uma correlação direta entre Pessoa Física e Tipo, como anteriormente usado na tabela de controle. Por não ser necessário especialização total, é uma boa abordagem.

Soluções Alternativas: A modelagem poderia seguir os padrões anteriores, utilizando uma tabela de controle. No fim, a indexação talvez não fizesse valer a pena, haja vista que essa consulta além de não ser tão utilizada, não fica lenta o suficiente na modelagem usada para ser necessário sua mudança.

10. **Atributos Multivalorados (Vícios, Habilidades): Solução Adotada:** Seguiu-se a modelagem padrão para atributos multivalorados, onde cria-se uma tabela, herdando a chave principal de sua entidade-mãe que fará parte da chave primária da próxima tabela, juntamente com o seu diferenciador. Por exemplo, para a tabela de vícios, as chaves serão compostas pelo CPF do auxiliado, bem como pelo vício em si. De forma análoga para as habilidades, onde a tabela de habilidades possuirá o CPF do indivíduo e, para cada habilidade, será gerada uma nova tupla.

Vantagem: A modelagem facilita a correlação dos valores, sendo amplamente utilizada para essas situações.

Desvantagens: Pode-se ter valores nulos e o processo de indexação em si consome processamento.

Soluções Alternativas: Dependendo da importância dada à estes pontos, como trata-se de atributos multivalorados, caso eles sejam considerados descartáveis, poderiam ser apresentados como uma grande string univalorada. Futuramente, em tratamento de aplicação, através de um parser, esses valores poderiam ser retornados.

11. **Mapeamento do atributo derivado ‘idade’ da entidade ‘Auxiliado’**

Solução adotada: O atributo ‘idade’ da entidade ‘Auxiliado’ foi mapeado como sendo derivado por seu baixo custo de cálculo e para evitar colocar informações redundantes no modelo. Portanto, como foi considerado atributo derivado ele não foi colocado na tabela ‘auxiliado’ no modelo relacional.

Vantagens: Diminui a quantidade de dados armazenados nesta entidade.

Desvantagens: O cálculo que é necessário para ter a informação do atributo, desacelera possíveis tratamentos.

Soluções Alternativas: Colocar o atributo não sendo derivado. Isso implicaria que todos os dias um cálculo deverá ser feito para todos os membros daquela entidade, o que pode ser custoso de acordo com o tamanho da entidade.

6 Mudanças realizadas em relação a segunda entrega

Para continuar com a terceira e última parte do projeto foram adequados os mapeamentos realizados para as tabelas Terceiros, Frente, Auxiliado, Centro Comunitário e Funcionário.

As alterações realizadas em Centro Comunitário e Funcionário foram para viabilizar a inserção de tuplas em suas respectivas tabelas, anteriormente o atributo presidente era chave secundária não nula em Centro Comunitário e o atributo CentroComunitario em

Funcionário não poderia receber valores nulos. Isto condicionava uma inserção a outra, e.g, não era possível realizar inserções de tuplas em Centro Comunitário sem que antes tivesse sido inserida uma tupla na tabela Funcionário, ao passo que não era possível inserir uma tupla na tabela Funcionário em que antes fosse inserida uma tupla em Centro Comunitário, formando uma interdependência que não permitia nenhuma inserção sem valores *default*. Isso foi resolvido retirando a obrigatoriedade de existir um valor não nulo no atributo CentroComunitario em Funcionário.

Como em Terceiro e Frente nós temos generalizações disjuntas, i.e, os atributos que definem as entidades específicas destas tabelas são necessariamente mono valorados, não é eficiente criar uma tabela de controle para manter exclusivamente os tipos, isto acarreta na realização de um maior número de buscas e prejudica a escalabilidade da base de dados construída.

Destarte, foram incluídos nestas tabelas atributos Tipo, que definem a qual entidade específica a tupla da entidade geral se refere. Isso pode vir a acarretar problemas como excesso de nulos de forma geral, pois não se trata de uma especialização total, entretanto em nossa aplicação é improvável que isto ocorra.

Ademais, como estamos tratando de uma generalização em *overlap* para a tabela do Auxiliado, o atributo de definição de Tipo deve ser multi valorado e, portanto, deve se relacionar a chave primária através de uma tabela de controle, cujo a chave consiste na composição entre este atributo e a chave primária referente a Auxiliado.

7 Implementação

Nesta seção serão definidas as minúcias que permeiam nossa aplicação, a qual foi implementada utilizando a linguagem de programação Python na versão estável 3.10.8 para as interfaces gráficas e interativas e para administrar nosso banco de dados foi empregado o SGBD (Sistema de Gerenciamento de Banco de Dados) foi utilizado o PostgreSQL, a integração e composição entre ambos foi feita utilizando o Docker e o driver Psycopg.

A aplicação consiste de um ambiente interativo que permite a realização de consultas pré-especificadas que serão expostas em maiores detalhes, e a realização de consultas personalizadas de acordo com a necessidade do usuário, dentre queries pré-estabelecidas. A interface de usuário foi feita baseada na biblioteca CustomTkinter, externa a linguagem de programação Python.

7.1 Script de Consultas

Os dados utilizados para as consultas (esquema, dados e consultas) pode ser encontrado no repositório da implementação, na pasta "data", neste link.

- **C1-Consultar informações do auxiliado referentes a sua reabilitação**

```

1      SELECT A.NOME, C.NOME, R.PRESENCA
2      FROM REABILITACAO R
3      JOIN AUXILIADO A
4      ON R.DEPENDENTE = A.CPF
5      JOIN CLINICA C
6      ON R.CLINICA = C.CNPJ
7      WHERE (SELECT AVG(R1.PRESENCA) FROM REABILITACAO R1) < R.PRESENCA;
```

Nesta consulta foi realizado o *inner join* das tabelas REABILITACAO, AUXILIADO e CLINICA, a fim de extrair de cada uma delas o nome do AUXILIADO, a CLINICA que ele está em REABILITACAO e sua presença. A junção é realizada garantindo que só serão expostos auxiliados cujo a presença seja maior que a média geral de todos os auxiliados, isso é garantido pelas cláusulas ON presentes na consulta, que garantem a junção apenas de tuplas que atendam aos requisitos da cláusula WHERE.

- **C2-Consulta de empresas parceiras com vagas de emprego supervisionado**

```

1      SELECT T.NOME, EP.NUFUNCIONARIOS, ES.LOCAL
2      FROM TERCEIROS T
3      JOIN EMPREGO_SUPERVISIONADO ES
4      ON T.NUCPFCNPJ = ES.EMPRESA_PARCEIRA
5      JOIN EMPRESAS_PARCEIRAS EP
6      ON ES.EMPRESA_PARCEIRA = EP.CNPJ
7      WHERE EP.MAXFUNCIONARIOS > EP.NUFUNCIONARIOS;
```

Neste procedimento são selecionados nome, número de funcionários trabalhando atualmente e local da vaga oferecida referentes a uma empresa parceira, estas informações estão presentes, respectivamente, nas tabelas TERCEIROS, EMPRESAS_PARCEIRAS E EMPREGO_SUPERVISIONADO. Como as informações que buscamos estão em tabelas distintas é utilizada a junção interna, representada em SQL pela cláusula JOIN, a garantia de coerência e objetividade das informações buscadas são garantidas pelas cláusulas ON, a qual realiza a junção apenas das tuplas que atendem as demandas feitas em WHERE e propaga isso de forma encadeada. A clausula WHERE garante que as tuplas selecionadas em EMPRESAS_PARCEIRAS

tenham um número máximo de funcionários maior que o número que já ocupa as vagas, i.e, pode receber novos trabalhadores.

- **C3-Consulta do status empregatício de um auxiliado**

```

1      SELECT A.NOME,
2      CASE
3          WHEN ES.DURACAO - CURRENT_DATE < 10
4          AND ES.DURACAO - CURRENT_DATE > 0
5              THEN 'PROXIMO_AO_FIM'
6          WHEN ES.DURACAO - CURRENT_DATE > 10
7              THEN 'FUNCIONARIO_EMPREGADO'
8          WHEN (ES.DURACAO - CURRENT_DATE < 0)
9          OR ES.DURACAO IS NULL
10             THEN 'FUNCIONARIO_DESEMPREGADO'
11      END STATUS_EMPREGO
12  FROM AUXILIADO A
13  LEFT JOIN EMPREGO_SUPERVISIONADO ES
14  ON A.CPF = ES.AUXILIADO
15  LEFT JOIN TERCEIROS E
16  ON ES.EMPRESA_PARCEIRA = E.NUCPFCONPJ;
```

Nesta consulta são retornadas tuplas com o nome do auxiliado associado ao seu status empregatício, baseado em uma subtração entre a duração do emprego supervisionado e a data atual. Isso foi feito através da cláusula CASE que pode classificar através da diretiva WHEN. Na aplicação, se o auxiliado tem menos de dez dias de emprego supervisionado planejados, é considerado que seu trabalho está PROXIMO AO FIM, caso contrário ele tem o status de FUNCIONARIO EMPREGADO caso tenha mais de dez dias até o fim de seu contrato ou está desempregado caso o atributo DURACAO seja nulo ou apresente uma data menor que a data atual.

- **C4-Consulta de todos os auxiliados empregados**

```

1      SELECT * from AUXILIADO A
2      JOIN EMPREGO_SUPERVISIONADO ES
3      ON A.CPF = ES.AUXILIADO;
```

Nesta consulta nos interessa recuperar todas as tuplas de auxiliados que estão relacionadas a um emprego supervisionado, i.e, nos interessa saber quais auxiliados estão empregados e suas informações. O retorno apenas das tuplas da tabela AUXILIADO referentes a participantes com emprego ativo é garantido pela junção

interna, representada pela cláusula JOIN em conjunto com a cláusula ON, a qual delimita as tuplas de acordo com a condição apresentada.

- **C5-Consulta de centros e de quantos auxiliados se beneficiam de sua tesouraria**

```

1      SELECT C.NOME, COUNT(*) FROM
2      CENTRO C JOIN TESOURARIA T
3      ON C.CNPJ = T.CENTRO
4      JOIN AUXILIADO A
5      ON A.FRENTE = T.FRENTE AND T.CENTRO = A.CENTRO
6      GROUP BY C.NOME;
```

Nesta consulta são retornadas as tuplas contendo os nomes de todos os centros que possuem tesouraria e quantos auxiliados são subsidiados pelas suas respectivas tesourarias. Nesta consulta foi utilizada a cláusula GROUP BY para realizar a contagem da quantidade de tuplas referente a grupos separados pelo nome dos centros selecionados. Além disso, as cláusulas JOIN associadas as cláusulas ON garantem que são selecionados apenas centros com tesourarias na primeira afirmação lógica, destas são segregadas apenas as tesourarias que possuem auxiliados na segunda afirmação lógica, por fim os centros comunitários são agrupados pelos seus respectivos nomes.

- **C6-Consulta os auxiliados que são apenas moradores de rua**

```

1      SELECT A.NOME, A.LOCAL FROM
2      AUXILIADO A JOIN TIPO_AUXILIADO TA
3      ON A.CPF = TA.CPF
4      WHERE A.CPF NOT IN
5      (SELECT DISTINCT A.CPF FROM
6      AUXILIADO A JOIN TIPO_AUXILIADO TA
7      ON A.CPF = TA.CPF
8      WHERE UPPER(TA.TIPO) = 'DEPENDENTE_QUIMICO');
```

Nesta seleção são retornadas tuplas com nome e local dos auxiliados que são, exclusivamente, moradores de rua. A exclusividade do tipo de auxiliado vem da cláusula WHERE acompanhada de NOT IN, que não inclui os auxiliados cujo tipo é DEPENDENTE_QUIMICO, a cláusula DISTINCT apenas evita repetições de tuplas selecionadas pela cláusula WHERE e garante otimização.

7.2 Aplicação

O repositório da aplicação pode ser encontrado neste link.

7.2.1 Requisitos do Sistema

Além da linguagem de programação Python devidamente instalada em versão 3.8.2 ou maior é necessário instalar as dependências e o *package manager* Pip referente a esta versão.

Todas as considerações feitas a seguir consideram que o usuário está utilizando um sistema operacional do tipo Linux baseado em Debian.

Para facilitar o uso da aplicação foram criados *scripts* para instalação de algumas dependências, portanto é necessário estar no diretório raiz do projeto e permitir que os *scripts* sejam executados através do seguinte comando no terminal:

```
sudo chmod +x scripts/*.sh
```

Os pacotes Tkinter utilizado para construção da interface gráfica e Psycopg2 para integração com o PostgreSQL tem como dependências os pacotes python-tk e libpq-dev respectivamente, para baixa-los basta utilizar o seguinte comando em sua shell:

```
sudo apt install python-tk libpq-dev
```

Para finalizar os pacotes basta baixa-los utilizando o administrador de pacotes nativo do Python, o pip, através de um arquivo de texto criado por comodidade e facilidade de uso:

```
pip install -r requirements.txt
```

Por fim, para realizar a instalação e a inicialização do Docker, ambiente de integração entre Python e o SGBD utilizado, basta realizar o seguinte comando em sua própria shell:

```
sudo scripts/install_docker_ubuntu.sh && sudo scripts/install.sh
```

7.2.2 Funcionamento geral

A fim de executar o programa basta utilizar o seguinte comando:

```
python src/main.py
```

Com isso, será exibida a interface vista na Figura 1, nela são perceptíveis os botões responsivos ao clique a esquerda, os quais desempenham as seguintes funções:

- **Mostrar tabela:** É exibido um botão no estilo *dropdown* contendo como opções todas as tabelas implementadas, ao selecionar uma tabela específica, são exibidas suas colunas, o tipo de cada uma e seu respectivo tamanho. É possível ver um exemplo da interface na Figura 2.

- **Rodar/Debugar SQL:** Permite a inserção direta de queries SQL através de uma interface de escrita, esta função está implementada única e exclusivamente para facilitar testes e correções e não estaria disponível ao usuário final. Funciona basicamente como um SQL injection liberado e implementado pelo sistema. Sua execução está referenciada pela Figura 3, onde é possível ver uma janela inferior interativa, a qual recebe as queries, e uma janela superior, responsável por exibir os resultados.
- **Rodar consultas.sql:** Realiza as consultas previstas no arquivo pré-estabelecido pelos desenvolvedores. Sua execução está exposta na Figura 4, que exibe uma janela superior para o resultado das consultas.
- **Cadastrar novo funcionário, cadastrar empresa e inserir centro:** São funcionalidades de inserção de tuplas e possuem interfaces semelhantes, nessas são exibidos os atributos de cada tabela referenciada (Funcionário, Empresa Parceira e Centro Comunitário), os quais possuem um pequeno espaço para serem preenchidos os valores e um botão para confirmar a inserção. A interface referente a Cadastrar novo funcionário, que serve como base para as demais, está explicitada na Figura 5.
- **Mostrar informações:** É exibido um botão no estilo *dropdown* com todas as tabelas possíveis e disponíveis para consulta, ao selecionar uma tabela é exibido um segundo botão *dropdown* contendo todos os atributos da tabela, assim é possível conferir, por exemplo, todos os nomes dos auxiliados inseridos. A interface de uso pode ser vista na Figura 6, já com os dois botões *dropdown* exibidos.

Vale ressaltar que nas funcionalidades de inserção as strings inseridas nos campos delimitados são passadas como parâmetros de funções secundárias e não uma simples concatenação de strings, evitando ataques maliciosos.

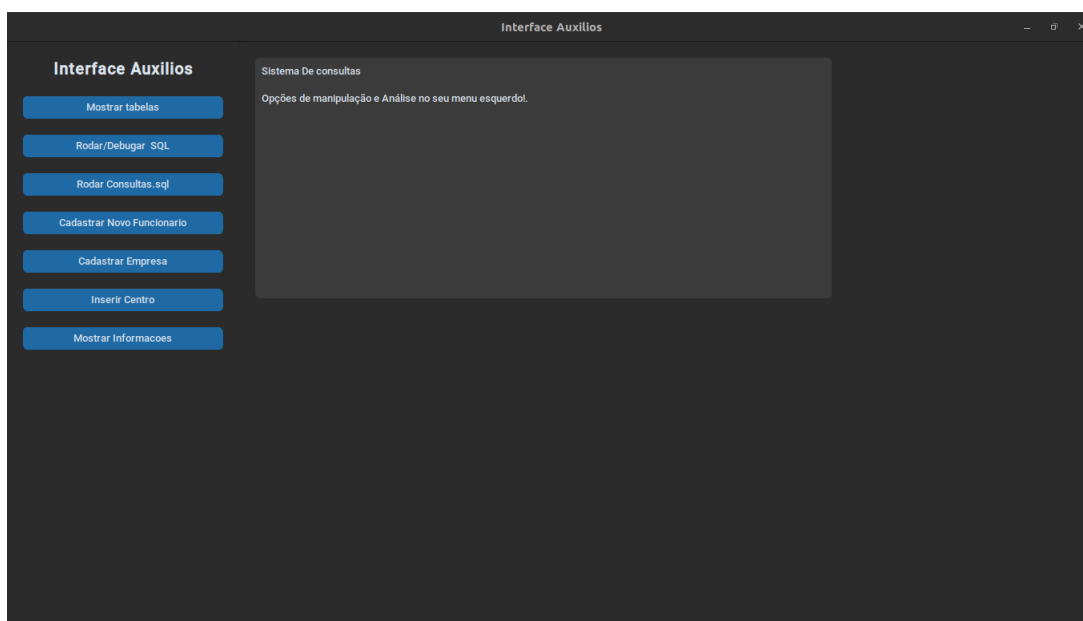


Figura 1 – Interface inicial. **Fonte:** Autores

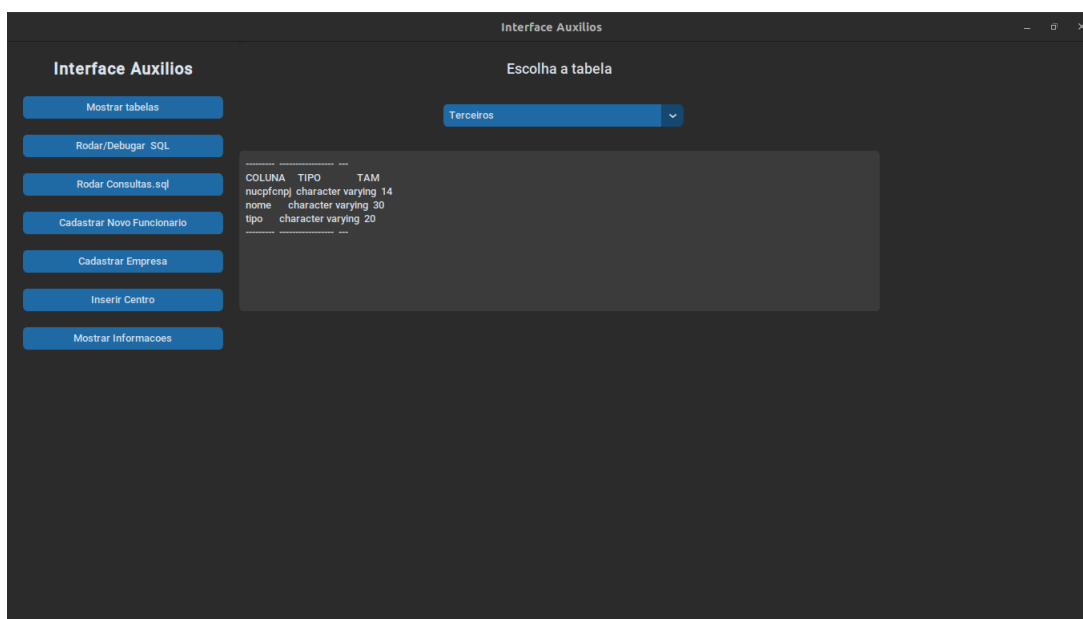


Figura 2 – Interface ao selecionar o botão Mostrar tabela. **Fonte:** Autores

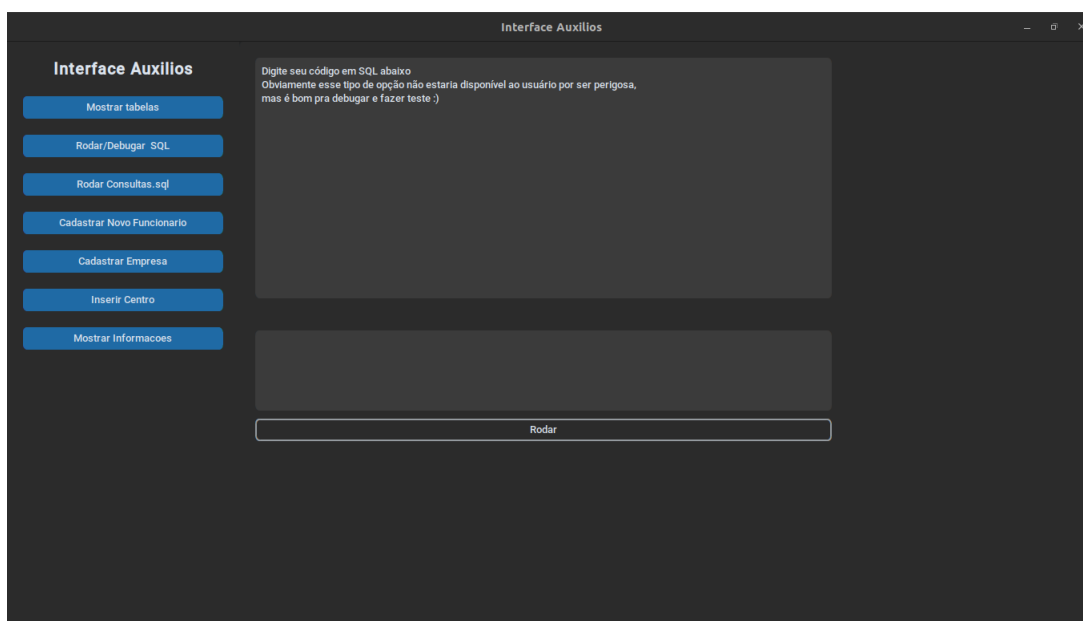


Figura 3 – Interface ao selecionar o botão Rodar/Debugar SQL. **Fonte:** Autores



Figura 4 – Interface ao selecionar o botão Rodar consultas.sql. **Fonte:** Autores

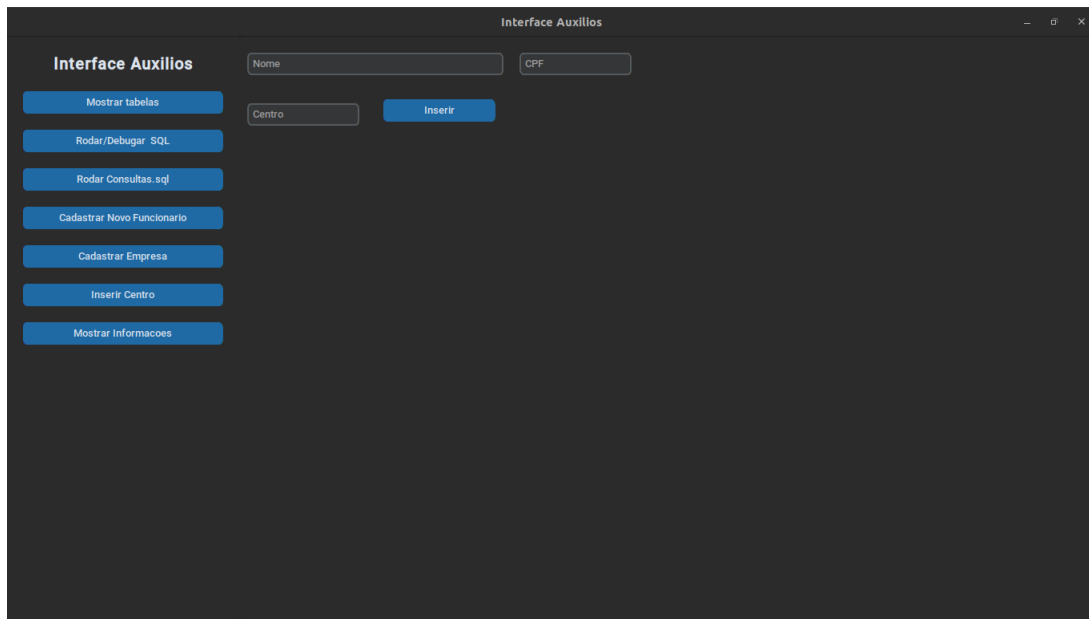


Figura 5 – Interface ao selecionar o botão Cadastrar novo funcionário. **Fonte:** Autores

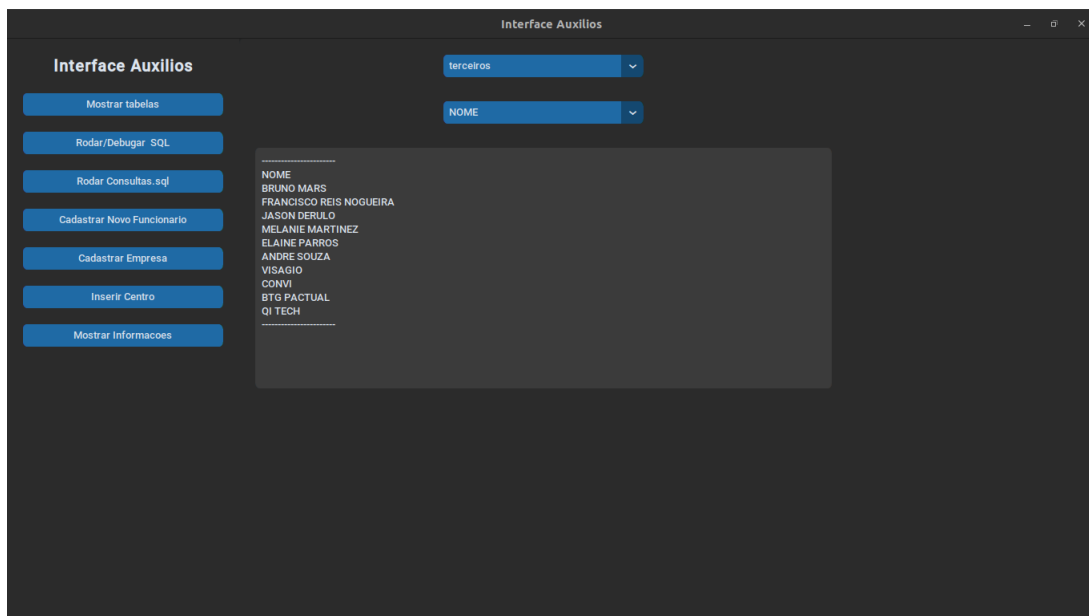


Figura 6 – Interface ao selecionar o botão Mostrar informações. **Fonte:** Autores

8 Conclusão e considerações finais

Após um semestre inteiro aprendendo os fundamentos de base de dados podemos concluir, em grupo, alguns aspectos relevantes da disciplina e do trabalho no geral. O tema de ações comunitárias fez o grupo a pensar em situações nas quais a importância dos conhecimentos aprendidos na matéria se destacaria com maior relevância.

Assim, escolhendo o tema de Auxílios a Dependentes Químicos e Moradores de rua nos fez pensar não somente na disciplina, mas sim como esse projeto poderia servir como base para uma implementação de um Centro Comunitário a fim de auxiliar aqueles que poderiam necessitar de ajuda.

O projeto começou com um dos maiores desafios que foi a criação da ideia e o desenvolvimento do Modelo Entidade Relacionamento (MER). O qual foi o mais complicado para o grupo, mas com os auxílios da Docente e Monitor PAE se mostrou como um dos mais importantes na definição de uma Base de dados.

No demais, o projeto se mostrou como uma grande forma de fazer o grupo a aprender realizando uma integração com toda a disciplina e também como deve ser implementada uma base dados corretamente, tanto para a academia quanto para o mercado. Dividir o projeto em três partes e cada parte ser sincronizada com o assunto da disciplina se mostrou muito eficaz no aprendizado.

Por fim, o projeto da disciplina é muito rico em informações e conteúdo para fazer o aluno da disciplina aprender muito bem os assuntos de Base de dados.