



Moisés Waidemann Molinillo Júnior, RM: 563719

Gabriel Sbrana Campos, RM: 565849

Thiago Rodrigues da Mota, RM: 563650

Fiapinhos

Solução

Dores identificadas

- Baixa adesão à telemedicina (consultas online)
- Dificuldade com tecnologia(apps/senhas)
- Alta taxa de abstenção (faltas):

Com base nessas dores vamos criar uma interface simples focada no público idoso, diminuindo a abstenção desse grupo em tele-consultas também.

Vamos criar uma plataforma que combina:

- Página Web Acessível Simples para pacientes visualizarem informações das consultas.
- Chatbot via WEB para enviar lembretes automáticos, permitir confirmações e fornecer ajuda com linguagem simples.
- Automação em Python que gerencia envio e registro das confirmações.
- Java Criação de uma API Restful.
- Banco de dados para centralizar pacientes, agendamentos, status e mais.

Java

Propósito Principal

Criação de uma API Restful para consumo do Front-end.

Principais funcionalidades do código

Classe: ConnectionFactory

Atributos:

- connection: Connection - É um objeto de conexão com o banco de dados

Métodos:

- getConnection(): Connection – retorna um conexão com um banco de dados
- closeConnection(): void – Fecha a conexão com o banco de dados

Interface: IDAO.

Métodos:

- findAll(): ArrayList - Ler todos os registros de uma tabela
- save(Object) : Object - Registrar uma linha em uma tabela
- findById(Long): Obejct - Ler um registro de uma tabela
- delete(Long): boolean - Deletar um registro de uma tabela
- update(Object): Obejct - Atualizar um registro de uma tabela

Todas as classes DAO implementam os métodos dessa interface.

Interface: IBO.

Métodos:

- `findAll(): ArrayList` - Esse método utiliza o método `findAll` de uma classe DAO para ler todos os registros de uma tabela.
- `save(Object) : Object` - Esse método utiliza o método `save` de uma classe DAO para registrar uma linha em uma tabela
- `findById(Object): Object` - Esse método utiliza o método `findById` de uma classe DAO para ler um registro de uma tabela
- `delete(Long): boolean` - Esse método utiliza o método `delete` de uma classe DAO para deletar um registro de uma tabela
- `update(Object): Object` - Esse método utiliza o método `update` de uma classe DAO para atualizar um registro de uma tabela

Todas as classes BO implementam os métodos dessa interface. Cada classe BO possui um objeto das classes DAO.

Interface: IRESOURCE

Métodos:

- findAll(): ArrayList - Este método utiliza o método findAll de uma classe BO, definindo status de resposta (Response).
- save(Object) : Object - Esse método utiliza o método save de uma classe BO, definindo status de resposta (Response)
- findById(Long) Object - Esse método utiliza o método findById de uma classe BO, definindo status de resposta (Response)
- delete(Long): boolean - Esse método utiliza o método delete de uma classe BO, definindo status de resposta (Response)
- update(Object, Long) Object - Esse método utiliza o método update de uma classe BO, definindo status de resposta (Response)

Todas as classes RESOURCE implementam os métodos dessa interface. Cada classe RESOURCE possui um objeto das classes BO.

Explicando as classes TO:

As classes TO possuem identificadores e atributos que são colunas das tabelas do banco de dados. Especificamente, as classe PacienteTO e MedicoTO possuem herança da classe pai PessoaTO.

Tabela de EndPoints (API Restful)

<u>URIs</u>	<u>Verbos</u>	<u>Status bom</u>	<u>Status ruim</u>	<u>Nome dos métodos</u>
http://localhost:8080/unidadeHC	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findAll</u>
http://localhost:8080/unidadeHC/{id}	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findById</u>
http://localhost:8080/unidadeHC/{id}	<u>PUT</u>	<u>201</u>	<u>400</u>	<u>update</u>
http://localhost:8080/unidadeHC	<u>POST</u>	<u>201</u>	<u>400</u>	<u>create</u>
http://localhost:8080/unidadeHC/{id}	<u>DELETE</u>	<u>204</u>	<u>404</u>	<u>delete</u>
http://localhost:8080/telefonePaciente	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findAll</u>
http://localhost:8080/telefonePaciente/{id}	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findById</u>
http://localhost:8080/telefonePaciente/{id}	<u>PUT</u>	<u>201</u>	<u>400</u>	<u>update</u>
http://localhost:8080/telefonePaciente	<u>POST</u>	<u>201</u>	<u>400</u>	<u>create</u>
http://localhost:8080/telefonePaciente/{id}	<u>DELETE</u>	<u>204</u>	<u>404</u>	<u>delete</u>
http://localhost:8080/paciente	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findAll</u>
http://localhost:8080/paciente/{id}	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findById</u>
http://localhost:8080/paciente/{id}	<u>PUT</u>	<u>201</u>	<u>400</u>	<u>update</u>
http://localhost:8080/paciente	<u>POST</u>	<u>201</u>	<u>400</u>	<u>create</u>
http://localhost:8080/paciente/{id}	<u>DELETE</u>	<u>204</u>	<u>404</u>	<u>delete</u>
http://localhost:8080/medico	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findAll</u>
http://localhost:8080/medico/{id}	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findById</u>
http://localhost:8080/medico/{id}	<u>PUT</u>	<u>201</u>	<u>400</u>	<u>update</u>
http://localhost:8080/medico	<u>POST</u>	<u>201</u>	<u>400</u>	<u>create</u>
http://localhost:8080/medico/{id}	<u>DELETE</u>	<u>204</u>	<u>404</u>	<u>delete</u>

Tabela de EndPoints (API Restful)

<u>URIs</u>	<u>Verbos</u>	<u>Status bom</u>	<u>Status ruim</u>	<u>Nome dos métodos</u>
http://localhost:8080/logradouro	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findAll</u>
http://localhost:8080/logradouro/{id}	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findById</u>
http://localhost:8080/logradouro/{id}	<u>PUT</u>	<u>201</u>	<u>400</u>	<u>update</u>
http://localhost:8080/logradouro	<u>POST</u>	<u>201</u>	<u>400</u>	<u>create</u>
http://localhost:8080/logradouro/{id}	<u>DELETE</u>	<u>204</u>	<u>404</u>	<u>delete</u>
http://localhost:8080/consulta	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findAll</u>
http://localhost:8080/consulta/{id}	<u>GET</u>	<u>200</u>	<u>404</u>	<u>findById</u>
http://localhost:8080/consulta/{id}	<u>PUT</u>	<u>201</u>	<u>400</u>	<u>update</u>
http://localhost:8080/consulta	<u>POST</u>	<u>201</u>	<u>400</u>	<u>create</u>
http://localhost:8080/consulta/{id}	<u>DELETE</u>	<u>204</u>	<u>404</u>	<u>delete</u>

Protótipo do sistema

Modelo de Entidade-Relacionamento (MER)

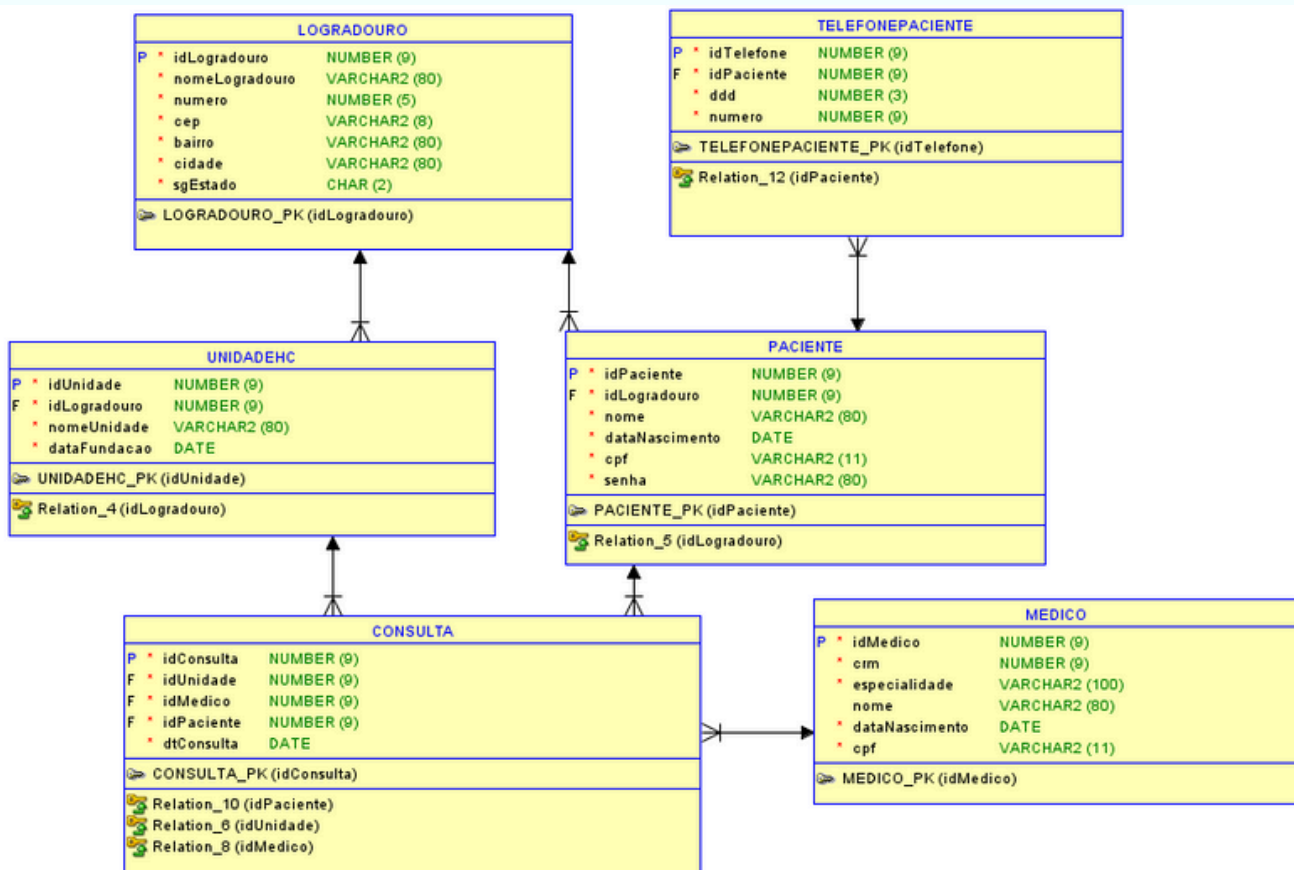
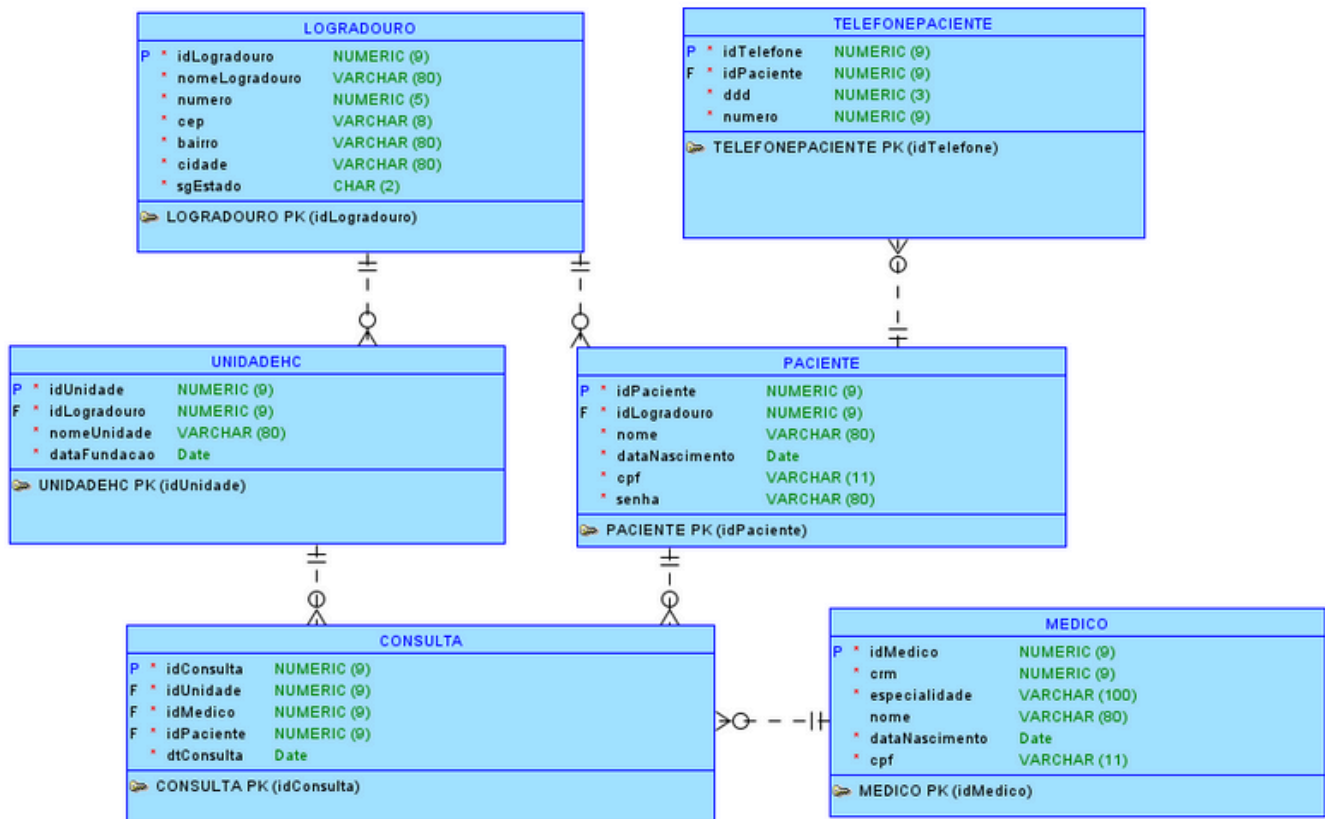


Diagrama de classe

