Documento de Projeto de Software

Projeto: Sistema de Filas de Prioridade Hospitalar. **Entrega:** 2. **Responsáveis**: Vitor dos Santos Correia, Luis Felipe Ferreira Martins, Cristian Brandão Tavares.

1. Introdução

Este documento apresenta o Projeto Integrador, uma iniciativa desenvolvida para consolidar os conhecimentos e habilidades adquiridos nas disciplinas de Estrutura de Dados, Programação Orientada a Objetos e Banco de Dados. O cerne deste projeto reside na implementação de um sistema de filas de prioridades, utilizando a eficiente estrutura de dados min-heap para gerenciar a ordem dos elementos. A base tecnológica do sistema é construída com TypeScript, garantindo robustez e escalabilidade, e integra-se a um banco de dados MySQL para persistência e gestão dos dados. O projeto serve como uma plataforma prática para aplicar conceitos teóricos, promovendo a integração entre essas áreas cruciais da computação e demonstrando a interdependência de seus princípios no desenvolvimento de soluções robustas e performáticas.

2.Justificativa e objetivos:

Atualmente, hospitais públicos em diversas regiões ainda dependem de métodos arcaicos e ineficientes para a gestão de filas de prioridade. A utilização de registros em papel e anotações manuais é uma prática comum, que, embora tradicional, acarreta uma série de desvantagens significativas, culminando em uma diminuição drástica da eficiência operacional.

Essa abordagem manual não apenas consome um tempo valioso da equipe administrativa e de enfermagem, que se vê constantemente engajada em tarefas repetitivas de cadastro, mas também gera um gasto desnecessário de recursos. A necessidade de preencher novos formulários a cada atendimento ou visita subsequente do mesmo paciente resulta em um consumo excessivo de papel.

Mais grave ainda, a duplicidade de esforços é inerente a esse sistema: a cada nova ida ao hospital, o paciente precisa ser recadastrado, e a pessoa responsável pelo primeiro registro é forçada a repetir a mesma tarefa, independentemente da frequência com que o paciente busca atendimento. Isso não só é ineficiente em termos de mão de obra, mas também aumenta a margem de erro e a desorganização dos prontuários.
Com a finalidade de mitigar esses problemas e otimizar o fluxo de atendimento, nossa equipe propôs o desenvolvimento de um sistema inovador para o cadastro e manuseio de pacientes. A premissa central dessa solução é a singularidade do registro: o paciente é cadastrado no sistema apenas uma única vez. Em visitas subsequentes, o processo se simplifica drasticamente. O paciente passa por uma triagem rápida e eficiente e, em seguida, é diretamente encaminhado para a fila de espera digital. O grande diferencial reside na gestão de prioridades, onde os atendimentos são organizados de forma inteligente, permitindo que os médicos chamem os pacientes de acordo com sua prioridade definida pelo sistema.
Essa iniciativa visa não apenas eliminar o desperdício de papel e a redundância de trabalho, mas também aprimorar significativamente a experiência do paciente, reduzindo o tempo de espera e garantindo que os casos mais urgentes sejam atendidos prontamente. Ao centralizar as informações e automatizar a gestão da fila, o projeto busca trazer mais eficiência, organização e humanização para o ambiente hospitalar.

3.Dados do Governo Utilizados:

- *.O Protocolo de Manchester*

4.Tecnologias Utilizadas:

- *Linguagens de programação: Typescript*
- *Gerência de Dados:* MySQL.

5. Plataforma de Implementação

O sistema em questão trata-se de um Sistema de Filas, e apresenta as seguintes características:

- Envolve grande quantidade de dados e a sua gerência deve ser feita usando um banco de dados.
- Composto por um banco de dados (MySQL).
- Há uma grande quantidade de interfaces (telas);

Levando-se em consideração essas características, decidiu-se implementar o sistema para o Projeto Integrador usando a linguagem de programação Typescript, o banco de dados relacional MySQL.

6.Arquitetura de Software

Assim, levando-se em consideração os requisitos para o sistema proposto, foram considerados como os principais atributos de qualidade a serem incorporados ao sistema os seguintes, apresentados juntamente com as táticas a serem aplicadas:

Usabilidade:

O sistema foi feito com estrutura e design extremamente claros para o usuário, reafirmando a praticidade de uso do mesmo.

Fácil Manutenção:

A organização do sistema deve se dar de modo que as responsabilidades das classe sejam bem divididas para que em mudanças futuras sejam de fácil manutenção. O uso de boa práticas como encapsulamentos, polimorfismos, classes abstratas e interfaces torna o sistema extremamente amigável com o desenvolvedor.

Segurança: O sistema possui segurança em seus dados tais como filtros que bloqueiam SQL injection, e outros ataques que podem corromper o sistema. Também é utilizado tokens e senhas de acesso aos dados privados.

Desempenho:

O desempenho do Projeto é alto, porém o seu desempenho preciso irá depender da quantidade de usuários e do volume de dados provindos de notas e avaliações enviados pelos usuários.

7.O Sistema do Projeto Integrador

Conforme discutido anteriormente, o sistema está organizado em três camadas: *Camada de Interface com o Usuário*, *Lógica de Negócio* e *Padrão de Projeto e Camada de Gerência de Dados.*

7.1.Camada de Interface com o Usuário

Aqui é definido o design de telas e como será a experiência do usuário no Projeto Integrador, os layouts foram desenhados com base em protótipos de fácil entendimento e simples de se usar.

O aplicativo foi definido nas seguintes telas:

- Tela da Recepção.
- Tela da Triagem.
- Tela da Chamada da fila.
- Tela do Atendimento.

7.2.Camada de Lógica de Negócio e Padrão de Projeto

O padrão adotado para a construção do sistema foi o modelo-visão-controlador, uma arquitetura de software que torna o desenvolvimento ágil e bem modularizado.

O **modelo** consiste nos dados da aplicação, regras de negócios, lógica e funções.

A **visão** pode ser qualquer saída de representação dos dados, como uma tela ou animação no celular.

O **controlador** faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão.

Baseado na necessidade de se criar um software que atenda os requisitos supracitados neste documento, o **MVC** apresentou-se como a forma ideal de arquitetura.

7.3.Camada de Gerência de Dados

A persistência dos objetos deste sistema é realizada no banco de dados relacional mysql, sendo desejável isolar os impactos da tecnologia de bancos de dados sobre o sistema. Assim, optou-se por adotar o padrão DAO e foi utilizado o utilitário de Persistência.

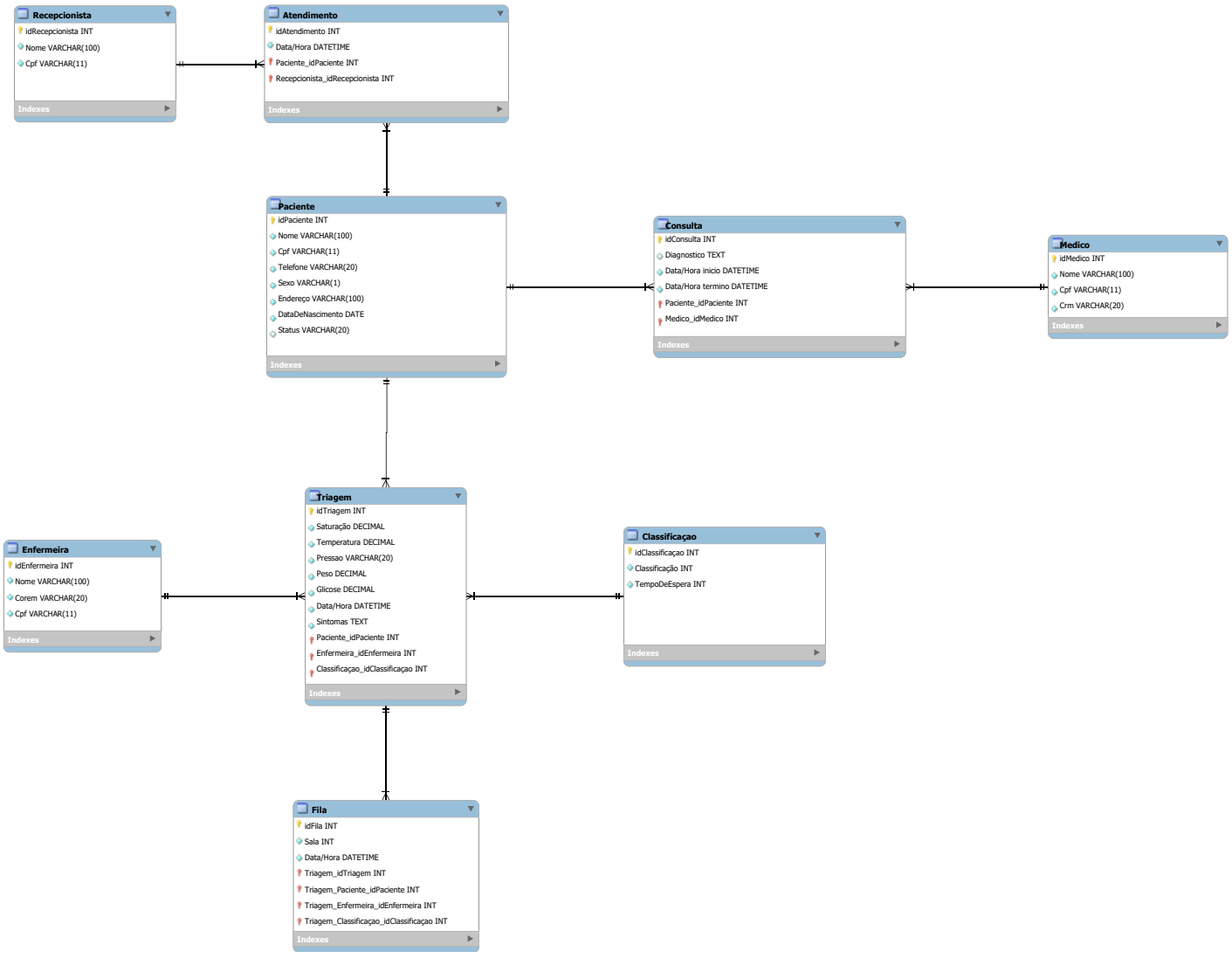
Uma API será utilizada para carregar dados do governo e um banco para armazenar dados e avaliações do usuário. Classes a serem persistidas devem herdar do modelo, que provê identificadores únicos para os objetos (Ids) a serem usados para mapear objetos em memória com as correspondentes linhas das tabelas Para cada classe de domínio a ser persistida, deve ser criada uma classe DAO correspondente. Tal informações são convertidas para JSON.

8. Conclusão

A solução proposta endereça diretamente as ineficiências causadas pelos métodos manuais de registro, como a redundância de cadastros. Com a centralização das informações e a automação da gestão de prioridades, o sistema não apenas promove uma redução significativa no tempo de espera e um aumento na eficiência operacional, mas também contribui para uma experiência mais humanizada e ágil para os pacientes. Este projeto, portanto, exemplifica como a tecnologia pode ser uma ferramenta transformadora para aprimorar a qualidade e a capacidade de resposta dos serviços de saúde

9. Anexos

Modelo Relacional



Benchmark

Critérios	Código grupo	Código aberto
Linguagem	TypeScript	C
Estrutura	Min-Heap	Lista Encadeada Simples
Lógica de priorização	Baseada nos níveis de prioridade SUS. Operações de inserção e extração do prioritário são O(log n).	A lógica para manter a ordem ou encontrar o próximo prioritário pode ser O(n).
Complexidade	→ Adicionar Paciente (com prioridade): O(log n) - Muito eficiente. A nova prioridade sempre sobe para sua posição correta no heap. → Chamar Próximo Paciente (extrair o de maior prioridade): O(log n) - Muito eficiente. O elemento raiz é removido e o heap é reorganizado. → Ver Próximo Paciente (sem remover): O(1) - Simplesmente olha o elemento raiz.	→ Adicionar Paciente: O(n) no pior caso, pois pode ser necessário percorrer a lista para encontrar a posição correta de inserção para manter a ordem. → Chamar Próximo Paciente do início : O(1). → A complexidade da inserção torna-se um problema.

fonte: https://github.com/Mahelchandupa/Hospital-Queue-Management-System

Links Uteis

Repositório do GitHub

Apresentação