Universidade Federal de Campina Grande Centro de Engenharia Elétrica e Informática Curso de Ciência da Computação

Alunos: André Abrantes Deuzimar Sepúlveda Lucas Ribeiro Vladwoguer Bezerra

Relatório do projeto de Sistemas de Informação 1

## 1-View:

A tecnologia utilizada foi o JSF(Java Server Faces), que atualmente oferece uma diversidade muito grande de recursos e componentes..

## 2-Controller:

## O controler do sistema se dá através de algumas classes:

MainBean: Representa as ações da página principal.

Pages: Enum onde estão as urls das páginas o que facilita a navegação.

LoginBean: Representa a operação de login.

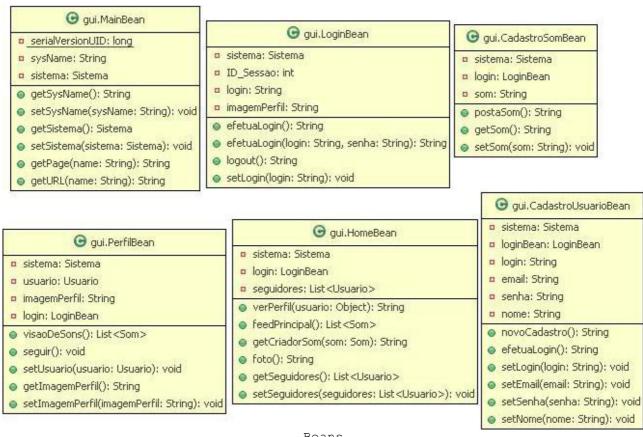
PesquisaBean: Controla a operação de pesquisar um usuário.

PerfilBean: Representa as operações de Perfil.

HomeBean: Representa as operações na página do usuário como Mostrar Feed

Principal e os seguidores.

CadastroSomBean: Controla as operações de cadastro de um novo som. CadastroUsuarioBean: Controla as operações de cadastro de Usuário.



Beans

## 3-Model:

Som: Classe que modela um os atributos e operações de uma nova música dentre os atributos principais estão o link e um OurTime(Classe que modela as Datas do sistema).

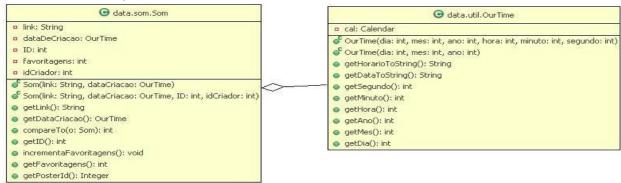


Diagrama Som e OurTime

Sistema: é a principal classe do projeto, ela controla as principais operações do blog e a iteração dos usuários, solicitação de amizade, além das listas de música e músicas favoritas. @ data.user.Usuario ⊕ data,system,Sistema nome: String usuarios: Map<Integer, Usuario> o conta: Conta sessoesAbertas: Map<Integer,Integer> memail: String sonsCadastrados: Map < Integer, Som > sons: List < Som > compositionRule3: Comparator<Integer> seguidores: List < Usuario > criarUsuario(login: String, senha: String, nome: String, email: String): void feedSecundario: List<Integer> abrirSessao(login: String, senha: String): int feedPrincipal: List < Integer > postarSom(idSessao: int, link: String, dataCriacao: String): int favoritagensDoUsuario: Map<Integer,Integer> seguirUsuario(idSessao: int, login: String): void rule: CompositionRule favoritarSom(idSessao: int, idSom: int): void sonsFavoritos: List<Integer> visaoDosSons: List<Integer> G data.util.SolicitacaoDeAmizade fontesDeSons: List<Integer> □ ID: int autenticaConta(senha: Object); boolean remetente: Usuario postaSom(som: Som): void destinatario: Usuario favoritaSom(idSom; int): void 🂣 SolicitacaoDeAmizade(iD: int, remetente: Usuario, destinatario: Usuario) getSonsFavoritos(): List<Integer> getFeedPrincipal(); List < Integer > getID(): int getRemetente(): Usuario addSonsNoFeedSecundario(sonsFavoritos: List<Integer>): void getFavoritagensDoUsuario(userId: Integer): Integer getDestinatario(): Usuario o equals(obj: Object): boolean setRule(rule: CompositionRule): void getFeedExtra(): List <Integer> @ data.som.Som link: String dataDeCriacao: OurTime □ ID: int a favoritagens; int □ idCriador: int Som(link: String, dataCriacao: OurTime) of Som(link: String, dataCriacao: OurTime, ID: int, idCriador: int) getLink(): String getDataCriacao(): OurTime compareTo(o: Som): int

Conta: Interface define as operações básicas de uma Conta. Usuário: define os atributos básicos e operações de um usuário.

Visão geral do sistema

aetID(); int

incrementaFavoritagens(): void
getFavoritagens(): int
getPosterId(): Integer