

DB

Let's Tech Up Together





Alain Delon

Quality Assurance

Daniel Jacon

Front End



Eduardo Sedrez

Back End

Eduardo Queiroz

Quality Assurance



Lucas Miguel

Back End

Ricardo Langbecker

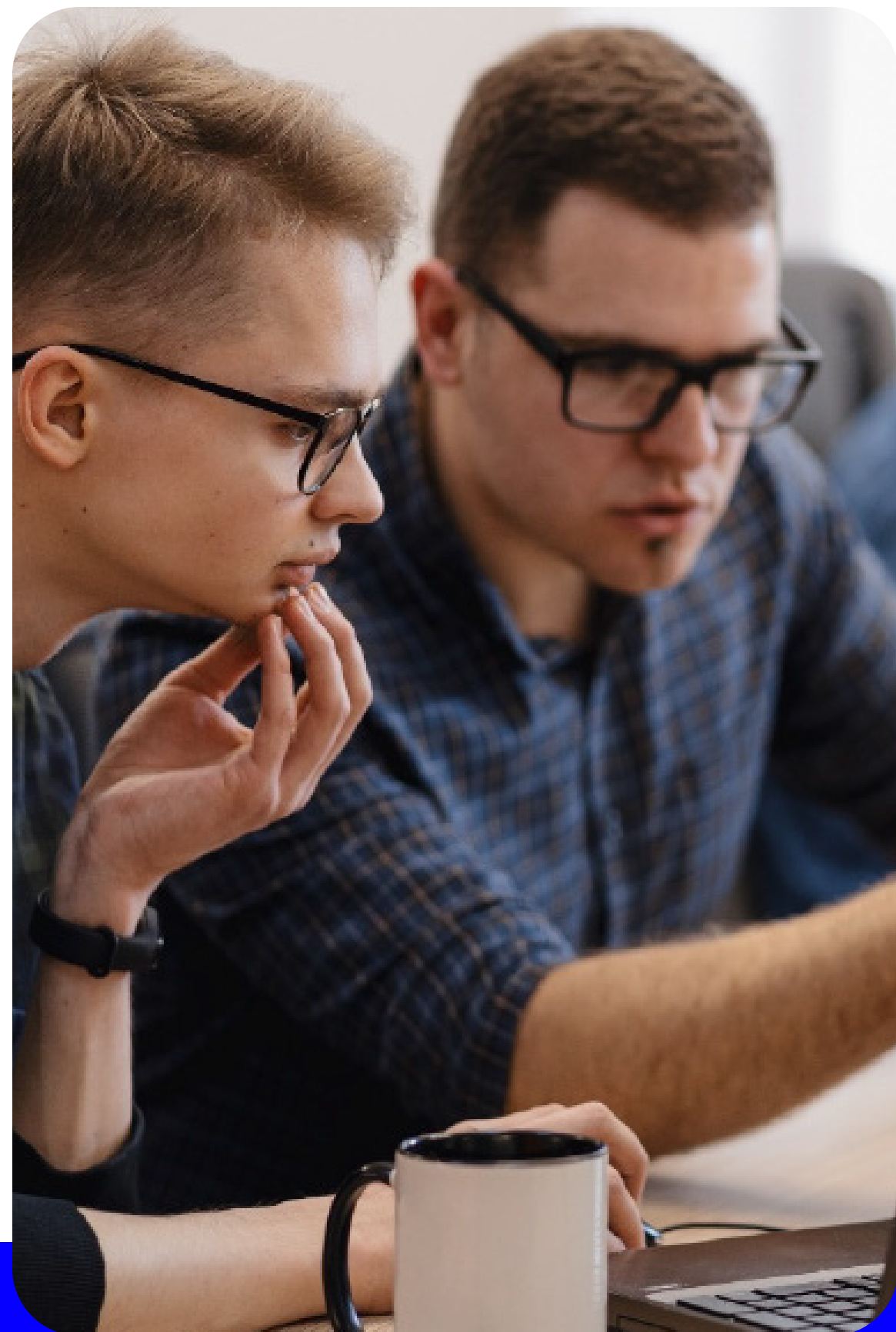
Back End



Cristina Jung

Product Owner

COLABORADORES

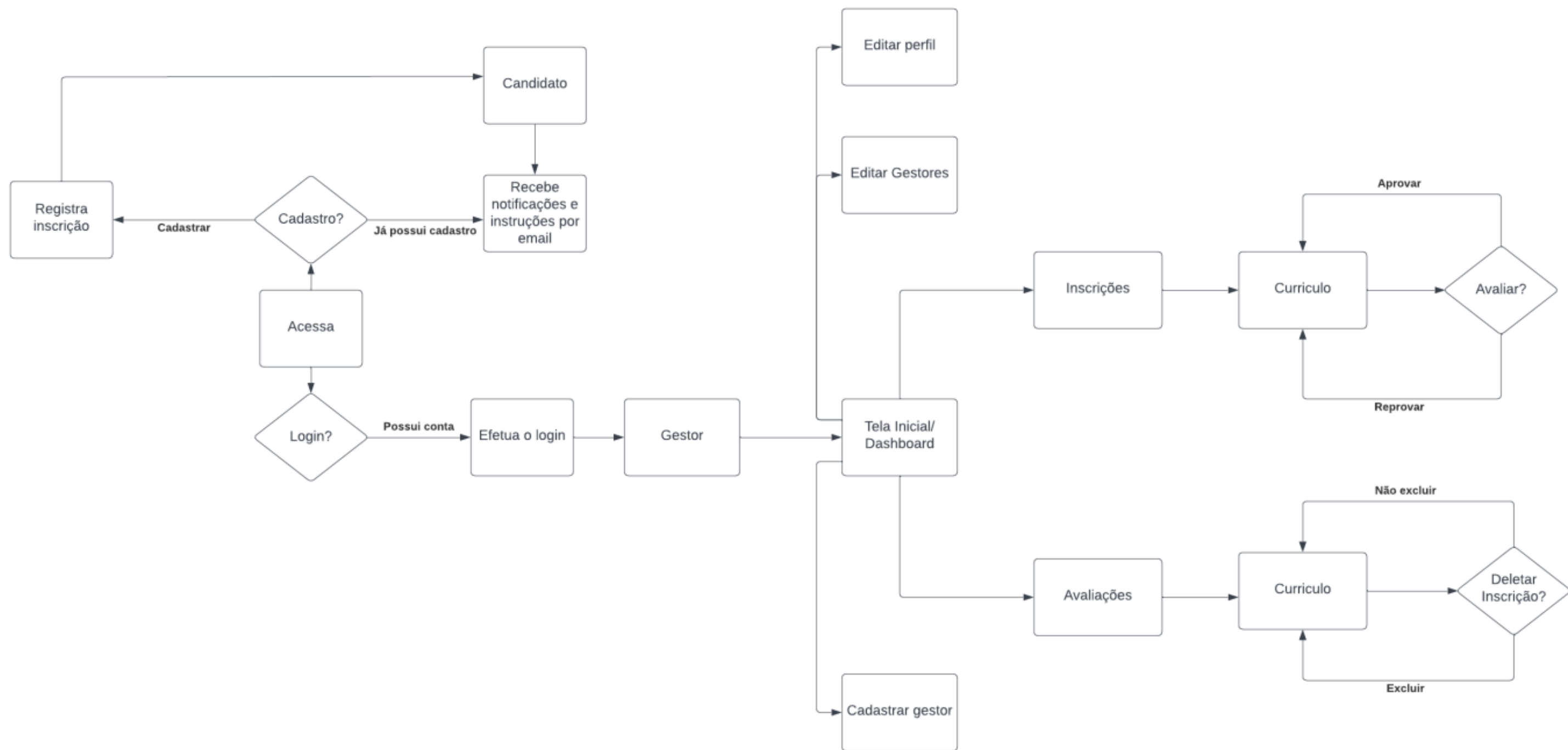


Projeto VemVemser

A empresa DBC Company possui um programa de aceleração de conhecimento técnico chamado VemSer. O cliente necessita de um sistema web que proporcione maior agilidade no gerenciamento das inscrições e avaliações de seus candidatos ao projeto VemSer, de forma online e automatizada.

A solução é desenvolver um site onde seja possível realizar inscrições, fazer avaliações e envio de emails resposta. O sistema permitirá o gerenciamento do gestor, seja ele um perfil de administrador ou colaborador, o qual poderá cadastrar, alterar, pesquisar, excluir perfis, listar candidatos e realizar avaliações.

Fluxo da aplicação



Projeto VemVemser

Confira a aplicação

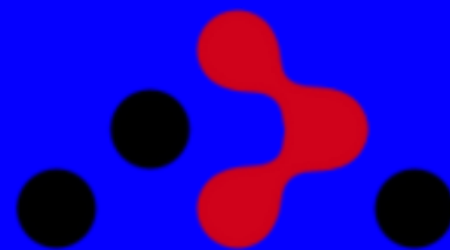


Tecnologias Utilizadas

ORACLE

TS TypeScript

▲ Vercel



REACT ROUTER

AJV



A X 1 O S

mockito

 Swagger™

 React

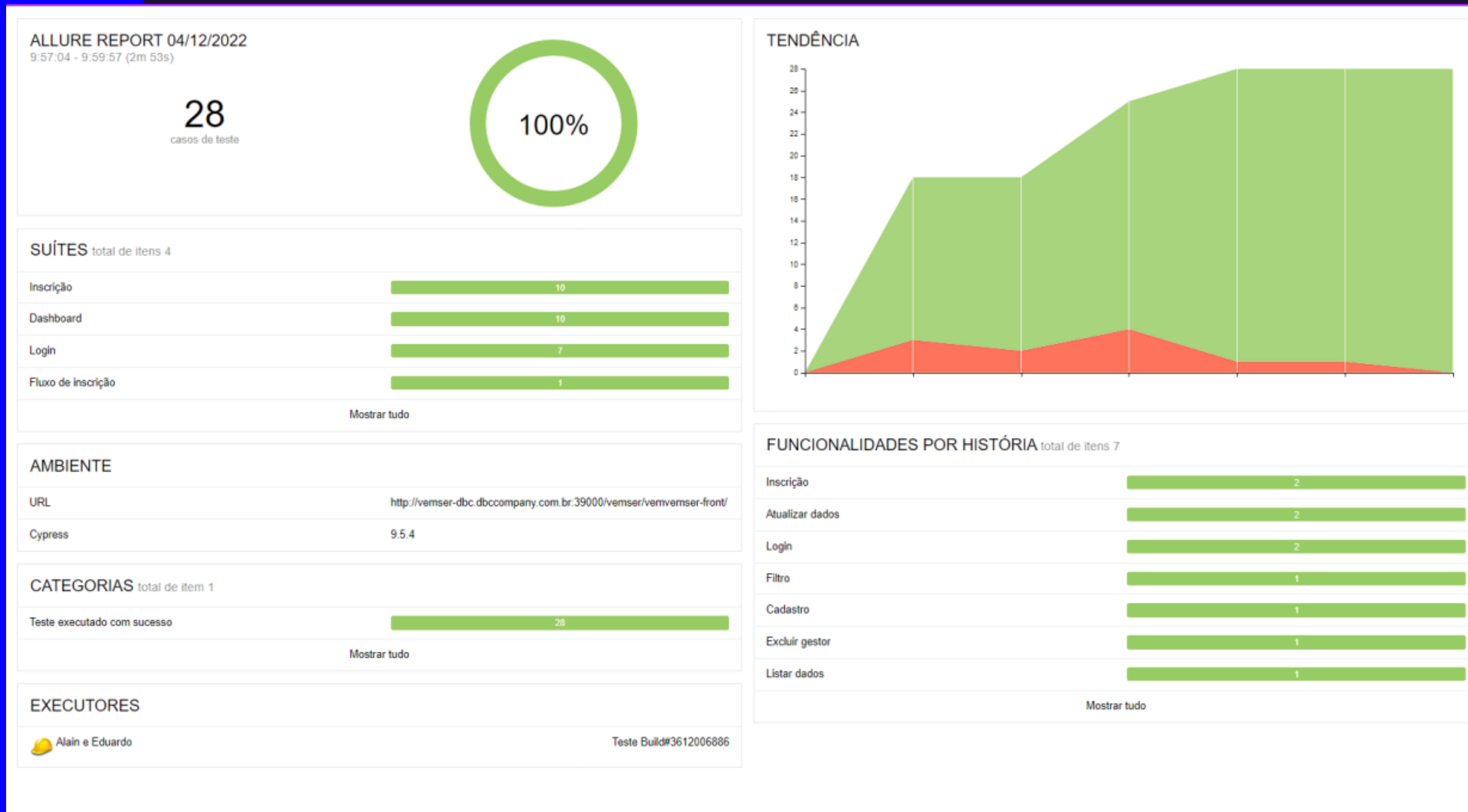


 spring

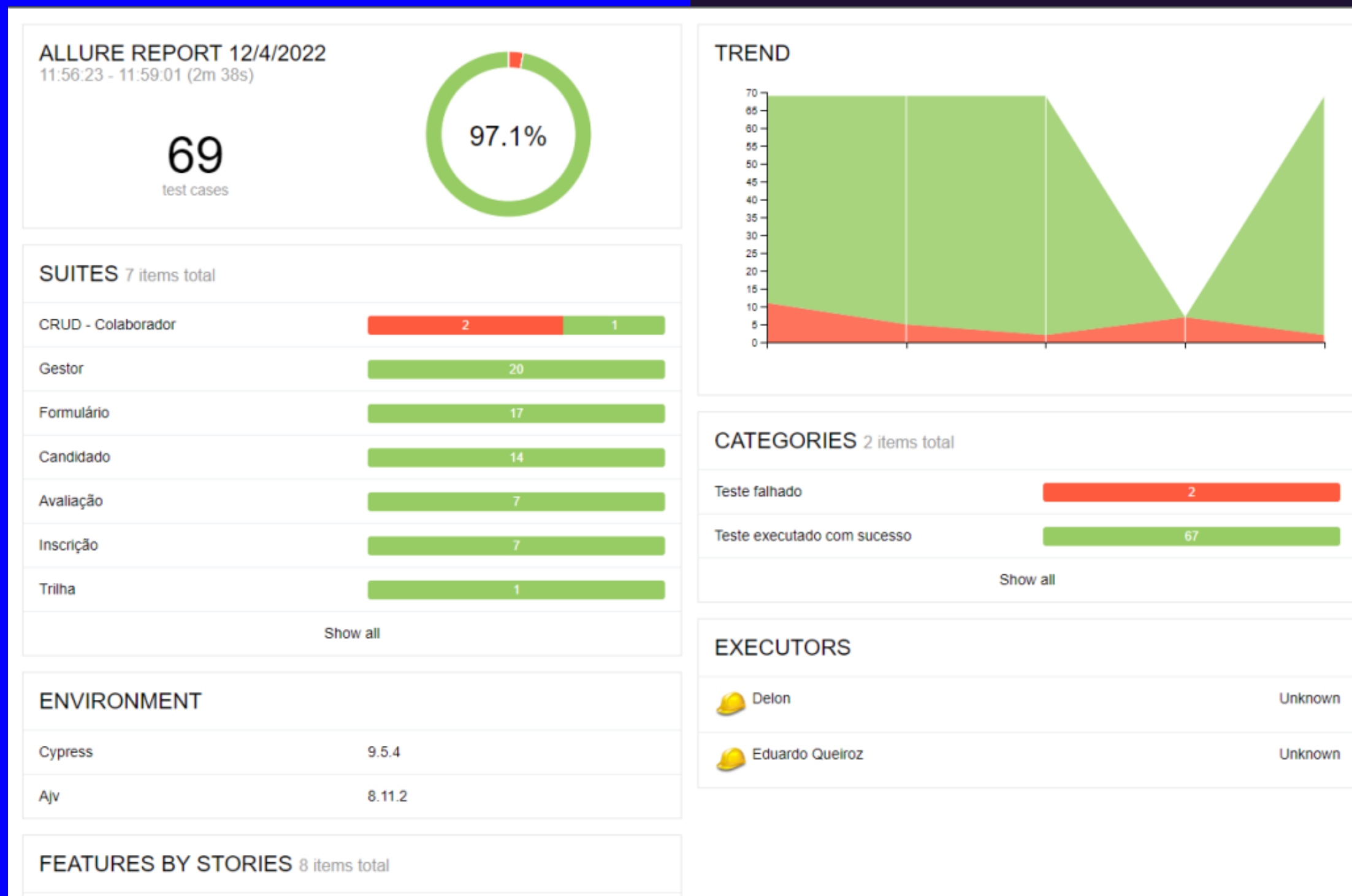
 Discord

 cypress

Testes - Front End



Testes - Back End



Models

SQL SCRIPT

Backend

QA

Frontend

+ Adicionar um cartão

To-Do

+ Adicionar um cartão

Doing

Plano de Testes

Avaliação [Teste Manual]

+ Adicionar um cartão

Liberado pra teste no front

+ Adicionar um cartão

Waiting Backend

+ Adicionar um cartão

Done

Perfil: alterar dados do usuário logado

Avaliações: estilização e integração com o BackEnd

Candidatos: estilização e validação

Inscrições/curriculo: estilização e integração com o BackEnd

Inscrições/home: estilização e integração com o Back End

Candidatos: integração com o Back End

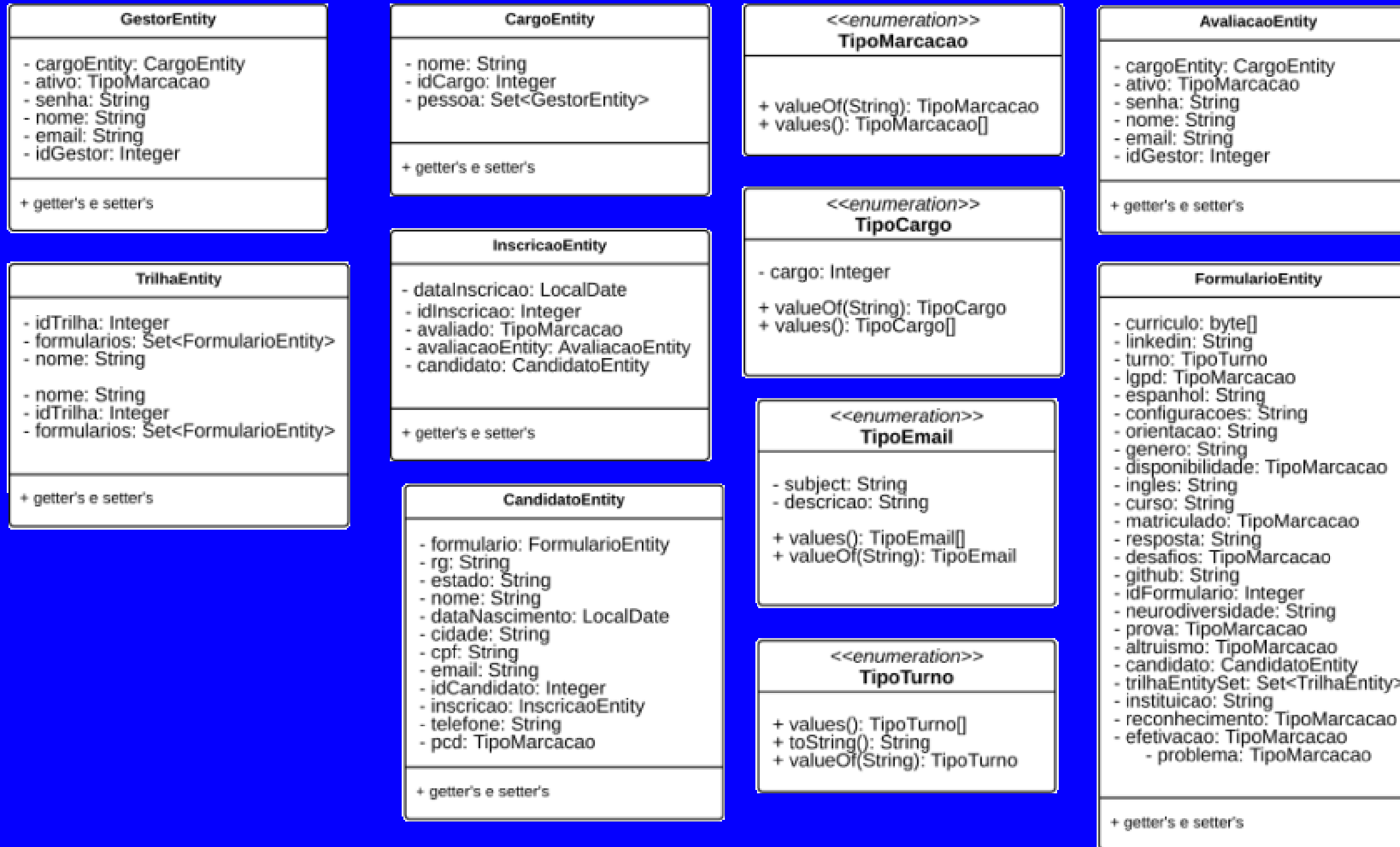
Dashboard/new-user: estilização da página Dashboard/new-user

+ Adicionar um cartão

Testing

+ Adicionar um cartão

Entidades



Services

AvaliacaoService
<ul style="list-style-type: none">- avaliacaoRepository: AvaliacaoRepository- inscricaoService: InscricaoService- gestorService: GestorService- emailService: EmailService- objectMapper: ObjectMapper
<ul style="list-style-type: none">- convertToEntity(AvaliacaoCreateDto): AvaliacaoEntity+ update(Integer, AvaliacaoCreateDto): AvaliacaoDto+ convertToDto(AvaliacaoEntity): AvaliacaoDto+ deleteById(Integer): void+ list(Integer, Integer, String, int): PageDto<AvaliacaoDto>- findById(Integer): AvaliacaoEntity+ create(AvaliacaoCreateDto): AvaliacaoDto+ findAvaliacaoByCandidatoEmail(String): List<AvaliacaoDto>

InscricaoService
<ul style="list-style-type: none">- inscricaoRepository: InscricaoRepository- candidatoService: CandidatoService- emailService: EmailService- objectMapper: ObjectMapper
<ul style="list-style-type: none">- findById(Integer): InscricaoEntity- convertToEntity(InscricaoCreateDto): InscricaoEntity+ convertToEntity(InscricaoDto): InscricaoEntity+ delete(Integer): void+ converterParaDTO(InscricaoEntity): InscricaoDto+ findDtoById(Integer): InscricaoDto+ findInscricaoPorEmail(String): List<InscricaoDto>+ create(InscricaoCreateDto): InscricaoDto+ listar(Integer, Integer, String, int): PageDto<InscricaoDto>+ setAvaliado(Integer): InscricaoEntity

GestorService
<ul style="list-style-type: none">- gestorRepository: GestorRepository- cargoService: CargoService- passwordEncoder: PasswordEncoder- emailService: EmailService- objectMapper: ObjectMapper
<ul style="list-style-type: none">+ findDtoById(Integer): GestorDto+ remover(Integer): void+ forgotPassword(GestorEmailDto): void+ cadastrar(GestorCreateDto): GestorDto+ convertToEntity(GestorDto): GestorEntity+ listar(Integer, Integer, String, int): PageDto<GestorDto>+ editar(Integer, GestorCreateDto): GestorDto+ findGestorbyNomeOrEmail(GestorEmailNomeCargoDto): List<GestorDto>+ contasInativas(): List<GestorDto>- convertToEntity(GestorCreateDto): GestorEntity+ convertToDto(GestorEntity): GestorDto+ findByEmail(String): GestorEntity+ editarSenha(Integer, GestorSenhaDto): GestorDto- findById(Integer): GestorEntity<ul style="list-style-type: none">+ desativarConta(Integer): GestorDto

CandidatoService
<ul style="list-style-type: none">- candidatoRepository: CandidatoRepository- formularioService: FormularioService- objectMapper: ObjectMapper
<ul style="list-style-type: none">+ listaAllPaginado(Integer, Integer, String, int): PageDto<CandidatoDto>+ findDtoById(Integer): CandidatoDto+ convertToDto(CandidatoEntity): CandidatoDto+ findCandidatoDtoByEmail(String): List<CandidatoDto>+ update(Integer, CandidatoCreateDto): CandidatoDto+ findById(Integer): CandidatoEntity+ cadastro(CandidatoCreateDto): CandidatoDto+ convertToEntity(CandidatoDto): CandidatoEntity+ deleteById(Integer): void- convertToEntity(CandidatoCreateDto): CandidatoEntity

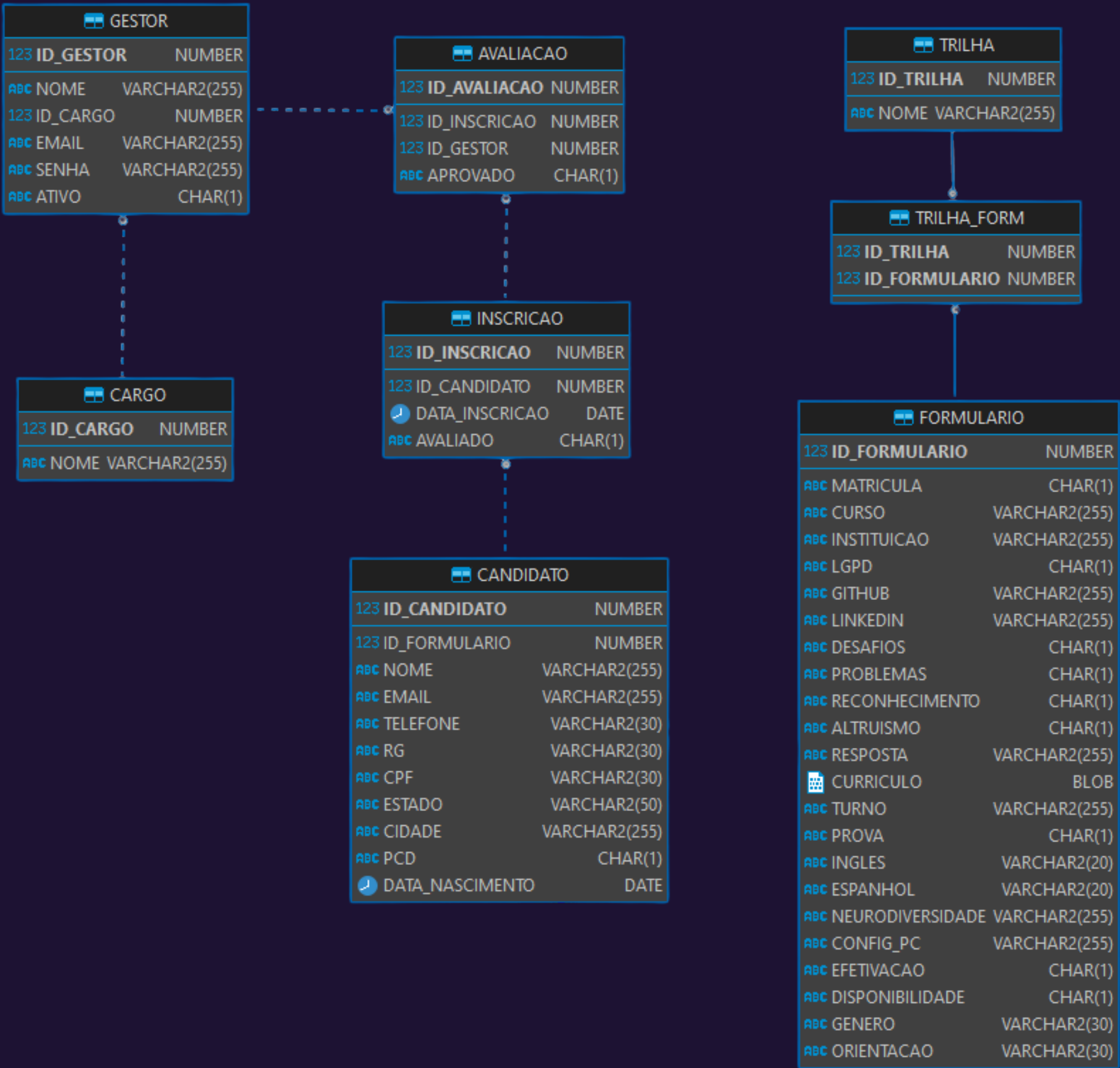
FormularioService
<ul style="list-style-type: none">- formularioRepository: FormularioRepository- trilhaService: TrilhaService- objectMapper: ObjectMapper
<ul style="list-style-type: none">+ findById(Integer): FormularioEntity+ retornarCurriculoDoCandidatoDecode(Integer): String+ listaAllPaginado(Integer, Integer, String, int): PageDto<FormularioDto>+ findDtoById(Integer): FormularioDto+ create(FormularioCreateDto): FormularioDto+ update(Integer, FormularioCreateDto): FormularioDto+ updateCurriculo(MultipartFile, Integer): void+ convertToDto(FormularioEntity): FormularioDto- convertToEntity(FormularioCreateDto): FormularioEntity+ deleteById(Integer): void+ convertToEntity(FormularioDto): FormularioEntity

EmailService
<ul style="list-style-type: none">- javaMailSender: JavaMailSender
<ul style="list-style-type: none">+ sendEmail(SendEmailDto, TipoEmail): void+ getContentFromTemplate(SendEmailDto, TipoEmail): String

CargoService
<ul style="list-style-type: none">- cargoRepository: CargoRepository- objectMapper: ObjectMapper
<ul style="list-style-type: none">+ findById(Integer): CargoEntity+ convertToDto(CargoEntity): CargoDto+ convertToEntity(CargoDto): CargoEntity

TrilhaService
<ul style="list-style-type: none">- tilhaRepository: TrilhaRepository- objectMapper: ObjectMapper
<ul style="list-style-type: none">+ delete(Integer): void- findById(Integer): TrilhaEntity+ list(): List<TrilhaDto>- convertToDto(TrilhaEntity): TrilhaDto+ convertToEntity(Set<TrilhaDto>): Set<TrilhaEntity>+ create(TrilhaCreateDto): TrilhaDto+ findListaTrilhas(List<Integer>): Set<TrilhaEntity>

Diagrama de Entidade e Relacionamento



Desafios e aprendizados

Trabalhar com arquivos

Conhecer formas de trabalhar com os mais diversos tipos de arquivos.

Comunicação em equipe

Manter a coordenação e o bom funcionamento do trabalho, independente das rotinas.

Integração entre as áreas da tecnologia

Integrar diferentes stacks com o objetivo de atingir um melhor desenvolvimento.

Obrigado
DB

