Let's Tech Up Together





Alain Delon

Quality Assurance







**Eduardo Sedrez** 

**Eduardo Queiroz** 

Quality Assurance





**Lucas Miguel** 

Back End

Back End

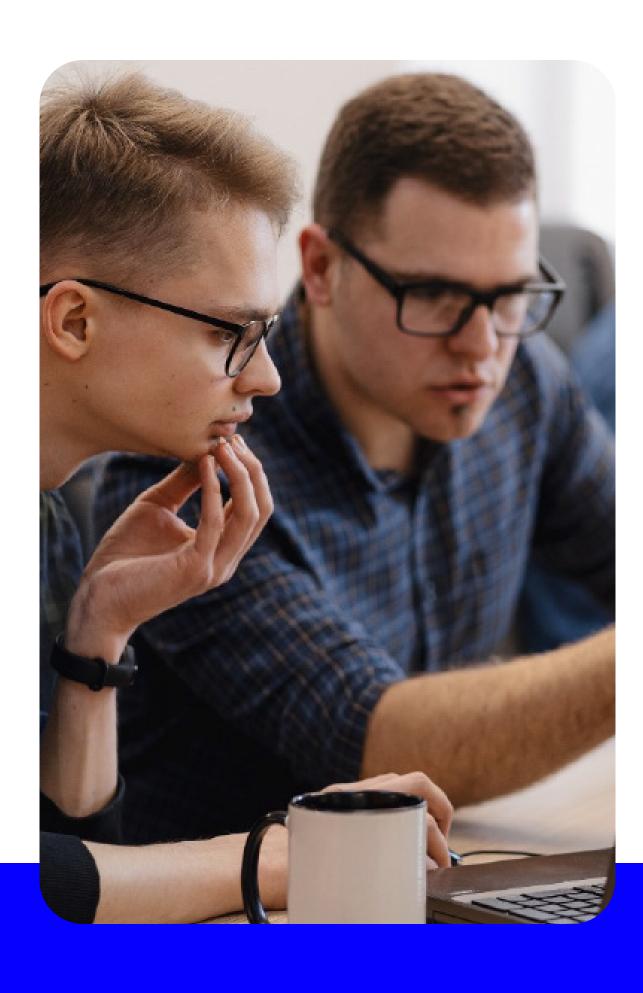
Ricardo Langbecker

Back End





Product Owner

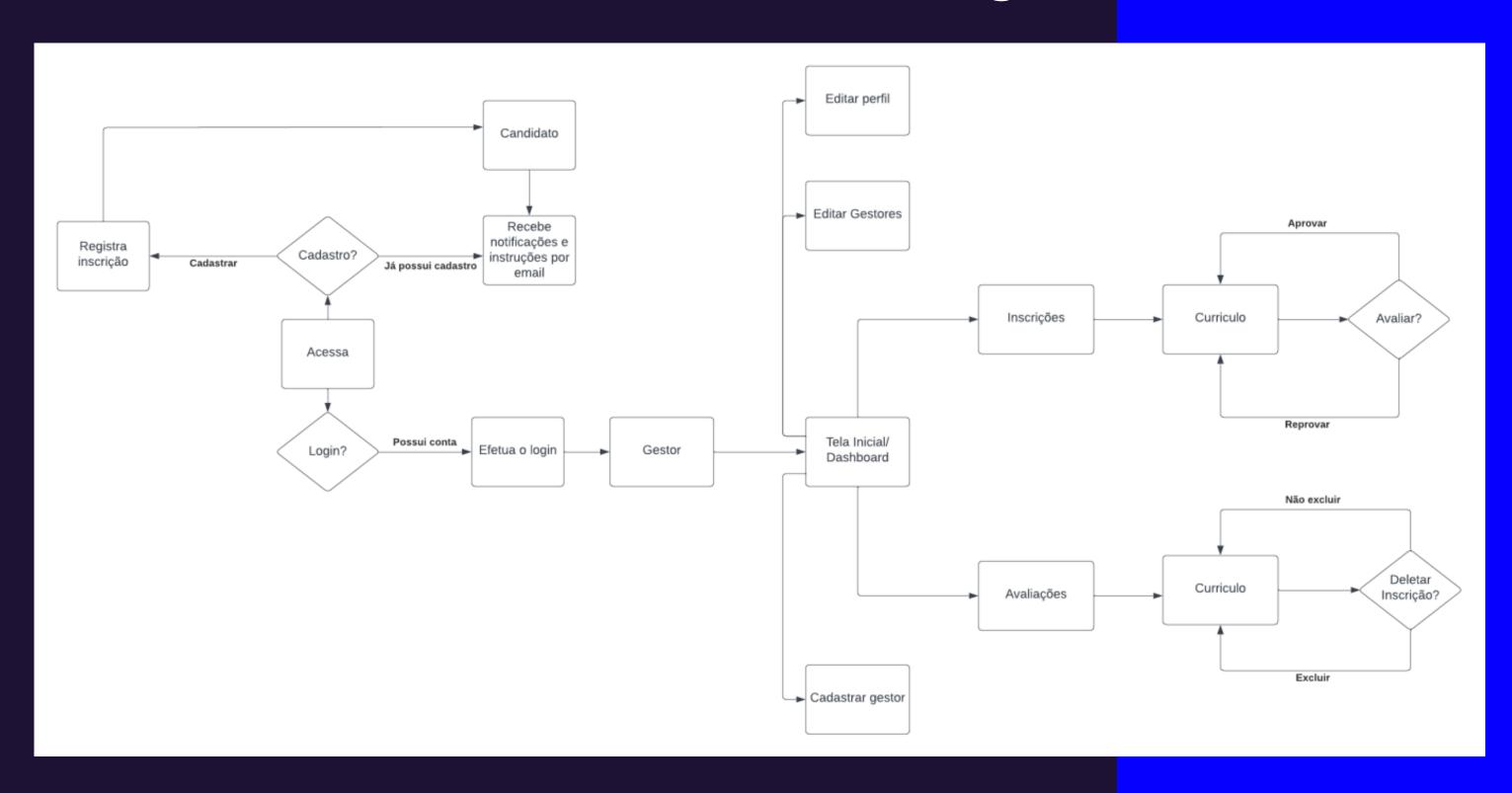


# Projeto VemVemser

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



## ORACLE





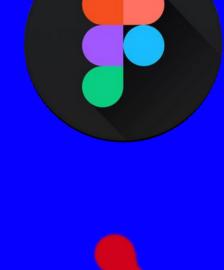
Vercel

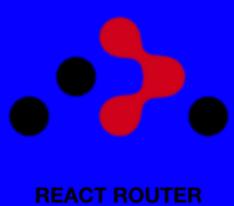
X 1 O S



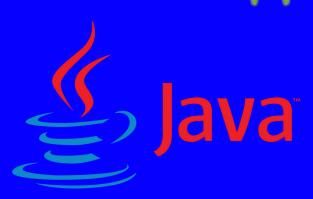












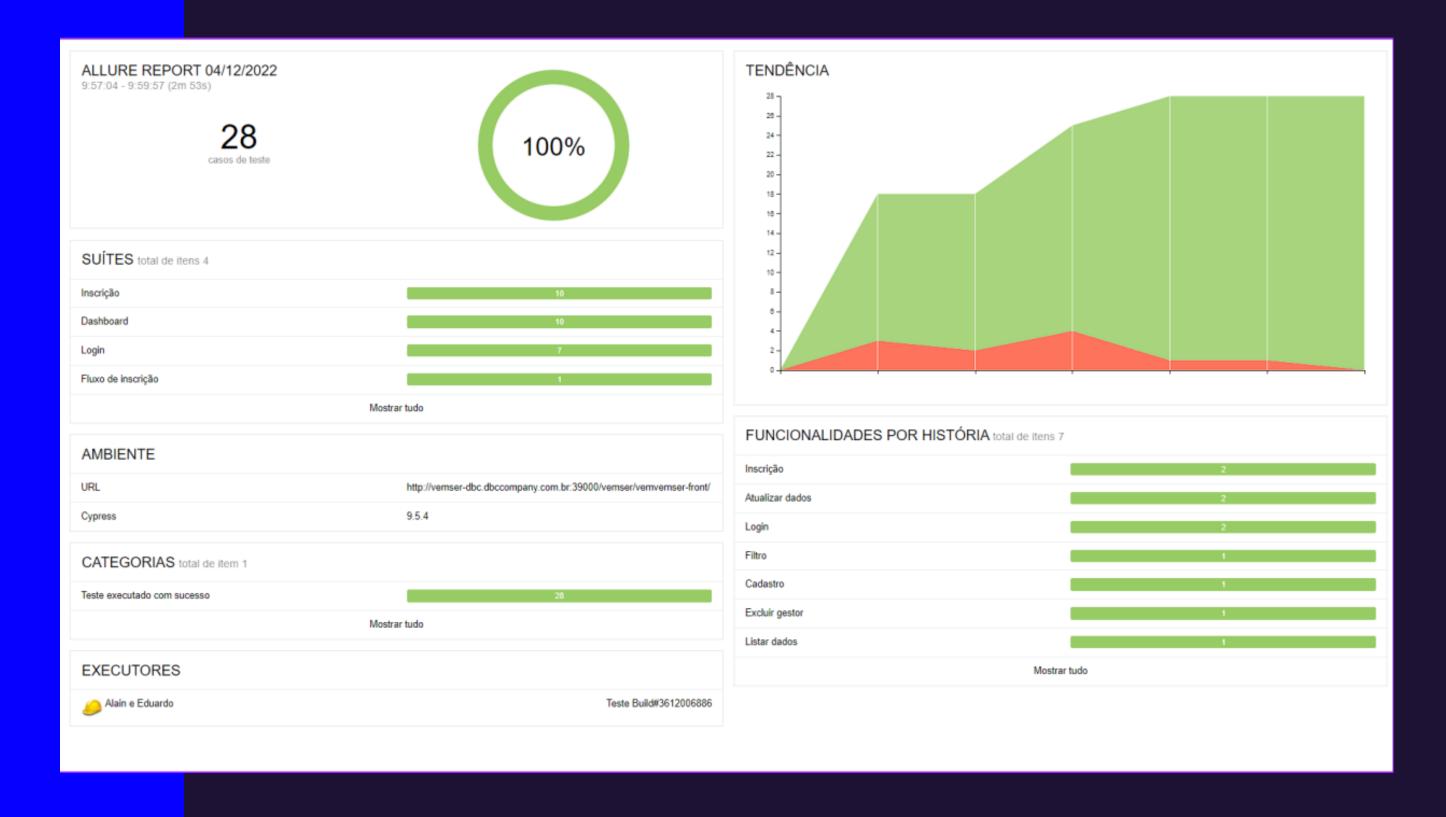




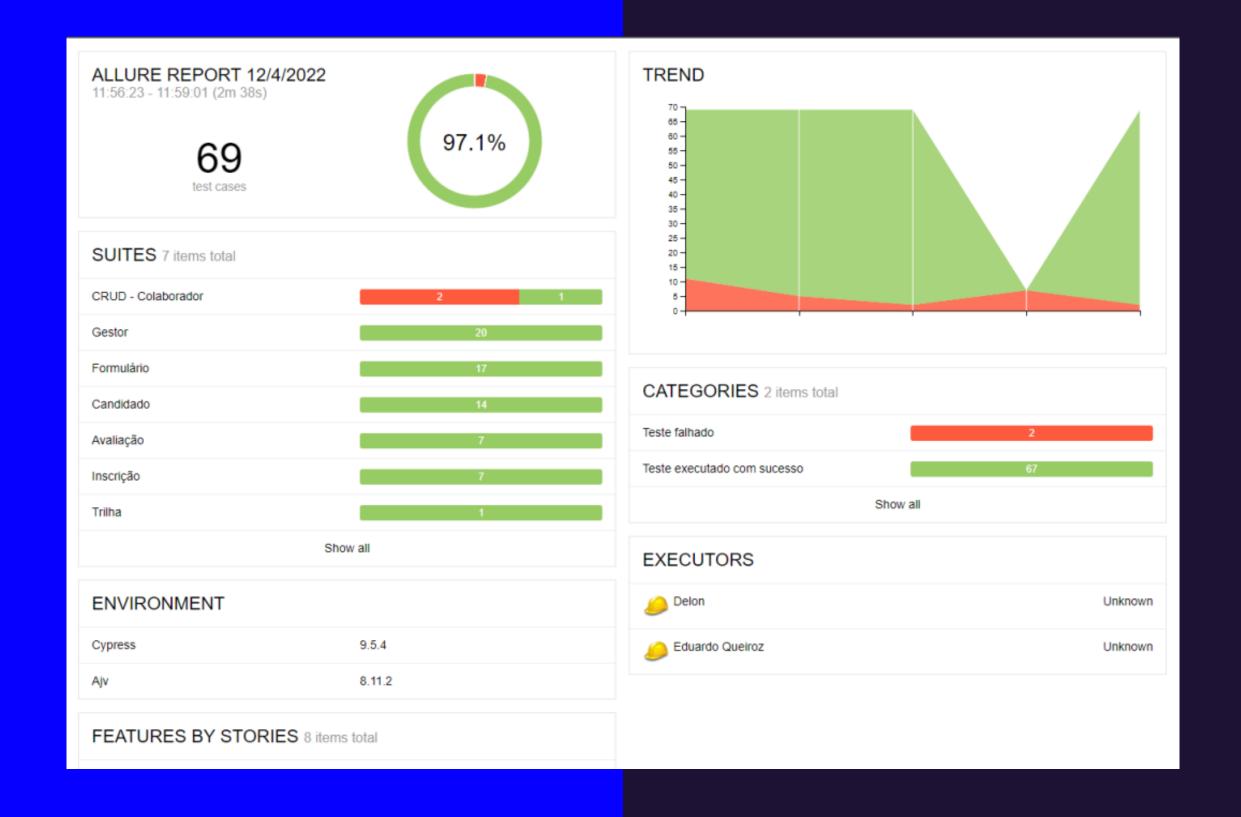




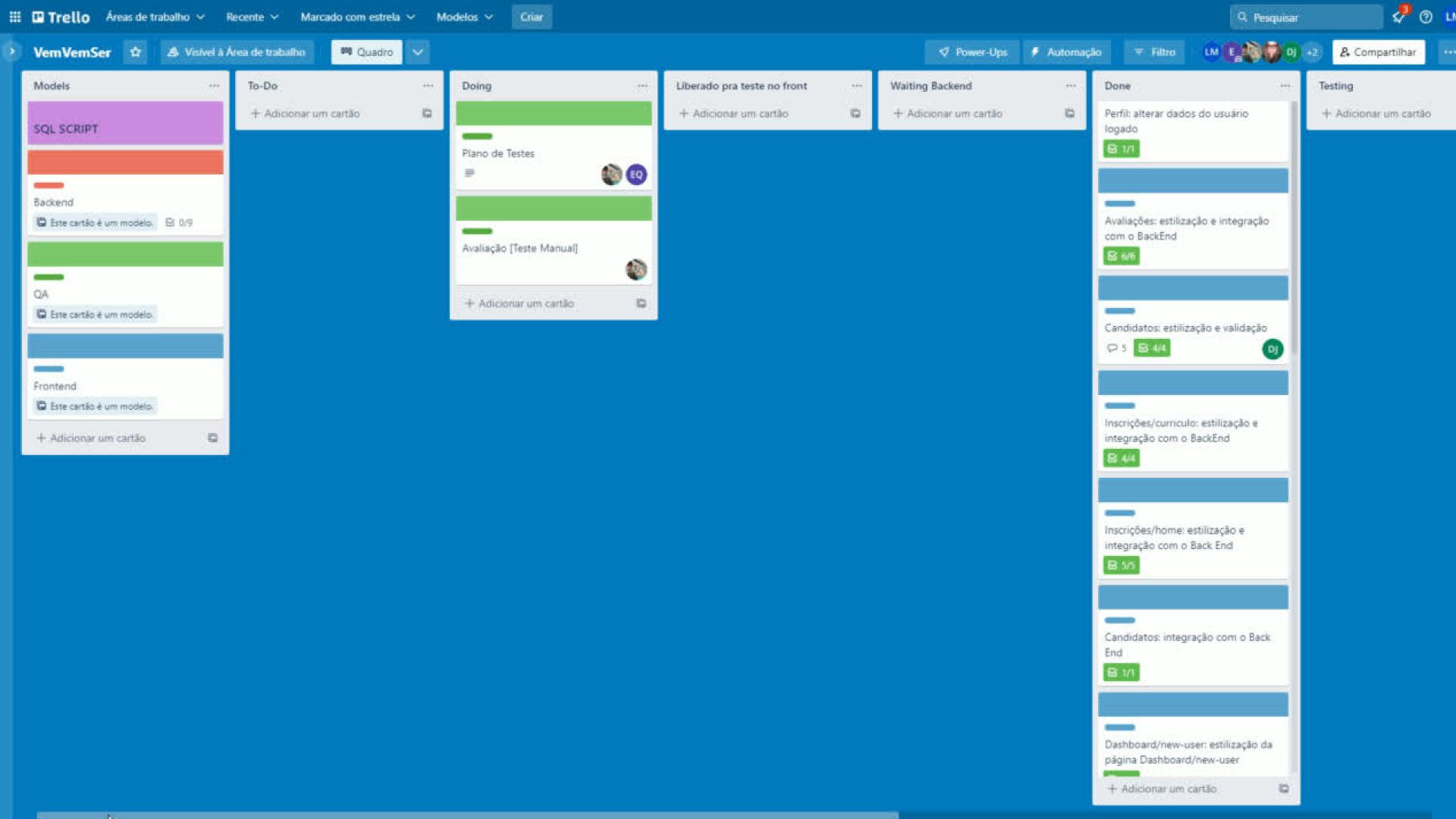
# **Testes - Front End**



# Testes - Back End







### Entidades

#### GestorEntity

- cargoEntity: CargoEntity ativo: TipoMarcacao
- senha: String
- nome: String
- email: String idGestor: Integer
- getter's e setter's

#### TrilhaEntity

- idTrilha: Integer
- formularios: Šet<FormularioEntity>
- nome: String
- nome: String
- idTrilha: Integer
- formularios: Šet<FormularioEntity>
- + getter's e setter's

#### CargoEntity

- nome: String idCargo: Integer
- pessoa: Set<GestorEntity>
- getter's e setter's

#### InscricaoEntity

- dataInscricao: LocalDate
- idInscricao: Integer
- avaliado: TipoMarcacao
- avaliacaoEntity: AvaliacaoEntity
- candidato: CandidatoEntity
- getter's e setter's

#### CandidatoEntity

- formulario: FormularioEntity
- rg: String
- estado: String
- nome: String
- dataNascimento: LocalDate
- cidade: String
- cpf: String
- email: String idCandidato: Integer
- inscricao: InscricaoEntity
- telefone: String
- pcd: TipoMarcacao
- getter's e setter's

#### <<enumeration>> TipoMarcacao

- + valueOf(String): TipoMarcacao + values(): TipoMarcacao[]
  - <<enumeration>>

#### TipoCargo

- cargo: Integer
- + valueOf(String): TipoCargo + values(): TipoCargo[]

#### <<enumeration>>

#### TipoEmail

- subject: String
- descricao: String
- + values(): TipoEmail[] + valueOf(String): TipoEmail
  - <<enumeration>> TipoTurno

#### + values(): TipoTurno[]

- + toString(): String + valueOf(String): TipoTurno

#### AvaliacaoEntity

- cargoEntity: CargoEntity
- ativo: TipoMarcacao
- senha: String
- nome: String
- email: String idGestor: Integer
- getter's e setter's

#### FormularioEntity

- curriculo: byte∏
- linkedin: String
- turno: TipoTurno
- Igpd: TipoMarcacao espanhol: String
- configuracoes: String
- orientacao: String genero: String
- disponibilidade: TipoMarcacao
- ingles: String
- curso: String
- matriculado: TipoMarcacao
- resposta: String
- desafios: TipoMarcacao
- github: String
- idFormulario: Integer
- neurodiversidade: String
- prova: TipoMarcacao
- altruismo: TipoMarcacao candidato: CandidatoEntity
- trilhaEntitySet: Set<TrilhaEntity>
- instituicaó: String
- reconhecimento: TipoMarcacao
- efetivacao: TipoMarcacao problema: TipoMarcacao
- detter's e setter's

# Services

#### AvaliacaoService

- avalicacaoRepository: AvaliacaoRepository
- inscricaoService: InscricaoService
- gestorService: GestorService
- emailService: EmailService
- objectMapper: ObjectMapper
- convertToEntity(AvaliacaoCreateDto):
- AvaliacaoEntity
- + update(Integer, AvaliacaoCreateDto): AvaliacaoDto
- + convertToDto(AvaliacaoEntity): AvaliacaoDto
- + deleteByld(Integer): void
- + list(Integer, Integer, String, int):
- PageDto<AvaliacaoDto>
- findById(Integer): AvaliacaoEntity + create(AvaliacaoCreateDto): AvaliacaoDto
- + findAvaliacaoByCanditadoÉmail(String):
- List<AvaliacaoDto>

#### CandidatoService

- candidatoRepository: CandidatoRepository
- formularioService: FormularioService
- objectMapper: ObjectMapper
- + listaAllPaginado(Integer, Integer, String, int): PageDto<CandidatoDto>
- + findDtoByld(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

#### InscricaoService

- inscricaoRepository: InscricaoRepository
- candidatoService: CandidatoService
- emailService: EmailService
- objectMapper: ObjectMapper
- 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

#### FormularioService

- formularioRepository: FormularioRepository
- trilhaService: TrilhaService
- objectMapper: ObjectMapper
- + findByld(Integer): FormularioEntity + retornarCurriculoDoCandidatoDecode(Integer):
- + listAllPaginado(Integer, Integer, String, int): PageDto<FormularioDto>
- + findDtoByld(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

#### GestorService

- gestorRepository: GestorRespository
- cargoService: CargoService
- passwordEncoder: PasswordEncoder
- emailService: EmailService
- objectMapper: ObjectMapper
- findDtoById(Integer): GestorDto
- remover(Intèger): 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
- findByld(Integer): GestorEntity
  - + desativarConta(Integer): GestorDto

#### **EmailService**

- javaMailSender: JavaMailSender
- + sendEmail(SendEmailDto, TipoEmail): void
- geContentFromTemplate(SendEmailDto, TipoEmail): String

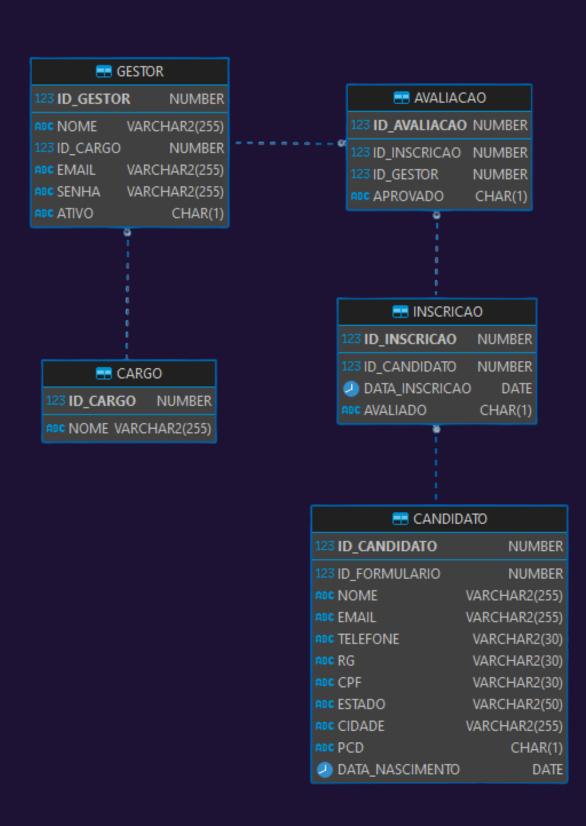
#### CargoService

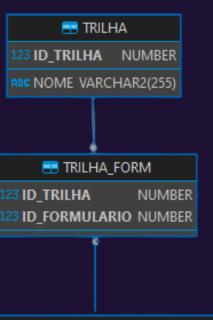
- cargoRespository: CargoRepository objectMapper: ObjectMapper
- + findByld(Integer): CargoEntity + convertToDto(CargoEntity): CargoDto
- + convertToEntity(CargoDto): CargoEntity

#### TrilhaService

- tilhaRepository: TrilhaRepository objectMapper: ObjectMapper
- + delete(Integer): void
- findByld(Integer): TrilhaEntity+ list(): List<TrilhaDto>
- convertToDTo(TrilhaEntity):
- TrilhaDto
- convertToEntity(Set<TrilhaDto>): Set<TrilhaEntity>
- + create(TrilhaCreateDto):
- TrilhaDtò
- + findListaTrilhas(List<Integer>): Set<TrilhaEntity>

# elacionamen 4





➡ FORMULARIO	
123 ID_FORMULARIO	NUMBER
ABC MATRICULA	CHAR(1)
ABC CURSO	VARCHAR2(255)
ABC INSTITUICAO	VARCHAR2(255)
ABC LGPD	CHAR(1)
ABC GITHUB	VARCHAR2(255)
ABC LINKEDIN	VARCHAR2(255)
ABC DESAFIOS	CHAR(1)
ABC PROBLEMAS	CHAR(1)
ABC RECONHECIMENTO	CHAR(1)
ABC ALTRUISMO	CHAR(1)
ABC RESPOSTA	VARCHAR2(255)
curriculo	BLOB
ABC TURNO	VARCHAR2(255)
ABC PROVA	CHAR(1)
ABC INGLES	VARCHAR2(20)
ABC ESPANHOL	VARCHAR2(20)
RBC NEURODIVERSIDADE	VARCHAR2(255)
ABC CONFIG_PC	VARCHAR2(255)
ABC EFETIVACAO	CHAR(1)
ABC DISPONIBILIDADE	CHAR(1)
ABC GENERO	VARCHAR2(30)
ABC ORIENTACAO	VARCHAR2(30)



# 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 B B

