DOMAIN DRIVEN DESIGN USING JAVA

Safe Work — Radar de bem-estar e fadiga de agenda para times híbridos.

Integrantes:

Nome: Enzo Monteiro Maciel
RM: 563734

Nome: Gabriel Cabral Mendes Mariano
RM: 523230

Nome: Matheus de Almeida Sousa
RM: 563537

SUMÁRIO

1. Objetivo e escopo do projeto

O projeto **Safe Work** tem como objetivo reduzir o **no-show** e melhorar o **bem-estar dos times** por meio de check-ins periódicos dos colaboradores, geração de indicadores de saúde do time e recomendações de ações para os líderes.

A **API Java**, desenvolvida com o framework **Quarkus**, é responsável por:

- Receber e armazenar dados de check-in dos colaboradores.
- Gerar e disponibilizar indicadores consolidados de bem-estar por time.
- Permitir o cadastro e a gestão de times.
- Expor serviços REST para serem consumidos pelo front-end da solução.

O escopo da API contempla módulos de:

- Autenticação de usuários.
- Gestão de times.
- Registro e consulta de check-ins.
- Cálculo e exposição de indicadores por time.

---

2. Visão geral da solução Safe Work

A solução Safe Work é composta por três camadas principais:

- **Front-end Web**: interface utilizada por colaboradores e líderes para registrar check-ins, visualizar indicadores e acompanhar a saúde do time.
- **API Java (Safe Work API)**: desenvolvida com Quarkus 3.15.1, expõe endpoints REST para o front-end consumir.
- **Banco de dados**: H2 em memória (na sprint), responsável pelo armazenamento de usuários, times, check-ins, recomendações e ações aplicadas.

Fluxo principal de uso:

1. O usuário acessa o front-end e realiza login, que chama o endpoint POST /api/auth/login.
2. Colaboradores registram seus check-ins em POST /api/checkins, informando humor, data/hora e comentários.

3. Gestores acessam dashboards que consomem dados de GET /api/teams e GET /api/indicadores/teams/{timeId} para acompanhar indicadores de bem-estar do time.
4. A partir dos indicadores, são sugeridas recomendações e ações para melhorar o clima e reduzir o absenteísmo.

---

3. Principais funcionalidades da solução

3.1 Autenticação

- Autenticação simples de usuários via POST /api/auth/login.
- Retorno de dados básicos do usuário autenticado, utilizado pelo front-end para controle de navegação.

3.2 Gestão de Times

- Cadastro de novos times.
- Listagem dos times existentes com suas informações principais (nome, líder, descrição).
- Integração com os módulos de check-ins e indicadores.

3.3 Registro de Check-ins

- Registro de check-ins periódicos pelos colaboradores, com campos como:
    - Data e hora do check-in.
    - Nível de humor/bem-estar.
    - Comentários adicionais.
    - Time ao qual o colaborador pertence.
- Consulta de check-ins cadastrados para uso em telas de histórico e indicadores.

3.4 Indicadores de Bem-estar

- Cálculo de indicadores agregados por time, como:
    - Média de humor em determinado período.
    - Tendência de melhora ou piora.
    - Possíveis alertas de risco para o time.

- Exposição dos indicadores via endpoint específico para ser consumido pelo front-end em gráficos e dashboards.

---

4. Arquitetura da aplicação Java

A API Safe Work foi desenvolvida utilizando **arquitetura em camadas**, com separação clara de responsabilidades.

Pacote base: com.safework.

- **Camada de domínio (domain.model)**
  Contém as entidades de negócio:
  - Usuario
  - Time
  - Checkin
  - Recomendacao
  - AcaoAplicada
  - IndicadorTimeDTO (objeto para transportar indicadores de um time)
- **Camada de repositórios (infra.repo)**
  Repositórios JPA responsáveis por interagir com o banco de dados H2, realizando operações de CRUD e consultas específicas.
- **Camada de serviços (domain.service)**
  Implementa as regras de negócio da aplicação, por exemplo:
  - Validação dos dados recebidos pelos recursos REST.
  - Orquestração entre diferentes repositórios.
  - Cálculo de indicadores de bem-estar por time.
- **Camada de recursos web (web.resource)**
  Disponibiliza os endpoints REST consumidos pelo front-end:
  - AuthResource
  - CheckinResource
  - TimeResource
  - IndicadorResource
- **Tratamento de exceções (web.exception)**
  Concentra classes para:

- Exceções de negócio (BusinessException).
- Mapeamento de erros para respostas HTTP amigáveis.
- Configuração de CORS para permitir o acesso do front-end.

Tecnologias utilizadas:

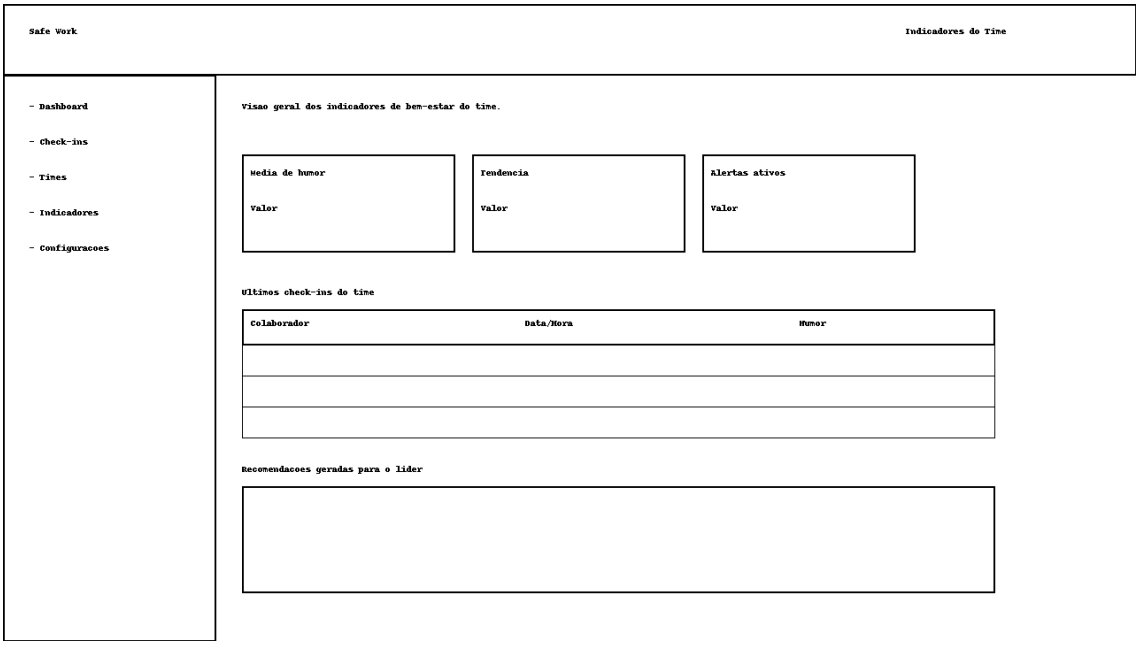
- Java 17
- Quarkus 3.15.1
- H2 Database (em memória)
- Hibernate ORM
- Bean Validation (Hibernate Validator)
- OpenAPI/Swagger UI para documentação automática

5. Tabela de Endpoints (API RESTful)

Recurso	Método	URI	Descrição	Códigos de resposta (exemplos)
Auth Resource	POST	/api/auth/login	Autentica o usuário e retorna seus dados básicos	200, 400, 401, 500
Checkin Resource	GET	/api/checkins	Lista todos os check-ins cadastrados	200, 500
Checkin Resource	POST	/api/checkins	Registra um novo check-in	201, 400, 500
Time Resource	GET	/api/teams	Lista todos os times cadastrados	200, 500
Time Resource	POST	/api/teams	Cadastra um novo time	201, 400, 500
Indicador Resource	GET	/api/indicadores/teams/{timeId}	Retorna indicadores consolidados de bem-estar do time	200, 404, 500

A documentação detalhada dos endpoints, com modelos de entrada e saída, é gerada automaticamente pelo Quarkus e está disponível pelo Swagger UI.

6. Protótipo


Safe Work
Indicadores do Time
- Dashboard
- Check-ins
- Times
- Indicadores
- Configuracoes
Visao geral dos indicadores de bem-estar do time.
Media de humor
Valor
Tendencia
Valor
Alertas ativos
Valor
Ultimos check-ins do time
Colaborador
Data/Hora
Humor
Recomendacoes geradas para o lider

Safe Work

Times

- Dashboard
- Check-ins
- Times
- Indicadores
- Configuracoes

Lista de times cadastrados na orgnmizacao.

+ Novo time

Nome do time	Lider	Colaboradores
Time 1	Lider 1	10
Time 2	Lider 2	10
Time 3	Lider 3	10
Time 4	Lider 4	10

Safe Work

Novo Check-in

- Dashboard
- Check-ins
- Times
- Indicadores
- Configuracoes

Registrar check-in de bem-estar do colaborador.

Como voce esta se sentindo hoje?

1 2 3 4 5

Data e hora do check-in

Comentario (opcional)

Salvar check-in

Safe Work

Login

- Dashboard
- Check-ins
- Times
- Indicadores
- Configuracoes

Acesse sua conta para registrar e acompanhar os check-ins do time.

E-mail

Senha

Lembrar de min

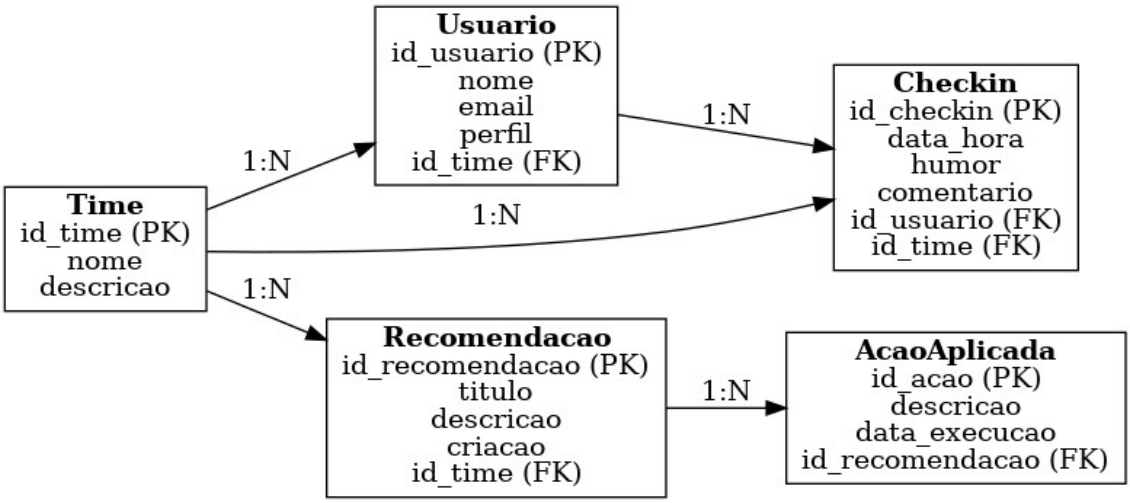
Esqueceu a senha?

Entrar

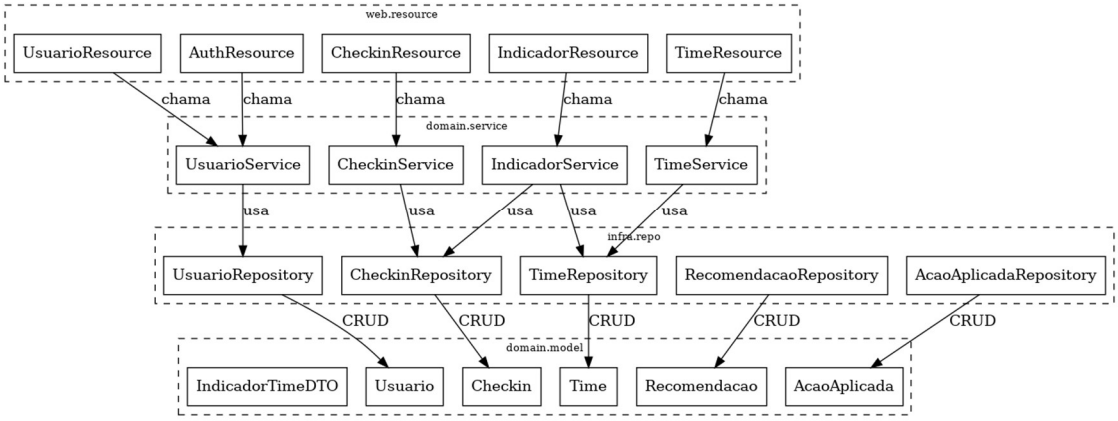
Dica:

Use seu e-mail corporativo
para acessar o Safe Work.

7. Modelo Entidade–Relacionamento (MER)



8. Diagrama de classes



9. Deploy da aplicação na nuvem (Render)

A API Safe Work foi implantada na plataforma **Render**, utilizando um **Web Service** configurado com **runtime Docker**.

9.1 Processo de build

- O projeto utiliza **Maven** para empacotamento da aplicação Quarkus (mvn package).
- O Dockerfile multi-stage, localizado no módulo safework-quarkus, realiza:
    1. Build da aplicação em uma imagem base com JDK.
    2. Geração dos artefatos na pasta target/quarkus-app.

3. Criação de uma imagem final com JRE, copiando apenas os artefatos necessários.

9.2 Configuração na Render

- **Repositório GitHub:** EnzoMMaciel10/safework-quarkus-api
- **Runtime:** Docker
- **Root Directory:** safework-quarkus
- **Porta de execução:** 8080 (detectada automaticamente pelo Render)
- **Branch de deploy:** main
- **Tipo de instância:** Free (para fins acadêmicos)

9.3 URL pública

A API está disponível publicamente em:

- **Base da API:** https://safework-quarkus-api.onrender.com/api
- **Swagger UI (documentação interativa):** https://safework-quarkus-api.onrender.com/q/swagger-ui

Essa configuração atende ao requisito de publicação em nuvem, permitindo que o front-end consuma a API de qualquer lugar.

10. Links de entrega

Para facilitar a correção e atender à "Forma de Entrega" da disciplina, seguem os principais links:

Repositório GitHub (código fonte Java):

https://github.com/EnzoMMaciel10/safework-quarkus-api

Deploy da API (Render):

https://safework-quarkus-api.onrender.com

Swagger UI – documentação dos endpoints:

https://safework-quarkus-api.onrender.com/q/swagger-ui

Link do Pitch (YouTube ou equivalente):

- link do pitch: https://youtu.be/dRZ0pQMAKjM

Link do Vídeo de Demonstração da aplicação funcionando:

link da demonstração: https://www.youtube.com/watch?v=f2WyMzDYynQ