

PROJETO INTEGRADOR – REQUISITOS DO CLIENTE – EMPRESA IACIT

2o. Semestre / 2020

Curso: Análise e Desenvolvimento de Sistemas (ADS)

Semestre: 2º

Tema do Semestre: CRUD Simples Não obrigatoriamente Web, Requisitos não funcionais (Java, BD Relacional)

Conhecimentos exigidos no Semestre (Hard Skills):

- Levantamento de requisitos
- Documentação de software (User Stories, Wireframes)
- Fundamentos de Banco de Dados (Relacional e NoSQL)
- Modelagem de dados (Relacional e NoSQL)
- Linguagem de programação
- Técnicas de apresentação

Problema (Desafio):

(Desenvolver um sistema de gerenciamento de **Controle de Jornada de Trabalho de Motoristas** de forma parametrizável visando a disponibilização de funcionalidades de planejamento, acompanhamento e controle. Adicionalmente, relatórios analíticos são pontos interessantes de agregação de valor ao projeto.)

Requisitos Funcionais:

- **Cadastros básicos:**
 - ✓ **Cadastro dos dados do motorista:** como o nome completo dele, CPF, matrícula e ID de identificação do equipamento do caminhão
 - ✓ **Cadastro dos dados do veículo:** como a placa, marca do rastreador e versão, bem como o ID de identificação de integração da tecnologia de rastreamento instalado no veículo;
 - ✓ **Escala dos motoristas:** (turno, carga horária diária, dias de trabalho e folga);
 - ✓ **Criar um cadastro de alertas:** Contendo as informações como ocorrência, sigla, ícone, descrição do alerta (nomenclatura da geração do alerta) e regra / parâmetro;
 - ✓ **Cadastro de Status:** Cadastrar todos os status e regras de negócio de acordo com cada situação ocorrida durante a jornada.

- ✓ **ID dos macros/ mensagens conforme a tecnologia de rastreamento e associação de eventos de jornada para cada macro/ mensagem** (status de início e fim de jornada, tempo de trabalho, tempo de espera, tempo de refeição, tempo de descanso, etc...)

Requisitos Não Funcionais:

- Linguagem Java
- Banco de Dados Relacional
- Documentações (Especificação, Plano de Teste, Manual do Usuário)
- Linguagem **Backend**: Java (JDK 14) com ORM Hibernate;
- Banco de Dados Relacional – Postgres;
- Repositório Git (*Sugestão: Utilizar o Fork de interface para o gerenciamento*);
- Documentações:
 - Levantamento de requisitos,
 - Modelo lógico do banco de dados (*Sugestão: BrModelo*),
 - Modelo conceitual do banco de dados (*Sugestão: MySQL Workbench - apenas para gerar o modelo conceitual -*)
 - Especificação,
 - Plano de teste,
 - Manual do usuário,
 - Wireframes (*Sugestão: Figma*).