SafeCar

Tech by Stellantis

PROJECT

GROUP SAFECAR BY FIAP

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Gabriel Mira, Jhonatas Oliveira e Nathalia Lopes

SafeCar

São Paulo
2022

Gabriel Mira, Jhonatas Oliveira e Nathalia Lopes

SafeCar

Trabalho apresentado no curso de graduação de Análise e Desenvolvimento de Sistemas da Faculdade de Informática e Administração Paulista - FIAP.

São Paulo
2022

SUMÁRIO

INTRODUCTION IN ENGLISH

The case of Global Solutions in partnership with FIAP and Stellanti, becoming a solution that consists of a solution based on Smart and practicality in the day to be intensified, with a solution that brings more safety and practicality to cars on a daily basis .

According to research carried out by ABRANET, it was found that about 42% of accidents caused by vehicles report results of drowsiness and/or the driver.

In view of all this data, we at Safe to develop a technology-based solution to develop an artificial intelligence, our idea is to develop an artificial intelligence that analyzes whether a user is learning at the wheel and possibly will make a call for a series of artificial intelligence in the steering wheel. health agent / be police where.

SafeCar system, fast, being the entire trip of the passengers thus generating, measurements of piloting time, complete trip warnings, useful warnings for applications in others.

The project corresponds to version 1.0, and is in the MVP phase, without some functionalities already in place and in operation, and the others, are the main ones with the next ones of the upcoming Sprints.

INTRODUÇÃO

O case da *Global Solutions* em parceria com a ***FIAP*** e Stellantis propôs uma solução baseada em Smart Mobility na qual consiste em trazer mais mobilidade para automóveis, intensificando sua segurança e praticidade no dia a dia.

De acordo com pesquisas realizadas pela ***ABRANET***, foi levantado que cerca de 42% dos acidentes causados por veículos, discorrem derivados de sonolência e/ou do piloto

Diante de todos esses dados, nós da SafeCar desenvolvemos uma solução baseada em tecnologia para intervir nestes acidentes, nossa ideia é desenvolver uma inteligência artificial que analisa se o usuário está dormindo no volante, disparando uma série de avisos e possivelmente, realizar um chamado para algum agente da saúde / polícia seja onde estiverem.

O sistema SafeCar, irá monitorar toda a viagem dos passageiros, gerando assim, métricas de tempo de pilotagem, avisos e estatísticas completas de sua viagem, sendo úteis para aplicações em outros ramos.

O projeto corresponde na versão 1.0, e está em fase de MVP, no qual algumas funcionalidades já estão em vigor e em funcionamento, e as demais, serão implementadas com o decorrer das próximas Sprints.

SafeCar by Stellantis

Nos dias atuais, a mobilidade é algo que tem sido cada vez mais presente no centro de todas as sociedades mundiais. Com ela, podemos melhorar o acesso à saúde, educação, trabalho e lazer.

Estima-se que 1,2 milhões de pessoas morrem todos os anos por acidentes de tráfego,sem contar os 50 milhões que se ferem, ou seja, 40% da sociedade mundial enfrentam esses problemas.

No Brasil, de acordo com a Associação Brasileira de Medicina do Tráfego (ABRAMET), cerca de 40% dos acidentes provocados no trânsito são derivados de sonolência dos motoristas no veículo, e mais da metade dos acidentes ocorrem em áreas urbanas. Estima-se que cerca de 11 bilhões de viagens são realizadas diariamente ao redor do mundo. Se reduzir em 1% o número de acidentes diariamente, impactariam as pessoas positivamente e reduziria a pressão nos sistemas de saúde, assim, gerando uma qualidade de vida melhor.

A ODS estabeleceu a seguinte meta:

“Até 2030, proporcionar o acesso a sistemas de transporte seguros, acessíveis, sustentáveis e o preço acessível para todos, melhorando a segurança rodoviária por meio de expansão dos transportes públicos, com especial atenção para as necessidades das pessoas em situação de vulnerabilidade, mulheres, crianças, pessoas com deficiência e idosos**”**

O conceito da mobilidade vai muito além do transporte, pois quando bem planejado e implementado, promove mais conforto, qualidade de vida, redução de agentes poluidores, conexão, velocidade, economia e promove um futuro melhor a todos.

Com base nas informações acima, uma das tecnologias mais utilizadas para desenvolvimento da mobilidade é a Inteligência Artificial. Ela pode melhorar a mobilidade das pessoas por meio de técnicas de otimização, como exemplo disso, o uso da sua otimização no trânsito é uma realidade e muitos de nós utilizamos através de aplicativos mobile, com dados que são atualizados em tempo real, todos com auxílio de inteligência artificial.

O grupo Stellantis junto à FIAP, busca por soluções para uma nova realidade sustentável e nós somos atores principais para contribuir com esta transformação no avanço da mobilidade.

Diante da proposta acima, surgiu a SafeCar. Com intuito de ajudar as pessoas a tornar a mobilidade mais acessível, segura e sustentável. Mas de qual forma poderíamos tornar isso possível ?

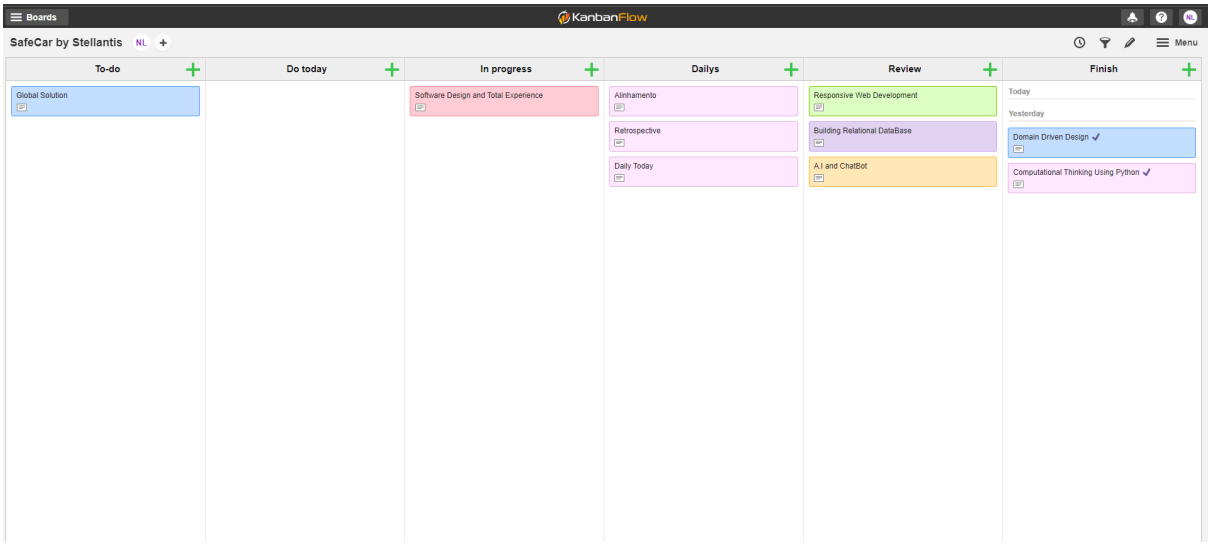
Foi desenvolvido um sistema de I.A visando a integridade e segurança dos pilotos e passageiros, identificando padrões de sonolência e cansaço por longos períodos de deslocamento, gerando métricas, alertas e dependendo da situação acionando unidades hospitalares e policiais para intervirem o quanto antes de um possível acidente acontecer.

Dessa forma a saúde de todos dentro do veículo será colocada como prioridade, trazendo uma experiência e segurança não só para uma família, mas também para a sociedade, para todos que compartilham das mesmas necessidades, seja indo do trabalho ou fazendo viagens de longa duração. Todos são beneficiados.

Proposta de intervenção SafeCar

- Diminuição de índice de mortes no tráfego
- Otimização com as tecnologias mais inovadoras do mercado
- Contribuição para Smart Mobility
- Aumento da segurança e conforto nos veículos
- Promover melhor qualidade de vida e bem-estar a todos
- Contribuição para um mundo mais sustentável
- Promover dominância de mercado em carros tecnológicos
- Impactos positivos na ODS da ONU
- Aumento da confiança dos clientes

KanbanFlow



Acima, está a metodologia de desenvolvimento escolhida pela equipe, onde contém todas as partes que foram desenvolvidas no projeto.

Requisitos Funcionais

- Aplicação web SafeCar
- Api conectando o banco de dados e aplicação
- Banco de dados
- SQL server
- Software de inteligência artificial
- Sistema de cadastro
- Sistema de Login
- Sistema identificador facial

Requisitos Não Funcionais

- Segurança da plataforma
- Segurança do software
- Desempenho do software
- Flexibilidade de identificação do software
- Velocidade na plataforma
- dinamismo na plataforma
- Responsividade da plataforma
- Fácil orientação na plataforma
- Sistema integrado
- Sistema disponível
- Alocação da plataforma no Netlify
- Servidor local
- Internet

Domain Driven Design

O core do projeto Safe-Car foi estruturado em Java, utilizamos tecnologias de criação de API'Restfull para criar uma conexão entre o banco de dados ORACLE e a aplicação web em React. Foi utilizado tecnologias como: TomCat para rodar um servidor localmente, Maven para criar a build estável de todas bibliotecas necessárias, Hibernate para permitir uma conexão com o banco de dados, JDBC para utilizá-los o banco ORACLE.

O projeto constitui classes POJO declarativas de objetos, como usuário, veiculo, plano de saúde, a classe de ConnectionFactory com todos os métodos de conexão necessários para o banco de dados, classes DAO onde ficam os métodos respectivos de cada funcionalidade com o banco de dados e por fim, as classes Resource que aplicam os métodos DAO só que em formato JSON, para a API, com métodos *GET, POST, PUT e DELETE.*

O projeto também possui um menu de escopo reduzido demonstrando as funcionalidades das classes DAO, ele é bem simples pois é apenas para teste.

Responsive Web Development

A implementação de todo o sistema e a parte dinâmica do projeto onde podemos visualizar todos os métodos é o front-end foi utilizado as boas práticas para iniciar um projeto do 0 em React, com todas páginas componentizadas, utilização de React-Router-DOM para gerenciar todas as rotas de páginas trazendo mais velocidade na troca das mesmas. Foi utilizado Styled-Components para gerar cada estrutura de CSS da aplicação, trazendo dinamismo à plataforma em React. O projeto está inteiramente constituído onde cada estrutura é um componente na plataforma. Está sendo consumido as API's do Java no qual interagem diretamente com o banco de dados ORACLE, existem APIs de validação de login, cadastro, listagem de planos de saúde, listagem de usuários cadastrados na plataforma. Nosso diferencial está na aplicação do software onde utilizamos uma Api de inteligência artificial para mapear toda a musculatura do rosto humano e identificar sinais de sonolência e cansaço no piloto. Foi entregue um MVP das primeiras Sprints em apenas 11 dias de desenvolvimento. O projeto ganhará futuramente mais estruturas especializando o diferencial e de fato se tornando uma solução para ser aplicada nos modelos de veículos futuramente.

O projeto possui imagens reativas e responsivas ao projeto, um design simples de tema escuro e de fácil localização, à ideia é transformá-lo em um site institucional onde os usuários/clientes poderão visualizar a solução e testar.

I.A Chatbot

Foi utilizado conceitos de Machine Learning e aprendizados de máquina para realizar uma análise de 2 datasets, envolvendo sensores de movimento de máquinas e velocidades com base nas estradas e ruas.

Foi realizado uma análise profunda dos dados, realizando limpezas no dataset, identificando valores nulos e label encoders, identificando possíveis agrupamentos como clusters.

A análise foi distribuída em modelos supervisionados e não-supervisionados, onde foi levantado cerca de 8 gráficos preditivos explicando as distribuições e médias de cada dado analisado, levando futuros dados e estatísticos sobre os assuntos do dataset, como tendências e posições.

Após o levantamento dos dados supervisionados, foram analisados modelos de aprendizado não supervisionados, onde encontramos possíveis agrupamentos, foi realizado uma reestruturação nos dados sendo eles normalizados e padronizados, focando em buscas exploratórias e associações, discorrendo de todas as ferramentas no alcance.

Building Relational DataBase

O Banco de Dados do projeto SafeCar, armazenará detalhes de chamados via ocorrência do nosso sistema. Contará com informações refinadas dos chamados, com todas características do carro, usuário, contatos, plano de Saúde, endereços, para que assim exista um atendimento mais prático e direto, salvando uma possível vítima de acidente.

As entidades principais do nosso sistema são:

T_SC_VEICULO:

- ***id_veiculo*** *NUMBER(10)* **-** Responsável por armazenar a identificação do carro.
- **qt_tripulantes** *NUMERIC(1)* **-** Responsável por exibir uma previsão de tripulantes no veículo.
- **nm_carro** *VARCHAR(60)* **-** Responsável por armazenar o nome do carro.
- **tp_veiculo** *VARCHAR(20)* **-** Responsável por informar o tipo de veículo (carro, caminhão, moto e etc).
- **ds_cor_veiculo** *VARCHAR(40)* **-** Responsavel por armazenar a cor do veiculo
- **ds_placa** *VARCHAR(7)* **-** Responsável por armazenar a placa do veículo
- **ds_modelo_veiculo** *VARCHAR(40)* **-** Responsável por armazenar a marca do veículo.
- **ds_veiculo** *VARCHAR(200)* **-** Responsável por informar uma descrição completa de informações relevantes particulares do veículo

ID_SC_USUARIO:

- **id_usuario** *NUMBER(10)* **-** Responsável por armazenar a identificação do usuário.

- **nm_usuario** *VARCHAR(40)* **-** Responsável por armazenar o nome do usuário.

- **ds_idade_usuario** *NUMBER(3)* **-** Responsável por armazenar a idade do usuário.

- **ds_profissao_usuario** *VARCHAR(30)* **-** Responsável por armazenar a profissão atual do usuário.

- **ds_salario_usuario** *NUMBER(10)* **-** Responsável por armazenar o salário do usuário.

- **ds_raca_usuario** *VARCHAR(20)* **-** Responsável por armazenar a raça do usuário.

- **ds_genero** *VARCHAR(30)* **-** Responsável por armazenar o gênero do usuário.

- **ds_nacionalidade** *VARCHAR(20)* **-** Responsável por armazenar a nacionalidade do usuário.

- **tp_sanguineo** *VARCHAR(30)* **-** Responsável por armazenar o tipo sanguíneo do usuário.

T_SC_CONTATO:

- **id_contato** *NUMBER(10)* **-** Responsável pela identificação do contato

- **nm_contato** *VARCHAR(30)* **-** Responsável por armazenar o nome do contato

- **nr_telefone** *NUMBER(9)* **-** Responsável por armazenar o número de telefone sem ddd e ddi.

- **ds_ddd** *NUMBER(2)* **-** Responsável por armazenar o ddd do telefone.

- **ds_ddi** *NUMBER(3)* **-** Responsável por armazenar o ddi do telefone
15
- **ds_email** *VARCHAR(50)* **-** Responsável por armazenar o email do contato

T_SC_PLANO_DE_SAUDE

- **id_plano_saude** *NUMBER(10)* **-** Responsável pela identificação do plano de saúde.

- **ds_empresa_plano_saude** *VARCHAR(30)* **-** Responsável por informar a empresa que fornece o plano de saúde.

- **nm_plano_saude** *VARCHAR(40)* **-** Responsável pelo nome do plano de saúde.

- **tp_plano_saude** *VARCHAR(30)* **-** Responsável pelo tipo de plano de saúde, adulto, infantil, idoso.

- **ds_descricao_plano** *VARCHAR(200)* **-** Responsável pela descrição completa do plano de saúde do usuário.

T_SC_ENDERECO

- **id_endereco** *NUMBER(10)* **-** Responsável pela identificação do endereço.

- **ds_estado** *VARCHAR(20)* **-** Responsável pelo estado do endereço

- **ds_cep** *NUMBER(8)* **-** Responsável pelo cep do endereço

- **ds_cidade** *VARCHAR(30)* **-** Responsavel pela cidade do endereço

- **ds_logradouro** VACHAR(60) **-** Responsável pelo logradouro do endereço

- **ds_numero** *VARCHAR(10)* **-** Responsavel pelo número do endereço

- **ds_pais** *VARCHAR(20)* **-** Responsável pelo pais do endereço

16

T_SC_CHAMADO

- **id_chamado** *NUMBER(10)* **-** Responsável pela identificação do chamado

- **tp_ocorrencia** *VARCHAR(20)* **-** Responsável pelo tipo de ocorrência do chamado exp: dormindo no volante

- **ds_grau_ocorrencia** *VARCHAR(20)* **-** Grau da ocorrência identificado pelo sistema.

- **loc_ocorrencia** *VARCHAR(30)* **-** Descrição das localizações da ocorrência, para um atendimento mais preciso

- **ds_coordenadas** *VARCHAR(20)* **-** Descrição das coordenadas onde o veículo foi visto pela última vez.

- **dt_chamado - DATE -** Dados do dia no qual o chamado foi realizado

T_SC_UND_HOSPITALAR

- **id_und_hospitalar** *NUMBER(10)* **-** Responsável pelo id da unidade hospitalar

- **nm_und_hospitalar** *VARCHAR(20)* **-** Responsável pelo nome da unidade

- **ds _endereco _ unidade** *VARCHAR(60)* **-** Responsavel pelo endereço

- **ds_cep_unidade** *NUMBER(8)* **-** Responsável pelo cep da unidade

- **ds_cidade** *VARCHAR(30)* **-** Responsavel pela cidade do endereço

- **ds_logradouro** *VACHAR(60)* **-** Responsável pelo logradouro do endereço

- **ds_numero** *VARCHAR(10)* **-** Responsavel pelo número do endereço

- **ds_estado** *VARCHAR(20)* **-** Responsável pelo estado do endereço

T_SC_CONTATO_EMERGENCIA

- **id_contato_emergencia** *NUMBER(10)* **-** Responsável por armazenar o id do contato de emergência

17

- **tp_contato_emergencia** *VARCHAR(30)* **-** Responsável por armazenar o tipo de contato, policial, ambulatório e entre outros.

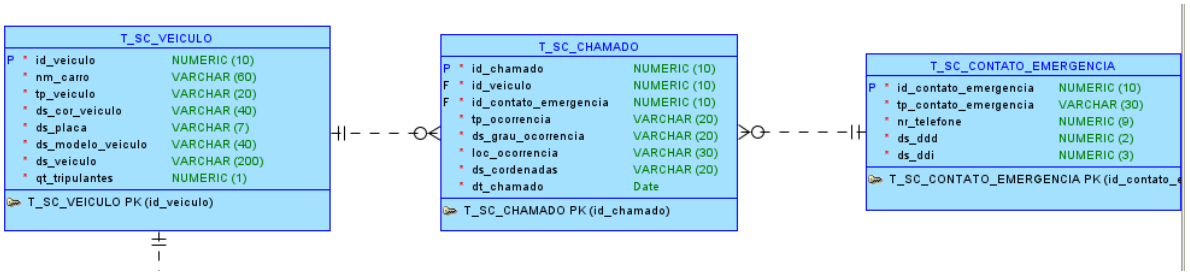
- **nr_telefone** *NUMBER(9)* **-** Responsável por armazenar o número de telefone sem ddd e ddi.

- **ds_ddd** *NUMBER(2)* **-** Responsável por armazenar o ddd do telefone.

- **ds_ddi** *NUMBER(3)* **-** Responsável por armazenar o ddi do telefone

Auto Relacionamento das tabelas

O auto relacionamento das tabelas, foi feito entre a tabela veículo, chamado e contato de emergência, sendo o veículo 1.n e contato emergência 1.n, sendo associados ao chamado



Vários veículos estão associados a vários contatos de emergência, e vários contatos de emergência estão associados a vários veículos.

Computational Thinking Using Python

Nós da SafeCar desenvolvemos um programa que mapeia e faz uma pesquisa da quantidade de acidentes que ocorreram nos anos anteriores. Com ele, o usuário consegue total flexibilidade para registrar, consultar dados cadastrados, ter acesso a um relatório comparativo de um ano escolhido, como o ano de 2019, que foi um dos anos onde ocorreram diversos casos de óbitos relacionados com acidentes de tráfego.

Sendo assim, seguimos utilizando o mesmo para que possamos ter uma visão ampla, analisando se os casos estão subindo ou diminuindo, também podendo trabalhar para que isso seja controlado.

O sistema foi desenvolvido e formado por 9 funções:

- menu principal(): Essa função cria o menu principal com as possíveis N opções para a melhor interação com o usuário.

- cria lista(): Essa função é responsável por criar uma lista que o código utilizará.

- cria relatório(): Essa função é responsável por criar o dicionário onde será armazenado as informações de entrada do usuário.

- cadastra ano mes(): Essa função é responsável por cadastrar o mês e ano onde ocorreram os acidentes, através do formato (mês-ano).

- consulta ano mes(): Essa função é responsável por consultar apenas um mês-ano específico que o usuário cadastrou.

- merge sort(): Essa função é responsável por ordenar a lista total de óbitos que ocorreram em um ano para uma melhor visualização do usuário.

- relatório comparativo(): Essa função é responsável por consultar os resultados de um ano inteiro, fazer e exibir um relatório comparativo com os casos que ocorreram em 2019 e além de também exibir uma tupla do total de óbitos de forma ordenada.
  19
- listar todos(): Essa função é responsável por listar todos os anos cadastrados de forma compreensível, independente do mês e do ano.

- principal(): Essa função será a principal, sendo responsável por executar todas as outras funções do programa.