

UNIVERSIDADE PAULISTA
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA



D2538H2 - JONATHAN SILVA SALES DE OLIVEIRA
N2984J6 - VINICIUS DA SILVA PATO
D654JJ4 - CARLOS HENRIQUE G LEMES ALVES
N272BH2 - ALEX WONG DE FREITAS

ATIVIDADES PRÁTICAS SUPERVISIONADAS

Aplicativo de Prevenção de Incêndios

SÃO PAULO
2021

D2538H2 - JONATHAN SILVA SALES DE OLIVEIRA

N2984J6 - VINICIUS DA SILVA PATO

D654JJ4 - CARLOS HENRIQUE G LEMES ALVES

N272BH2 - ALEX WONG DE FREITAS

ATIVIDADES PRÁTICAS SUPERVISIONADAS

Aplicativo de Prevenção de Incêndios

Trabalho de Desenvolvimento de Sistemas
Distribuídos feito durante o oitavo semestre da
graduação em Ciência da Computação.

Orientador: Prof. Lauro Tomiatti

SÃO PAULO

2021

SUMÁRIO

1	OBJETIVO E MOTIVAÇÃO DO TRABALHO	5
2	INTRODUÇÃO	7
3	FUNDAMENTOS DAS TECNOLOGIAS DE SISTEMAS DISTRIBUÍDOS	9
4	PLANO DE DESENVOLVIMENTO DA APLICAÇÃO	11
4.1	Banco de Dados	11
4.2	Business Intelligence (BI)	12
4.3	Sistema Distribuído – Back-end	13
4.3.1	Spring Boot	13
4.3.2	Spring Framework	13
4.3.3	Spring Data JPA	14
4.3.4	H2 Database	14
4.3.5	Spring Security	15
4.3.6	Spring Web	15
4.3.7	JWT, Validation API, Lombok	16
4.3.8	Maven	16
4.4	Sistema Distribuído – Front-end	16
4.4.1	HTML, CSS e JavaScript	16
4.4.2	NPM	16
4.4.3	Node.js	17
4.4.4	React	17
4.4.5	Create React App	17

1 OBJETIVO E MOTIVAÇÃO DO TRABALHO

A proposta da atividade prática supervisionada é o desenvolvimento de uma aplicação de sistema distribuído para dispositivo móvel, no qual a aplicação deve ser relativa ao gerenciamento de informações urbanas sendo elas alguns problemas recorrentes do meio urbano como enchentes, poluição, desmatamento, invasão térmica entre outros problemas no qual a tecnologia seria de grande ajuda para agilizar a resolução desses problemas com objetivo de amenizar os impactos, até coleta de dados para futuros estudos estatísticos, análise de dados e ciência de dados para implementação de algoritmo de machine learning para análise preditiva.

Sendo assim, o objetivo principal desta atividade que foi proposta a nós é desenvolver esta aplicação mobile para diminuir os impactos urbanos que são de extrema recorrência a ambientes urbanos, o objetivo de forma prática será benéfico tanto para órgãos públicos, os habitantes dos centros urbanos e os agentes dos centros urbanos como policiais, bombeiros, garis, lixeiros, carteiros entre outras membros que fazem todos os centros urbanos funcionar adequadamente, pois os mesmos são de certa forma a base do funcionamento desta grande cadeia.

2 INTRODUÇÃO

De acordo com que foi proposto e apresentado anteriormente o nosso grupo entrou em acordo e possuía o foco principal nos incêndios nos centros urbanos cujo é bem recorrente porém não possuem tanta visibilidade por habitantes e órgãos públicos, realizando análises de dados fornecidos pelo corpo de bombeiros do Estado de São Paulo¹ podemos concluir que de 2016 até 2021 houve uma queda considerável no número de incêndios mas infelizmente o número de incêndios em vegetação ainda permanecesse muito alto, notícias recentes mostram que acidentes domésticos estão relacionado a incêndios, devido à grande crise em que o Brasil está passando no momento algumas famílias estão recorrendo a não usar gás natural como fonte de combustível para cozinhas mas sim lenhas devido ao alto custo do gás, consequentemente com a manipulação deste método o número de acidentes aumentou segundo notícias do G1, Yahoo Notícias, R7 e BBC.

Já os incêndios nas grandes vegetações são devido à grande expansão dos centros urbanos, para que grandes empresas, empresários ou até mesmo moradores destes centros urbanos consigam expandir suas casas, empresas para centros comerciais.

De acordo com que foi sobreposto, a alta demanda e o pouco investimento de tecnologia para resolução deste grande problema nos centros urbanos, nossa solução tem o intuito de auxiliar estes profissionais para diminuir o impacto trazido pelos incêndios e fazer com que o resgate e controle do acidente seja o mais breve possível com o auxílio de moradores dos centros urbanos, isso mesmo, cidadãos terá a praticidade de reportar incêndios ao corpo de bombeiros para que os mesmos cheguem e resolvam o problema, para evitar trote deste recorrências nosso aplicativo exigirá dados pessoais do usuário para que o ator do trote seja punido de acordo com a lei de acordo com o artigo 266 do código penal diz que interromper ou perturbar o serviço telefônico poderá incorrer pena de detenção de um a seis meses ou multa.

¹ SECRETARIA DE SEGURANÇA PÚBLICA DE SÃO PAULO, **Dados Estatísticos do Corpo de Bombeiros do Estado de São Paulo**, disponível em: <<http://www.ssp.sp.gov.br/Estatistica/CorpoBombeiro.aspx>>, acesso em: 10 nov. 2021.

3 FUNDAMENTOS DAS TECNOLOGIAS DE SISTEMAS DISTRIBUÍDOS

Há diversos fundamentos de tecnologia de sistemas distribuídos, dentre elas a monolítica e a de microsserviços, os sistemas monolíticos como os sistema está inteiro em um único bloco, seu desenvolvimento é mais ágil porém são muitas vezes são bastante robustos, único que apresenta um único processo em que diferentes componentes ligados a um único programa de uma única plataforma, mas há algumas desvantagens como a manutenção, dentre elas a manutenibilidade, conforme uma aplicação monolítica cresce, diversas funções são adicionadas a um mesmo código e processo, o que pode acarretar em quedas em cascatas da aplicação como um todo, o código se torna complexo e difícil de dar manutenção, as entregas por sua vez acabam se tornando mais críticas, menos frequentes e até estáveis, a escalabilidade, por se tratar de um único código, todas as funcionalidades precisam ser escalada como um todo, normalmente escalada verticalmente, adicionando mais máquina (processador, memória, ...) para aplicação, ou horizontalmente por modelos de balanceador de carga a linha de código robusta e a falta de flexibilidade de linguagens de programação.

Sendo assim foi decidido em nossa aplicação desenvolver a arquitetura em microsserviços, uma arquitetura de microsserviços, os serviços são refinados e os protocolos são leves, o objetivo é que os serviços possam consigam fornecer suas funcionalidades independentemente de fatores externos.

O acoplamento fraco reduz todos os tipos de dependências e as complexidades em torno delas, já que os desenvolvedores de serviço não precisam se preocupar com os usuários do serviço, eles não forçam suas mudanças para os usuários do serviço.

Portanto, permite que as organizações que desenvolvem software cresçam rapidamente e se tornem grandes, além de usar os serviços prontos para uso com mais facilidade juntamente apresenta mais vantagens como a facilidade e rapidez na atualização e implementação dos serviços, pois os códigos são divididos em pequenas aplicações, aumento da flexibilidade da infraestrutura.

Atualmente, quando temos a necessidade de entrega de um serviço as APIs são essenciais para a entrada de um serviço cada vez mais rico trazendo uma gama de funcionalidades para aplicação satisfazendo nossas necessidades e até mesmo na agilidade de desenvolvimento e de entrega do serviço proposto.

Com a necessidade da utilização de APIs, como todos sabemos o formato de dados mais comum do retorno de dados dessas APIs é o JSON, adotamos este formato também para representar os dados a serem transmitidos, foi acordado utilizar este formato de dado pois JSON

é uma notação que permite estruturar dados em formato de texto para serem utilizados em diferentes tipos de sistemas, pois isto já traz uma valorização no conceito de interação entre sistemas para sistemas distribuídos, pois como vamos utilizar APIs de terceiros e de grande maioria das APIs e outros sistemas que permitem interação com terceiros adotam o formato JSON para o retorno dos dados.

4 PLANO DE DESENVOLVIMENTO DA APLICAÇÃO

Nosso plano de desenvolvimento é fazer back-end feito em Spring Boot com Spring Data usando Java para fazer um servidor de back-end que fornece dados através de APIs.

Sobre Spring Boot é um framework Java Open Source que tem como objetivo facilitar esse processo em aplicações Java, consequentemente, ele traz mais agilidade para o processo de desenvolvimento, uma vez que os desenvolvedores conseguem reduzir o tempo gasto com as configurações iniciais, com o Spring Boot conseguirmos abstrair e facilitar a configuração de servidores, gerenciamento de dependências, configuração de bibliotecas.

O Spring Data é um projeto Spring com proposta de unificar e facilitar acesso a diferentes tecnologias de armazenamento de dados, devido a este fato foi o que nos motivou a escolher o mesmo pois facilitaria o armazenamento de dados de nosso serviço, as principais vantagens do Spring Data além do benefício de conseguir interagir com diferentes tipos de banco de dados de forma padronizada, assim podemos escrever muito menos código repetido, pois as consultas podem ser definidas junto aos métodos de pesquisa e suas documentações, além disso as consultas JPQL são compiladas assim que o contexto Spring é criado e na primeira vez que a consulta é utilizada o que facilita a detecção de erros de sintaxe.

Para o Front End foi decidido utilizar a ferramenta Create React App devido a facilidade de desenvolver em React, React Router Dom para gerenciamento de rotas e Bootstrap, basicamente o front-end vai pegar os dados através de requisições nas APIs do back-end estabelecendo um sistema distribuído, todo esse desenvolvimento será feito através da IDE Visual Studio Code.

4.1 Banco de Dados

Foi utilizado um banco de dados relacional para armazenamento dos dados do sistema. Seu desenho lógico foi feito a partir da definição das entidades e seus relacionamentos.

Manipulações diretas ao banco serão realizadas apenas pelos desenvolvedores, o impacto dos demais usuários será apenas de maneira indireta em funções da aplicação.

Com o propósito de demonstração, a equipe vai utilizar um banco de dados em memória para facilitar a inicialização do mesmo e apresentação de algo, junto com scripts de inicialização dos dados.

Caso fosse ser utilizado em um ambiente de produção, a equipe optaria por utilizar o banco de dados Microsoft SQL Server por ter mais familiaridade com seu uso.

4.2 Business Intelligence (BI)

O BI, é o processo de coleta e transformação de dados brutos em conhecimento para uma área de negócio ou cenário específico para auxiliar os tomadores de decisão a tomarem decisões em seu negócio, em base do conhecimento adquirido nas análises realizadas no BI convertendo esses dados brutos em informação.

O BI possui um conjunto de técnicas, processos e ferramentas para que seja possível realizar a entrega de um projeto, o conhecimento de banco de dados tanto teoricamente quanto tecnicamente é essencial, conhecimento nas regras de negócio em que está aplicando o BI, conhecimento de Business Intelligence como por exemplo, coletar dados de um banco de dados relacional e criar um OLAP (Online Analytical Processing) e de algumas ferramentas específicas de BI como por exemplo o Power BI.

O Power BI, é um serviço de análise de dados da Microsoft, o objetivo é fornecer visualizações gráficas interativas e recursos de BI, nosso fluxo de BI foi feito da seguinte forma, para a extração dos dados foi feita a conexão do Power BI diretamente em nosso banco de dados para realizar todo processo de ETL (Extrair, Transformar e Carregar), o cenário ideal seria a criação de um Data Warehouse em nosso banco de dados para que todos os dados brutos de diversas fontes fossem implementadas lá, o Data Warehouse nada mais é que a base do BI, é o ambiente que podemos guardar informações relativas às atividades de um modelo de negócio em banco de dados consolidados voltados para análise, sendo assim o Data Warehouse ou DW é um OLAP (Online Analytical Processing) cujo seu principal objetivo é realizar análises.

Porém em nosso projeto não foi implementado a criação de um DW pois como a fonte de dados era vinda de apenas uma fonte foi decidido que fosse conectado diretamente ao banco de dados para realização das análises e por questão de custo, pois a criação de um DW geraria mais custo.

Por fim, foi decidido o uso do Power BI pois devido à complexidade de codificação para geração de gráficos visuais para apresentar as análises com os dados gerados, o Power BI tem a possibilidade de integrar os relatórios desenvolvidos em ambientes web e sendo foi decidido o uso do Power BI em nossa aplicação, pois possui uma grande vantagem, é de fácil desenvolvimento e economizaria tempo de desenvolvimento diminuindo a complexidade do código e consequentemente a personalização e ajustes do projeto seria mais ágil comparado ao desenvolvimento da visualização gráfica utilizando o Back-end e o Front-end.

4.3 Sistema Distribuído – Back-end

A aplicação é representada por um sistema distribuído em duas partes, sendo elas o front-end e o back-end.

O back-end é feito em linguagem Java e é composto por uma feita em Spring Boot através de um projeto gerado pelo Spring Initializr com Java na versão 11, Maven e diversas dependências relacionadas ao Spring Boot.

4.3.1 Spring Boot

Spring Boot² é um framework que facilita a geração de aplicativos baseados em Spring de forma fácil, podendo ser configurado com diversas dependências e fazendo autoconfiguração e inclusão de diversos projetos do Spring que podem ser encaixados na aplicação de forma rápida e eficiente.

4.3.2 Spring Framework

O Spring Framework³ possibilita o desenvolvimento e configuração de aplicações Java de forma mais eficiente.

Algumas das tecnologias fornecidas pelo Spring são gerenciamento e instanciação de componente, injeção de dependências, conversão de tipos, ORM, Serialização de JSON, transações, fornecimento de repositórios e controllers, template engines como thymeleaf, validação, anotações de código, eventos entre outras coisas através de seus diversos projetos de fácil acoplamento.

² VMWARE, **Spring.io**, disponível em: <<https://spring.io/>>, acesso em: 5 set. 2021.

³ *Ibid.*

4.3.3 Spring Data JPA

Spring Data JPA⁴ possibilita que o desenvolvedor consiga lidar com diversos tipos de bancos de dados de uma forma padronizada para manipulação e acesso.

Além disso existem outros projetos internos que fornecem capacidade de mapear os objetos e classes do programa em Java com as tabelas, linhas, e colunas de um banco de dados, criando um tipo de Object-Relational Mapping.

Com o Spring Data JPA, você ganha acesso ao Hibernate que seria uma implementação do Java Persistence API que facilita o mapeamento de objetos e classes para serem tratados como entidades com as tabelas de um banco de dados. E com a possibilidade de consultar e alterar os dados do banco de dados através das classes e objetos.

Além disso, o Spring Data JPA permite a criação de Repositórios para realizar as alterações no banco de dados de acordo com as entidades definidas dentro da aplicação.

O Spring Data JPA tem a capacidade de gerar automaticamente as tabelas de um banco de dados baseados nas entidades definidas através de um conceito chamado Data Definition Language e isso é utilizado no projeto para inicializar as tabelas para depois preencher com o conteúdo do script de inicialização de dados.

4.3.4 H2 Database

H2 Database é um banco de dados que fica em memória que foi utilizado nessa aplicação. Devido ao fato de ser em memória, fica fácil de fazer demonstrações uma vez que se tenha um script de inicialização de tabela e um script de inserção de dados.

Esse banco de dados pode ser facilmente trocado dentro do framework Spring por outros como MySQL, MongoDB, Sqlite ou SQL Server caso seja necessário, como por exemplo em um ambiente de produção.

⁴ *Ibid.*

4.3.5 Spring Security

O Spring Security⁵ é um dos projetos do Spring Framework que fornece ferramentas de autenticação, autorização e criptografia para serem utilizados no desenvolvimento de aplicações que precisam criptografar seus dados e senhas, uso de conceitos de autenticação e autorização de usuários baseados em cargos.

No caso do nosso projeto, ele é utilizado para gerar chaves de JSON Web Tokens, criptografar senhas e para autenticação e autorização para acesso a APIs e partes do sistema tanto no front-end quanto no back-end.

Esse framework pode ser utilizado para criptografar os dados armazenados para proteger os dados dos usuários, mas no projeto desenvolvido essa funcionalidade ainda não foi implementada.

4.3.6 Spring Web

Spring Web⁶ é um framework que é utilizado na aplicação para gerar Controllers do tipo REST para fornecer APIs que podem ser usadas pelo front-end para fazer alterações no banco de dados através de Repositórios, podendo também para fazer a autenticação de usuários retornando JSON Web Tokens após o usuário ser autenticado e ao mesmo tempo sendo capaz de validar um JSON Web Token.

Além disso através do Spring Web é possível configurar funcionalidades como Cross-Origin Resource Sharing para permitir o compartilhamento de serviços e recursos com requisições de origens externas possibilitando que o front-end se comunique com as APIs do back-end.

⁵ *Ibid.*

⁶ *Ibid.*

4.3.7 JJWT, Validation API, Lombok

São dependências extras adicionadas ao projeto do Spring Boot para fornecer ferramentas de manipulação de JSON Web Tokens, validação de dados recebidos na API.

Lombok é um plugin que fornece anotações para implementação automática de construtores, getters e setters de classes, além de poder gerar alguns métodos que são usados com frequência.

4.3.8 Maven

Apache Maven é uma ferramenta para gerenciamento de projetos que facilita o gerenciamento de dependências, plugins, scripts de builds, ciclos entre diversas outras coisas de um projeto desenvolvido com ele.

É utilizado dentro do nosso projeto para gerenciar as dependências e plugins, além de cuidar dos scripts relacionados a build e run do projeto.

4.4 Sistema Distribuído – Front-end

4.4.1 HTML, CSS e JavaScript

As páginas web da aplicação que representam sua View foram feitas com HTML para montar a estrutura, CSS para estilos e JavaScript para lógica com auxílio de outros frameworks e bibliotecas como Bootstrap, React e Create React App.

4.4.2 NPM

NPM é um gerenciador de pacotes que é utilizado para gerenciar as dependências de uma aplicação que utilize Node.js.

4.4.3 Node.js

Node.js é um ambiente de execução de JavaScript que é utilizado para rodar código JavaScript fora do Browser utilizando o motor de JavaScript V8.

Ele é utilizado pela nossa aplicação em diversos momentos devido aos módulos adicionados e ao fato da nossa aplicação rodar através do Create React App que utiliza bibliotecas que precisam do Node.js como Webpack.

4.4.4 React

No desenvolvimento da aplicação foi escolhido o uso da Biblioteca de JavaScript React⁷ para gerar componentes que podem ser usados para montar a interface do usuário, além de utilizar o conceito de estado para atualizar o conteúdo dos componentes de acordo com o que o usuário faz dentro da aplicação.

4.4.5 Create React App

Create React App⁸ é um projeto do Facebook que permite criar uma aplicação web de página única, facilitando o desenvolvimento de uma aplicação, nessa aplicação é possível usar React para adicionar componentes e renderizar eles através do uso de React e JSX para criar uma aplicação que consegue modificar seu conteúdo dentro de uma única página.

Esse projeto foi utilizado para criar o molde inicial do projeto na parte do front-end e a partir disso foi desenvolvido o resto.

⁷ **React A JavaScript library for building user interfaces**, React - Facebook, disponível em: <<https://reactjs.org/>>, acesso em: 9 set. 2021.

⁸ **Create React App**, Facebook, disponível em: <<https://create-react-app.dev/>>, acesso em: 5 set. 2021.

REFERÊNCIAS

SECRETARIA DE SEGURANÇA PÚBLICA DE SÃO PAULO. **Dados Estatísticos do Corpo de Bombeiros do Estado de São Paulo**. Disponível em: <<http://www.ssp.sp.gov.br/Estatistica/CorpoBombeiro.aspx>>. Acesso em: 10 nov. 2021.

VMWARE. **Spring.io**. Disponível em: <<https://spring.io/>>. Acesso em: 5 set. 2021.

Create React App. Facebook. Disponível em: <<https://create-react-app.dev/>>. Acesso em: 5 set. 2021.

React A JavaScript library for building user interfaces. React - Facebook. Disponível em: <<https://reactjs.org/>>. Acesso em: 9 set. 2021.