

GLOBAL SOLUTIONS – EnergyX

Projeto: Sistema de Monitoramento e Emissão de Alertas para Operadores de Usinas Nucleares

INTEGRANTES (2TDSPS)

Felipe Amador RM: 553528

Leonardo de Oliveira RM: 554024

Sara Sousa RM: 552656

Sumário

Solução Geral do Projeto	3
Funcionalidades da Solução Geral do Projeto	3
Aplicação Móvel:.....	3
Aplicação .NET (Relatório Diário de Trabalho):.....	3
Deep Learning & Visão Computacional:	3
Projeto DevOps & CloudComputing.....	4
Etapas Principais	4
Descrição das Principais Ferramentas	5
SQL Developer (Banco de Dados Oracle)	5
IntelliJ IDEA (Backend Java Spring Boot)	5
Sobre o Banco de Dados Oracle	6
Confiabilidade e Escalabilidade.....	6
Suporte a Transações Complexas.....	6
Compatibilidade e Integração.....	6
Instalações.....	6
Arquitetura	7
Legenda Técnica da Arquitetura.....	7
Legenda de Fluxo de Informações da Arquitetura	8
LINKS do Contéudo.....	9

Solução Geral do Projeto

O Sistema de Monitoramento e Emissão de Alertas foi desenvolvido para operadores de usinas nucleares, com o objetivo de garantir a segurança e o monitoramento constante das condições operacionais da usina. Este sistema envolve a coleta de dados em tempo real, como temperatura, pressão, radiação, e fluxo de refrigeração, com emissão de alertas para condições adversas e críticas.

Além disso, o projeto conta com um módulo de verificação do uso de EPIs (Equipamentos de Proteção Individual) pelos operadores, utilizando tecnologias de deep learning e visão computacional.

Funcionalidades da Solução Geral do Projeto

Aplicação Móvel:

Login e Cadastro do Operador

Monitoramento de Temperatura, Pressão, Fluxo de Refrigeração e Níveis de Radiação

Status Geral (Normal, Alerta, Crítico)

Integração com o Backend API (Java) e Banco de Dados Oracle

Aplicação .NET (Relatório Diário de Trabalho):

Login e Cadastro do Operador

Tela de Relatório Diário de Trabalho

Integração com Banco de Dados Oracle para armazenar os dados do relatório

Deep Learning & Visão Computacional:

Reconhecimento de EPIs, como máscara, óculos, e macacão, garantindo que os operadores estejam utilizando o equipamento adequado.

Projeto DevOps & CloudComputing

Este projeto tem como objetivo desenvolver a parte backend da nossa solução para monitoramento e gerenciamento de energia renovável, utilizando uma API construída com Java Spring Boot. O backend será hospedado em uma máquina virtual Windows, enquanto o banco de dados Oracle será configurado em uma VM Linux, ambos na Azure. A API será documentada e testada utilizando o Swagger, permitindo realizar operações CRUD (Create, Read, Update e Delete) diretamente pela interface. O foco deste estágio do projeto é garantir a implementação do backend, a persistência dos dados no banco Oracle e a validação dos endpoints da API por meio de testes interativos no Swagger.

Etapas Principais

Configuração de Infraestrutura na Azure: Provisionar e configurar uma máquina virtual (VM) com Windows para hospedar o backend Java Spring Boot e uma VM com Linux para o Oracle Database, para conectividade entre as VMs.

Desenvolvimento do Backend API: Implementar a aplicação backend em Java Spring Boot utilizando o IntelliJ IDEA.

Banco de Dados: Conectar ao Oracle Database para persistência de dados utilizando o SQL Developer.

Integração e Testes de CRUD: Implementar e testar as operações CRUD (Create, Read, Update e Delete | POST, GET, UPDATE e DELETE) através do Swagger usando o navegador Edge, garantindo que todas as funcionalidades de persistência estejam funcionando corretamente com o backend.

Swagger: é uma ferramenta que facilita a documentação e o teste de APIs RESTful. Ele é amplamente utilizado para criar uma interface interativa e amigável onde os desenvolvedores e usuários podem testar os endpoints da API diretamente no navegador, sem a necessidade de escrever código de cliente ou usar ferramentas externas.

Descrição das Principais Ferramentas

SQL Developer (Banco de Dados Oracle)

Motivo da Escolha:

Ferramenta Oficial da Oracle: SQL Developer é a ferramenta oficial da Oracle para gerenciar e desenvolver bancos de dados Oracle, oferecendo suporte completo a todas as funcionalidades do Oracle Database.

Interface Amigável: Possui uma interface gráfica intuitiva para executar scripts SQL, modelar tabelas e gerenciar dados, o que facilita o trabalho de desenvolvimento e manutenção do banco de dados.

Recursos de Modelagem de Dados: Oferece ferramentas para modelagem e visualização de esquemas de banco de dados, facilitando o design da estrutura de dados.

Fonte Confiável: <https://www.oracle.com/br/database/sqldeveloper/technologies/what-is-sql-developer/#:~:text=O%20Oracle%20SQL%20Developer%2C%20uma,cliques%20e%20pressionamentos%20de%20tecla.>

IntelliJ IDEA (Backend Java Spring Boot)

Motivo da Escolha:

Desenvolvimento Avançado: IntelliJ IDEA é amplamente reconhecida como uma das melhores IDEs para desenvolvimento em Java, oferecendo suporte completo a frameworks como Spring Boot e outras bibliotecas de desenvolvimento.

Refatoração e Análise de Código: Proporciona ferramentas de refatoração e análise de código que ajudam a manter a qualidade e a organização do projeto.

Plugins e Integração: Possui uma vasta biblioteca de plugins que permitem integrar ferramentas adicionais, como suporte a Docker, controle de versão (Git), e integração contínua.

Fonte Confiável: <https://www.jetbrains.com/pt-br/idea/>

Sobre o Banco de Dados Oracle

Banco de Dados para a Persistência

Para o projeto, escolhemos o Oracle Database como a solução de banco de dados principal. A decisão foi baseada em várias características e benefícios que tornam o Oracle uma excelente opção para aplicações críticas e de grande porte, como sistemas de energia e monitoramento industrial.

Confiabilidade e Escalabilidade

O Oracle Database é reconhecido mundialmente por sua confiabilidade e capacidade de lidar com grandes volumes de dados. Ele é projetado para suportar operações de alta carga com segurança e estabilidade, sendo ideal para aplicações que necessitam de um desempenho consistente e sem interrupções.

Suporte a Transações Complexas

O Oracle é altamente eficiente no gerenciamento de transações complexas e simultâneas, o que é essencial para um sistema que envolve processos críticos, como a monitoração em tempo real de dados e registros de operações. Ele garante consistência e integridade transacional com seu mecanismo de rollback e controle de concorrência.

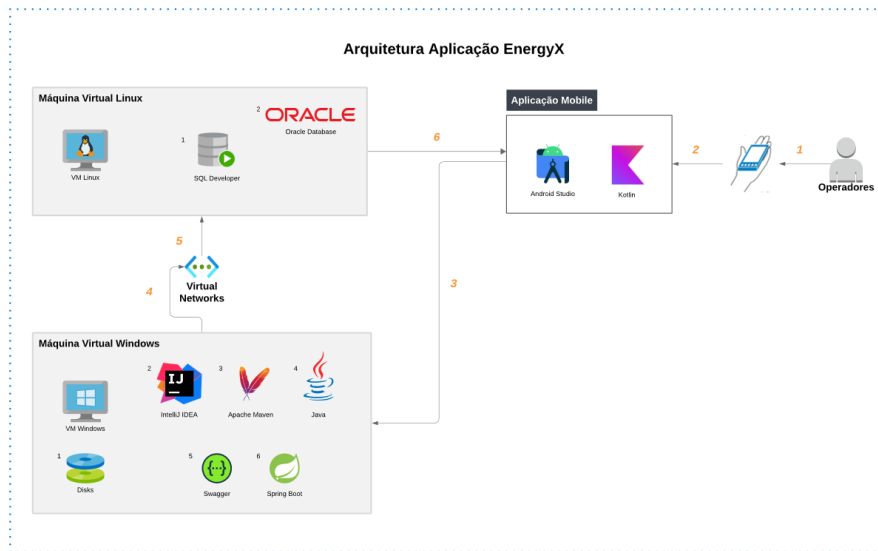
Compatibilidade e Integração

O Oracle Database é amplamente compatível com diversas tecnologias e linguagens de programação, incluindo Java, que será utilizado no backend do projeto. Essa compatibilidade facilita a integração e a comunicação entre a aplicação e o banco de dados, aumentando a produtividade e simplificando o desenvolvimento.

Instalações

Ferramenta	Versão Recomendada	Link de Download	Passos para Instalação
Java	17 (LTS)	Oracle JDK 17	Baixar, configurar JAVA_HOME e PATH no sistema
JDK	17 (LTS)	Já incluído no Java 17	Instalar junto com o Java, verificar com <code>java -version</code>
IntelliJ IDEA	2023	IntelliJ IDEA 2023	Baixar e instalar, configurar JDK 17 no projeto
Maven	3.9.9	Maven 3.9.9	Baixar, extrair e configurar MAVEN_HOME e PATH
Oracle Database	19c (LTS)	Oracle Database 19c	Baixar, instalar e configurar ORACLE_HOME e PATH
SQL Developer	24.3	SQL Developer 24.3	Baixar, extrair, apontar JDK e executar

Arquitetura



Legenda Técnica da Arquitetura

Máquina Virtual Linux

1 - SQL Developer: Ferramenta cliente utilizada para gerenciar e executar consultas SQL no Oracle Database, instalada localmente na máquina virtual Linux. É usada para acessar e administrar o banco de dados de forma gráfica e interativa.

2 - Oracle Database: Sistema de gerenciamento de banco de dados relacional (RDBMS) hospedado nesta máquina virtual, que armazena e gerencia os dados da aplicação.

Virtual Networks

Estrutura de rede configurada na Azure para permitir a comunicação segura entre as máquinas virtuais Linux e Windows, isolando os recursos na nuvem e garantindo conectividade privada entre elas.

Máquina Virtual Windows

1 - Disks: Armazenamento persistente alocado na máquina virtual para manter o sistema operacional, aplicações e arquivos do backend. Foi implementado um disco adicional para suportar aplicações.

2 - IntelliJ IDEA: IDE utilizada para desenvolvimento do backend em Java, oferecendo ferramentas para codificação, testes e deploy.

3 - Apache Maven: Ferramenta de automação para gerenciamento de dependências e build do projeto backend, facilitando a compilação, empacotamento e execução.

4 - Java: Ambiente de desenvolvimento e execução usado pelo backend, incluindo o OpenJDK instalado para rodar aplicações Spring Boot.

5 - Swagger: Ferramenta integrada ao backend para documentação e teste de APIs RESTful. Permite aos desenvolvedores visualizar e interagir com os endpoints expostos.

6 - Spring Boot: Framework do backend, responsável pela lógica de negócio e exposição das APIs REST. Conecta-se ao Oracle Database na máquina Linux para operações de persistência.

Aplicação Mobile

1 - Android Studio: Plataforma de desenvolvimento utilizada para criar aplicativos Android, oferecendo um ambiente integrado e completo com ferramentas avançadas para programação, design, testes e depuração.

2 - Kotlin: Linguagem de programação empregada para integrar o aplicativo ao backend, proporcionando navegação fluida entre as páginas e melhorando a experiência do usuário.

Legenda de Fluxo de Informações da Arquitetura

1 - Acesso pelo Celular: O operador utiliza seu dispositivo móvel para abrir a aplicação EnergyX, que serve como interface para monitoramento e controle.

2 - Login na EnergyX: O operador entra na aplicação para acessar o painel de monitoramento dos reatores, onde pode visualizar dados em tempo real e realizar ações específicas.

3 - Requisição ao Backend: A aplicação mobile faz requisições à API do backend para obter informações sobre o status dos reatores ou enviar comandos de controle.

4 - Conexão Segura via Virtual Network: A comunicação entre a aplicação mobile, o backend e o banco de dados é realizada através de uma Virtual Network (VNet), que garante um canal seguro e isolado entre as máquinas virtuais Windows e Linux na nuvem Azure.

5 - Validação no Banco de Dados: As requisições chegam ao Oracle Database na máquina virtual Linux. Os dados solicitados ou enviados pelo operador são processados e validados para garantir consistência e segurança.

6 - Resposta para a Aplicação Mobile: Após a validação no banco de dados, as informações solicitadas são enviadas de volta ao backend e, em seguida, retornam para a aplicação mobile. O operador recebe as atualizações em tempo real, incluindo funcionalidades ativas, status do sistema e notificações importantes.

LINKS do Contéudo

Repositório no Github

<https://github.com/sousa-sara/energyx-devops-cloud>

Vídeo de Instalação de VMs e conexão com o BD

<https://www.youtube.com/watch?v=ikUDZuAsRPo>