



**MATHEUS GALVÃO FERREIRA**

**DESENVOLVIMENTO FULL-STACK EM SISTEMA DE  
BARRAGENS**

**LAVRAS – MG  
2022**

**MATHEUS GALVÃO FERREIRA**

**DESENVOLVIMENTO FULL-STACK EM SISTEMA DE BARRAGENS**

Relatório Técnico apresentado à Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação, para obtenção do título de Bacharel.

Prof<sup>a</sup>. Dra. Renata Teles Moreira  
Orientadora

**LAVRAS - MG**  
**2022**

# MATHEUS GALVÃO FERREIRA

## DESENVOLVIMENTO FULL-STACK EM SISTEMA DE BARRAGENS

Relatório Técnico apresentado à Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação, para obtenção do título de Bacharel.

Aprovado em 12 de setembro de 2022

Prof<sup>a</sup>. Dra. Renata Teles Moreira  
Prof. Dr. Bruno de Silva Abreu  
Prof. Dr. Rafael Serapilha Durelli

DCC/UFLA  
DCC/UFLA  
DCC/UFLA



Prof<sup>a</sup>. Dra. Renata Teles Moreira  
Orientadora

LAVRAS - MG  
2022

## **AGRADECIMENTOS**

Agradeço primeiramente a minha família, em especial meus pais, João Carlos e Rosângela, por todo amor, apoio e carinho durante toda a minha vida. À minha irmã Juliana e amigos, que de alguma forma puderam tornar toda a minha trajetória mais agradável.

À professora Renata Teles Moreira, a orientação e disposição. À Universidade Federal de Lavras, em especial ao Departamento de Ciência da Computação e seus profissionais.

À TCS, especialmente aos profissionais e ao meu time que me ajudaram no meu desenvolvimento pessoal e profissional.

## RESUMO

O objetivo deste documento é relatar o cotidiano vivenciado durante o período do projeto desenvolvendo um software legado relacionado a estruturas geológicas que envolve o represamento de grandes volumes de água, sedimentos e rejeitos para prevenir catástrofes e riscos ambientais. O desenvolvimento de ferramentas e softwares para gestão e processamento desses dados são imprescindíveis para inspecionar o real estado em que se encontram as barragens em todo o território nacional. O objetivo deste trabalho é explicar as principais técnicas e soluções utilizadas durante o projeto, para manter os padrões de projeto e atender as melhorias e necessidades dos usuários que trabalham com a aplicação. Fazendo análise de sistemas, algoritmos, atendendo as regras de negócio e utilização de metodologias ágeis para manter a constância e qualidade a cada entrega de melhorias do sistema legado. O projeto foi realizado na empresa TCS - *Tata Consultancy Services Limited*, que desenvolve melhorias do sistema chamado GRG - Gestão de Riscos Geotécnicos que pertence a empresa VALE S.A. O sistema tem como objetivo processar os dados e informações inseridas pelos técnicos e retornar as reais condições que se encontra uma estrutura geotécnica, e então, classificá-la de acordo com as normas e regras previstas, que vigoram as leis, os impactos ambientais e sociais que elas causam, portanto, a manutenção e melhorias realizadas no sistema são de extrema importância para a gestão de estruturas geotécnicas, a usabilidade dos usuários e desenvolvedores. O trabalho realizado no projeto teve relação direta com alguns conhecimentos obtidos nas aulas e proporcionou uma grande experiência de mercado, utilizando conceitos de linguagens de programação como Java, SQL, e padrões de projetos.

**Palavras-chave:** Desenvolvimento de software. Padrões de Projeto. Gestão de Riscos Geotécnicos.

## LISTA DE FIGURAS

Figura 1- Requisito e suas atividades .....	12
Figura 2 - Trilha de release para o deploy em Homologação e Produção.....	13
Figura 3 - Versão do pacote implantado nos ambientes de homologação e produção .....	13
Figura 4 - Tela inicial do GRG.....	14
Figura 5 - Tela de seleção de estrutura .....	15
Figura 6 - Tela com informações de uma estrutura geotécnica.....	15
Figura 7 - Diagrama de arquitetura dos ambientes Produção, Homologação e Desenvolvimento .....	16
Figura 8 - Esquema do funcionamento da interface .....	18
Figura 9 - <i>Product Backlog</i> das funcionalidades do GRG .....	24
Figura 10 – Tela Novo Volume VI.....	25
Figura 11 - Tela de pop-up para cadastro de documento .....	26
Figura 12 - Tela da aba do psb com novo volume VI .....	26
Figura 13 - Implementação do Nível de Alerta .....	27
Figura 14 - Exemplificação em código .....	27
Figura 15 - Atualização em banco de dados.....	28
Figura 16 - Implementação dos novos fatores de segurança .....	28
Figura 17 - Script para criar colunas necessárias. ....	29
Figura 18 - Tela de do PSB com Ciência do empreendedor .....	29
Figura 19 - Seleção de grupo de documentos para o relatório .....	29
Figura 20 - Atividade de execução de script .....	30

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>8</b>
<b>2</b>	<b>SOBRE A ORGANIZAÇÃO .....</b>	<b>10</b>
2.1	TATA CONSULTANCY SERVICES (TCS) .....	10
2.2	PROCESSO ORGANIZACIONAL .....	10
2.2.1	REFINAMENTO FUNCIONAL.....	10
2.2.2	REFINAMENTO TÉCNICO .....	11
2.2.3	PLANNING POKER .....	11
2.2.4	DESENVOLVIMENTO .....	11
2.2.5	DEPLOY EM AMBIENTE DE HOMOLOGAÇÃO E PRODUÇÃO.....	12
2.3	PRODUTO .....	13
2.3.1	ARQUITETURA DO SISTEMA .....	16
2.3.2	INTERFACES DO GRG .....	17
<b>3</b>	<b>CONCEITOS E TECNOLOGIAS UTILIZADAS.....</b>	<b>19</b>
3.1	FRAMEWORK SCRUM .....	19
3.2	DEVOPS .....	21
3.2.1	FERRAMENTA AZURE DEVOPS .....	21
3.3	SMARTGWT .....	22
3.4	LINGUAGEM DE PROGRAMAÇÃO JAVA .....	23
3.5	GIT .....	23
<b>4</b>	<b>PRODUTO DESENVOLVIDO.....</b>	<b>24</b>
4.1	ADEQUAÇÃO À NORMA N95 .....	24
4.2	NOVO VOLUME VI – PROCESSO DE GESTÃO DE RISCO .....	25
4.3	NOVA OPÇÃO CAMPO NÍVEL ALERTA EMERGÊNCIA .....	27
4.4	INSERIR NOVOS CAMPOS FATOR DE SEGURANÇA .....	28
4.5	CIÊNCIA POR PARTE DO EMPREENDEDOR .....	29
4.6	SCRIPTS .....	30
<b>5</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>31</b>
	<b>REFERÊNCIAS .....</b>	<b>32</b>

## 1 INTRODUÇÃO

O objetivo deste documento é descrever as atividades realizadas e o produto desenvolvido no decorrer do trabalho como desenvolvedor *full-stack* na empresa TCS, trabalhando num projeto da empresa VALE. A empresa TCS pertence ao Grupo TATA, de origem indiana, e é provedora de serviços e produtos de tecnologia da informação, entre outros serviços, em todo o mundo. A TCS atua com empresas parceiras e tem diversas instalações pelo Brasil.

Dentre as ferramentas utilizadas pela equipe de desenvolvimento da qual o autor participava, podem ser citadas o framework Scrum e o DevOps, as quais eram gerenciadas através da ferramenta Azure DevOps, que corroboram para a cultura ágil que a empresa busca seguir. A linguagem de programação utilizadas nos projetos que ficaram sob responsabilidade da equipe de desenvolvimento foi Java, tanto para o *front-end* quanto o *back-end*; utilizando a biblioteca GWT (*Google Web ToolKit*) para desenvolvimento do *front-end* e o framework *Spring Boot*. Para as manipulações em banco de dados foi aplicada a linguagem SQL (*Standard Query Language*).

A equipe em que o autor estava alocado era composta por três desenvolvedores da TCS e o líder técnico, que era a referência técnica dentro da equipe. Além disso, havia um *Product Owner* e um *Scrum Master*, sendo o *Product Owner* funcionário da empresa cliente VALE e o *Scrum Master* da própria TCS. De forma sucinta, no dia a dia o autor como desenvolvedor participava de todas as reuniões cabíveis à equipe de desenvolvimento, como reuniões diárias conhecidas como ‘Daily’, para repassar as atividades que foram feitas, estão sendo feitas e se possuía algum impedimento para dar continuidade. A *Planning Poker*, que se trata de avaliar as tarefas a serem realizadas durante a sprint e dar uma nota com base na sequência de Fibonacci (L UBIENIECKI, 2021), refinamentos funcionais e técnicos que visam compreender os requisitos propostos pelo *Product Owner* juntamente com a equipe de desenvolvimento. Vale acrescentar que a equipe também realizou contribuições para as entregas ao cliente, prototipação de telas, desenvolvendo tarefas que eram livremente escolhidas entre os desenvolvedores.

Durante o trabalho em questão, foram realizadas diversas tarefas, como a criação de requisitos, que demandou o desenvolvimento *full-stack*, edição de componentes, regras de negócio e PDFs, que necessitavam de atuação voltadas para *front-end* e *back-end* do projeto. Outra atividade tratada neste documento é a correção de bugs, que levou a uma abordagem mais

focada em compreender as regras de negócio e o banco de dados e no atendimento direto ao cliente.

O restante do documento é organizado como a seguir. No Capítulo 2 é apresentada a empresa, seu processo organizacional e o produto desenvolvido no âmbito deste trabalho. No Capítulo 3 é feito um recorte dos temas abordados no documento como um todo, elaborando uma breve definição para cada elemento com base em referências atuais. No Capítulo 4 é abordado o produto desenvolvido durante o trabalho em questão. Por fim, no Capítulo 5 são apresentadas as considerações finais acerca das atividades desempenhadas durante o trabalho.

## 2 SOBRE A ORGANIZAÇÃO

Este Capítulo trata sobre a organização e informações da empresa em questão e sobre o software desenvolvido com suas especificações.

### 2.1 Tata Consultancy Services (TCS)

A *Tata Consultancy Services* (TCS) é uma organização de serviços de TI, consultoria e soluções de negócios que tem feito parceria com muitas das maiores empresas do mundo em suas jornadas de transformação há mais de 50 anos. A TCS oferece uma consultoria cognitiva modelo único de entrega *Location Independent Agile*, reconhecido como referência de excelência em desenvolvimento de software. Parte do grupo Tata, o maior grupo empresarial multinacional da Índia, a TCS tem mais de 606.000 associados em 55 países.

No Brasil, a TCS está localizada na cidade sede de Barueri e conta com mais de 4 mil funcionários. O autor atuou como analista desenvolvedor júnior na área de desenvolvimento de software em um projeto da empresa parceira VALE chamado GRG - Gestão de Riscos Geotécnicos. A equipe do projeto era composta por 5 integrantes, sendo 3 desenvolvedores, um *Scrum Master*, um líder técnico e um *Product Owner*. Durante o projeto, o autor atuou como desenvolvedor Java realizando estórias, resolvendo bugs encontrados em produção, scripts SQL, versionamento de código e realização de implantação nos ambientes. Além disso, era de responsabilidade da equipe participar dos ritos que compõem a metodologia do Scrum, como: *daily*, *review*, retrospectiva e *planning*.

### 2.2 Processo Organizacional

A empresa visa um processo capaz de garantir a qualidade de entrega dos seus produtos mantendo os prazos estabelecidos de acordo entre o cliente. Neste processo, para entrega dos requisitos do projeto é realizado um planejamento utilizando Scrum, uma metodologia de desenvolvimento ágil que está descrita no capítulo 3. As seções seguintes detalham os passos de cada requisito até serem implementados em ambiente de produção.

#### 2.2.1 Refinamento Funcional

No refinamento funcional, o time de desenvolvimento se reúne com o *Product Owner* e os principais envolvidos nas novas funcionalidades (requisitos) que devem ser adicionadas e implementadas no sistema. Neste evento, o *Product Owner* explica sobre as funcionalidades e

quais os critérios necessários para atender à nova funcionalidade. O time técnico atua nesta reunião fazendo perguntas sobre o que as novas funcionalidades irão precisar, desde a criação de novas páginas, botões e questionando sobre como serão as novas regras de negócio para compreender o fluxo até atingir os critérios de aceitação.

### **2.2.2 Refinamento Técnico**

No refinamento técnico, o time de desenvolvimento se reúne para detalhar os requisitos propostos no refinamento funcional, buscando mitigar a melhor maneira de dividir as principais funcionalidades em atividades menores. Essas atividades são escritas de modo que todos consigam compreendê-la e desempenhá-la durante a *sprint*. Neste evento, o requisito e atividades são bem detalhados e são realizados, se necessário, protótipos de tela, investigação no código para procurar comportamentos semelhantes para auxiliar no desenvolvimento, se existe a necessidade de se criar tabela ou coluna em banco de dados e observando se todas as atividades irão atender aos critérios de aceitação.

### **2.2.3 Planning Poker**

Após realizar o refinamento técnico, o time se reúne para pontuar as atividades do requisito para verificar se comporta dentro de uma *sprint* ou se será necessário dividir em outros requisitos. Nesta reunião, cada membro do time atribui uma pontuação a uma determinada tarefa. Então, quando a pontuação é muito divergente entre os integrantes, os integrantes são convidados a argumentar sobre a pontuação que atribuíram à atividade, expondo o motivo que acredita que tal tarefa mereça maior pontuação devido a sua complexidade ou não.

### **2.2.4 Desenvolvimento**

Ao iniciar as atividades de um requisito, é criado um roteiro de testes numa planilha que busca atender todos os critérios esperados pelo requisito, abordando diversas condições e situações a serem testadas pelo time de desenvolvimento e, posteriormente, pelo analista de teste do cliente. Na Figura 1 é apresentado um exemplo do modelo padrão adotado.

Figura 1- Requisito e suas atividades

User Story	Novo Volume VI - Processo de Gestão de Risco (PGRBM).
Task	00 - Criar Roteiro de Testes
Task	01 - Criar nova Aba em Documentação
Task	02 - Criar tela de Popup "Há Processos de Gestão de Riscos"
Task	03 - Adicionar em "Lista de Documentos" tipo de projeto "Processos de Gestão de Riscos"
Task	04 - Criar novo Volume VI: Processos de Gestão de Riscos
Task	05 - Adicionar novo Volume VI no botão "Gerar PDF"
Task	06 - Criar nova seção "Processos de Gestão de Riscos (PGRBM)" no botão Download
Task	07 - Criar PDF para Volume VI
Task	99 - Executar Roteiro de Testes

Fonte: TCS

Após a criação do roteiro de testes é criada a *branch* a partir da *master* para o requisito com o seu ID e em qual sprint ela está, por exemplo:

sprints/sprint\_01/features/123456

Depois desse passo, inicia-se o processo de desenvolvimento do requisito, aplicando os conceitos necessários da linguagem de programação, padrões de projeto etc.

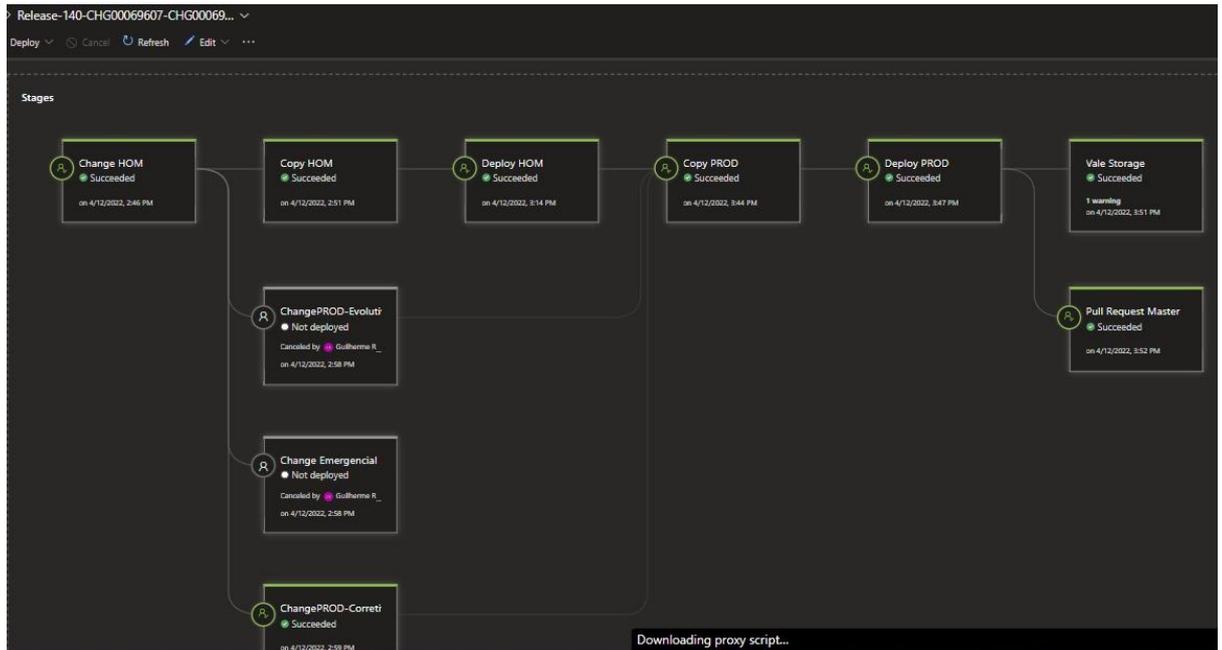
### 2.2.5 Deploy em ambiente de Homologação e Produção

Após a finalização do requisito, ele é informado ao cliente e, então, preparado para ser disponibilizado em ambiente de homologação para testes do cliente. Para isto é criada uma *branch* do pacote com o seguinte padrão:

releases/evolutivas/<ano-mes-dia>

Por meio de um *Pull Request*, são colocados os artefatos necessários para levar o requisito para o ambiente de testes. Para isso é necessário executar o Pipeline no Azure Devops no *build* do projeto para, então, gerar uma *release* que emitirá um número de mudança para Homologação. Posteriormente, esse requisito passa por uma aprovação gerando um número de mudanças para o ambiente de Produção. Esta mudança deverá ser preenchida no *ServiceNow* com aprovação do responsável pelo Projeto e, então, programada para ser implementada em ambiente de produção. Na Figura 2 é ilustrada a trilha da release para o *deploy* em homologação e produção.

Figura 2 - Trilha de release para o deploy em Homologação e Produção



Fonte: TCS

Após a implantação do pacote com a versão, tanto em ambiente de homologação quanto produção, é possível visualizar na tela inicial de login a versão do pacote implantado (Figura 3).

Figura 3 - Versão do pacote implantado nos ambientes de homologação e produção



Fonte: TCS

### 2.3 Produto

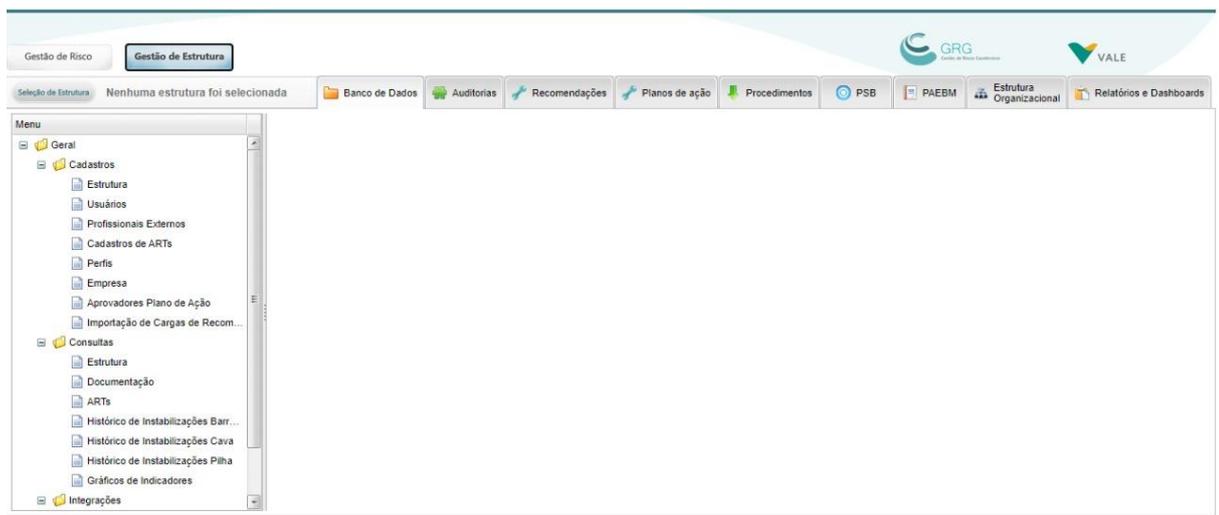
A aplicação GRG - Gestão de Risco Geotécnico tem como objetivo realizar o cálculo e gestão de riscos geotécnicos e apoiar diretamente a tomada de decisões sobre investimentos direcionados para reduzir esses riscos. Através do GRG é possível obter uma visão e análise de riscos geotécnicos, priorizando os planos de ação de acordo com o grau de risco, investimento e outros parâmetros, evitando perdas potenciais (impactos ambientais e perdas financeiras) em projetos da Vale.

O sistema Gestão de Riscos Geotécnicos possui como módulo principal o Gestão de Estruturas. Este módulo é responsável por permitir o cadastro de informações técnicas e

documentação das estruturas de barragens, fornecendo como principal saída o documento de Plano de Segurança de Barragens.

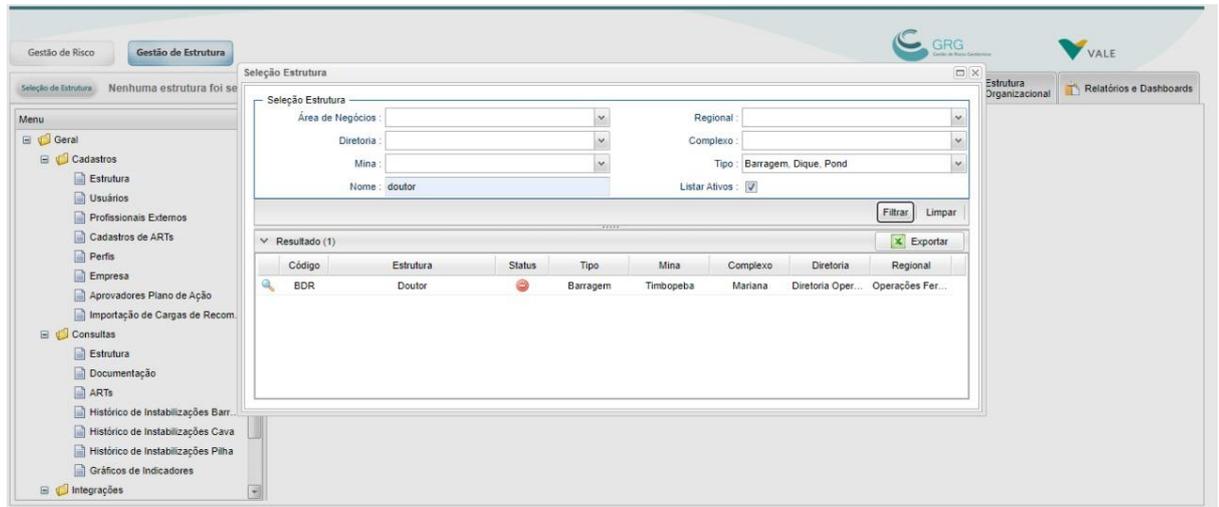
Inúmeras funcionalidades do GRG só são habilitadas mediante a seleção de uma estrutura geotécnica. A seleção da estrutura geotécnica deve ser feita a partir de um botão localizado na porção superior esquerda da tela. A partir do “Botão Seleção de Estrutura” é aberta uma tela para a seleção da estrutura geotécnica que poderá ser realizada a partir de filtros associados a: departamento, complexo, mina, tipo de estrutura geotécnica e/ou nome da estrutura etc. Após o preenchimento dos campos de interesse sobre a estrutura, deve-se selecionar o “Botão Filtrar”, de modo a aparecer a listagem das estruturas que atendem aos critérios previamente estabelecidos. Em seguida, deve-se clicar sobre a lupa localizada na primeira coluna do *grid* para habilitar diversas funcionalidades do sistema. Nas Figuras 4 e 5 estão apresentados exemplos de seleção de estruturas geotécnicas.

Figura 4 - Tela inicial do GRG



Fonte: GRG

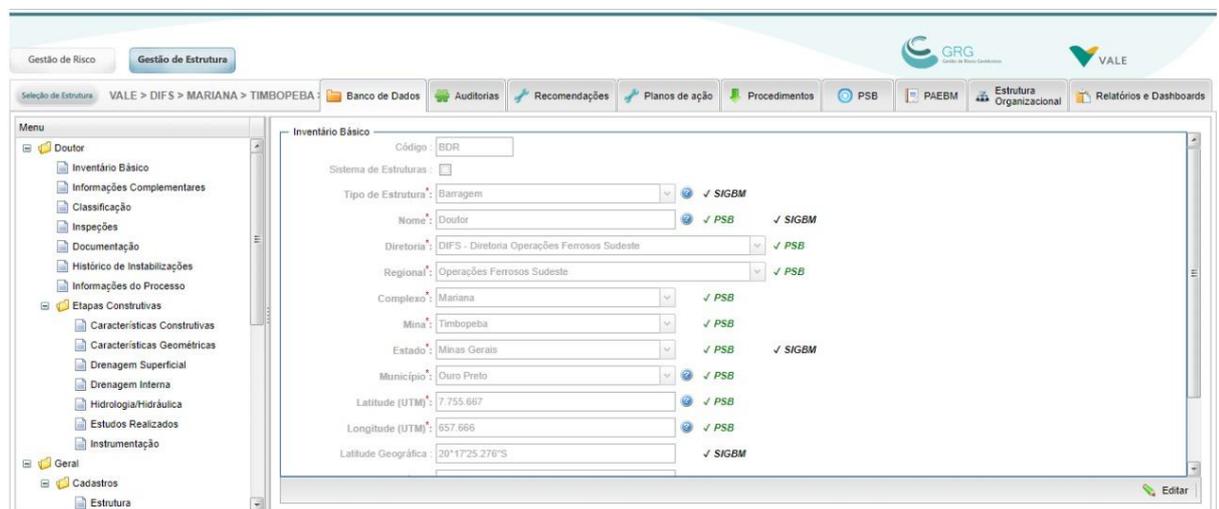
Figura 5 - Tela de seleção de estrutura



Fonte: GRG

O módulo Gestão de Estruturas conta com interfaces que permitem cadastrar ou atualizar as informações técnicas, *upload* e *download* de documentação das estruturas, fornecendo como principal saída o documento de Plano de Segurança de Barragens. Na Figura 6 é mostrada uma tela do GRG com informações de uma estrutura geotécnica.

Figura 6 - Tela com informações de uma estrutura geotécnica



Fonte: GRG

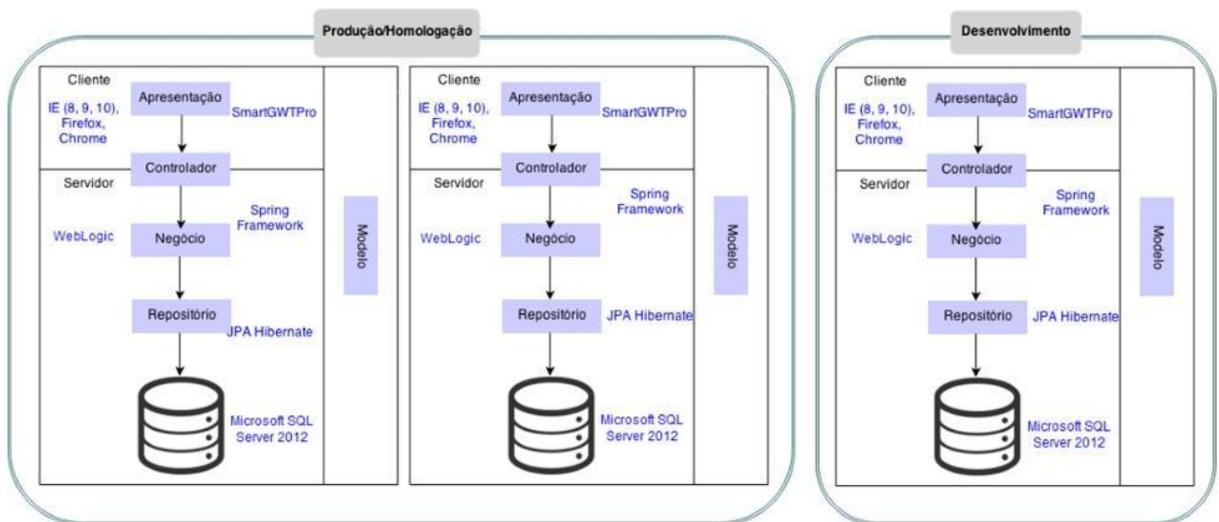
O GRG é uma aplicação *web*, desenvolvida em linguagem Java e frameworks como SmartGWT, Spring, Hibernate e JPA. Além disso, o GRG utiliza servidores Weblogic e banco de dados SQL.

### 2.3.1 Arquitetura do Sistema

A arquitetura do GRG possui servidores Weblogic e banco de dados SQL. Na Figura 7 é mostrado o diagrama de arquitetura do GRG.

Figura 7 - Diagrama de arquitetura dos ambientes Produção, Homologação e

#### Desenvolvimento



Fonte: TCS

Para o desenvolvimento do projeto são utilizados três ambientes distintos, nomeados como: Desenvolvimento (utilizado para testes de sistemas), Homologação (utilizado para homologação de novas funcionalidades) e Produção (ambiente de produção do GRG).

Os servidores utilizam o WebLogic Server Version: 12.1.2.0.3. No entanto, a aplicação não utiliza recursos específicos dessa versão e pode ser homologada em outras versões, desde que compatível com Java 7.

A base de dados utilizada no projeto é o Microsoft SQL Server 2012 R2. Entretanto, não são utilizados recursos específicos dessa versão e a aplicação poderia ser homologada para outra versão.

O sistema é projetado em cinco camadas, sendo elas:

- Apresentação (GRG): Esta camada é responsável por prover a interface gráfica do GRG.
- Controlador (*GRGFacade*): Esta camada funciona como controlador das requisições do cliente para o servidor. Nesta camada, não é desenvolvido qualquer tipo de lógica de negócio, apenas interface entre as chamadas da camada de apresentação para a camada de negócios.

Para o GRG, todas as chamadas do lado cliente ao servidor são feitas de forma assíncrona devido ao paradigma de funcionamento do GWT.

- Negócio (*GRGService*): Esta camada tem como principal responsabilidade a implementação das regras de negócio relacionadas aos objetos de persistência.
- Repositório (*GRGRepository*): A camada Repositório do sistema GRG utiliza o framework hibernate 4.2.6 para implementar a interface com o banco de dados. Nesta camada deve ser garantido que a informação obtida a partir da base de dados seja passada de forma correta para as camadas acima. Da mesma forma, deve ser garantido que informações provenientes das camadas acima estejam formatadas, para serem escritas de volta no banco de dados.

A implementação dos comandos com o banco de dados foca no padrão JPA, preferencialmente não utilizando recursos específicos do Hibernate.

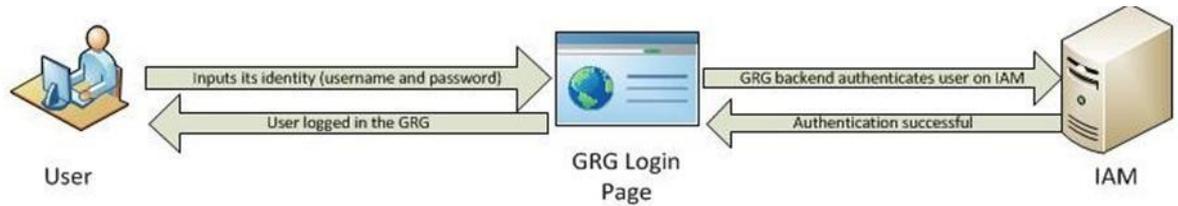
- Modelo (*GRGModel*): Na camada Model estão as classes e enumerações que representam as tabelas do banco de dados. Através dessa camada é possível tratar as informações a serem consultadas e atualizadas no banco de dados como objetos de classes. Os métodos desenvolvidos nas classes desta camada são basicamente acessórios, não apresentando nenhuma lógica elaborada. Quando existem operações diferentes, tais funções se restringem apenas a comandos simples como obtenção de primeiro ou último elemento de uma lista. As classes desta camada são anotadas com as informações necessárias para que o JPA entenda sua ligação com o banco de dados.

A modelagem foi feita seguindo as boas práticas e padrão difundido na literatura, sempre perseguindo a máxima objetividade e limpeza das informações persistidas, e o mínimo de redundância de dados.

### 2.3.2 Interfaces do GRG

O GRG usa o IAM (*Identity Access Management*) como interface para realizar autenticação de usuários. O IAM é framework de autenticação de acessos, que verifica se um usuário e senha possui autorização para acessar o sistema, como pode ser observado na Figura 8.

Figura 8 - Esquema do funcionamento da interface



Fonte: TCS

Onde:

1. Usuário informa sua chave e senha;
2. O GRG realiza a validação do usuário na base de dados do GRG;
3. O GRG faz a autenticação do usuário no IAM;
4. O IAM retorna a resposta da autenticação;
5. A página principal do GRG é apresentada ao usuário e as telas associadas ao perfil do usuário ficam disponíveis para acesso.

### 3 CONCEITOS E TECNOLOGIAS UTILIZADAS

Este Capítulo tem como objetivo apresentar algumas das tecnologias utilizadas no desenvolvimento da aplicação.

#### 3.1 Framework Scrum

Com o mercado de Tecnologia da Informação mais competitivo, onde a entrega rápida dos sistemas com qualidade e flexibilidade se torna a prioridade, surge o novo conceito de metodologia ágil na gestão dos projetos, o Scrum (MARTINS, 2007).

Originalmente, o Scrum foi desenvolvido para ser implementado em empresas de fabricação de automóveis. Porém, pode ser utilizado por qualquer empresa que queira implementar processos de gerenciamento de projetos, tais como agências de publicidade, projetos de arquitetura, bancos e outros (SILVA, 2010).

De acordo com Schwaber e Sutherland (2020) o Scrum é composto por eventos, papéis e artefatos. Os eventos do Scrum são uma oportunidade para inspecionar e alterar os artefatos do Scrum. Esses eventos são necessários para a transparência, se os eventos não forem conduzidos conforme especificado, a oportunidade de inspecionar e alterar os artefatos do Scrum é perdida. Todos os eventos são realizados no mesmo horário e local, para eliminar a complexidade extra.

As *Sprints* são eventos de duração fixa com ciclo de duas semanas, por exemplo. Uma nova *Sprint* começa imediatamente após a conclusão do *Sprint* anterior. Todos os demais eventos necessários para alcançar o sucesso da *sprint*, incluindo a *Sprint Planning*, as *Daily Scrums*, a *Sprint Review* e a *Sprint Retrospective*, acontecem dentro das *Sprints*.

A *Sprint Planning* é o evento em que será organizado todo o planejamento para a *sprint* pelo *Scrum Team*. O *Product Owner* garante que os participantes estão preparados para discutir os itens mais importantes do *Product Backlog* e alinhar as expectativas da *Sprint*. O *Scrum Team* pode também convidar outras pessoas a participar no *Sprint Planning* para darem conselhos.

As *Daily Sprint* são reuniões diárias com o intuito de inspecionar o desenvolvimento do time de desenvolvimento, alinhando as principais atividades realizadas e as dificuldades no decorrer da *sprint* para garantir as expectativas se a *sprint* será completamente finalizada. Na *daily* o time de desenvolvimento basicamente responde três perguntas: “O que está fazendo atualmente?”, “O que foi feito anteriormente?” e “se possui algum impeditivo?”. Com este evento, os envolvidos podem planejar o andamento da *Sprint* e melhorar as tomadas de decisão.

Na *Sprint Review* são apresentados os resultados produzidos pelo *Scrum Team*, demonstrando as principais funcionalidades desenvolvidas e o que que foi agregado para o ambiente.

E por fim, o último evento é a *Sprint Retrospective* com o objetivo de aumentar a qualidade e os esforços realizados na *sprint* e garantir a melhor continuação nas entregas nas próximas *sprints*. Neste evento o *Scrum Team* identifica os pontos bem-sucedidos da *sprint* e os pontos que podem ser melhorados ou evitados, sendo possível acrescentar ao *Sprint Backlog* para o Sprint seguinte.

O Scrum é composto por 3 papéis fundamentais que irão compor o *Scrum Team*. São eles (Schwaber e Sutherland, 2020):

- *Scrum master*: é responsável pela implementação do Scrum e a garantir que todos os envolvidos possam compreender a teoria e a prática do Scrum, tanto dentro da *Scrum Team* como dentro da organização. Ele é responsável por treinar a equipe e garantir a autogestão, remover os impedimentos encontrados e a importância da realização de todos os eventos.
- *Product Owner*: é responsável por maximizar o valor do produto resultante do trabalho da *Scrum Team*. A forma como isto é feito pode variar muito entre organizações, *Scrum Teams* e indivíduos
- *Team*: são os que compõem o time de desenvolvimento que irá atuar no desenvolvimento das atividades a serem realizadas durante a *sprint*.

Para apresentar as vantagens do uso do framework é possível citar (Pereira, 2007):

- Cria um ambiente propício para definição de mudanças de requisitos e inovação durante o ciclo de desenvolvimento do produto, assim como mais colaborativo e produtivo entre desenvolvedores e cliente, resultando em entregas mais rápidas de produto, melhor adaptados à realidade do cliente e com a qualidade esperada.
- Facilita o gerenciamento do projeto, uma vez que existem maior integração e comprometimento da equipe do projeto, que conseqüentemente se sente mais motivada: a moral da equipe é elevada.
- Reforça o planejamento constante do projeto, o que minimiza os riscos, considerando que o planejamento é mais importante do que o plano. Não se deve parar de planejar até que se tenha encontrado a satisfação do cliente com a entrega do produto.
- Valoriza a satisfação do cliente e autonomia do *Scrum Team*.

## 3.2 DevOps

“DevOps é a união de pessoas, processos, e produtos para permitir a entrega contínua de valor aos usuários finais. Está além da automação ou infraestrutura como código” (Play Fresco, 2022). A cultura do DevOps promove uma melhor comunicação e colaboração entre equipes, quebrando assim os silos formados entre as equipes (VALENTE, 2020). Os objetivos do DevOps são:

- Desenvolver softwares de alta qualidade;
- Implementar em ciclos frequentes;
- Reduzir o tempo para passar da ideia para a implementação.

Quatro princípios por trás do manifesto ágil que se aplicam aos DevOps:

- A prioridade mais alta é satisfazer o cliente por meio de entrega antecipada e contínua de software;
- Entrega de software em funcionamento com frequência;
- Negócios e os desenvolvedores devem trabalhar juntos diariamente durante todo o projeto;
- Software funcionando é a principal medida de progresso.

### 3.2.1 Ferramenta Azure DevOps

A ferramenta Azure DevOps pode ser utilizada para implementar os conceitos de DevOps. Esta ferramenta surgiu por volta de 2008, para solucionar conflitos entre as áreas de Desenvolvimento e Operações. A plataforma de computação na nuvem da Microsoft permite executar qualquer tipo de aplicação, e este conceito é amplamente adotado atualmente, pois é uma ferramenta que proporciona tudo que um projeto ou uma empresa de software precisa (Play Fresco, 2022).

O Azure DevOps controla toda gestão de ciclo de vida do desenvolvimento de software. Nela existem alguns módulos que auxiliam no acompanhamento de desenvolvimento fazendo toda a gestão de uma aplicação, onde pode-se planejar e distribuir as atividades de maneira eficiente e inteligente entre as equipes, com ferramentas simples e confiáveis para entrega contínua. Os principais serviços do *Azure DevOps* são: *Azure Boards*, *Azure Pipelines*, *Azure Repos*, *Azure Test Plans* e *Azure Artifacts* (SATO, 2013).

O *Azure Board* é responsável pelo gerenciamento de backlogs e demandas e fornece às equipes processos do tipo Agile, Scrum e Kanban, exibições de calendário, painéis configuráveis e relatórios integrados e personalizáveis, fazendo que todos os envolvidos

acompanhem com rapidez, clareza no desenvolvimento do projeto permitindo atualizar e filtrar histórias de usuário, bugs, recursos e épicas.

O *Azure Pipelines* fornece a Integração Contínua e a Entrega Contínua (CI/CD), também conhecida como pipelines CICD. Este módulo permite automatizar o processo de entrega de software colocando em produção de forma contínua e sem perder a qualidade da entrega. Esta automatização é uma sequência de etapas que precisam ser executadas para colocar uma aplicação de software em produção, como executar testes automatizados, procedimentos de *build*, implementação em ambientes de testes e de produção.

O *Azure Repos* é um conjunto de ferramentas de controle de versão utilizado para gerenciar o código. O *Azure Repos* mantém a rastreabilidade das alterações realizadas no código permitindo monitorar, coordenar as alterações da equipe e ter um histórico de desenvolvimento.

O *Azure Test Plans* é um conjunto de ferramentas avançadas utilizado para testar o aplicativo com testes manuais/exploratórios e testes contínuos, acompanhando a qualidade do código em todo seu processo de ciclo de vida.

O *Azure Artifacts* é uma plataforma de pacotes *open source* para comunidade de desenvolvedores restrito à empresa, sem acesso ao público, ou seja, o *Azure Artifacts* é um serviço que permite que as equipes gerenciem suas dependências utilizadas no desenvolvimento do software.

### 3.3 SmartGWT

O *SmartGWT* é um framework baseado no GWT (*Google Web Toolkit*) que possui uma gama de componentes disponíveis na sua biblioteca de *widgets* para aplicações *UI*. Tais componentes, em alguns casos, como no dos grids e gráficos, apresentam funções interativas para que o usuário consiga alterar, por exemplo, o tipo de gráfico a ser exibido ou a ordem das colunas de um grid dinamicamente no lado cliente. Os gráficos desenhados através da biblioteca são feitos utilizando os padrões HTML5 e SVG, tornando-os compatíveis com os navegadores mais utilizados (Smartgwt, 2022).

O *SmartGWT* encapsula HTML5 e Ajax, assim, o código escrito na camada de apresentação é feito na linguagem Java e este é traduzido de Java para *JavaScript* através do framework. Assim, a aplicação tem a característica de apresentar respostas rápidas ao usuário já que o código *JavaScript* roda apenas no lado cliente.

A IDE para desenvolvimento utilizando o *SmartGWT* é de fácil aprendizado e é facilmente integrada a plataformas de desenvolvimento Java, como no caso desse projeto no qual foi utilizado o *Eclipse Java EE IDE for Web Developers*, versão Juno Service Release 2.

A versão do *SmartGWT* utilizada pelo GRG é a v9.1p\_2014-07-03/Pro Deployment (2014-07-03) do *SmartGWT*.

### 3.4 Linguagem de programação Java

Java é uma linguagem de programação orientada a objetos, cuja principal característica é que programas criados nela são compilados em um código intermediário entre o código-fonte e o código de máquina, denominado *Bytecode*. Esse *Bytecode* é, então, executado por uma máquina virtual Java, permitindo aos desenvolvedores criarem aplicações uma única vez e depois executá-las em qualquer uma das plataformas suportadas pela tecnologia.

Por ser uma linguagem multiplataforma, ganhou espaço em diversos dispositivos, o que exige de muitas empresas de software que disponibilizam programas que são feitos nessa linguagem, tornando a instalação da máquina virtual Java muito importante na maioria dos sistemas operacionais atuais.

A máquina virtual Java também pode executar programas dentro dos navegadores, através de um plugin. Essa característica, aliás, tem sido um dos fatores que mais contribuíram para a adoção do Java atualmente (BRITO, 2014).

No sistema desenvolvido, a linguagem Java foi utilizada tanto no *back-end* para modelar entidades específicas quanto no *front-end* para criar componentes com interação do usuário.

### 3.5 Git

Git é um sistema de controle de versão distribuído de software-livre que foi desenvolvido com o intuito de evitar erros encontrados e contribuir com um ambiente colaborativo entre equipes na disponibilização do código-fonte (Git, 2022). Um sistema de controle de versão garante o registro de todas as modificações realizadas no código-fonte. Desta forma, se um erro for encontrado, os desenvolvedores podem voltar numa linha do tempo imaginária em que o erro começou a cometer e então aplicar uma nova solução para o problema.

## 4 PRODUTO DESENVOLVIDO

Este capítulo tem como objetivo descrever algumas funcionalidades desenvolvidas para o produto GRG. Vale ressaltar que tais funcionalidades foram desenvolvidas no sistema sob responsabilidade do time de desenvolvimento.

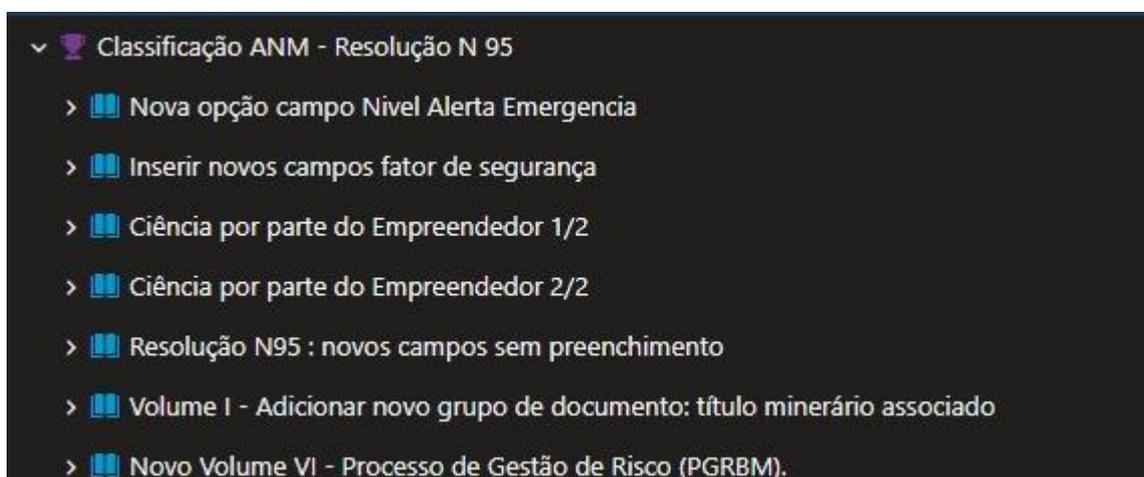
### 4.1 Adequação à Norma N95

Para se adequar a Resolução N° 95, de 7 de fevereiro de 2002, que consolida os atos normativos que dispõem sobre segurança de barragens de mineração (BRASIL, 2022). Às novas leis, normas e prioridades da organização, muitas mudanças nas regras de negócio da aplicação do GRG precisaram ser realizadas ou desenvolvidas novas funcionalidades para se adequar às exigências governamentais e, ainda sim, garantir o bom funcionamento no fluxo de utilização para os usuários. Nesta fase, as antigas leis e regras conhecidas com Classificação ANM deveriam ser adequadas às novas normas, chamada N95, onde foi necessário implementar novas funcionalidades para o produto, como por exemplo:

- Adicionar novos campos que atendam a nova lei;
- Criar volumes de documentos que englobam tipos específicos de projetos relacionados com o tipo da barragem;
- Vincular a assinaturas do responsável técnico a uma ou mais determinadas estruturas geotécnicas;
- Remoção de cálculos automáticos realizados pelo sistema.

Na Figura 9 é mostrado o *Product Backlog* com exemplos de funcionalidades que foram planejadas para o produto GRG.

Figura 9 - *Product Backlog* das funcionalidades do GRG



Fonte: TCS

## 4.2 Novo Volume VI – Processo de Gestão de Risco

A funcionalidade “Novo Volume VI - Processo de Gestão de Risco”, trata-se de inserir um novo grupo de documentos que irá carregar para dentro do sistema arquivos no formato de PDF e que deverão estar organizados no PSB para a obtenção de relatórios de acordo com o seu determinado volume.

Portanto, para atender às novas exigências foi necessário criar um volume para tipos de documentos chamados de processos de Gestão de Risco. Para esta funcionalidade foi necessária a criação de uma nova aba de documentação e nela inserir a pergunta “há processo de gestão de riscos?” com botão para inserir um arquivo com as principais informações necessárias. Na Figura 10 é possível visualizar como é o modelo desejado.

Figura 10 – Tela Novo Volume VI



Fonte: GRG

No botão ‘arquivos’ existem diversos campos a serem preenchidos entre obrigatórios e não obrigatórios, além de ser possível carregar um ou mais arquivos. Na Figura 11 é mostrado o *pop-up* de como deve ser a tela para cadastrar o documento e fazer o seu *upload*.

Figura 11 - Tela de pop-up para cadastro de documento

Há Processos de Gestão de Risco?

Dados do Projeto

Tipo do Projeto:  Empresa Projetista:

Título do Projeto:

Dados do Documento

Tipo Documento:  Ciclo de Vida:

Empresa Resp.:  Número VALE:

Revisão VALE:  Volume:  Número Contratada:

Revisão Contrat.:  Data Emissão:

Título:

Palavras-Chave:

Local Físico de Armaz.:  EL. da Crista de Projeto (m):

Processo Ambiental:  Núm. Protocolo:

Data Vencimento:

Selecionar Estruturas

Nome do arquivo:

Documento:  No file chosen

Documentos Cadastrados							
	Título	Tipo Documento	Número VALE	Revisão VALE	Data	Data Cadastro	Palavras-Chave
	Fechamento da Galeria de Drena...	Desenho	1850CM-X-00...	0	17/05/2013	17/05/2013	
	Fechamento da Galeria de Drena...	Desenho	1850CM-X-00...	0	17/05/2013	17/05/2013	

Fonte: GRG

Este novo grupo de documentos deverá aparecer também na aba do PSB, que é área onde pode baixar, visualizar documentos, relatórios e histórico pertencentes a uma estrutura geotécnica.

Figura 12 - Tela da aba do psb com novo volume VI

Matheus Galvão Ferreira\_CONTR

Gestão de Risco | Gestão de Estrutura

Seleção de Estrutura: VALE > DICS > SOSSEGO > SEQUEIRINH

Banco de Dados | Auditorias | Recomendações | Planos de ação | Procedimentos | PSB | PAEBM | Estrutura Organizacional | Relatórios e Dashboards

Menu

- Sistema Sossego
  - Volume I
    - Parte 1: Informações Gerais
    - Parte 2: Documentação Técnica
  - Volume II: Planos e Procedimentos
  - Volume III: Registros e Controles
  - Volume IV
  - Volume V: Plano de Ação Emergencial
  - Volume VI: Processos de Gestão de ...
  - Controle De Revisões
  - Protocolo PSB

Gerar PDF | Download | Incluir Informações de Estruturas Internas

Processos de Gestão de Risco | Controle de Revisões

Resultado (0)

Tipo do Projeto	Tipo de Documento	Título do Documento	Título do Projeto	Data	Número VALE	Número Contratada
Nenhum registro para exibir						

Fonte: GRG

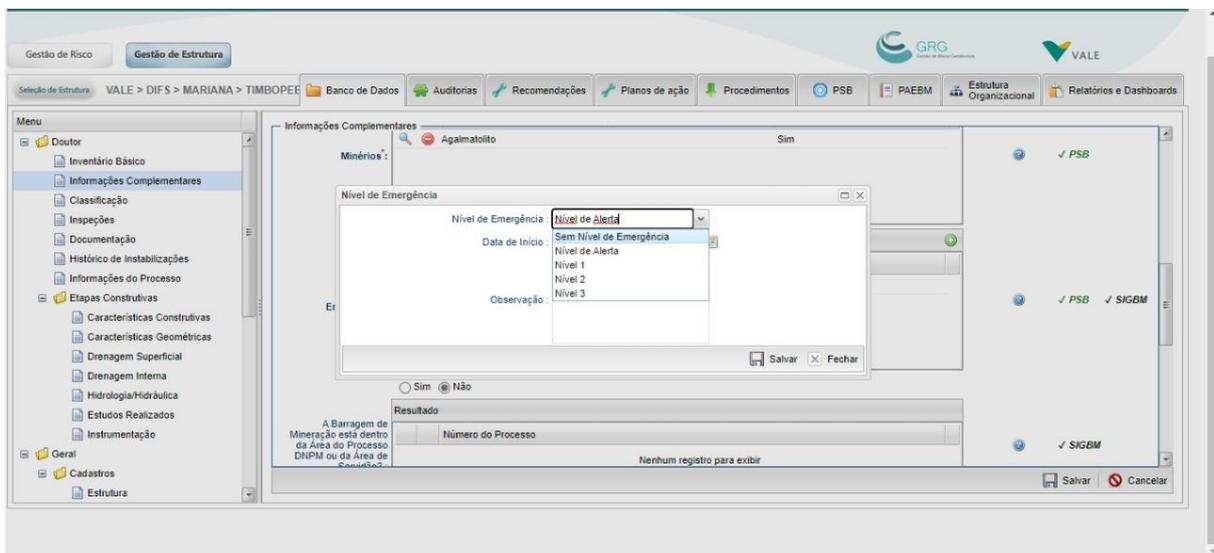
### 4.3 Nova opção campo Nível Alerta Emergência

Nesta nova funcionalidade, aos pesquisar por uma determinada estrutura geotécnica, no menu de “informações complementares” haver uma nova opção chamada “Nível de Alerta” entre as já existentes:

1. Sem Nível de Emergência
2. Nível 1
3. Nível 2
4. Nível 3

A nova opção deverá estar entre o “Sem Nível de Emergência” e “nível 1”, como demonstrado na Figura 13 com a implementação realizada.

Figura 13 - Implementação do Nível de Alerta



Fonte: GRG

Em Código e em banco de dados foi necessário atualizar o valor para adequar a nova exigência. Na Figura 14 é mostrado um trecho de código que exemplifica como fica em código e a atualização necessária em banco de dados para validação (Figura 15).

Figura 14 - Exemplificação em código

```

30 import java.util.ArrayList;
5
6 public class NivelEmergencia extends Enumeration
7 {
8     /**
9      *
10     */
11     private static final long serialVersionUID = 1L;
12     public static final NivelEmergencia SEM_NIVEL_EMERGENCIA = new NivelEmergencia(0, "Sem Nível de Emergência");
13     public static final NivelEmergencia NIVEL_ALERTA = new NivelEmergencia(1, "Nível de Alerta");
14     public static final NivelEmergencia NIVEL_1 = new NivelEmergencia(2, "Nível 1");
15     public static final NivelEmergencia NIVEL_2 = new NivelEmergencia(3, "Nível 2");
16     public static final NivelEmergencia NIVEL_3 = new NivelEmergencia(4, "Nível 3");
17

```

Fonte: GRG

Figura 15 - Atualização em banco de dados

```

updateNivelAlerta.sql - not connected
1 UPDATE nivelAlerta
2 SET nivelEmergencia = 0
3 WHERE nivelEmergencia = 1;
4
5 SELECT * FROM nivelAlerta na
6 WHERE nivelEmergencia = 1;

```

Fonte: TCS

#### 4.4 Inserir novos campos fator de segurança

Esta nova funcionalidade insere três novos campos relacionados a fator de segurança de barragens, são eles:

1. Fator de Segurança Drenado;
2. Fator de Segurança Não Drenado de Pico;
3. Fator de Segurança Sísmico.

Estes novos campos estão ao pesquisar por uma determinada estrutura geotécnica, dentro do menu de “informações complementares”, como mostrado na Figura 16.

Figura 16 - Implementação dos novos fatores de segurança

Fonte: GRG

No banco de dados foi necessário a inserção de novas colunas na tabela de Barragem com os três novos valores, como está exemplificado na Figura 17.

Figura 17 - Script para criar colunas necessárias.

```

criar_colunas.sql - not connected*
1 ALTER TABLE GRG.dbo.barragem ALTER COLUMN fatorSegurancaDrenado numeric(12,2) NULL;
2
3 ALTER TABLE GRG.dbo.barragem ADD fatorSegurancaSismico numeric(12,2) NULL;
4
5 EXEC GRG.sys.sp_rename N'GRG.dbo.barragem.fatorSegurancaNaoDrenado', N'fatorSegurancaNaoDrenadoPico', 'COLUMN';
6 ALTER TABLE GRG.dbo.barragem ALTER COLUMN fatorSegurancaNaoDrenadoPico numeric(12,2) NULL;
7

```

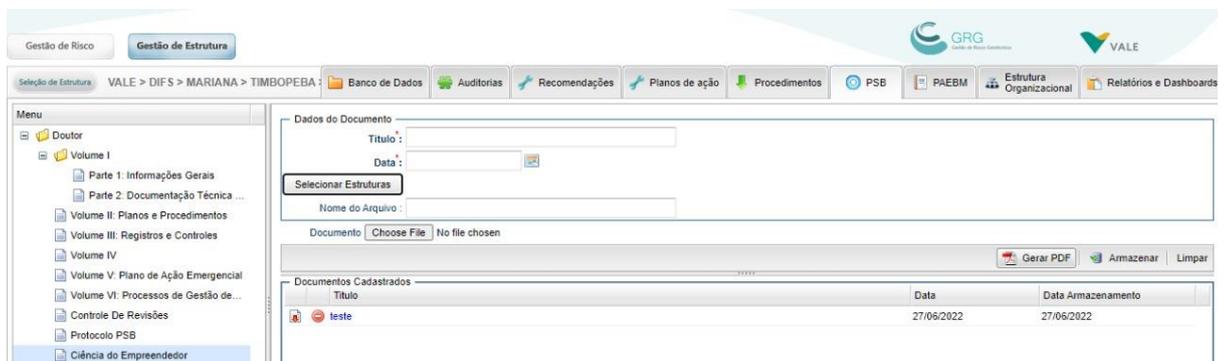
Fonte: TCS

#### 4.5 Ciência por parte do empreendedor

Nesta funcionalidade cria no menu no PSB um novo grupo para inserção de documentos que contêm a assinatura do empreendedor responsável por uma ou mais determinadas estruturas geotécnicas. Esta funcionalidade insere uma tela com o botão para selecionar diversas estruturas entre barragem e diques somente.

Ao selecionar as estruturas, o usuário pode anexar um documento somente no formato PDF que contenha a assinatura do empreendedor. Ao clicar em “GERAR PDF” deve-se aparecer quais grupos de documentos deseja gerar o PDF com o documento de “assinatura do empreendedor” anexado no final do relatório (Figura 18 e Figura 19).

Figura 18 - Tela de do PSB com Ciência do empreendedor



Fonte: GRG

Figura 19 - Seleção de grupo de documentos para o relatório



Fonte: GRG

## 4.6 Scripts

As atividades de scripts consistem em fazer alguma atualização em banco de dados na qual o usuário não possui permissão para alterar ou para corrigir eventuais erros que possam acontecer (Figura 20).

Figura 20 - Atividade de execução de script



584059	Alterar nome de Complexo	Closed
585804	Criar Script	Closed
585806	Validação em QA	Closed
585808	Execução em PROD	Closed

Fonte: TCS

As atividades de scripts geralmente são inseridas no decorrer da *sprint*, lembrando que o time de desenvolvimento possui pontuação residual para atender determinadas atividades de emergência e bugs. Os scripts possuem o mesmo processo de implantação de novas funcionalidades.

## 5 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento deste sistema, o autor pôde desenvolver inúmeras habilidades técnicas como melhor e aplicar em situações reais de trabalhos, conceitos de orientação a objetos, padrões de projetos, linguagens de programação, que apoiaram para adquirir autonomia nas tomadas de decisões de como as novas funcionalidades deveriam estar sendo aplicadas para o projeto. Além disso, as habilidades interpessoais como comunicação ativa frente às reuniões de refinamento técnico e funcional que contribuíram para encontrar soluções viáveis para o projeto e a negociar atividades durante o processo de desenvolvimento nas sprints. Estas habilidades juntas contribuíram para o seu desenvolvimento pessoal e, principalmente, profissional, além de contribuir como a experiência para o currículo.

Por se tratar de um sistema legado e algumas de suas tecnologias usadas como o *SmartGWT*, não possuir muita documentação e exemplos, tornou o desenvolvimento do sistema desafiador e o processo de pesquisar, testar e entender o comportamento da aplicação, muito mais delongada que a própria implementação em muito dos casos. Além de se tratar de um sistema que envolve leis estaduais e federais que sofrem alterações que impactam muitas vezes nos cálculos e tomada de decisão do sistema tornando o trabalho de verificação altamente minucioso.

Este documento apresentou as estratégias e passos necessários para o desenvolvimento das funcionalidades do sistema GRG, buscando atender as reais necessidades do cliente e usuários da aplicação. Após seu desenvolvimento e implantação em ambiente de produção, foi possível constatar que as novas funcionalidades cumpriram com o objetivo esperado, agregando valor, de modo a atender as novas leis que incidem sobre o sistema de gestão de estruturas geotécnicas.

## REFERÊNCIAS

Beck, K. et al. Manifesto para Desenvolvimento Ágil de Software. 2001. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>.

Ken Schwaber & Jeff Sutherland. O Guia do Scrum. 2020. Disponível em: <<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf>>.

Lubieniecki, A. O que é e como funciona o Planning Poker? 2021. Disponível em: <<https://www.luby.com.br/o-que-e-e-como-funciona-o-planning-poker/>>.

Brito, E. **Java: Entenda para que serve o software e os problemas da sua ausência.** 2014. Disponível em: <<https://www.techtudo.com.br/noticias/2014/11/java-entenda-para-que-serve-o-software-e-os-problemas-da-sua-ausencia.ghtml>>.

Guedes, M. Para que serve um framework? 2019. Disponível em: <<https://www.treinaweb.com.br/blog/para-que-serve-um-framework/>>.

**Play Fresco. Cultura DevOps. 2022. Disponível em:** <<https://play.fresco.me/course/997>> Acessado em 23 ago. 2022.

Git. 2022. Disponível em: <https://git-scm.com/>.

Smartgwt. 2022. Disponível em: <<http://www.smartclient.com/smartgwt/showcase/#main>>.

VALENTE, Marcos Tulio. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade.** 2020. Editora: Independente, capítulo 10. Disponível em: <<https://engsoftmoderna.info/cap10.html>>.

BRASIL. **RESOLUÇÃO Nº 95, DE 7 DE FEVEREIRO DE 2022**, Diário oficial da União. Disponível em: <<https://in.gov.br/web/dou/-/resolucao-n-95-de-7-de-fevereiro-de-2022-380760962>>.

SATO, Danilo. **DevOps na Prática. Entrega de Software Confiável e Automatizada.** Editora: Casa do Código. 2013.

MARTINS, José Carlos Cordeiro. **Técnicas para gerenciamento de projetos de software**. Rio de Janeiro, Editora Brasport, 1ª Edição, 2007.

Pereira, P.; Torreão, P.; Maçal, A. S. (2007) Entendendo Scrum para Gerenciar Projetos de Forma Ágil. In.: Mundo PM.

Silva, M. A. C.; Roriz Filho, H.; Silva, H. F. N. (2010) Análise do BA durante o Processo Scrum. XVII Simpósio de Engenharia de Produção. Bauru – SP.

LUBIENIECKI, A. O que é e como funciona o Planning Poker? 2021. Disponível em: <<https://www.luby.com.br/o-que-e-e-como-funciona-o-planning-poker/>>.