Desafio Parte 2 - Mandacaru Broker

PROFESSOR: Ricardo Vilela

Link do projeto

Marcos Vinicius Andrade de Sousa

Wendel Manfrini de Andrade Mendes

Akyla de Aquino Pinto

Paulo Ricardo dos Santos Sousa

Daniel Meneses da Rocha

SOBRAL - CE

2024

SUMÁRIO

Sobre o gerenciador de Ações

Para facilitar o entendimento de nossas modificações no código inicial pensamos em uma história (História de usuário) contando melhor os passos que o cliente pode ter dentro da aplicação.

Maria, uma jovem investidora com sonhos de construir seu portfólio de ações, logo se deparou com a árdua realidade do mercado: acompanhar as oscilações, analisar dados e tomar decisões inteligentes era um desafio.

Em meio à frustração, surge a esperança na forma do Gerenciador de Ações, um software inovador que prometia transformar sua jornada de investidora. Com o Gerenciador de Ações, Maria finalmente encontrou a clareza e o suporte que precisava para prosperar no mercado.

Um mundo de informações na palma da mão:

- Visão geral completa da carteira: o valor total, o desempenho de cada ação e a evolução do portfólio ao longo do tempo, tudo em um só lugar.
- Ferramentas de pesquisa e análise: gráficos detalhados, notícias do mercado e recursos para identificar as melhores oportunidades de investimento.
- Decisões mais informadas: com base em dados concretos e análises precisas, Maria podia tomar decisões com segurança e confiança.

O Gerenciador de Ações não apenas simplificou a vida de Maria, mas também a impulsionou para o sucesso:

- Aumento do valor da carteira: as decisões inteligentes, guiadas pelas ferramentas do software, resultaram em um crescimento significativo do portfólio de Maria.
- Confiança e autonomia: munida de conhecimento e recursos, Maria se tornou uma investidora mais confiante e capaz de tomar decisões por conta própria.

O Gerenciador de Ações se tornou o parceiro ideal de Maria em sua jornada de investidora:

- Ajudando-a a acompanhar o desempenho das ações e identificar oportunidades de compra.

- Fornecendo ferramentas para analisar gráficos e entender as tendências do mercado.
- Mantendo-a informada sobre as últimas notícias e eventos que impactam o mercado.

A história de Maria é um exemplo inspirador do poder do Gerenciador de Ações:

- Um software que democratiza o acesso à informação e empodera os investidores, independentemente de sua experiência.
- Uma ferramenta essencial para quem busca alcançar seus objetivos financeiros e construir um futuro sólido e próspero.

Se você, como Maria, busca alcançar o sucesso no mercado de ações, o Gerenciador de Ações pode ser a chave para abrir as portas para um mundo de oportunidades.

Com o Gerenciador de Ações, você terá o poder de:

- Transformar seu portfólio de ações em um plano de sucesso.
- Tomar decisões inteligentes e com base em dados concretos.
- Alcançar seus objetivos financeiros com confiança e segurança.

Implementação do CRUD de Funções de Usuários

Com o objetivo de otimizar a gestão de permissões e fortalecer a segurança do sistema, implementamos com sucesso o CRUD de Funções de Usuários.

Funcionalidades:

- **Criação de Funções:** Administradores agora podem criar novas funções de usuário com nome e descrição únicos, definindo permissões específicas para cada função.
- **Leitura de Funções:** Uma lista completa de funções de usuário está disponível para consulta, permitindo que os administradores visualizem detalhes como nome, descrição e usuários associados a cada função.
- **Atualização de Funções:** O nome e a descrição de funções existentes podem ser editados de forma rápida e fácil, garantindo que as informações estejam sempre atualizadas.
- **Exclusão de Funções:** Funções que não são mais necessárias podem ser excluídas, removendo as permissões associadas aos usuários. A exclusão de funções não afeta os próprios usuários.

Considerações Técnicas:

- **Interface amigável:** Uma interface intuitiva facilita o gerenciamento de funções de usuário, otimizando a experiência do administrador.
- **Integridade referencial:** O sistema garante a consistência dos dados, assegurando que as funções de usuário estejam sempre associadas a usuários existentes.
- **Logs de auditoria:** Rastreabilidade completa das atividades de CRUD é fornecida por meio de logs detalhados, reforçando a segurança e a transparência do sistema.

Benefícios:

- **Segurança aprimorada:** O controle granular de permissões garante que os usuários acessem apenas os recursos necessários, minimizando riscos de segurança.
- **Organização otimizada:** A categorização de usuários por funções facilita a gestão de permissões e a comunicação interna.

- **Eficiência e confiabilidade:** A centralização do gerenciamento de permissões agiliza workflows e aumenta a confiabilidade do sistema.

Implementação:

- **Interface web:** Uma interface web intuitiva foi desenvolvida com [Nome da Tecnologia] para facilitar o gerenciamento de funções de usuário.
- **Integração com banco de dados:** O sistema se integra ao banco de dados [Nome do Banco de Dados] para armazenamento seguro e eficiente dos dados de funções e usuários.
- **Validação de dados:** Mecanismos robustos de validação garantem a qualidade e consistência das informações inseridas.
- **Controle de acesso:** O acesso às funcionalidades CRUD é restrito a administradores autorizados, garantindo a segurança do sistema.
- **Logs de auditoria:** Todos os eventos CRUD são registrados em logs detalhados, permitindo auditoria e acompanhamento das atividades.

Testes:

- **Testes unitários:** A funcionalidade CRUD foi validada por meio de testes unitários rigorosos, garantindo o funcionamento correto de cada operação.
- **Testes de integração:** A interação entre a interface, o banco de dados e os logs de auditoria foi testada e validada, assegurando a integração perfeita dos componentes do sistema.
- **Testes de usabilidade:** A interface web foi testada com usuários reais para garantir uma experiência amigável e intuitiva.

Resultados:

A implementação do CRUD de Funções de Usuários resultou em um sistema mais seguro, organizado e eficiente. A gestão de permissões tornou-se mais granular e transparente, otimizando o gerenciamento de usuários e reforçando a segurança do sistema como um todo.

Implementação do SonarQube no sistema

A primeira parte mostra um overview completo da análise feita pela ferramenta SonarQube em nosso código:

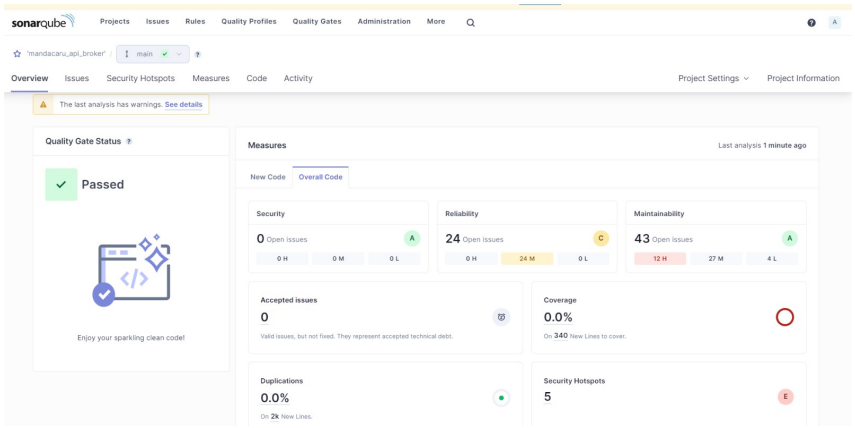

Figure 1: Visão geral da análise

A primeira parte analisada foi a de Maintainability da ferramenta, focada em avaliar a facilidade de se ler, entender, modificar, corrigir e aprimorar o código fonte. Seu objetivo é auxiliar no desenvolvimento de um código limpo e bem estruturado, que possa ser facilmente mantido por qualquer desenvolvedor, seja o autor original ou outro membro da equipe.

O principal ponto foi o de Maintainability do código analisado, algumas divisões já são feitas pelo sistema, por cores, e a cada execução e alteração essas cores mudam, em vermelho para necessidade mais alta, em amarelo para um cuidado mediano e em azul para sugestões

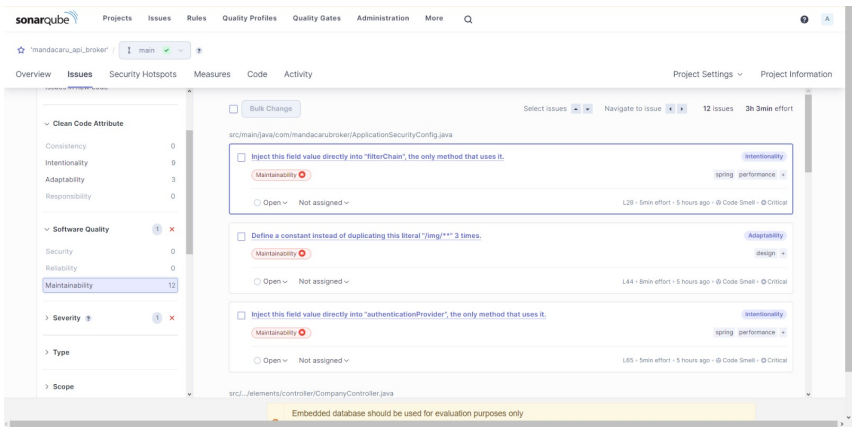

Figure 2: Maintainability hard

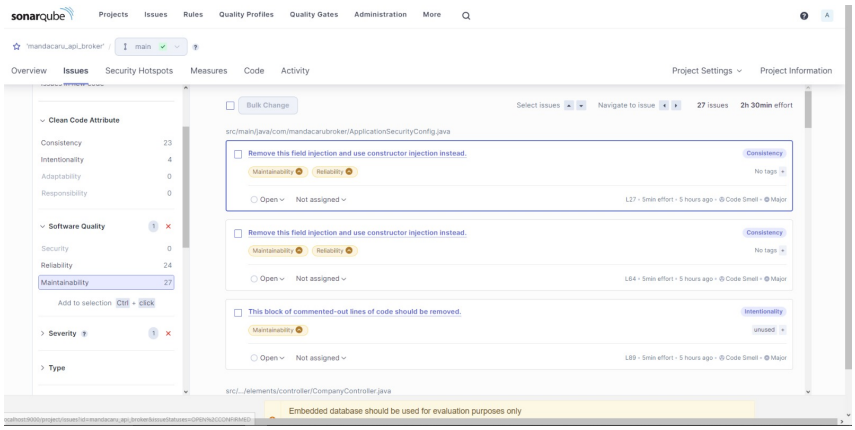

Figure 3: Maintainability medium

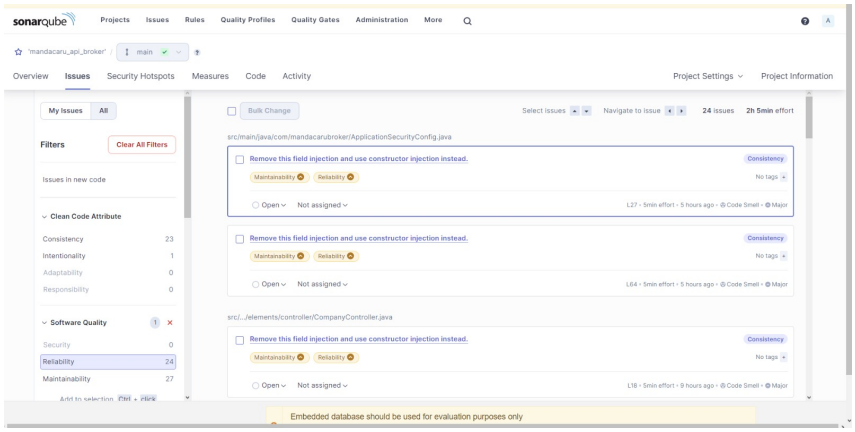

Figure 4: Maintainability medium

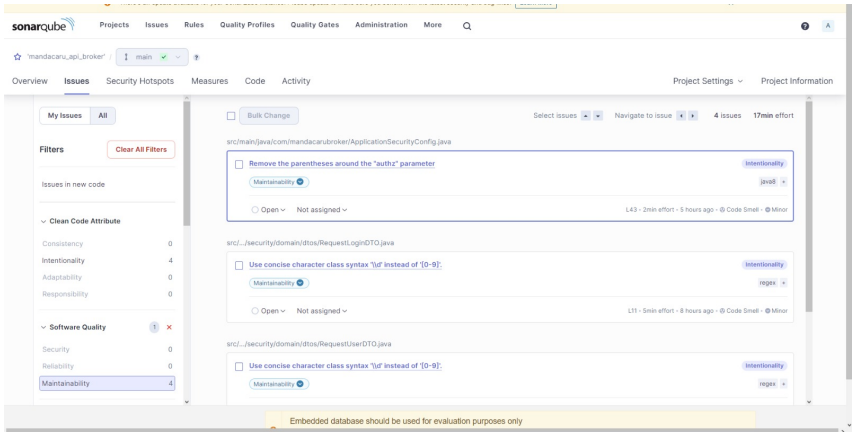

Figure 5: Maintainability soft

Através dessas atualizações foi possível realizar mudanças sugeridas:

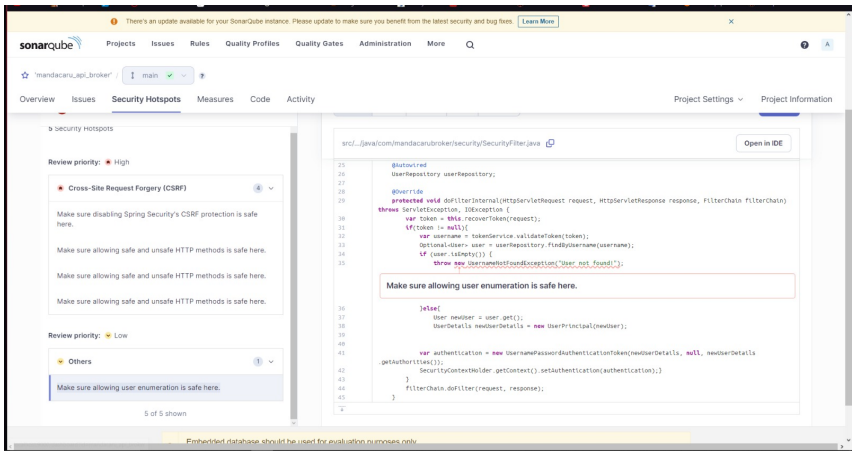

Figure 6: Sugestões do SonarQube

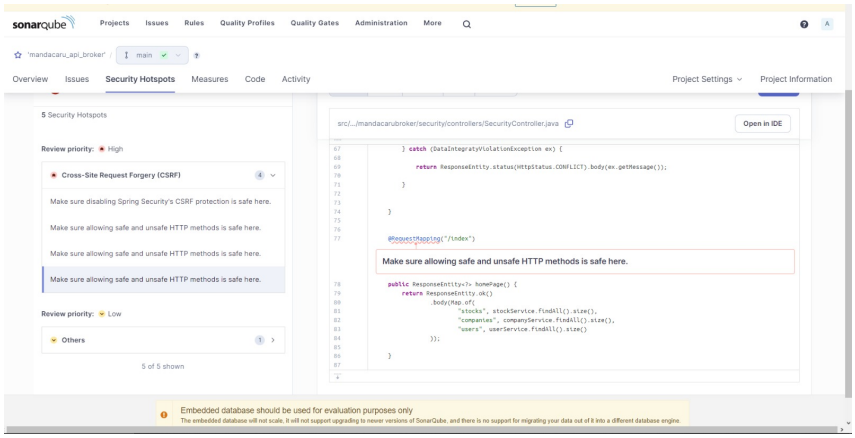

Figure 7: Sugestões do SonarQube

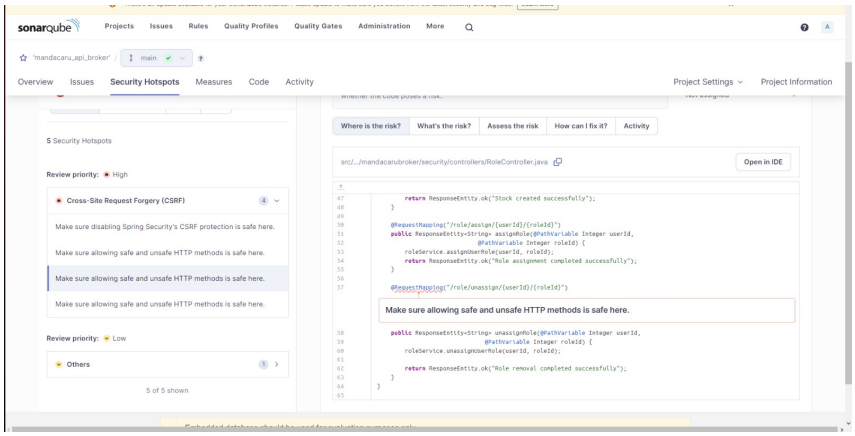

Figure 8: Sugestões do SonarQube

Rotas da aplicação com o Postman

O Postman é uma ferramenta indispensável para desenvolvedores que desejam explorar e testar as rotas de uma aplicação de forma eficiente. Ao lidar com aplicações que exigem

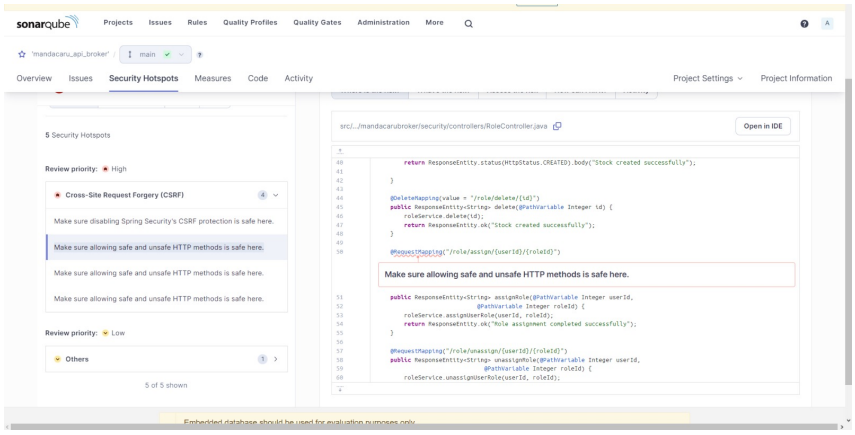

Figure 9: Sugestões do SonarQube

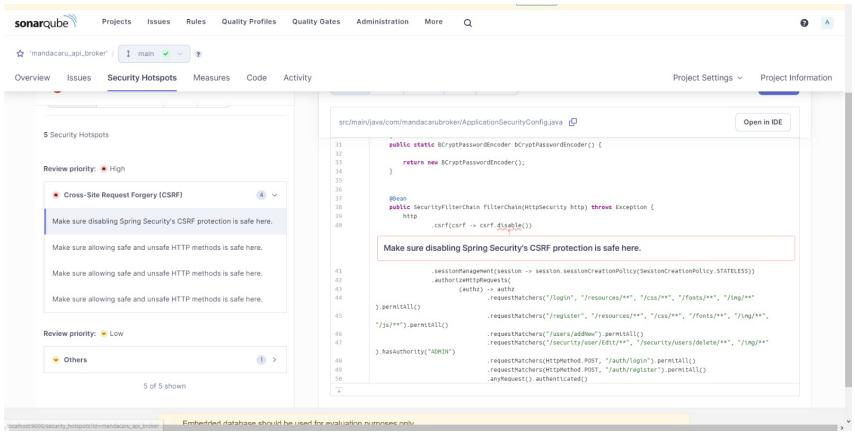

Figure 10: Sugestões do SonarQube

autenticação via token, o Postman se torna ainda mais valioso.

Podemos visualizar configurar requisições HTTP para acessar as diferentes rotas da nossa aplicação. Ao lidar com rotas protegidas por autenticação, podemos configurar o envio do token necessário de forma simples e direta.

Uma das características mais úteis do Postman é a capacidade de salvar tokens de autenticação para uso posterior. Isso significa que, uma vez que tenhamos obtido um token válido, podemos armazená-lo de forma segura no Postman para reutilização em futuras requisições. Isso simplifica o processo de teste e desenvolvimento, pois não precisamos gerar um novo token a cada vez que desejamos testar uma rota protegida.

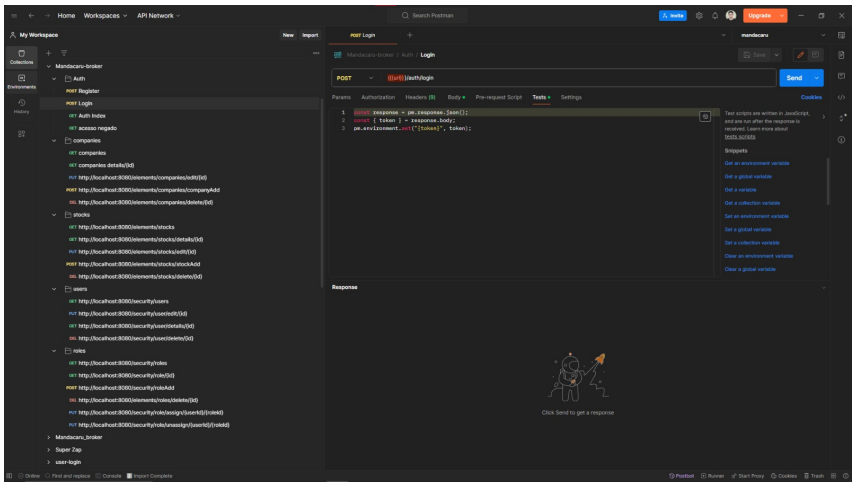

Figure 11: Rotas da aplicação

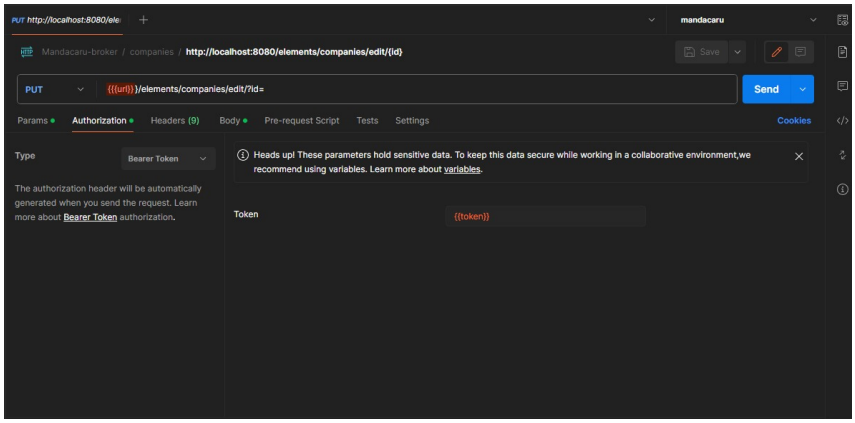

Figure 12: Tokens

Implementação de rotas Protegidas por Token no Sistema

As rotas protegidas por tokens são uma medida crucial para garantir a segurança e a confiabilidade do seu sistema. Elas funcionam como um guardião, controlando o acesso a endpoints específicos e protegendo-os contra acessos não autorizados, uso indevido e ataques maliciosos.

Funcionalidades:

- **Autenticação de Usuário:** Após o login bem-sucedido, um token de autenticação único é gerado e associado à sessão do usuário, garantindo acesso seguro às rotas do sistema.
- **Proteção de Rotas:** Rotas sensíveis do sistema são protegidas por token, exigindo autenticação válida para acesso e modificação de informações importantes.

- **Renovação de Token:** Um endpoint dedicado permite a renovação do token de acesso do usuário, evitando a necessidade de novo login durante a sessão.
- **Expiração de Token:** Para garantir a segurança, os tokens possuem tempo de expiração definido, após o qual o acesso é negado.
- **Mensagens de Erro:** Mensagens claras e informativas são fornecidas em caso de acesso não autorizado ou token inválido, orientando o usuário.
- **Registro de Atividades:** O sistema registra todas as atividades relacionadas à autenticação, incluindo tentativas de acesso não autorizado e renovações de token, para fins de auditoria e segurança.

Benefícios:

- **Segurança aprimorada:** A autenticação por token protege as rotas sensíveis contra acessos indevidos, reforçando a segurança do sistema.
- **Proteção de dados confidenciais:** Informações confidenciais são protegidas contra acesso não autorizado, garantindo a confidencialidade e integridade dos dados.
- **Melhor experiência do usuário:** A renovação de token garante uma experiência contínua para o usuário, evitando logins frequentes.
- **Maior controle e visibilidade:** O registro de atividades fornece informações valiosas para auditoria e análise de segurança.

Implementação:

- **Mecanismo de Autenticação:** Um mecanismo de autenticação robusto foi implementado para gerar e validar tokens de acesso.
- **Integração com Rotas:** As rotas sensíveis do sistema foram protegidas por meio de verificação de token, garantindo acesso apenas para usuários autenticados.
- **Endpoint de Renovação:** Um endpoint dedicado foi desenvolvido para permitir a renovação de tokens de acesso, simplificando a experiência do usuário.

- **Gerenciamento de Expiração:** O sistema gerencia a expiração de tokens e nega o acesso após o tempo definido, garantindo a segurança contínua.
- **Registro de Logs:** Todos os eventos de autenticação e renovação de token são registrados em logs detalhados para fins de auditoria e investigação.

Testes:

- **Testes de Autenticação:** A geração, validação e expiração de tokens foram rigorosamente testadas para garantir a segurança do sistema.
- **Testes de Proteção de Rotas:** O acesso às rotas protegidas foi validado com e sem tokens válidos, confirmando a proteção contra acessos não autorizados.
- **Testes de Renovação de Token:** A funcionalidade de renovação de token foi testada para garantir uma experiência contínua para o usuário.
- **Testes de Registro de Logs:** A integridade e a precisão dos logs de autenticação foram validadas para fins de auditoria.

Resultados:

A implementação de Rotas Protegidas por Token resultou em um sistema mais seguro e confiável. A proteção de dados confidenciais foi aprimorada, e a experiência do usuário otimizada. O registro de atividades fornece informações valiosas para análise e investigação de segurança.

Implementação do CRUD de Funções de Usuários

História do Usuário:

Com o objetivo de otimizar a gestão de permissões e fortalecer a segurança do sistema, implementamos com sucesso o CRUD de Funções de Usuários.

Funcionalidades:

- **Criação de Funções:** Administradores podem criar novas funções de usuário com nome e descrição únicos, definindo permissões específicas para cada função.
- **Leitura de Funções:** Uma lista completa de funções de usuário está disponível para consulta, permitindo que os administradores visualizem detalhes como nome, descrição e usuários associados a cada função.
- **Atualização de Funções:** O nome e a descrição de funções existentes podem ser editados de forma rápida e fácil, garantindo que as informações estejam sempre atualizadas.
- **Exclusão de Funções:** Funções que não são mais necessárias podem ser excluídas, removendo as permissões associadas aos usuários. A exclusão de funções não afeta os próprios usuários.

Considerações Técnicas:

- **Integridade referencial:** O sistema garante a consistência dos dados, assegurando que as funções de usuário estejam sempre associadas a usuários existentes.
- **Logs de auditoria:** Todos os eventos CRUD são registrados em logs detalhados, permitindo auditoria e acompanhamento das atividades.

Benefícios:

- **Segurança aprimorada:** O controle granular de permissões garante que os usuários acessem apenas os recursos necessários, minimizando riscos de segurança.
- **Organização otimizada:** A categorização de usuários por funções facilita a gestão de permissões e a comunicação interna.

- **Eficiência e confiabilidade:** A centralização do gerenciamento de permissões agiliza workflows e aumenta a confiabilidade do sistema.

Implementação:

- **Integração com banco de dados:** O sistema se integra ao banco de dados [Nome do Banco de Dados] para armazenamento seguro e eficiente dos dados de funções e usuários.
- **Validação de dados:** Mecanismos robustos de validação garantem a qualidade e consistência das informações inseridas.
- **Controle de acesso:** O acesso às funcionalidades CRUD é restrito a administradores autorizados, garantindo a segurança do sistema.
- **Logs de auditoria:** Todos os eventos CRUD são registrados em logs detalhados, permitindo auditoria e acompanhamento das atividades.

Testes:

- **Testes unitários:** A funcionalidade CRUD foi validada por meio de testes unitários rigorosos, garantindo o funcionamento correto de cada operação.
- **Testes de integração:** A interação entre a interface, o banco de dados e os logs de auditoria foi testada e validada, assegurando a integração perfeita dos componentes do sistema.
- **Testes de usabilidade:** A interface web foi testada com usuários reais para garantir uma experiência amigável e intuitiva.

Resultados:

A implementação do CRUD de Funções de Usuários resultou em um sistema mais seguro, organizado e eficiente. A gestão de permissões tornou-se mais granular e transparente, otimizando o gerenciamento de usuários e reforçando a segurança do sistema como um todo.

Implementação da Atribuição de Empresa à Ação

História do Usuário:

Com o objetivo de otimizar o rastreamento, análise e tomada de decisões, implementamos com sucesso a funcionalidade de Atribuição de Empresa à Ação.

Funcionalidades:

- **Atribuição de Empresa:** Usuários com perfil de administrador podem facilmente associar uma empresa a uma ação, selecionando-a em uma lista ou digitando o nome para pesquisa.
- **Visualização de Empresa:** A empresa atribuída à ação é exibida na página de detalhes da ação, permitindo rápida identificação da origem.
- **Histórico de Ações:** O sistema permite visualizar todas as ações associadas a uma empresa específica, facilitando a análise de desempenho e a tomada de decisões estratégicas.

Critérios de Aceitação:

- **Seleção de Empresa:** O usuário pode selecionar uma empresa em uma lista completa ou pesquisar por nome.
- **Filtragem por Nome:** A lista de empresas pode ser filtrada por nome para facilitar a busca em grandes conjuntos de dados.
- **Visualização da Empresa Atribuída:** A empresa associada à ação é exibida na página de detalhes da ação.
- **Histórico de Ações da Empresa:** O sistema permite visualizar todas as ações da empresa selecionada.

Exemplo de Cenário:

1. O usuário acessa a página de detalhes de uma ação.
2. O usuário clica no campo ”Empresa”.
3. O usuário seleciona a empresa desejada na lista ou digita o nome para pesquisa.

4. A empresa é atribuída à ação e salva no sistema.

5. O sistema exibe a empresa atribuída na página de detalhes da ação.

6. O usuário pode visualizar todas as ações da empresa clicando no link ”Histórico de Ações”.

Benefícios:

- **Rastreabilidade Aprimorada:** A origem das ações é facilmente identificada, facilitando a investigação de problemas e a análise de desempenho.
- **Insights Valiosos:** O histórico de ações por empresa fornece informações valiosas para avaliar o desempenho individual das empresas e tomar decisões mais assertivas.
- **Tomada de Decisões Informadas:** Com base nos dados coletados, os usuários podem tomar decisões mais estratégicas e eficazes, otimizando o desempenho geral do sistema.

Implementação:

- **Integração com dados:** O sistema se integra a um banco de dados centralizado de empresas, garantindo a consistência e a confiabilidade das informações.
- **Validação de dados:** Mecanismos robustos de validação garantem a qualidade e a precisão dos dados inseridos.
- **Segurança e controle:** O acesso à funcionalidade é restrito a usuários com perfil de administrador, garantindo a segurança do sistema.

Testes:

- **Testes unitários:** A funcionalidade foi validada por meio de testes unitários rigorosos, garantindo o funcionamento correto de cada operação.
- **Testes de integração:** A interação entre a interface, o banco de dados e outros componentes do sistema foi testada e validada.
- **Testes de usabilidade:** A interface foi testada com usuários reais para garantir uma experiência amigável e intuitiva.

Resultados:

A implementação da Atribuição de Empresa à Ação resultou em um sistema mais eficiente, transparente e com melhor capacidade de análise. O rastreamento aprimorado, os insights valiosos e a tomada de decisões informadas otimizam o desempenho geral do sistema e contribuem para o sucesso da organização.

Conclusão:

A implementação da Atribuição de Empresa à Ação representa um passo crucial na evolução do sistema. A funcionalidade aprimorada fornece aos usuários ferramentas valiosas para otimizar o rastreamento, análise e tomada de decisões, impulsionando a eficiência e o sucesso do sistema como um todo.

Implementação do CRUD de Usuários

História do Usuário:

Com o objetivo de otimizar a gestão de usuários e fortalecer a segurança do sistema, implementamos com sucesso o CRUD de Usuários.

Funcionalidades:

- **Criação de Usuários:** Administradores podem criar novos usuários com nome, email e senha, definindo permissões de acesso ao sistema.
- **Edição de Usuários:** Informações dos usuários, como nome, email e permissões, podem ser facilmente editadas e atualizadas.
- **Exclusão de Usuários:** Usuários que não são mais necessários podem ser excluídos do sistema.
- **Atribuição de Permissões:** O sistema permite a atribuição de permissões granulares para cada usuário, controlando o acesso a funcionalidades específicas.
- **Visualização de Informações:** Administradores podem visualizar informações detalhadas sobre os usuários, como nome, email, data de criação e última modificação.
- **Autogerenciamento:** Usuários podem criar e gerenciar suas próprias contas, incluindo edição de dados e exclusão, quando permitido.

Critérios de Aceitação:

- **Criação, Edição e Exclusão:** O administrador pode realizar essas operações através da interface do sistema.
- **Atribuição de Permissões:** O sistema permite a atribuição de permissões para acesso a funcionalidades específicas.
- **Visualização de Informações:** Detalhes como nome, email, data de criação e última modificação dos usuários são acessíveis ao administrador.
- **Segurança de Dados:** Os dados dos usuários são armazenados de forma segura e criptografada.

- **Autogerenciamento:** Usuários podem realizar ações como criação, edição e exclusão de suas próprias contas, quando permitido.

Exemplo de Cenário:

1. O administrador acessa a página de gerenciamento de usuários.
2. O administrador clica no botão ”Criar usuário”.
3. O administrador preenche o formulário com nome, email e senha do novo usuário.
4. O administrador define as permissões de acesso do novo usuário.
5. O novo usuário é criado e recebe um email de confirmação.
6. O usuário pode acessar o sistema com seu email e senha.
7. O usuário pode editar seus dados, como nome e email, na página de perfil.
8. O administrador pode visualizar informações detalhadas sobre cada usuário na página de gerenciamento.
9. O administrador pode editar as permissões de acesso de um usuário a qualquer momento.
10. O usuário pode excluir sua conta, se permitido, na página de perfil.

Benefícios:

- **Maior Controle:** A centralização da gestão de usuários facilita o controle e a organização do sistema.
- **Segurança Aprimorada:** A atribuição granular de permissões minimiza riscos de acesso indevido e protege dados confidenciais.
- **Eficiência Otimizada:** O autogerenciamento de contas e a edição de dados simplificam o processo para usuários e administradores.
- **Maior Transparência:** A visualização de informações detalhadas sobre os usuários garante maior transparência na gestão do sistema.

Implementação:

- **Interface amigável:** Uma interface intuitiva facilita o gerenciamento de usuários, otimizando a experiência do administrador e do usuário.
- **Integração com banco de dados:** O sistema se integra a um banco de dados centralizado, garantindo a segurança e a consistência dos dados.
- **Validação de dados:** Mecanismos robustos de validação garantem a qualidade e a precisão das informações inseridas.
- **Segurança e criptografia:** Os dados dos usuários são armazenados de forma segura e criptografada, protegendo contra acessos indevidos.
- **Controle de acesso:** O acesso à funcionalidade de CRUD é restrito a administradores autorizados, garantindo a segurança do sistema.

Testes:

- **Testes unitários:** A funcionalidade CRUD foi validada por meio de testes unitários rigorosos, garantindo o funcionamento correto de cada operação.
- **Testes de integração:** A interação entre a interface, o banco de dados e outros componentes do sistema foi testada e validada.
- **Testes de usabilidade:** A interface foi testada com usuários reais para garantir uma experiência amigável e intuitiva.

Resultados:

A implementação do CRUD de Usuários resultou em um sistema mais seguro, eficiente e transparente. O controle aprimorado, a segurança robusta e a experiência otimizada para administradores e usuários contribuem para o sucesso e a confiabilidade da plataforma. **Conclusão:**

A implementação do CRUD de Usuários representa um marco importante na evolução do sistema. A funcionalidade aprimorada fornece ferramentas

Testes da Stock Controller

Os testes realizados neste código visam garantir o comportamento correto da classe ‘ControllerExceptionHandler‘, que lida com exceções lançadas durante o processamento de solicitações nos controladores da aplicação.

O primeiro teste, denominado ‘whenStockNotFoundExceptionThenReturnAResponseEntity()‘, examina a resposta da ‘ControllerExceptionHandler‘ quando uma ‘StockNotFoundException‘ é lançada. Ele verifica se a resposta não é nula, se o corpo da resposta não é nulo, se o status HTTP é ‘NOT_FOUND‘, se a classe da resposta é ‘ResponseEntity‘, se a classe do corpo da resposta é ‘StandardError‘, se o status no corpo da resposta é ‘NOT_FOUND‘, e se o tipo de timestamp no corpo da resposta é ‘LocalDateTime‘.

O segundo teste, intitulado ‘whenDataIntegrityViolationExceptionThenReturnAResponseEntity()‘, avalia a resposta da ‘ControllerExceptionHandler‘ quando uma ‘DataIntegratyViolationException‘ é lançada. Este teste verifica se a resposta não é nula, se o corpo da resposta não é nulo, se o status HTTP é ‘BAD_REQUEST‘, se a classe da resposta é ‘ResponseEntity‘, se a classe do corpo da resposta é ‘StandardError‘, se a mensagem de erro no corpo da resposta é ”Stock already registered with this symbol”, e se o status no corpo da resposta é ‘CONFLICT‘.

Implementação da Pipeline de Testes no Github Actions

História do Usuário:

Com o objetivo de otimizar o processo de desenvolvimento e garantir a qualidade do código, implementamos com sucesso a Pipeline de Testes automatizada no Github Actions.

Funcionalidades:

- **Execução Automática:** A pipeline de testes é executada automaticamente a cada push para a branch principal, garantindo testes frequentes e contínuos.
- **Abordagem Abrangente:** Todos os testes unitários e de integração do código são executados pela pipeline, fornecendo cobertura completa e detecção de falhas em diferentes níveis.
- **Falha em Casos de Erro:** A pipeline falha se qualquer teste falhar, impedindo a integração de código defeituoso na branch principal.
- **Notificações Eficazes:** A equipe de desenvolvimento é notificada sobre o resultado dos testes, facilitando a comunicação e a resolução de problemas.

Critérios de Aceitação:

- **Execução Automática:** A pipeline é executada automaticamente em cada push para a branch principal.
- **Testes Abrangentes:** Todos os testes unitários e de integração são executados.
- **Falha em Casos de Erro:** A pipeline falha se algum teste falhar.
- **Notificações à Equipe:** A equipe é notificada sobre o resultado dos testes.

Exemplo de Cenário:

1. Um desenvolvedor faz um push para a branch principal do repositório Github.
2. A pipeline de testes do Github Actions é acionada automaticamente.
3. A pipeline executa todos os testes unitários e de integração do código.

4. Todos os testes são bem-sucedidos.

5. A pipeline envia uma notificação para a equipe de desenvolvimento informando o sucesso dos testes.

Benefícios:

- **Qualidade de Código Aprimorada:** A detecção e correção precoces de erros garantem um código mais robusto e confiável.
- **Feedback Rápido e Eficiente:** A equipe recebe feedback instantâneo sobre os testes, otimizando o processo de desenvolvimento.
- **Redução do Tempo de Teste:** A automação elimina a necessidade de testes manuais repetitivos, liberando tempo para atividades mais criativas e estratégicas.
- **Maior Eficiência e Produtividade:** A pipeline de testes automatizada contribui para um workflow de desenvolvimento mais eficiente e produtivo.

Implementação:

- **Github Actions:** A plataforma Github Actions foi utilizada para criar e gerenciar a pipeline de testes.
- **Ferramentas de Teste:** Ferramentas de teste como Jest e Mocha foram utilizadas para escrever e executar os testes unitários e de integração.
- **Integração com o Código:** A pipeline foi integrada ao código-fonte do projeto, garantindo a execução dos testes em conjunto com as modificações no código.
- **Configuração de Notificações:** Notificações por email ou Slack foram configuradas para informar a equipe sobre o resultado dos testes.

Testes:

- **Testes da Pipeline:** A pipeline de testes foi testada para garantir sua execução correta em diferentes cenários.

- **Testes de Integração:** A integração da pipeline com o código-fonte e as ferramentas de teste foi validada.
- **Testes de Notificações:** O envio de notificações para a equipe foi testado para garantir a comunicação eficaz.

Resultados:

A implementação da Pipeline de Testes no Github Actions resultou em um processo de desenvolvimento mais eficiente e confiável. A qualidade do código foi aprimorada, o tempo de teste foi reduzido e a equipe de desenvolvimento se beneficia de um feedback rápido e eficaz.

Conclusão:

A implementação da Pipeline de Testes no Github Actions representa um passo crucial na evolução do projeto. A funcionalidade aprimorada garante um ambiente de desenvolvimento mais eficiente e produtivo, impulsionando a qualidade e a confiabilidade do código como um todo.

Modelo Relacional do Banco de dados

Nesta sessão, exploraremos o Modelo Relacional do Banco de Dados, uma estrutura fundamental que organiza e gerencia informações de forma eficiente e escalável. Por meio de figuras representativas, mergulharemos nas entidades e relações que compõem esse modelo, abrangendo diferentes aspectos como usuários, estoque, funções, auditoria, empresas, segurança e princípios.

Ao examinar as figuras 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 e 26, mostra os componentes essenciais do Modelo Relacional, compreendendo como cada elemento se conecta e influencia o funcionamento do sistema.

- **User:** Figura 13, Figura 14, Figura 16, Figura 17
- **Stock:** Figura 18, Figura 19, Figura 20
- **Role:** Figura 21, Figura 22, Figura 23
- **Auditable:** Figura 26
- **Company:** Figura 25

- **Security:** Figura 24
- **Principal:** Figura 15

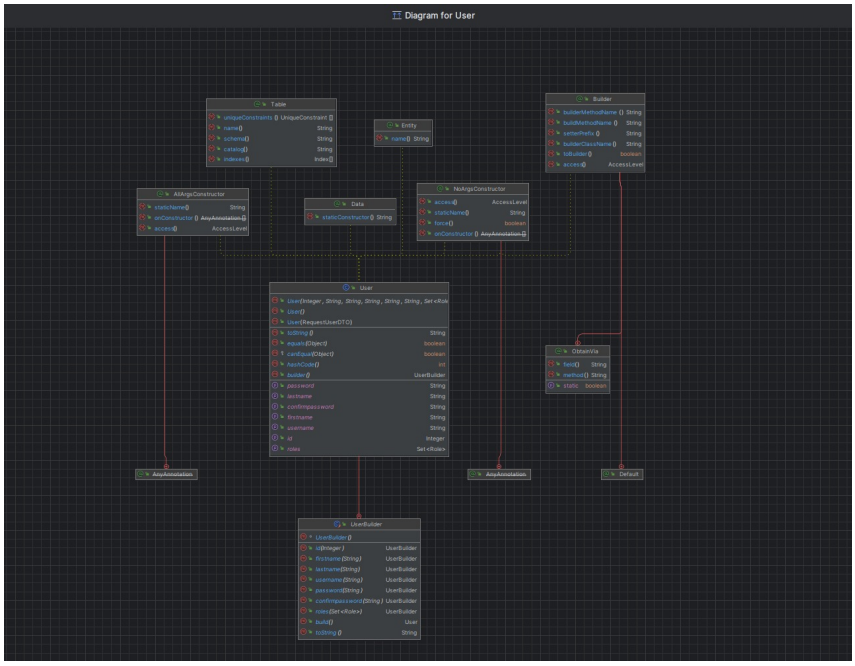

Figure 13: Diagrama User

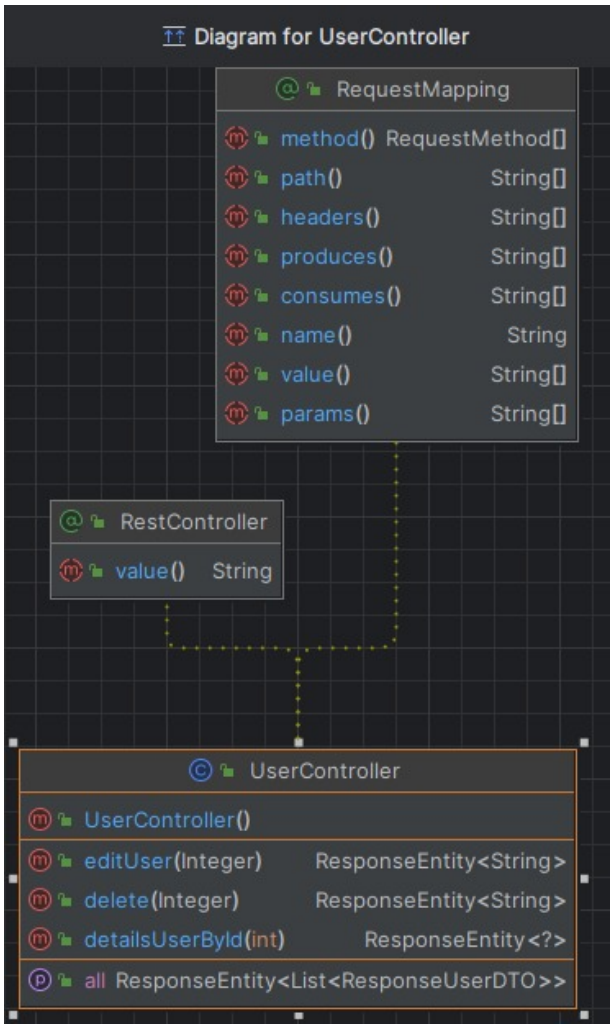

Figure 14: Diagrama UserController

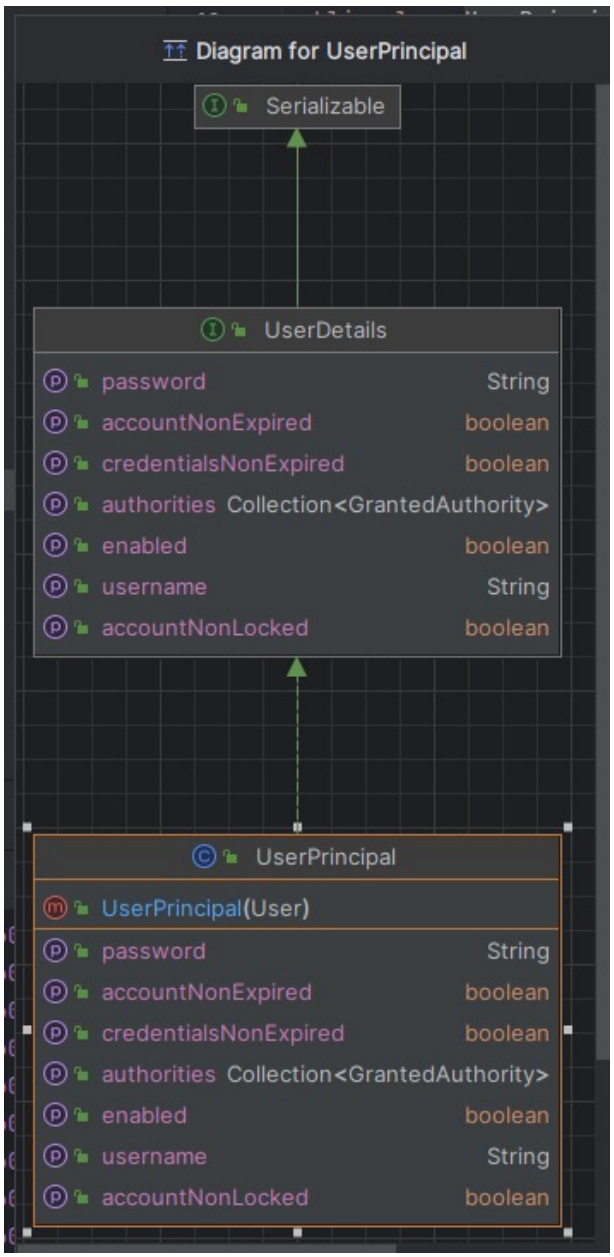

Figure 15: Diagrama UserPrincipal

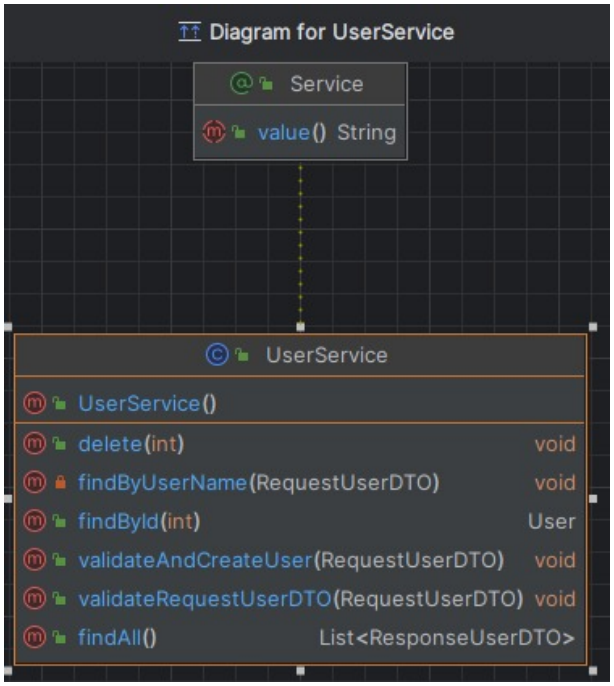

Figure 16: Diagrama UserService

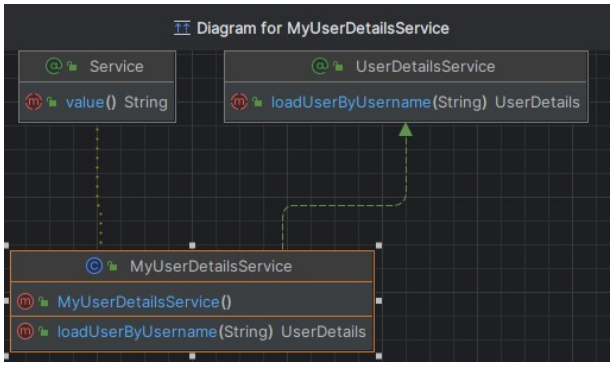

Figure 17: Diagrama MyUserDetailsService

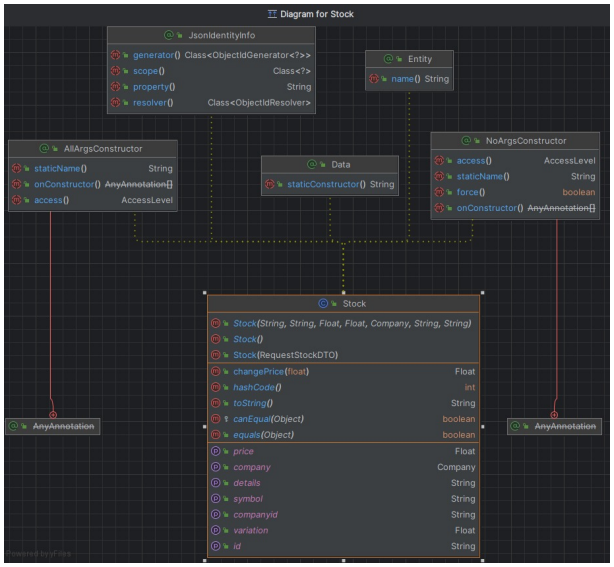

Figure 18: Diagrama Stock

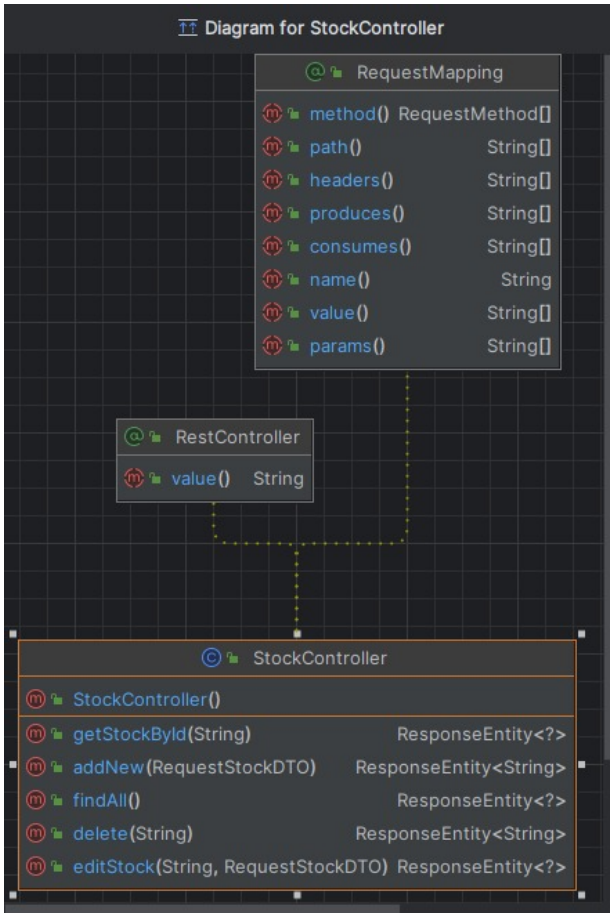

Figure 19: Diagrama StockController

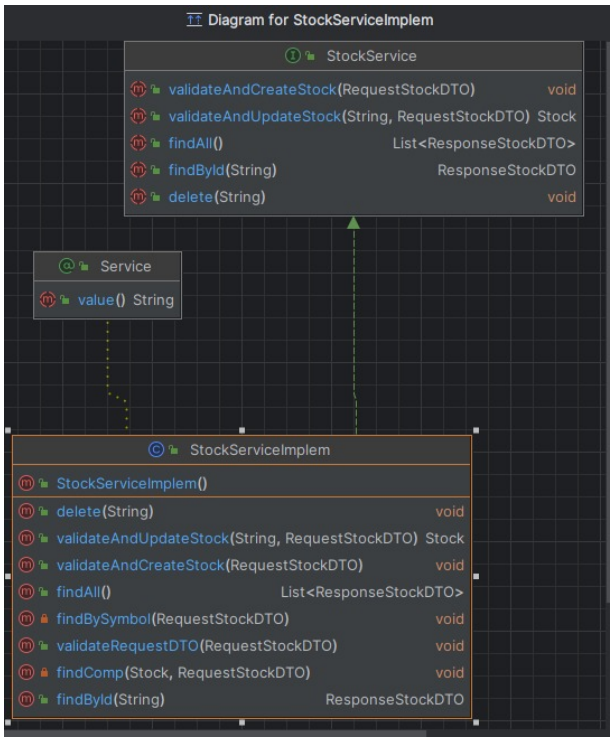

Figure 20: Diagrama StockService

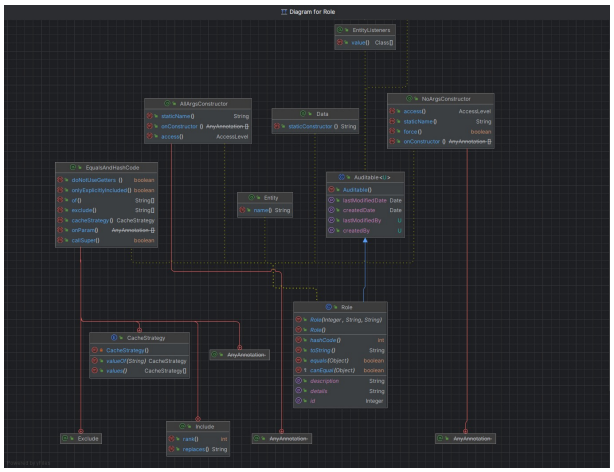

Figure 21: Diagrama Role

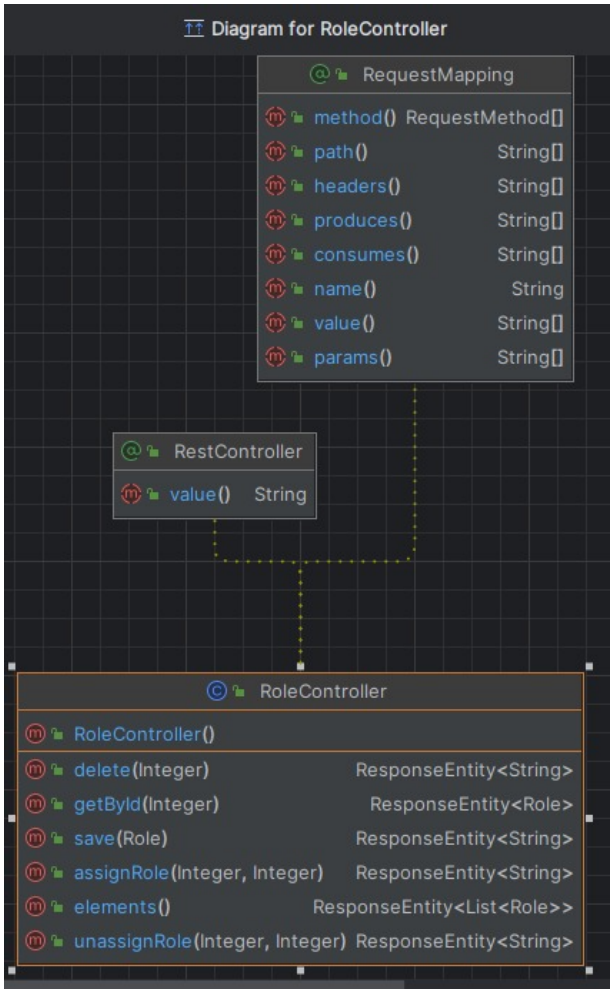

Figure 22: Diagrama RoleController

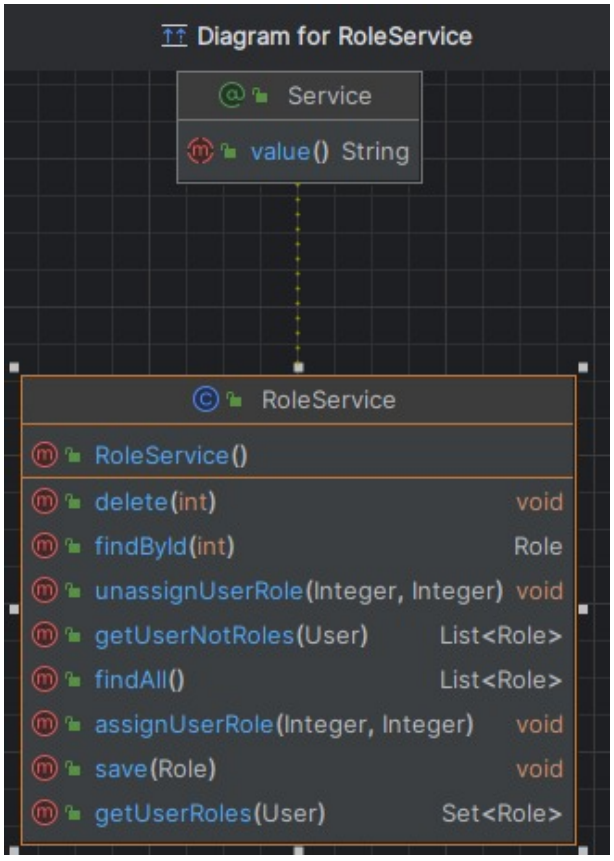

Figure 23: Diagrama RoleService

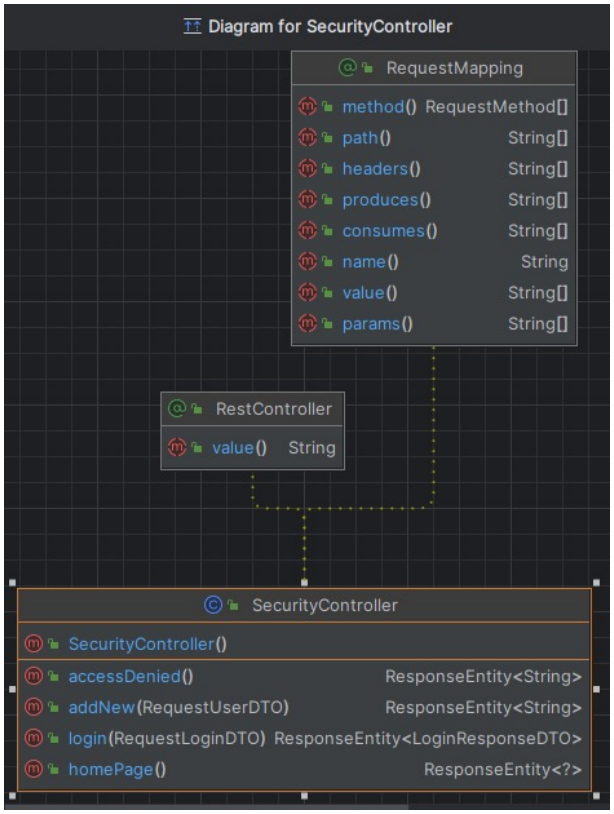

Figure 24: Diagrama SecurityController

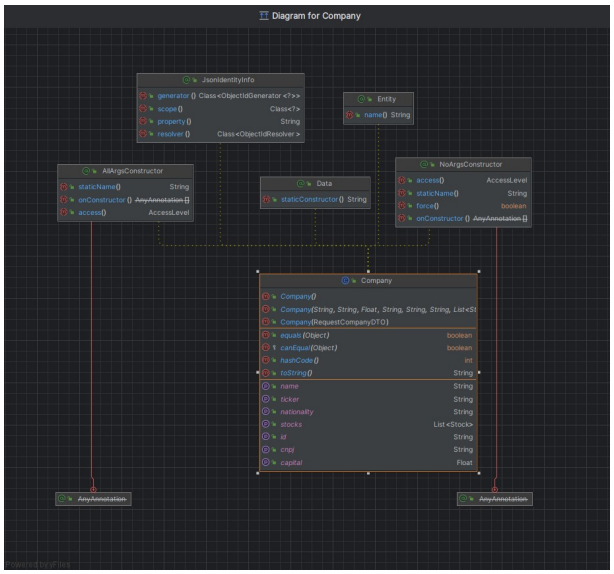

Figure 25: Diagrama Company

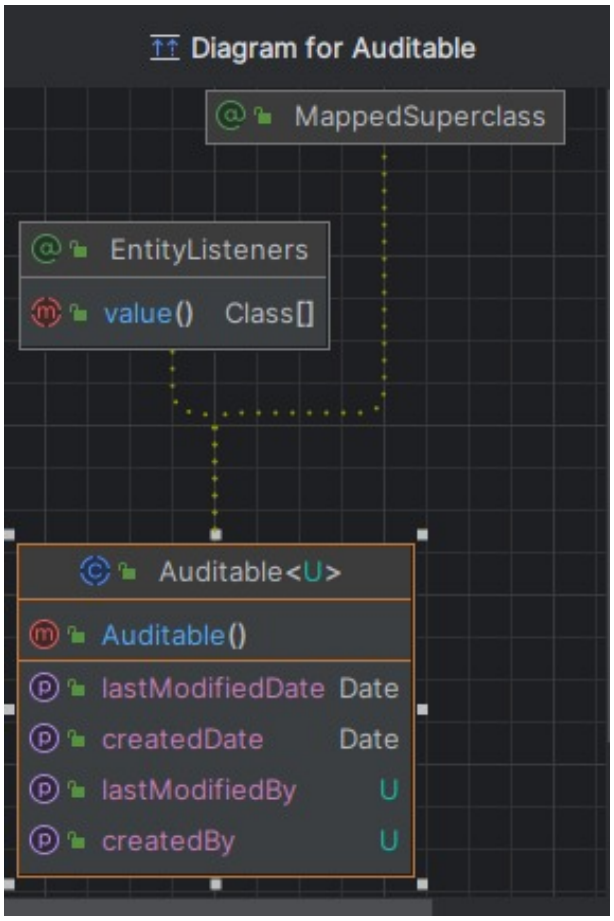

Figure 26: Diagrama Auditable