**Sprint 1: Configuração Inicial e Estrutura do Projeto**

* Configurar o ambiente de desenvolvimento.
* Inicializar o repositório Git e criar a estrutura básica do projeto.
* Implementar a camada de Banco para distribuir transações e informações das contas.
* Definir as rotas de API relacionadas ao Banco.

**Sprint 2: Desenvolvimento do Seletor**

* Implementar a camada de Seletor para cadastrar e verificar a viabilidade dos validadores.
* Implementar a lógica de seleção de validadores para as transações ativas.
* Definir as rotas de API relacionadas ao Seletor.

**Sprint 3: Implementação do Validador**

* Implementar a camada de Validador para validar transações conforme as regras do ecossistema.
* Desenvolver a lógica de validação de transações.
* Definir as rotas de API relacionadas ao Validador.

**Sprint 4: Refinamento e Testes**

* Refatorar e otimizar o código conforme necessário.
* Realizar testes de unidade e integração para garantir o funcionamento correto das funcionalidades.
* Corrigir eventuais bugs e problemas identificados durante os testes.

**Sprint 5: Implementação das Regras de Seleção e Validação**

* Implementar as regras de seleção de validadores no Seletor.
* Desenvolver as regras de validação de transações no Validador, conforme especificado no escopo do projeto.

**Sprint 6: Finalização e Documentação**

* Finalizar a implementação de todas as funcionalidades do sistema.
* Documentar o código-fonte, incluindo comentários e explicação da lógica de funcionamento.
* Elaborar a documentação do projeto, incluindo esta descrição das sprints e a documentação oficial do sistema.

**Sprint 7: Testes Finais, Preparação para Implantação e Testes de Desempenho**

* Realizar testes finais para garantir que todas as funcionalidades estejam completas e funcionando corretamente.
* Preparar o ambiente de produção para implantação, incluindo configuração de servidores e preparação de ambientes de teste e produção.
* Garantir que a equipe esteja pronta para a implantação do sistema.
* Configurar três máquinas virtuais ou servidores para simular o acesso concorrente.
* Realizar testes de desempenho, simulando o acesso concorrente ao webservice a partir das três máquinas diferentes.
* Analisar os resultados dos testes de desempenho e identificar possíveis gargalos ou problemas de escalabilidade.
* Fazer ajustes no código ou na infraestrutura, se necessário, para otimizar o desempenho e garantir a escalabilidade do sistema.