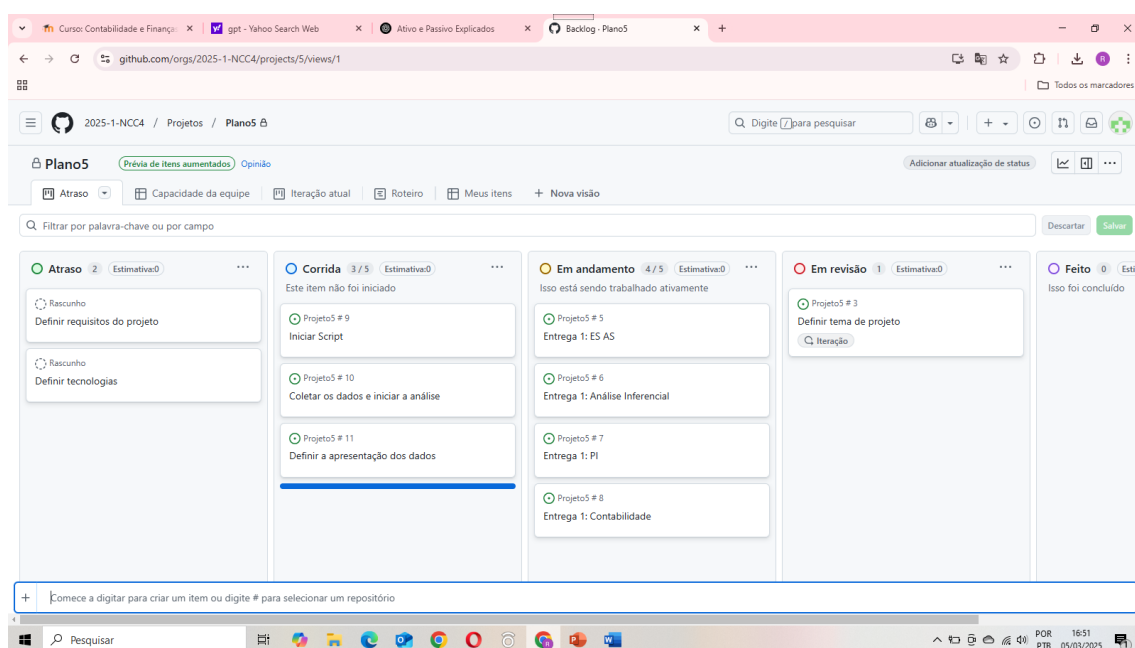


Escolhemos trabalhar com a metodologia ágil Scrum, que tem como objetivo facilitar e tornar mais leve a entrega de um determinado projeto. Optamos por essa metodologia porque, como estudamos na mesma sala quatro dias por semana, podemos realizar nossas *daily's* presencialmente. Apesar de não termos um *tech lead*, nos adaptamos bem, pois estamos todos focados e determinados a entregar um excelente resultado final para o projeto interdisciplinar.

Pensando nas entregas, decidimos dividir as responsabilidades, deixando cada participante 100% responsável por uma entrega. Dessa forma, a carga de trabalho não fica pesada para ninguém. Conforme avançamos para as próximas etapas, faremos a troca de matérias, permitindo que cada um participe de todas as fases do projeto.

Após definirmos o projeto e o público-alvo, iniciamos a divisão de tarefas utilizando o *Project* do GitHub.



Estudamos os primeiros **requisitos** necessários para iniciar o projeto e os dividimos em tarefas. Conforme o projeto for ficando mais estruturado, adicionaremos mais tarefas com um nível de complexidade maior.

Requisitos Funcionais – O que o sistema deve fazer:

RF01 – Coletar automaticamente dados econômicos do **Banco Central e do Banco Neon**.

RF02 – Monitorar e exibir indicadores como **câmbio, inflação, taxa SELIC e PIB**.

RF03 – Oferecer um **dashboard interativo** para visualização de dados em tempo real.

RF04 – Implementar **alertas automáticos** para variações bruscas de mercado.

RF05 – Integrar algoritmos de **machine learning** para prever tendências cambiais.

RF06 – Permitir **simulações de cenários financeiros** (ex: impacto da alta do dólar).

RF07 – Gerar **relatórios personalizados** para empresas.

RF08 – Enviar **notificações via e-mail, WhatsApp ou app** sobre oscilações de mercado.

RF09 – Implementar um sistema de **assinatura e controle de planos Freemium/Pago**.
RF10 – Oferecer **integração com sistemas ERP e BI** utilizados por empresas.

Requisitos não funcionais – Como o sistema deve funcionar:

RNF01 – O sistema deve ser desenvolvido em **Python**.

RNF02 – O banco de dados deve ser **SQL**, garantindo alta performance.

RNF03 – A interface do dashboard deve ser desenvolvida.

RNF04 – O sistema deve garantir **segurança e criptografia** nas transações de dados financeiros.

RNF05 – Os algoritmos de previsão devem utilizar **Scikit-learn para machine learning**.

RNF06 – A plataforma deve suportar **alta disponibilidade e escalabilidade**, pois será usada por empresas.

RNF07 – Integração via **API do Banco Neon** para dados financeiros.

RNF08 – O tempo de resposta para consultas de dados deve ser **inferior a 2 segundos**.

Requisitos de domínio – Especificidades do negócios e do setor:

RD01 – O sistema deve seguir as normas do **Banco Central do Brasil** sobre dados financeiros.

RD02 – O cálculo de riscos financeiros deve estar alinhado com **estratégias de hedge cambial**.

RD03 – O modelo preditivo deve considerar **fatores macroeconômicos**, como taxa de juros global.

RD04 – A estrutura dos relatórios deve seguir padrões utilizados por **analistas financeiros**.

RD05 – O modelo de monetização deve atender às **necessidades de pequenas e grandes empresas**.