#### **1. Introdução**

No nosso projeto, buscamos aplicar conceitos de programação e controle de versão, usando Git para gerenciar o desenvolvimento. O controle de versão é muito importante para manter o histórico de alterações no código e facilitar o trabalho em equipe.

#### **2. Objetivos da Atividade**

Os principais objetivos desta atividade foram:

* Entender e aplicar o controle de versão utilizando Git.
* Trabalhar em equipe de forma colaborativa, utilizando o GitHub como ferramenta.

#### **3. Descrição da Atividade**

Nós começamos a atividade discutindo o que cada um iria fazer e como organizaríamos o trabalho. Decidimos criar um robô chamado "RobocopMamonas", que é um robô com ações programadas para buscar e atacar inimigos, desviar de balas, e reagir ao colidir com paredes. O robô foi desenvolvido com uma lógica que permite buscar alvos, atirar, e se movimentar de acordo com a posição do inimigo.

Durante o desenvolvimento, utilizamos o Git para controlar as versões do código. Criamos um repositório no GitHub, onde cada integrante do grupo conseguiu contribuir e revisar o código dos outros. Dividimos o projeto em duas branches principais: main, onde mantivemos o código estável, e develop, onde realizamos as implementações e testes das novas funcionalidades antes de integrá-las na main.

#### **4. Estrutura do Git Utilizada**

No Git, seguimos a estrutura GitFlow para organizar nosso trabalho:

* **Repositório:** Criamos o repositório no GitHub e o organizamos com duas branches principais, main e develop
* **Branches:**
  + **main:** Guardava a versão estável e finalizada do código.
  + **develop:** Usada para implementar e testar novas funcionalidades antes de serem aprovadas e mescladas na main.
* **Commits:** Fizemos commits frequentes, cada um com uma mensagem clara, para garantir que as mudanças fossem documentadas de forma compreensível.

Os commits foram padronizados usando “Conventional Commits”. Esse padrão de mensagens de commit é amplamente usado no mercado profissional atualmente. Uma alteração nesse padrão escolhida pelo grupo foi enviar as mensagens de commit em português.

Referência: <https://www.conventionalcommits.org/en/v1.0.0/>

Resumo: https://github.com/iuricode/padroes-de-commits

* **Feature branches:** Seguindo o padrão do GitFlow, quando uma nova funcionalidade estava em desenvolvimento, uma branch específica para ela era criada. Partindo da branch “develop”.

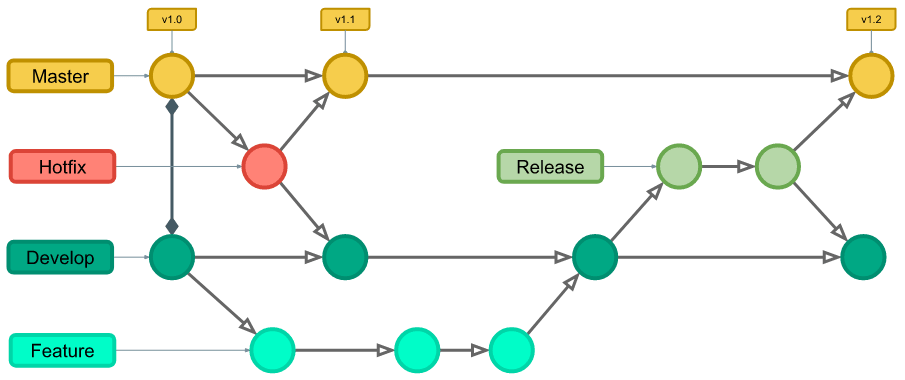


Imagem: estrutura de branches utilizada; GitFlow.

#### **5. Resultados e Aprendizados**

Essa atividade foi muito proveitosa para todos nós. Conseguimos aprender bastante sobre como programar um robô no Robocode e também sobre como usar o Git para trabalhar em equipe. Houve alguns desafios, como ajustar a lógica de movimentação do robô e fazer com que o radar funcionasse da melhor forma, mas superamos isso com testes e colaboração entre o grupo.

#### **6. Conclusão**

Em resumo, a atividade nos ajudou a entender melhor como é desenvolver um projeto de software de forma colaborativa. O uso do Robocode e do Git nos preparou para futuras experiências, seja em projetos acadêmicos ou profissionais, e nos mostrou a importância do controle de versão e da comunicação clara dentro de um time.