HC4xServer Flow

Objetivos

O software que utilizamos para versionamento é o Git e subimos no GitHub para hospedar os repositórios e arquivos e trabalhar com a equipe. Anteriormente, todos os commits, push e qualquer coisa que jogássemos no Git ou GitHub do projeto estava sendo direto na branch Main, porém como temos os ambientes de Produção, Desenvolvimento e Teste, e queremos manter a organização e versionamento do projeto corretamente, é interessante que possamos implementar um fluxo que estabeleça funções específicas para cada ambiente, sendo cada um em uma branch diferente.

O objetivo do fluxo é implementar uma estratégia e modelo para auxiliar na organização e versionamento dos códigos do projeto. Sendo assim, a organização será feita em Branches (ramificações) e o fluxo irá fluir entre elas.

O fluxo que iremos utilizar tem como base o Git Flow. Estamos trabalhando com entrega contínua, muitas mudanças de estrutura do projeto como um todo e uma equipe pequena para desenvolvimento, sendo assim, a estrutura do Git Flow poderia tornar o processo um pouco complicado e o desenvolvimento mais lento, então o Git Flow será adaptado de forma que no fluxo do HC4xServer terão as duas branches principais, a Develop (desenvolvimento) e a Main (produção), que duram para sempre; e apenas uma outra branch de suporte, a Release (teste).

Como vai funcionar?

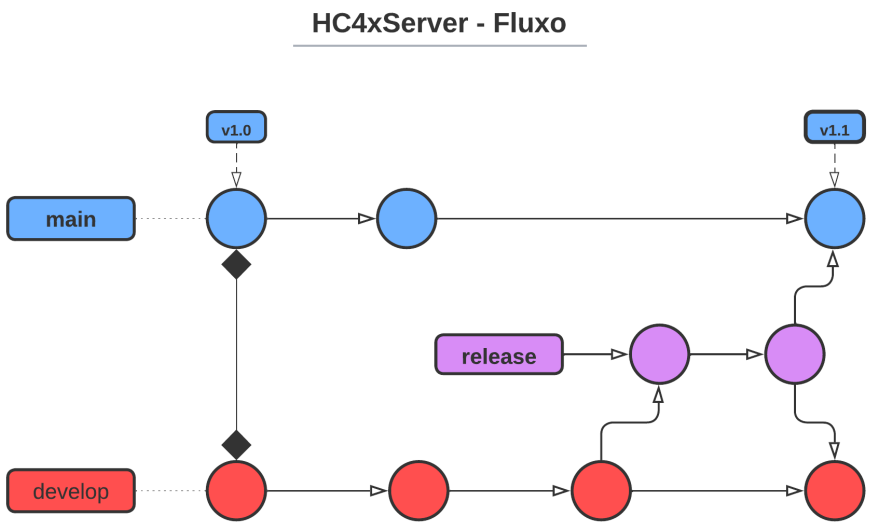
O fluxo irá funcionar trabalhando com as duas branches principais: ***main*** e ***develop***; e a branch de suporte ***release.*** Então ao invés de trabalharmos somente com a ***main*** para registrar o histórico e subir o projeto, como era antes, iremos utilizar também a ***develop.*** Na branch ***main*** vai ficar tudo o que for certo para produção e a branch ***develop*** será a branch em que iremos desenvolver e servirá como uma ramificação de integração dos recursos.

Branch ***main***:
Principal branch, aqui é onde temos todo o código de produção. Todas as novas funcionalidades que estão sendo desenvolvidas, em algum momento, serão mescladas ou associadas a ***main***. As formas de interagir com essa branch são através de uma nova Release.

Branch ***develop***:
É a branch onde fica o código do próximo deploy. Ela serve como uma linha do tempo com os últimos desenvolvimentos, isso significa que ela possui funcionalidades que ainda não foram publicadas e que posteriormente vão ser associadas com a branch ***main***.

Branch ***release***:
Uma vez que uma etapa de desenvolvimento esteja concluída, teremos em nossa Branch ***develop*** o que foi desenvolvido. Então, se quisermos ter todas essas novas funcionalidades na Branch ***main***, teremos que criar uma Branch de ***release***.
A Branch ***release*** serve como ponte para fazer o merge da ***develop*** para a ***main***. Ela funciona como ambiente de homologação e é removida após realizar os testes do merge com a Master. Caso seja encontrado algum bug e haja alguma alteração, ela também deve ser sincronizada com a Develop.

Abaixo podemos analisar a estrutura do fluxo:



O workflow foi feito utilizando o aplicativo de diagramação LucidChart.

Referências:

https://www.alura.com.br/artigos/git-flow-o-que-e-como-quando-utilizar