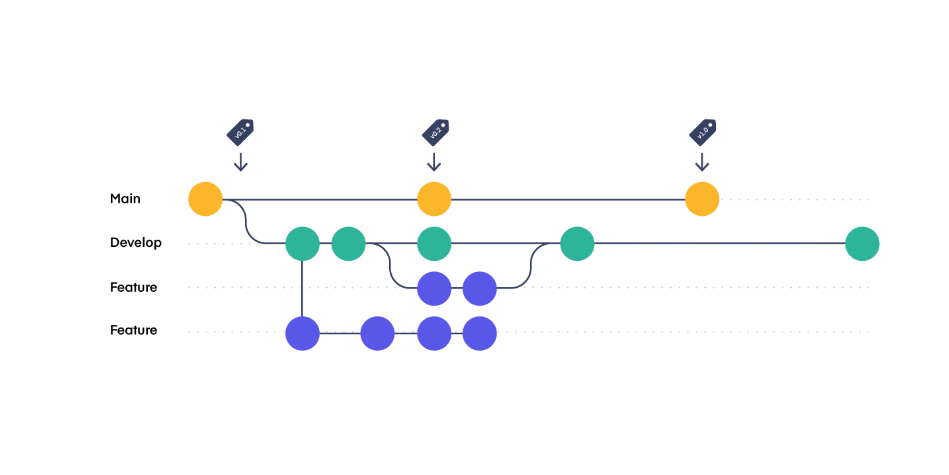
Git Branching Strategy :



The branching model depicted in the image supports team collaboration and CI/CD practices in several ways:

1. **Separation of Concerns**: By having separate branches for development and features, it allows multiple team members to work on different tasks simultaneously without interfering with each other's work. This separation helps in maintaining a clean and stable "Main" branch.
2. **Feature Isolation**: The feature branches allow individual developers or teams to work on new features or bug fixes in isolation. This means that the rest of the team can continue to develop, test, and release other features independently.
3. **Continuous Integration**: Developers can regularly merge changes back into the "Develop" branch, which can be set up to trigger automated builds and tests. This helps catch integration issues early and improves the quality of the code in the development branch.
4. **Release Management**: The "Main" branch can be used as the stable release branch that reflects the production-ready state of the application. Only well-tested and approved changes from the "Develop" branch should be merged into "Main", ensuring that the release branch always maintains a high level of stability.
5. **Continuous Deployment**: Once changes are merged into "Main", they can be automatically deployed to production systems, providing a seamless path from development to deployment. This is especially useful in a CI/CD pipeline where you want to automate the release process as much as possible.
6. **Code Reviews and Quality Control**: Before merging feature branches into "Develop" or "Main", code reviews can be conducted. This process helps improve code quality and share knowledge within the team.
7. **Rollback and Hotfixes**: If any issues are identified in the "Main" branch after a release, the team can easily create a hotfix branch from "Main" to address the issues. It also allows quick rollbacks to previous stable states if necessary.

Overall, the branching strategy facilitates a structured workflow where development, testing, and releases can happen in a controlled and collaborative manner, supporting both team collaboration and the principles of CI/CD.