

# GitHub

## Grupo T02-11

- 222037610 - Gabriel Lima Da Silva
- 222037620 - Wanjo Christopher Paraizo Escobar
- 222006267 - Karoline Luz da Conceição
- 222006113 - João Marcos Moraes de Andrade

## 1. CENÁRIO

O GitHub é uma plataforma de desenvolvimento colaborativo que permite aos desenvolvedores compartilhar e colaborar em projetos de software. Com ele, podemos aproveitar várias ferramentas, como **usuário**, **repositório**, **commits** e **issues**, que desempenham papéis cruciais em todas as fases do ciclo de vida de desenvolvimento de um software.

O **usuário** no contexto do GitHub desempenha um papel fundamental no ambiente de desenvolvimento colaborativo. Ele é a porta de entrada para a vasta gama de ferramentas e recursos disponíveis nesta plataforma. Ao se cadastrar como usuário, terá que atribuir seu **nome**, **email** e **senha** para ter acesso à plataforma. Além de poder **cadastrar**, **remover**, **buscar** e **editar** um repositório terá que **listar** todos os usuários cadastrados.

O **repositório** do GitHub é o local central onde armazenamos todo o código-fonte, documentação e recursos associados ao nosso software. É o ponto focal da colaboração com atributos iniciais como seu **nome** e **data de criação**. Dentro do repositório é possível **adicionar**, **remover**, **editar**, **buscar** e **listar repositórios**.

**Commits** representam alterações individuais feitas ao código-fonte, permitindo um histórico detalhado de todas as edições. Com cada commit, documentamos o que foi alterado, por que foi alterado e por quem foi alterado. Seu cadastro depende principalmente de seu **nome**, **identificação**, **descrição**, **usuário** e **data de envio**.

Além disso, o GitHub fornece uma maneira eficaz de gerenciar tarefas e melhorias em nosso projeto através das issues. As **issues** são usadas para rastrear problemas, solicitações de recursos e tarefas pendentes. Elas servem como uma plataforma para discussão, colaboração e acompanhamento do progresso. Ao serem adicionadas depende de informações como **nome**, **identificação**, **descrição**, **status** e **usuário**.

Por fim, o gerenciamento de commit e issue ocorre por uma classe abstrata **Item Controle**, onde é possível **adicionar**, **remover**, **editar**, **buscar** e **listar** ambas ferramentas

## 2. LISTA DE OBJETOS

<i>Objeto</i>	<i>Atributos</i>	<i>Métodos</i>
Usuario	nome : String email : String senha : String repositorios: List<Repositorio>	gets-e-sets() : void Usuario( <i>nome : String, email : String, senha : String</i> ) addRepositorio( <i>repositorio : Repositorio</i> ): boolean deletarRepositorio ( <i>repositorio : Repositorio</i> ) : boolean editarRepositorio( <i>repositorio : Repositorio</i> ) : void buscarRepositorio ( <i>nome : String</i> ) : List<Repositorio> toString(): String
Repositorio	nome : String dtCriacao : String itensControles: List<ItemControle>	Repositorio( <i>nome : String, dtCriacao : String, usuario : Usuario</i> ) gets-e-sets() : void addItemControle( <i>itemControle: ItemControle</i> ): boolean deletarItemControle( <i>itemControle: ItemControle</i> ): boolean buscarItemControle( <i>itemControle: ItemControle</i> ): List<ItemControle> editarItemControle ( <i>itemControle: ItemControle</i> ) : void toString(): String
<<abstract>> ItemControle	nome : String id : int descricao : String commits: List<Commit> issues: List<Issue>	ItemControle( <i>nome : String, id : int, descricao : String</i> ) gets-e-sets() : void addCommit( <i>commit: Commit</i> ): boolean deletarCommit( <i>commit : Commit</i> ): boolean buscarCommit( <i>nome : String</i> ): List<Commit>

		editarCommit(commit : Commit): void addIssue(issue : Issue): boolean deletarIssue(issue : Issue): boolean buscarIssue(nome : String): List<Issue> editarIssue(issue : Issue): void toString(): String
Commit	dtEnvio : String	Commit(nome : String, id : int, descricao : String, autor : Usuario, dtEnvio : String) gets-e-sets() : void toString(): String
Issue	status : String	Issue(nome : String, id : int, descricao : String, status : String, responsavel : Usuario) gets-e-sets() : void toString(): String