Curso: Introdução a HTML, CSS, JavaScript e Experiência do Usuário

ENVIO DE ATIVIDADES:

GITHUB CLASSROOM

UM GUIA PRÁTICO

EXPRESSO.DEV

CEFET-MG

ENVIANDO TAREFAS:

GITHUB CLASSROOM

UM GUIA PRÁTICO

“A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original”.
Albert Einstein

EXPRESSO.DEV

CEFET-MG

ENVIANDO TAREFAS:

GITHUB CLASSROOM
UM GUIA PRÁTICO

“Bem-vindo ao nosso e-book, criado para auxiliar nossos alunos no uso do GitHub Classroom! Este guia foi desenvolvido para facilitar o processo de entrega de tarefas durante o curso de Programação Web, garantindo que você aproveite ao máximo essa ferramenta.

O GitHub Classroom funciona como um ambiente virtual de aprendizado, permitindo a criação e gerenciamento de repositórios para cada atividade. Além de ajudar no envio das tarefas, ele também introduz conceitos essenciais do GitHub, como repositórios, commits e versionamento de código, que são fundamentais para qualquer desenvolvedor.

Neste e-book, explicamos passo a passo como acessar sua sala, aceitar tarefas, trabalhar nos repositórios e enviar suas entregas corretamente. Se você nunca utilizou o GitHub antes, não se preocupe! Nosso objetivo é tornar esse processo simples e intuitivo, para que você possa focar no aprendizado da programação.

Sinta-se à vontade para compartilhar este material com colegas e revisitar sempre que precisar. Vamos juntos explorar o GitHub e aprimorar suas habilidades no desenvolvimento web! 🚀”

EXPRESSO.DEV

CEFET-MG

INFORMAÇÕES DO CURSO

Instagram - @expresso.dev.cefetmg

Inscrições e mais informações ℹ️

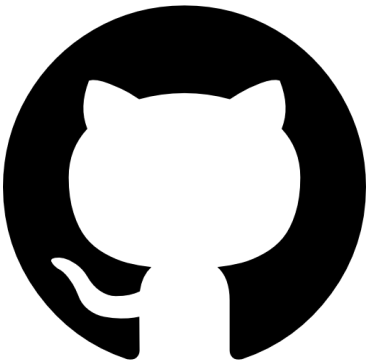
Dúvidas frequentes ❓

EXPRESSO DEV

EXPRESSO.DEV

CEFET-MG

ÍNDICE



EXPRESSO.DEV

CEFET-MG

1. INTRODUÇÃO

GITHUB CLASSROOM:
O QUE É?

Ao longo do curso, vocês terão diversas atividades práticas para consolidar o que estão aprendendo em aula. Mas como serão feitas as entregas?

É aqui que entra o **GitHub Classroom**! Essa ferramenta pode ser comparada a plataformas como o Google Sala de Aula, pois permite organizar tarefas e acompanhar o progresso da turma. No entanto, o GitHub Classroom é especialmente projetado para programação, oferecendo uma estrutura completa para versionamento de código, envio de projetos e feedbacks de forma dinâmica.

O GitHub é uma das plataformas mais utilizadas no mundo para hospedagem e gerenciamento de projetos de software, sendo amplamente adotado por desenvolvedores, empresas e pesquisadores. Além de facilitar a colaboração, ele também é uma excelente ferramenta para construir e divulgar portfólios profissionais.

Ao utilizarmos o GitHub Classroom para as entregas do curso, vocês não apenas aprenderão a enviar suas tarefas, mas também terão um primeiro contato prático com conceitos essenciais do **Git** e do **GitHub**, como **versionamento de código, commits e repositórios**. Essa experiência será valiosa para o futuro, seja para projetos acadêmicos, desenvolvimento profissional ou colaboração em equipe.

Antes de tudo, é importante realçar a **diferença entre Git e Github**: Git é um sistema de versionamento de código. Ou seja, com ele é possivel verificar qual versão o código está, realizar réplicas organizadas dependendo do seu estado (desenvolvimento ou publicada, por exemplo). O GitHub é um repositório que hospeda repositórios Git.

Nos próximos tópicos, vamos guiá-los pelo processo de configuração, envio de atividades e utilização dos principais recursos da plataforma. Vamos começar? 🚀

2. ACESSANDO O GITHUB CLASSROOM

ANTES DE COMEÇAR:
REQUISITOS NECESSÁRIOS

Para utilizar o GitHub Classroom, **você precisará de uma conta no GitHub**. Se ainda não tiver uma, **crie sua conta antecipadamente por aqui**.

Além disso, é importante ter um nome de usuário apropriado, pois ele será usado para identificar suas entregas no curso. Evite apelidos muito informais ou difíceis de reconhecer. Atenção, que muitas empresas pedem a conta de seu Github durante o processo seletivo para analisar os projetos no qual você já participou.

Nos próximos passos, vamos guiá-lo pelo processo de acesso e configuração do ambiente.

PRIMEIROS PASSOS:
ACESSANDO A SALA

Para acessar a sala virtual onde serão disponibilizadas as tarefas, basta **clicar no link abaixo**. Esse link corresponde ao convite para a realização da primeira atividade prática.

ACESSO: DINOPRÁTICA & SALA GITHUB CLASSROOM

ETAPAS:

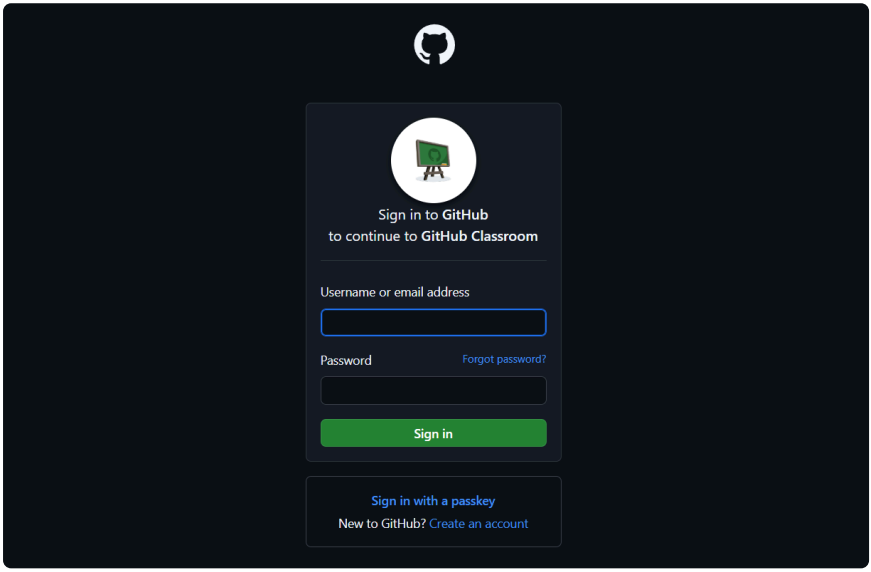
Aqui temos um resumo do passo a passo necessário para acessar a sala. Ele será detalhado em sequência.

1. Acesse o link de convite.
2. Faça login na sua conta do GitHub (ou crie uma, caso ainda não tenha).
3. Você verá uma lista de alunos. Selecione o seu nome para vincular sua conta corretamente.
4. Aceite a tarefa clicando na opção correspondente.
5. Após essa etapa, seu repositório será criado automaticamente, já contendo os arquivos base da atividade. Você poderá editar esses arquivos conforme necessário.

2. ACESSANDO O GITHUB CLASSROOM

PASSO A PASSO:
ACESSO DETALHADO

1. Acesse o link do convite e efetue login em sua conta do github.

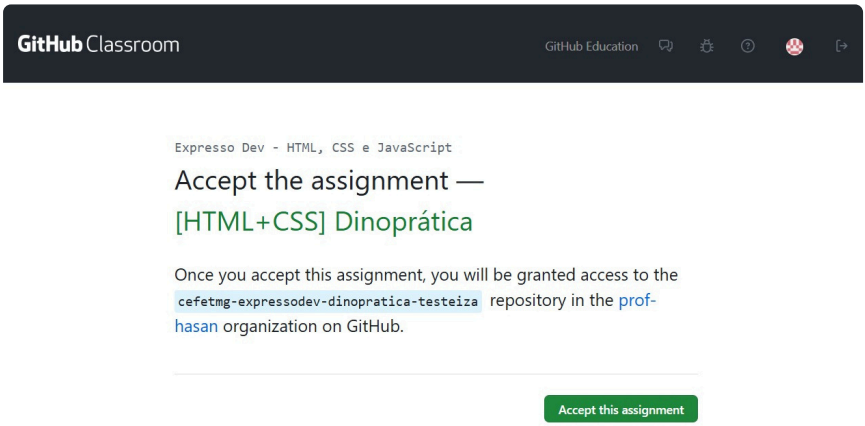


2. Selecione o seu nome na lista de alunos para vincular sua conta corretamente.

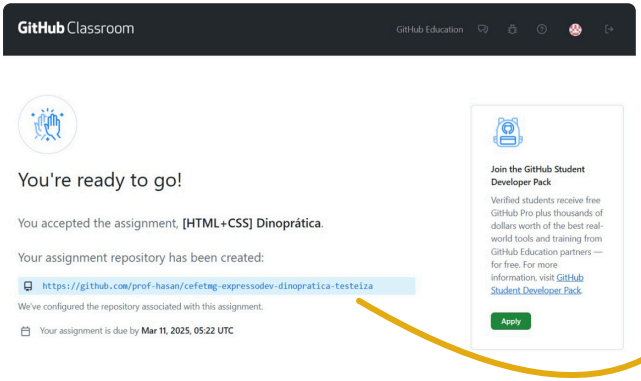
2. ACESSANDO O GITHUB CLASSROOM

PASSO A PASSO:
ACESSO DETALHADO

3. Aceite a tarefa clicando na opção correspondente.



4. Após essa etapa, seu repositório será criado automaticamente, já contendo os arquivos base da atividade.



Você pode acessá-lo através do link destacado.

Repare que o nome do repositório será um prefixo padrão característico da prática, seguido do seu **username**.

3. AMBIENTE & CONFIGURAÇÕES

REPOSITÓRIOS
UM OLHAR MAIS DETALHADO

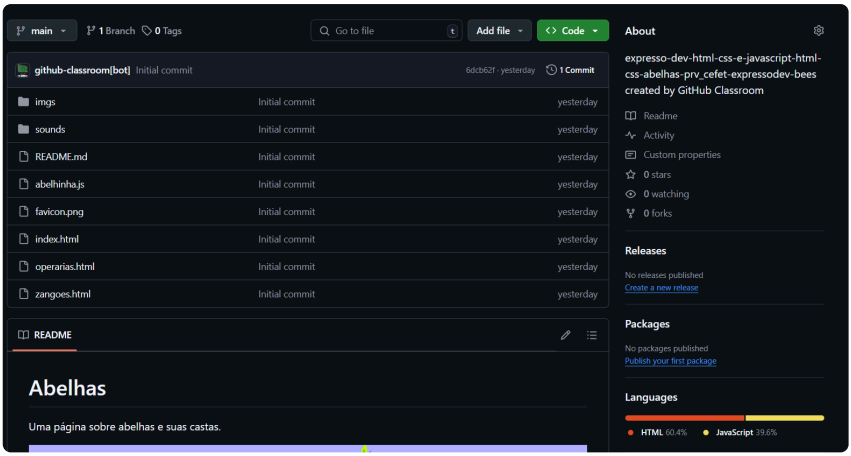
O repositório funciona como uma "pasta" virtual, acessível tanto para você quanto para os professores. É nele que você deve adicionar e modificar os arquivos conforme as demandas da atividade.

Se precisar de ajuda durante o processo, consulte os próximos tópicos deste guia ou entre em contato com o professor. 😊

ESTRUTURA DO REPOSITÓRIO

Ao acessar o repositório, você verá uma lista de arquivos e pastas. Alguns dos mais comuns incluem:

- **README.md:** Se presente, este arquivo geralmente contém instruções importantes sobre a atividade. ***Leia sempre antes de começar!***
- **Pastas e arquivos-base:** Esses arquivos já foram configurados pelos professores e devem ser utilizados para a realização da tarefa. ***Evite excluí-los ou movê-los sem necessidade.***



Um exemplo prático: Repositório da prática de Abelhas.

4. AÇÕES & ENTREGAS

REPOSITÓRIOS

PRINCIPAIS AÇÕES

Para realizar as entregas, será necessário editar os arquivos já existentes no repositório ou adicionar novos, conforme a atividade proposta. Esse processo pode ser feito de duas formas principais:

- Diretamente pelo navegador, utilizando o **GitHub Web**.
- Através de aplicativos a serem instalados no seu computador que permitem gerenciar os arquivos. Iremos apresentar o **GitHub Desktop**, mas o princípio é o mesmo caso queira usar outro, como o **Gitkraken** e até o **VisualCode** ou o terminal do Windows/Linux.

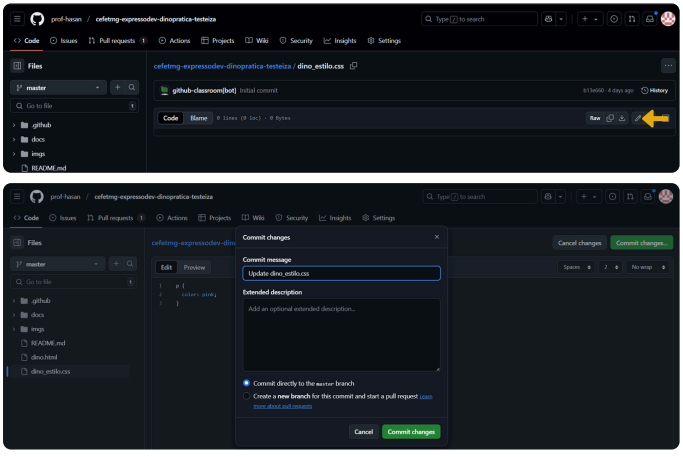
A seguir, vamos explorar como cada um desses métodos funciona.

É possível (e até recomendável) realizar entregas parciais, enviando periodicamente seu código para o GitHub. Essa é uma boa prática de uso do repositório, pois garante um backup do trabalho realizado e mantém um histórico de edições ao longo do tempo.

GITHUB WEB

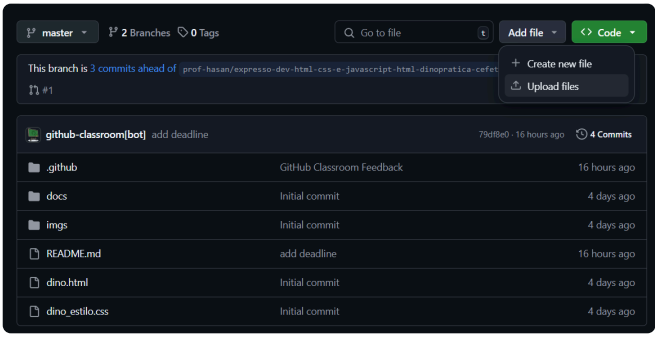
Se você prefere realizar todas as alterações diretamente no navegador, o GitHub Web permite editar arquivos, criar novos e enviar suas mudanças sem precisar instalar nada.

- Acesse o repositório da sua atividade no GitHub.
- Clique no arquivo que deseja modificar e pressione o ícone de lápis 📝 para editá-lo.
- Após finalizar as mudanças, desça clique no botão "commit" e escreva uma mensagem curta descrevendo a alteração (exemplo: "Adicionando estilos ao cabeçalho"), e clique em Commit changes.

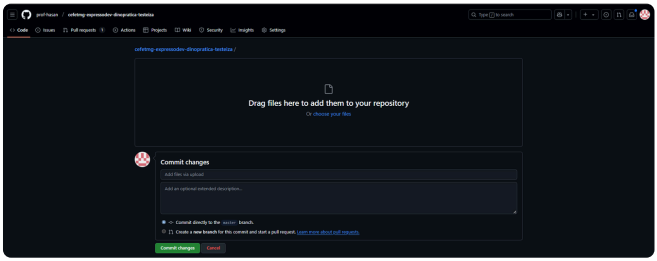


4. AÇÕES & ENTREGAS

Se precisar adicionar um novo arquivo, clique em Add file > Create new file, nomeie o arquivo corretamente e edite o conteúdo antes de salvar.



Depois, basta selecionar os arquivos de interesse ou arrastá-los para região especificada.



⚠ CUIDADOS IMPORTANTES

1. Não envie pastas compactadas (.zip, .rar, etc.):
O GitHub não lida bem com arquivos compactados, e os professores precisam acessar os arquivos individualmente. Portanto, sempre envie os arquivos no formato original, sem compactá-los antes.

2. Mantenha a estrutura de arquivos original:
Se a atividade forneceu um repositório com pastas organizadas, não altere essa estrutura. Adicione ou edite os arquivos dentro dos diretórios indicados, sem mover, renomear ou excluir pastas essenciais.

3. Certifique-se de que os arquivos foram realmente enviados
Após fazer o upload, confira se todos os arquivos apareceram corretamente no repositório e se estão no local certo.

Manter arquivos duplicados (como versões zipadas ou cópias em outras pastas) no Git **é uma má prática**. O Git já armazena eficientemente o histórico de edições e gerencia versões de arquivos de forma automática. A duplicação de arquivos pode gerar confusão sobre qual versão está atualizada, além de dificultar ou até impossibilitar a gestão de conflitos de versão código entre os desenvolvedores.

4. AÇÕES & ENTREGAS

EDIÇÃO LOCAL & GITHUB DESKTOP

Embora seja possível editar arquivos diretamente pelo navegador ou fazer uploads manuais, esse método pode se tornar pouco prático para projetos maiores. Por isso, recomendamos que você edite os arquivos localmente, utilizando o **GitHub Desktop**, que facilita o gerenciamento das alterações e mantém um fluxo de trabalho mais organizado.

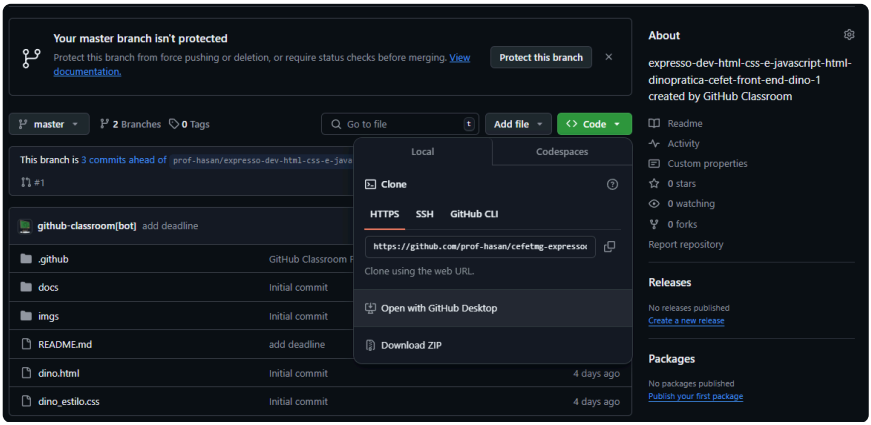
PASSO A PASSO

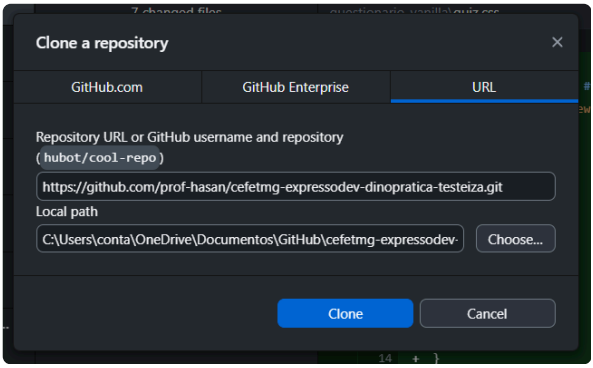
1.Baixe e instale o GitHub Desktop

- Caso ainda não tenha o programa, faça o download em *desktop.github.com* e siga as instruções de instalação.

2. Clone o repositório da atividade

- Acesse o repositório da tarefa no GitHub Web.
- Clique no botão Code e selecione Open with GitHub Desktop.



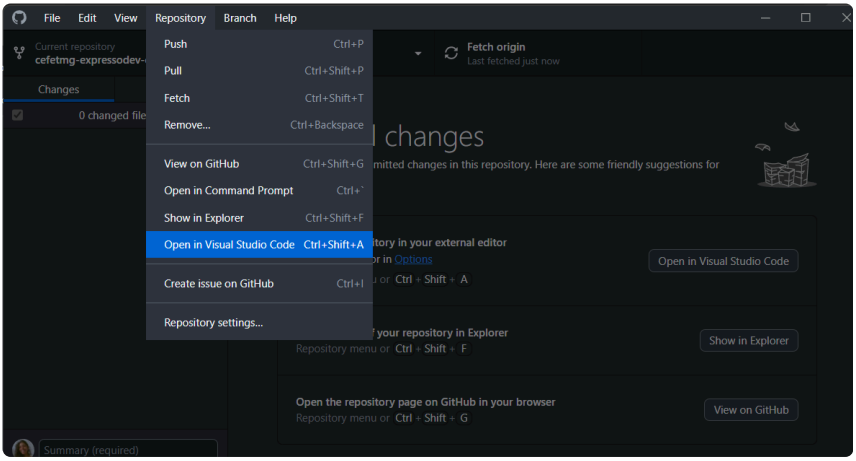


- Escolha um local no seu computador para salvar o repositório e clique em Clone.

4. AÇÕES & ENTREGAS

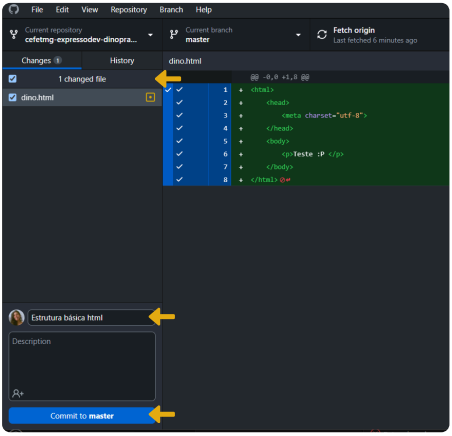
3. Edite os arquivos localmente

- Abra os arquivos no editor de código de sua preferência (como VS Code, Sublime Text ou outro) e faça as modificações necessárias conforme a atividade.



4. Salve e registre suas alterações (Commit)

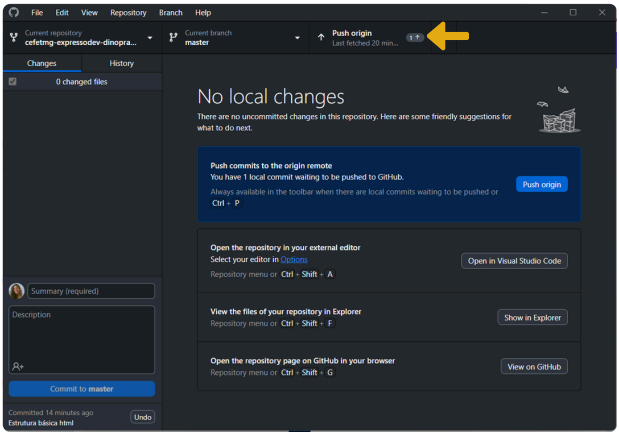
- Após editar e salvar os arquivos, abra o GitHub Desktop.
- Na aba Changes, verifique os arquivos modificados.
- No campo "Summary", escreva uma breve descrição das mudanças (exemplo: "Adicionei estilos ao cabeçalho").
- Clique em Commit to main para salvar as alterações no repositório local.



4. AÇÕES & ENTREGAS

5. Envie as alterações para o GitHub (Push)

- Após realizar o commit, clique no botão Push origin para sincronizar suas mudanças com o repositório no GitHub.



Após o envio, recomendamos que você acesse o repositório pelo GitHub Web para conferir se todas as alterações foram registradas corretamente e se os arquivos estão no local certo.

E O QUE É UM COMMIT?

Um **commit** no Git representa um **registro de mudanças feitas nos arquivos de um repositório**. Sempre que você modifica um arquivo e deseja salvar essa alteração de forma controlada, é necessário fazer um commit.

Cada commit inclui uma mensagem descritiva, que ajuda a entender quais mudanças foram feitas. Por exemplo, ao corrigir um erro no código, a mensagem do commit pode ser: *"Corrigido erro na validação do formulário"*.

O commit não envia as alterações para o GitHub automaticamente. Para isso, é necessário fazer o push, que envia os commits para o repositório online.

Manter arquivos duplicados (como versões zipadas ou cópias em outras pastas) no Git **é uma má prática**. O Git já armazena eficientemente o histórico de edições e gerencia versões de arquivos de forma automática. A duplicação de arquivos pode gerar confusão sobre qual versão está atualizada, além de dificultar ou até impossibilitar a gestão de conflitos de versão código entre os desenvolvedores.

5. FEEDBACKS & MELHORIAS

REPOSITÓRIOS

VISUALIZAÇÃO DE FEEDBACK

Após o envio das tarefas, os professores poderão deixar comentários com feedbacks e sugestões de melhoria. Para visualizá-los, acesse a aba "Pull Requests" no repositório da atividade. Caso haja feedback disponível, você encontrará uma seção específica para isso.

Nessa área, podem estar comentários de professores e monitores sobre seu código, apontando ajustes ou sugestões, mas nem sempre haverá mensagens. Por isso, é importante conferir periodicamente se há novas orientações sobre sua entrega.