Versionamento de código com Git e Git Hub

Introduzir ao versionamento de código com Git e Git hub

Conhecer as ferramentas Instalar, configurar e Autenticar, Primeiros Passos com Git e Git Hub Dicas e Materiais de Apoio

Pré-requisitos

* Computador com acesso à internet
* Vontade de aprender

Percuso

Visão geral do curso e ferramentas

Instalação, configuração e Autenticação

Primeiros passos com Git e Git Hub

Dicas de Materiais de Apoio

O que é versionamento de código

Versionamento é em programação, que a cada alteração/versão de código ser gerado uma nova versão.

Sistemas de Controle de versão

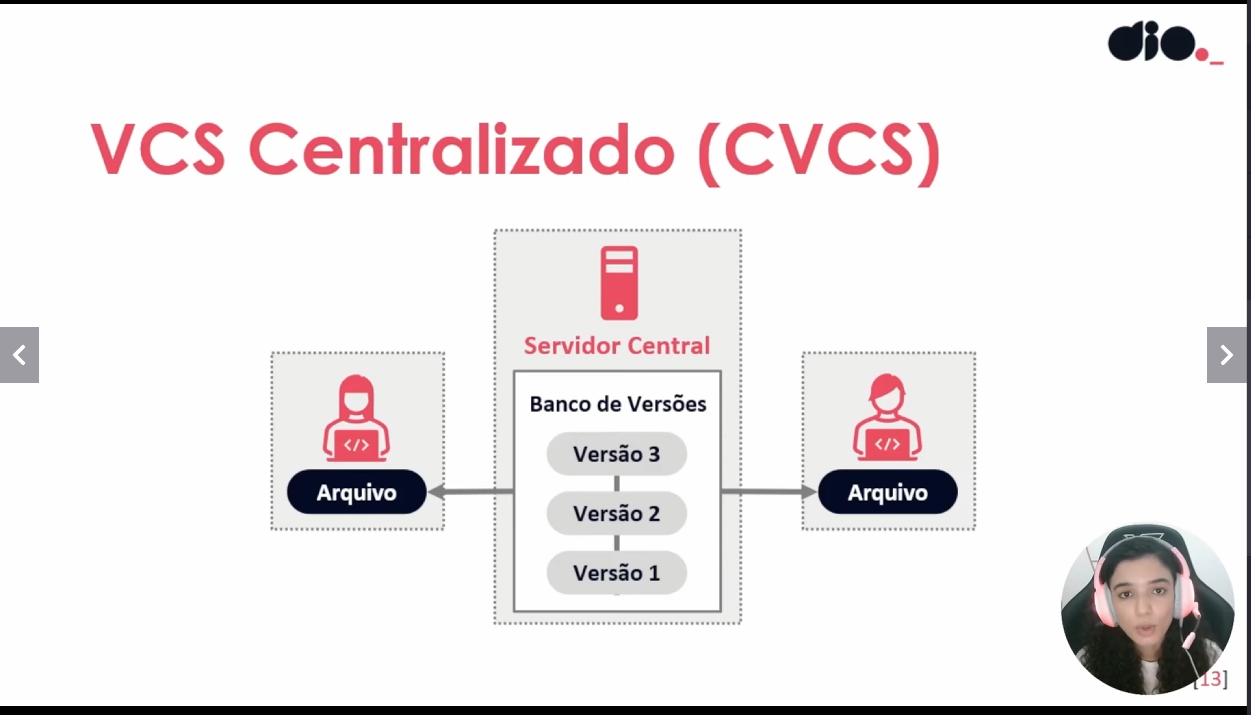
Controlam as versões de um arquivo ao longo do tempo.

* Registra o histórico de atualizações de um arquivo;
* Gerencia quais foram as alterações, a data, autor, etc;
* Organização, controle e segurança;

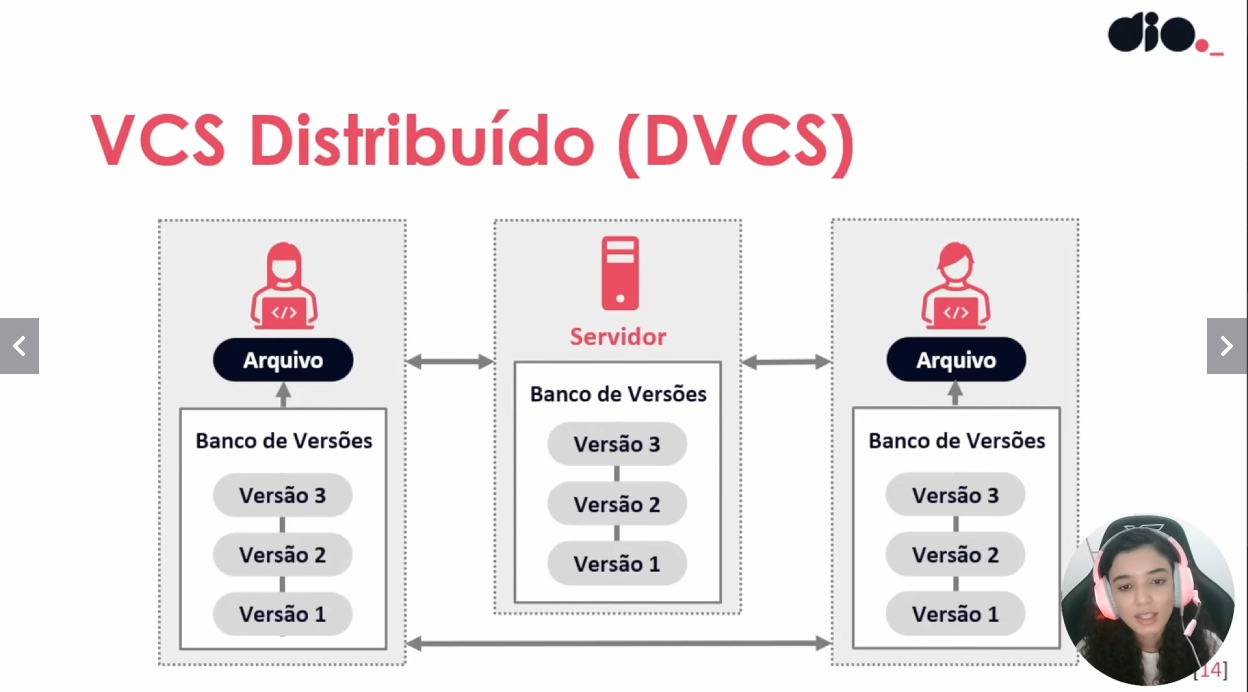
Tipos de Sistemas de Controle de versão

Dentre os sistemas de controle de versão (VCS), temos:

* VCS Centralizado (CVCS); -> CVS, Subversion.
* VCS Distribuídos (DVCS); - > Git, Mercurial



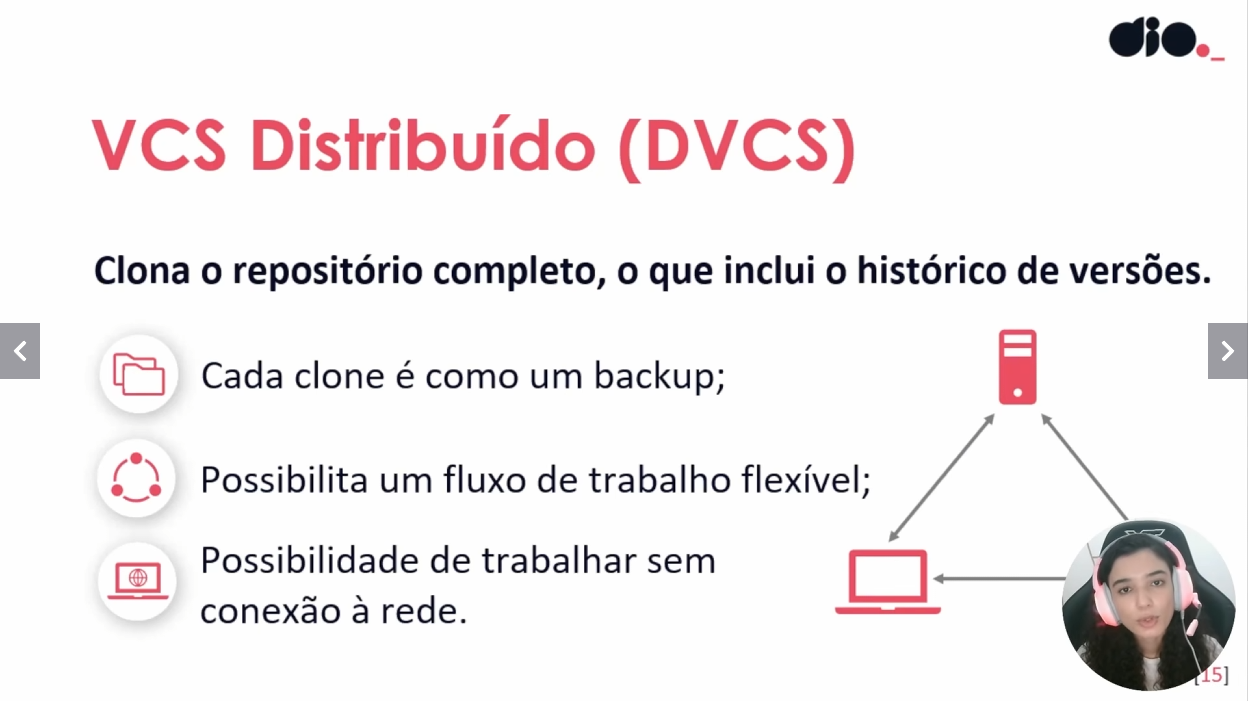
Nesse CVCS, esse modelo é caracterizado por conter apenas um servidor que conterá todos os arquivos responsáveis pelo controle de versão, e sua principal desvantagem, em caso de queda ou problemas com o servidor central você fica impossibilitado de salvar ou alterar alguma versão, e para casos de corrompimento de arquivos ou perda de dados, você perdera o projeto e para solucionar isso sugiram então os VCS Distribuídos (DVCS)



Nesse modelo se caracteriza principalmente que nesse modelo cada repositório ou cada “banco de versão”, sendo duplicado localmente, ou seja, os participantes possuem uma cópia do que está no servidor principal.

O VCS Distribuídos (DVCS), clona o repositório completo, o que inclui o histórico de versões.

* Cada clone é como um backup;
* Possibilita um fluxo de trabalho flexível;
* Possibilidade de trabalhar sem conexão à rede;



O que é Git

Sistema de controle de versão Distribuído.

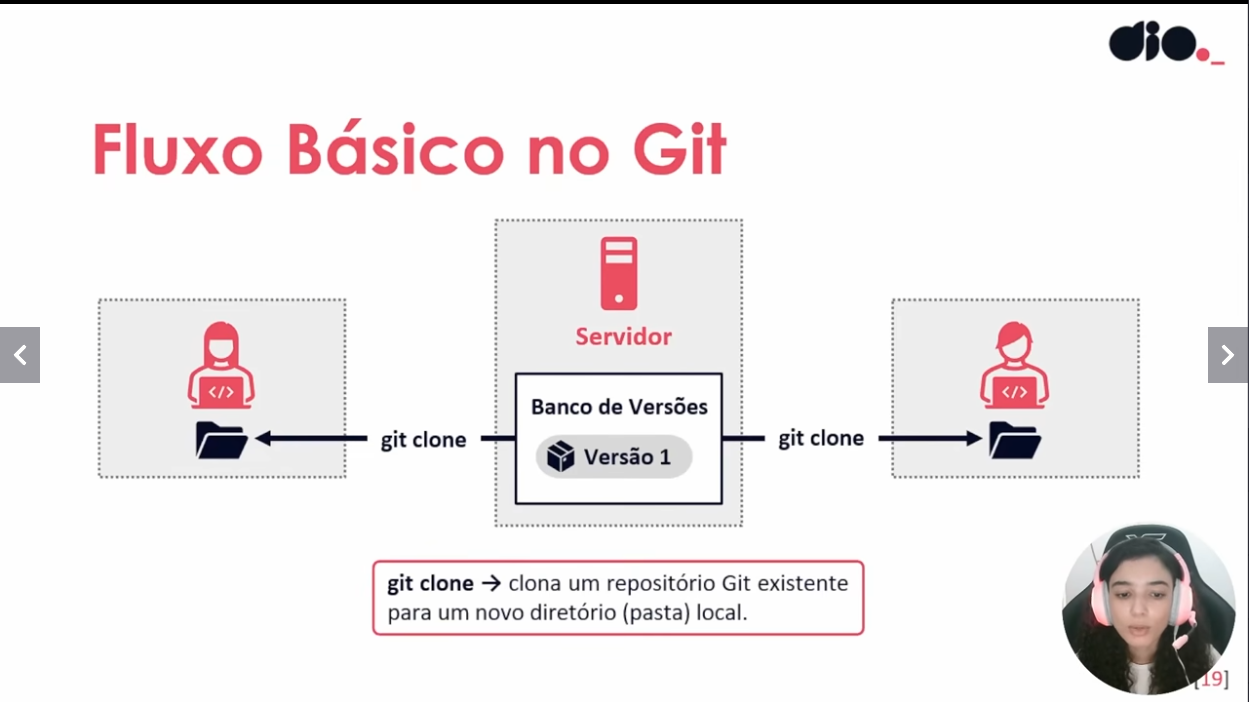
* Gratuito e Open Source (Código Aberto);
* Ramificação (Branching ) e fusões (merging) eficientes);
* Leve e rápido

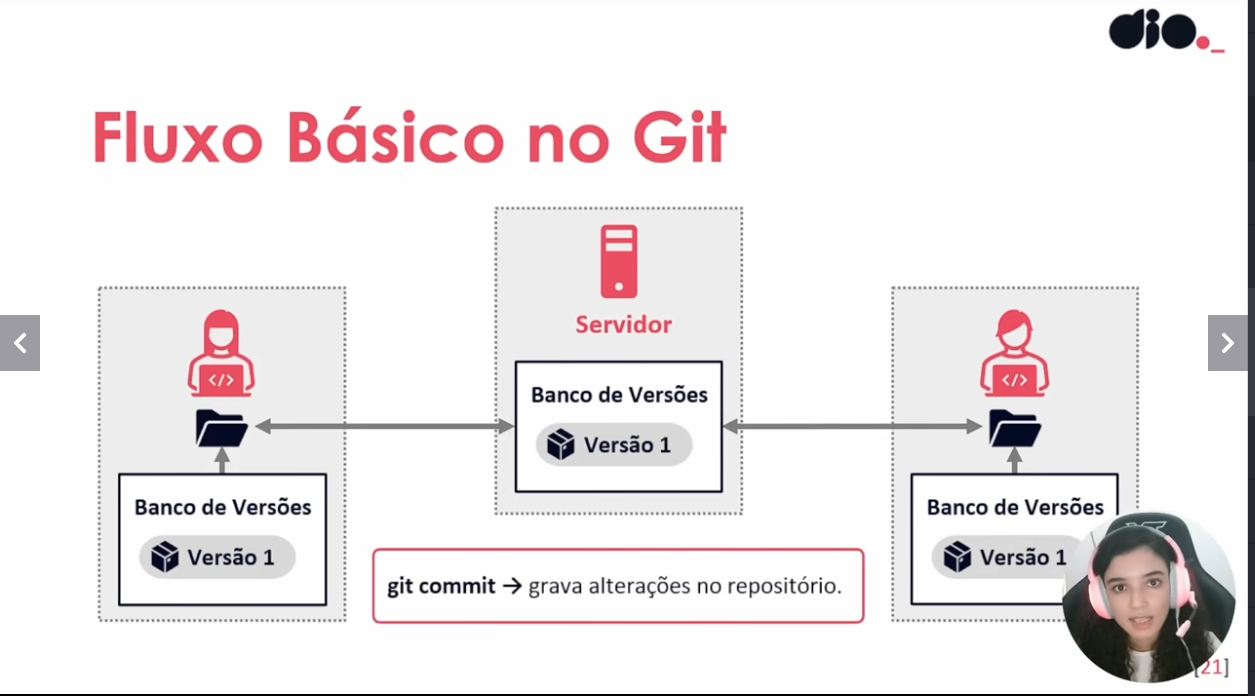
<https://git-scm.com/>

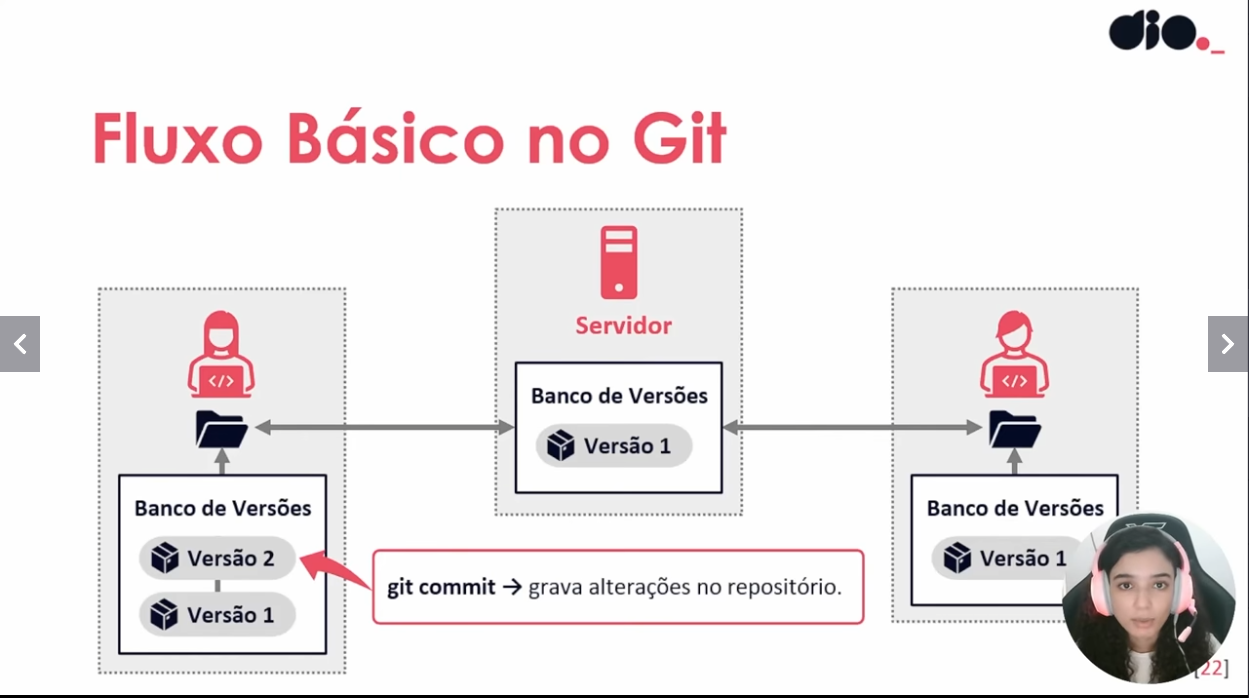
Breve histórico do Git

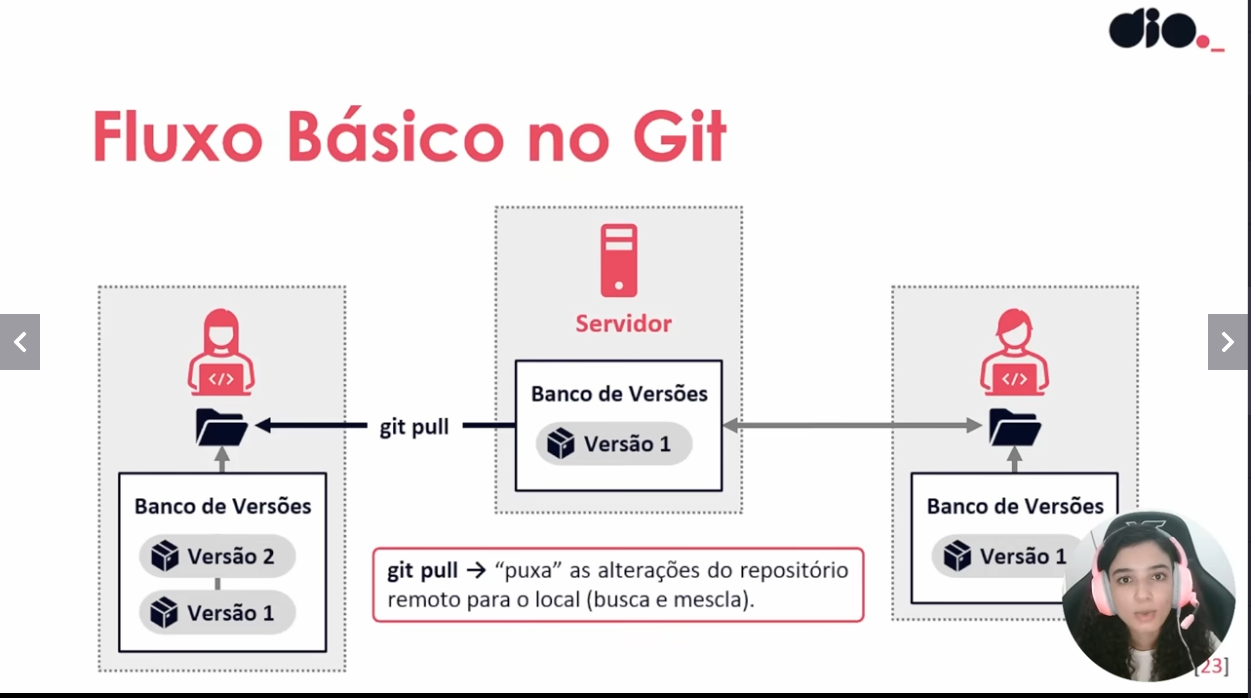
2002 -> O projeto do núcleo (Kernel) do Linux, que é open source, começa a utilizar o bitkeeper, um DVCS proprietário;

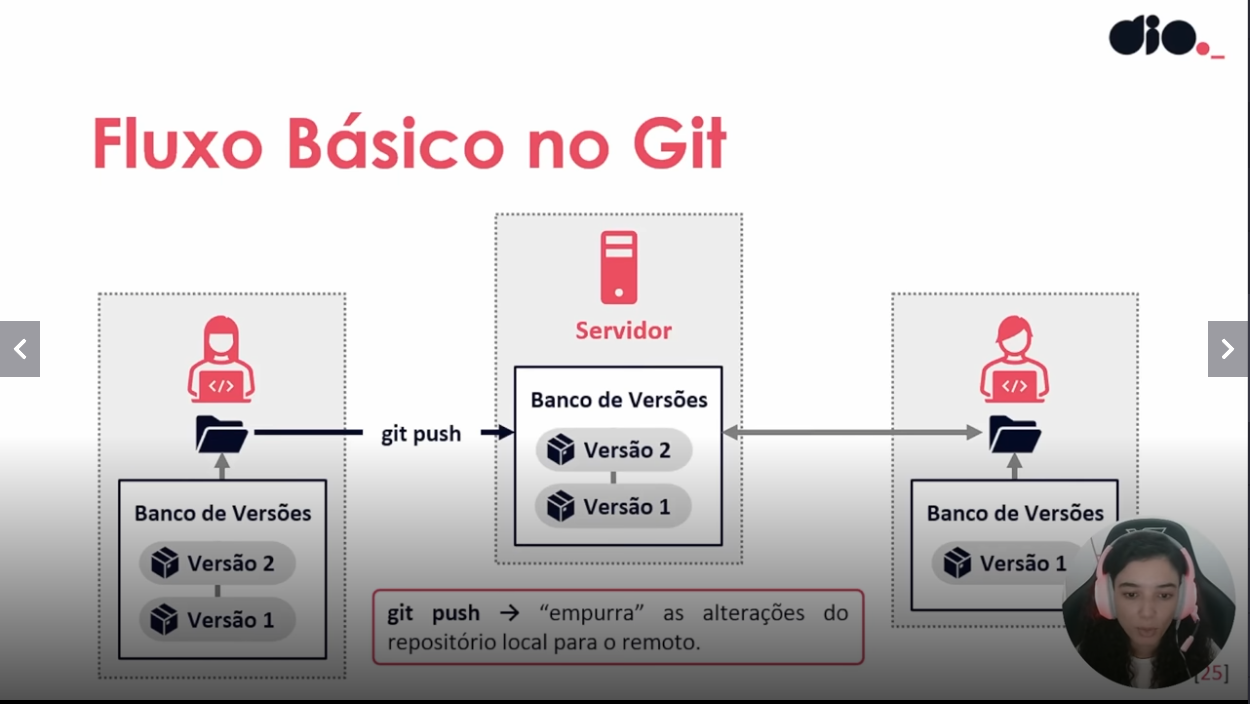
2005 -> Após conflitos com a comunidade, o BitKeeper recinde a licença gratuita. O que leva a Linux Torvalds, o criado do Linux, e sua equipe desenvolverem sua própria ferramenta, o Git.











O que é GitHub?

Plataforma de hospedagem de código para controle de versão com Git, e colaboração

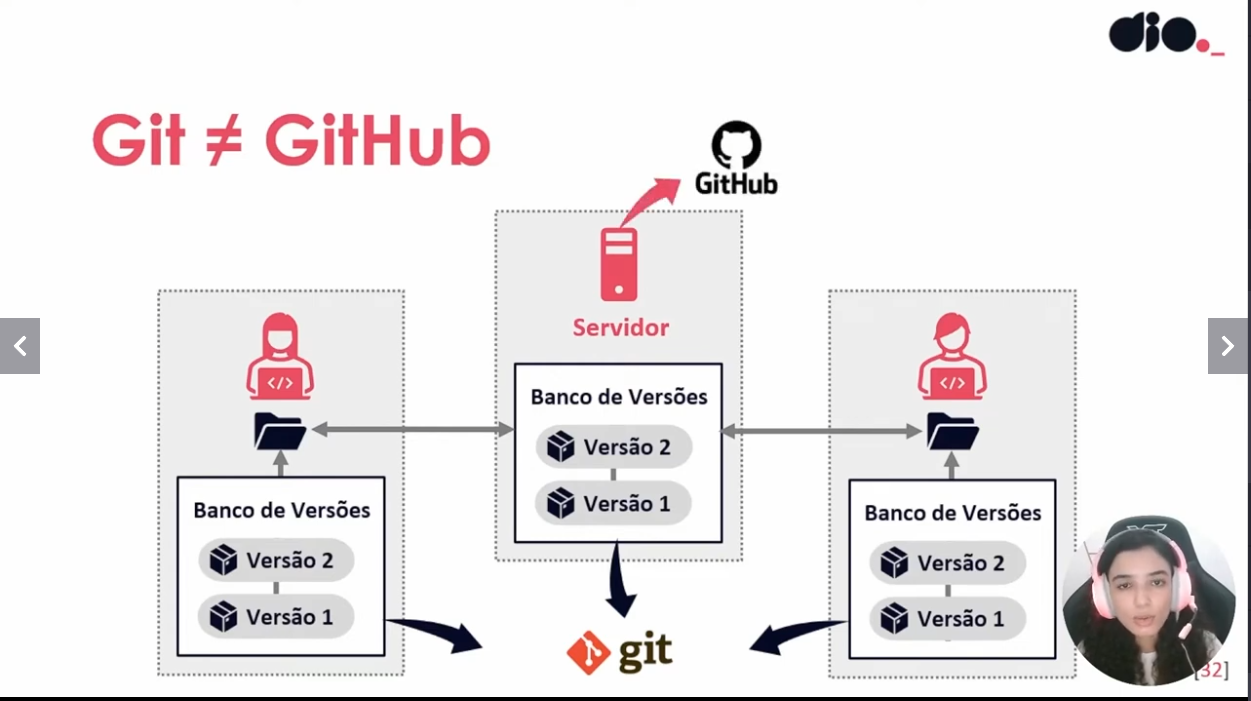
* Comunidade ativa;
* Utilizado mundialmente;
* Mascote “Octocat”.



Breve Histórico do GitHub

2008 -> Desenvolvido por Chris Wanstrath, J. Hyett, Tom preston-Werner e Scott Chacon.

2018 -> Vítima de um dos maiores ataques de DDoS (Ataque distribuído de negação de serviço); comprado pela Microsoft Corporation por US $ 7,5 Bilhões.



Autenticando via Token

Após as configurações padrões do git e github caso deseje realizar a configuração via token que foi gerado no git hub

Siga os seguintes passos

1 gerar o token no git hub

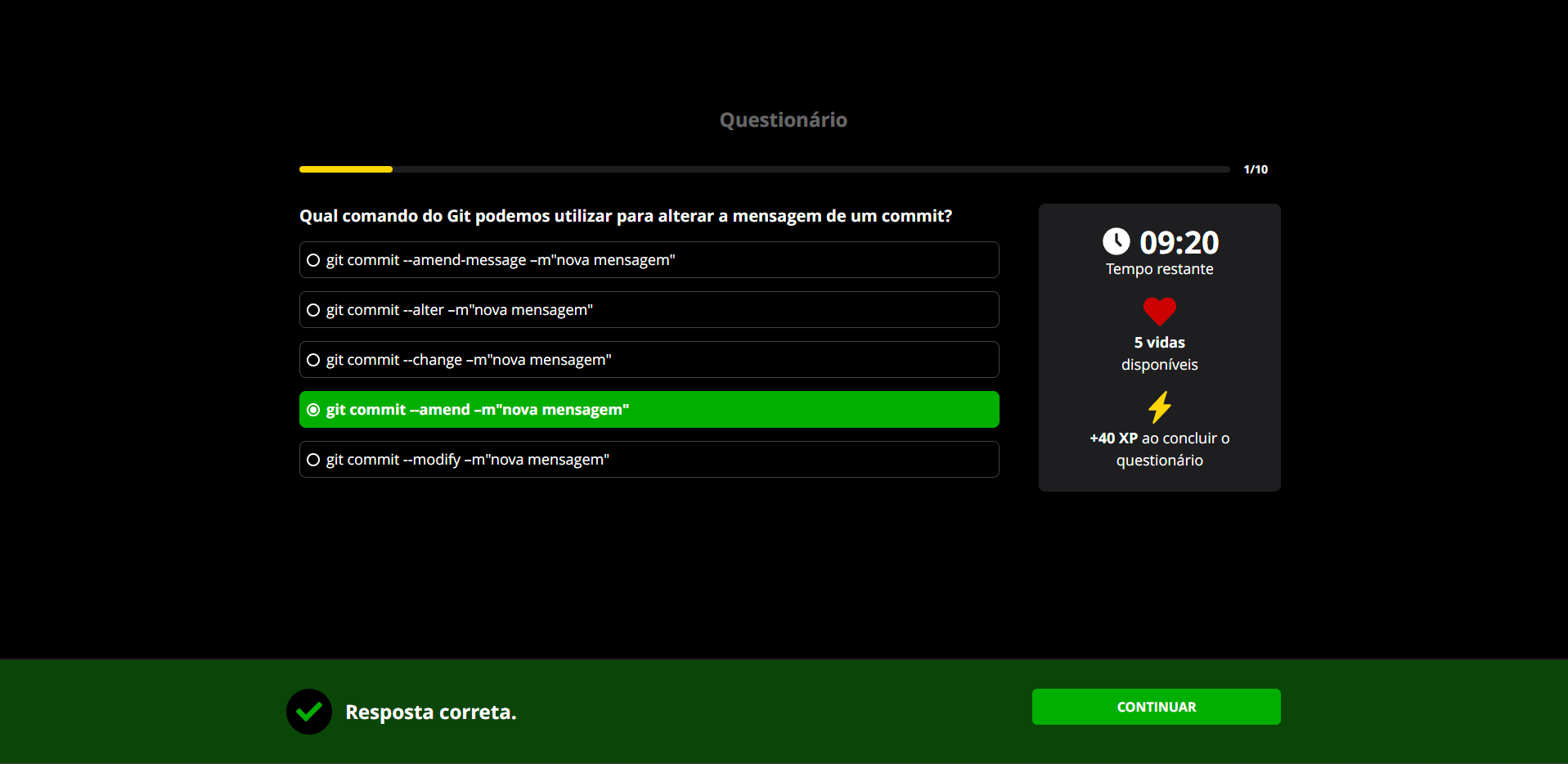
2 inserir o usuário do git hub e email

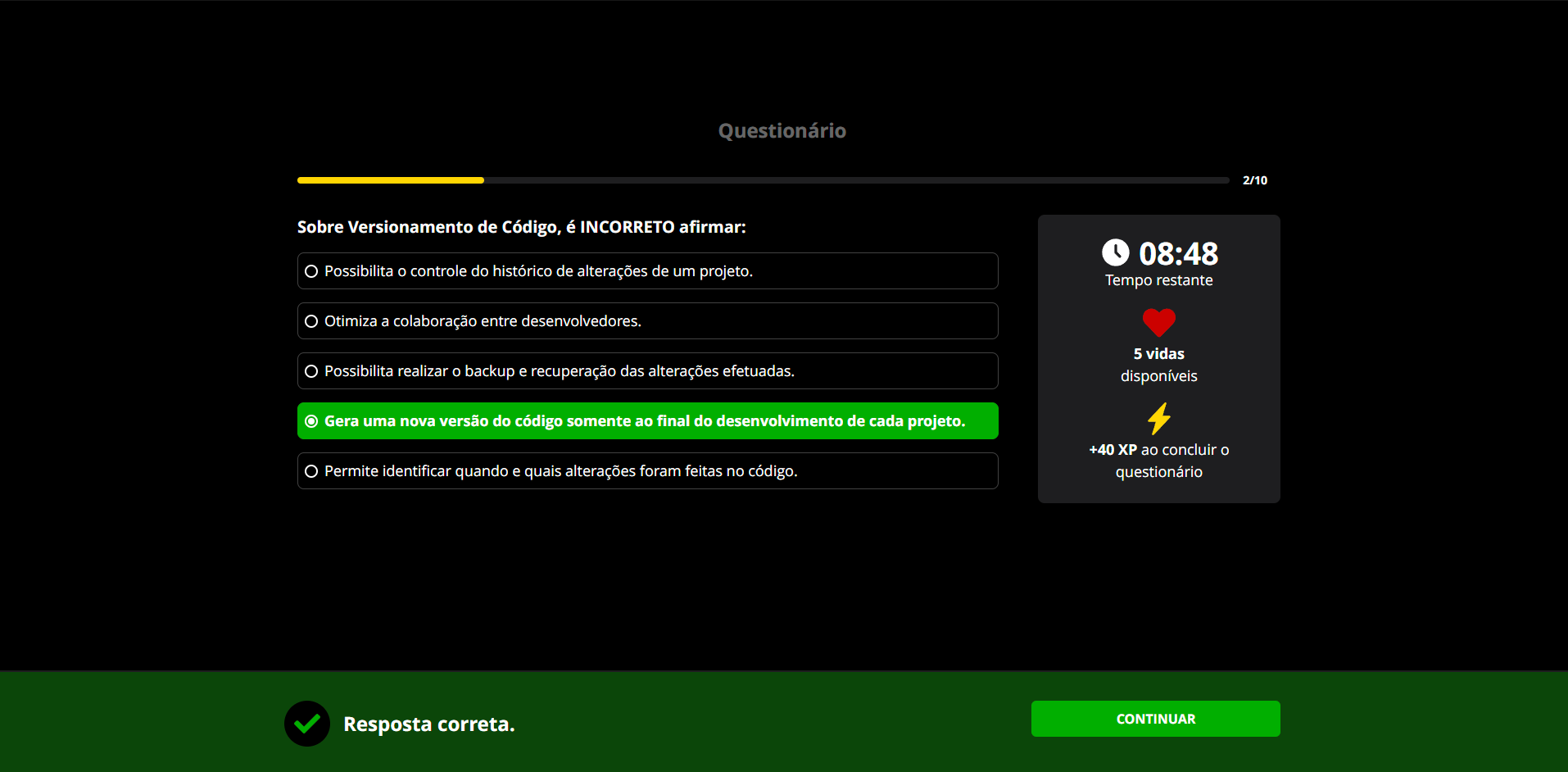
3 inserir o token que foi gerado

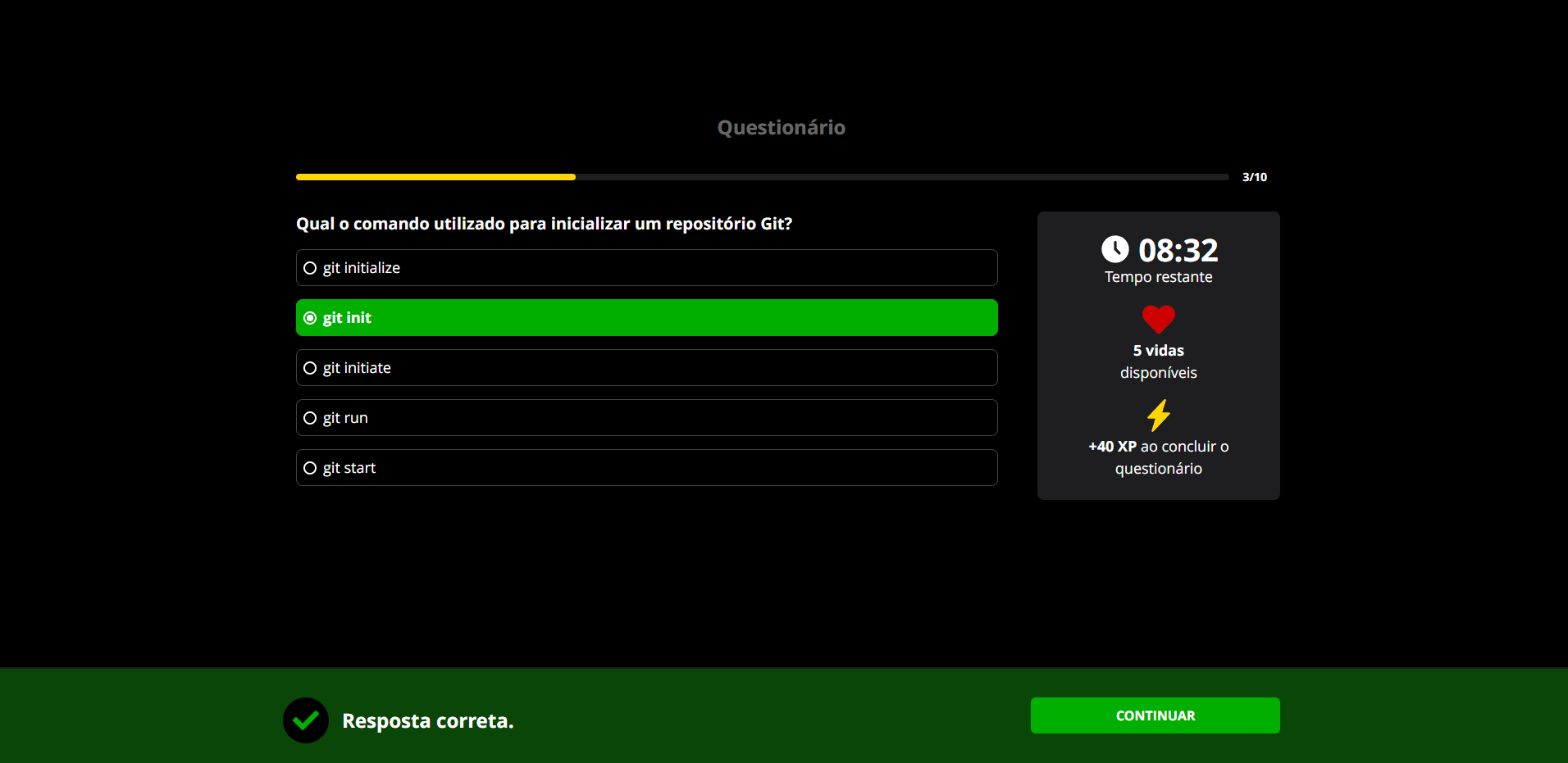
4 dentro do git bash e ou terminal git inserir os seguintes comandos

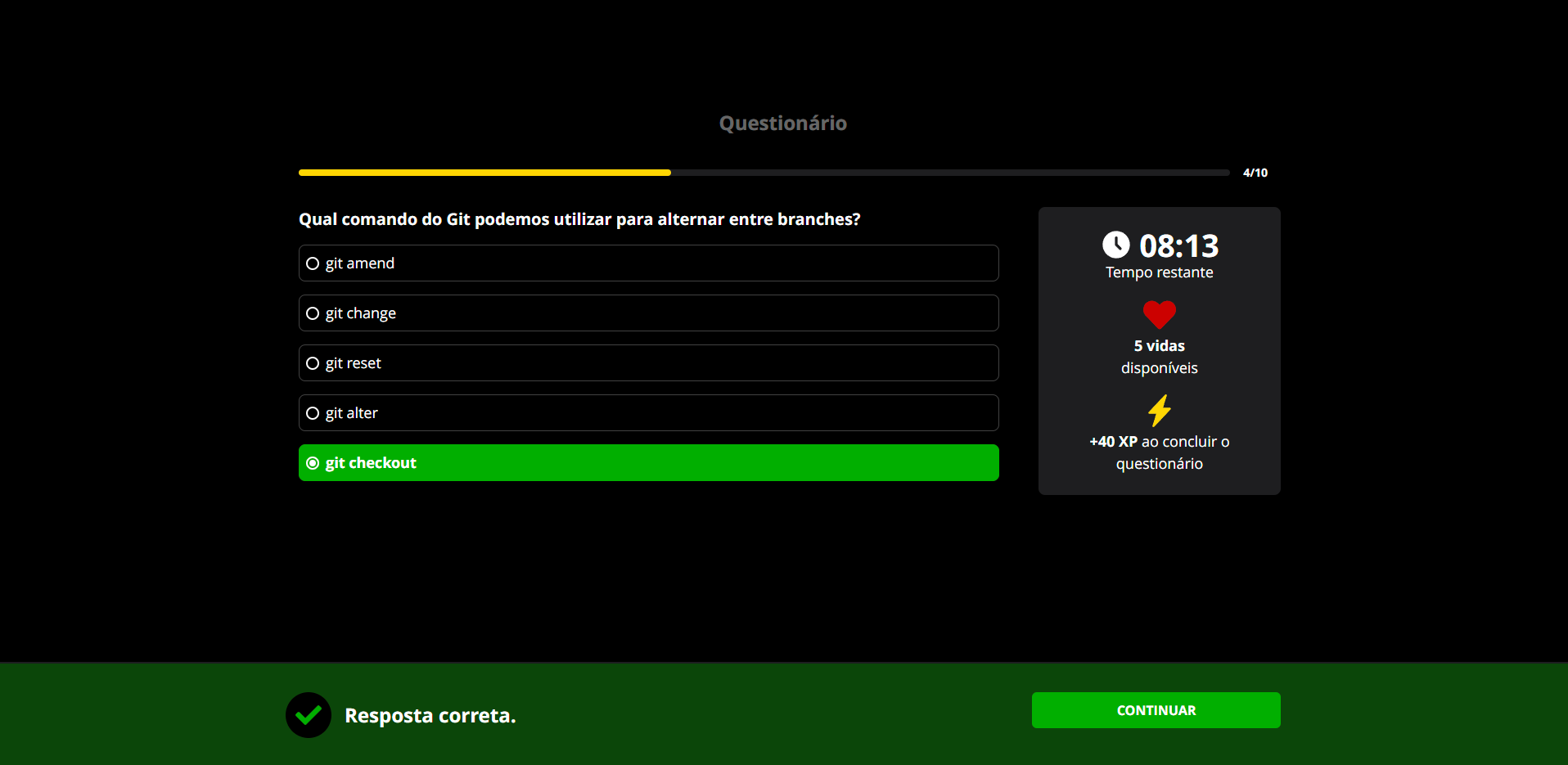
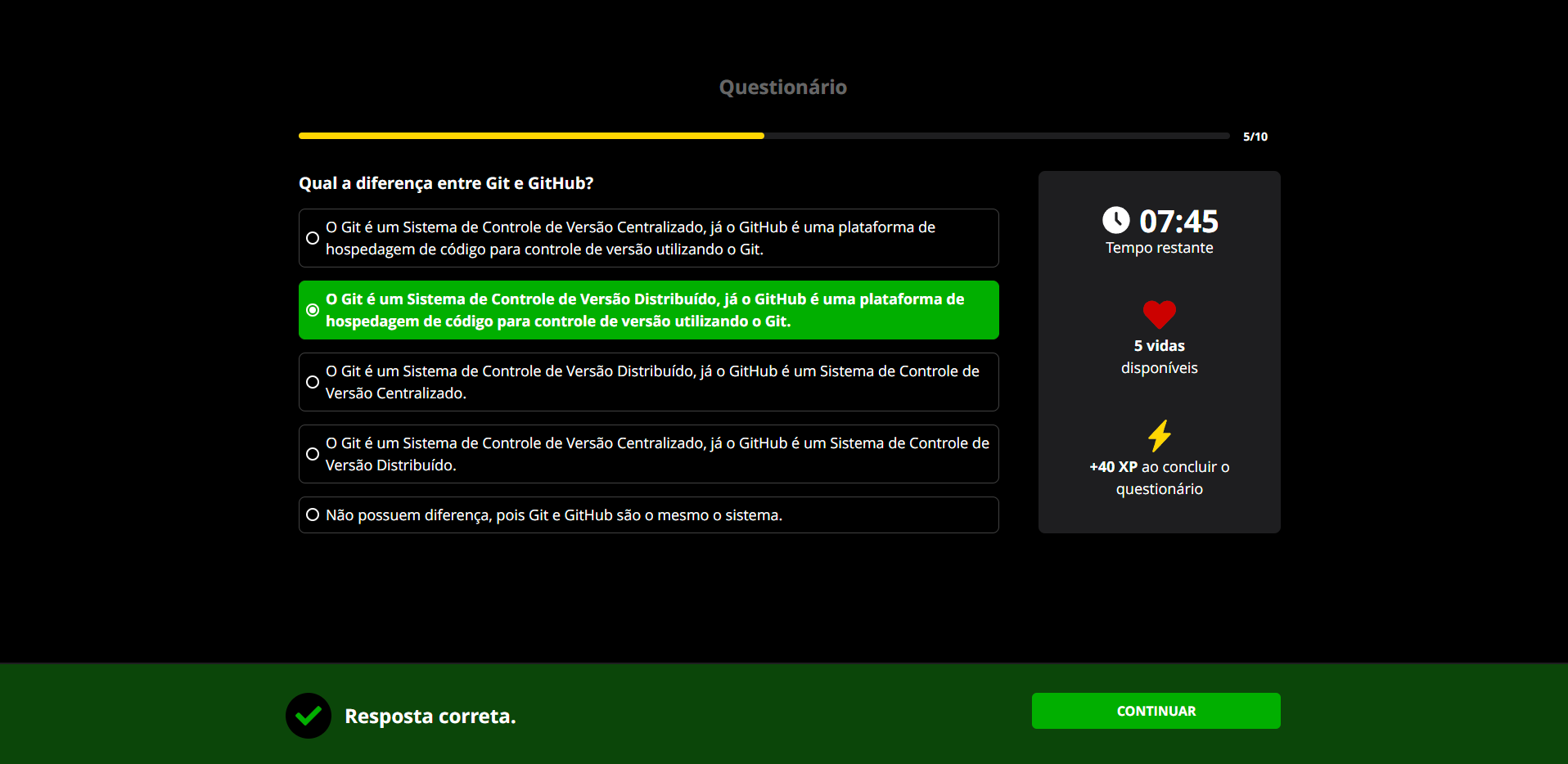
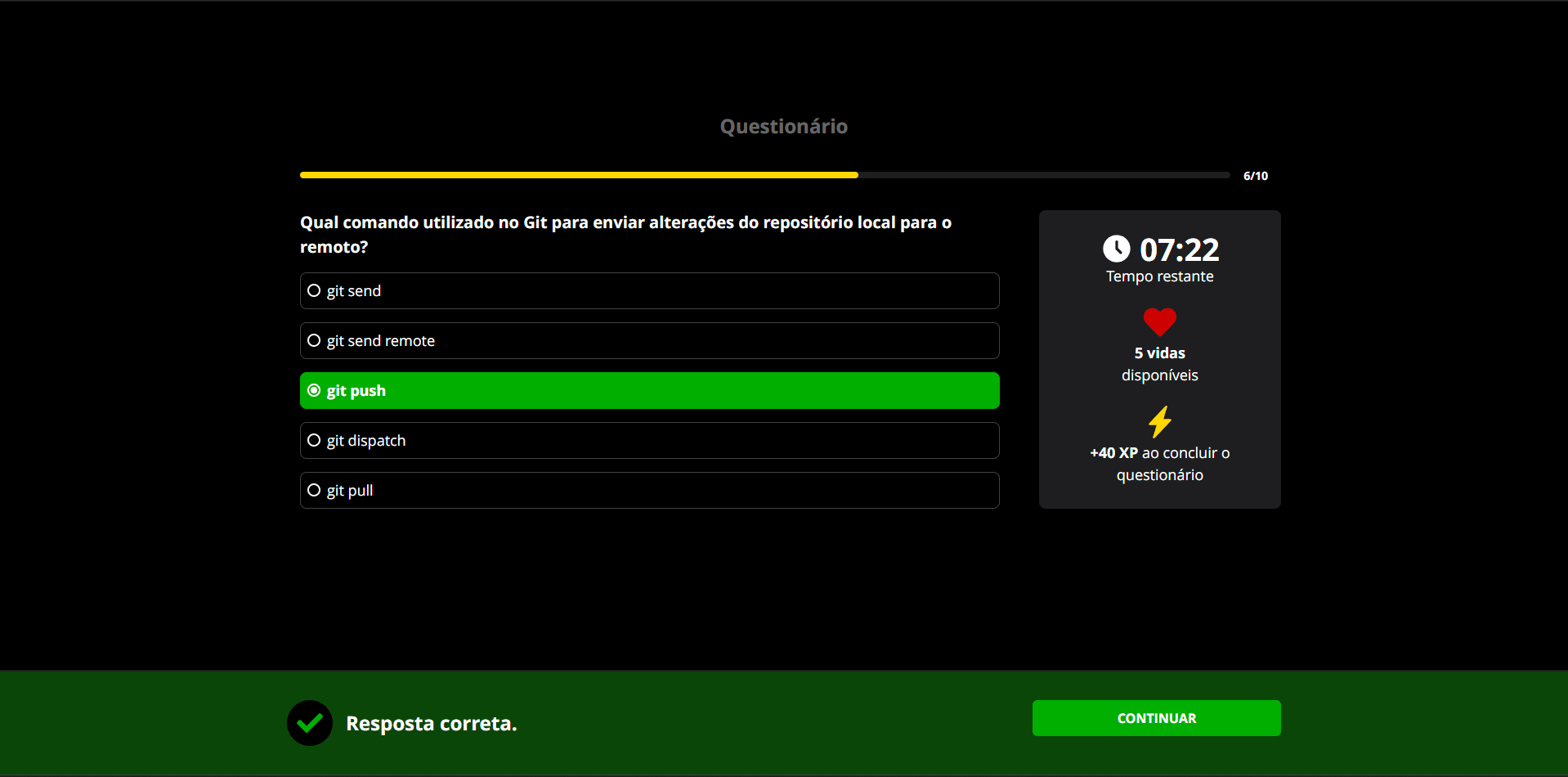
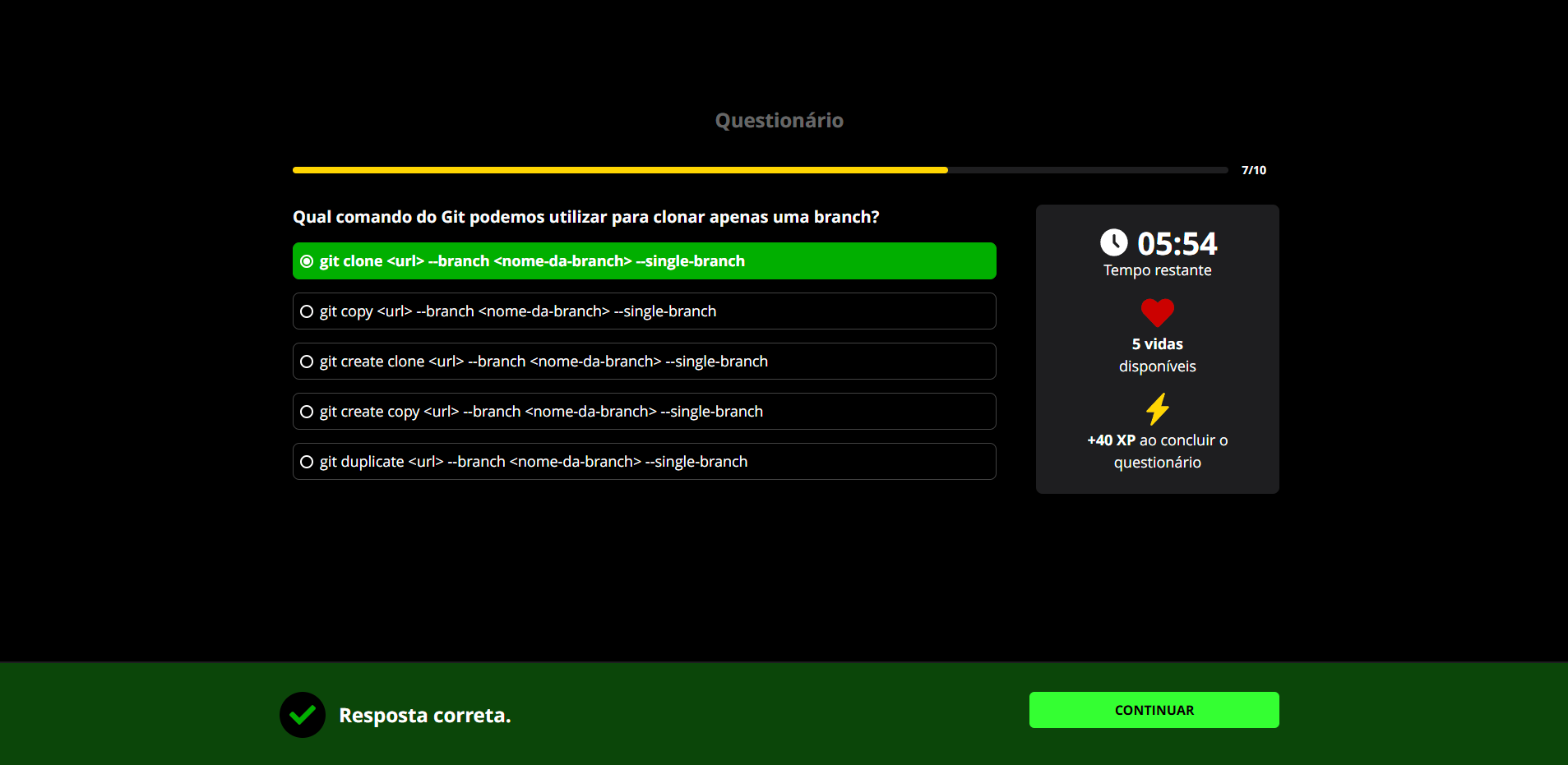
git config credential.helper cache (para esse caso e a configuração para que essa credencial não fique salva)

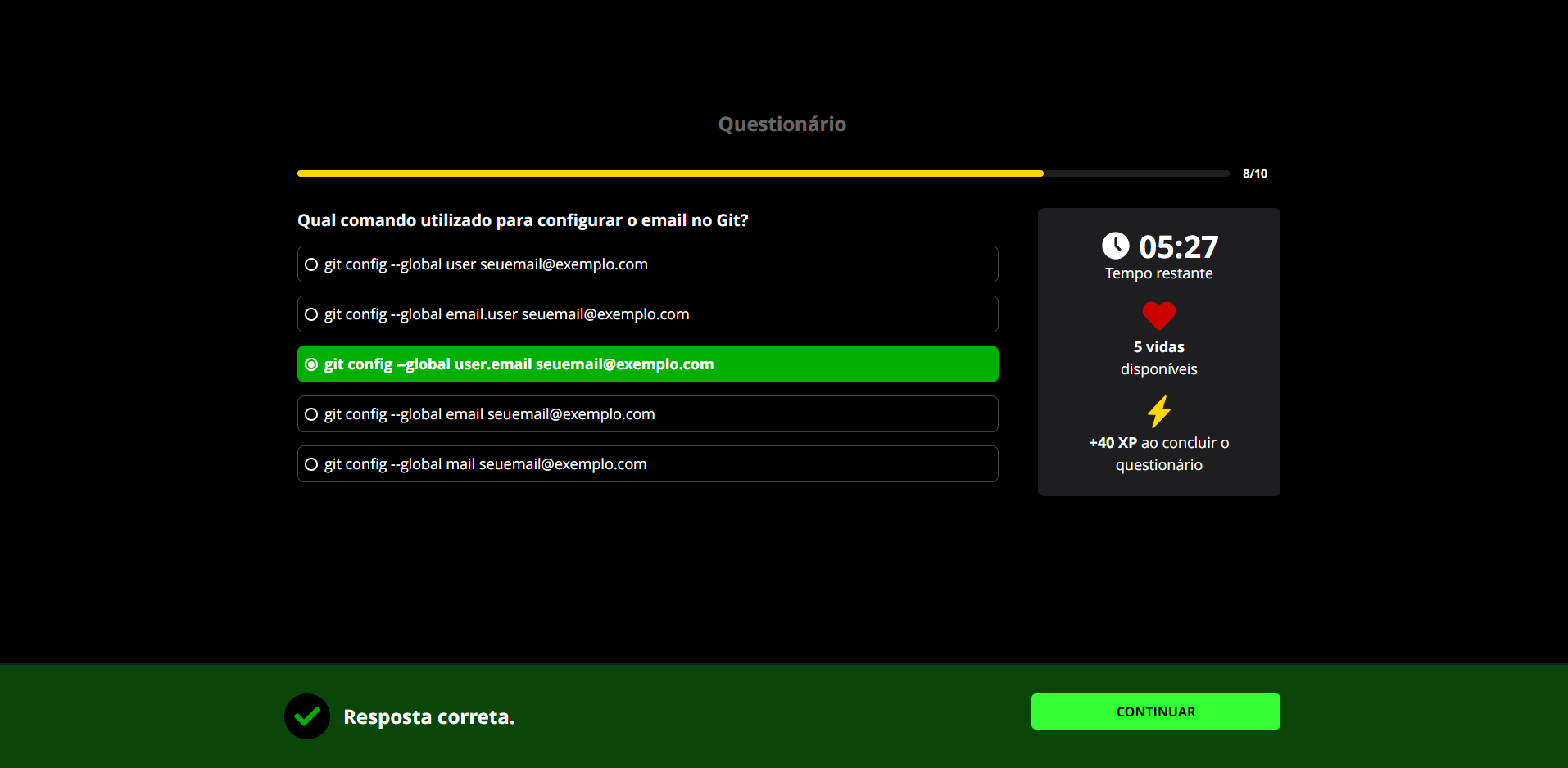
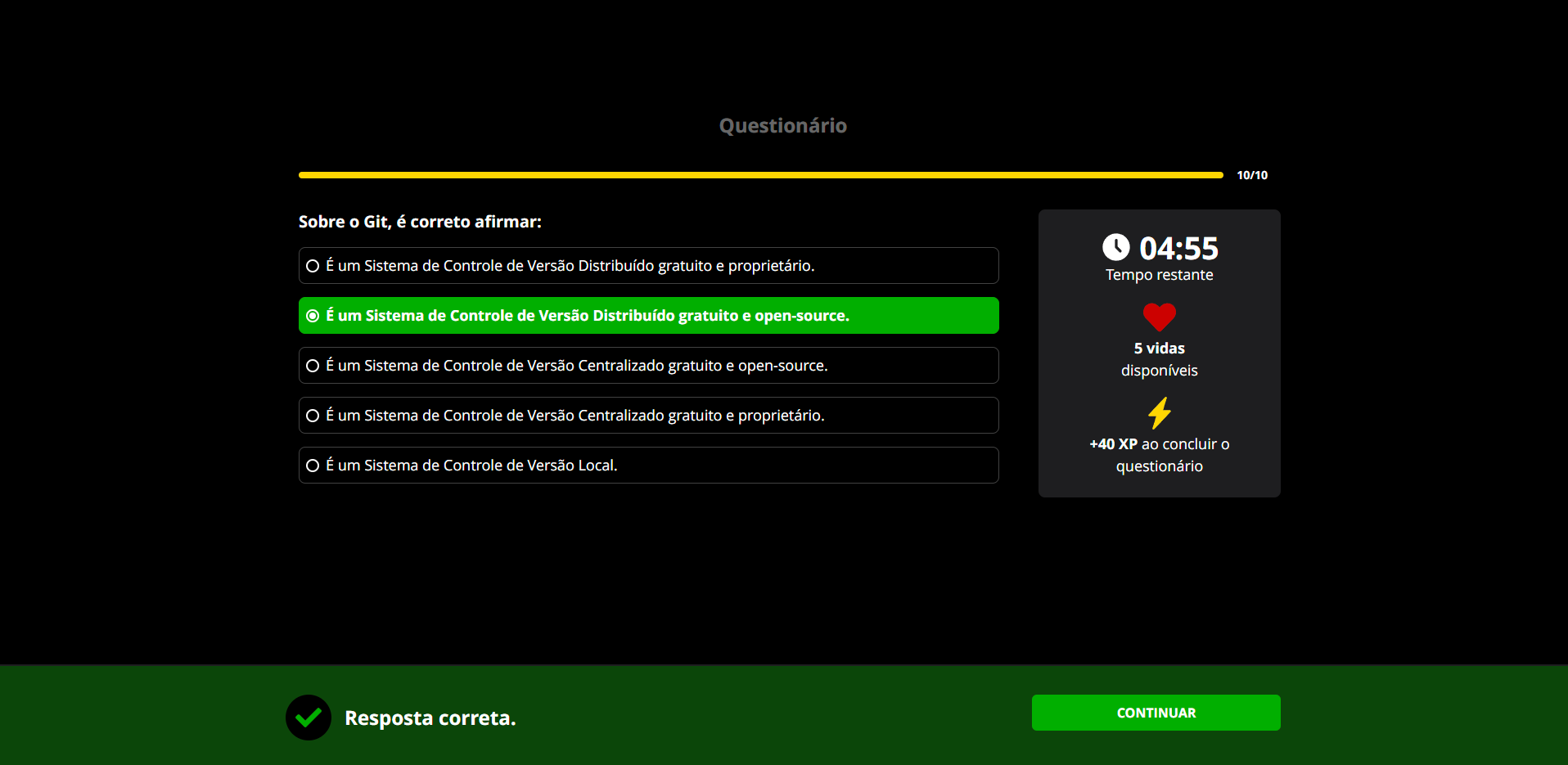
git config credential.helper store (para esse caso e para salvar a credencial gerada com a adição do local (global, system ou local ), ele guarda aquelas configurações.)

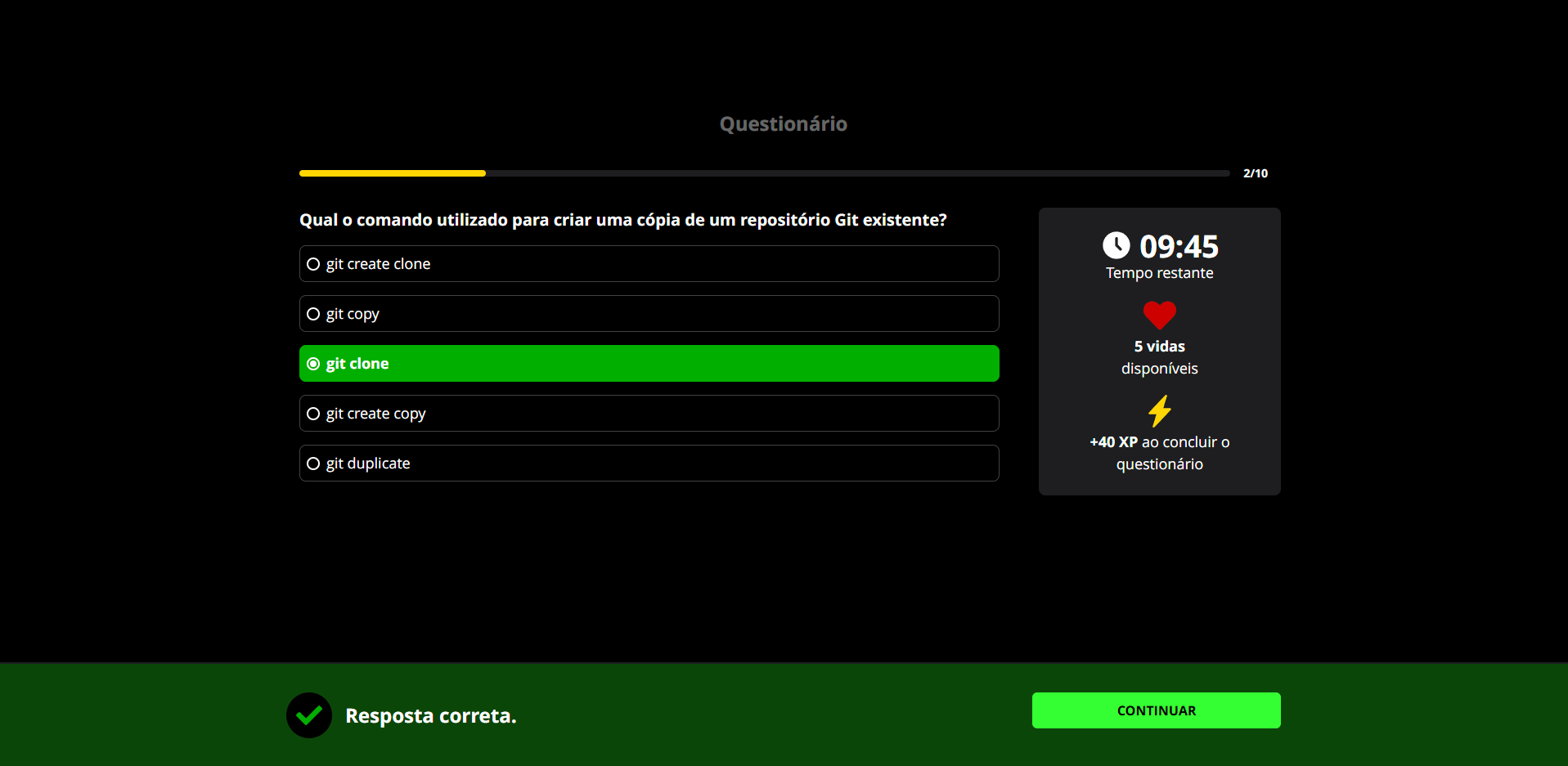










**Materiais Complementares**

Nossos materiais complementares e de apoio têm como objetivo apresentar informações para facilitar e enriquecer a sua jornada de aprendizado. Para isso, links úteis (como slides, repositórios e páginas oficiais) serão disponibilizados, além de dicas sobre como se destacar na DIO e no mercado de trabalho 😉

**Repositório Git**

O Git é um conceito essencial no mercado de trabalho atualmente, por isso sempre reforçamos sua importância em nossa metodologia educacional. Por isso, todo código-fonte desenvolvido durante este conteúdo foi versionado no seguinte endereço para que você possa consultá-lo a qualquer momento:

<https://github.com/elidianaandrade/dio-curso-git-github>

**Slides**

[Versionamento de Código com Git e GitHub.pptx](https://academiapme-my.sharepoint.com/:p:/g/personal/renato_dio_me/EYjkgVZuUv5HsVgJUEPv1_oB_QWs8MFBY_PBQ2UAtLqucg?e=262HGK)

**Dicas/Links Úteis**

Por fim, disponibilizamos alguns links úteis para que você possa se desenvolver ainda mais através de referências oficiais das tecnologias, páginas de documentação e/ou fóruns de discussão relevantes. Nesse contexto, seguem algumas sugestões:

* **Artigos/Fórum**: você pode compartilhar conteúdos técnicos através de [Artigos](https://web.dio.me/articles) (visíveis globalmente na plataforma da DIO). Por outro lado, você também pode compartilhar suas conquistas e dúvidas usando os Fóruns (que são específicos para cada experiência educacional na DIO, como um Bootcamp por exemplo);
* **Rooms**: caso você esteja inscrito(a) em uma experiência educacional na DIO (como um Bootcamp, por exemplo) você terá acesso ao Rooms. O Rooms é uma ferramenta de bate-papo em tempo real onde todos os inscritos podem interagir, compartilhando dúvidas e dicas (que podem conter imagens e snippets de código-fonte);
* **Pesquise na Web**: pode parecer óbvio, mas é importante frisar a importância das engines de busca no dia-a-dia de um profissional de TI. Caso não encontre o que procura dentro da DIO, pesquise sobre o assunto (conceito, dúvida, erro etc) na Internet (dê um Google), pois na maioria das vezes você será levado à páginas incríveis como o [StackOverflow](https://stackoverflow.com/) que salvarão o seu dia 😎