



Codificar com o **GitHub Codespaces**





Codificar com o GitHub Codespaces

O GitHub Codespaces é um ambiente de desenvolvimento **totalmente configurado hospedado na nuvem**. Usando o GitHub Codespaces, seu workspace, juntamente com todos os ambientes de desenvolvimento configurados, fica disponível em qualquer computador com acesso à Internet.

Objetivos de aprendizagem

Ao final deste módulo, **você será capaz de fazer o seguinte:**

- Descreva o GitHub Codespaces.
- Explique o ciclo de vida do GitHub Codespaces e como executar cada etapa.
- Defina as diferentes personalizações que você pode fazer com o GitHub Codespaces.
- Identifique as diferenças entre GitHub.dev e o GitHub Codespaces.

Pré-requisitos

Uma conta do GitHub

Este módulo faz parte destes roteiros de aprendizagem

[GitHub Foundations](#)

Introdução

O ciclo de vida do Codespace

Personalizar seu Codespace

Codespaces versus editor GitHub.dev

Exercício – Codificar com o Codespaces e o Visual Studio Code

Resumo





Introdução

O GitHub Codespaces é um ambiente de desenvolvimento instantâneo e baseado em nuvem que usa um contêiner para fornecer linguagens, ferramentas e utilitários de desenvolvimento comuns.

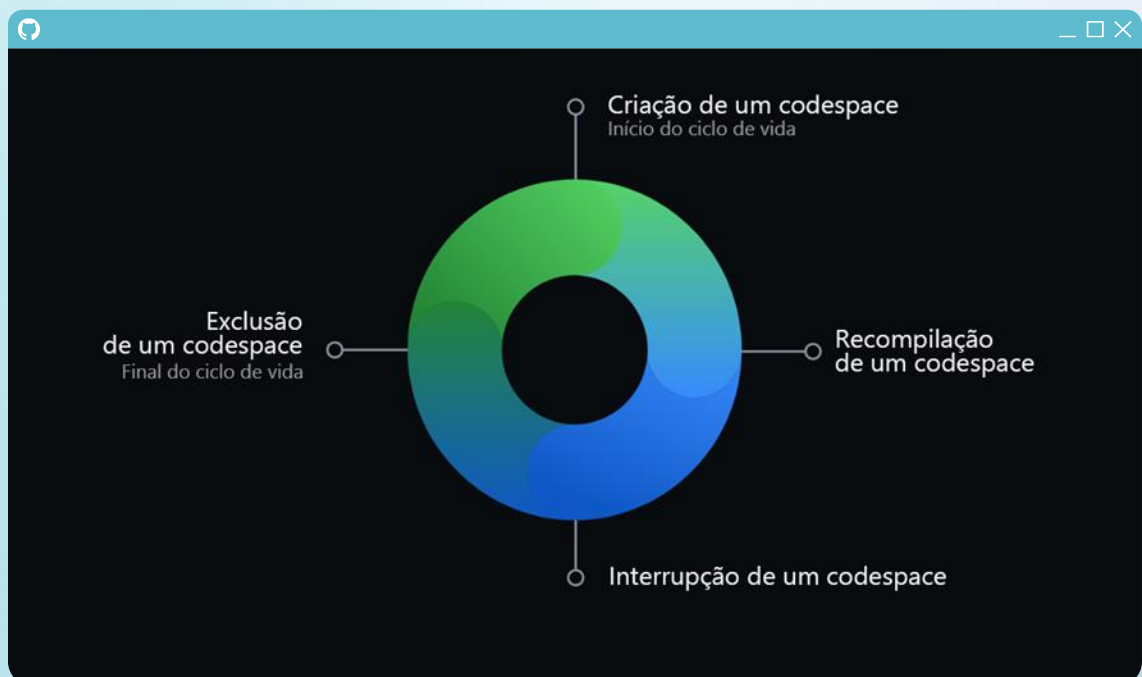
Neste módulo, nós iremos:

- Explorar o ciclo de vida e os processos dos Codespaces.
- Examinar as maneiras de personalizar a configuração do Codespace.
- Comparar as diferenças entre o GitHub Codespaces e GitHub.dev.
- Concluir um exercício para praticar a codificação no Codespaces.

O ciclo de vida do Codespace

O GitHub Codespaces **é configurável, permitindo que você crie um ambiente de desenvolvimento personalizado** para seu projeto. Ao configurar um ambiente de desenvolvimento personalizado para o projeto, você pode ter uma configuração reproduzível do Codespace para todos os usuários do projeto.

O ciclo de vida de um Codespace **começa quando você cria um Codespace e termina quando você o exclui**. Você pode se desconectar e reconectar a um Codespace ativo sem afetar os processos em execução. Você pode parar e reiniciar um Codespace sem perder as alterações feitas no projeto.





Criar um Codespace

Você pode criar um Codespace no GitHub.com, no Visual Studio Code ou usando a CLI do GitHub.

Há quatro maneiras de criar um Codespace:

- Usando um modelo do GitHub ou qualquer repositório de modelos no GitHub.com para iniciar um novo projeto.
- De um branch em seu repositório para um novo trabalho de recursos.
- De um pull request aberto para explorar o trabalho em andamento.
- Usando um commit no histórico do repositório para investigar um bug em um momento específico.

Você pode usar um Codespace temporariamente para testar o código ou pode voltar ao mesmo Codespace para continuar um trabalho de recursos de longa execução.

Você pode criar mais de um Codespace por repositório ou até mesmo por branch. No entanto, há limites quanto ao número de Codespaces que você pode criar e o que você pode executar ao mesmo tempo.

Quando você atinge o número máximo de Codespaces e tenta criar outro, uma mensagem é exibida. A mensagem informa que um Codespace existente precisar ser removido/excluído antes que um novo Codespace possa ser criado.

Você pode criar um Codespace sempre que desenvolver no GitHub Codespaces ou pode manter um Codespace de execução prolongada para um recurso. Se estiver iniciando um novo projeto, crie um Codespace usando um modelo e publique-o em um repositório no GitHub posteriormente.

Ao criar um Codespace sempre que trabalhar em um projeto, você deverá efetuar push das alterações regularmente para garantir que os novos commits estejam no GitHub.

Ao usar um Codespace de execução longa para um novo projeto, efetue pull do branch padrão do repositório sempre que começar a trabalhar no Codespace para permitir que seu ambiente obtenha as confirmações mais recentes. O fluxo de trabalho é semelhante a trabalhar com um projeto em um computador local.

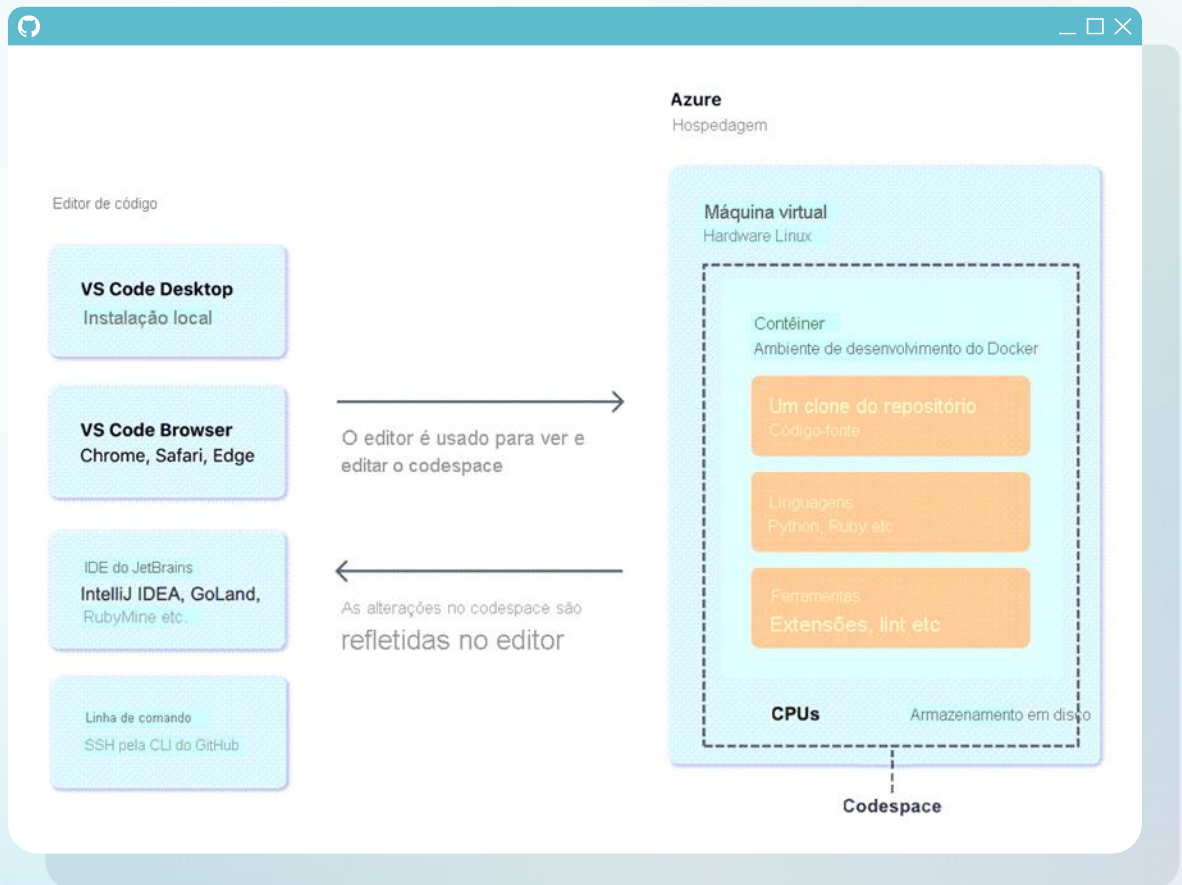
Administradores de repositório podem habilitar pré-compilações do GitHub Codespaces para um repositório para acelerar a criação do Codespace.

Para obter orientações detalhadas e diretrizes passo a passo, confira os recursos Guia para iniciantes para aprender a codificar com o GitHub Codespaces e Desenvolver em um Codespace, localizados na unidade Resumo no final deste módulo.





Processo de criação de um Codespace



Ao criar um GitHub Codespace, **ocorrem quatro processos:**

1. Uma máquina virtual e um armazenamento são atribuídos ao seu Codespace.
2. Um contêiner é criado.
3. É feita uma conexão com o Codespace.
4. É feita uma configuração pós-criação.

Salvar alterações em um Codespace

Quando você se conecta a um Codespace por meio da Web, o Salvamento Automático é habilitado automaticamente para salvar as alterações após um período específico ter passado. Quando se conecta a um Codespace usando o Visual Studio Code em execução na área de trabalho, você precisa habilitar o salvamento automático.

O trabalho é salvo em uma máquina virtual na nuvem. Você pode fechar e parar um Codespace e voltar ao trabalho salvo posteriormente. Se tiver alterações não salvas, você receberá um prompt para salvá-las antes de sair. No entanto, se o Codespace for excluído, seu trabalho será perdido. Para salvar o trabalho, você precisa fazer commit das alterações e efetuar push delas para seu repositório remoto ou publicar o trabalho em um novo caso você tenha criado seu Codespace usando um modelo.





Abrir um Codespace existente

Você pode reabrir qualquer um de seus Codespaces ativos ou parados no GitHub.com, em um IDE do JetBrains, no Visual Studio Code ou usando a CLI do GitHub.

Para retomar um Codespace existente, acesse o repositório onde o Codespace existe, selecione a chave , e, em seguida, **Retomar este codespace**. Ou você pode abrir <https://github.com/codespaces> no navegador, selecionar o repositório e, em seguida, o Codespace existente.

Tempos limite para um Codespace

Se um Codespace estiver inativo ou se você sair dele sem pará-lo explicitamente, o aplicativo atingirá o tempo limite após um período de inatividade e deixará de ser executado. O tempo limite padrão é atingido após 30 minutos de inatividade. Quando um Codespace atinge o tempo limite, seus dados são preservados da última vez em que alterações foram salvas.

Conexão com a Internet usando o GitHub Codespaces

Um Codespace requer uma conexão com a Internet. Se a conexão com a Internet for perdida durante seu trabalho em um Codespace, você não poderá acessar o Codespace. No entanto, as alterações não confirmadas serão salvas. Quando restabelecer a conexão com a Internet, você poderá acessar o Codespace no mesmo estado em que ele foi deixado quando a conexão foi perdida.

Se você tiver uma conexão instável com a Internet, faça commit e efetue push de suas alterações com frequência.

Fechar ou parar um Codespace

Se você sair do Codespace sem executar o comando Parar ou deixar o Codespace em execução sem interação, o Codespace e os processos em execução continuarão durante o período de tempo limite de inatividade.

O período de tempo limite de inatividade padrão é de 30 minutos.

Você pode definir sua configuração de tempo limite pessoal para os Codespaces que criar, mas isso pode ser anulado por uma política de tempo limite da organização.

Somente Codespaces em execução incorrem em encargos de CPU.

Um Codespace parado incorre apenas em custos de armazenamento.

Você pode parar e reiniciar um Codespace para aplicar alterações. Por exemplo, se você alterar o tipo de computador usado no Codespace, precisará parar e reiniciá-lo para que a alteração entre em vigor. Quando você fecha ou para o Codespace, todas as alterações não confirmadas são preservadas até você se conectar novamente ao Codespace.

Você também pode parar o Codespace e optar por reiniciá-lo ou excluí-lo se encontrar um erro ou algo inesperado.





Recompilar um Codespace

Você pode recompilar o Codespace para implementar alterações em sua configuração de contêiner de desenvolvimento. Para a maioria dos usos, você pode criar um Codespace como alternativa à recompilação de um Codespace. Quando você recompila o Codespace, as imagens do cache aceleram o processo de recompilação. Você também pode executar uma recompilação completa para limpar o cache e recompilar o contêiner com imagens novas.

Quando você recompila o contêiner em um Codespace, as alterações feitas fora do diretório `/workspaces` são limpas. Alterações feitas dentro do diretório `/workspaces`, incluindo o clone do repositório ou modelo com base no qual você criou o Codespace, são preservadas em uma recompilação.

Excluir um Codespace

Você pode criar um Codespace para uma tarefa específica. Após efetuar push de alterações para um branch remoto, você poderá excluir com segurança esse Codespace.

Se você tentar excluir um Codespace com git commits que não foram enviadas por push, o editor notificará você de que há alterações que não foram enviadas por push para um branch remoto. Você pode efetuar push de qualquer alteração desejada e excluir o Codespace. Você também pode excluir o Codespace e as alterações não confirmadas ou exportar o código para um novo Branch sem criar um Codespace.

Codespaces parados que permanecem inativos por um período especificado são excluídos automaticamente. Codespaces inativos são excluídos após 30 dias, mas você pode personalizar os intervalos de retenção do Codespace.

Personalizar seu Codespace

O GitHub Codespaces é um ambiente dedicado para você.

Você pode configurar seus repositórios com um contêiner de desenvolvimento para definir o ambiente padrão do GitHub Codespaces e personalizar sua experiência de desenvolvimento em todos os seus Codespaces com dotfiles e a Sincronização de Configurações.

O que você pode personalizar

Há várias maneiras de personalizar um Codespace.

Vamos examinar cada uma delas.

- **Sincronização de Configurações:**
Você pode sincronizar suas configurações do Visual Studio Code (VS Code) entre o aplicativo da área de trabalho e o cliente Web do VS Code.
- **Dotfiles:**
Você pode usar um repositório dotfiles para especificar scripts, preferências de shell e outras configurações.





- **Renomear um Codespace:**

Quando você cria um Codespace, um nome de exibição gerado automaticamente é atribuído a ele. Se você tiver vários Codespaces, o nome de exibição ajudará a diferenciar entre eles. Você pode alterar o nome de exibição do Codespace.

- **Alterar seu shell:**

Você pode alterar seu shell em um Codespace para manter a configuração com a qual está acostumado. Ao trabalhar em um Codespace, você pode abrir uma nova janela de terminal com um shell de sua escolha, alterar o shell padrão para novas janelas de terminal ou instalar um novo shell. Você também pode usar dotfiles para configurar o shell.

- **Alterar o tipo de computador:**

Você pode alterar o tipo de computador que está executando o Codespace para usar os recursos apropriados ao trabalho que está fazendo.

- **Definir o editor padrão:**

Você pode definir o editor padrão para Codespaces em sua página de configurações pessoais. Defina sua preferência de editor para que, ao criar um Codespace ou abrir um Codespace existente, ele seja aberto no editor padrão.

- Visual Studio Code (aplicativo da área de trabalho)
- Visual Studio Code (aplicativo cliente Web)
- JetBrains Gateway – para abrir Codespaces em um IDE do JetBrains
- JupyterLab – A interface Web do Project Jupyter

- **Definir a região padrão:**

Você pode definir sua região padrão na página de configurações do perfil do GitHub Codespaces para personalizar o local onde os dados são mantidos.

- **Definir o tempo limite:**

Um Codespace deixará de ser executado após um período de inatividade. Por padrão, esse período é de 30 minutos, mas você pode especificar um período de tempo limite padrão maior ou menor nas configurações pessoais no GitHub. A configuração atualizada se aplica a todos os Codespaces que você criar ou aos Codespaces existentes na próxima inicialização.

- **Configurar a exclusão automática:**

Codespaces inativos são excluídos automaticamente. Você pode escolher por quanto tempo Codespaces parados são retidos, até, no máximo, 30 dias.

Informações adicionais e instruções passo a passo sobre a personalização estão localizadas na unidade Resumo no final deste módulo.

Adicionar ao Codespace com extensões ou plug-ins

Você pode adicionar plug-ins e extensões em um Codespace para personalizar sua experiência no JetBrains e no VS Code.





Extensões do VS Code

Se você trabalha em seus Codespaces e no aplicativo da área de trabalho do VS Code ou no cliente Web, pode adicionar as extensões necessárias do Marketplace do Visual Studio Code. Consulte [Suporte ao desenvolvimento remoto e ao GitHub Codespaces](#) na documentação do VS Code para obter informações sobre como as extensões são executadas no GitHub Codespaces.

Se você já usa o VS Code, use a Sincronização de Configurações para sincronizar automaticamente extensões, configurações, temas e atalhos de teclado entre a instância local e os Codespaces que você criar.

Plug-ins do JetBrains

Se você trabalha nos Codespaces em um IDE do JetBrains, pode adicionar plug-ins do Marketplace do JetBrains.



Codespaces versus editor GitHub.dev

Provavelmente, você está se perguntando, **quando devo usar o GitHub Codespaces e quando devo usar GitHub.dev?**

Você pode usar GitHub.dev para navegar por arquivos e repositórios de código-fonte do GitHub e fazer e confirmar alterações de código. Você pode abrir qualquer repositório, bifurcação ou PR no editor GitHub.dev.

Se quiser fazer um trabalho mais pesado, como testar o código, use o GitHub Codespaces. Ele tem computação associada para que você possa criar o código, executar o código e ter acesso ao terminal. GitHub.dev não tem computação. Com o GitHub Codespaces, você obtém o poder de uma VM (Máquina Virtual) pessoal com acesso ao terminal, da mesma maneira que poderia usar seu ambiente local, só que na nuvem.





Comparação entre Codespaces e GitHub.dev

A seguinte tabela lista as diferenças principais entre Codespaces e GitHub.dev:

	GitHub.dev	Codespaces do GitHub
Custo	Gratuita	Cota mensal gratuita de uso para contas pessoais.
Disponibilidade	Disponível para todos no GitHub.com	Disponível para todos no GitHub.com.
Inicialização	GitHub.dev abre instantaneamente com o pressionar de uma tecla e você pode começar a usá-lo imediatamente, sem ter que esperar pela configuração ou instalação.	Quando você cria ou retoma um Codespace, uma VM é atribuída ao Codespace. O contêiner é então configurado com base no conteúdo de um arquivo devcontainer.json. Essa configuração leva alguns minutos para criar o ambiente de desenvolvimento.
Computação	Não há recursos de computação associados, portanto você não pode criar e executar seu código ou usar o terminal integrado.	Com o GitHub Codespaces, você tem a potência de uma VM dedicada para executar e depurar seu aplicativo.
Acesso ao terminal	Nenhum	O GitHub Codespaces fornece um conjunto comum de ferramentas por padrão, o que significa que você pode usar o Terminal exatamente como faria no ambiente local.
Extensões	Apenas um subconjunto de extensões que podem ser executadas na Web aparecerão na visualização de extensões e poderão ser instaladas	Com o GitHub Codespaces, você pode usar a maioria das extensões do Marketplace do Visual Studio Code.





Continuar trabalhando em Codespaces

Você pode começar seu fluxo de trabalho no GitHub.dev e continuar em um Codespace. Se você tentar acessar a Visualização Executar e Depurar ou o Terminal, verá uma notificação de que eles não estão disponíveis em GitHub.dev.

Para continuar o trabalho em um Codespace, selecione **Continuar Trabalhando...** Selecione **Criar Codespace** para criar um Codespace no branch atual. Antes de selecionar esta opção, você precisa fazer commit de quaisquer alterações.

Exercício

Codificar com o Codespaces e o Visual Studio Code

Agora que você entende o ciclo de vida e os processos do Codespaces, é hora de praticar a codificação no Codespaces e no Visual Studio Code (VS Code).

Use as instruções a seguir para realizar esse exercício.

Instruções

1. Clique com o botão direito do mouse no link do [exercício do GitHub](#) para abri-lo em uma nova guia.
2. Na página inicial do exercício do GitHub, clique com o botão direito do mouse no botão **Iniciar curso** para abri-lo em uma nova guia.
 1. Na nova guia, a maioria das solicitações são preenchidas automaticamente para você.
 2. Para proprietário, escolha sua conta pessoal ou uma organização para hospedar o repositório.
 3. É recomendável criar um repositório público, pois repositórios privados usam minutos do GitHub Actions.
3. Role para baixo e selecione o botão **Criar repositório** na parte inferior do formulário.
4. Após o repositório ser criado, aguarde cerca de 20 segundos e atualize a página. Siga as instruções passo a passo no LEIAME do novo repositório.

Após concluir o exercício no GitHub, volte aqui para:

- Concluir uma verificação de conhecimentos.
- Revisar um resumo do que você aprendeu neste módulo.
- Receber um selo pela conclusão deste módulo.





Resumo

Neste módulo, você aprendeu sobre o GitHub Codespaces, um ambiente de desenvolvimento totalmente configurado hospedado na nuvem.

Agora você deve estar apto a:

- Descreva o GitHub Codespaces.
- Explique o ciclo de vida do GitHub Codespaces e como executar cada etapa.
- Defina as diferentes personalizações que você pode fazer com o GitHub Codespaces.
- Identifique as diferenças entre GitHub.dev e o GitHub Codespaces.

Recursos

Confira estes links com mais informações sobre os assuntos abordados neste módulo:

- [Um guia para iniciantes para aprender a codificar com o GitHub Codespaces](#)
- [Desenvolver em um Codespace](#)
- [Personalizando seu Codespace](#)

Para escrever códigos com mais rapidez e eficiência, use a IA do GitHub Copilot, que sugere linhas de código e funções inteiras com base no contexto do seu projeto.

Aprenda mais sobre a integração GitHub Copilot no GitHub Codespaces em: <https://aka.ms/githubcopilotcodespaces>

Para te ajudar a começar a usar as ferramentas que aumentarão sua produtividade e reduzirão seus esforços de desenvolvimento, disponibilizamos os links para acessá-las:

GitHub Copilot: <https://aka.ms/githubcopilotquickstart>

GitHub Codespaces: <https://aka.ms/githubcodespacesquickstart>

GitHub Actions: <https://aka.ms/githubactionsquickstart>

