

# Verificação de conhecimentos

3 minutos

Escolha a melhor resposta para cada pergunta e selecione **Verificar suas respostas**.

## Verificar seus conhecimentos

1. Qual é a melhor maneira de garantir que você esteja integrando as versões mais seguras de suas dependências de projeto? \*

☐ Configurar os arquivos de pacote para sempre usar as versões mais recentes das dependências.

☐ Verifique atentamente cada detalhe de segurança do projeto antes de adicioná-lo às suas dependências. Para isso, confirme o status da versão em vários sites de consultoria.

☒ Habilite o **Dependabot** para o repositório.

✓ **O Dependabot verifica os manifestos de dependência do repositório e notifica você por meio de solicitação de pull sempre que uma versão confiável for marcada como insegura.**

2. Imagine que um dos seus projetos de origem usa segredos em uma pasta chamada `.secrets`. Você gostaria de garantir que os arquivos mantidos nesta pasta em computadores de desenvolvimento não sejam confirmados inadvertidamente no repositório. Qual desses arquivos melhor ajuda a impor essa política? \*

☐ `SECURITY.md`

☒ `.gitignore`

✓ **`.gitignore` pode ser usado para ajudar a impor os arquivos incluídos em confirmações por ferramentas que o obedecem. No entanto, o cliente impõe essa política e não necessariamente impede que os usuários confirmem arquivos que violam a política.**

☐ `CONTRIBUTING.md`

3. Suponha que alguém confirme inadvertidamente uma chave de API confidencial armazenada na pasta `.secrets`. Qual é a maneira correta de remover essas informações do GitHub? \*



Use `git` para remover a confirmação indesejada e atualizar as referências históricas. Em seguida, entre em contato com o suporte do GitHub para executar a coleta de lixo e invalidar o cache Git.

✓ Essa abordagem será o processo correto para remover os dados a partir de agora. No entanto, se você achar que alguém pode acessar a chave quando ela estiver disponível, você deverá substituir a chave por uma nova. Como prática recomendada, considere os dados confidenciais comprometidos e substitua a chave.



Exclua o arquivo confidencial do GitHub. Em seguida, confirme um arquivo vazio na mesma localização para substituí-lo.



É uma pegadinha. Depois que você confirma algo no GitHub, ele fica lá infinitamente. É por isso que hashes globalmente exclusivos são usados para identificar tudo.

---

## Unidade seguinte: Resumo

Continuar >

---