Curso ATDEV120 - Desenvolvimento em Equipa com o GitHub

Duração: 28h

Introdução

Este curso dá aos participantes as competências para tirar partido do ecossistema GitHub em fluxos de trabalho de equipa eficientes, dominando pipelines de integração e deployment contínuos para entregar software de qualidade rapidamente. Os formandos ganham à-vontade com Git para controlo de versões, GitHub Projects para gestão ágil e GitHub Actions para automatização, compreendendo ao mesmo tempo práticas DevSecOps essenciais com o GitHub Advanced Security.

Público-alvo

Este curso destina-se a qualquer pessoa envolvida no desenvolvimento de aplicações.

No final do curso

Após a conclusão deste curso, os formandos deverão ser capazes de:
• Utilizar o Git, branches e pull requests no trabalho em equipa.
• Gerir projetos e sprints com GitHub Issues e Projects.
• Implementar integração e deployment contínuos com GitHub Actions.
• Aplicar deployment patterns e gerir configuração e segredos.
• Gerir infrastructure as code e pacotes com o GitHub Packages.
• Alargar a equipa de desenvolvimento com o GitHub Copilot.

Pré-requisitos

Os exercícios não exigem competências reais de programação; servem para ilustrar o fluxo de trabalho do GitHub.

Exames de certificação

(não existem exames)

Conteúdo programático em detalhe

Introdução ao ecossistema GitHub
• O GitHub Flow vs. o DevOps tradicional
• Compreender a hierarquia: enterprise, organizações e equipas
• Contas pessoais vs. repositórios da organização
• LAB: criar a sua organização e configurar um repositório
Utilização do Git
• Conceitos de controlo de versões
• Controlo de versões distribuído com Git
• Clientes de Git: linha de comandos, Visual Studio ou Visual Studio Code
• Criar um repositório local
• Configurar o seu repositório Git
• Verificar o estado do repositório

- Colocar ficheiros em staging
- Fazer commit de ficheiros
- Comparar alterações em ficheiros
- Reverter commits
- LAB: trabalhar com um repositório local

Colaboração em equipa com repositórios remotos

- Colaboração em equipa com repositórios remotos
- Clonar um repositório remoto
- Sincronizar com um repositório remoto (push/pull)
- Resolver conflitos no pull do Git
- LAB: colaborar num repositório partilhado do GitHub

Trabalhar isoladamente com branches

- Acrescentar branches ao seu fluxo de trabalho
- Estratégias de branching: o GitHub Flow
- Merge vs. rebase — manter um histórico linear
- Diferentes estratégias de merge (squash, rebase e merge commit)
- Gerir os seus branches pela interface e pela CLI
- LAB: implementar um fluxo de feature branch

Colaboração em equipa com pull requests

- Proteger os seus branches com branch protection rules
- Revisões de código e discussões em fio
- Usar code owners para atribuição automática
- Pull requests em rascunho vs. prontos para revisão
- LAB: gerir o ciclo de vida de um pull request e a revisão de código

Receitas de Git

- Git worktrees para gerir vários checkouts
- Guardar alterações em stash para mudar rapidamente de contexto
- Git cherry-pick para aplicar commits específicos
- Rebase interativo para limpar o histórico de commits
- Git reflog para recuperar commits ou branches perdidos
- Git hooks para automatizar o fluxo de trabalho
- Aliases próprios para comandos personalizados
- Git bisect para caça automatizada a erros
- Remover segredos e dados sensíveis do histórico
- Um .gitignore global para ambientes mais limpos

Alargar a equipa de desenvolvimento com o GitHub Copilot

- Introdução ao GitHub Copilot Coding Agent
- Definir requisitos de funcionalidades com issues do GitHub
- Atribuir ao Copilot a revisão de um pull request
- Orientar o Copilot
- Rever alterações com o GitHub Codespaces
- LAB: juntar o GitHub Copilot à equipa de desenvolvimento do seu projeto

Gestão de projeto com GitHub Issues e Projects

- Introdução aos GitHub Issues (labels, responsáveis e menções)

- Converter issues em tarefas (tasklists)
- Usar o GitHub Projects (tabelas, quadros e roadmaps)
- Campos próprios e automatização
- LAB: montar um project board e preencher o backlog

Planear e executar sprints

- Gerir sprints com campos de iteração e milestones
- Planeamento de capacidade e visualização da carga de trabalho
- Automatizar o progresso com workflows e ligação a pull requests
- Monitorizar a saúde do projeto com burndown charts e insights
- LAB: planear e executar um sprint no GitHub Projects

Integração contínua com GitHub Actions

- Fazer continuous integration
- Introdução aos workflows em YAML
- Eventos (triggers), jobs e steps
- Usar o GitHub Marketplace para obter actions
- Runners alojados pelo GitHub vs. runners próprios
- LAB: criar um workflow de CI para construir e testar código

Deployment contínuo com GitHub Actions

- Gestão de releases nas GitHub Actions
- Definir ambientes (desenvolvimento, QA e produção)
- Montar deployment protection rules
- Autenticar-se junto dos fornecedores de nuvem (OIDC)
- LAB: criar um workflow de CD para publicar no Azure

Deployment patterns

- Publicar é mais do que publicar em ambientes de QA e de produção
- Explorar a arquitetura de microserviços
- Minimizar a indisponibilidade da publicação com a publicação blue/green
- Usar canary releases
- Expor funcionalidades novas com feature toggles
- Usar os GitHub environments para exposição progressiva
- LAB: implementar um workflow de publicação faseada

Gestão de configuração e de segredos

- Segredos do repositório vs. segredos do ambiente
- Usar variáveis de configuração
- Gerir segredos nos GitHub Codespaces
- Passar segredos às actions com segurança
- LAB: usar segredos com segurança numa GitHub Action

Implementar deployment contínuo seguro

- Gestão de dependências com o Dependabot
- Static application security testing (SAST) com o CodeQL
- Secret scanning (evitar fugas de credenciais)
- Push protection
- LAB: ativar o Dependabot e corrigir um alerta de segurança

Infrastructure as code

- O que é infrastructure as code?
- Escrever templates de Bicep
- Publicar Bicep com a CLI do Azure
- Usar a Desired State Configuration
- Usar continuous deployment para criar a sua infraestrutura
- LAB: publicar a infraestrutura da sua aplicação

GitHub Packages

- Compreender o registry do GitHub Packages
- Publicar um pacote a partir de um workflow
- Consumir pacotes num workflow
- Estratégias de versionamento
- LAB: publicar e consumir um pacote privado