

Module Quiz: Software collaboration

1. What is Revision History in Version Control?

1 / 1 ponto

- ☐ A record of all users in a project
- ☐ A record of all bugs in a project
- ☒ A record of all changes in a project

✓ Correto

Correct! Revision History is a record of all changes within a project. It allows you to pinpoint who made the changes, when they were made and what was changed.

2. When using a Distributed Version Control System, a connection to a central server is always required to perform an action.

1 / 1 ponto

- ☐ True
- ☒ False

✓ Correto

Correct! In a DVCS, you can work in an offline state and only connect to a server to push and pull changes.

3. A Version Control system allows you to roll back your files back to a specific point in time.

1 / 1 ponto

- ☒ True
- ☐ False

✓ Correto

Correct! Revert/Rollback is a common feature in version control systems.

4. When using a Centralised Version Control system, for other developers to see your changes you must commit them to a central server.

1 / 1 ponto

- ☒ True
- ☐ False

✓ Correto

Correct! All actions must be done through the central server.

5. Version Control is inefficient when multiple developers work on one project.

1 / 1 ponto

- ☐ True
- ☒ False

✓ Correto

Correct! Version Control combined with testing and automation allows a development team to work efficiently.

6. _____ automatically packages an application and prepares it for deployment.

1 / 1 ponto

- ☐ Continuous Deployment
- ☒ Continuous Delivery
- ☐ Continuous Integration

✓ Correto

Correct! Continuous Delivery automatically packages the application and prepares it for deployment.

7. Which of the following are distributed Version Control Systems? Select all that apply.

1 / 1 ponto

☒ Git

☒ **Correto**

Correct! Git is a distributed version control system.

☒ Mercurial

☒ **Correto**

Correct! Mercurial is a distributed version control system.

☐ Subversion

☐ Concurrent Versions System

8. What are the main benefits of having a staging environment?

1 / 1 ponto

☒ It provides a safe place to test and verify changes prior to being deployed to a live production environment.

☒ **Correto**

Correct! Staging is its own environment and any changes made to it do not affect the production environment in any way. It is completely isolated.

☒ A staging environment mimics the production environment which allows for more accurate testing to find any possible issues.

☒ **Correto**

Correct! Having the same setup as the production environment gives peace of mind that the changes being introduced are not going to cause issues when released to production.

☒ It can allow teams to demo or showcase new features in a controlled environment.

☒ **Correto**

Correct! Staging can be used to demo new features and changes without impacting the production site.

☐ It allows developers to add new features directly to production.