



### CUESTIONARIO AUTOEVALUACIÓN TEMA 3

## INGENIERÍA DEL SOFTWARE CON MATLAB: GESTIÓN DE VERSIONES, PRUEBAS UNITARIAS Y DISTRIBUCIÓN DE CÓDIGO

Responda marcando la opción que considere correcta. Cada pregunta acertada suma 1 punto, cada pregunta fallada resta 0,25 puntos. Las preguntas en blanco no contabilizan.

1. MATLAB admite integración con el sistema de gestión de versiones *ClearCase* de forma nativa
  - a) Verdadero
  - b) Falso
  
2. Si tras tener habilitado e integrado en MATLAB un sistema de control de versiones, lo desactivamos. ¿Qué pasará con los repositorios asociados?
  - a) Se mantendrán
  - b) Se eliminarán
  - c) Depende de la configuración del usuario
  
3. MATLAB se distribuye junto con la última versión de:
  - a) Git
  - b) Apache Subversion
  - c) Visual SourceSafe
  
4. En el sistema *Subversion* es recomendable bloquear un archivo antes de comenzar a editarlo porque...
  - a) ... si no lo hacemos no será posible recuperar la versión anterior a los cambios
  - b) ... *subversión* trabaja sobre un repositorio central y puede haber problemas de edición conjunta
  - c) El bloqueo de archivos no es propio de *Subversion* sino de *Git*
  
5. Para poder integrar *Git* con MATLAB es preciso primero instalar este software en el sistema
  - a) Verdadero
  - b) Falso



6. El símbolo con el que MATLAB indica que hay cambios que notificar al repositorio común en *Git* es:
  - a) Un cuadrado azul
  - b) Un cuadrado verde
  - c) Un círculo azul
  - d) Un círculo verde
  
7. Hasta la aparición en el año 2013 de un marco propio, las pruebas en MATLAB se efectuaban, sobre todo, con marcos basados en:
  - a) Doctest
  - b) xUnit
  - c) MATLAB xUnit Test Framework
  
8. El marco de pruebas propio de MATLAB se llama:
  - a) MATLAB xUnit Test Framework
  - b) MATLAB Unit Testing
  - c) JUnit for MATLAB
  
9. Las pruebas basadas en clases se emplean en MATLAB para escribir pruebas de gran complejidad
  - a) Verdadero
  - b) Falso
  
10. En las pruebas basadas en scripts, solo se permite una función de comprobación. Ésta es:
  - a) isEqual()
  - b) runtests()
  - c) assert()



11. Al ejecutar el código

```
assert (salida == esperado, 'Mensaje')
```

la cadena de caracteres 'Mensaje' se mostrará cuando:

- a) No se verifique la igualdad `salida == esperado`
- b) Se verifique la igualdad `salida == esperado`

12. Al ejecutar el código

```
assert (3 == 2 'Mensaje');  
disp('Finalizado');
```

como un script ordinario, ¿se llegará a imprimir el mensaje 'Finalizado'?

- a) Si
- b) No
- c) Depende de la configuración

13. En las pruebas basadas en funciones de MATLAB, los casos de prueba se codifican en forma de:

- a) Funciones principales
- b) Funciones anidadas
- c) Funciones locales

14. En caso de tener que cambiar el *path* del entorno MATLAB antes de ejecutar los casos de prueba, y emplear pruebas basadas en funciones, esto se hará dentro de la función:

- a) `setupOnce`
- b) `teardownOnce`
- c) `setup`
- d) `teardown`

15. En el caso de querer detener una batería de pruebas si una condición no se verifica, emplearíamos las siguientes funciones de calificación:

- a) Verificaciones
- b) Supuestos
- c) Aseveraciones
- d) Aseveraciones fatales



16. Si se desea distribuir un algoritmo escrito en M en forma de librería cerrada (de forma que el código fuente no quede accesible) debe hacerse...
- a) ...en forma de librería dinámica *.dll*
  - b) ...en forma de librería Java *.jar*
  - c) No es posible realizar tal distribución al ser M un lenguaje compilado Just-in-time
17. La herramienta que permite compilar aplicaciones MATLAB para su distribución en forma ejecutable es:
- a) MATLAB Coder
  - b) Deploytool
  - c) MATLAB Compiler