Formación financiada por:

XUNTA DE GALICIA | CONSELLERÍA DE EMPREGO, COMERCIO E EMIGRACIÓN

Financiado pola Unión Europea NextGenerationEU

GOBIERNO DE ESPAÑA | MINISTERIO DE EDUCACIÓN, FORMACIÓN PROFESIONAL Y DEPORTES

Plan de Recuperación, Transformación y Resiliencia

Testing de Software y TDD

Teleformación • 40 horas de duración

CLUSTER TIC GALICIA

Testing de Software y TDD

Objetivos principales del curso

- Aprende la importancia del testing en las prácticas CI/CD y en el desarrollo ágil
- Interioriza los principios de testing.
- Aprende a escribir test unitarios y de integración con JUnit 5
- Aprende cuándo y cómo usar dobles de prueba con EasyMock y con Mockito
- Aprende desarrollo dirigido por tests como técnica de diseño
- Pon en práctica el proceso ATDD y TDD
- Aprende a escribir test de funcionalidades que acceden a bases de datos
- Conoce y practica los distintos enfoques de testing. Tests unitarios vs tests de integración. De dentro a fuera vs de fuera a dentro.

Conocimientos necesarios del alumno y requisitos técnicos:

- Conocimientos de programación orientada a objetos en cualquier lenguaje
- Cuenta en Github
- Un ordenador personal. Puesto que gran parte del curso consiste en clases prácticas de programación es mucho mejor un PC que un móvil para seguir las clases prácticas

Metodología:

El curso se compone de clases pregrabadas en video donde el profesor desarrolla tanto los conceptos teóricos como prácticos. Por tanto, no hay horario de clases, puedes consultar las lecciones en cualquier momento.
Los alumnos desarrollarán ejercicios en los que podrán aplicar todo lo aprendido durante el curso. El profesor corrige los ejercicios y les da feedback acerca de cómo lo han hecho.
El curso supone unas 40 horas de estudio por parte de los alumnos aproximadamente. Videos, lecturas y tiempo de resolución de práctica.
Las dudas se consultan al profesor mediante foro y el profesor responde en 24 horas laborables.
Finalizado el plazo, el profesor deja de participar en el curso, pero los alumnos siguen teniendo acceso a los materiales.

CLUSTER TIC GALICIA

Temario

¿Qué aprenderás con nosotros?

1. Introducción

1.1 Introducción
1.2 ¿Por qué hacer tests?
1.3 Cómo seguir el curso
1.4 Sobre el entorno de desarrollo
1.5 Instalación del entorno en Windows
1.6 Instalación del entorno en Linux
1.7 Vista jerárquica de paquetes en Eclipse)

2. Base teórica

2.1 ¿Qué aprenderemos en esta sección?
2.2 Introducción al agilismo
2.3 El testing en las prácticas CI/CD
2.4 Conceptos de TDD y ATDD
2.5 Tipos de tests
2.6 Principios de testing

3. Test unitarios

3.1 ¿Qué aprenderemos en esta sección?
3.2 Introducción a JUnit 5
3.3 ¿Qué es un tests unitario y cómo se escribe?
3.4 Plantilla Given-When-Then para Eclipse
3.5 Escribir los primeros tests unitarios – TextNormalizer
3.6 Introducir bugs en TextNormalizer
3.7 Ejercicio: probar EmailValidator. Enunciado
3.8 Ejercicio: probar EmailValidator. Solución
3.9 Pruebas de lanzamiento de excepciones.
3.10 Ejercicio: probar lanzamiento de excepción en EmailValidator. Enunciado
3.11 Ejercicio: probar lanzamiento de excepción en EmailValidator. Solución
3.12 Introducción a los tests parametrizados de JUnit 5
3.13 Tests parametrizados – Mejorando EmailValidatorTest
3.14 Tests parametrizados – Mejorando TextNormalizerTest
3.15 Test parametrizados – Probar InvoiceLine con @MethodSource
3.16 Usar Clock en lugar de Instant.now o new Date()
3.17 El término “test fixture”
3.18 Código fuente de la sección Test Unitarios

Temario

4. Desarrollo dirigido por test (TDD)

4.1 ¿Qué aprenderemos en esta sección?

4.2 El algoritmo TDD

4.3 TDD – Beneficios e inconvenientes

4.4 ATDD – Repaso

4.5 ATDD – TDD – Proceso en práctica (Contador de palabras) – Parte 1

4.6 ATDD – TDD – Proceso en práctica (Contador de palabras) – Parte 2

4.7 La importancia de interiorizar el proceso y dar pasos pequeños

4.8 Mejorando Product con TDD – parte 1

4.9 Mejorando Product con TDD – parte 2

4.10 Mejorando InvoiceLine con TDD

4.11 Ejercicio del triángulo – Enunciado

4.12 Ejercicio del triángulo – Solución

4.13 Aplicar TDD en nuestro día a día

5. Dobles de prueba

5.1 ¿Qué aprenderemos en esta sección?

5.2 Dobles de prueba – Introducción

5.3 Dobles de prueba – Ejemplo básico

5.4 Tipos de dobles de prueba

5.5 Cuándo usar un doble de prueba

5.6 Dobles de prueba – Frameworks

5.7 EasyMock – Getting Started

5.8 Mockito – Getting Started

5.9 CurrencyConverter con Mockito – parte 1

5.10 CurrencyConverter con Mockito – parte 2

5.11 CurrencyConverter con Mockito – Validación de estado vs comportamiento

6. Test de integración

6.1 ¿Qué aprenderemos en esta sección?

6.2 Test de integración – Introducción

6.3 Test de integración en el desarrollo ágil con TDD

6.4 Preparar base de datos H2

6.5 Probar funcionalidades que requieran acceso a BBDD – Problemática

6.6 Tests de acceso a BD – Solución con Spring Test

Temario

6.7 Tests de acceso a BD – Introducción a Database Rider
6.8 Tests de acceso a BD – Introducción a JDBDT (opcional)
6.9 Tests de acceso a BD – Database Rider – Ejemplo práctico
6.10 Testing con BBDD en memoria – Ventajas e inconvenientes
6.11 Tests Unitarios vs Tests de Integración – Enfoques

7. Proyecto de ejemplo

7.1 ¿Qué haremos en esta sección?
7.2 Aplicación parking – Descripción problema
7.3 Esqueleto del proyecto
7.4 Emitir un ticket – Test de aceptación
7.5 Emitir un ticket – Parte 1 – Preparamos el test de integración
7.6 Emitir un ticket – Parte 2 – Implementamos y metemos un bug
7.7 Emitir un ticket – Parte 3 – Resolvemos bug y terminamos la historia
7.8 Calcular importe a pagar – Tests de aceptación
7.9 Calcular importe a pagar – Domain – Parte 1
7.10 Calcular importe a pagar – Domain – Parte 2
7.11 Calcular importe a pagar – Infrastructure y application
7.12 Aplicación Parking – Resumen final

Perfil del docente y empresa proveedora

Sergio García Trapiello

Ingeniero de software. Llevo desde 2009 construyendo todo tipo de productos software en distintos lenguajes (portales y aplicaciones Web, aplicaciones de escritorio, apps móviles, sistemas backend, Web Services y APIs, frameworks y arquitecturas de desarrollo, etc.)
Estudié Ingeniería Técnica Informática en la Universidad de Oviedo y después cursé el Master de Ingeniería Web en la misma Universidad. Desde entonces he trabajado como ingeniero y desarrollador en distintas empresas y distintos roles. Y también por mi cuenta como freelance.
A nivel profesional, lo que me apasiona es resolver problemas con el foco puesto en la calidad del software. Empecé a aplicar TDD allá por el año 2012 y desde entonces soy un convencido del enfoque TDD y ATDD. Me encanta el proceso de identificar las necesidades, limar las ambigüedades traduciendo los requisitos a test de aceptación y luego empezar a desarrollar haciendo TDD (de hecho creo que ya no sé hacerlo de otra forma xd).

TrainingIT | https://www.trainingit.es/

TrainingIT es una iniciativa para ofrecer formación especializada IT de alta calidad. Descubre nuestros cursos online creados por especialistas en sus materias. Queremos formar a los mejores profesionales para que no se diga que en España no hay talento.

CLUSTER TIC GALICIA

Resumen de características del curso

Necesitarás dedicarle **4-5 horas semanales** (~40 horas en total).

Nivel: Iniciación-Intermedio.

Curso 100% en **castellano.**

Acceso directo al tutor para resolver todas tus dudas a través de foro.

Diploma emitido por la Consellería de Emprego, Comercio e Emigración después de revisar que el alumno ha cumplido con los requisitos exigidos para superar el curso con la cualificación de APTO. El Clúster TIC Galicia no es responsable de la fecha de emisión de dicho diploma.

CLUSTER TIC GALICIA

Formación financiada por:

XUNTA DE GALICIA | CONSELLERÍA DE EMPREGO, COMERCIO E EMIGRACIÓN

Financiado pola Unión Europea NextGenerationEU

GOBIERNO DE ESPAÑA | MINISTERIO DE EDUCACIÓN, FORMACIÓN PROFESIONAL Y DEPORTES

Plan de Recuperación, Transformación y Resiliencia

¿Tienes dudas?
Contacta con nosotros:

Tel.: +34 881 939 651

E-mail: info@clusterticgalicia.com

CLUSTER TIC GALICIA