

Formación en Desarrollo de Software

Edición 2019



Formación Certificada en Desarrollo de Software	5
iSQI - Certified Agile Test Driven Development (TDD)	5
DASA DevOps Fundamentals	6
Cursos de Desarrollo de Software	7
Introducción a DevOps	7
Introducción a Test Driven Development (TDD)	8
Testing para Desarrolladores	9
Workshop de Desarrollo de Software	10
Herramientas en la Gestión de Configuración	10



El objetivo de este programa de formación es capacitar a los equipos de desarrollo software en metodologías, mejores prácticas y herramientas para crear software con la combinación adecuada de calidad y productividad.

Formación Certificada en Desarrollo Software

iSQI - Certified Agile Test Driven Development (TDD)

Objetivos del curso:

El objetivo de esta formación es conseguir que los desarrolladores apliquen las mejores prácticas de desarrollo de software ágil. Después de la capacitación, podrán conocer las diferencias entre código pobre y código de calidad, siendo capaces de detectar defectos de diseño y sugerir soluciones mejorando el código en términos de facilidad de mantenimiento. Serán conscientes de las consecuencias de realizar código en un único bloque o fragmento, dándose cuenta de lo importante que es preocuparse por el código que escriben. Algunos de los participantes iniciarán la formación con la creencia de que el código está escrito para ordenadores, pero finalizarán dándose cuenta de que la calidad del código está en escribirlo para los lectores, ya que son los que van a mantenerlo. “Las máquinas actuales no son capaces de quejarse de la mala calidad del código”.

Los participantes comprenderán TDD, un método profesional para desarrollar software fiable fácil de mantener. También podrán utilizar TDD durante el desarrollo de pequeñas funcionalidades y problemas concretos en ámbitos bien definidos. Podrán mejorar en sus prácticas cotidianas con una base sólida en TDD. Las personas que se certifiquen adquirirán las habilidades necesarias para hacer frente a proyectos de gran envergadura y alcances indefinidos.

Junto con la información teórica precisa, este curso se centra principalmente en tratar TDD en un nivel práctico. Un ejemplo de cómo se logra esto es un ejercicio integral, en el que los participantes aprenden a través de sus propios errores, bajo la supervisión de un experto. Al finalizar del curso, los participantes serán capaces de:

- Hacer evaluaciones de la calidad del código de producción y de la calidad de las pruebas automatizadas.
- Mejorar la calidad del código existente sin alterar su comportamiento.
- Comportamientos dirigidos por pruebas.
- Producir baterías de pruebas de alta calidad y producir código legible.
- Reconocer y evitar la sobrerregulación (gold-plating).
- Evaluar si se introdujo deuda técnica en el desarrollo así como sus consecuencias.
- Estar cualificado para aprender las técnicas más avanzadas de manera efectiva.

Dirigido a:

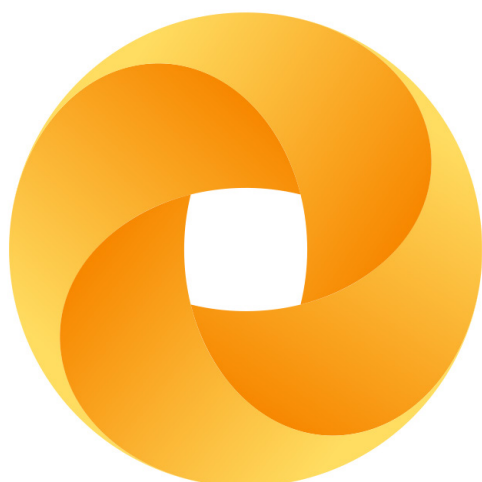
Este es un curso para desarrolladores que aspiran a ser más eficaces y a mejorar continuamente sus habilidades. Básicamente los participantes deben querer aportar más valor a sus usuarios mediante la entrega de mejores productos.

Pre-requisitos:

Como pre-requisito de este curso, los desarrolladores deben tener capacidad de codificar en Java o C#, no siendo obligatorio un conocimiento profundo en estos lenguajes. Basta un conocimiento básico de programación orientada a objetos, por ejemplo, la comprensión de los conceptos de herencia y composición. No es necesario la experiencia previa con los marcos o modelos de pruebas, pruebas automatizadas o TDD.

Examen: requerido para la obtención de la certificación. El examen se realiza el último día del curso al finalizar el temario con una duración de 4 horas y 30 minutos.

Duración: 16 horas distribuidas en 2 días.



TEMARIO DEL CURSO DIA 1

- Introducción a los tests unitarios
- Refactoring
- Naming and duplication
- SOLID design principles
- Design Patterns
- Introducción a TDD

TEMARIO DEL CURSO DIA 2

- Practicando TDD con state-based validation
- Diseño emergente y arquitectura
- Integration tests
- Mock objects, test stubs, test spies
- Practicando TDD con interaction-based tests
- Test doubles frameworks
- End-to-end tests
- Conclusiones

DASA DevOps Fundamentals

El curso DevOps Fundamentals ofrece una introducción extensa a los principios centrales de Agile DevOps que cubren las competencias de conocimiento y habilidades esenciales que han sido definidas por DevOps Agile Skills Association (DASA). Durante la formación se describen los conceptos y terminología clave de DevOps, estudios de casos reales, ejemplos y debates interactivos y ejercicios exhaustivos en cada módulo.

Objetivos del curso:

- Conocer los orígenes de DevOps, sus conceptos y principios básicos. Saber lo que significa para usted como profesional y para su organización.
- Comprender la esencia y elementos clave de una cultura DevOps.
- Conocer los aspectos importantes al crear una cultura DevOps y sus modelos operativos.
- Comprender la necesidad de los equipos autónomos y el gobierno dentro de los equipos de DevOps.
- Comprender el impacto de DevOps en la arquitectura con respecto a la implementación.
- Comprender Agile, Scrum y Kanban y cómo estas prácticas se relacionan entre sí.
- Comprender cómo se relacionan los procesos ITSM con las prácticas en una cultura DevOps.
- Comprender cómo se usa Lean para optimizar los procesos.
- Saber cómo proporcionar un mapa de flujo de valor para un proceso determinado.
- Comprender la forma de obtener ideas nuevas e innovadoras.
- Conocer el impacto de la automatización en los procesos de entrega de software.
- Comprender los beneficios y los principios básicos de la entrega continua (CD).
- Describir los principios clave del Cloud para las organizaciones de DevOps.
- Conocer la relevancia de la monitorización y el registro de datos en DevOps.

Dirigido a:

- Personas involucradas en el desarrollo TI, operaciones de TI o gestión de servicios de TI.
- Las personas cuyo papel está relacionado con DevOps y la entrega continua, como por ejemplo los siguientes roles TI: Ingenieros de DevOps, Product Owners, Especialistas en integración continua, Gerentes y responsables de operaciones, Responsables de incidencias y cambios, Administradores del sistema, Administradores de red, Gerentes y responsables de negocio, Arquitectos de automatización, Arquitectos empresariales.

Duración: 24 horas distribuidas en 3 días.

Pre-requisitos: Se recomienda estar familiarizado con los principios Agile, Scrum, Lean e ITSM.

Examen: Tras la finalización de la formación, el asistente podrá realizar un examen para la obtención del certificado DASA DevOps Fundamentals siguiendo el esquema de certificación DevOps Agile Skills Association (DASA).

TEMARIO DEL CURSO:

<p>MÓDULO 1 - INTRODUCCIÓN DEL CURSO</p> <ul style="list-style-type: none"> • Descripción general y objetivos del curso. • Relación entre el Modelo de Competencia y los Módulos del Curso. • Agenda y Tipo de actividades. • Examen. • Material del curso y glosario técnico. 	<p>MÓDULO 2 - INTRODUCCIÓN A DEVOPS</p> <ul style="list-style-type: none"> • Origen de DevOps. • Conceptos básicos de DevOps. • DevOps Agile Skills Association (DASA).
<p>MÓDULO 3 - INTRODUCCIÓN DEL CURSO</p> <ul style="list-style-type: none"> • Esencia de una cultura DevOps. • Elementos clave de DevOps. • Implementación de una cultura DevOps. 	<p>MÓDULO 4 - INTRODUCCIÓN A DEVOPS</p> <ul style="list-style-type: none"> • Modelo Organizacional. • Equipos autónomos. • Arquitectura para DevOps.
<p>MÓDULO 5 - PROCESOS</p> <ul style="list-style-type: none"> • Conceptos básicos del proceso. • DevOps en relación con ITSM. • Agile y Scrum. • 12 Principios del Manifiesto Ágil. • Optimización de procesos con Lean. • Optimización del valor empresarial y análisis comercial utilizando Story Mapping. 	<p>MÓDULO 6 - AUTOMATIZACIÓN</p> <ul style="list-style-type: none"> • Conceptos de Automatización. <ul style="list-style-type: none"> • Automatización para la entrega de software. • Conceptos básicos de entrega continua. • Conceptos de automatización de entrega continua. • Temas de enfoque de Automatización de entrega continua. • Data Center Automation. <ul style="list-style-type: none"> • Surgimiento de la tecnología y los principios de la nube. • Conceptos de servicios en la nube en una organización DevOps. • Conceptos de Aprovisionamiento Automatizado. • Características del producto de la plataforma y vencimiento de la aplicación.
<p>MÓDULO 7 - MEDIDA Y MEJORA</p> <ul style="list-style-type: none"> • Importancia de la medición. • Elegir las métricas correctas. • Monitorización y registro de información. 	

Cursos de Desarrollo Software

Introducción a DevOps

Objetivos del curso: conocer los procesos, patrones y técnicas que ofrece la filosofía DevOps para reducir el tiempo medio de ciclo, esto es, el tiempo transcurrido desde la concepción de una idea hasta su puesta en producción. A la finalización de la acción formativa, los asistentes podrán:

- Saber lo que es DevOps y qué patrones seguir para poder entregar el SW con plazos menores y mayor calidad.
- Aplicar técnicas para mejorar la colaboración entre los equipos de desarrollo y operaciones.
- Unificar procesos e incentivos con el objetivo de tener unas metas compartidas para todos los equipos.
- Integrar varias herramientas para soportar el ciclo de vida completo de un proyecto.
- Agilizar el proceso de entrega de SW.

Dirigido a: Todos los perfiles de los equipos de desarrollo y explotación de Sistemas Software: Ingenieros Software, Programadores, Responsables y Técnicos de los sistemas de software en producción, Responsables de mejora de procesos TI.

Duración: 16 horas, distribuidas en 2 días.

Pre-requisitos: conocimientos de programación y construcción de Software.

TEMARIO DEL CURSO:

CAPÍTULO 1 - IDENTIFICACIÓN DE LA PROBLEMÁTICA O SITUACIÓN ACTUAL

- Existencia de silos.
- Objetivos distintos entre distintos equipos.
- Procesos optimizados pero no alineados.
- Metodologías ágiles no aplican al equipo de operaciones.
- Métricas usadas que entran en conflicto.
- Actualizaciones poco frecuentes y batch size elevado.
- Tiempo de ciclo muy alto.
- Búsqueda de culpables si algo no va bien.

CAPÍTULO 2 - BENEFICIOS QUE SE PUEDEN OBTENER APLICANDO DEVOPS

- Liberar recursos gracias a la automatización.
- Tener objetivos y métricas compartidos por todos los departamentos implicados en el proceso de delivery.
- Evitar la existencia de silos.
- Gestionar el cambio de forma unificada.
- Tener un batch size reducido.
- Explotar antes el SW desarrollado.
- Mitigar el riesgo en las actualizaciones.
- Alta reducción en la necesidad de mantener complejos procesos de rollback.
- Eliminar tareas manuales (proceso de releasing y provisioning automatizado).

CAPÍTULO 3 - CÓMO CONSEGUIR LOS BENEFICIOS DE APLICAR DEVOPS

- Describiendo patrones para compartir y alinear metas e incentivos.
- Redefiniendo el concepto de calidad y aplicando técnicas para mejorarla.
- Automatizando los procesos de release:
 - Delivery pipelines.
 - Creación y uso de Baselines.
 - Versionado de artefactos.
 - Packaging tools.
 - Independizar configuración.
 - Task-based development.
 - Almacén de binarios y trazabilidad a tareas.
 - Scripts de BBDD y gestión de ChangeSets.
- Gestionando la infraestructura como código.
 - Provisioning con Puppet.
- Definiendo pruebas de aceptación:
 - Acordando criterios de aceptación.
 - Automatizando con Thucydides.
 - ATDD.

CAPÍTULO 4 EJEMPLO PRÁCTICO

- Se utilizarán varias herramientas OpenSource o de uso libre para modelar el proceso de delivery de SW y hacerlo de forma automatizada, eliminando tantos procesos manuales como sea posible.
- Se construirá una tool-chain utilizando:
 - JIRA, FishEye, Subversion, Eclipse, TestLink, Jenkins, Maven, Artifactory, XL Release, XL Deploy, Puppet, MCollective.
- Se utilizará una aplicación de ejemplo sobre la que se crearán nuevas versiones (código, base de datos, configuración, infraestructura) y se promocionarán de forma automatizada entre distintos entornos.





Introducción a Test Driven Development (TDD)

Un desarrollador profesional necesita usar métodos y herramientas profesionales. TDD es un método de desarrollo que produce código de calidad y excelentes baterías de pruebas automáticas cuando se practica correctamente.

TDD es una práctica del método XP (eXtremeProgramming). Las prácticas no tienen sentido si no se entienden los valores y principios que promueven. Por eso en realidad este es un curso de Introducción a XP. Además de practicar TDD, practicaremos PairProgramming e incluso CollectiveCodeOwnership.

Objetivos del curso:

La forma de impartir el curso es programando prácticamente todo el tiempo con la supervisión y la guía del experto. La diferencia con leer libros o ver vídeos es que hay a tu lado un experto para corregir tus errores y acelerar el proceso de aprendizaje, minimizando riesgos a la hora de luego poner en práctica el método en el día a día.

Dirigido a: Este es un curso para personas que sepan programar. Fundamentalmente programadores pero también testers que quieran conocer más sobre calidad de código y pruebas automáticas. Lo que hace falta es venir con ganas y la mente abierta. Si nunca has probado TDD antes, te sorprenderá.

Duración: 16 horas, distribuidas en 2 días.

Pre-requisitos:

Cada alumno debe asistir al curso con su portátil y las herramientas de desarrollo que use habitualmente, ya instaladas. A ser posible que también tenga instalado Git o Mercurial o ambas.

Se trabaja en pares con lo cual es buena idea crear un usuario nuevo para el curso, como para poder rotar de unas máquinas a otras sin estar preocupados por privacidad de la cuenta.



TEMARIO DEL CURSO:

CAPÍTULO 1 INTRODUCCIÓN

- Introducción a XP: Valores, Principios y Prácticas
- Introducción a los tests unitarios
- Refactoring
- Naming and duplication
- Less Surprise Principle y SOLID design principles
- DesignPatterns
- Introducción a TDD Agil, etc.

CAPÍTULO 2 MÉTRICAS DEL SOFTWARE

- Practicando TDD con state-based validation
- Diseño emergente y arquitectura
- Integration tests
- Mock objects, test stubs, test spies
- Practicando TDD con interaction-based tests
- Test doubles frameworks
- End-to-end tests
- Conclusiones



Testing para Desarrolladores

Objetivos del curso: Los asistentes a esta sesión formativa tras la realización de la misma serán capaces de:

- Comprender la utilidad y necesidad de las pruebas integradas en el proceso general de desarrollo software, y no sólo como una actividad al final de dicho proceso.
- Conocer las actividades de prueba que le corresponde al desarrollador y cómo abordarlas.
- Entender los conceptos clave y fundamentales para realizar de forma efectiva los casos de prueba que componen el plan de pruebas de una aplicación o sistema.
- Comprender los aspectos que intervienen en una estrategia de calidad de producto software.
- Conocer algunas prácticas y recomendaciones del mundo del desarrollo software (automatización de las pruebas, automatización del desarrollo, integración continua, TDD).

Durante la formación se tratará de qué, por qué, cómo, con qué, y cuándo probar el software por el desarrollador, y se compartirán experiencias relacionadas con el testing realizado por desarrolladores.

Dirigido a:

- Ingenieros Software, Arquitectos Software, Programadores y profesionales de desarrollo de software y aplicaciones.

Duración: 8 horas distribuidas en 1 día.

Pre-requisitos: es imprescindible contar con experiencia en proyectos de desarrollo de software.

TEMARIO DEL CURSO:

1.- LA RESPONSABILIDAD DEL DESARROLLADOR DE LA CALIDAD DEL SOFTWARE
¿qué debo hacer?

2.- REQUISITOS FUNCIONALES Y NO FUNCIONALES DEL SOFTWARE. CARACTERÍSTICAS TÉCNICAS
¿qué probar?

3.- HERRAMIENTAS DEL TESTING PARA EL DESARROLLADOR
¿con qué probarlo?

4.- TÉCNICAS BÁSICAS DE PRUEBA. METODOLOGÍAS.
¿cómo probarlo?

5.- EL TESTING EN DESARROLLO PREDICTIVO VS AGILE
¿quién prueba?

6.- LA RESPONSABILIDAD DEL TESTER
¿cómo me ayuda?

7.- EL FUTURO DEL TESTING Y LA CALIDAD DEL SOFTWARE
¿y después qué?

8.- AUTOMATIZACIÓN COMO RESPUESTA A LA COMPLEJIDAD



Workshop de Desarrollo Software

Herramientas de Gestión de Configuración

Este workshop sobre herramientas de gestión de configuración del software es aplicable a las principales herramientas comerciales y open source.

Objetivos del Workshop:

- Comprender y valorar el proceso de gestión de configuración dentro del ciclo de vida del desarrollo de software.
- Comprender las ventajas del uso de herramientas especializadas en el proceso de gestión de configuración.

Dirigido a:

- Responsables y especialistas en Desarrollo software,
- Responsables y especialistas en Pruebas,
- Analistas e Ingenieros de Desarrollo,
- Programadores,
- Técnicos de Sistemas (Explotación).

Duración: 8 horas, distribuidas en 1 día.

TEMARIO DEL CURSO:

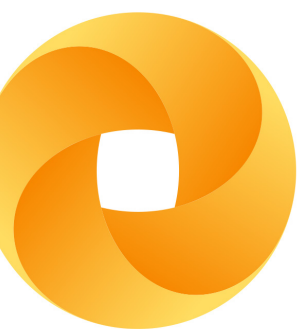
CONTENIDOS DEL CURSO

- La gestión de configuración dentro del ciclo de vida del Software. Conceptos y terminología.
- Descripción general de la arquitectura de la herramienta y la organización de equipos para su utilización.

- Batería de casos Prácticos que simulan un amplio abanico de situaciones reales, sobre actividades como:
 - Gestión de líneas base.
 - Check-in y check-out.
 - Procesos y herramientas relacionadas.
 - Gestión de cambios.
 - Informes de estado.







C/ Santa Leonor 65, Edificio C Planta 4ª
28037 Madrid (España)
Teléfono: +34 91 144 06 00
Fax: +34 91 304 49 15
Email: formacion@mtp.es