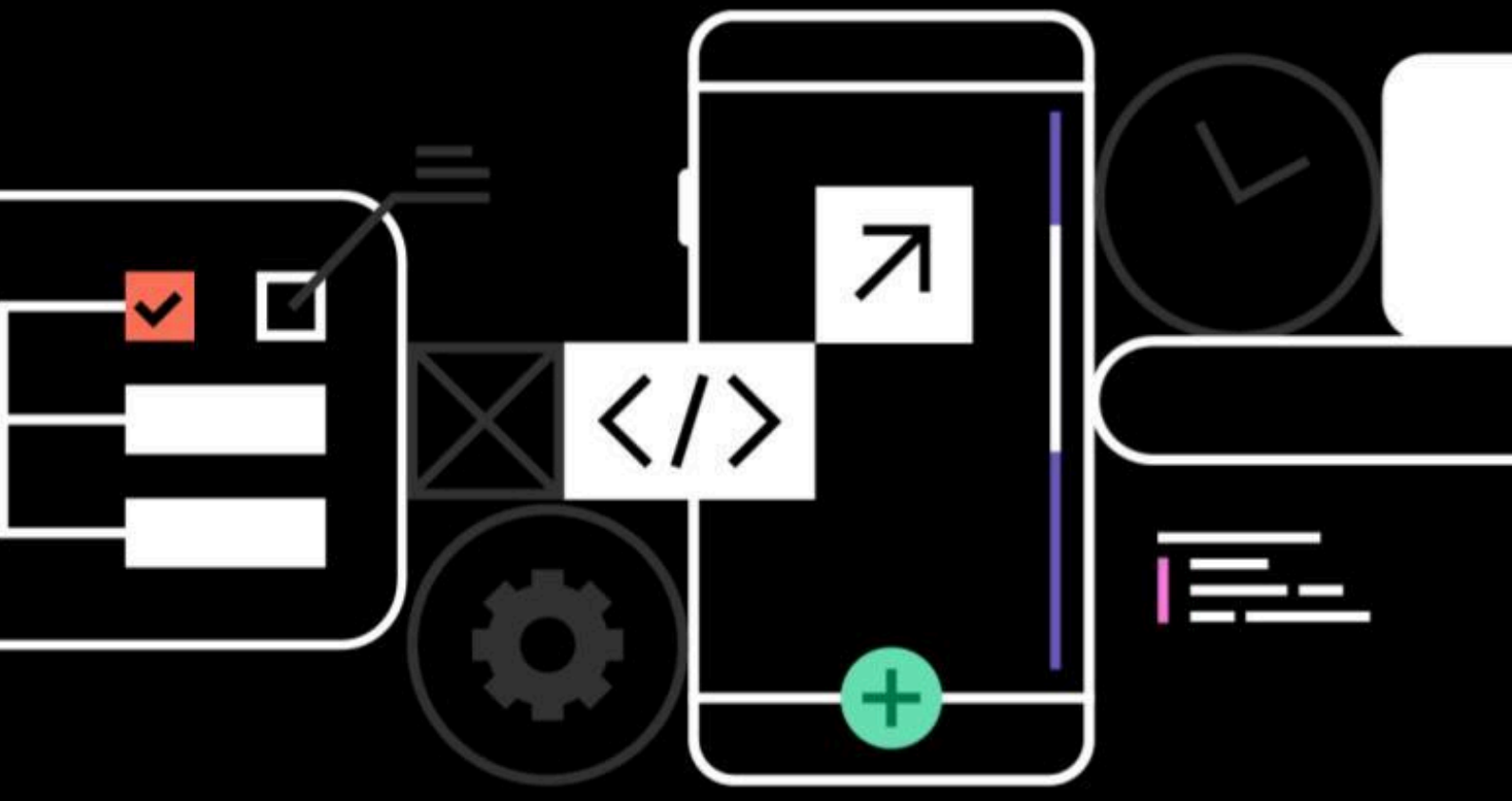
DIGITALHOUSE

Testing QA Automatizado

Programa



Testing QA Automatizado

Fundamentación

Esta certificación proporciona una formación integral en las áreas más relevantes del desarrollo de software, abarcando desde los fundamentos del mundo IT hasta técnicas avanzadas de testing automatizado. A través de un enfoque práctico, las personas cursantes aprenderán los principios básicos de informática, redes y segurida, asegurando una sólida comprensión del entorno tecnológico actual. Además, profundizarán en la programación imperativa con JavaScript y la programación orientada a objetos en Java, cubriendo desde la solución de problemas cotidianos hasta el desarrollo de software eficiente y sostenible mediante el uso de patrones de diseño y buenas prácticas. La certificación también explora los principios de calidad de software, proporcionando las herramientas necesarias para garantizar productos robustos y confiables mediante técnicas de testing manual y automatizado. La metodología Design Thinking se incorpora para fomentar un enfoque centrado en el usuario, impulsando la innovación a través de la colaboración y la iteración rápida. En conjunto, esta certificación prepara a las personas para abordar los desafíos del desarrollo de software moderno con un enfoque completo y adaptable.

Objetivos de aprendizaje

Al completar la certificación, las personas cursantes podrán desarrollar programas eficientes en JavaScript y aplicar el paradigma orientado a objetos en Java. Serán capaces de implementar técnicas de testing de software, tanto manuales como automatizadas, utilizando herramientas clave. Además, dominarán la metodología Design Thinking para crear soluciones centradas en el usuario y gestionarán procesos de calidad en todas las etapas del ciclo de desarrollo de software.

Contenidos

Introducción al mundo IT

Se espera que los estudiantes puedan:

- Obtener conocimientos técnicos de informática general.
- Brindar las herramientas necesarias para que el estudiante pueda utilizar un sistema de control de versiones utilizando la terminal (CLI) a lo largo de toda su carrera.

- Comprender las estructuras y tecnologías que utiliza el hardware y software de una computadora.
- Entender los conceptos básicos de las redes de interconexión profundizando en cómo funciona Internet y su entorno de tecnologías.
- Identificar los aspectos generales de la seguridad de los sistemas informáticos.
- Brindar al futuro profesional conocimientos acerca de la importancia de la informática en la sociedad, los códigos de ética, moral y práctica profesional.
- Aplicar las herramientas y marcos del agilismo en casos prácticos para experimentar cómo funcionan.

Programación Imperativa con JavaScript

Se espera que los estudiantes puedan:

- Desarrollar programas que resuelvan problemas cotidianos.
- Ampliar sus capacidades de programación de manera autónoma.
- Adquirir habilidades de pensamiento computacional para resolver problemas y diseñar algoritmos.
- Manejar las estructuras básicas de JavaScript y su entorno de desarrollo (Node.js y Visual Studio Code).
- Comprender y aplicar variables, tipos de datos, operadores, funciones y control de flujo.
- Trabajar con estructuras de datos complejas como arrays y objetos literales.
- Implementar estructuras de control como ciclos y condicionales.
- Desarrollar aplicaciones completas con proyectos integradores.
- Utilizar métodos avanzados de arrays y funciones, como *arrow functions* y *callbacks*.
- Aplicar algoritmos de búsqueda y ordenación.
- Emplear buenas prácticas de desarrollo de software.

POO en Java

Se espera que los estudiantes puedan:

- Desarrollar la capacidad de programar en diversas áreas del desarrollo de software utilizando el paradigma orientado a objetos.
- Comprender y aplicar los fundamentos de la programación orientada a objetos en Java.
- Conocer y utilizar patrones de diseño para crear software eficiente y sostenible.

- Crear clases, objetos, interfaces y manejar relaciones entre clases como la herencia y el polimorfismo.
- Aplicar principios SOLID en el diseño de software.
- Manejar excepciones y utilizar colecciones de datos como arrays, sets, y mapas.
- Implementar patrones de diseño como Factory, Singleton, State, Composite, Observer y Strategy.
- Integrar y aplicar los conocimientos adquiridos a través de prácticas y proyectos

Testing en Calidad de Software

Se espera que los estudiantes puedan:

- Conocer y comprender la importancia del testing y la calidad del software.
- Comprender y profundizar acerca del ciclo de vida del proceso testing.
- Comprender y ejercitar diferentes técnicas de prueba.
- Crear y ejecutar casos de pruebas de manera adecuada.
- Comprender cómo reportar errores y la importancia de los mismos.
- Conocer los diferentes niveles y tipos de prueba para determinar la conveniencia de su utilización.
- Conocer y dominar herramientas de automatización que faciliten y agilicen el proceso de pruebas de calidad de software.

Design Thinking

Se espera que los estudiantes puedan:

- Conocer la importancia de desarrollar metodologías de trabajo en equipo enfocadas en los usuarios para alcanzar resultados exitosos.
- Comprender los beneficios de la agilidad en el desarrollo de soluciones, reconociendo el error temprano como un aliado de la innovación.
- Conocer las bases y pilares de la metodología Design thinking, sus etapas y principales herramientas aplicables en cada una de ellas.
- Aplicar la metodología en casos prácticos a lo largo de la cursada para experimentar los pasos de Design thinking.

Testing Automatizado

Se espera que los estudiantes puedan:

- Conocer e implementar procesos de calidad dentro de un proceso de desarrollo de software.
- Reconocer casos de pruebas automatizables.
- Implementar un proyecto de automatización con Selenium y JAVA
    - Iniciar un proyecto de automatización.
    - Implementar Page Object como patrón de diseño en un lenguaje orientado a objetos.
    - Armado y ejecución de suites dentro de un proyecto.
    - Generación de reportes.
- Implementar un proyecto de automatización con RestAssured y JAVA.
- Definición de procesos de calidad en los distintos momentos de desarrollo tales como:
    - Plan de pruebas.
    - Plan para la liberación de nuevas versiones.
    - Mantenimiento de la prueba - Procesos de revisión.
    - Almacenamiento de pruebas .
    - Métricas y Reportes.
- Principios para implementar testing en dispositivos móviles.