



# Certified Tech Developer

The Ultimate Degree

---

## Materia: Testing I

---

### Fundamentación

Todo el tiempo hablamos de calidad, pero ¿sabemos realmente qué es la calidad? Para comprender este término podemos basarnos en la definición formal: *“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”* (IEEE Std 610, 1990).

En la industria tecnológica, la calidad cobra importancia en el objetivo principal de todo equipo de trabajo, este es, la entrega de manera eficiente y en un corto plazo de un producto que tenga la mayor calidad posible. Si bien todo el equipo es responsable de la calidad del producto de software, el tester o probador del sistema será la persona encargada de guiar y liderar las actividades relacionadas con la calidad para asegurarse que se cumplen las necesidades del cliente.

Estas actividades están comprendidas en el **ciclo de vida de las pruebas de software**, el cual nos ayuda a organizar el trabajo. Conocer y profundizar acerca de estas actividades nos permitirá entregar un producto de calidad.

### Objetivos de aprendizaje

- 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.



**Certified Tech Developer**

The Ultimate Degree

**DigitalHouse** >  
Coding School

- Comprender y profundizar sobre prueba de componente (TDD vs. BDD).
- Conocer y ejercitar API testing.
- Conocer diferentes frameworks de automatización de pruebas y modelar casos de prueba automáticos.

## Metodología de enseñanza-aprendizaje

Desde Digital House, proponemos un modelo educativo que incluye entornos de aprendizaje sincrónicos y asincrónicos con un enfoque que vincula la teoría y la práctica, mediante un aprendizaje activo y colaborativo.

Nuestra propuesta incluye clases en vivo con tu grupo de estudiantes y docentes, a los que podrás sumarte desde donde estés. Además, contamos con un campus virtual a medida, en el cual encontrarás las clases virtuales, con actividades, videos, presentaciones y recursos interactivos, para realizar a tu ritmo antes de cada clase en vivo.

A lo largo de tu experiencia de aprendizaje en Digital House lograrás desarrollar habilidades técnicas y blandas, como ser el trabajo en equipo, creatividad, responsabilidad, compromiso, comunicación efectiva y autonomía.

En Digital House utilizamos la metodología de “aula invertida”. ¿Qué quiere decir? Cada semana te vamos a pedir que te prepares para la que sigue, leyendo textos, viendo videos, realizando actividades, entre otros recursos. De esta forma, cuando llegues al encuentro en vivo, estarás preparado para abordar el tema de manera más rica.

Utilizamos actividades y estrategias basadas en los métodos participativos y activos para ponerte en movimiento, ya que uno solo sabe lo que hace por sí mismo. Por ese motivo, organizamos las clases para que trabajes en ella de verdad y puedas poner en práctica las distintas herramientas, lenguajes y competencias que hacen a la formación de un programador. Concebimos la clase como espacio de trabajo.

Una de las cuestiones centrales de nuestra metodología de enseñanza es el aprendizaje en la práctica. Por ese motivo, a lo largo de la cursada estarán muy presentes las ejercitaciones, es

decir, la práctica de actividades de diversos tipos y niveles de complejidad que te permitirán afianzar el aprendizaje y comprobar que lo hayas asimilado correctamente. De esta forma, se logra un aprendizaje más significativo y profundo, la asimilación de los conocimientos de manera más eficaz y duradera, relacionar lo aprendido con la realidad de los desarrolladores web, fomentar la autonomía y el autoconocimiento, mejorar el análisis, la relación y la comprensión de conceptos, además de ayudar a ejercitar una multitud de competencias.

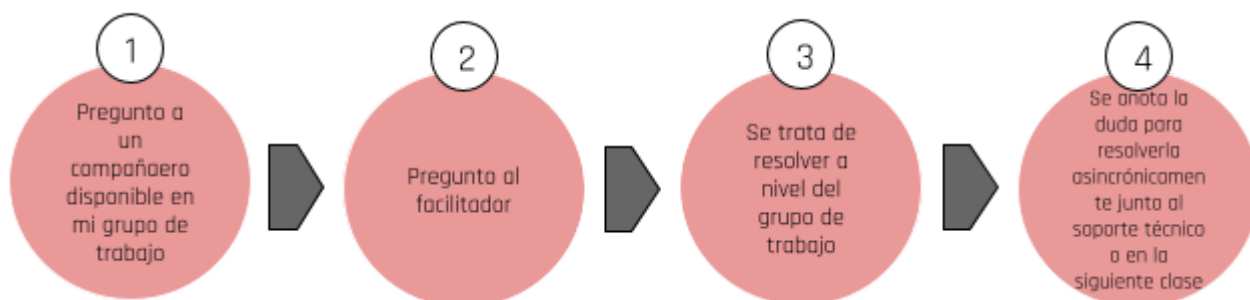
El aprendizaje entre pares es uno de los elementos centrales de nuestra metodología, por eso, en cada clase te propondremos que trabajes en mesas de trabajo junto a tus compañeros, a lo largo de la cursada iremos variando la composición de los grupos para potenciar la cooperación. Lo que se propone es un cambio de mirada sobre el curso en cuestión, ya no se contempla al estudiante transitando su camino académico de manera individual, sino como parte de un equipo que resulta de la suma de las potencialidades de cada uno. La distribución en grupos de trabajo fomenta la diversidad y el aprovechamiento del potencial de cada integrante para mejorar el rendimiento del equipo.

La explicación recíproca como eje del trabajo cotidiano no solo facilita el aprendizaje de los compañeros, sino que sobre todo potencia la consolidación de conocimientos por parte de quien explica. Se promueve la responsabilidad, la autonomía, la proactividad, todo en el marco de la cooperación. Lo que lleva a resignificar la experiencia de aprendizaje y a que la misma esté vinculada con emociones positivas.

El trabajo cooperativo permite entablar relaciones responsables y duraderas, aumenta la motivación y el compromiso, además de promover un buen desarrollo cognitivo y social. La cooperación surge frente a la duda. Si un estudiante tiene una pregunta, le consulta a algún miembro de su grupo asignado que esté disponible. Si la duda continúa, se convoca al facilitador. Si no lo resuelven, el facilitador pedirá a todos que se detengan para cooperar como equipo en la resolución del conflicto que ha despertado la duda. Así debatirán todos los integrantes de la mesa buscando la solución. Si aún así no pueden resolverlo, anotarán la duda que será abordada asincrónicamente por el soporte técnico o de forma sincrónica en la siguiente clase por parte del profesor.

El trabajo comienza junto al docente, frente a la duda:

### COOPERACIÓN



Todos los días, finalizada la jornada, los estudiantes reconocerán a uno de los integrantes del grupo con quienes compartió ese día. El criterio para ese reconocimiento es la cooperación.

Cada grupo tendrá un facilitador que será elegido a partir de los reconocimientos y generando un sistema de rotación donde cualquiera pasar por ese rol. El facilitador no es una figura estática, sino que cumple un rol dinámico y versátil. El facilitador es un estudiante que moviliza el alcance de los objetivos comunes del equipo poniendo en juego la cooperación. Es aquel que comparte con la mesa su potencial en favor del resto del equipo, y que por lo tanto promueve la cooperación.

## Información de la materia

- Modalidad 100% a distancia.
- Cantidad de semanas totales: 9
- Cantidad de encuentros sincrónicos semanales: 3
- Clases virtuales en nuestro campus Playground: 18
- Cantidad de clases en vivo: 18

## Requisitos y correlatividades

¿Qué materias tiene que tener aprobada el alumno previamente?

- Introducción a la Informática (Bimestre 1, Año 1)
- Programación Imperativa (Bimestre 1, Año 1)

¿Qué materias cursa después?

- Proyecto Integrador (Bimestre 4, Año 1)
- Testing II (Bimestre 2, Año 2)

## Modalidad de trabajo

Nuestra propuesta educativa está diseñada especialmente para esta modalidad 100% a distancia, mediante un aprendizaje activo y colaborativo siguiendo nuestro pilar de "aprender haciendo".

Los entornos de aprendizaje son tanto sincrónicos como asincrónicos, con un enfoque que vincula teoría y práctica, por lo que ambas están presentes en todo momento.

Contamos con un campus virtual propio en el cual vamos a encontrar actividades, videos, presentaciones y recursos interactivos con instancias de trabajo individual y en equipo para profundizar en cada uno de los conceptos.

Además, realizaremos encuentros online y en vivo con el grupo de estudiantes y docentes, a los que podremos sumarnos desde donde estemos a través de una plataforma de videoconferencias con nuestra cámara y micrófono para generar una experiencia cercana.

## Metodología de evaluación

La evaluación formativa es un proceso continuo que genera información sobre la formación de nuestros estudiantes y de nosotros como educadores.

A su vez, se genera conocimiento de carácter retroalimentador, es decir, tiene una función de conocimiento ya que nos permite conocer acerca de los procesos de enseñanza y aprendizaje. También tiene una función de mejora continua porque nos permite saber en qué parte del proceso nos encontramos, validar si continuamos por el camino planificado o necesitamos tomar nuevas decisiones para cumplir los objetivos propuestos.

Por último, la evaluación desempeña un papel importante en términos de promover el desarrollo de competencias muy valiosas.

Nuestro objetivo es correrlos de la evaluación tradicional, donde muchas veces resulta un momento difícil, aburrido y tenso. Para ello, vamos a utilizar la gamificación, la cual es una técnica donde se aplican elementos de juego para que el contenido sea más atractivo, los

participantes se sientan motivados e inmersos en el proceso, utilicen los contenidos de aprendizaje como retos que realmente quieren superar y aprendan del error.

**A su vez, para registrar dicha formación, se utiliza un conjunto de instrumentos, para los cuales es fundamental utilizar la mayor variedad posible y técnicas de análisis.**

## Criterios de aprobación

- Realizar las actividades de Playground (80% de completitud)
- Asistencia a los encuentros sincrónicos (90% de asistencia)\*
- Obtener un puntaje de 7 o más en la evaluación final.
- Obtener un puntaje de 7 o más en la nota final de la materia.

## Contenidos

### **Módulo 1: Fundamentos de testing y Gestión de Defectos.**

Introducción al proceso de prueba tradicional. Comenzando con la historia de la prueba, lo que un tester debe tener en mente en sus actividades diarias, el proceso formal de pruebas y terminando con el conocimiento de los tipos y niveles de prueba para saber qué pruebas realizar de acuerdo al contexto del producto a probar. Uno de los objetivos principales de todo tester es encontrar defectos. También en este módulo se aprenderá qué es un defecto, cómo reportar y gestionar el mismo.

#### **Clase 1: Primeros pasos**

- Un poco de historia
- Los 7 principios del testing
- Aspecto psicológico del testing
- Los 3 pilares del desarrollo de software

#### **Clase 2: Gestión de defectos**

- ¿Qué es un defecto?
- Errores, defectos y fallas
- Ciclo de vida de un defecto



- Gestión de defectos
- "Defectos naturales"

## Clase 3: Fundamentos de Testing y Gestión de Defectos - Revisión

- Cierre de semana (Clase 1 y 2)

## Módulo 2: Diseño e implementación de la prueba

Durante el diseño de la prueba, las condiciones de prueba se transforman en casos de prueba. En este módulo se presentan las distintas técnicas de prueba para generar un caso de prueba eficiente.

Durante la etapa de implementación de la prueba debemos ser capaces de contestar la siguiente pregunta, ¿está todo preparado para realizar la prueba? Luego de realizar las configuraciones necesarias se procede a la ejecución de las pruebas. En este módulo conoceremos cómo organizar, controlar y reportar la ejecución de las pruebas.

## Clase 4: Diseño de la prueba

- Casos de prueba
- Testing negativo vs testing positivo
- "Happy Path"
- Casos de Uso

## Clase 5: Niveles y Tipos de Prueba

- Modelos de Desarrollo de Software
- Niveles de Prueba
  - Prueba de Componente
  - Prueba de Integración
  - Prueba de Sistema
  - Prueba de Aceptación
- Tipos de Prueba
  - Prueba Funcional
  - Prueba No Funcional



## Clase 6: Diseño de la Prueba y Niveles y Tipos de Prueba - Revisión

- Cierre de semana (Clase 4 y 5)

## Clase 7: Técnicas de pruebas

- Caja negra vs Caja blanca
- Análisis de valores límites
- Partición de equivalencia
- Prueba de tabla de decisión
- Prueba de transición de estados
- Pruebas basadas en la experiencia

## Clase 8: Implementación y ejecución de la prueba

- Ejecución de casos de prueba
- Creación de suites
  - Smoke suites
  - Regression suites

## Clase 9: Pruebas - Revisión

- Cierre de semana (Clase 7 y 8)

## Módulo 3: Análisis y Planificación de la prueba

Durante el análisis de la prueba, se analiza la base de prueba para identificar "qué probar". Estas actividades de análisis de prueba no sólo verifican si los requisitos son consistentes, están debidamente expresados y son completos, sino que también validan si los requisitos satisfacen adecuadamente las necesidades de los clientes, usuarios y otros implicados.

En este módulo aprenderemos cómo estar involucrados en las pruebas desde etapas tempranas y cómo los distintos puntos de vista en los procesos de revisión ayudan a mejorar la calidad del producto.

La planificación de la prueba implica actividades que definen los objetivos de la prueba y el enfoque para cumplir con los objetivos de la prueba. En este módulo aprenderemos cómo se



organizan las pruebas a través de distintos entornos de trabajo hasta llegar al entregable final.

## **Clase 10: Pruebas estáticas y dinámicas**

- Pruebas estáticas
- Pruebas estáticas vs Pruebas dinámicas
- Proceso de revisión
- Requerimientos

## **Módulo 4: Planificación de la prueba**

### **Clase 11: Organización de la prueba**

- Pruebas en entornos de desarrollo

### **Clase 12: Evaluación I**

## **Módulo 5: Unit Testing**

En los últimos tiempos, la metodología test driven development se ha ido imponiendo como una forma de trabajo debido a que las pruebas son requeridas en etapas tempranas del proyecto. En este módulo iniciaremos en el camino de este proceso de desarrollo que consiste en codificar pruebas, desarrollar y refactorizar de forma continua el código construido.

### **Clase 13: Introducción al Debugging**

- Debugging

### **Clase 14: Introducción a la prueba de componente**

- Introducción a unit testing - JavaScript y Jest



## Clase 15: Cierre de la semana

- Cierre de semana (Clase 13 y 14)

## Clase 16: Prueba de componente

- Prueba y cobertura de sentencia
- Prueba y cobertura de decisión
- Herramientas de coberturas de prueba
- Test driven development

## Clase 17: Test Unitario

- Implementación de test unitarios en JavaScript

## Clase 18: Cierre de Semana

- Cierre de semana (Clase 16 y 17)

## Módulo 6: BE testing

Introducir al alumno al concepto de API —siglas de “application programming interface”—. Cómo funcionan las mismas y cuáles son las mejores herramientas y maneras de comprobar su funcionamiento.

## Clase 19: Introducción - API testing

- Back end testing
- APIs
- Protocolo HTTP
- Postman I

## Clase 20: API Testing

- Postman II
- GET y POST
- Tests (js)
- Colecciones en postman/variables de entorno



**Certified Tech Developer**

The Ultimate Degree

**DigitalHouse** >  
Coding School

## **Clase 21: Cierre de la semana**

- Cierre de semana (Clase 20 y 21)

## **Módulo 7: Introducción a Automation**

Introducir al alumno dentro del mundo de automatización de prueba, haciendo énfasis en mostrar cómo mejora la calidad de software y acorta los tiempos de prueba.

## **Clase 22: Fundamentos de automatización de la prueba**

- Introducción a automation
- Patrones de diseño
- Page Object Model
- Introducción a Selenium

## **Clase 23: Automatización de la prueba**

- Criterios para la automatización y selección de de herramientas
- Selenium WebDriver
- Creando nuestro primer proyecto
- Nuestro primer test

## **Clase 24: EVALUACIÓN II**

## **Clase 25: Práctica Integradora I**

## **Clase 26: Práctica Integradora II**

## **Clase 27: Cierre de Semana - RETRO**