



# Certified Tech Developer

The Ultimate Degree

---

## Infraestructura II

---

### Fundamentación

El mundo de la tecnología ha cambiado significativamente de la mano de la introducción de nuevas herramientas y metodologías, que son consecuencia de la necesidad de responder de manera eficaz y eficiente a las necesidades del mercado.

Dichos cambios no son exclusivos del mundo del desarrollo de software. Para que la entrega de valor y las respuestas a las necesidades del mercado se den de manera ágil y predecible, estas nuevas tendencias han alcanzado al mundo de la infraestructura. De este modo, los analistas de infraestructura pueden acompañar a los desarrolladores y trabajar ya no en silos o aislados, sino en conjunto con un objetivo común.

Infraestructura II tiene por objetivo integrar los conceptos abordados en Infraestructura I, mediante la incorporación de tres nuevas verticales: infraestructura como código, build y release pipelines y monitoreo —que si bien son disciplinas en sí mismas, servirán para integrar y orquestar los conocimientos adquiridos durante Infraestructura I—.

De este modo, te brindamos una visión integral de los procesos y las prácticas de infraestructura moderna para permitirte desempeñar potencialmente el rol de Analista de Infraestructura Jr.

## Objetivos de aprendizaje

- Entender la cultura DevOps, su origen y cómo beneficia a las organizaciones.
- Conocer la importancia de la disciplina de infraestructura como código y las herramientas que pueden utilizarse para tal fin.
- Comprender la importancia de los pipelines en el ecosistema DevOps y cómo las tecnologías o prácticas adquiridas durante Infraestructura I (scripting, configuration management y Docker) pueden utilizarse como amalgama para las actividades ejecutadas en el contexto de un Pipeline.
- Definir la necesidad de monitoreo y entender por qué miramos a los sistemas de manera integral. Reconocer la diferencia entre el monitoreo de infraestructura y el monitoreo de aplicaciones y cómo monitorear el stack de manera integral.

## 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 el trabajo en equipo, la creatividad, la responsabilidad, el compromiso, la comunicación efectiva y la 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, etcétera. 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 ellas 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, lograrás 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. Todas estas herramientas te ayudarán 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 proponemos 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.

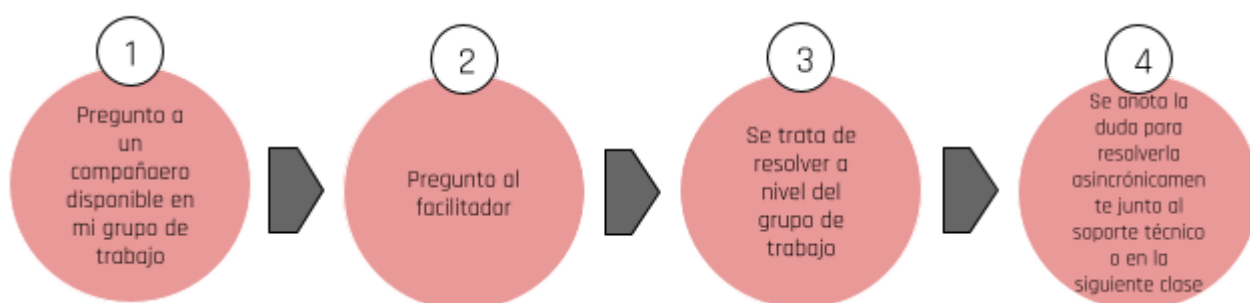
La explicación recíproca como eje del trabajo cotidiano no solo facilita el aprendizaje de los compañeros, sino que sobre todo potencia la consolidación de conocimientos por parte de quien explica. Se promueve la responsabilidad, la autonomía, la proactividad, todo en el marco de la cooperación. Esto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

El trabajo cooperativo permite entablar relaciones responsables y duraderas, aumenta la motivación y compromiso y promueve un buen desarrollo cognitivo y social. La cooperación surge frente a la duda. En caso de tener una pregunta, le consultarás a algún miembro de su grupo asignado que esté disponible. Si la duda continúa, se convocará 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 aun 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 puede 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 quien 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: 27
- Cantidad de clases en vivo: 27

## Requisitos y correlatividades

Para cursar Infraestructura II es necesario haber aprobado Introducción a la Informática e Infraestructura I. Asimismo, para cursar Infraestructura III, tiene que estar aprobada Infraestructura II.

## 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 corrernos de la evaluación tradicional, donde muchas veces resulta un momento difícil, aburrido y tenso. Para ello, vamos a utilizar la gamificación, una técnica donde se aplican elementos de juego para que el contenido sea más atractivo y te sientas más motivado e inmerso en el proceso, utilices los contenidos de aprendizaje como retos que realmente quieras superar y aprendas del error.

**A su vez, para registrar dicha formación, utilizamos 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: Inmersión

Nos acercaremos de manera introductoria al mundo de la infraestructura moderna y expondremos las bases y pilares sobre los que se asienta. Presentaremos la cultura DevOps, entendida como práctica y como cultura. Además, analizaremos el contexto y las problemáticas que dan lugar a su surgimiento.

### **Clase 1: Bienvenidos a Infraestructura II**

- Definir y explicar el objetivo de la materia
- Enumerar y recorrer los temas vistos en Introducción a la Informática e Infraestructura I.
- Ubicar en tiempo y espacio los temas vistos en un mapa del mundo de la infraestructura moderna.
- Presentar los contenidos de Infraestructura II y explicar cómo aumentan los contenidos ya adquiridos.



## Clase 2: ¿Qué es DevOps?

- Explicar qué es DevOps.
- Qué problemas resuelve.
- Qué herramientas hacen a la disciplina o práctica.
- Qué perfiles y personas actúan en el ecosistema DevOps.
- Qué se espera de una persona que se desempeña en este ecosistema.

## Clase 3: Cierre de la semana

## Módulo 2: Infraestructura como código

Estudiaremos el concepto de la infraestructura como código y el ecosistema de herramientas que la componen. A lo largo de las clases pondremos el foco en algunas de ellas: CloudFormation, Ansible y Terraform.

## Clase 4: Infraestructura como código: La disciplina

- Explicar el concepto de infraestructura como código.
- ¿Qué existía antes de infraestructura como código?
- Dos paradigmas para la infraestructura como código: código imperativo vs. código declarativo.
- Ecosistema de herramientas IaC: diferencias entre tecnologías y cuáles debemos conocer.
- Principio de idempotencia.

## Clase 5: Infraestructura como código en AWS: CloudFormation

- ¿De qué se trata AWS CloudFormation?
- ¿Cómo usar esta herramienta?
- ¿Dónde la usamos?
- ¿Quién la usa?
- ¡Hora de practicar!

## **Clase 6: Cierre de la semana**

## **Clase 7: Infraestructura como código: Ansible**

- ¿De qué se trata Ansible?
- ¿Cómo usar esta herramienta?
- ¿Dónde la usamos?
- ¿Quién la usa?
- ¡Hora de practicar!

## **Clase 8: Infraestructura como código: Terraform**

- ¿De qué se trata Terraform?
- ¿Cómo usar esta herramienta?
- ¿Dónde la usamos?
- ¿Quién la usa?
- ¡Hora de practicar!

## **Clase 9: Cierre de la semana**

## **Clase 10: Infraestructura como código: Terraform (continuación)**

- Detalles de TF (DSL de Terraform).
- Escribimos un template más complejo.
- Que es un provisioner.
- Que es un módulo.

## **Clase 11: Examen parcial**

## **Clase 12: Cierre de la semana**

## **Módulo 3: Pipelines**

Presentaremos los pipelines, abordando qué son, para qué sirven y el rol que juegan en la infraestructura moderna. Además de introducirnos al mundo de los pipelines,



desagregaremos los procesos de entrega continua y realizaremos ejercitaciones.

## **Clase 13: Introducción al mundo de los pipelines**

- ¿Qué son y para qué sirven?
- ¿Qué rol juegan en el mundo de la infraestructura moderna?
- ¿Qué tecnologías existen?
- Presentación de Jenkins (y diferentes formas de ejecutarlo).
- Scripting como amalgama.

## **Clase 14: Pipelines: build y continuous integration**

- ¿Qué es el proceso de build y qué incluimos en él?
  - Maven: Compilación y gestión de configuraciones.
  - Testing: Unit testing.
  - Docker: Construcción de imágenes.
- ¿Qué es continuous integration y por qué es importante conocer el concepto?
  - Triggers.
- ¿Qué es un artefacto?
  - ¿Dónde almacenamos el producto del proceso de Build?
  - Principio de inmutabilidad.

## **Clase 15: Cierre de la semana**

## **Clase 16: Pipelines: release y continuous delivery**

- ¿Qué es el proceso de release y que incluimos en él?
- ¿Qué es continuous delivery y continuous deployment?
- ¿Cómo se ve un proceso de release?
  - Stages, ¿qué son y para qué sirven?
  - Gateways, ¿qué son y para qué sirven?
- Inmutabilidad en acción.

## Clase 17: Pipelines: End-to-End

- Una mirada integral de los pipelines.
- Combinando el deployment de infraestructura con el deployment de una aplicación.

## Clase 18: Cierre de la semana

### Módulo 4: Monitoreo

Estudiaremos sobre monitoreo y la diferencia entre monitorear infraestructura y monitorear aplicaciones. Se presentan herramientas para realizar el monitoreo, la importancia de qué buscar al momento de monitorear y la importancia de las métricas.

## Clase 19: Monitoreo: Introducción

- ¿Qué significa monitorear?
- ¿Qué diferencias hay entre monitorear infraestructura y monitorear aplicaciones?
- ¿Qué buscamos cuando monitoreamos?
- La importancia de las métricas

## Clase 20: Monitoreo: Monitoreando infraestructura

- ¿Por qué es importante monitorear infraestructura?
- ¿Qué miramos cuando monitoreamos infraestructura?
- ¿Qué herramientas o plataformas utilizamos para monitorear Infraestructura?
  - Nagios

## Clase 21: Cierre de la semana

## Clase 22: Monitoreo: Monitoreando aplicaciones

- ¿Por qué es importante monitorear aplicaciones?
- ¿Qué miramos cuando monitoreamos aplicaciones?
- ¿Cuál es la diferencia de monitorear aplicaciones e infraestructura?



- ¿Qué herramientas o plataformas utilizamos para monitorear aplicaciones?
  - NewRelic
  - Prometheus
- La importancia de mirar el todo y no solo las partes: Observabilidad.
  - ¿Cómo reaccionamos a aquello que observamos?

## **Clase 23: Cierre de la semana**

## **Clase 24: Examen final**

### **Módulo 5: Cierre de la materia**

Con el objetivo de consolidar los conocimientos aprendidos durante la materia, en este último espacio ejercitaremos una visión holística del ecosistema de la infraestructura moderna.

## **Clase 25: Armemos un pipeline**

Realizaremos una actividad que integre todo lo aprendido durante Infraestructura I e Infraestructura II, mediante el armado de un pipeline que incluya la mayoría de los conceptos y tecnologías abordados durante ambas materias.

## **Clase 26: Otros conceptos de infraestructura moderna**

Retomaremos los conceptos desarrollados en la materia y se suman otros para aportar en la construcción del mundo de la infraestructura moderna. El objetivo principal de esta clase es dar a conocer tecnologías, prácticas y terminologías que no se abordaron en el programa de Infraestructura I o Infraestructura II, con el objetivo de dar contexto al alumno de cara a una posible entrevista de trabajo.

## **Clase 27: Cierre de la semana**