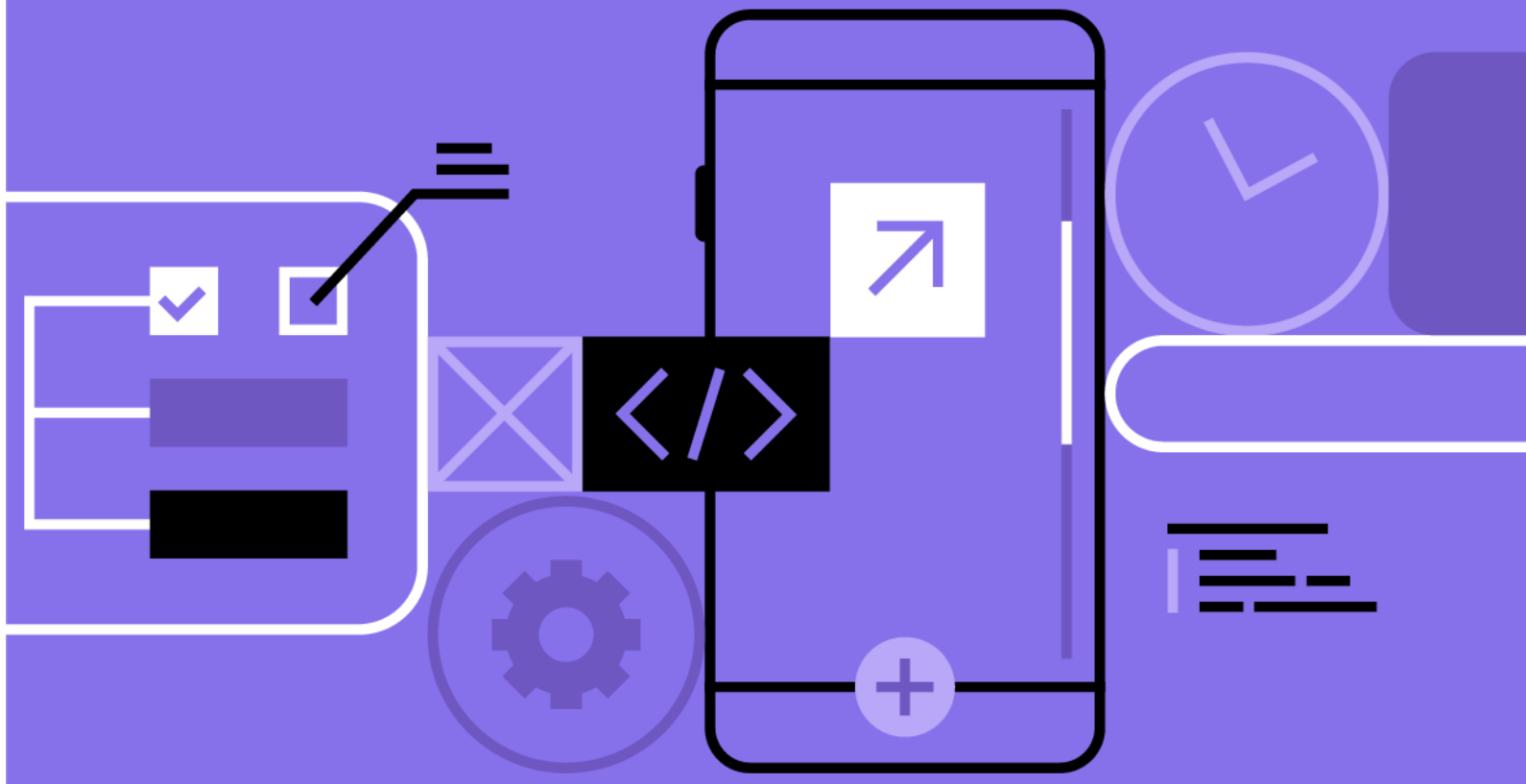
DIGITALHOUSE

Certified Tech Developer

The Ultimate Degree



Especialización en Back End III

Especialización en Back End III

Fundamentación

Una de las características que podemos encontrar en una arquitectura de microservicios es la diversidad en los lenguajes de programación utilizados en la construcción de cada uno de ellos.

Es por ello que en esta materia estudiaremos **Go**, uno de los lenguajes más modernos, creado recientemente en el 2009 por Rob Pike, Ken Thompson y Robert Griesemer. Sus características lo convierten en un lenguaje cada vez más utilizado en la construcción de componentes de back end, como microservicios y procesos que involucren gran escala de datos —por ejemplo, el big data—. En gran parte, gracias a sus cualidades, como el manejo de concurrencia nativa, gran cantidad de librerías nativas para el desarrollo de web APIs, encriptado y manejo de big data.

Objetivos de aprendizaje

- Adquirir los conocimientos teóricos y prácticos de la sintaxis del lenguaje Go.
- Diseñar e implementar una API en el lenguaje Go.
- Integrar un microservicio desarrollado en Go en una solución basada en Spring Cloud.

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, 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, entre otros recursos. De esta forma, cuando llegues al encuentro en vivo, estarás en condiciones de abordar el tema y aprovechar esa instancia al máximo.

Empleamos 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. En otras palabras, concebimos la clase como un espacio de trabajo.

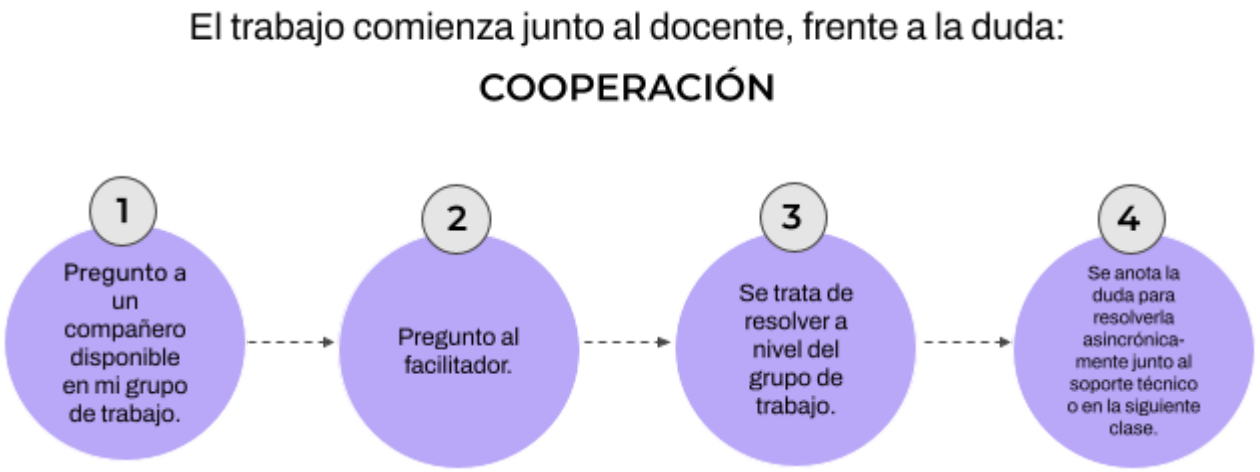
Una de las cuestiones centrales de nuestra metodología de enseñanza es el aprendizaje en la práctica. Por ese motivo, a lo largo de la cursada estarán muy presentes las ejercitaciones, es decir, la práctica de actividades de diversos tipos y niveles de complejidad que te permitirán afianzar el aprendizaje y comprobar que lo hayas asimilado correctamente. De esta forma, se logra la incorporación de los contenidos de una forma más significativa y profunda, la asimilación de los conocimientos se vuelve 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 ayuda a ejercitar múltiples 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 entre 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. En caso de que no lo resuelvan, el facilitador pedirá a todos que se detengan para cooperar como equipo en la resolución del conflicto que ha despertado la pregunta. 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.



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

Cada grupo tendrá un facilitador que será elegido a partir de los reconocimientos y desarrollarán un sistema de rotación donde cualquiera pueda pasar por dicho rol. El facilitador no es una figura estática, sino que cumple un rol dinámico y versátil: se trata de 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 clases virtuales en Playground: 27.
- Cantidad de clases en vivo totales: 27.

Requisitos y correlatividades

Para cursar esta materia de la Especialización en Back End es requisito haber aprobado la segunda materia de la especialización. A su vez, la aprobación de esta es requisito para cursar el Proyecto Integrador 2.

Modalidad de trabajo

Nuestra propuesta educativa está diseñada especialmente para la modalidad 100% a distancia, mediante un aprendizaje activo y colaborativo bajo nuestro lema “aprender haciendo”. Es por esto que 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. Esto 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 diferenciarnos de la evaluación tradicional, que 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 del lenguaje Go

Clase 1: Introducción a la sintaxis Go

- Organización de un proyecto Go
    - Modules
    - Package

- Variables
    - Tipos de datos
    - Declaración en bloque
    - Constantes
    - Conversión de tipo de datos
- Operadores
- Condicionales y bucles
- Estructura de datos
    - Arrays
    - Slices
    - Maps

Clase 2: Funciones

- Definiendo una función
- Ellipsis
- Múltiples retornos
- Retorno de funciones

Clase 3: Integración

Clase 4: Estructuras

- Definición e instanciación
- Etiquetas
- Estructuras y JSON
- Métodos

Clase 5: Composición, punteros e interfaces

- Punteros
- Composición (embedding structs)
- Interfaces, interfaz vacía, typecasting, type assertion

Clase 6: Integración

Clase 7: Manejo de errores e interrupciones

- ¿Qué es un error en Go?
- Diferencias con otros lenguajes como Java y JavaScript
- Errores personalizados
- Package errors
    - As()
    - Is()
    - Unwrap()
- Funciones multiretorno y errores
- Panic
- Creando nuestros panic

- Estructura de un panic
- Casos de panic
    - Index out of bounds
    - Receptores nulos
- Defer y recover

Clase 8: Operaciones de I/O y archivos

- Paquete fmt
- Paquete os
- Paquete io

Clase 9: Integración

Clase 10: Concurrencia y paralelismo

- Goroutines
- Canales

Clase 11: Taller de código - Inicio desafío I

- Consigna del desafío

Clase 12: Taller de código - Entrega desafío I

Módulo 2: Go Web

Clase 13: APIs I

- Arquitectura web
- Package JSON
    - Marshal

    - Unmarshal
- Package NET/HTTP

Clase 14: APIs II

- Método GET
- Web Context
- Muxing Router

Clase 15: Integración

Clase 16: APIs III

- Método POST
- ShouldBind vs. Bind
- Headers
- Enviar, recibir y validar un token
- Controlador, servicio, repositorio
- Estructura de proyecto en Go
- Método PUT
- Método PATCH
- Método DELETE

Clase 17: Configuración de entorno y persistencia en archivos

- Introducción
- Instalación y uso de gotenv
- Token en variable de entorno

Clase 18: Integración

Clase 19: Middleware, autenticación y documentación

- Middleware
- Autenticación
- Swagger con Swaggo
- Anotaciones para documentar
- Generar documentación y visualizarla

Clase 20: Capa de acceso a datos

- Package Database/SQL
- Instalación de driver MySQL
- Implementación
- Capa repository
- Implementación
    - Store
    - GetOne
    - Update
    - GetAll
    - Delete
    - GetFullData

Clase 21: Integración

Clase 22: Taller de coding I - Desafío II

- Desarrollar la capa de Entidades
- Desarrollar capa Repository

Clase 23: Taller de coding II - Desafío II

- Desarrollar la capa de Service
- Desarrollar capa de Controller

Clase 24: Taller de coding III - Entrega desafío II

- Ajustes y entrega del desafío

Clase 25: Go en una arquitectura Spring Cloud I

- Arquitectura a implementar
- Paso 1: Implementar microservicio en Go
- Paso 2: Registrar microservicio Go en Eureka

Clase 26: Go en una arquitectura Spring Cloud II

- Paso 3: Construir docker para el microservicio en Go
- Paso 4: Incorporar el microservicio Go dentro de la arquitectura Spring Cloud
- Paso 5: Invocar desde un microservicio Java al microservicio Go

Clase 27: Cierre