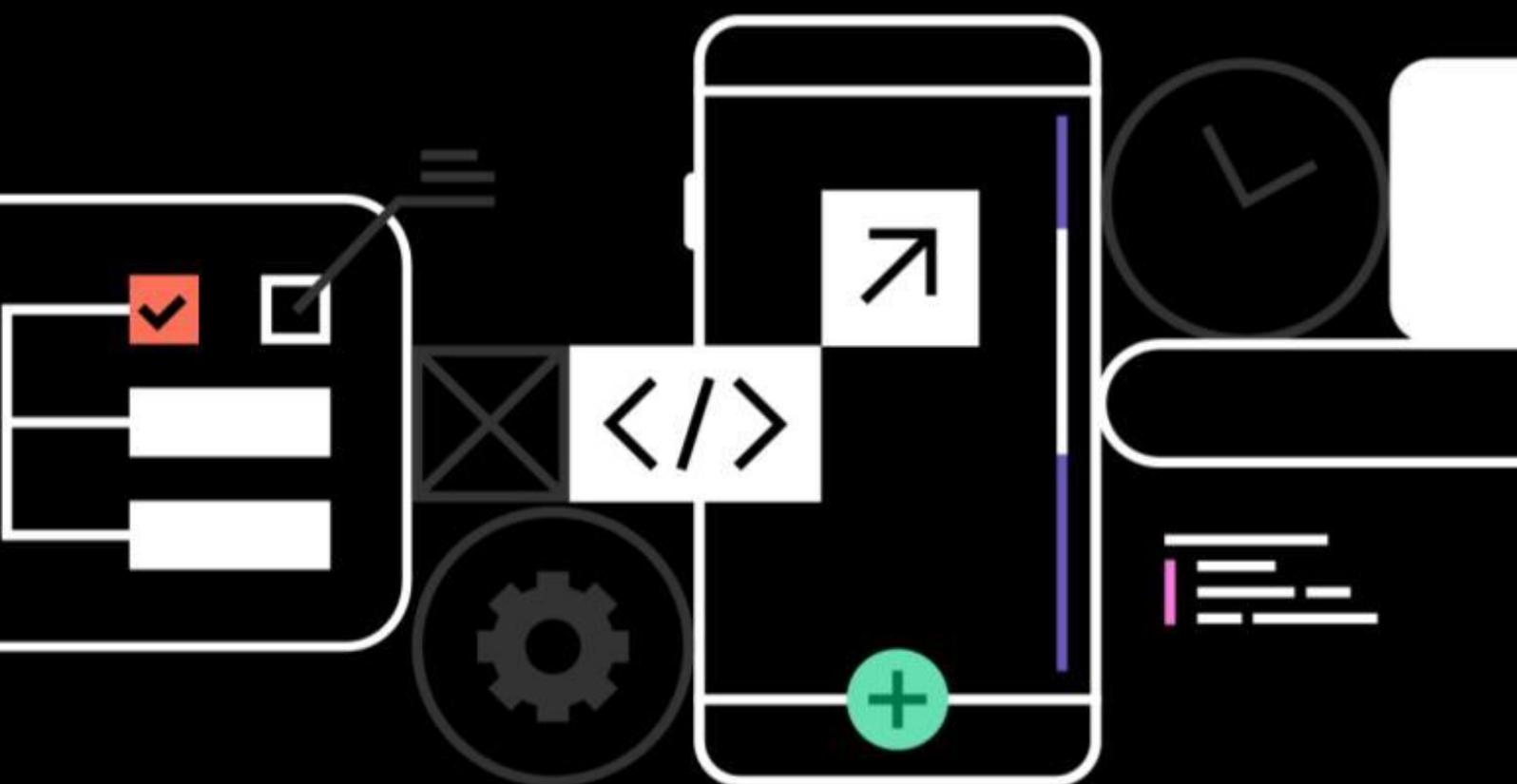


DIGITALHOUSE

Fundamentos de GO

 Programa



Fundamentos de GO

Objetivos de aprendizaje

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

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 este curso 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.

Metodología de enseñanza-aprendizaje

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

Nuestra propuesta incluye contenidos en formato audiovisual, contenidos descargables, desafíos prácticos y cuestionarios que van a desafiar tus nuevos conocimientos adquiridos en nuestro campus virtual.

Empleamos actividades y estrategias basadas en los métodos 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 desarrollador.

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 colaborativo es uno de los elementos centrales de nuestra metodología, por eso te convocaremos a participar de nuestra comunidad de aprendizaje junto con otros estudiantes y docentes 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 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 la comunidad que esté disponible y resolverán el desafío en conjunto con los docentes y otras personas participantes de la misma comunidad.

Requisitos

Para cursar este curso es requisito contar con los siguientes conocimientos:

- Desarrollo de aplicaciones con Microservicios.
- Visual Studio Code.
- GitLab / GitHub o similares

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 están creados 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 vas a encontrar actividades, videos, presentaciones y recursos interactivos con instancias de trabajo individual e invitaciones a la comunidad para profundizar en cada uno de los conceptos.

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 (100% de completitud).
- Aprobación de los checkpoints de conocimiento de cada módulo de aprendizaje.
- Aprobación del cuestionario final del curso.

Contenidos

Módulo 1 -Bienvenida

Te damos la bienvenida a nuestro curso y haremos un test de autoevaluación de conocimientos.

Clase 1: Bienvenida

- Programa del curso
- Presentación del curso
- Cuestionario de autoevaluación

Módulo 2: Fundamentos del lenguaje GO

Clase 2: Introducción a la sintaxis de GO

- Instalación de GO
- Introducción
- GO Path (estructura de carpetas)
- Modules & Packages
- Variables en GO
- Variables en GO - Aplicación
- Constants
- Bloques
- Variable Size
- Unit Variable Size
- Float Variable Size
- String Variable Size
- Convert to String

- Convert to Number
- Operators
- Operators | Aplicación
- Condicional If
- Condicional Switch
- Bucle for
- Arrays
- Arrays | Aplicación
- Slices
- Slices | Aplicación
- Maps
- Maps | Aplicación

Clase 3: Funciones

- Functions
- Functions | Aplicación

Clase 4: Estructuras

- Structs
- Tags y JSON
- Convenciones para nombrar elementos
- Métodos
- Structs - Ejemplo
- Format

Clase 5: Checkpoint de conocimientos

- Checkpoint de conocimientos

Clase 6: Composición, punteros e interfaces

- Pointers
- Pointers | Aplicación
- Interfaces
- Interfaces | Aplicación

Clase 7: Errores e interrupciones

- Errors
- Errors | Demo
- Methods (As, Is, Unwrap)
- Panic, Defer y Recover

Clase 8: Packages

- Package internos
- Package time
- Package OS
- Package log
- Package externos
- Package UUID
- Package custom

Clase 9: Concurrencia y paralelismo

- Goroutines & Channels
- Goroutines
- Channels

Clase 10: Generics

- Generics

Clase 11: Checkpoint de conocimientos

- Checkpoint de conocimientos

Módulo 3: GO Web

Clase 12: APIs I

- Introducción
- Arquitectura web
- Principios de REST
- GO Web Basic

- Context

Clase 13: APIs II

- Arquitectura por capas
- Estructura del proyecto
- Estructura del proyecto | Aplicación
- Capa Repositorio
- Capa Servicio
- Capa Controller

Clase 14: APIs III

- Bootstrap Package
- Transport Package
- Handler
- Dynamic params
- Método Get
- Método Patch
- Errors
- Responses

Clase 15: Bases de datos, environments & autorización

- Database
- Repositorio: GetAll & Create
- Repositorio: Get & Update
- Arreglos en el package Response
- Environment
- Environment Token
- Gin Gonic
- Gin Gonic | Aplicación

Clase 16: Checkpoint de conocimientos

- Checkpoint de conocimientos

Módulo 4: Cierre del curso

Clase 17: Cierre del curso

- Cuestionario final
- Cierre del curso