



UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA

Campus de la UNA
SAN LORENZO-PARAGUAY

RESOLUCIÓN N° 0637/2026

POR LA CUAL SE AUTORIZA LA APERTURA DE LA SEGUNDA CONVOCATORIA 2026, DEL CURSO DE INTEGRACIÓN BACK-END (BASADO EN NODE.JS).

13 de mayo de 2026

VISTO Y CONSIDERANDO: El Memorando DGCITIC/061/2025, del Director, Lic. Juan Fernando Duré, de la Dirección de Gestión del Centro de Innovación en TIC de la FP-UNA, en el cual solicita la apertura de la segunda convocatoria del Curso de Integración Back-End (basado en Node.js), nivel intermedio, presentado por la Prof. Lic. Leticia Carolina Mendieta Guillen.

Que la propuesta está diseñada en la Integración Back-End en Desarrollo Web, en la que ofrece una exploración en profundidad de los aspectos del lado del servidor del desarrollo web esenciales para crear sitios web y aplicaciones dinámicas y basadas en datos. A través de un plan de estudios integral, los estudiantes profundizarán en la programación del lado del servidor, gestión de bases de datos, desarrollo de API, y la integración, así como las estrategias de implementación, asegurando que emergen con una comprensión holística de las tecnologías de Back-end y sus aplicaciones.

Que el curso está estructurado en base a 40 horas (8 semanas de duración) a ser desarrolladas en la modalidad virtual.

Periodo estimado de Inscripción: 11/05/2026 al 03/07/2026

La fecha de **inicio**: 22/06/2026, la fecha de **finalización**: 15/08/2026.

Se estima dar apertura con una convocatoria de 5 (cinco) matriculados como mínimo y 50 matriculados como máximo

La Ley N° 4995/2013 de Educación Superior.
El Estatuto de la Universidad Nacional de Asunción.

POR TANTO: en uso de sus facultades y atribuciones legales,

LA DECANA DE LA FACULTAD POLITÉCNICA
RESUELVE:

Art. 1° Autorizar la apertura de la Segunda Convocatoria 2026, del Curso de Integración Back-End (basado en Node.js), ofrecido por la FP-UNA.

Art. 2° Comunicar, copiar y archivar.

Prof. Abg. Joel Arsenio Benítez Santacruz
Secretario de la Facultad

Prof. Ing. Silvia Teresa Leiva León, MSc.
Decana





Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA

ANEXO RESOLUCIÓN N° 0637/2026

Pág. 1/10

Universidad Nacional de Asunción

Facultad Politécnica

Centro de Innovación en TIC



Título: *Integración Back-End (basado en Node.js)*

Modalidad: *Virtual*

Docente

Prof. Lic. Leticia Carolina Mendieta Guillén

Sede Central, San Lorenzo

Segunda Convocatoria - Mayo, 2026





Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA

ANEXO RESOLUCIÓN N° 0637/2026

Pág. 2/10

Inteligencia Artificial

Información básica

1	Título	<i>Integración Back-End (basado en Node.js)</i>
2	Código	-
3	Año propuesto	2026
4	Semestre propuesto	<i>Primero</i>
5	Departamento	<i>Centro de Innovación TIC (FP-UNA)</i>
6	Año Objetivo/Carrera	<i>Programa de Desarrollo de Software</i>
7	Horas de créditos	<i>Crédito total: -</i>
		<i>Créditos de clase: -</i>
		<i>Créditos experimentales/prácticos: -</i>
		<i>Créditos de diseño: 0</i>
	<i>Otros: 0</i>	
8	Formato de clase (tipo)	<i>Clases interactivas, sesiones prácticas.</i>

[Horario y Lugar]

1	Días	<i>Lunes y martes</i>
2	Horario	<i>Lunes 19:00 a 22:00 h dedicación clases síncronas Martes 19:00 a 21:00 h dedicación clases asíncronas</i>
3	Ubicación	<i>Online</i>

[Información del instructor/a]

1	Nombre	<i>Lic. Leticia Carolina Mendieta Guillén.</i>
2	Oficina (si aplica)	-
3	Contacto (correo)	<i>letiziag294@fpuna.edu.py</i>
4	Contacto (teléfono)	<i>(0972) 224-028</i>

* Si el curso se imparte en equipo con asistente(s), añada más cuadros para describirlos a todos.





Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN FACULTAD POLITÉCNICA

ANEXO RESOLUCIÓN N° 0637/2026

Pág. 3/10

Prerrequisitos

- **Conocimientos básicos de programación:** Familiaridad con la lógica básica de programación, como estructuras de control (if, loops) y conceptos como variables y funciones. Idealmente, con experiencia previa en JavaScript.
- **Conocimientos de JavaScript:** Los estudiantes deben tener una comprensión sólida de JavaScript, ya que Node.js es un entorno basado en este lenguaje.
- Deben estar cómodos trabajando con funciones, objetos, arrays y promesas.
- **Uso de terminal o línea de comandos:** Familiaridad con el uso de la terminal para ejecutar comandos en sistemas operativos como Windows, macOS o Linux. Esto incluye comandos básicos para navegar en directorios, instalar paquetes con npm, y ejecutar scripts de Node.js.
- **Conocimientos básicos de HTTP y APIs:** Los estudiantes deben tener nociones sobre solicitudes web y respuestas y de cómo funcionan las peticiones HTTP (GET, POST, PUT, DELETE) y cómo interactúan con APIs.
- **Familiaridad con bases de datos:** Los estudiantes deben tener nociones básicas sobre cómo funcionan las bases de datos relacionales y no relacionales (como PostgreSQL o MongoDB).
- **Uso de editores de código:** Es esencial que los estudiantes se sientan cómodos utilizando editores de texto o IDEs como **Visual Studio Code**.

Descripción del curso

Este curso avanzado forma parte del Programa de Desarrollo de Software y está diseñado para desarrollar competencias sólidas en la creación, integración y despliegue de soluciones de back-end utilizando Node.js. A lo largo de 40 horas intensivas, los participantes aprenderán a crear y conectar sistemas back-end modernos usando Node.js, con un enfoque práctico, actual y ético.

Características clave del curso:

- Clases interactivas:

Las sesiones sincrónicas semanales permiten la participación activa del estudiante a través de demostraciones en vivo, resolución colaborativa de problemas y espacio para preguntas y retroalimentación directa. Se fomenta el trabajo en equipo y el intercambio de experiencias entre los participantes.

- Actividades prácticas:

Cada módulo incluye ejercicios aplicados y proyectos pequeños que simulan escenarios reales del desarrollo profesional. Los estudiantes aprenderán haciendo, aplicando cada concepto técnico a situaciones concretas como el manejo de usuarios, acceso a datos o integración de servicios.

- Material asincrónico flexible:

Los estudiantes disponen de recursos digitales como videos explicativos, lecturas complementarias y retos de programación que pueden completar a su propio ritmo, permitiendo un aprendizaje más autónomo y adaptado a distintos estilos.

- Aprendizaje integral:

Se cubren temas clave del desarrollo back-end como:

- Programación del lado del servidor con Node.js.
- Conexión con bases de datos (PostgreSQL, MongoDB).
- Autenticación segura con JWT y control de accesos.
- Integración de APIs externas.
- Pruebas del sistema y depuración de errores.
- Implementación en la nube y despliegue final de aplicaciones.





UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA

Campus de la UNA
SAN LORENZO-PARAGUAY

ANEXO RESOLUCIÓN N° 0637/2026

Pág. 4/10

- Desarrollo profesional:

Además de los aspectos técnicos, se refuerzan habilidades blandas como el trabajo colaborativo, la comunicación efectiva en equipos de desarrollo y la resolución ética de problemas. Se alienta a los estudiantes a seguir buenas prácticas de documentación, revisión de código y mantenimiento sostenible de aplicaciones.

- Aspectos éticos y profesionales:

A lo largo del curso se destaca la importancia de desarrollar software de manera responsable, cuidando la privacidad y seguridad de los usuarios, respetando la legalidad en el uso de datos y fomentando la inclusión. También se promueven valores como el trabajo colaborativo, la transparencia en el desarrollo y la mejora continua.

Objetivo del curso

Al finalizar con éxito este curso los estudiantes serán capaces de:

- Diseñar y desarrollar servidores y aplicaciones del lado del servidor utilizando Node.js.
- Integrar bases de datos relacionales y no relacionales para el manejo eficiente de información.
- Implementar mecanismos de autenticación y seguridad en aplicaciones web.
- Consumir e integrar APIs de terceros y preparar aplicaciones para su despliegue en entornos de producción.

Subtemas:

- **Programación del lado del servidor con Node.js:**
Introducción al entorno de ejecución de Node.js, creación de servidores web, manejo de rutas, middlewares, y organización del código usando patrones como MVC.
- **Conexión con bases de datos (PostgreSQL, MongoDB):**
Uso de controladores y ORMs para conectar aplicaciones con bases de datos relacionales y no relacionales. Inserción, consulta y actualización de datos.
Modelado de esquemas y relaciones.
- **Autenticación segura con JWT y control de accesos:**
Implementación de procesos de login con tokens JWT, protección de rutas privadas, autorización basada en roles y medidas contra vulnerabilidades comunes como ataques de inyección y CSRF.
- **Integración de APIs externas:**
Consumo de servicios REST desde el servidor, gestión de claves de acceso (API keys), manejo de errores, y transformación de datos de terceros.
- **Pruebas del sistema y depuración de errores:**
Introducción a pruebas unitarias y de integración con herramientas como Jest y Postman. Técnicas de depuración, logging, y análisis de errores comunes en producción.
- **Implementación en la nube y despliegue final de aplicaciones:**
Configuración de entornos de producción, despliegue de aplicaciones en plataformas como Render, variables de entorno, manejo de logs y supervisión del sistema en vivo.

Política de calificación

Criterios de evaluación:

- Tareas, prácticas y cuestionarios (40%)

Los estudiantes deberán completar ejercicios prácticos, cuestionarios y desafíos semanales diseñados para reforzar los conocimientos adquiridos en cada módulo del curso. Estas actividades permitirán evaluar la aplicación de conceptos clave como la creación de rutas en Express, conexión con bases de datos, autenticación, consumo de APIs y pruebas del sistema.

- Proyecto final (40%)

Al finalizar el curso, los estudiantes deberán desarrollar una aplicación back-end funcional que integre los conocimientos adquiridos. El proyecto deberá incluir:





UNIVERSIDAD NACIONAL DE ASUNCIÓN FACULTAD POLITÉCNICA

Campus de la UNA
SAN LORENZO-PARAGUAY

ANEXO RESOLUCIÓN N° 0637/2026

Pág. 5/10

- Un servidor con Express y Node.js
- Conexión a base de datos (PostgreSQL o MongoDB)
- Autenticación con JWT
- Consumo de al menos una API externa
- Despliegue en la nube

El proyecto será evaluado según criterios de funcionalidad, estructura del código, buenas prácticas, seguridad y documentación.

- Examen final teórico-práctico (20%)

Se aplicará un examen que combine preguntas teóricas (conceptos, flujos, seguridad) con ejercicios prácticos breves (ej. escritura de rutas o funciones middleware). Este examen busca medir la comprensión integral de los temas abordados durante el curso.

- Calificación final

La calificación final será el resultado ponderado de cada uno de los criterios mencionados. Para aprobar el curso, el estudiante deberá obtener un mínimo de 70/100 puntos en la evaluación total.

- Créditos extra

Se podrá otorgar hasta un 5% adicional en la nota final por:

- Participación activa en foros y sesiones interactivas
- Propuestas técnicas innovadoras en el proyecto final
- Ayuda técnica y colaborativa en la comunidad del curso

Libros de texto y otros materiales necesarios

Referencias en línea y documentación oficial:

- [Node.js Documentation](#) – Documentación de Node.js.
- [Express.js Guide](#) – Documentación de Express.
- [MDN Web Docs – JavaScript](#) – Documentación de Javascript.
- [MongoDB University](#) – Cursos gratuitos para aprender sobre MongoDB.
- [PostgreSQL Documentation](#) – Documentación de PostgreSQL.
- [JWT](#) – Documentación de JWT.
- [Render](#) – Documentación de Render.
- [JEST](#) - Guía de Jest para Node.js
- [Symbaloo](#)– Escritorio virtual accesible desde cualquier dispositivo con conexión a internet.

Software requerido (SW)

- Visual Studio Code – Editor de código ligero y extensible, gratuito. <https://code.visualstudio.com/>
- Node.js y npm – Entorno de ejecución JavaScript para el servidor y gestor de paquetes.
- <https://nodejs.org/>
- Postman – Herramienta para pruebas de APIs (GET, POST, etc.).
- Postman
- MongoDB Compass / PgAdmin – Para la visualización y administración de bases de datos MongoDB y PostgreSQL, respectivamente.

Requisitos de hardware (HW)

- Computadora personal con al menos 4 GB de RAM, sistema operativo actualizado y conexión a internet.
- Permisos para instalar software y configurar entornos de desarrollo local (Node.js, MongoDB, etc.).





Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA

ANEXO RESOLUCIÓN Nº 0637/2026

Pág. 6/10

Otros recursos opcionales:

- **GitHub** – Para gestionar y compartir proyectos de código

Tareas y exámenes

Las tareas se enfocan en la resolución de ejercicios prácticos.

Tarea 1 – Configuración de entorno y creación de un servidor básico en Node.js (5%)

- Instalar Node.js y crear un proyecto básico.
- Configurar el entorno de desarrollo en la máquina local de manera adecuada.
- Crear y ejecutar el primer script.

Tarea 2 – Conexión con base de datos (PostgreSQL o MongoDB) (5%)

- Instalar y configurar una base de datos local (PostgreSQL o MongoDB).
- Conectar la base de datos con la aplicación Node.js.
- Crear una ruta que permita insertar y recuperar datos de la base de datos (por ejemplo, gestionar usuarios).

Tarea 3 – Implementación de autenticación con JWT (5%)

- Implementar un sistema básico de autenticación con JSON Web Tokens (JWT).
- Crear rutas protegidas que solo permitan acceso a usuarios autenticados.
- Practicar el manejo de sesiones y tokens en Node.js.

Tarea 4 – Consumo de API externa (5%)

- Realizar peticiones a una API externa pública (PokeAPI).
- Mostrar los datos obtenidos en una ruta de Express.
- Procesar la información de la API y devolverla en el formato adecuado.

Tarea 5 – Gestión de rutas y controladores (5%)

- Implementar múltiples rutas en la aplicación Express.
- Crear controladores para manejar la lógica de negocio.
- Organizar el proyecto utilizando una estructura modular para escalabilidad.

Tarea 6 – Implementación de pruebas unitarias (5%)

- Introducir el concepto de pruebas unitarias en el back-end
- Implementar pruebas para las rutas y funciones de la API.
- Asegurar que el código está cubierto por pruebas.

Tarea 7 – Despliegue de la aplicación en la nube (10%)

- Desplegar la aplicación en un entorno de producción, como **Render**.
- Verificar que la aplicación se ejecute correctamente desde la nube.





Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN FACULTAD POLITÉCNICA

ANEXO RESOLUCIÓN Nº 0637/2026

Pág. 7/10

Examen Final (20%)

Contenido:

- Preguntas de opción múltiple y desarrollo sobre los conceptos fundamentales cubiertos en el curso, como rutas en Express, autenticación, manejo de bases de datos y APIs.

Proyecto Final (40%)

El proyecto final consistirá en la creación de una **aplicación back-end completa** que integre todos los elementos aprendidos. El proyecto debe incluir:

- **Servidor Express** configurado adecuadamente.
- **Conexión a base de datos** (PostgreSQL o MongoDB) para almacenamiento persistente de datos.
- **Autenticación JWT** para manejo de usuarios.
- **Consumo de API externa** y manejo adecuado de respuestas.
- **Implementación de pruebas unitarias** para asegurar la calidad del código.
- **Despliegue en la nube** (Render u otra opción).
- **Documentación técnica** y explicación del código.

Actividades del curso

1. Clases interactivas

Formato: Exposición dinámica de conceptos clave con participación activa de los estudiantes.

Actividades:

- Preguntas y respuestas en vivo sobre Node.js, Express, bases de datos y seguridad.
- Encuestas rápidas o mini-quizzes para verificar comprensión de temas como JWT o APIs.
- Demostraciones en vivo de código (por ejemplo, creación de un servidor Express o conexión a una base de datos).
- Aplicación de la técnica de gamificación mediante juegos de repaso (como Kahoot o quizzes competitivos).

2. Sesiones de debate y análisis

Formato: Discusión crítica sobre temas relevantes del desarrollo back-end.

Actividades:

- Estudio de casos sobre vulnerabilidades reales (ej. mala gestión de autenticación).
- Análisis de código de APIs mal estructuradas para identificar fallos de seguridad, escalabilidad o claridad.
- Debates sobre frameworks y lenguajes: ¿cuándo usar Node.js, Django o Flask?
- Reflexión sobre la ética en el desarrollo de software: privacidad de los datos, seguridad y responsabilidad profesional.

3. Sesiones prácticas y desafíos de codificación

Formato: Aplicación práctica de los conocimientos mediante ejercicios guiados y desafíos semanales.

Actividades:

- Laboratorios para construir rutas, controladores y middlewares en Express.
- Ejercicios prácticos para integrar y consultar bases de datos PostgreSQL y MongoDB.
- Desafíos como, por ejemplo: implementar autenticación, consumir una API externa o crear pruebas automáticas.
- Retroalimentación personalizada del código por parte del docente o entre pares

4. Proyecto integrador

Formato: Desarrollo progresivo de una aplicación back-end completa durante las 8 semanas.

Actividades:

- Definición de una idea de aplicación (ej. sistema de gestión, blog, plataforma educativa).





UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA

Campus de la UNA
SAN LORENZO-PARAGUAY

ANEXO RESOLUCIÓN N° 0637/2026

Pág. 8/10

- Implementación de rutas, controladores, base de datos, autenticación y consumo de APIs externas.
- Despliegue en la nube (Render u otra).
- Presentación final del proyecto, incluyendo demo funcional y reflexión escrita o verbal sobre el proceso de desarrollo.

Cronograma del curso

Semana	Tema	Tipo de clases	Materiales
1	Introducción al Desarrollo Back-End	Clase interactiva Debate	Node.js Documentation Symbaloo
2	Programación del lado del servidor con Node.js y Express	Clase interactiva Taller	Node.js Documentation Express.js Guide Symbaloo
3	Bases de Datos Relacionales con PostgreSQL	Clase interactiva Taller	PostgreSQL Documentation Symbaloo
4	Bases de Datos No Relacionales con MongoDB	Clase interactiva Taller	MongoDB University Symbaloo
5	Autenticación y Seguridad con JWT	Clase interactiva Taller	JWT Node.js Documentation Express.js Guide Symbaloo
6	Integración de APIs Externas	Clase interactiva Taller	PokeAPI Symbaloo
7	Pruebas y Depuración de Aplicaciones	Clase interactiva Taller	JEST Postman Symbaloo
8	Despliegue en la Nube y Proyecto final	Taller Examen final	Render

Contenidos del curso

Semana 1: Introducción al Desarrollo Back-End

Objetivo:

Comprender los fundamentos del desarrollo del lado del servidor, las diferencias entre back-end y front-end, y explorar las tecnologías actuales, incluyendo Node.js y Python.

Materiales:

- [Node.js Documentation](#)
- [Symbaloo](#)

Actividades:

- Clase teórica: Fundamentos del back-end, arquitectura cliente-servidor, comparativa Node.js vs Python.
- Práctica: Instalar Node.js, VS Code y crear un "Hello World" en Node.js.
- Foro: ¿Qué tecnología back-end te interesa más y por qué?

Semana 2: Programación del lado del servidor con Node.js y Express

Objetivo:
Aprender a construir un servidor básico usando Node.js y Express, manejar rutas, middlewares y estructurar proyectos de forma modular.

Materiales:

- [Node.js Documentation](#)
- [Express.js Guide](#)
- [Symbaloo](#)

Actividades:

- Clase teórica: Servidores con Express, definición de rutas y middlewares.
- Práctica: Crear un servidor con rutas básicas y middleware personalizado.
- Foro: Comparte tu código base y explica la función de un middleware.





UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA

Campus de la UNA
SAN LORENZO-PARAGUAY

ANEXO RESOLUCIÓN N° 0637/2026

Pág. 9/10

Semana 3: Bases de Datos Relacionales con PostgreSQL

Objetivo:

Conocer los fundamentos de SQL y conectar una base de datos PostgreSQL a una aplicación Express usando un ORM o librería nativa

Materiales:

- [PostgreSQL Documentation](#)
- [Symbaloo](#)

Actividades:

- Clase teórica: Introducción a SQL, conexión a PostgreSQL, consultas básicas.
- Práctica: Conectar una app Express a PostgreSQL y hacer operaciones CRUD.
- Foro: ¿Qué ventajas encuentras en usar una base de datos relacional?

Semana 4: Bases de Datos No Relacionales con MongoDB

Objetivo:

Entender el modelo de datos NoSQL con MongoDB y conectarlo a una aplicación Node.js usando Mongoose.

Materiales:

- [MongoDB University](#)
- [Symbaloo](#)

Actividades:

- Clase teórica: ¿Qué es NoSQL?, esquemas con Mongoose, comparativa SQL vs NoSQL.
- Práctica: Crear modelos con Mongoose y realizar operaciones CRUD.
- Foro: Comparte tu modelo de usuario y una consulta con filtros.

Semana 5: Autenticación y Seguridad con JWT

Objetivo:

Implementar autenticación con JSON Web Tokens, proteger rutas y aplicar buenas prácticas de seguridad en aplicaciones web.

Materiales:

- [JWT](#)
- [Node.js Documentation](#)
- [Express.js Guide](#)
- [Symbaloo](#)

Actividades:

- Clase teórica: Autenticación con JWT, hash de contraseñas, protección de rutas.
- Práctica: Implementar login y generación de tokens con JWT.
- Foro: ¿Por qué es importante proteger nuestras rutas y cómo lo haces?





Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN FACULTAD POLITÉCNICA

ANEXO RESOLUCIÓN Nº 0637/2026

Pág. 10/10

Semana 6: Integración de APIs Externas Objetivo:

Comprender cómo consumir APIs públicas desde una aplicación Node.js y procesar las respuestas en rutas de Express.

Materiales:

- [PokeAPI](#)
- [Symbaloo](#)

Actividades:

- Clase teórica: ¿Qué es una API externa? Estructura de una petición HTTP. Métodos GET y POST.
- Ejercicio práctico:
- Crear una ruta en Express que consuma datos de una API pública (PokeAPI).
- Procesar la respuesta (por ejemplo, mostrar solo nombre, email o habilidad).
- Devolver el resultado como JSON desde la app local.
- Desafío (opcional): Mostrar información combinada de dos fuentes o aplicar filtros antes de retornar el resultado.

Semana 7: Pruebas y Depuración de Aplicaciones Objetivo:

Aplicar pruebas básicas en el Back-End con Jest y Postman, y aprender técnicas para depurar errores y mejorar la calidad del código.

Materiales:

- [JEST](#)
- [Postman](#)
- [Symbaloo](#)

Actividades:

- Clase teórica: Tipos de pruebas (unitarias, integración), depuración de código.
- Práctica: Escribir pruebas simples para una API y detectar errores.
- Foro: ¿Qué aprendiste depurando tu app?

Semana 8: Implementación y Despliegue en la Nube Objetivo:

Desplegar una aplicación back-end en la nube, gestionar variables de entorno y preparar la aplicación para producción.

Materiales:

- Render - – Documentación de Render.

Actividades:

- Clase teórica: Configuración de entorno, despliegue, variables de entorno y logging.
- Práctica: Subir la aplicación completa a un entorno en la nube.
- Evaluación final: Presentación del proyecto funcional y retroalimentación grupal

