



# TRABAJO FINAL INTEGRADOR

PROGRAMACIÓN III - 2025 2do cuatrimestre  
TECNICATURA UNIVERSITARIA EN DESARROLLO WEB

## OBJETIVOS

El Trabajo Final Integrador tiene como objetivos que el estudiante:

- Ponga en práctica todos los conocimientos adquiridos durante el cursado de la asignatura desarrollando una API REST.
- Defina la estructura de los documentos y las relaciones entre estos.
- Interactúe con una API REST intercambiando información.

## CONDICIONES DE ENTREGA

- El Trabajo Final Integrador deberá ser:
  - Realizado en forma grupal.
  - Los grupos deberán contar con un mínimo de 4 y un máximo de 6 alumnos.
  - Cargado en la sección del Campus Virtual correspondiente, en un archivo comprimido en formato ZIP, RAR, TAR.GZ u otro formato similar. También se acepta subir archivo TXT con URL a repositorio GitHub.
  - Deberá indicarse número de grupo, apellido y nombre de los integrantes.
  - Entregado antes de la fecha límite informada en el campus.
- Las soluciones deben ser de autoría propia. Aquellas que se detecten como idénticas entre diferentes grupos serán desaprobadas para todos los involucrados.
- Uso de Inteligencia Artificial (IA): Las herramientas de IA (como generadores de código o texto) pueden ser utilizadas únicamente como asistentes para el aprendizaje, depuración o para generar código repetitivo. No deben ser usadas para generar la solución completa.
- Se valorarán la exactitud, eficiencia y prolijidad (identación y otras buenas prácticas) de las soluciones planteadas, así como también la calidad de la exposición realizada.
- Los integrantes del equipo deberán realizar un video de exposición del proyecto, detallando su funcionamiento y las decisiones de diseño tomadas. Este video deberá:
  - Mostrar la pantalla del dispositivo donde se visualice el código y la aplicación en ejecución.
  - Exponer la voz de todos los integrantes explicando el proyecto.
  - Idealmente, capturar en video a los integrantes mientras realizan la explicación.
  - Cargarse en YouTube, Google Drive, Vimeo u otro.

## ENUNCIADO

La empresa para la que usted trabaja ha inaugurado una nueva unidad de negocios llamada “**PROGIII**”. Dado su excelente desempeño en proyectos anteriores —en particular en el sistema de *Gestión de Reservas de Casas de Cumpleaños*— ha sido asignado como parte del equipo responsable de diseñar y desarrollar la **API REST** que se integrará con un cliente web previamente desarrollado. La misma deberá contemplar autenticación, autorización y validación de datos para la gestión de reservas de salones de cumpleaños.

## FECHAS DE ENTREGA

- Primera entrega (avance funcional mínimo): 09/10/2025
  - BREAD (Browse, Read, Edit, Add, Delete) completo de alguna entidad del API contemplando las mejores prácticas vistas en clase.
- Entrega final (versión completa): 06/11/2025
  - Con todos los requerimientos completados.
- Recuperatorio de entrega final (versión completa): 18/11/2025
  - Con todos los requerimientos completados.

## REQUISITOS FUNCIONALES

Roles y permisos:

### Cliente

- Iniciar sesión (autenticación).
- Limitado para:
  - Reservas (crear, listar).
- Listado de:
  - Salones.
  - Servicios.
  - Turnos.
- Recepción de notificaciones automáticas cuando se confirma una reserva.

### Empleado

- Iniciar sesión (autenticación).
- Listado de:
  - Reservas.
  - Clientes.
- BREAD completo para:
  - Salones, Servicios, Turnos.

### Administrador

- Iniciar sesión (autenticación).

- BREAD completo para:
  - Reservas, Salones, Servicios, Turnos, Usuarios.
- Generación de:
  - Informes estadísticos (a través de procedimientos almacenados).
  - Reportes de reservas en: PDF – CSV – otros (No JSON).
- Recepción de notificaciones automáticas cuando se realiza una reserva.

### ASPECTOS TÉCNICOS REQUERIDOS

- Autenticación con JWT.
- Autorización por roles.
- Uso del framework Express.
- Persistencia de datos en MySQL.
- Buen manejo de errores y respuestas HTTP apropiadas.
- Documentación del API haciendo uso de Swagger.
- Validaciones utilizando middleware como express-validator.

### RESTRICCIONES Y REGLAS DE NEGOCIO:

- Una reserva puede ser modificada únicamente por un administrador.
- Las estadísticas deben generarse exclusivamente mediante procedimientos almacenados (stored procedures).
- Los informes en PDF deben contener los datos de reservas con sus servicios, salón, turno y cliente.
- Los “delete” no serán borrados físicos, se utilizarán “soft delete”, es decir se utilizará el campo **activo** para indicar si el registro de la tabla está borrado o no.

### MODELO DE DATOS

- **salones:** salon\_id, titulo, direccion, latitud, longitud, capacidad, importe, creado, modificado, activo.
- **servicios:** servicio\_id, descripcion, importe, creado, modificado, activo.
- **turnos:** turno\_id, orden, hora\_desde, hora\_hasta, creado, modificado, activo.
- **reservas:** reserva\_id, fecha\_reserva, salon\_id, usuario\_id, turno\_id, foto\_cumpleaniero, tematica, importe\_salon, importe\_total, creado, modificado, activo.
- **reservas\_servicios:** reserva\_servicio\_id, reserva\_id, servicio\_id, importe, creado, modificado.
- **usuarios:** usuario\_id, nombre, apellido, nombre\_usuario, contrasenia, tipo\_usuario, celular, foto, creado, modificado, activo.

**EXTRAS**

Cada grupo podrá agregar una funcionalidad extra al desarrollo.

Lista de ejemplos:

- Sistema de encuestas: después de realizado el cumpleaños permitir al cliente completar una encuesta de satisfacción (agregar una tabla que se relacione con la reserva).
- Sistema de comentarios/observaciones: permitir a los empleados u administradores agregar comentarios/observaciones sobre las reservas (ej. “Pago 50% de la reserva”).
- Reinicio de contraseña para los usuarios.
- Dashboard de estadísticas simple: HTML, CSS, JS.
- Recordatorio al cliente 24hs antes de la reserva.
- Sistema de auditoría: historial de acciones por usuario básico y sencillo.
- Registro de usuario tipo “cliente”.
- Registro de invitados.