

## Trabajo Práctico Final

*El fin de este trabajo práctico es que puedas probarte como desarrollador de software. A diferencia de los trabajos prácticos anteriores donde la consigna está completamente pautada, este trabajo práctico te invita a que lleves adelante tu propio proyecto, sin un enunciado de ejercicios definido. La propuesta, el diseño y la implementación del proyecto serán de tu autoría, aunque siempre con el acompañamiento docente.*

### Objetivos

- Integrar en un único trabajo todos los conocimientos adquiridos a lo largo de las materias de programación cursadas y en curso.
- Adquirir un mayor entendimiento de las metodologías de trabajo aplicadas durante la elaboración de software, incluyendo pero no limitándose a:
  - Análisis del proyecto.
  - Definición del alcance del proyecto.
  - Planificación del proyecto.
  - Presentación oral de informes de avance.
  - Elaboración de diagramas UML para facilitar el entendimiento del sistema.
  - Entrega semanal de un ejecutable con un incremento visible y comprobable de sus funcionalidades, y libre de errores.

### Requisitos mínimos del proyecto

- El sistema a abordar debe presentar una arquitectura de Cliente/Servidor. Dentro de la materia, se evaluará únicamente lo correspondiente al lado del servidor.
- El servidor debe estar plenamente desarrollado utilizando NodeJS y Express, utilizando siempre que se pueda, las técnicas vistas durante las primeras unidades, así como también aprovechando las nuevas sintaxis y funcionalidades provistas a partir de ES5, como por ejemplo, uso de let/const, promesas/async/await, etc.
- El servidor debe mostrar una clara separación en capas. Entre ellas, debe contar con una capa de persistencia que se conecte a una base de datos (de libre elección), y correctamente modularizada, de acuerdo a los lineamientos de diseño vistos durante las clases correspondientes a la unidad de Arquitectura y Diseño.
- Implementar la persistencia de sus variables de configuración utilizando archivos.
- Resolver al menos 1 caso de uso de complejidad de moderada a alta por integrante del grupo. Operaciones de Alta, Baja, Modificación y Consulta se consideran de baja complejidad. Consideramos de complejidad moderada a alta (dependiendo del tipo de solución que elijas) aquellos que transforman la información del sistema para generar nueva información (módulos estadísticos, importadores y exportadores de datos).

- El sistema debe estar acompañado de la documentación que detalle toda la funcionalidad desarrollada, los puntos de acceso al servidor, y el comportamiento esperado, tal como se muestra en el apunte sobre Documentación de servidores.

## Modalidad de trabajo

El trabajo se realizará en grupos de 4 personas, exclusivamente.

La temática del proyecto será definida íntegramente por cada grupo. Deberán idear y presentar las funcionalidades deseadas de su proyecto, y luego elegir cuáles de ellas se incluirán en el trabajo práctico. A fin de enriquecer los proyectos, se dispondrá de un tiempo para brainstorming en clase entre estudiantes y docentes. En esta etapa, los docentes podrán ayudar en forma de asesores u orientadores a delimitar el alcance de los proyectos.

Cada semana, deberá presentarse el estado de avance del proyecto (en forma oral), el proyecto funcionando, y la documentación sobre la funcionalidad desarrollada, si correspondiese.

El orden de las presentaciones podrá ser definido de manera informal y de común acuerdo entre los docentes y los presentadores, de antemano, o al comienzo de cada clase.

## Forma de evaluación

Antes que nada, para poder considerar al trabajo aprobado, éste deberán contar con el diseño pre-aprobado por el docente, validando que los casos de uso propuestos son suficientes para el dar cuenta de los conocimientos adquiridos durante la materia.

Una vez considerado esto, se procede a contabilizar todos los avances del trabajo en pos de medir el porcentaje de completitud del mismo. Para ello, se tomarán como ítems separados todos los requisitos funcionales y no funcionales pedidos, y sobre eso se medirá dicho porcentaje. Se tendrá en cuenta el grado de avance de los casos de uso propuestos, la calidad del diseño implementado y de su codificación. Un caso de uso y su implementación no se considerarán completos si su código contiene errores, si no hay validaciones de datos comprobables a través de casos de prueba o módulos de testeo o si su documentación está ausente o incompleta.

Finalmente, la nota (de 1 a 10) se calcula a partir del porcentaje obtenido:

Porcentaje de completitud	Nota
Menos de un 70%	2
Entre 70% y 89%	4 a 6
A partir de 90% y hasta 100% o más	7 a 10