

## Trabajo intermedio y final (Obligatorios)

Conforme transcurre la cursada se irá realizando un trabajo final para ir integrando los conceptos aprendidos en cada unidad. El trabajo pretende tener una continuidad desde la unidad 1 hasta la unidad 8, y puede ser realizado a elección del alumno mediante el uso de Tkinter o usando sólo Python a elección.

**Nivel de exigencia:** Alto

**Objetivos:** Probar los conocimientos adquiridos y que el alumno comience a pasar de la teoría a la práctica.

**Participantes:** El trabajo puede ser realizado de forma individual pero para afianzar el trabajo en equipo y la participación, se permite que los alumnos puedan conformar grupos de hasta 5 personas.

### Alcance de la aplicación.

Para la aprobación final del curso el alumno debe realizar una aplicación que podrá ir completando a medida que se van presentando los temas en el nivel intermedio en las distintas unidades. No es requisito realizarla en tkinter, puede ser realizada sin el uso de una GUI.

La aplicación deberá contener tanto lo aprendido en el nivel inicial como lo presentado en el nivel intermedio. A continuación se indican los puntos que debe poseer indicando para los alumnos que se incorporan en el nivel intermedio los puntos presentados en el nivel inicial.

1) Según los temas dados en el nivel inicial.

- Crear un abmc (Crud) de datos, que permita cargar como mínimo dos campos (titulo y descripción)
- Guardar los registros en una base de datos del tipo SQLite3.
- Validar el código del campo título para admitir sólo alfanuméricos. Se puede usar la siguiente regex o utilizar una propia: `patron="^[A-Za-z]+(?:[ _][A-Za-z]+)*$"`.
- El código debe cumplir con PEP8

2) Temas adicionados durante la cursada del nivel intermedio.

La funcionalidad de interacción con la base de datos y validación de campos debe ubicarse en módulos aparte.

La app debe realizarse según el paradigma de POO (Dado a partir de la unidad 3)

Se puede implementar el patrón MVC (Introducido en la unidad 5) (OPCIONAL)

Se debe agregar el trabajo con excepciones (Introducido en la unidad 7)

Se debe documentar mediante pydoc la app (Introducido en la unidad 8)

## Entrega parcial

Entrega parcial - Unidad 4: Esta entrega es a los fines que chequear el trabajo de cada alumno y poder brindar ayuda en el caso de que se note que el alumno o los alumnos no están participando. No es considerada con nota será evaluada con:

10 si va todo bien

6 Si hay temas a modificar en cuyo caso se realizara una devolución

0 Si no se ha presentado trabajo.

Se espera que a esta altura los alumnos ya tengan definidos los grupos de trabajo si existen.

## Entrega final

En función a lo entregado las notas finales serán fijadas según el siguiente criterio:

	Nota mínima				Nota máxima
	bueno	bueno	muy bueno	excelente	sobresaliente
	6	7	8	9	10
Código según pep 8					
abmc (crud)					
Uso de modulos					
Validación de campos					
Paradigma de POO					
Implementación de excepciones					
Uso de clases para módulo de conexión a base de datos					
Documentación de la app					
Patron mvc					

## Comentario Importante para ser evaluados

Todos los alumnos que integren un grupo deben:

- Agregar el trabajo en la sección de entrega
- Comentar quienes integran el grupo.

## **TEMAS ADICIONALES**

- 1) Instalación de anaconda y configuración para el trabajo con numpy, pandas, matplotlib, tensorflow, keras,.....
- 2) Instalación y primera app con Django (para interpretación de patrón MVC)
- 3) Programación en python dentro de la app de Blender.
- 4) Instalación de Fenics y primer ejemplo para el análisis de la ecuación diferencia de Poisson con condiciones de contorno. Visualización 2D y 3D