

# Instituto de formación técnica superior N° 18

## Backend - Proyecto integrador

---

**Curso:** 2° Año

**Profesor:** Bonini Juan Ignacio

**Carga horaria:** 9 hs cátedra/semana

**Régimen:** Cuatrimestral

Para obtener la aprobación del proyecto, es fundamental entregar el código dentro del plazo establecido.

Adicionalmente, será necesario llevar a cabo una presentación individual del proyecto, donde se exhiba tanto su funcionamiento como el código realizado por el alumno.

Si el trabajo se llevó a cabo en equipo, el estudiante deberá describir la funcionalidad que lo distingue de los demás de manera individual.

Cabe destacar que la presentación deberá ajustarse a un límite máximo de 20 minutos de duración.

El proyecto integrador a entregar será una aplicación web con temática a elección, orientada a la gestión y que cumpla con los siguientes requisitos:

1. Debe realizarse de manera individual, aunque tienen la opción de realizarlo en grupo hasta un máximo de 5 participantes.
2. Si optan por realizar el proyecto en grupo, se entrega de forma individual y cada miembro del grupo debe realizar una funcionalidad adicional.
3. El proyecto debe subirse a un repositorio git en la nube y debe estar accesible para los demás integrantes del grupo y para el docente.
4. Deben realizar tantas aplicaciones de Django como sean necesarias. Como mínimo tres aplicaciones, lo ideal son cinco o más.
5. Deben existir al menos 10 rutas distintas, algunas de ellas con parámetros de consulta y otras con parámetros de ruta y deben estar bien aplicados. \* Ver punto 13 para el CRUD
6. Se deberá aplicar Programación Orientada a Objetos lo más que se pueda.
7. Las vistas deberán ser basadas en clases
8. Se deben utilizar templates que cumplan con las siguientes características:
  - 8.1. Deben estar asociados a las vistas. \* `TemplateView`
  - 8.2. Debe existir la relación de herencia entre los templates (utilizando el `base.html`).
  - 8.3. Deben aplicarse filtros en los templates.
  - 8.4. Los templates deberán utilizar archivos estáticos (js, css, etc). \* Archivos físicos
9. Se deben utilizar Django Forms que cumplan con las siguientes características:
  - 9.1. Tener validaciones tanto en el front-end como en el back-end (excluyente).
  - 9.2. Deberán estar asociados a los templates.
  - 9.3. Al menos un formulario basado en clases.
  - 9.4. Debe haber formularios asociado a un modelo y no asociado a modelos.
10. Todas las aplicaciones deben poseer sus modelos o incluso más de un modelo por aplicación cuando corresponda. Deben poseer una relación de uno a muchos, muchos a muchos, uno a uno, dependiendo del uso que se le quiera dar.
11. El proyecto debe funcionar utilizando un servidor de base de datos local dentro de los soportados. **Se puede utilizar SQLite**, pero será un plus instalar y configurar PostgreSQL. Además, debe poseer las migraciones necesarias para su correcto funcionamiento. (El `makemigrations` lo tienen que hacer ustedes)
12. Se debe poder acceder al Admin de Django y registrar todos los modelos y estos deberán cumplir con las siguientes características:
  - 12.1. Se deberá poder buscar por más de un campo a elección.
  - 12.2. Se deberán poder filtrar
  - 12.3. Se deberán poder ordenar
13. El proyecto debe poseer páginas a las que solo se pueda acceder mediante autenticación y la misma debe ser validada tanto en el front-end como en el back-end.
14. Se deberá crear altas, bajas y modificaciones en al menos 3 aplicaciones.