**🧩 SPRINT 2 — Gestión Académica Completa (Post-Matrícula)**

**🎯 Objetivo general**

Implementar las funcionalidades internas del sistema académico que el administrador utilizará después de la matrícula:  
**gestionar estudiantes, grupos, modalidades, carreras, asignaturas y horarios.**

**🧱 Módulos que desarrollarás**

**1️⃣ Gestión de Estudiantes**

**Objetivo:** permitir al admin ver y administrar los estudiantes ya matriculados.

**Tareas:**

* Endpoint /api/estudiantes → listar todos los estudiantes (JOIN con usuario y matrícula).
* Endpoint /api/estudiantes/:id → ver detalles.
* Endpoint /api/estudiantes/:id (PUT) → editar datos (teléfono, apoderado, etc.).
* Vista React GestionEstudiantes.jsx:
  + Tabla con filtros (por grupo, modalidad, carrera).
  + Botones **Editar / Eliminar / Ver Detalle**.
  + Barra de búsqueda por nombre o DNI.

**2️⃣ Gestión de Catálogos Académicos**

**Objetivo:** mantener actualizadas las opciones que aparecen en la matrícula.

**Tareas:**

* CRUD de **Modalidades** (/api/modalidades)
* CRUD de **Grupos** (/api/grupos)
* CRUD de **Carreras** (/api/carreras)
* Interfaz React Catalogos.jsx:
  + Usa pestañas (Modalidades | Grupos | Carreras).
  + Formularios simples con validaciones.

**Ejemplo de tablas afectadas:**  
Modalidad, Grupo, Carrera, Asignatura.

**3️⃣ Gestión de Asignaturas**

**Objetivo:** asignar materias a los grupos.

**Tareas:**

* Endpoint /api/asignaturas
* Relacionar cada asignatura con un grupo (grupoId).
* CRUD en frontend: agregar, editar, eliminar.

**4️⃣ Gestión de Horarios (nuevo módulo)**

**Objetivo:** registrar y consultar horarios de clase.

**Tabla Prisma:**

model Horario {

id Int @id @default(autoincrement())

grupoId Int

grupo Grupo @relation(fields: [grupoId], references: [id])

asignaturaId Int

asignatura Asignatura @relation(fields: [asignaturaId], references: [id])

docente String

dia String

horaInicio String

horaFin String

aula String?

createdAt DateTime @default(now())

}

**Endpoints:**

| **Método** | **Ruta** | **Descripción** |
| --- | --- | --- |
| GET | /api/horarios | Listar todos los horarios |
| GET | /api/horarios/:grupoId | Mostrar horarios por grupo |
| POST | /api/horarios | Crear nuevo horario |
| PUT | /api/horarios/:id | Editar horario |
| DELETE | /api/horarios/:id | Eliminar horario |

**Frontend GestionHorarios.jsx:**

* Tabla con día, hora, grupo, asignatura, docente, aula.
* Botón “Agregar Horario”.
* Filtros por grupo o día.

**5️⃣ (Opcional) Vistas para docentes**

Si luego agregas el rol **DOCENTE**, podrás:

* Mostrar su horario personal.
* Listar sus asignaturas.
* Registrar asistencia o notas.

**🔗 Flujo completo del Sprint 2**

1. El **admin** entra a su panel.
2. Gestiona **catálogos** (modalidades, grupos, carreras).
3. Crea **asignaturas** por grupo.
4. Define **horarios** para cada grupo y asignatura.
5. Visualiza los **estudiantes matriculados** por grupo o modalidad.
6. El sistema queda listo para Sprint 3 (Pagos + Reportes).

**🧮 Resultado esperado**

✅ Base académica totalmente configurable.  
✅ CRUDs funcionales de modalidades, grupos, carreras, asignaturas y horarios.  
✅ Listado y edición de estudiantes.  
✅ Estructura lista para pagos y reportes.