

Documentación General de Endpoints – Sistema de Gestión de Horarios

Autenticación

Método	URL	Descripción
POST	<code>/api/login/</code>	Login con credenciales (<code>correo</code> , <code>password</code>) y retorno de tokens JWT.

Módulos de Gestión – CRUD Protegido (solo SuperAdmin/Admin)

Todos estos módulos están protegidos y requieren token JWT y rol `SuperAdmin`.

Facultad

Método	URL	Descripción
GET	<code>/api/superadmin/facultades/</code>	Lista todas las facultades.
POST	<code>/api/superadmin/facultades/</code>	Crea una nueva facultad.
GET	<code>/api/superadmin/facultades/{id}/</code>	Obtiene una facultad específica.
PUT	<code>/api/superadmin/facultades/{id}/</code>	Actualiza una facultad.
DELETE	<code>/api/superadmin/facultades/{id}/</code>	Elimina una facultad.

Carrera

Método	URL	Descripción
--------	-----	-------------

GET	<code>/api/superadmin/carreras/</code>	Lista todas las carreras.
POST	<code>/api/superadmin/carreras/</code>	Crea una nueva carrera.
...	(CRUD completo)	

Asignatura

| URL base: `/api/superadmin/asignaturas/` (CRUD completo) |

Aula

| URL base: `/api/superadmin/aulas/` (CRUD completo) |

- Incluye validación para uso general (aulas para materias comunes).
-

HoraClase

| URL base: `/api/superadmin/horas-clase/` (CRUD completo) |

Relaciones y Asociaciones

Asignatura-Carrera

Método	URL	Descripción
GET	<code>/api/superadmin/asignatura-carrera/</code>	Lista todas las asociaciones.
POST	<code>/api/superadmin/asignatura-carrera/</code>	Asocia una asignatura a una carrera.
GET	<code>/api/superadmin/asignatura-carrera/asignatura/{id_asignatura}/</code>	Lista todas las carreras asociadas a una asignatura.

GET	<code>/api/superadmin/asignatura-carrera/carrera/{id_carrera}/</code>	Lista todas las asignaturas asociadas a una carrera.
-----	---	--

Usuario-Carrera

Método	URL	Descripción
GET	<code>/api/superadmin/usuario-carrera/</code>	Lista todas las asociaciones.
POST	<code>/api/superadmin/usuario-carrera/</code>	Asocia un usuario a una carrera.
GET	<code>/api/superadmin/usuario-carrera/usuario/{id}/</code>	Carreras asignadas a un usuario.

Usuario-Asignatura (Docente)

| URL base: `/api/superadmin/usuario-asignatura/` |

- Se impide que un docente esté duplicado en la misma asignatura y paralelo.
 - GET por `?usuario=id`, `?asignatura=id` y ambos combinados.
-

Usuario-Asignatura-Estudiante

| URL base: `/api/superadmin/usuario-asig-estudiante/` |

- Se evita duplicidad.
- Valida la capacidad del aula según el paralelo.
- Endpoints especiales:
 - `?usuario=id`: Lista las asignaturas del estudiante.
 - `?asignatura=id`: Lista los estudiantes inscritos en una materia.
- Respuesta JSON incluye:

json

CopyEdit

```
{
  "total": 25,
  "resultados": [...]
}
```



Horarios

| URL base: </api/superadmin/horario/> |

- CRUD completo con validaciones:
 - No se permite conflicto de hora con docente ni aula.
 - Materias comunes solo se asignan a aulas de uso general.
 - Validación de horario por paralelo.
- Respuesta GET ordenada por día de la semana y hora.

Método	URL	Descripción
GET	/api/superadmin/horario/	Lista todos los horarios disponibles según rol.
GET	/api/superadmin/horario/docente/{id}/	Lista todos los horarios de un docente específico.
GET	/api/superadmin/horario/estudiante/{id}/	Lista todos los horarios de un estudiante.

Los usuarios autenticados verán automáticamente **solo sus horarios** si son **Docente** o **Estudiante**.



Lógica General de Roles

Rol	Puede ver
SuperAdmin	Todos los datos.

Admin	Todos los datos.
Docente	Solo sus asignaturas y horarios.
Estudiante	Solo sus asignaturas inscritas y horarios.

✓ 1. Generación de horarios (SuperAdmin/Admin)

Los horarios se crean a través del endpoint:

```
bash
CopyEdit
POST http://localhost:8000/api/superadmin/horario/
```

Con un JSON como este:

```
json
CopyEdit
{
  "id_asignatura": 3,
  "id_aula": 4,
  "id_hora_clase": 5,
  "id_usuario": 2,
  "paralelo": 1,
  "semestre_lectivo": "2025-A"
}
```

Esto representa lo siguiente:

- El **docente con ID 2** va a dictar la **asignatura con ID 3**
- En el **aula 4**, el **día y hora** indicados en **id_hora_clase 5**
- Pertenece al **paralelo 1** y al **semestre_lectivo "2025-A"**

Este horario queda guardado en la tabla **horario**, la cual vincula **docente + asignatura + paralelo + hora + aula + semestre**.

✓ 2. ¿Cómo el estudiante consulta su horario?

Tu sistema no genera horarios *por separado* para estudiantes. En su lugar, los estudiantes **ven los horarios de sus materias inscritas** (según la tabla `usuario_asignatura_estudiante`) y el paralelo que les corresponde.

Cuando el estudiante hace un GET a `/api/superadmin/horario/`, el sistema hace esto:

- Consulta todas sus **asignaturas inscritas** en `usuario_asignatura_estudiante`
- Para cada asignatura, filtra el horario que tenga:
 - `id_asignatura` igual
 - `paralelo` igual al paralelo del estudiante
- Retorna esos horarios (día, aula, etc.)

📌 Ejemplo:

El estudiante está inscrito en:

json

CopyEdit

```
{
  "id_usuario": 7,
  "id_asignatura": 3,
  "paralelo": 1
}
```

Y en la tabla `horario` hay un horario para la `asignatura 3`, `paralelo 1` → entonces **ese es su horario**.

✓ 3. Consulta de horarios para cada rol

Rol	¿Qué ve?
SuperAdmin/Admin	Todos los horarios (GET <code>/api/superadmin/horario/</code>)
Docente	Solo los horarios en los que él es <code>id_usuario</code>
Estudiante	Solo ve los horarios de las asignaturas inscritas en su paralelo



¿Qué papel juegan estas tablas?

Tabla	¿Qué representa?	¿Quién la usa?
<code>asignatura_carrera</code>	Vincula qué asignaturas pertenecen a cada carrera	Referencia académica , usada por admins
<code>usuario_carrera</code>	Indica a qué carrera pertenece un usuario (estudiante o docente)	Base para sugerencias
<code>usuario_asignatura</code>	Qué asignaturas dicta un docente + en qué paralelo	Planificación docente
<code>usuario_asignatura_estudiante</code>	Qué asignaturas cursa un estudiante + en qué paralelo	Base para ver su horario



¿Entonces hay que asignar manualmente las asignaturas?

✓ **Sí. Pero te explico por qué es correcto así:**

Aunque un estudiante tenga una carrera y esa carrera tenga asignaturas, NO necesariamente está inscrito en todas ni en todas al mismo tiempo.

Ejemplo:

- Carrera: Ingeniería en Computación
- Tiene 40 asignaturas en total (1ro a 9no semestre)
- Estudiante Joel está en **3er semestre**

Entonces tú **NO debes** asignarle automáticamente todas las 40 asignaturas.

Solo le asignas (manualmente o por interfaz) las 5 o 6 del semestre que cursa, así:

json

CopyEdit

```
{
  "id_usuario": 7,
  "id_asignatura": 8,
  "paralelo": 1
}
```

Lo mismo pasa con los **docentes**, solo dictan algunas materias específicas, y **también** deben asignarse manualmente.

¿Cómo se conecta todo?

text

CopyEdit

FACULTAD

└─ CARRERA

└─ ASIGNATURA (asignatura_carrera)

└─ HORARIO

└─ docente (usuario_asignatura)

└─ estudiantes (usuario_asignatura_estudiante)

└─ USUARIO (usuario_carrera)

¿Se puede automatizar esta asignación?

Sí, más adelante podrías:

- Crear un endpoint o lógica que diga:
 - “Asigna automáticamente al estudiante X las asignaturas del semestre 3 de su carrera”.

Pero como cada estudiante puede tener:

- Materias atrasadas
- Repeticiones
- Paralelos distintos

... es mucho más seguro hacerlo **de forma controlada con tu endpoint actual**.