

FICCT School Sprint 1

⌵ Subject	INF412
-----------	--------

Flujo Completo de Implementación Frontend - FICCT School Backend

Este documento detalla todos los endpoints, parámetros y casos de uso que el frontend debe implementar para interactuar correctamente con el backend.

Índice

1. Autenticación y Usuarios
2. Gestión de Estudiantes
3. Gestión de Profesores
4. Gestión de Cursos
5. Gestión de Materias
6. Gestión de Periodos Académicos
7. Gestión de Calificaciones
8. Gestión de Inscripciones
9. Asignaciones de Profesores
10. Registro de Actividades del Sistema

1. Autenticación y Usuarios

1.1. Iniciar Sesión

- **URL:** `/api/auth/login/`
- **Método:** POST
- **Body:**

```
{
  "email": "usuario@example.com",
  "password": "contraseña"
}
```

- **Respuesta exitosa:**

```
{
  "access": "token_jwt_access",
  "refresh": "token_jwt_refresh"
}
```

1.2. Refrescar Token

- **URL:** `/api/auth/token/refresh/`
- **Método:** POST
- **Body:**

```
{
  "refresh": "token_jwt_refresh"
}
```

```
}
```

- **Respuesta exitosa:**

```
{
  "access": "nuevo_token_jwt_access"
}
```

1.3. Listar Usuarios

- **URL:** `/api/auth/users/`
- **Método:** GET
- **Parámetros opcionales:**
 - `search` : Búsqueda por nombre o email
 - `page` : Número de página
 - `page_size` : Elementos por página

1.4. Crear Usuario

- **URL:** `/api/auth/users/`
- **Método:** POST
- **Body:**

```
{
  "email": "usuario@example.com",
  "password": "contraseña",
  "first_name": "Nombre",
  "last_name": "Apellido",
  "groups": [1]
}
```

1.5. Cambiar Contraseña

- **URL:** `/api/auth/users/{id}/change_password/`
- **Método:** POST
- **Body:**

```
{
  "current_password": "contraseña_actual",
  "new_password": "nueva_contraseña"
}
```

1.6. Listar Roles

- **URL:** `/api/auth/groups/`
- **Método:** GET

1.7. Asignar Rol a Usuario

- **URL:** `/api/auth/groups/{id}/users/`
- **Método:** POST
- **Body:**

```
{
  "user_ids": [1, 2, 3]
}
```

2. Gestión de Estudiantes

2.1. Listar Estudiantes

- **URL:** `/api/auth/students/`
- **Método:** GET
- **Parámetros opcionales:**
 - `search` : Búsqueda por nombre o ID
 - `grade_level` : Filtrar por nivel
 - `page` : Número de página
 - `page_size` : Elementos por página

2.2. Obtener Estudiante

- **URL:** `/api/auth/students/{id}/`
- **Método:** GET

2.3. Crear Estudiante

- **URL:** `/api/auth/students/`
- **Método:** POST
- **Body:**

```
{
  "user": {
    "email": "estudiante@example.com",
    "password": "contraseña",
    "first_name": "Nombre",
    "last_name": "Apellido"
  },
  "student_id": "P1-001",
  "enrollment_date": "2025-02-01",
  "parent_name": "Nombre del Padre",
  "parent_contact": "71234567",
  "parent_email": "padre@example.com",
  "emergency_contact": "71234567"
}
```

2.4. Actualizar Estudiante

- **URL:** `/api/auth/students/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

2.5. Obtener Perfil Académico de Estudiante

- **URL:** `/api/auth/students/{id}/profile/`
- **Método:** GET
- **Respuesta:**

```
{
  "user": { /* datos del usuario */ },
  "student_id": "P1-001",
  "academic_metrics": {
    "current_average": 85.5,
    "attendance_percentage": 95.0
  }
}
```

```
}  
}
```

3. Gestión de Profesores

3.1. Listar Profesores

- **URL:** `/api/auth/teachers/`
- **Método:** GET
- **Parámetros opcionales:**
 - `search` : Búsqueda por nombre o ID
 - `specialization` : Filtrar por especialización
 - `page` : Número de página
 - `page_size` : Elementos por página

3.2. Obtener Profesor

- **URL:** `/api/auth/teachers/{id}/`
- **Método:** GET

3.3. Crear Profesor

- **URL:** `/api/auth/teachers/`
- **Método:** POST
- **Body:**

```
{  
  "user": {  
    "email": "profesor@example.com",  
    "password": "contraseña",  
    "first_name": "Nombre",  
    "last_name": "Apellido"  
  },  
  "teacher_id": "T001",  
  "specialization": "Matemáticas",  
  "qualification": "Licenciado en Educación",  
  "years_of_experience": 5,  
  "date_joined": "2020-02-01",  
  "employment_status": "active"  
}
```

3.4. Actualizar Profesor

- **URL:** `/api/auth/teachers/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

3.5. Obtener Cursos Asignados al Profesor

- **URL:** `/api/auth/teachers/{id}/courses/`
- **Método:** GET

4. Gestión de Cursos

4.1. Listar Cursos

- **URL:** `/api/academic/courses/`

- **Método:** GET
- **Parámetros opcionales:**
 - `search` : Búsqueda por nombre o código
 - `year` : Filtrar por año
 - `active` : Filtrar por estado activo (true/false)
 - `page` : Número de página
 - `page_size` : Elementos por página

4.2. Obtener Curso

- **URL:** `/api/academic/courses/{id}/`
- **Método:** GET

4.3. Crear Curso

- **URL:** `/api/academic/courses/`
- **Método:** POST
- **Body:**

```
{
  "name": "1ro de Primaria",
  "code": "P1",
  "description": "Primer grado de primaria",
  "year": 2025,
  "capacity": 30,
  "is_active": true
}
```

4.4. Actualizar Curso

- **URL:** `/api/academic/courses/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

4.5. Eliminar Curso

- **URL:** `/api/academic/courses/{id}/`
- **Método:** DELETE

4.6. Listar Estudiantes de un Curso

- **URL:** `/api/academic/courses/{id}/students/`
- **Método:** GET

5. Gestión de Materias

5.1. Listar Materias

- **URL:** `/api/academic/subjects/`
- **Método:** GET
- **Parámetros opcionales:**
 - `search` : Búsqueda por nombre o código
 - `course` : Filtrar por ID de curso
 - `page` : Número de página
 - `page_size` : Elementos por página

5.2. Obtener Materia

- **URL:** `/api/academic/subjects/{id}/`
- **Método:** GET

5.3. Crear Materia

- **URL:** `/api/academic/subjects/`
- **Método:** POST
- **Body:**

```
{
  "name": "Matemáticas",
  "code": "MAT",
  "description": "Matemáticas básicas",
  "credit_hours": 5
}
```

5.4. Actualizar Materia

- **URL:** `/api/academic/subjects/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

5.5. Eliminar Materia

- **URL:** `/api/academic/subjects/{id}/`
- **Método:** DELETE

5.6. Listar Cursos que Enseñan una Materia

- **URL:** `/api/academic/subjects/{id}/courses/`
 - **Método:** GET
-

6. Gestión de Periodos Académicos

6.1. Listar Periodos

- **URL:** `/api/academic/periods/`
- **Método:** GET
- **Parámetros opcionales:**
 - `search` : Búsqueda por nombre
 - `active` : Filtrar por estado activo (true/false)
 - `page` : Número de página
 - `page_size` : Elementos por página

6.2. Obtener Periodo

- **URL:** `/api/academic/periods/{id}/`
- **Método:** GET

6.3. Crear Periodo

- **URL:** `/api/academic/periods/`
- **Método:** POST
- **Body:**

```
{
  "name": "Año Escolar 2025",
  "start_date": "2025-02-01",
  "end_date": "2025-11-30",
  "is_active": true
}
```

6.4. Actualizar Periodo

- **URL:** `/api/academic/periods/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

6.5. Eliminar Periodo

- **URL:** `/api/academic/periods/{id}/`
- **Método:** DELETE

6.6. Obtener Periodo Actual

- **URL:** `/api/academic/periods/current/`
- **Método:** GET

7. Gestión de Calificaciones

7.1. Listar Calificaciones

- **URL:** `/api/academic/grades/`
- **Método:** GET
- **Parámetros opcionales:**
 - `student` : Filtrar por ID de estudiante
 - `course` : Filtrar por ID de curso
 - `subject` : Filtrar por ID de materia
 - `period` : Filtrar por ID de periodo
 - `min_value` : Filtrar por valor mínimo
 - `max_value` : Filtrar por valor máximo
 - `page` : Número de página
 - `page_size` : Elementos por página

7.2. Obtener Calificación

- **URL:** `/api/academic/grades/{id}/`
- **Método:** GET

7.3. Crear Calificación

- **URL:** `/api/academic/grades/`
- **Método:** POST
- **Body:**

```
{
  "student": 1,
  "course": 1,
  "subject": 1,
  "period": 1,
```

```
"value": 85,
"comments": "Buen desempeño"
}
```

7.4. Actualizar Calificación

- **URL:** `/api/academic/grades/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

7.5. Eliminar Calificación

- **URL:** `/api/academic/grades/{id}/`
- **Método:** DELETE

7.6. Obtener Calificaciones de un Estudiante

- **URL:** `/api/academic/grades/student_grades/`
- **Método:** GET
- **Parámetros requeridos:**
 - `student_id` : ID del estudiante
- **Parámetros opcionales:**
 - `period_id` : ID del periodo (por defecto el periodo activo)
- **Respuesta:**

```
{
  "Matemáticas - 1ro de Primaria": [
    {
      "id": 1,
      "value": 85,
      "comments": "Buen desempeño",
      "created_at": "2025-05-10T14:30:00Z"
    }
  ],
  "Lenguaje - 1ro de Primaria": [
    {
      "id": 2,
      "value": 90,
      "comments": "Excelente trabajo",
      "created_at": "2025-05-11T14:30:00Z"
    }
  ]
}
```

8. Gestión de Inscripciones

8.1. Listar Inscripciones

- **URL:** `/api/academic/enrollments/`
- **Método:** GET
- **Parámetros opcionales:**
 - `student` : Filtrar por ID de estudiante
 - `course` : Filtrar por ID de curso
 - `subject` : Filtrar por ID de materia
 - `period` : Filtrar por ID de periodo

- `status` : Filtrar por estado ('active', 'withdrawn', 'completed', 'pending')
- `page` : Número de página
- `page_size` : Elementos por página

8.2. Obtener Inscripción

- **URL:** `/api/academic/enrollments/{id}/`
- **Método:** GET

8.3. Crear Inscripción

- **URL:** `/api/academic/enrollments/`
- **Método:** POST
- **Body:**

```
{
  "student": 1,
  "course": 1,
  "subject": 1,
  "period": 1,
  "status": "active"
}
```

8.4. Actualizar Inscripción

- **URL:** `/api/academic/enrollments/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

8.5. Eliminar Inscripción

- **URL:** `/api/academic/enrollments/{id}/`
- **Método:** DELETE

8.6. Obtener Inscripciones de un Estudiante

- **URL:** `/api/academic/enrollments/student_enrollments/`
- **Método:** GET
- **Parámetros requeridos:**
 - `student_id` : ID del estudiante

9. Asignaciones de Profesores

9.1. Listar Asignaciones

- **URL:** `/api/academic/teacher-assignments/`
- **Método:** GET
- **Parámetros opcionales:**
 - `teacher` : Filtrar por ID de profesor
 - `course` : Filtrar por ID de curso
 - `subject` : Filtrar por ID de materia
 - `period` : Filtrar por ID de periodo
 - `page` : Número de página
 - `page_size` : Elementos por página

9.2. Obtener Asignación

- **URL:** `/api/academic/teacher-assignments/{id}/`
- **Método:** GET

9.3. Crear Asignación

- **URL:** `/api/academic/teacher-assignments/`
- **Método:** POST
- **Body:**

```
{
  "teacher": 1,
  "course": 1,
  "subject": 1,
  "period": 1,
  "is_primary": true
}
```

9.4. Actualizar Asignación

- **URL:** `/api/academic/teacher-assignments/{id}/`
- **Método:** PUT/PATCH
- **Body:** Misma estructura que la creación (campos a actualizar)

9.5. Eliminar Asignación

- **URL:** `/api/academic/teacher-assignments/{id}/`
- **Método:** DELETE

10. Registro de Actividades del Sistema

10.1. Listar Registros (Solo Admin)

- **URL:** `/api/logs/`
- **Método:** GET
- **Parámetros opcionales:**
 - `user` : Filtrar por ID de usuario
 - `action` : Filtrar por tipo de acción (CREATE, UPDATE, DELETE, ERROR, LOGIN, etc.)
 - `table_name` : Filtrar por nombre de tabla
 - `level` : Filtrar por nivel (INFO, WARNING, ERROR)
 - `start_date` : Filtrar registros desde una fecha (YYYY-MM-DD)
 - `end_date` : Filtrar registros hasta una fecha (YYYY-MM-DD)
 - `page` : Número de página
 - `page_size` : Elementos por página

10.2. Obtener Estadísticas de Registros (Solo Admin)

- **URL:** `/api/logs/statistics/`
- **Método:** GET
- **Respuesta:**

```
{
  "total_logs": 1250,
  "recent_logs": 78,
```

```
"by_action": [
  {"action": "CREATE", "count": 450},
  {"action": "UPDATE", "count": 320},
  {"action": "DELETE", "count": 120}
],
"by_table": [
  {"table_name": "Student", "count": 210},
  {"table_name": "Grade", "count": 450}
],
"by_level": [
  {"level": "INFO", "count": 1100},
  {"level": "WARNING", "count": 100},
  {"level": "ERROR", "count": 50}
]
}
```

Flujos Principales del Frontend

Flujo de Autenticación

- 1. Usuario ingresa credenciales (`/api/auth/login/`)
- 2. Guardar tokens en almacenamiento local
- 3. Configurar interceptor para incluir token en solicitudes
- 4. Implementar renovación automática de tokens (`/api/auth/token/refresh/`)
- 5. Cerrar sesión (eliminar tokens)

Flujo de Gestión de Estudiantes

- 1. Listar estudiantes (`/api/auth/students/`)
- 2. Ver perfil detallado (`/api/auth/students/{id}/profile/`)
- 3. Consultar calificaciones (`/api/academic/grades/student_grades/?student_id={id}`)
- 4. Consultar inscripciones (`/api/academic/enrollments/student_enrollments/?student_id={id}`)
- 5. Crear/editar estudiante

Flujo de Gestión de Profesores

- 1. Listar profesores (`/api/auth/teachers/`)
- 2. Ver perfil detallado (`/api/auth/teachers/{id}/`)
- 3. Consultar cursos asignados (`/api/auth/teachers/{id}/courses/`)
- 4. Crear/editar profesor

Flujo de Gestión Académica

- 1. Crear/gestionar periodos académicos (`/api/academic/periods/`)
- 2. Crear/gestionar cursos (`/api/academic/courses/`)
- 3. Crear/gestionar materias (`/api/academic/subjects/`)
- 4. Asignar profesores a cursos/materias (`/api/academic/teacher-assignments/`)
- 5. Inscribir estudiantes (`/api/academic/enrollments/`)
- 6. Registrar calificaciones (`/api/academic/grades/`)

Dashboard Administrativo

- 1. Consultar estadísticas de actividad (`/api/logs/statistics/`)
- 2. Monitorear acciones recientes (`/api/logs/`)
- 3. Ver resumen de periodos académicos (`/api/academic/periods/`)

Panel del Profesor

- 1. Ver cursos asignados (`/api/auth/teachers/{id}/courses/`)
- 2. Listar estudiantes por curso (`/api/academic/courses/{id}/students/`)
- 3. Gestionar calificaciones (`/api/academic/grades/`)

Portal del Estudiante

- 1. Ver perfil y métricas académicas (`/api/auth/students/{id}/profile/`)
- 2. Consultar calificaciones (`/api/academic/grades/student_grades/?student_id={id}`)
- 3. Ver cursos y materias inscritas (`/api/academic/enrollments/student_enrollments/?student_id={id}`)