

ARQUITECTURA DEL PROYECTO DEL EXAMEN

Vas a trabajar con tres aplicaciones web Spring Boot:

- **API_DATA_REST_ESTUDIANTES**: que se proporciona por completo y que solo debes ejecutar y utilizar. No debes modificar nada de este proyecto.
- **API_REST_EVALUACIONES**: que debes crear en base a especificaciones.
- **MVC_Avellaneda**: que debes crear en base a especificaciones.

RECURSOS INICIALES

- **API_DATA_REST_ESTUDIANTES**: carpeta con proyecto entero.
- **API_REST_EVALUACIONES**:
 - Los **ENTITIES** del modelo.
 - El recurso **import.sql** para tener datos iniciales.
 - Debes trabajar con una base de datos **H2 en memoria**.
- **MVC_Avellaneda**:
 - Los **ENTITIES** del modelo.
 - El recurso **import.sql** para tener datos iniciales.
 - Debes trabajar con una base de datos **H2 en memoria**.
 - **Plantillas Thymeleaf**.

API REST DATA ESTUDIANTES

- **Este proyecto está completo. No debes modificarlo. Ejecútalo cuando vayas a usarlo.**
- De todos los endpoints de este API REST, **deberás realizar peticiones desde el proyecto MVC solo a determinados endpoints.**
 - **Dichos endpoints requieren un token de autenticación.** Da igual el role.
 - Permite autenticación con:
 - **usuario admin (password admin)**
 - **usuario user (password user)**

ESPECIFICACIONES DE LOS ENDPOINTS QUE UTILIZARÁS EN EL PROYECTO MVC:

1) GET <http://localhost:8081/estudiantes/search/findByNia?nia=XXX>

Ejemplo:

GET http://localhost:8081/estudiantes/search/findByNia?nia=NIA_1002

Ejecuta el endpoint y verás el Json obtenido como respuesta. Ejemplo de respuesta:

```
{
  "nia": "NIA_1002",
  "nombre": "Juan",
  "primerApellido": "Pérez",
  "segundoApellido": "López",
  "email": "juan.perez2@example.com",
  "movil": "601123456",
  "direccion": "Avenida Gran Vía 23, Madrid",
  "fechaNacimiento": "1999-08-22",
  "edad": 25,
  "_links": {
    "self": {
      "href": "http://localhost:8081/estudiantes/2"
    },
    "estudiante": {
      "href": "http://localhost:8081/estudiantes/2"
    }
  }
}
```

2) GET

<http://localhost:8081/estudiantes/search/findByNombreAndPrimerApellidoAndSegundoApellido?nombre=XXX&primerApellido=XXX&segundoApellido=XXX>

Ejemplo:

GET
<http://localhost:8081/estudiantes/search/findByNombreAndPrimerApellidoAndSegundoApellido?nombre=Juan&primerApellido=Pérez&segundoApellido=Gómez>

Ejecuta el endpoint y verás el Json obtenido como respuesta. Ejemplo de respuesta:

```
{
  "nia": "NIA_1001",
  "nombre": "Juan",
  "primerApellido": "Pérez",
  "segundoApellido": "Gómez",
  "email": "juan.perez@example.com",
  "movil": "600123456",
  "direccion": "Calle Mayor 10, Madrid",
  "fechaNacimiento": "2000-05-15",
  "edad": 24,
  "_links": {
    "self": {
      "href": "http://localhost:8081/estudiantes/1"
    },
    "estudiante": {
      "href": "http://localhost:8081/estudiantes/1"
    }
  }
}
```

PARA PROBAR LOS ENDPOINTS: El proyecto contiene en la carpeta resource un archivo **endpoints.http** para realizar las pruebas a los endpoints. También puedes usar un cliente postman si prefieres.

SPRING MVC - AVELLANEDA

La aplicación web MVC del IES Avellaneda, tiene su propia autenticación y se conectará al **API_DATA_REST_ESTUDIANTES** para gestionar estudiantes.

Esta es la información de autenticación. Existen estos usuarios y roles en la base de datos:

- ROLE_TEACHER	USUARIO: profe	CONTRASEÑA: admin	ROLE_TEACHER
- ROLE_STUDENT	USUARIO: alumno	CONTRASEÑA: user	ROLE_STUDENT

Se proporcionan la siguientes plantillas Thymeleaf:

QUE NO DEBES MODIFICAR	QUE DEBES COMPLETAR
<ul style="list-style-type: none">- home- login- error- info- principal- ciclos-info- estudiantes/filtro	<ul style="list-style-type: none">- estudiantes/informe- estudiantes/registro

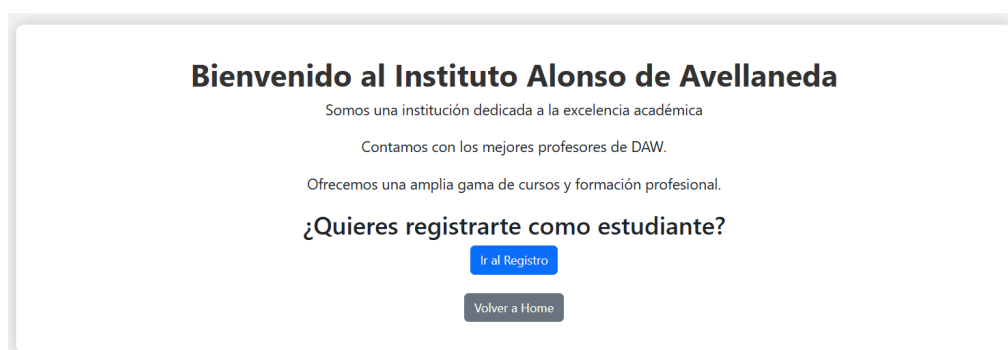
La conexión a <http://localhost:8080/> redirigirá a la página **home**:



Puedes entrar a la aplicación como **invitado** o **iniciando sesión**.

ENTRAR A LA APLICACIÓN COMO INVITADO

Se mostrará una página con información del instituto Avellaneda.

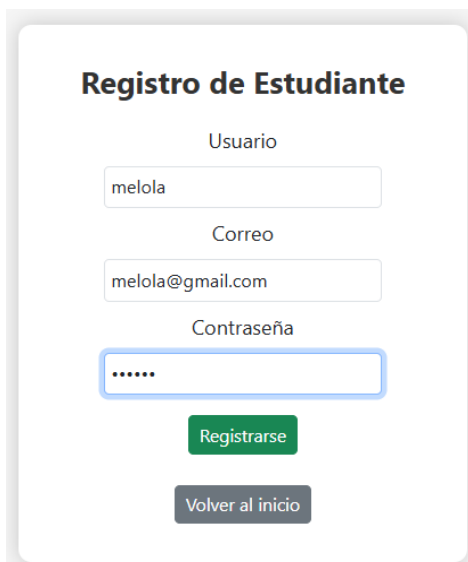


(*) **NAVEGACIÓN:** home → info (entrar como invitado) → volver a home

(*) **NAVEGACIÓN:** home → info → Ir a registro → volver a home (sin hacer registro)

REGISTRO / INICIO DE SESIÓN

1. Se mostrará la página **estudiantes/registro:**



Registro de Estudiante

Usuario
melola

Correo
melola@gmail.com

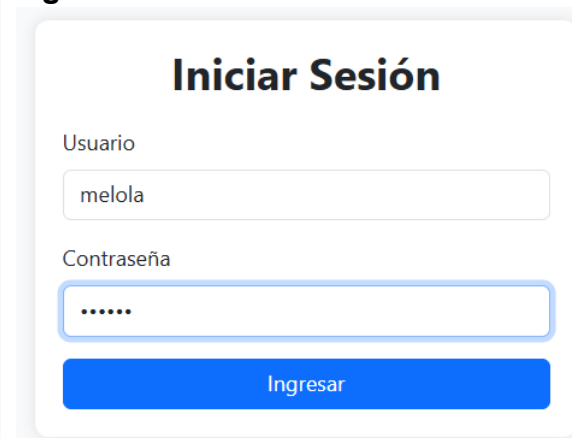
Contraseña

Registrarse

Volver al inicio

No hay que especificar el role, ya que cuando se registre un usuario, **automáticamente su role será ROLE_STUDENT.**

2. Tras el registro aparecerá la página de **login:**



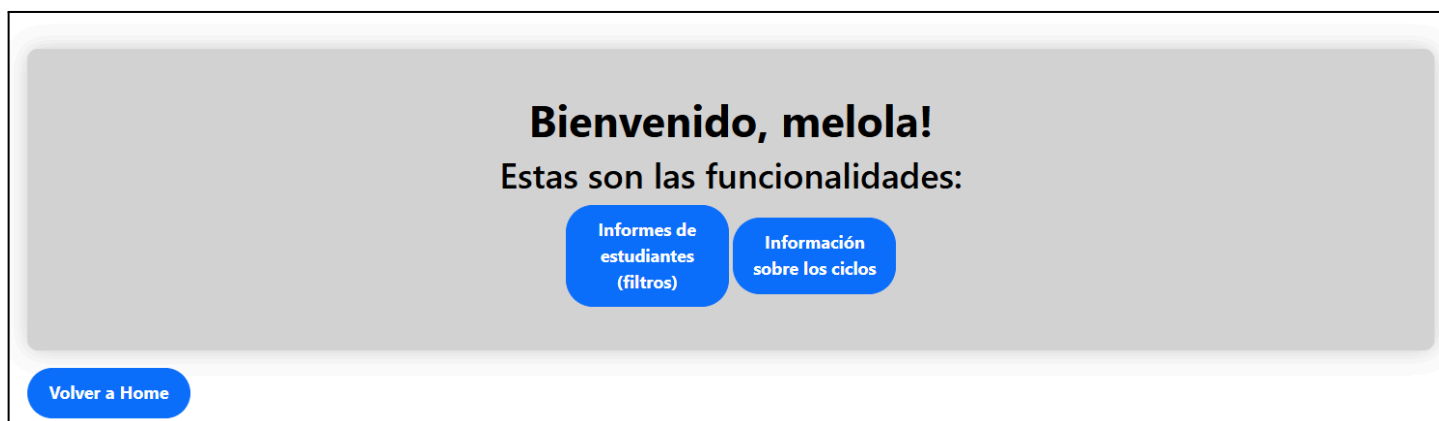
Iniciar Sesión

Usuario
melola

Contraseña

Ingresar

3. Si el inicio de sesión ha ido bien, aparecerá la página **principal:**



Bienvenido, melola!

Estas son las funcionalidades:

Informes de estudiantes (filtros)

Información sobre los ciclos

Volver a Home

Si no te funciona el registro de usuario!!!! recuerda que tienes el **usuario alumno** y el **usuario profe** ya creados para tus pruebas.

(*) **NAVEGACIÓN:** login → principal

INFORMACIÓN SOBRE LOS CICLOS


Cualquier usuario puede obtener información sobre los ciclos.

(*) **NAVEGACIÓN:** principal (elegir opción “Información sobre ciclos”) → ciclos-info → volver a principal

Ciclos de Informática en España - FP	
Consulta los ciclos formativos disponibles en Grado Medio y Grado Superior dentro del ámbito de la informática.	
Grado Medio	
Nombre del Ciclo	Duración
Sistemas Microinformáticos y Redes	2 años
Grado Superior	
Nombre del Ciclo	Duración
Administración de Sistemas Informáticos en Red (ASIR)	2 años
Desarrollo de Aplicaciones Multiplataforma (DAM)	2 años
Desarrollo de Aplicaciones Web (DAW)	2 años
Ciberseguridad	2 años
Big Data e Inteligencia Artificial	2 años
Volver a la página principal	

INFORMES DE ESTUDIANTES

- Solo un usuario con **ROLE_TEACHER** podrá ejecutar esa acción.
 - En caso contrario, saldrá una **página de error** (que se proporciona y no debes tocar) indicando que no tienes permisos:

 **Error**

Puede que no tengas permisos suficientes para ejecutar la aplicación.

Puede que sea otro tipo de error. Revisa la consola

[Volver a la página principal](#)

- Si el usuario tiene **ROLE_TEACHER** , aparecerá la página de **estudiantes/filtro**

Informes de Estudiantes

[Ver informe del estudiante con NIA_1002](#)

[Ver informe del estudiante Juan Pérez Gómez](#)

[Volver a la página principal](#)

Implementa el endpoint **/filtrar** encargado de toda la lógica para después mostrar el informe del estudiante filtrado. **Muestra por consola el estudiante** obtenido tras realizar la petición remota vía WebClient al API REST ESTUDIANTES proporcionado.

NO puedes usar directamente WebClient en el endpoint.

Completa la plantilla informe con la información del estudiante obtenido en el filtro:

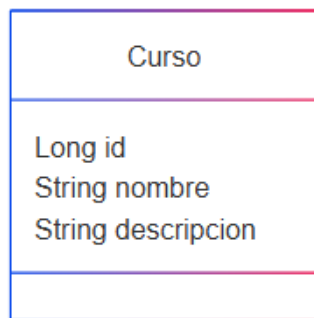
INFORME DEL ESTUDIANTE NIA_1002		INFORME DE JUAN PÉREZ GÓMEZ																																					
<h3>Detalles del Estudiante</h3> <table border="1"><tr><td>NIA</td><td>NIA_1002</td></tr><tr><td>Nombre</td><td>Juan</td></tr><tr><td>Primer Apellido</td><td>Pérez</td></tr><tr><td>Segundo Apellido</td><td>López</td></tr><tr><td>Email</td><td>juan.perez2@example.com</td></tr><tr><td>Móvil</td><td>601123456</td></tr><tr><td>Dirección</td><td>Avenida Gran Vía 23, Madrid</td></tr><tr><td>Fecha de Nacimiento</td><td>1999-08-22</td></tr><tr><td>Edad</td><td>25 años</td></tr></table> <p>Volver a los informes</p>		NIA	NIA_1002	Nombre	Juan	Primer Apellido	Pérez	Segundo Apellido	López	Email	juan.perez2@example.com	Móvil	601123456	Dirección	Avenida Gran Vía 23, Madrid	Fecha de Nacimiento	1999-08-22	Edad	25 años	<h3>Detalles del Estudiante</h3> <table border="1"><tr><td>NIA</td><td>NIA_1001</td></tr><tr><td>Nombre</td><td>Juan</td></tr><tr><td>Primer Apellido</td><td>Pérez</td></tr><tr><td>Segundo Apellido</td><td>Gómez</td></tr><tr><td>Email</td><td>juan.perez@example.com</td></tr><tr><td>Móvil</td><td>600123456</td></tr><tr><td>Dirección</td><td>Calle Mayor 10, Madrid</td></tr><tr><td>Fecha de Nacimiento</td><td>2000-05-15</td></tr><tr><td>Edad</td><td>24 años</td></tr></table> <p>Volver a los informes</p>		NIA	NIA_1001	Nombre	Juan	Primer Apellido	Pérez	Segundo Apellido	Gómez	Email	juan.perez@example.com	Móvil	600123456	Dirección	Calle Mayor 10, Madrid	Fecha de Nacimiento	2000-05-15	Edad	24 años
NIA	NIA_1002																																						
Nombre	Juan																																						
Primer Apellido	Pérez																																						
Segundo Apellido	López																																						
Email	juan.perez2@example.com																																						
Móvil	601123456																																						
Dirección	Avenida Gran Vía 23, Madrid																																						
Fecha de Nacimiento	1999-08-22																																						
Edad	25 años																																						
NIA	NIA_1001																																						
Nombre	Juan																																						
Primer Apellido	Pérez																																						
Segundo Apellido	Gómez																																						
Email	juan.perez@example.com																																						
Móvil	600123456																																						
Dirección	Calle Mayor 10, Madrid																																						
Fecha de Nacimiento	2000-05-15																																						
Edad	24 años																																						

Para obtener los informes, **debe realizarse la conexión al API Rest Data Estudiantes mediante WebClient síncrono y bloqueante:**

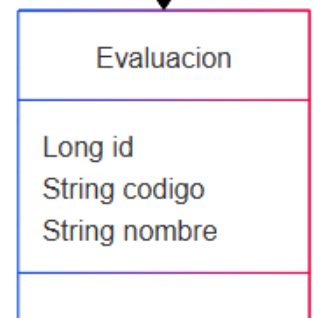
- El login y password estarán especificadas en un fichero llamado config.properties con los siguientes valores:
 - `api.estudiantes.auth.login=admin`
 - `api.estudiantes.auth.password=admin`
- Las urls de los endpoints para obtener los informes son (requieren autenticación):
 - http://localhost:8081/estudiantes/search/findByNia?nia=NIA_1002
 - <http://localhost:8081/estudiantes/search/findByNombreAndPrimerApellidoAndSegundoApellido?nombre=Juan&primerApellido=Pérez&segundoApellido=Gómez>
- Debe existir un **Servicio de autenticación** que se encargará de la autenticación (login) en el API Rest. Si la autenticación ha ido bien, obtendrá el token JWT, píntalo
 - Será necesario añadir el token en la petición de webclient de la siguiente manera:
`.header("Authorization", "Bearer " + token)`

API REST - EVALUACIONES

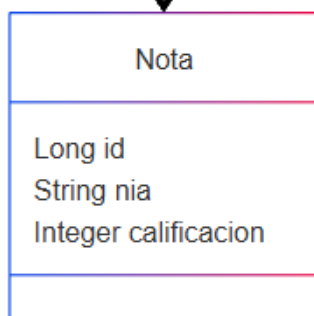
El **modelo de datos** que se proporciona con los entities es el siguiente:



1
tiene
*



1
contiene
*



- Un **curso** tiene muchas **evaluaciones** (1ev, 2ev...).
- Una **evaluación** contiene muchas **notas**.
- Cada **nota** pertenece a una única evaluación y está vinculada a un estudiante mediante su NIA.

SELECT id,nombre, descripcion FROM CURSO;

ID	NOMBRE	DESCRIPCION
1	1DAW	1º curso de Desarrollo de Aplicaciones Web
2	2DAW	2º curso de Desarrollo de Aplicaciones Web

select codigo,nombre, curso_id from evaluacion;

CODIGO	NOMBRE	CURSO_ID
1ev_1	Primera Evaluación	1
2ev_1	Segunda Evaluación	1
3ev_1	Tercera Evaluación	1
ordinaria_1	Ordinaria	1
extraordinaria_1	Extraordinaria	1
1ev_2	Primera Evaluación	2
2ev_2	Segunda Evaluación	2
ordinaria_2	Ordinaria	2
extraordinaria_2	Extraordinaria	2

Notas del estudiante por curso y evaluación:

CURSO	CODIGO_EVALUACION	CALIFICACION	NIA
1DAW	1ev_1	8	12345
1DAW	1ev_1	7	12346
1DAW	1ev_1	6	12347
1DAW	2ev_1	9	12348
1DAW	2ev_1	5	12345
1DAW	3ev_1	4	12346
1DAW	ordinaria_1	8	12348
1DAW	extraordinaria_1	7	12345
1DAW	extraordinaria_1	6	12346
1DAW	extraordinaria_1	5	12347
2DAW	1ev_2	7	12400
2DAW	1ev_2	9	12401
2DAW	1ev_2	6	12402
2DAW	2ev_2	8	12400
2DAW	2ev_2	7	12401
2DAW	2ev_2	5	12402

REQUISITOS OBLIGATORIOS DE TODOS LOS ENDPOINTS A DESARROLLAR:

- Usar de **DTO** para peticiones y respuestas que reciban y devuelvan JSON.
- Usar de **ResponseEntity**.
- **No está permitido usar JPQL (@Query)** en los repositorios.
- Usar **Optional** en vez de trabajar directamente con entidades.

Endpoint sin cumplir gran parte de lo especificado sobre la entrada/salida ⇒ **NO** se calificará.

<u>Obtener todos los cursos</u>	GET /cursos
<pre>GET http://localhost:8082/cursos Show Request HTTP/1.1 200 (Headers) ...Content-Type: application/json... [{ "nombre": "1DAW", "descripcion": "1º curso de Desarrollo de Aplicaciones Web" }, { "nombre": "2DAW", "descripcion": "2º curso de Desarrollo de Aplicaciones Web" }]</pre>	
<u>Obtener todas las evaluaciones con sus notas:</u> GET /evaluaciones/detalles	
<p>RESPUESTA: OK. Contenido en el body del request (no lo copio entero):</p> <pre>[{ "nombre": "1ev_1", "curso": "1DAW", "notas": [{ "nia": "12345", "calificacion": 8 }, { "nia": "12346", "calificacion": 7 }, { "nia": "12347", "calificacion": 6 }] }, { "nombre": "2ev_2", "curso": "2DAW", "notas": [{ "nia": "12400", "calificacion": 8 }, { "nia": "12401", "calificacion": 7 }, { "nia": "12402", "calificacion": 5 }] }, { "nombre": "ordinaria_2", "curso": "2DAW", "notas": [] }, { "nombre": "extraordinaria_2", "curso": "2DAW", "notas": [] }]</pre>	

Obtener el promedio de calificaciones por evaluación `GET /evaluaciones/{codigo}/promedio`

GET http://localhost:8082/evaluaciones/1ev_1/promedio

Show Request

HTTP/1.1 200

(Headers) ...Content-Type: application/json...

```
{
  "codigo": "1ev_1",
  "nombre": "Primera Evaluación",
  "curso": "1DAW",
  "promedio": 7.0
}
```

Código de evaluación incorrecto (excepción `EvaluacionNoEncontradaException`).

GET http://localhost:8082/evaluaciones/4ev_1/promedio

Show Request

HTTP/1.1 404

(Headers) ...Content-Type: application/json...

```
{
  "status": 404,
  "error": "Evaluación no encontrada",
  "mensaje": "No se encontró la evaluación con código 4ev_1",
  "timestamp": "2025-02-21T11:40:30.9546874"
}
```

Evaluación sin notas (excepción `EvaluacionSinNotasException`)

GET http://localhost:8082/evaluaciones/ordinaria_2/promedio

Show Request

HTTP/1.1 400 |

(Headers) ...Content-Type: application/json...

```
{
  "status": 400,
  "error": "Evaluación sin notas",
  "mensaje": "La evaluación ordinaria_2 del curso 2DAW no tiene notas registradas.",
  "timestamp": "2025-02-21T11:41:28.7803725"
}
```

Modificar la calificación de un estudiante en una evaluación

PATCH /evaluaciones/{codigo}/notas/{nia}

▶ PATCH http://localhost:8082/evaluaciones/1ev_1/notas/12345

Content-Type: application/json

```
{  
  "calificacion": 10  
}
```

Submit Request

Open in Editor

HTTP/1.1 200

> (Headers) ...Content-Type: application/json...

```
{  
  "evaluacionCodigo": "1ev_1",  
  "evaluacionNombre": "Primera Evaluación",  
  "cursoNombre": "1DAW",  
  "calificacion": 10  
}
```

La calificación a actualizar debe estar comprendida entre 1 y 10.

PATCH http://localhost:8082/evaluaciones/1ev_1/notas/12345

Content-Type: application/json

```
{  
  "calificacion": 0  
}
```

Submit Request

Open in Editor

> (Headers) ...Content-Type: application/json...

```
{  
  "status": 400,  
  "error": "Error de validación",  
  "mensaje": "La calificación debe ser al menos 1.",  
  "timestamp": "2025-02-21T12:06:22.5607429"  
}
```

PATCH http://localhost:8082/evaluaciones/1ev_1/notas/12345

Content-Type: application/json

```
{  
  "calificacion": 11  
}
```

Submit Request

Open in Editor

> (Headers) ...Content-Type: application/json...

```
{  
  "status": 400,  
  "error": "Error de validación",  
  "mensaje": "La calificación no puede ser mayor a 10.",  
  "timestamp": "2025-02-21T12:07:54.7326715"  
}
```

Excepción: EvaluacionNoEncontradaException

PATCH http://localhost:8082/evaluaciones/4ev_1/notas/12345

Content-Type: application/json

```
{
  "calificacion": 10
}
```

mit Request

Open in Editor

(Headers) ...Content-Type: application/json...

```
{
  "status": 404,
  "error": "Evaluación no encontrada",
  "mensaje": "4ev_1",
  "timestamp": "2025-02-21T12:08:47.4239558"
}
```

Excepción: NotaNoEncontradaException

PATCH http://localhost:8082/evaluaciones/1ev_1/notas/12340

Content-Type: application/json

```
{
  "calificacion": 10
}
```

mit Request

Open in Editor

(Headers) ...Content-Type: application/json...

```
{
  "status": 404,
  "error": "Nota no encontrada",
  "mensaje": "No se encontró la nota para el estudiante con NIA: 12340 en la evaluación: 1ev_1",
  "timestamp": "2025-02-21T12:09:46.6527848"
}
```