

Vas a participar en la **mejora y ampliación** de:

## EJERCICIO 1

El proyecto **DAW EVENT HUB** compuesto de dos aplicaciones:

- **EventHubDataRest (API Data REST)** — catálogo para consultar
- **EventHubMVC** — seguridad + consumo Data REST + carrito + lógica “de interfaz”

## EJERCICIO 2

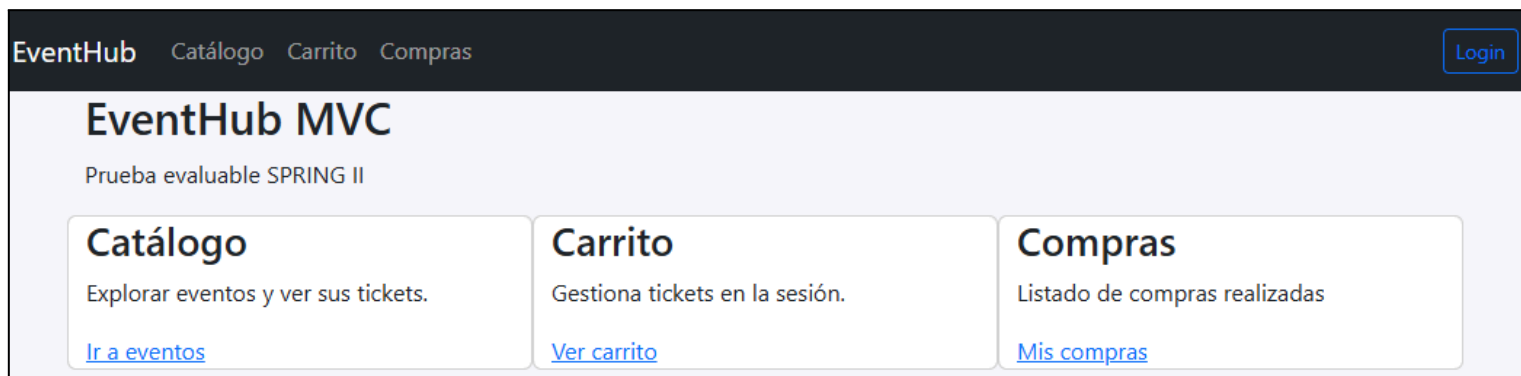
El proyecto **Clínica API REST** — seguridad JWT + lógica + agregados

### EJERCICIO 1- API DATA REST

- Catálogo público y de **solo lectura**. No debes modificarlo
- El modelo del catálogo está compuesto de **tres entidades** (Location, Event y TicketType)
- Para probar el catálogo, se entrega un fichero **endpoints.http** con ejemplos de llamadas que puedes ejecutar directamente.

### EJERCICIO 1- MVC

**EventHubMVC** es una aplicación Spring MVC de compra de entradas a eventos.



- Con BD **H2 en disco** y consume de **EventHubDataRest** para:
  - Listar eventos
  - Filtrar eventos por ubicación
  - Obtener el detalle de un evento
  - Obtener los tipos de ticket asociados a un evento

### 1. SECURIZAR LA APLICACIÓN

Debes securizar la aplicación para que **cualquier página de la aplicación requiera autenticación** (excepto la página de login y recursos estáticos necesarios).

Modifica el proceso de autenticación para que **un usuario pueda iniciar sesión utilizando su nombre de usuario o su email** (ambos deben funcionar). Debe ser posible iniciar sesión tanto con el Username como con el Email de los usuarios de prueba.

Login

Username or email

You can sign in using your username or your email address.

Password

Sign in

Test users

user@demo / user123

admin@local / admin123

Modifica el formulario de autenticación para añadir esta funcionalidad acorde a la captura. Los estilos no tienen por qué ser exactamente iguales...

Usuarios de prueba:

Username: user@demo · Email: user@demo.local · Pwd: user123

Username: admin@local · Email: admin@demo.local · Pwd: admin123

## 2. MEJORA DE LA FUNCIÓN DE COMPRAR

Actualmente, al añadir entradas al carrito, el usuario no recibe ninguna confirmación visual. Implementa la funcionalidad necesaria para que, tras añadir un tipo de entrada, se muestre en la página de detalle del evento un **mensaje de éxito** indicando **la cantidad de entradas añadidas** (según el valor introducido en el formulario) "Added X ticket(s) to your cart.". Observa la captura:

DevOps Meetup 2025

Evento pasado para probar filtros por fecha.

Code: EVT\_MAD\_TECH\_2025

TECH

Inicio: 10/10/2025 18:00

Added 1 ticket(s) to your cart.

Tipo de entradas (Ticket types)

3 tipos disponibles

Tipo	Precio	Cantidad disponible	Cantidad a comprar	
GENERAL	10.00 €	300	1	Añadir
VIP	25.00 €	30	1	Añadir
STUDENT	5.00 €	80	1	Añadir

Ver carrito

Al recargar la página el mensaje debe de desaparecer.

## 3. CARRITO. Validación server-side real de compra

Actualmente, la aplicación limita la cantidad de entradas, según la categoría (10, 4 ó 2), que se pueden añadir mediante controles en la interfaz.

Tipo de entradas (Ticket types) 3 tipos disponibles

Tipo	Precio	Cantidad disponible	Cantidad a comprar
GENERAL	10.00 €	300	20 <input type="button" value="Añadir"/>
VIP	25.00 €	<div> <span>!</span> El valor debe ser inferior o igual a 10         </div>	
STUDENT	5.00 €	80	1 <input type="button" value="Añadir"/>

Este enfoque **NO es suficiente en un entorno real**, ya que el servidor **no puede confiar en los datos enviados por el cliente** y el estado del carrito podría quedar corrupto si se manipula el formulario o se envía la petición manualmente.

**Adapta la aplicación para garantizar que, al añadir entradas al carrito:**

- **No se pueda superar el número máximo permitido** por la categoría del tipo de entrada.
- Si el tipo de entrada ya existe en el carrito, la **cantidad total resultante tampoco supere dicho límite**.
- **En caso de intento no válido:**
  - El carrito **no debe modificarse**.
  - Debe mostrarse un **mensaje informativo** en la página de detalle del evento *"You can buy at most 4 ticket(s) for VIP. You already have 3 in your cart."*
  - El mensaje debe mostrarse una sola vez y desaparecer al recargar la página.

**CASO DE PRUEBA:** si el máximo de entradas tipo VIP es 4 y el carrito ya contiene 3 entradas de ese tipo, al intentar añadir 2 más debe rechazarse ( aunque en la página html se permita introducir 2)

**1.Compro tres entradas cuando se permiten 4 entradas para el tipo VIP. Todo OK**

**Carrito**

Items 1 línea(s)

Ticket	Tipo	Precio	Cantidad	Total
TT_EVT_MAD_TECH_2025_VIP	VIP	25.00 €	3 <input type="button" value="Actualizar"/>	75.00 € <input type="button" value="Eliminar"/>

0 elimina la línea

Consejo: revisa cantidades antes de confirmar. [Añadir más tickets](#)

**2.Compro dos entradas más de tipo VIP. No da error de validación html porque te deja porque el límite max está a 4:**

Tipo	Precio	Cantidad disponible	Cantidad a comprar
GENERAL	10.00 €	300	1 <input type="button" value="Añadir"/>
VIP	25.00 €	30	2 <input type="button" value="Añadir"/>
STUDENT	5.00 €	80	1 <input type="button" value="Añadir"/>

**3.Al añadir, se comprueba que has sobrepasado el límite y se muestra el mensaje de aviso:** *You can buy at most 4 ticket(s) for VIP. You already have 3 in your cart.*

DevOps Meetup 2025

Volver al listado

Evento pasado para probar filtros por fecha.

Code: EVT\_MAD\_TECH\_2025

TECH

Inicio: 10/10/2025 18:00

You can buy at most 4 ticket(s) for VIP. You already have 3 in your cart.

Tipo de entradas (Ticket types)

3 tipos disponibles

Tipo	Precio	Cantidad disponible	Cantidad a comprar	
GENERAL	10.00 €	300	1	Añadir
VIP	25.00 €	30	1	Añadir
STUDENT	5.00 €	80	1	Añadir

Ver carrito

## 4. CARRITO. Descuento por completar entradas VIP

Las entradas de categoría **VIP** tienen un límite máximo de **4 unidades por compra**.

Para incentivar que el usuario complete dicho máximo, se debe aplicar una regla de descuento con las siguientes condiciones:

- Si **antes de añadir nuevas entradas**, el carrito ya contiene **3 entradas VIP**, las nuevas entradas VIP añadidas se benefician de un **10 % de descuento**.
- El descuento se aplica **únicamente a las entradas añadidas en esa operación**, no a las que ya estaban en el carrito.
- Si el carrito **no cumplía previamente** la condición (por ejemplo, estaba vacío), **no se aplica descuento**, aunque se añadan 4 entradas VIP de una sola vez.

**Ejemplo:**

- Si el carrito contiene 3 entradas VIP y se añade 1 más, esa nueva entrada VIP se añade con un 10 % de descuento.
- Si el carrito está vacío y se añaden 4 entradas VIP de una sola vez, **no se aplica descuento**.

**En el carrito debe mostrarse una línea diferenciada** para las entradas con descuento (aunque correspondan al mismo tipo de entrada) y, tras la operación, debe **mostrarse un mensaje informativo en el detalle del evento** indicando que **se ha aplicado el descuento** (ver capturas).

Carrito

Items

2 línea(s)

Ticket	Tipo	Precio	Cantidad	Total
TT_EVT_MAD_TECH_2025_VIP	VIP	25.00 €	<div>3</div> <div>0 elimina la línea</div>	<div>Actualizar</div> <div>75.00 €</div> <div>Eliminar</div>
TT_EVT_MAD_TECH_2025_VIP	VIP	22.500 €	<div>1</div> <div>0 elimina la línea</div>	<div>Actualizar</div> <div>22.500 €</div> <div>Eliminar</div>

Consejo: revisa cantidades antes de confirmar.

[Añadir más tickets](#)

10% discount applied to the new VIP tickets.

Tipo de entradas (Ticket types)

3 tipos disponibles

Tipo	Precio	Cantidad disponible	Cantidad a comprar
GENERAL	10.00 €	300	<div>1</div> <div>Añadir</div>
VIP	25.00 €	30	<div>1</div> <div>Añadir</div>
STUDENT	5.00 €	80	<div>1</div> <div>Añadir</div>

Ver carrito

5. EDICIÓN DE PERFIL

Implementa una funcionalidad “**Editar usuario**” que permita a un usuario autenticado modificar los datos de su perfil desde la aplicación web. **Se editarán únicamente fullName, email y password.**

La solución debe estar integrada en el flujo MVC existente y cumplir los siguientes criterios:

<div>EventHub Catálogo Carrito Compras Editar usuario</div>	<div>Debe añadirse un acceso a esta funcionalidad desde la interfaz cuando el usuario esté autenticado (en el menú).</div> <div>Los datos actuales del usuario deben mostrarse en el formulario de edición. <b>Utiliza y amplía profile/edit.html</b></div>
---	---

## Editar perfil

Nombre completo

Email

### Cambiar contraseña (opcional)

Nueva contraseña

Si rellenas este campo, deberás confirmar la contraseña.

Confirmar contraseña

## Editar perfil

Perfil actualizado correctamente

Nombre completo

Email

### Cambiar contraseña (opcional)

Nueva contraseña

Si rellenas este campo, deberás confirmar la contraseña.

Confirmar contraseña

**La contraseña solo debe modificarse si el usuario decide cambiarla.**

**Al cargarse el formulario de edición aparecerá solo el nombre completo y el email, las contraseñas no aparecen visibles. Observa la captura.**

**Tras una actualización correcta, el usuario debe ser redirigido al formulario indicando que se ha actualizado correctamente el perfil.**

## Editar perfil

Nombre completo

User Demo muy largo asdfasdfsdfasdfsdf asdf asdf asdfasdf asdfasdfsdf asdfa

el tamaño debe estar entre 0 y 100

Email

user@demo.local

debe ser una dirección de correo electrónico con formato correcto

el tamaño debe estar entre 0 y 50

### Cambiar contraseña (opcional)

Nueva contraseña

Deja en blanco para no cambiarla

Si rellenas este campo, deberás confirmar la contraseña.

Confirmar contraseña

Repite la nueva contraseña

Las contraseñas no coinciden

Guardar cambios

Cancelar

### VALIDACIONES:

El **nombre completo** y el **email** deben **validarse** correctamente, acorde a las restricciones del modelo.

**Implementa todas las validaciones que puedes ver en la captura de pantalla.**

El **email no puede coincidir** con el de otro usuario del sistema.

Los **errores de validación** deben mostrarse en el formulario acorde a las capturas de pantalla.

## Editar perfil

Nombre completo

User Demo update

Email

admin@demo.local

El email ya está en uso por otro usuario

## EJERCICIO 2 - CLÍNICA API REST

### MODELO COMPLETO DEL DOMINIO (Con BD H2 en disco)

Tabla	Descripción	Relaciones principales
patients	Pacientes registrados en el sistema	1 Patient → 0..N Appointments
doctors	Médicos del sistema	1 Doctor → 0..N Appointments 1 Doctor → 0..N DoctorSpecialties
medical_services	Catálogo de servicios médicos facturables	1 MedicalService → 0..N InvoiceLines
appointments	Citas médicas entre paciente y doctor	N Appointments → 1 Patient N Appointments → 1 Doctor 1 Appointment → 0..1 Invoice
doctor_specialties	Especialidades de un médico (entidad intermedia)	N DoctorSpecialties → 1 Doctor
invoices	Facturas asociadas a una cita	1 Invoice → 1 Appointment 1 Invoice → 1..N InvoiceLines
invoice_lines	Líneas de detalle de una factura	N InvoiceLines → 1 Invoice N InvoiceLines → 1 MedicalService
app_users	Usuarios de la aplicación	N Users ↔ N Roles
roles	Roles del sistema (ADMIN, STAFF, BILLING)	N Roles ↔ N Users

Se proporciona el proyecto con una estructura inicial que debes de modificar y ampliar en base a las siguientes especificaciones. **PARTICIPAS EN EL DESARROLLO DE UN PROYECTO INCOMPLETO Y CON BUGS.**

Todos los endpoints son **públicos**. Si tenéis problemas para configurar la **SEGURIDAD**, dejadlos públicos.

**USUARIOS DE PRUEBA** (login y pwd iguales): `admin`, `staff`, `billing`

### 1. ENTIDADES JPA

Debes **diseñar correctamente las relaciones bidireccionales JPA** junto con los métodos java correspondientes, entre **Doctor** y **DoctorSpecialty**. **CORRIGE** los errores JPA. Haz la **VALIDACIÓN**.

### 2. EMITIR FACTURA DE UNA CITA (crear Invoice + líneas)



<pre>POST http://localhost:8083/api/appointments/56/invoice Content-Type: application/json Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJyb2xlcyI6I6  {   "lines": [     {       "medicalServiceId": 6,       "qty": 1     },     {       "medicalServiceId": 5,       "qty": 2     }   ] }</pre>	<p><b>POST</b> <code>http://localhost:8083/api/appointments/{id}/invoice</code></p> <p>Crear factura a partir de una cita y sus servicios médicos. La creación de la factura y sus líneas debe ser atómica (Prueba con la cita con id 56)</p> <p><b>PARTE DE InvoiceController Y VE COMPLETANDO LOS FUENTES PERTINENTES. <u>Completa, no machaques.</u></b></p> <p><b>La factura debe persistirse junto con todas sus líneas en la base de datos en la misma operación.</b> Mantener bidireccionalidad.</p> <p>El usuario solo con <b>rol ADMIN o BILLING</b>.</p>
<pre>HTTP/1.1 201 (Headers) ...Content-Type: application/json...  {   "id": 30,   "appointmentId": 56,   "status": "PENDING",   "issuedAt": "2026-02-02T13:55:12.2412359",   "subtotal": 110.00,   "taxTotal": 23.1000,   "total": 133.1000,   "lines": [     {       "id": 33,       "medicalServiceId": 6,       "medicalServiceName": "Plan nutrición",       "qty": 1,       "unitPrice": 60.00,       "vatRate": 0.21,       "lineTotal": 72.6000     },     {       "id": 34,       "medicalServiceId": 5,       "medicalServiceName": "Vacuna gripe",       "qty": 2,       "unitPrice": 25.00,       "vatRate": 0.21,       "lineTotal": 60.5000     }   ] }</pre>	<p>No puede haber servicios duplicados en el json request; si no → <b>BadRequestException (400)</b></p> <p>La cita debe existir; si no → <b>NotFoundException (404)</b>.</p> <p>La cita no puede estar ya facturada; si no → <b>ConflictException (409)</b>.</p> <p>La cita no puede estar <b>CANCELLED</b>. si no → <b>BussinessRuleException (422)</b>.</p> <p>La cita debe estar <b>COMPLETED</b> si no → <b>BussinessRuleException (422)</b>.</p> <p>Cada <b>medicalServiceId</b> debe existir y estar <b>active=true</b>; si no: inexistente → 404 / inactivo → 422 (el primero que falle)</p> <p>Las líneas de factura se emitirán sin aplicar ningún tipo de descuento.</p> <p>El tipo de IVA aplicado en las líneas de factura será siempre del 21% (VAT_21). El cálculo de impuestos y totales debe realizarse en el servidor. Los importes económicos deberán redondearse a dos decimales siguiendo el redondeo estándar.</p>
<p><b>EJEMPLOS DE RESPUESTAS CON ERROR (Obligatorio completar y usar <code>ApiErrorResponse</code>)</b></p>	

<div>POST http://localhost:8083/api/appointments/6/invoice</div> <div>Show Request</div> <div>HTTP/1.1 409</div> <div>&gt; (Headers) ...Content-Type: application/json...</div> <div>{   "message": "Invoice already exists for appointment: }</div>	<div>HTTP/1.1 400</div> <div>&gt; (Headers) ...Content-Type: application/json...</div> <div>{   "message": "Duplicate medicalServiceId in lines: 5" }</div>
<div>POST http://localhost:8083/api/appointments/2/invoice</div> <div>Show Request</div> <div>HTTP/1.1 422</div> <div>&gt; (Headers) ...Content-Type: application/json...</div> <div>{   "message": "Cannot issue invoice for a CANCELLED appointm }</div>	<div>POST http://localhost:8083/api/appointments/3/invoice</div> <div>Show Request</div> <div>HTTP/1.1 422</div> <div>&gt; (Headers) ...Content-Type: application/json...</div> <div>{   "message": "Appointment must be COMPLETED to issue an invoice." }</div>

3. INFORME DE SERVICIOS MÁS FACTURADOS

<div>GET http://localhost:8083/api/reports/top-services? from=2026-01-01T12:00:00&amp; to=2026-12-31T00:00:00&amp;status=PAID&amp;size=2 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJyb2xl</div> <div>Submit Request Open in Editor</div> <div>[   {     "serviceId": 3,     "serviceName": "ECG / Cardiología",     "linesCount": 3,     "unitsSold": 3,     "totalBilled": 435.60   },   {     "serviceId": 1,     "serviceName": "Consulta general",     "linesCount": 6,     "unitsSold": 6,     "totalBilled": 363.00   } ]</div>	<div>Endpoint de reporting que devuelve un <b>ranking PAGINADO de los servicios médicos más facturados</b> en un intervalo de tiempo, filtrando por la <b>fecha de emisión de la factura</b>.</div> <div>COMPLETA LAS CLASES QUE SE PROPORCIONAN:</div> <div><ul style="list-style-type: none"><li>- ReportController</li><li>- ReportService</li><li>- TopServiceReport (mantén campos acorde captura)</li><li>- InvoiceLineRepository</li></ul></div> <div>La solución debe usar JPQL con proyección a DTO. (joins + group by + agregaciones)</div> <div>El filtro se hace por factura, no por cita (Invoice.issuedAt)</div> <div>GET /api/reports/top-services</div> <div><ul style="list-style-type: none"><li>• from (obligatorio) → fecha/hora mínima (incluida).</li><li>• to (obligatorio) → fecha/hora máxima (incluida).</li><li>• status (opcional) → si se informa, solo se consideran facturas con ese estado.</li></ul></div> <div>Solo acceso usuario con rol BILLING.</div> <div>El ranking se calcula solo con facturas emitidas (issuedAt != null) en el rango de fechas.</div>
--	---

<pre>GET http://localhost:8083/api/reports/top-services? from=2026-01-01T12:00:00&amp; to=2026-12-31T00:00:00&amp;status=PAID&amp;size=10 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJyb2xlcyI6WyJm mit Request Open in Editor Show Request HTTP/1.1 400 (Headers) ...Content-Type: application/json... {   "message": "El tamaño de página debe estar entre 1 y 5" }</pre>	<p>Los servicios más facturados deben aparecer primero (importe total facturado).</p> <p>Como resultado debe devolverse una lista del DTO <b>TopServiceReport</b>, con estos campos:</p> <ul style="list-style-type: none"><li>• <b>serviceId</b>: id del servicio</li><li>• <b>serviceName</b>: nombre del servicio</li><li>• <b>linesCount</b>: número de líneas de factura asociadas al servicio</li><li>• <b>unitsSold</b>: unidades totales vendidas del servicio</li><li>• <b>totalBilled</b>: importe total facturado del servicio</li></ul> <p>El número máximo de resultados lo marca el tamaño de la paginación: por defecto 3, mínimo 1, máximo 5</p>
<pre>GET http://localhost:8083/api/reports/top-services? from=2026-01-01T12:00:00&amp; to=2026-12-31T00:00:00 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJyb2xlcyI6WyJm</pre>	<pre>HTTP/1.1 200 (Headers) ...Content-Type: application/json...  [   {     "serviceId": 1,     "serviceName": "Consulta general",     "linesCount": 12,     "unitsSold": 12,     "totalBilled": 726.00   },   {     "serviceId": 3,     "serviceName": "ECG / Cardiología",     "linesCount": 4,     "unitsSold": 4,     "totalBilled": 580.80   },   {     "serviceId": 2,     "serviceName": "Revisión dermatológica",     "linesCount": 5,     "unitsSold": 5,     "totalBilled": 484.00   } ]</pre>