

# Informe del Proyecto Integrador - Microservicio de Reuniones Zoom

---

## Links de los repositorios:

Estructura del microservicio: <https://github.com/TaniaPerezD/microservicio-reuniones.git>

Consumo con el endpoint del proyecto integrador:

[https://github.com/TaniaPerezD/TaniaPerez\\_2daEvaluacionBackend.git](https://github.com/TaniaPerezD/TaniaPerez_2daEvaluacionBackend.git)

## 1. Análisis del Proyecto Integrador (15 pts)

### 1.1 Identificación clara del endpoint (5 pts)

El endpoint consumido es:

POST /api/collections/usuarios/auth-with-password

Ya que este nos proporciona el token JWT, además de manera interna, con la extensión de pocketbase con Go, se consultan las colecciones de “usuarios” y “reuniones”

El endpoint desarrollado es:

POST /api/custom/create-meeting

Este endpoint está implementado dentro de PocketBase como un endpoint personalizado protegido por autenticación. Su objetivo principal es permitir que un director de carrera genere reuniones de Zoom a través de un microservicio externo, una vez que se haya solicitado una reunión por parte del estudiante.

### 1.2 Justificación de la elección del endpoint (5 pts)

Se escogió el endpoint /api/collections/usuarios/auth-with-password porque permite acceder a la información de los usuarios después de autenticarlos, esto es clave, ya que solo los usuarios que tengan el rol de “director” pueden manejar la información de las reuniones, por otro lado, se escogieron los endpoints internos de “usuarios” y “reuniones” para completar los datos necesarios, como el id del estudiante que pide la reunión o guardar el enlace generado por el microservicio.

Además, el endpoint a desarrollar fue elegido porque resuelve una necesidad clave en la gestión académica: la coordinación eficiente de reuniones entre estudiantes y directores de carrera. Automatizar la generación de reuniones Zoom facilita la organización y manejo del tiempo del director de carrera.

### 1.3 Descripción técnica del endpoint (5 pts)

- **Método:** POST
- **Ruta:** /api/custom/create-meeting
- **Protección:** Requiere autenticación con rol de usuario director.
- **Formato de entrada:** JSON, con datos como agenda, duration, start\_time, timezone, e información del estudiante invitado.
- **Formato de salida:** JSON, con el ID del registro en PocketBase y los datos devueltos por el microservicio Zoom (por ejemplo: join\_url, start\_url).
- **Proceso:**
  - Valida el rol del usuario autenticado.
  - Lee el payload del request.
  - Envía la solicitud al microservicio de reuniones.
  - Guarda la respuesta como registro en PocketBase.
  - Devuelve una respuesta combinada al cliente.

## 2. Diseño del Microservicio (15 pts)

### 2.1 Objetivo del microservicio claramente definido (5 pts)

El objetivo principal del microservicio es conectarse con la API oficial de Zoom para generar reuniones virtuales automáticamente, en nombre del director de carrera. Además de crearlas, el microservicio está diseñado para permitir acciones futuras como editar, eliminar o listar reuniones activas, garantizando un control completo sobre la gestión de las sesiones.

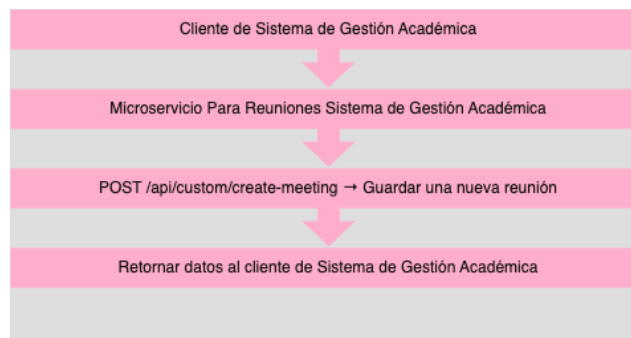
### 2.2 Elección y justificación de la tecnología de comunicación (5 pts)

Se utilizó comunicación HTTP basada en REST, empleando el formato JSON. Esta elección se debe a:

- Facilidad de integración con otras APIs como la de Zoom.

- Compatibilidad nativa con PocketBase.
- Bajo acoplamiento y escalabilidad para futuras integraciones.
- Acceso y manipulación sencilla desde herramientas de prueba como Postman o curl.

## 2.3 Diagrama del flujo de integración (5 pts)

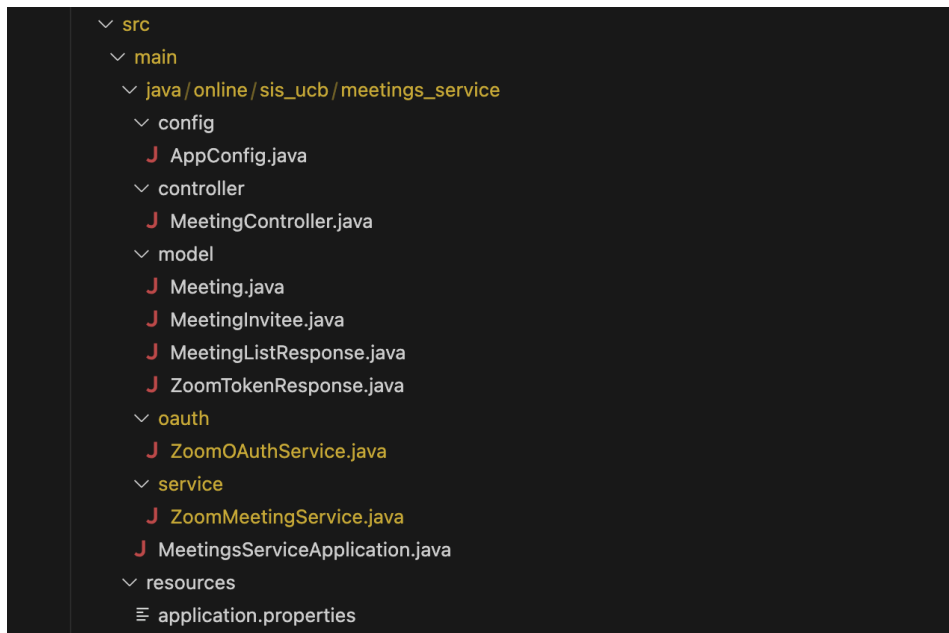


### Flujo resumido:

1. El estudiante solicita una reunión desde la plataforma.
2. El director accede y aprueba la reunión.
3. Se hace una llamada POST /api/custom/create-meeting.
4. Este endpoint llama al microservicio.
5. El microservicio se comunica con la API de Zoom para crear la reunión.
6. Se recibe la respuesta de Zoom (URL, hora, etc.).
7. Se guarda en PocketBase.
8. Se responde al frontend con la confirmación.

## 3. Implementación Técnica (40 pts)

Estructura del proyecto



### 3.1 Desarrollo del microservicio funcional (15 pts)

El microservicio fue desarrollado en Spring boot siguiendo principios de modularidad y claridad. Se expone un endpoint como:

POST /api/meetings/create/:email

Este endpoint recibe la configuración de la reunión, se autentica contra la API de Zoom (vía JWT u OAuth), y crea la reunión en nombre del director autenticado.

### 3.2 Consumo correcto del endpoint externo (10 pts)

La API oficial de Zoom fue consumida exitosamente usando HTTP con autenticación basada en JWT. La integración incluye:

- Generación correcta del token JWT.
- Configuración del payload según el formato requerido por Zoom.
- Manejo de errores y respuestas de estado.
- Extracción de los datos útiles (start\_url, join\_url, id, etc.).

### 3.3 Procesamiento y transformación de los datos (10 pts)

Los datos recibidos del frontend (hora local, duración, estudiante) son transformados en el formato aceptado por Zoom, lo que incluye:

- Conversión de hora local (America/La\_Paz) a UTC.
- Serialización completa del JSON antes de enviarlo a Zoom.

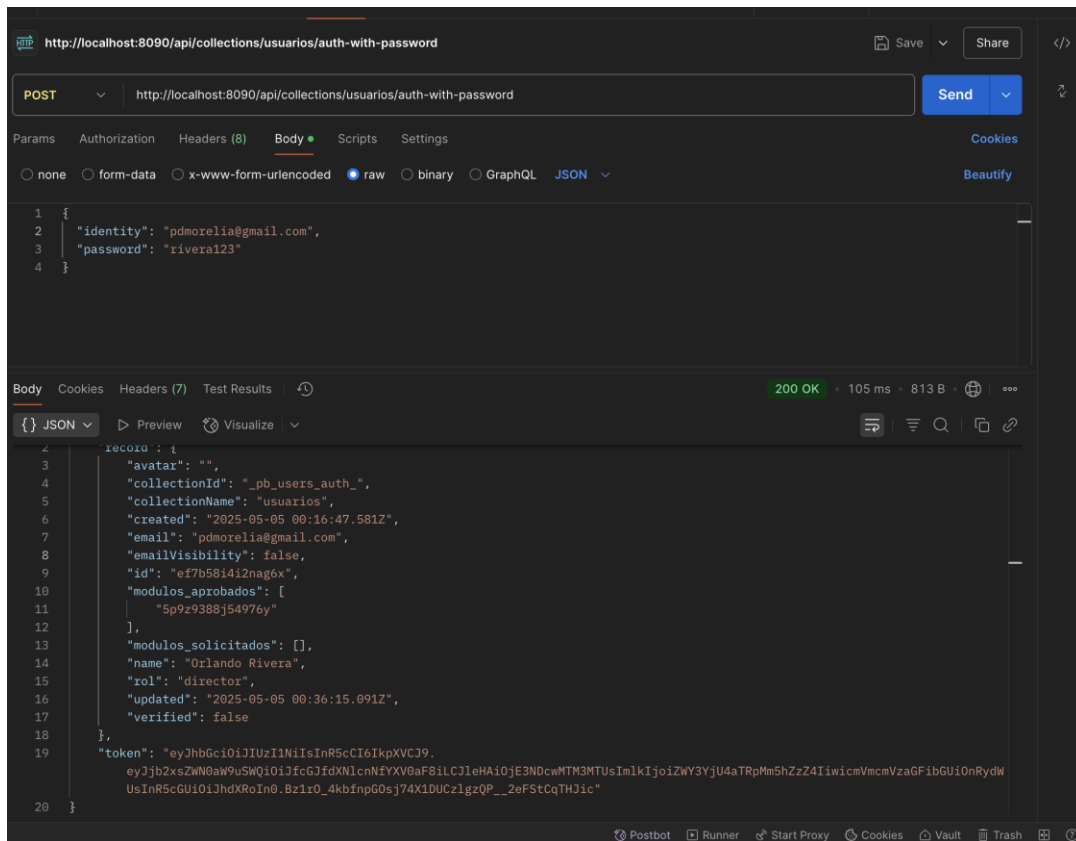
## 4. Pruebas y Documentación (15 pts)

### 4.1 Evidencias funcionales (5 pts)

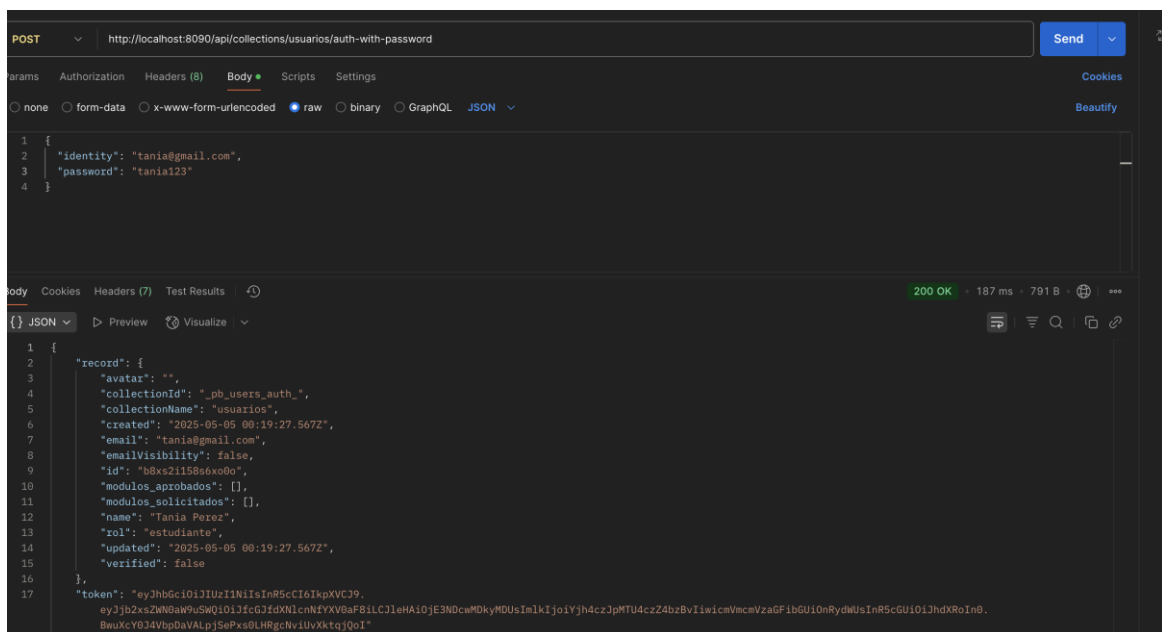
Se incluyen capturas de:

- Solicitudes hechas desde Postman.
- Reuniones creadas en la cuenta Zoom.
- Registros guardados en PocketBase.

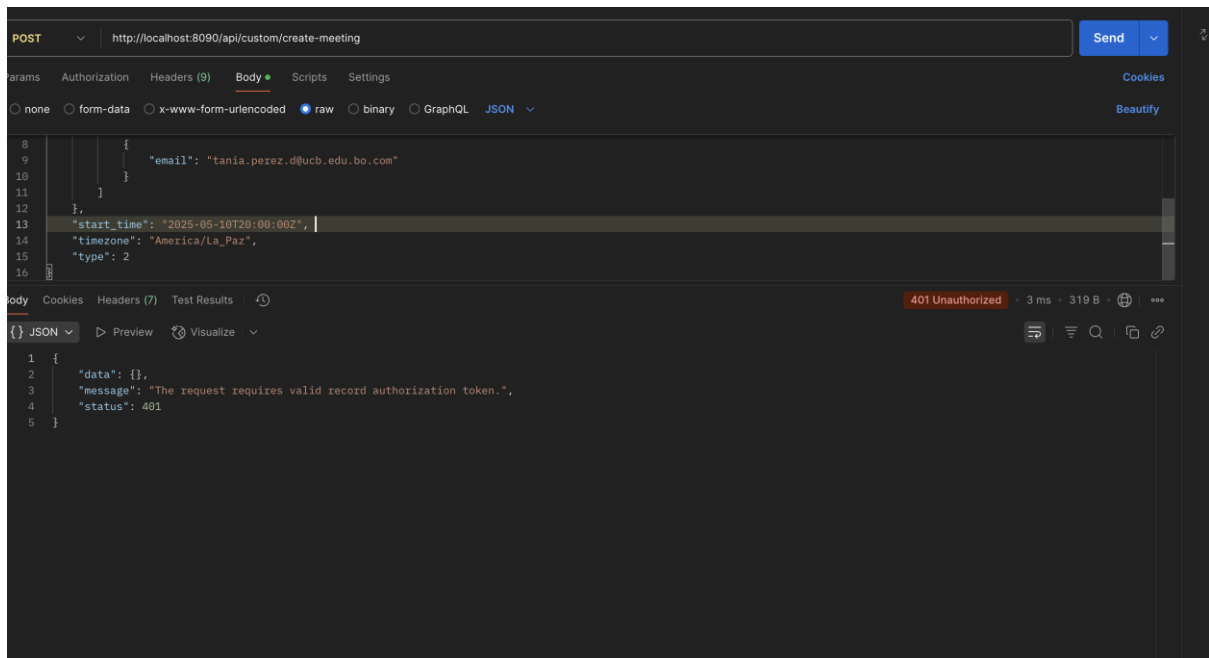
## Inicio de sesión como director



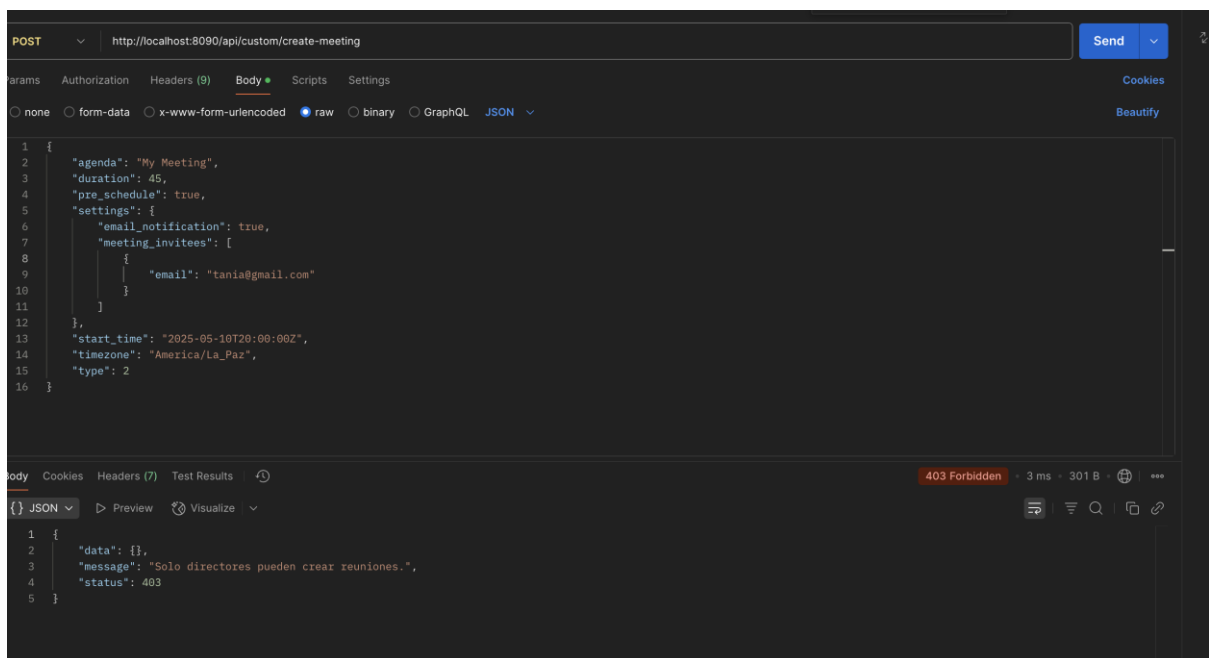
## Inicio de sesión como estudiante



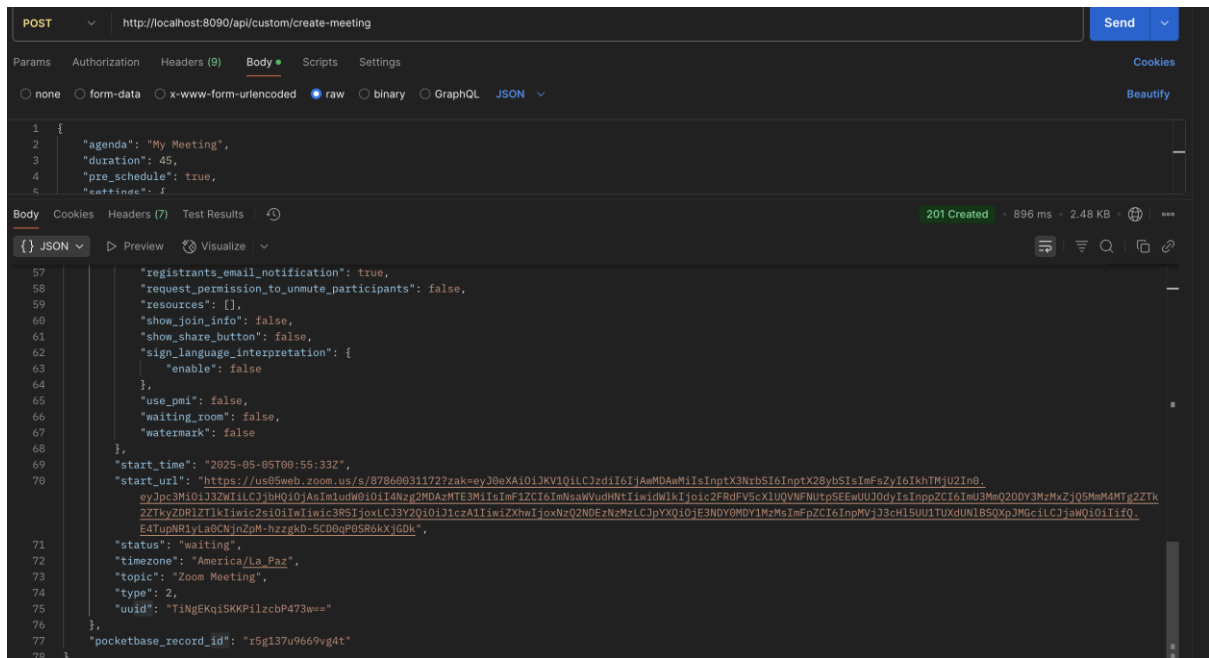
## Prueba del endpoint al no contar con un token JWT



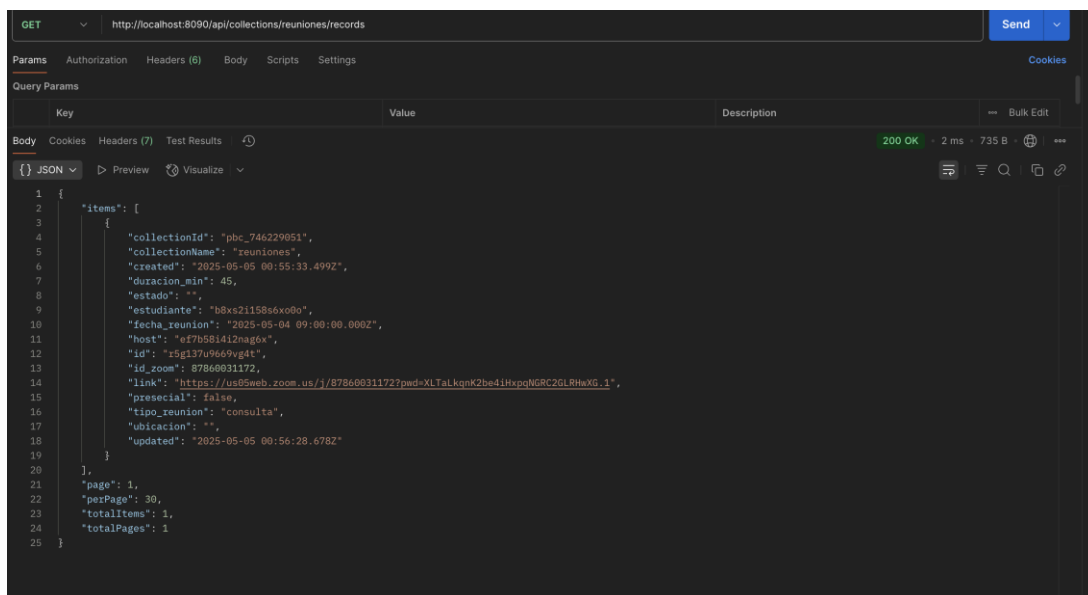
Prueba del endpoint con un token JWT, pero con rol de estudiante



Prueba del endpoint con un token JWT y rol de director de carrera



Verificación de la creación de la reunión en la colección de pocketbase y en la cuenta de Zoom





Edit reuniones record

id
r5g137u9669vg4t

estudiante
Tania Perez
Open picker

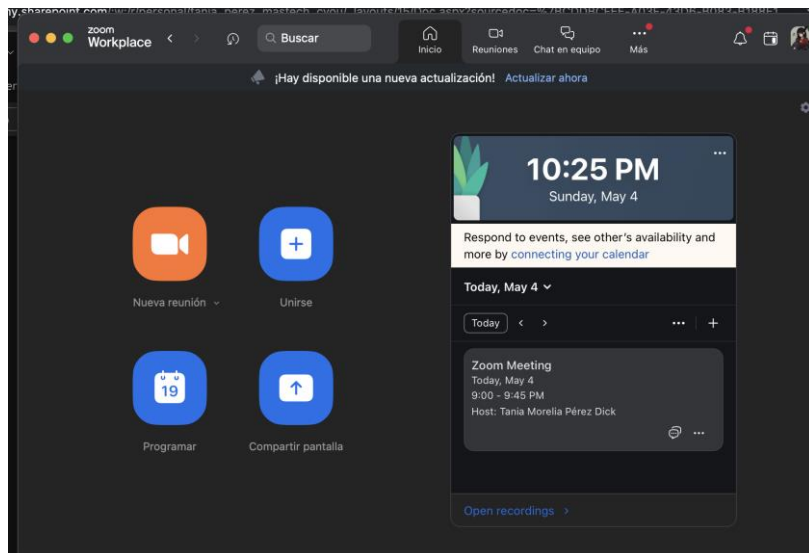
host
Orlando Rivera
Open picker

tipo\_reunion
consulta

fecha\_reunion
2025-05-04 09:00:00

link
https://us05web.zoom.us/j/87860031172?pwd=XLtaLkqnK2be4iHxpqNGRC2GLRHwXG.1

Cancel
Save changes



## 4.2 Pruebas unitarias o manuales explicadas (5 pts)

Las pruebas fueron manuales y consistieron en:

- Validación del login y autorización del director.
- Envío de requests válidos y con errores.
- Verificación de reuniones en Zoom.
- Comprobación de persistencia en PocketBase.