



# PaTareas

Pablo Peiró Serra

IES Benigasló

Ciclo DAM

Curso 2024-2025

Tutor: Gerard Falcó



---

# ¿QUÉ ES PaTareas?

- Aplicación móvil que permite crear, organizar y visualizar tareas personales o académicas, con notificaciones y estadísticas.



# PaTareas

---

# RESUMEN DEL PROYECTO



App Android desarrollada con Java y arquitectura MVVM.



Backend con Spring Boot y PostgreSQL.



Comunicación mediante API REST con autenticación.

---

# OBJETIVOS DEL PROYECTO

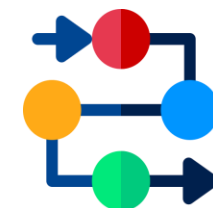
Interfaz intuitiva



Motivar al usuario



Organización



---

# ESTUDIO DE SOLUCIONES EXISTENTES

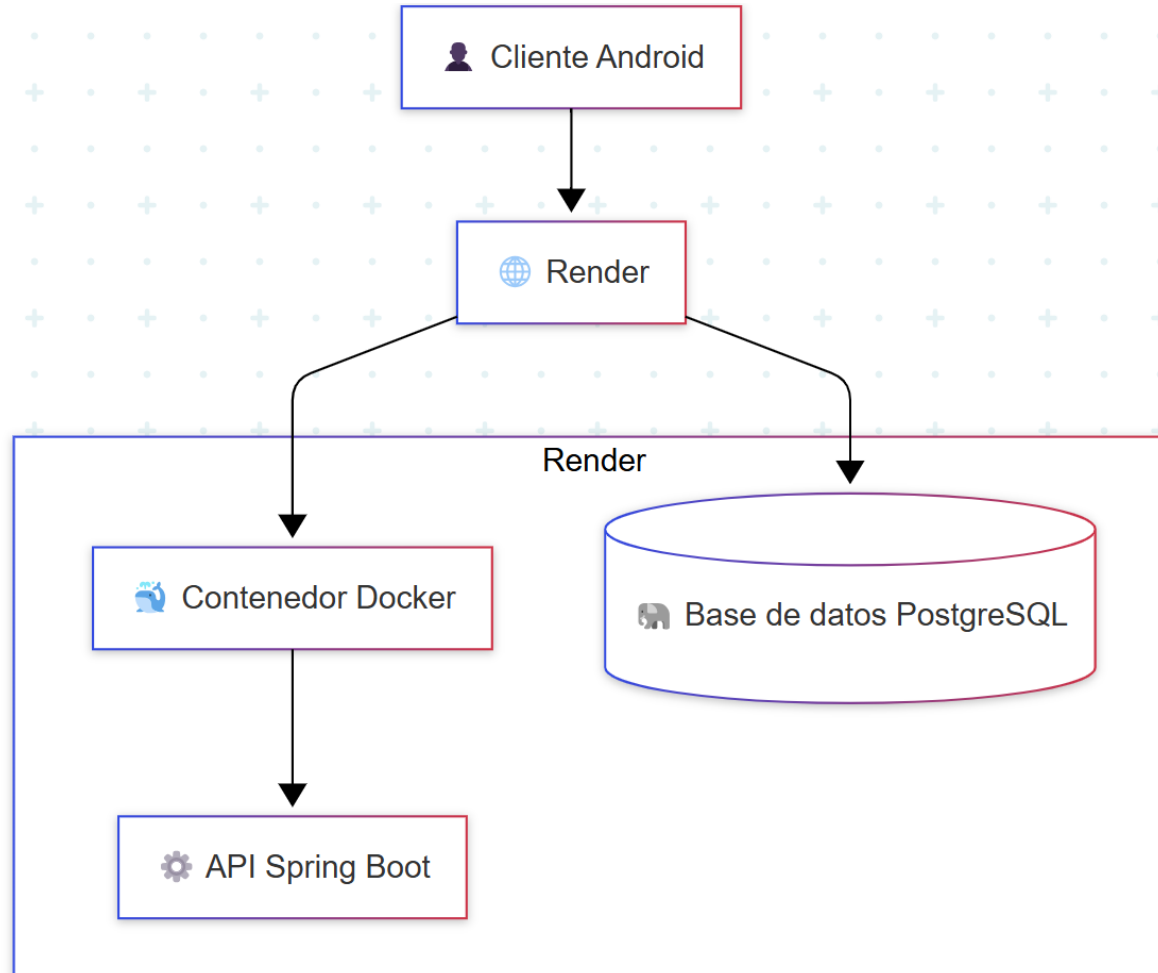
App	Ventajas	Limitaciones
Todoist	Funcional, profesional	Funciones avanzadas de pago
Google Tasks	Integración con Google	No tiene recordatorios ni gráficos
Microsoft ToDo	Buen diseño	Requiere Microsoft 365
<b>PaTareas</b>	Gratuita, con estadísticas	Enfocada a uso individual

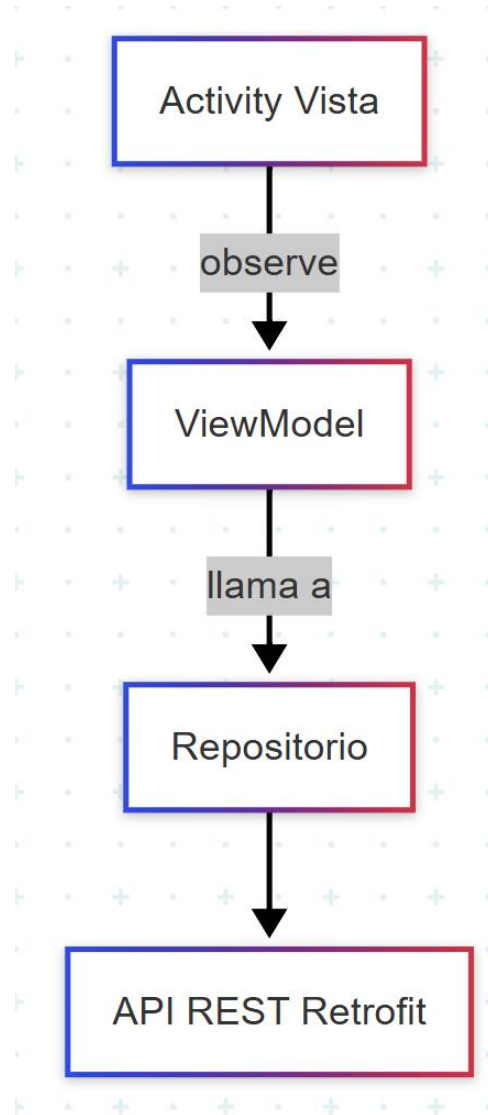


# PLANIFICACIÓN DEL PROYECTO

Fases

# ARQUITECTURA DESPLIEGUE





---

# ARQUITECTURA MVVM APLICADA

- Separar vista y lógica me permitió tener código limpio, reutilizable.



```
public LiveData<List<Tarea>> getTareasPorUsuario(UUID usuarioId, String username, String password) {  
    setBasicAuth(username, password);  
    MutableLiveData<List<Tarea>> tareasLiveData = new MutableLiveData<>();  
  
    apiService.getTareasPorUsuario(usuarioId).enqueue(new Callback<List<Tarea>>() {  
        @Override  
        public void onResponse(Call<List<Tarea>> call, Response<List<Tarea>> response) {  
            if (response.isSuccessful() && response.body() != null) {  
                tareasLiveData.postValue(response.body());  
            } else {  
                tareasLiveData.postValue(null);  
            }  
        }  
    })  
}
```

# EJEMPLO DE VIEWMODEL (MVVM)

La lógica de negocio se separa de la interfaz mediante ViewModel, siguiendo el patrón MVVM.

---

# TECNOLOGÍAS UTILIZADAS



**Área**



Frontend



Backend



Base de datos



Control de versiones



**Tecnologías / Herramientas**

Android SDK, Java, XML, LiveData


Spring Boot, JPA, Retrofit, Postman

PostgreSQL, Docker, pgAdmin

GitHub

---

# APP



Login

Nombre

Contraseña

Login

No tienes cuenta? REGISTRATE

Registrar

+

—

1:1

—

✓ No se si funciona (Completada)

no

Baja

09/05/2025

✓ Hola com estas (Completada)

No se si funcionara esto o que

Baja

07/05/2025

✓ Funciona (Completada)

A la hora de que la tarea esta completada

Medio

03/05/2025

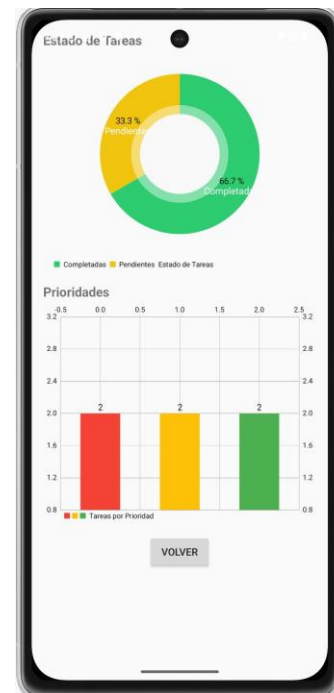
✓ Funciona (Completada)

no funciona

Alta

20/04/2025

VOLVER



Funciona

No funciona lo de completar

Alta

13/05/2025

Nueva tarea

Esta es la descripcion de la Tarea

Medio

10/05/2025

✓

+

---

# API REST

✓ Backend  
desarrollado con  
Spring Boot

✓ CRUD completo  
para usuarios y  
tareas

✓ Seguridad con  
autenticación  
BasicAuth

✓ Comunicación  
desde Android  
mediante Retrofit

✓ Datos  
gestionados  
desde PostgreSQL

---

```
@GET("tareass/usuario/{id}")
Call<List<Tarea>> getTareasPorUsuario(@Path("id") UUID usuarioId);

@POST("tareass/usuario/{id}")
Call<Tarea> crearTarea(@Path("id") UUID usuarioId, @Body Tarea tarea);

@POST("usuarios/register/")
Call<ResponseBody> registrarUsuario(@Body UserRegisterRequest request);
```

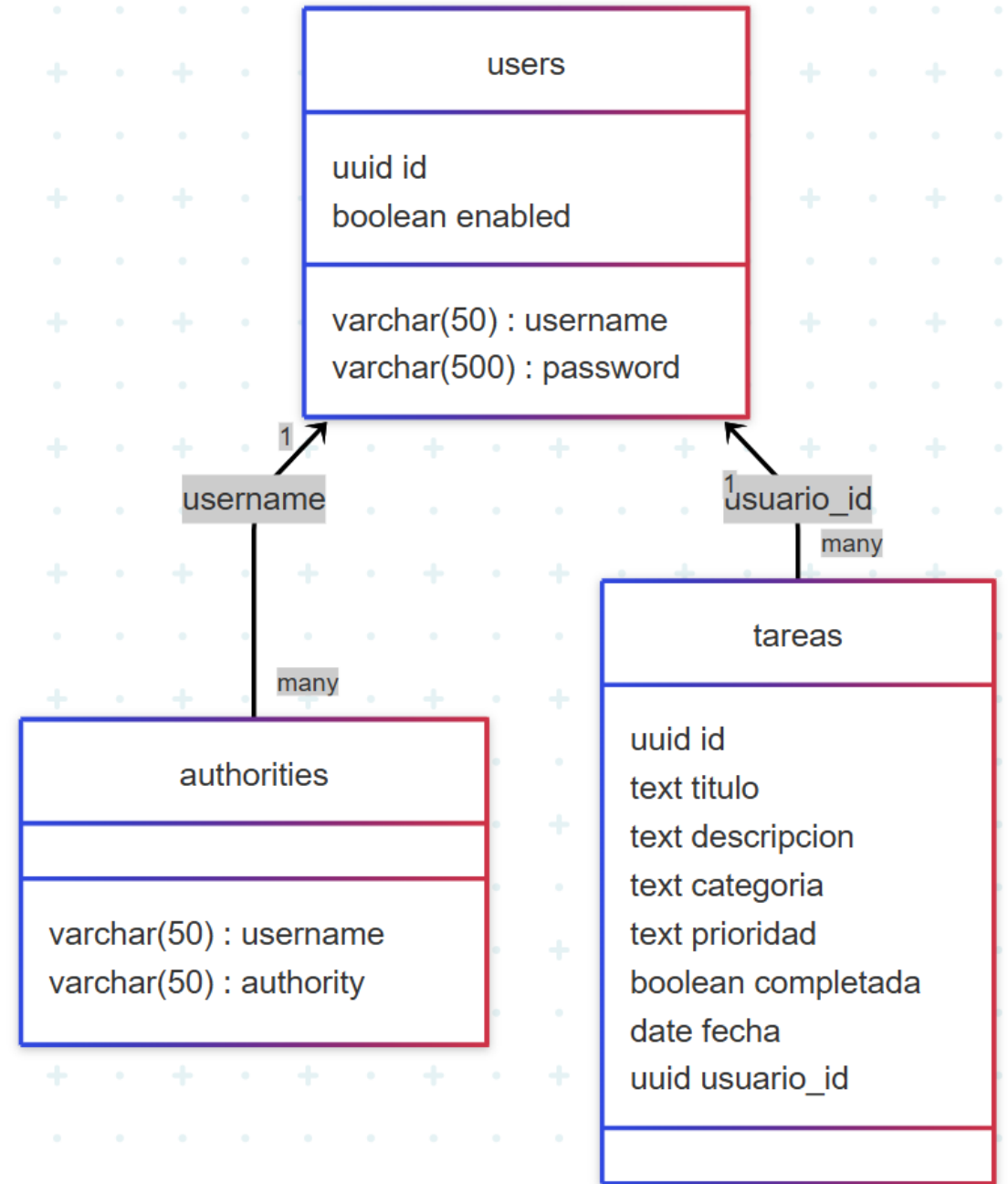
## EJEMPLO DE ENDPOINT EN EL BACKEND

- La API REST permite la creación de tareas mediante el siguiente endpoint:

---

# BASE DE DATOS

- ✓ PostgreSQL desplegado con Docker
- ✓ Gestión de datos en la nube (Render)
- ✓ Integración con Spring Boot (JPA)
- ✓ Estructura relacional: usuarios y tareas
- ✓ Administración con pgAdmin



---

# CONCLUSIONES



La aplicación funciona  
de forma estable



Proyecto desarrollado  
con herramientas  
accesibles y gratuitas



Aprendizaje práctico  
sobre desarrollo  
multiplataforma

---

GRACIAS POR SU  
ATENCIÓN

