



ANÁLISIS Y DISEÑO

# SPRINT 1 DEL PROYECTO

GRUPO 6





ECONOFARMA

# INTRODUCCIÓN

# LISTADO DE TAREAS

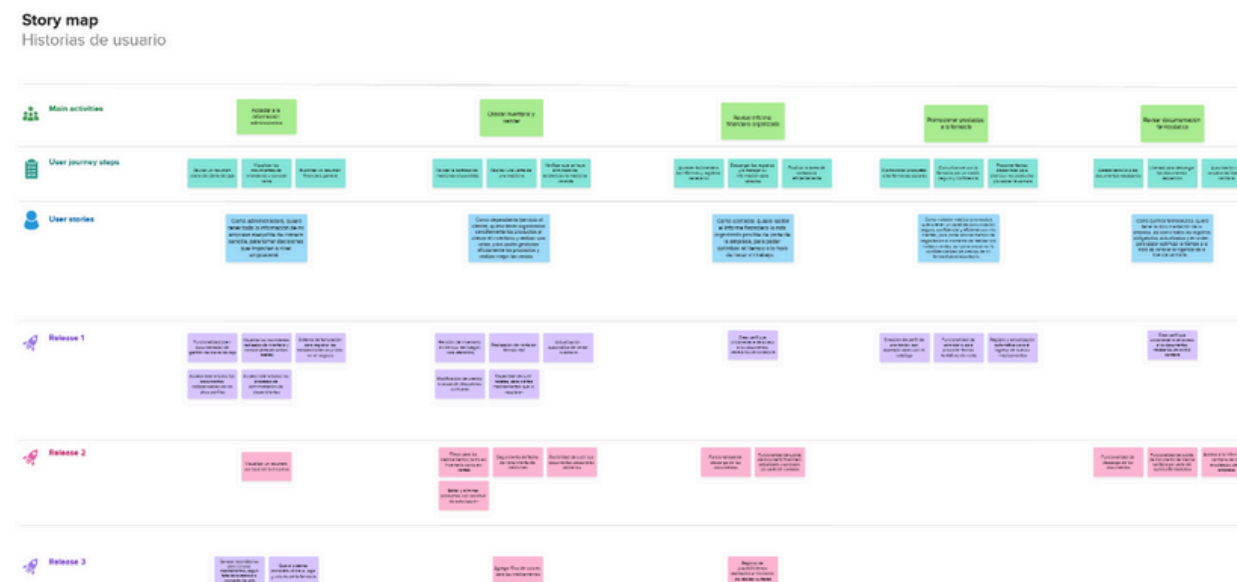


## Frontend:

- Diseñar calendario con eventos. (8 puntos)
- Implementar botones para: (4 puntos)
- Agregar evento
- Eliminar evento
- Editar evento
- Cancelar evento
- Diseñar vista de actividades del mes. (2 puntos)

## Backend:

- Configurar conexión con la base de datos de eventos. (4 puntos)
- Implementar almacenamiento y eliminación de eventos. (4 puntos)



# SPRING BACKLOG

Trabajamos con la herramienta JIRA para facilitar el proceso de gestión del proyecto.

- Calendario
- Tareas completadas



# Jira

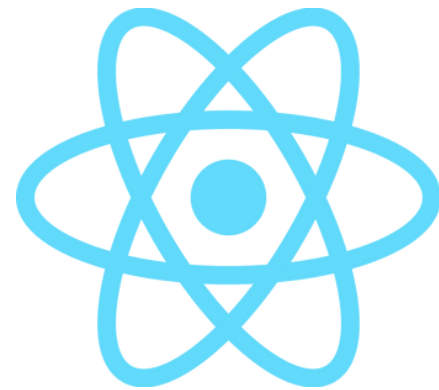


# FRAMEWORKS

## FRONTEND



Vite  
Javascript  
TypeScript

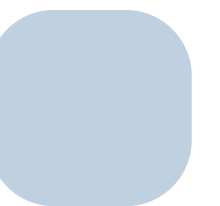


React  
Javascript

## BACKEND

express 

Express  
Javascript





# INFRAESTRUCTURA

## DEL PROYECTO

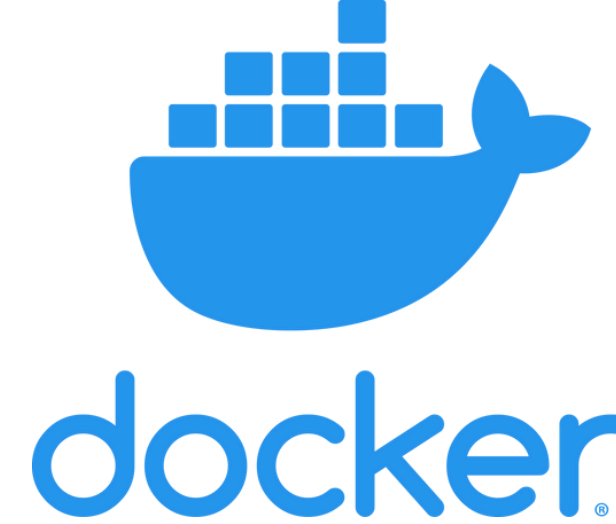
Elección de la herramienta de contenerización: Docker  
Archivos de configuración. Enlace a los repositorios de GitHub:

- [Documentación](#)
- [Frontend](#)
- [Backend](#)



# RESULTADOS

# Backend



## Docker-compose

```
services:
  # Servicio de la base de datos PostgreSQL
  > Run Service
  db:
    image: postgres:16-alpine #Versión Postgres 16
    container_name: postgres_db
    environment:
      POSTGRES_USER: ${DB_USER} #Credenciales,
      POSTGRES_PASSWORD: ${DB_PASSWORD}
      POSTGRES_DB: ${DB_NAME}
    volumes:
      - db_data:/var/lib/docker/postgresql/data
    ports:
      - "2222:5432" #Puertos designados. 2222 p
```

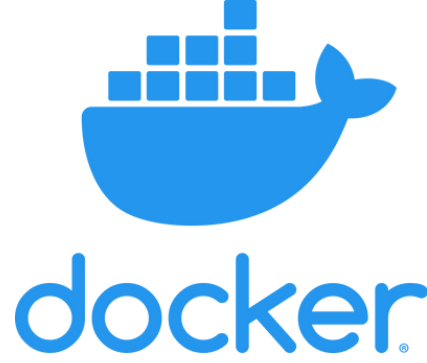
```
# Servicio del backend (Express)
> Run Service
backend:
  build:
    context: .
    dockerfile: Dockerfile #Se carga a partir
  container_name: express_backend
  environment: #Se acceden a las credenciales
    PORT: ${PORT}
    DB_HOST: ${DB_HOST}
    DB_USER: ${DB_USER}
    DB_PASSWORD: ${DB_PASSWORD}
    DB_NAME: ${DB_NAME}
  depends_on:
    - db #Se coloca que depende de la imagen d
  ports:
    - "3000:3000" #Puerto computadora local : p
```

(Hola, soy Renato. Estoy enfermo de la garganta, no tengo voz👍)

## Dockerfile

```
1 FROM node:22.14.0-alpine
2 WORKDIR /app
3 #Se elige node 22; y se le
4
5 # Se le pasan a Docker las
6 COPY package*.json ./
7 RUN npm install
8
9 # Copia el código fuente en
10 COPY ./express-backend .
11
12 #Se expone el puerto 3000
13 EXPOSE 3000
14 CMD ["npm", "start"]
15
```





# Frontend

## Docker-compose

```
1  > Run All Services
1  services:
2    > Run Service
2    frontend:
3      build:
4        context: .
5        dockerfile: Dockerfile #Se indica el Dockerfile a utilizar
6        container_name: react_vite_frontend
7      ports:
8        - "3001:3000" #Se expone el puerto 3000 del contenedor al 3001 del host
9      environment:
10        API_URL: ${VITE_API_URL} #Utiliza la variable de entorno VITE_API_URL
11
```

## Dockerfile

```
1  FROM node:22.14.0-alpine
2  WORKDIR /app
3  #Se eligió node 22. Se le dice al Dockerfile que use la versión 22 de node
4
5  # Instala las dependencias, que se encuentran en el package.json
6  COPY package*.json ./
7  RUN npm install
8
9  # Copia el código fuente
10 COPY . .
11
12 # Genera el build de producción
13 RUN npm run build
14
15 #Se expone el puerto 3000, del contenedor al host
16 EXPOSE 3000
17 CMD ["npm", "run", "preview"]
18
```

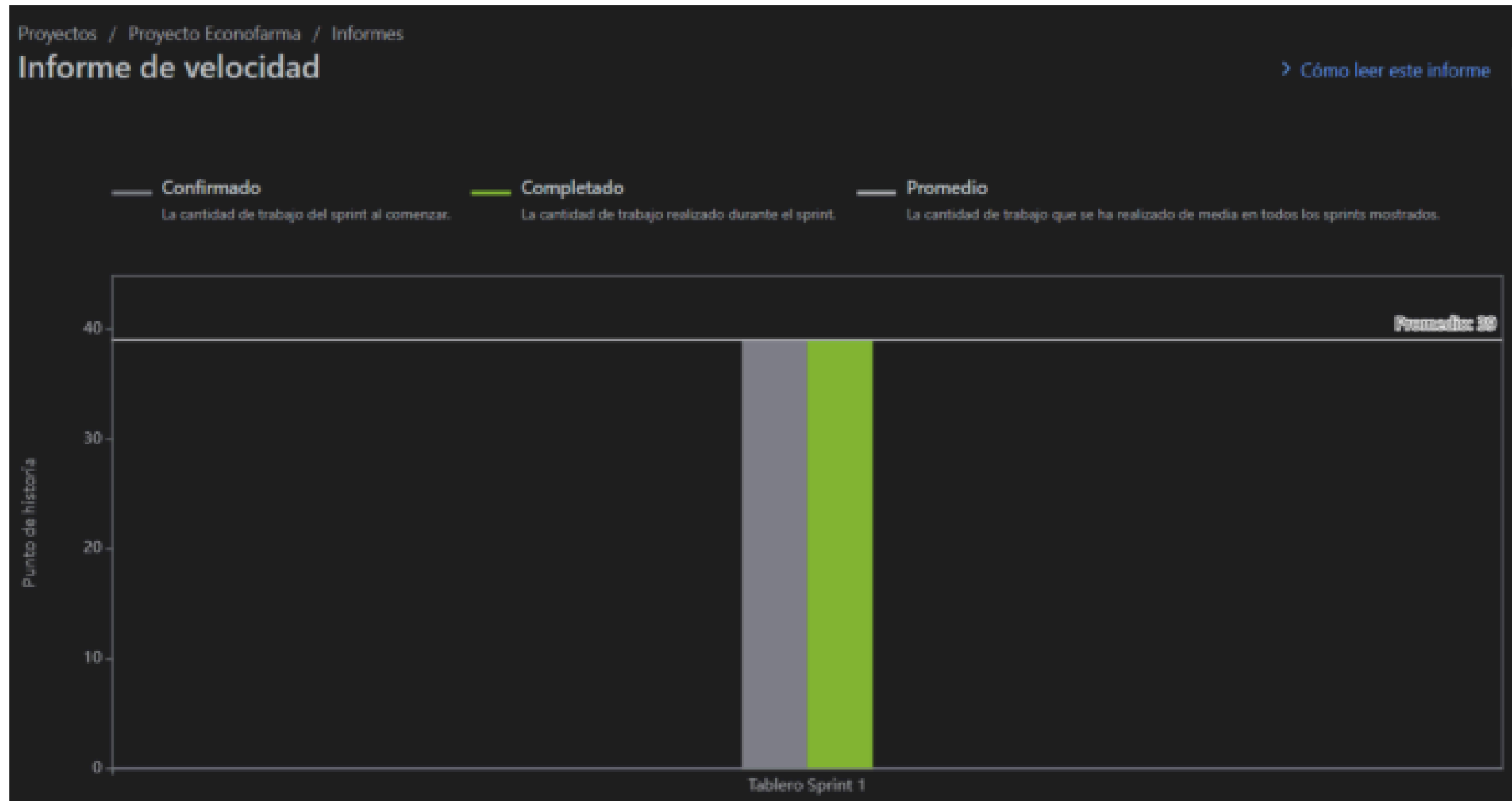
# MÉTRICAS

## BURNDOWN



# MÉTRICAS

## VELOCIDAD



# RETROSPECTIVA

## Retrospectiva sprint 1

Fecha: 06/04/2024

Éxito del sprint: 6 de 10

### ¿Qué fue bien?

Las investigaciones y búsqueda de información que se realizaron fueron variadas, con catedráticos de diferentes cursos que permitieron cambiar las perspectivas.

Las reuniones con el cliente permitieron comprender la modalidad de trabajo con los productos y laboratorios para poder ir completando la base de datos.

Cada quien busco información extra y no se conformo con lo minimo para poder entender lo que se necesitaba en el sprint.

### ¿Qué fue mal?

Cada quien se había asignado tareas individuales y que se debían trabajar grupal ya que los primeros sprints ayudan a estructurar el proyecto.

Hubieron cambios con las tecnologías a utilizar lo que genero un atraso bastante significativo en el proceso de desarrollo.

La comunicación grupal fue baja, lo que genero problemas principalmente no se conocían las herramientas a usar y esto genero que la retroalimentación y el apoyo en equipo fuera bastante bajo.

### Ideas

Podemos probar involucrar más al equipo en el establecimiento del entorno

Tomar en consideración refinar de manera continua el backlog.

Seguir usando las historias de usuario para la creación de tareas.

### ¿Cómo mejorar?

Definir prioridades de una mejor manera, fechas límite, retroalimentación continua y otras

Dejar de asumir que todos entienden los items del backlog sin discusión.

Mejorar las descripciones de las tareas con todas sus responsabilidades y los posibles caminos que pueden tomar.



¡MUCHAS  
GRACIAS!

