


RFI I. Proyecto Votación - Metodologías de Gestión

Organización de la empresa

ESTRUCTURA DE LA ORGANIZACIÓN DEL EQUIPO

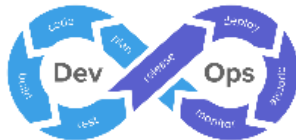
Opciones consideradas

- ❌ - Estructura jerárquica (vertical)
 - ✅ - Estructura plana (horizontal) 
 - ❌ - Estructura matricial
 - ❌ - Estructura en red
 - ❌ - Estructura en células o squad (modelo Spotify)
 - ❌ - Estructura funcional
 - ❌ - Estructura por proyectos
- Comunicación rápida y directa
 - Mayor autonomía y toma de decisiones distribuida
 - Cultura de trabajo más colaborativa
 - Mayor innovación y creatividad

Organización de la empresa

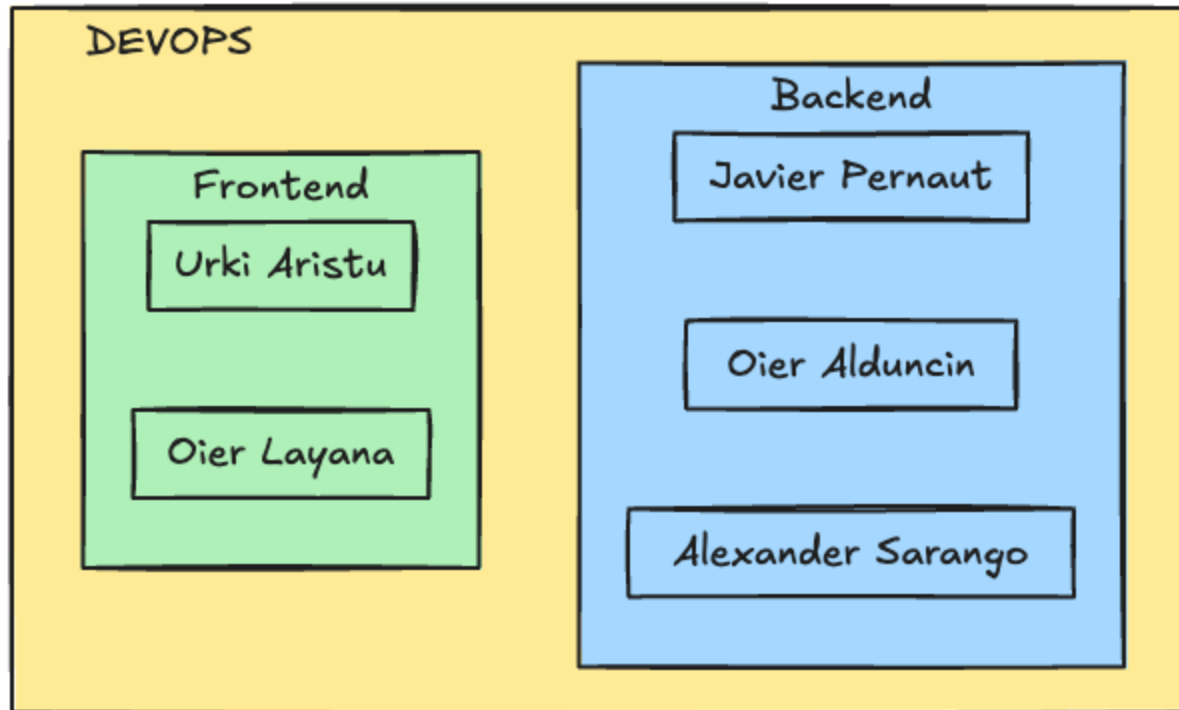
METODOLOGÍA

Opciones consideradas



- + Flujo de trabajo continuo y visual
- + Reducción de desperdicio y mejora en eficiencia
- + Permite priorizar tareas de forma flexible
- + Fácil de integrar con otras metodologías
- + Mayor libertad que otras metodologías

Miembros y roles



Herramientas para la gestión de proyectos

HERRAMIENTAS DE GESTIÓN



Integración nativa con GitHub

Automatización mediante GitHub Actions.

Markdown y vista de tabla personalizable.

Permite conectar issues y pull requests directamente.

Funcionalidad más limitada en comparación con otras herramientas.

No es tan intuitivo para quienes no usan GitHub regularmente.

Menos opciones de personalización visual.

TABLERO KANBAN

The screenshot shows a Kanban board for the project 'GTIO_Votacion'. The board is organized into five columns: Backlog, Ready, In progress, In review, and Done. Each column has a title, a count of items, and an estimate. The 'Backlog' column is empty. The 'Ready' column has one item: 'GTIO_Votacion #13 Creación de proyecto'. The 'In progress' column has one item: 'GTIO_Votacion #33 Presentación RFI I'. The 'In review' column is empty. The 'Done' column has ten items, including 'GTIO_Votacion #7 Crear el navbar y las secciones de la pagina web', 'GTIO_Votacion #11 Crear sección de votaciones', 'GTIO_Votacion #9 Crear sección de participantes', 'GTIO_Votacion #12 Crear sección de resultados', 'GTIO_Votacion #8 Crear sección de home', 'GTIO_Votacion #10 Crear navbar con movimiento entre secciones', 'GTIO_Votacion #1 Definir la arquitectura del programa', 'GTIO_Votacion #23 Documentación', 'GTIO_Votacion #17 Definir metodología', and 'GTIO_Votacion #3 Definir los miembros del equipo y cuáles son sus roles'. The board also features a search bar, a filter by keyword or by field, and buttons for adding, discarding, and saving items.

oielay / Projects / GTIO_Votacion

GTIO_Votacion

Backlog Priority board Team items Roadmap In review My items + New view

Filter by keyword or by field

Discard Save

Backlog 0/20 Estimate: 0
This item hasn't been started

Ready 1 Estimate: 0
This is ready to be picked up

GTIO_Votacion #13
Creación de proyecto

In progress 1/10 Estimate: 0
This is actively being worked on

GTIO_Votacion #33
Presentación RFI I

In review 0/10 Estimate: 0
This item is in review

Done 27 Estimate: 0
This has been completed

GTIO_Votacion #7
Crear el navbar y las secciones de la pagina web

GTIO_Votacion #11
Crear sección de votaciones

GTIO_Votacion #9
Crear sección de participantes

GTIO_Votacion #12
Crear sección de resultados

GTIO_Votacion #8
Crear sección de home

GTIO_Votacion #10
Crear navbar con movimiento entre secciones

GTIO_Votacion #1
Definir la arquitectura del programa

GTIO_Votacion #23
Documentación

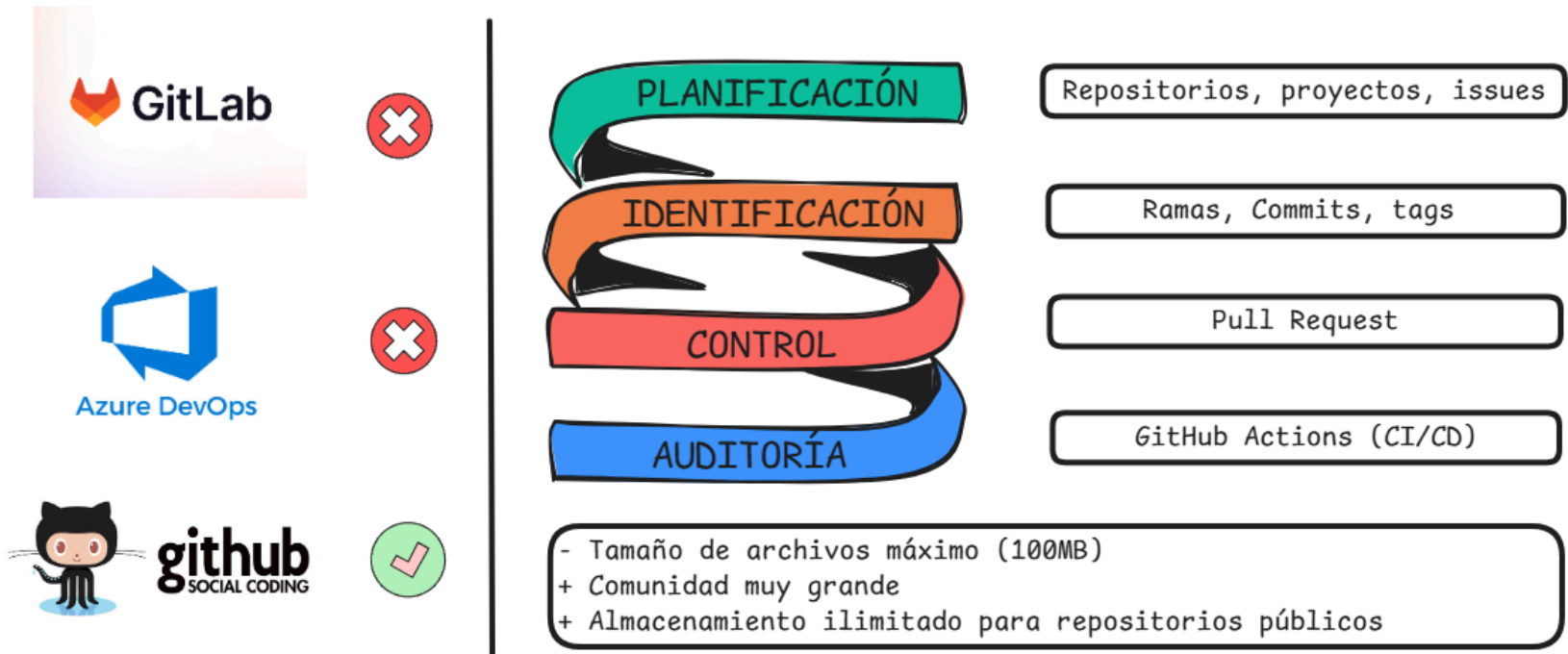
GTIO_Votacion #17
Definir metodología

GTIO_Votacion #3
Definir los miembros del equipo y cuáles son sus roles

+ Add item

Herramientas de gestión de la configuración

GESTIÓN DE LA CONFIGURACIÓN



Herramientas de gestión de la configuración

ESTRUCTURA DE COMMITS

OPCIONES CONSIDERADAS:

1. Sin estructura estándar
2. Convenciones personalizadas
3. Convención basada en Conventional Commits

JUSTIFICACIÓN

1. Estandariza los mensajes
2. Mejora colaboración.
3. Asocia cada commit a una issue.

DECISIÓN:

Convención de mensajes de commit basada en Conventional Commits

ESTRUCTURA:

`<tipo>(<área>): <mensaje corto> (#<número_issue>)`

Ejemplos:

- feat (frontend): crear seccion de participantes (#123)
- fix (backend): corregir guardado de voto en BBDD (#456)
- docs (rfi1): actualizar ADR009 Estructura de Commits (#789)

Herramientas de gestión de la configuración

ARQUITECTURA DEL PROYECTO

MICROSERVICIOS

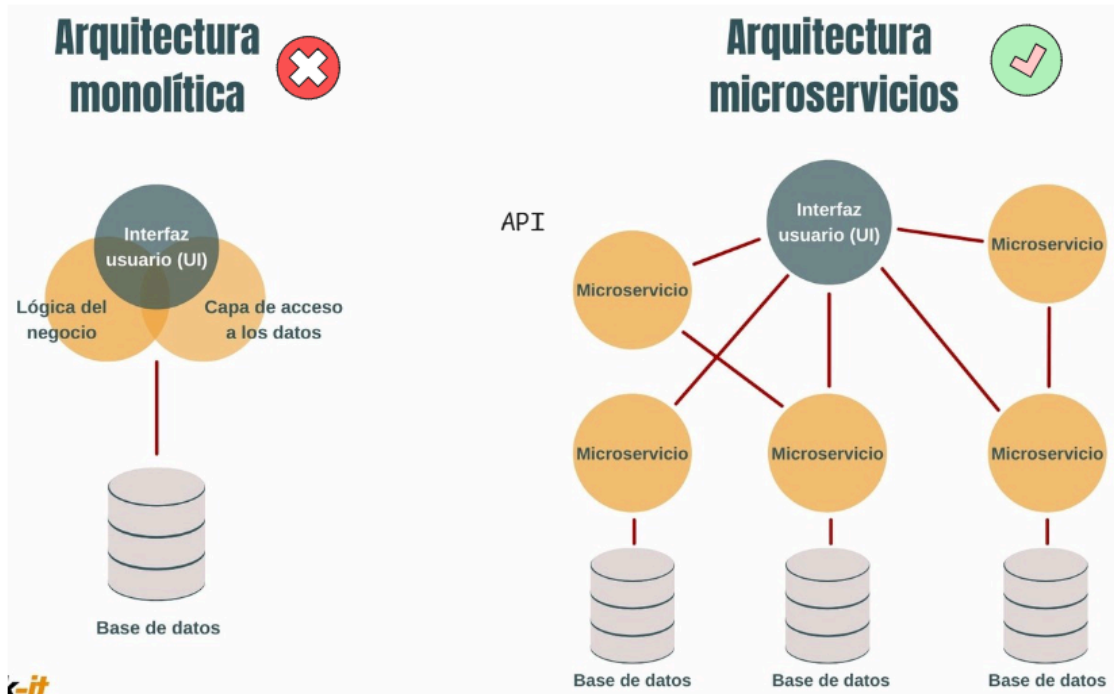
- Complejidad
- Consistencia en los datos
- + Tecnologías diferentes
- + Escalabilidad individual
- + Facilidad de mantenimiento

FRONTEND

- + Un único proyecto
- + Arquitectura de componentes

BACKEND

- + Varios proyectos (APIs)
- + Arquitectura de microservicios
- + Arquitectura en capas



Herramientas de gestión de la configuración

PULL REQUESTS

OPCIONES:

1. No revisión ❌

- -Inconsistente
- -Riesgo
- +Rápido

2. Revisión post-fusión ❌

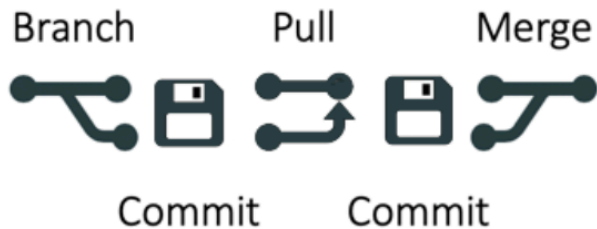
- +Rapidez en desarrollo
- -Errores en rama principal

3. Revisión pre-fusión ✅

- -Tedioso
- +Calidad

JUSTIFICACION:

- Cambios revisados
- Facilita la colaboración
- Mejora la trazabilidad
- Pocos errores en main
- Historial de cambios limpio



REGLAS:

1. Cada PR asociado a una issue padre.
2. Uso de estructura de commits en título.
3. División de cambios en commits asignados a sub-issues.
4. Ejecución de pruebas y validaciones antes de solicitar revisión.
5. Revisión por al menos un miembro antes de aprobación.

Despliegue en entorno de test local

DESPLIEGUE



1. Primero se despliega manualmente con npm por simplicidad y falta de proyecto dockerizado
2. Tras dockerizar se hará despliegue con docker



PASOS PARA DESPLIEGUE EN LOCAL

1. `npm install`
2. `npm run dev`
3. Acceder a `http://localhost:4321`

Arquitectura del programa

BACKEND



+ Experiencia previa

+ Gran rendimiento

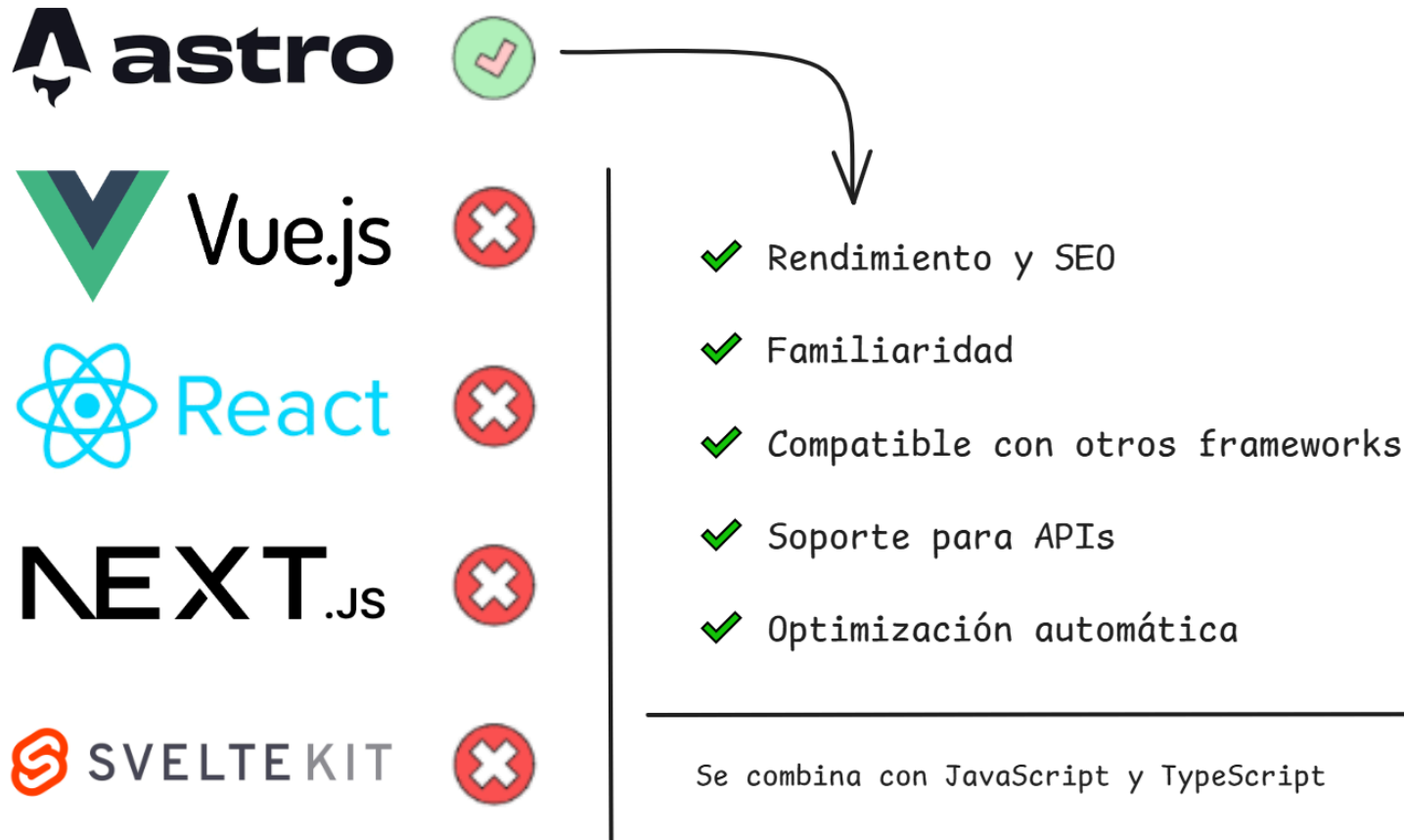
+ Seguridad avanzada integrada

+ Soporte nativo para micro-servicios y contenedores

+ Open-source y mantenimiento a largo plazo

Arquitectura del programa

FRONTEND



Arquitectura del programa

BASE DE DATOS



- Popular
- + Compatibilidad
- + Entity Framework

- + Entorno Microsoft
- + Microsoft Azure
- + Visual Studio

¿Azure SQL ó alojar BD?

Demo