



Plataforma Interactiva **SpaceX**

Datos Técnicos + Comunidad Social en Tiempo Real

Contenido

01 Introducción

Visión general y objetivos del proyecto

03 Funcionalidades

Bloque informativo y social

05 Planificación de Sprints

Metodología ágil y gestión del proyecto

02 Análisis y Arquitectura

Fuentes de datos y estructura técnica

04 Stack Tecnológico

Arquitectura, modelo de datos y seguridad

06 Conclusión

Logros y próximos pasos

 **Nota:** Este documento detalla el desarrollo completo de una plataforma web que integra datos técnicos de SpaceX con funcionalidades sociales en tiempo real.

01

Introducción

Visión general del proyecto y su propuesta de valor única en el ecosistema espacial

Introducción al Proyecto

Visión

Crear una aplicación web que combine **información técnica y estadísticas** de los lanzamientos y cohetes de SpaceX con un **espacio social** donde los usuarios puedan interactuar mediante foros y chat en tiempo real.

Dos Pilares Fundamentales

Visualización de Datos

Consumir datos de la API pública de SpaceX y presentarlos de forma atractiva y comprensible

Comunidad Social

Permitir registro de usuarios, creación de comunidad, debates en foros y comunicación privada

Propuesta de Valor

Una **solución moderna y funcional** que no solo sirve como recurso educativo sobre SpaceX, sino también como ejemplo práctico de cómo se construyen aplicaciones web full-stack con integración de APIs, bases de datos y comunicación en tiempo real.

Diferenciación

- ✓ Primera plataforma que combina datos oficiales de SpaceX con comunidad social integrada
- ✓ Experiencia inmersiva tipo "mission control center"
- ✓ Tecnologías de vanguardia (Next.js 14, Socket.io, Turso)

Objetivo del Proyecto



Bloque Informativo

⌚ Datos Actualizados

Visualizar información en tiempo real de la API de SpaceX de forma ordenada y atractiva

📈 Estadísticas Visuales

Mostrar gráficos interactivos (lanzamientos por año, tasa de éxito, comparativas)

🔍 Exploración Avanzada

Buscador con filtros múltiples para encontrar lanzamientos, cohetes y misiones



Bloque Social

👤 Gestión de Usuarios

Registro, perfiles personalizables (foto, bio), sistema de amigos con solicitudes

💬 Foros Temáticos

Espacios de debate sobre cohetes, misiones, Marte, Starship con likes y comentarios

⚡ Chat en Tiempo Real

Mensajes privados (amigos) y grupales con indicador de conexión y notificaciones

◎ **Meta Final:** Una aplicación web completa que mezcla lo técnico con lo social, creando un ecosistema único para entusiastas del espacio.

Público Objetivo



Fans del Espacio

Entusiastas de SpaceX y la exploración espacial que buscan **toda la información en un solo lugar**, actualizada y presentada de forma atractiva.

[Lanzamientos](#)[Estadísticas](#)[Comunidad](#)

Estudiantes Tech

Estudiantes y profesionales tecnológicos que quieren ver **ejemplos prácticos** de integración de APIs, visualización de datos y desarrollo de chats en tiempo real.

[Next.js](#)[APIs](#)[WebSocket](#)

Entusiastas de la Ciencia

Personas apasionadas por la ciencia que buscan un **espacio para debatir**, compartir conocimiento y conectar con otros aficionados.

[Foros](#)[Debate](#)[Chat](#)

Necesidades Comunes

- ✓ Información precisa y actualizada
- ✓ Interfaz intuitiva y visualmente atractiva
- ✓ Comunidad activa y comprometida

Segmentación

- **Primario (60%):** Fans de SpaceX y espacio
- **Secundario (25%):** Estudiantes y desarrolladores
- **Terciario (15%):** Entusiastas de la ciencia



02

Análisis y Arquitectura

Fuentes de datos y estructura técnica del sistema

Análisis de la Fuente de Datos

API Pública de SpaceX

Utilizamos la **API oficial y gratuita** de SpaceX que proporciona datos reales y actualizados sobre todos los aspectos de la empresa: lanzamientos, cohetes, cápsulas, naves de apoyo e información corporativa.

Tipos de Datos Disponibles

Lanzamientos

Fecha, cohete utilizado, resultado (éxito/fallo), enlaces a vídeos, detalles de la misión

Cohetes

Especificaciones técnicas: altura, peso, capacidad de carga, motores, imágenes

Cápsulas Dragon

Información sobre Dragon 1 y 2, historial de vuelos, tripulación

Naves de Apoyo

Datos de barcos de recuperación (GO Navigator, etc.)

Integración con la Aplicación

Peticiones HTTP

La app realiza peticiones a la API para obtener datos actualizados

Procesamiento

Los datos se procesan y transforman para su visualización

Visualización

Se generan listas, fichas de detalle y gráficos estadísticos

Características Técnicas

RESTful API: Endpoints bien estructurados

JSON Format: Datos en formato estándar

Rate Limiting: Gestión de límites de peticiones

Caché: Sistema de caché para optimizar rendimiento

Error Handling: Manejo robusto de errores

Estructura General del Sistema



Framework Unificado

Next.js 14

App Router

✓ Frontend (UI/UX)

✓ Backend (API Routes)

✓ Conexión BD



Persistencia

Turso

SQLite Cloud

✓ Usuarios

✓ Mensajes

✓ Foros



Tiempo Real

Socket.io

WebSockets

✓ Chat privado

✓ Chat grupal

✓ Notificaciones

Flujo de Datos

1

Usuario → Frontend
Interacción con la interfaz React

2

API Routes → Backend
Procesamiento de peticiones

3

Base de Datos / API SpaceX
Persistencia y datos externos



Arquitectura Monolítica

Hemos optado por una **arquitectura monolítica** con Next.js como framework principal, que simplifica el desarrollo y despliegue:

- ✓ **Ventaja:** Un solo código base para frontend y backend
- ✓ **Ventaja:** Despliegue simplificado en Vercel
- ✓ **Excepción:** Chat en tiempo real como microservicio separado

03

Funcionalidades

Características principales de la plataforma



Bloque Informativo (Parte 1)

Página de Inicio

- Resumen del proyecto y su propósito
- Información del **próximo lanzamiento** con cuenta atrás en tiempo real
- Accesos directos a secciones principales
- Diseño responsive y visualmente atractivo

Fichas de Lanzamiento

- Detalles completos de cada misión
- Imágenes, vídeos y enlaces externos
- Enlace directo al foro para comentar
- Información del cohete y carga útil

Lista de Lanzamientos

- Catálogo completo de lanzamientos (pasados y futuros)
- **Filtros avanzados:** por año, cohete, resultado (éxito/fallo)
- Vista en grid/tabla responsive
- Paginación y ordenamiento

Catálogo de Cohetes

- Todos los modelos de SpaceX (Falcon 9, Falcon Heavy, Starship)
- Especificaciones técnicas detalladas
- Comparativas visuales entre modelos
- Galería de imágenes y diagramas

Bloque Informativo (Parte 2)

Cápsulas y Naves

- Información sobre cápsulas Dragon 1 y 2
- Historial de vuelos y misiones
- Naves de apoyo: GO Navigator, GO Searcher, etc.
- Datos de recuperación y reusabilidad

Estadísticas

- Gráficos interactivos:
- Lanzamientos por año
- Tasa de éxito vs fracaso
- Cohetes más utilizados
- Carga útil total lanzada

Buscador Avanzado

- Búsqueda global en toda la plataforma
- Filtros combinables: nombre, año, tipo de cohete, resultado
- Autocompletado y sugerencias
- Historial de búsquedas recientes

Widget Próximo Lanzamiento

- Componente reutilizable visible en múltiples páginas
- **Cuenta atrás en tiempo real** hasta el próximo lanzamiento
- Información resumida: fecha, cohete, misión
- Enlace directo a detalles completos

Bloque Social (Parte 1)

Sistema de Usuarios

Autenticación

Registro e inicio de sesión seguro

Perfiles

Nombre, foto, biografía personalizable

Edición

Modificar datos y preferencias

Sistema de Amigos

Buscador

Encontrar otros usuarios por nombre

Solicitud

Enviar, aceptar, rechazar o cancelar

Lista de Amigos

Ver conexiones y estado online

Flujo de Amistad

1

Búsqueda

Usuario A busca y encuentra a Usuario B

2

Solicitud

A envía solicitud (estado: pendiente)

3

Notificación

B recibe notificación de nueva solicitud

4

Decisión

B acepta o rechaza la solicitud

5

Conexión

Si acepta: amigos y chat habilitado

ⓘ

Estados: Pendiente | Aceptada | Rechazada | Bloqueada

Bloque Social (Parte 2)

Foro Temático

Secciones

Lanzamientos Marte Starship Tecnología

Hilos

Crear temas y responder a otros

Conexión

Hilos vinculados a lanzamientos

Likes

Sistema de me gusta en posts

Seguridad Social

- ✓ Chats privados solo entre participantes
- ✓ Sistema de bloqueo de usuarios
- ✓ Moderación de contenido en foros
- ✓ Reportes de comportamiento inapropiado

Chat en Tiempo Real

Mensajes Privados

1-a-1 solo entre amigos

Chats Grupales

Grupos generales o por temas

Notificaciones

Alertas de mensajes nuevos

Estado

Indicador de usuarios online

Sistema de Notificaciones

- Nueva solicitud de amistad
- Nuevo mensaje privado
- Respuesta en tu hilo del foro
- Like en tu publicación
- Mención en comentario

Valor Añadido e Innovación



Diferenciadores Clave

💡 Mezcla Única

Primera plataforma que integra **datos técnicos oficiales** de SpaceX con una **comunidad social activa**. No existe otra solución similar en el mercado.

⚡ Tiempo Real Dual

Datos de la API actualizados al instante + **chat en vivo** durante los lanzamientos. Experiencia inmersiva tipo "mission control".

👥 Espacio Colaborativo

Un sitio donde los fans pueden **compartir conocimiento**, discutir sobre misiones y conectar con otros entusiastas del espacio.



Tecnologías Modernas

✓ Next.js 14

Framework full-stack con App Router

✓ React + Tailwind CSS

UI moderna, responsive y rápida

✓ Socket.io

Comunicación bidireccional en tiempo real

✓ Turso (SQLite)

Base de datos moderna y escalable

🏆 **Proposición Única:** Transformamos una API informativa en una **experiencia comunitaria** donde el usuario no solo consume datos, sino que participa activamente a través de foros y chats durante los eventos de lanzamiento.

🎓 Valor Educativo

Ejemplo práctico de **desarrollo web moderno** con integración de APIs, bases de datos y comunicación en tiempo real.

↗ Escalabilidad

Arquitectura preparada para **crecer**: más usuarios, nuevas funcionalidades, integración con otras APIs espaciales.

04

Stack Tecnológico

Tecnologías y arquitectura del proyecto

Arquitectura del Proyecto



Framework Principal

Next.js 14

Frontend + Backend unificados

- ✓ React (UI)
- ✓ Tailwind CSS
- ✓ API Routes
- ✓ App Router



Base de Datos

Turso

SQLite en la nube

- ✓ Relacional
- ✓ Edge-deployed
- ✓ Baja latencia
- ✓ Escalable



Tiempo Real

Node.js + Socket.io

Microservicio separado

- ✓ WebSockets
- ✓ Chat privado
- ✓ Chat grupal
- ✓ Notificaciones

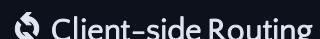
</> Frontend (Next.js + React)



Framework CSS utility-first para diseño rápido y consistente



Componentes reutilizables para UI modular y mantenible



Navegación fluida sin recargas de página

Backend (API Routes)



API routes para registro, login y gestión de sesiones



Validación de datos y protección contra vulnerabilidades



Procesamiento de peticiones y operaciones CRUD

Modelo de Datos Conceptual

Usuario

- ID (PK)
- nombre
- email (unique)
- password_hash
- avatar_url
- bio
- created_at
- is_online

Amistad

- ID (PK)
- usuario_id (FK)
- amigo_id (FK)
- estado
 - pendiente
 - aceptada
 - bloqueada
- created_at

PublicacionForo

- ID (PK)
- titulo
- autor_id (FK)
- contenido
- categoria
 - lanzamientos
 - marte, starship
- created_at
- likes_count

ComentarioForo

- ID (PK)
- autor_id (FK)
- publicacion_id (FK)
- contenido
- created_at
- parent_id (FK)
- (respuestas anidadas)

MensajeChat

- ID (PK)
- emisor_id (FK)
- receptor_id (FK)
- tipo_receptor
 - usuario
 - grupo
- contenido
- created_at
- leido

LikePublicacion

- ID (PK)
- usuario_id (FK)
- publicacion_id (FK)
- created_at

Notificacion

- ID (PK)
- usuario_id (FK)
- tipo
- contenido
- leida
- created_at

Seguridad y Privacidad



Autenticación Segura

🔒 Hashing de Contraseñas

bcryptjs para almacenamiento seguro (nunca en texto plano)

· Protección de Rutas

Middleware de autenticación para rutas privadas

.. Sesiones

Tokens JWT con expiración configurable



Protección de Datos

💬 Privacidad de Chats

Solo participantes pueden leer mensajes privados

⚡ Validación de Inputs

Sanitización para prevenir XSS e inyección

🔒 HTTPS

Comunicación encriptada end-to-end

Control de Acceso

- Rutas públicas vs privadas
- Permisos por rol de usuario
- Verificación de propiedad
- Rate limiting en APIs

Prevención de Ataques

- XSS (Cross-Site Scripting)
- CSRF (Cross-Site Request)
- SQL Injection
- DoS (Denial of Service)

Cumplimiento

- GDPR (protección de datos)
- Política de privacidad
- Términos de servicio
- Derecho al olvido

⚠️ Principio: "Security by Design" – La seguridad se integra desde el inicio, no como afterthought.

Pruebas y Control de Calidad

Pruebas Unitarias

Pruebas de funciones individuales, especialmente de la API:

- ✓ Registro de usuarios
- ✓ Sistema de amistades
- ✓ CRUD de posts
- ✓ Validación de datos

 Herramienta: Jest

Pruebas E2E

Flujos completos de usuario:

Ejemplo de Flujo:

Registro → Login → Buscar amigo → Enviar solicitud → Aceptar → Iniciar chat

 Herramienta: Cypress

Pruebas de Componentes

Verificación de componentes React:

- ✓ Renderizado correcto
- ✓ Manejo de estados
- ✓ Eventos y callbacks
- ✓ Props y validación

 Herramienta: React Testing Library

Pruebas de Carga

Rendimiento del chat con múltiples usuarios:

- ✓ 10+ usuarios simultáneos
- ✓ 100+ mensajes por minuto
- ✓ Tiempo de respuesta <100ms
- ✓ Estabilidad del servidor

 Herramienta: Artillery / k6

Responsive

Pruebas en múltiples dispositivos y tamaños de pantalla

Cross-browser

Compatibilidad con Chrome, Firefox, Safari, Edge

Accesibilidad

Cumplimiento WCAG 2.1 AA

Despliegue y Mantenimiento

Plataforma de Despliegue

Vercel

Plataforma optimizada para Next.js con despliegue continuo

Turso Cloud

Base de datos SQLite distribuida globalmente

Servidor Socket.io

Despliegue en Railway/Render para el microservicio de chat

Entornos

Desarrollo

Localhost, datos de prueba, logs detallados

Staging

Pre-producción, pruebas finales antes de deploy

Producción

Entorno real, datos reales, usuarios finales

Control de Versiones

Git + GitHub

Repositorio con flujo de trabajo basado en ramas

Estrategia de Ramas

main (producción) ← develop ← feature/*

Pull Requests

Revisões de código antes de merge

Mantenimiento

- ✓ **Monitoreo:** Logs, métricas de rendimiento, alertas
- ✓ **Backups:** Copias de seguridad automáticas de BD
- ✓ **Actualizaciones:** Parches de seguridad, mejoras
- ✓ **Soporte:** Canal de reporte de bugs y sugerencias

Roles y Responsabilidades



Ismael

Diseño y Frontend



Diseño UI/UX

Figma, prototipos, experiencia de usuario



React Components

Desarrollo de componentes reutilizables



Tailwind CSS

Estilos, responsive design, animaciones



Responsive

Adaptación a móviles, tablets, desktop



Figma: SpaceX Interactive Platform



Elías

Base de Datos



Turso

Configuración, esquemas, migraciones



Modelos

Diseño de entidades y relaciones



Queries

Consultas SQL optimizadas



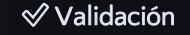
Lucilene

Backend



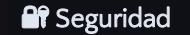
API Routes

Endpoints de Next.js, lógica de negocio



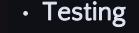
Validación

Validación de datos, sanitización



Seguridad

Autenticación, autorización, protección



Testing

Pruebas unitarias de APIs



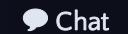
José

Tiempo Real y Comunidad



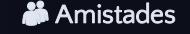
Socket.io

Servidor WebSocket, eventos en tiempo real



Chat

Mensajes privados y grupales



Amistades

Sistema de solicitudes y gestión

05

Planificación de Sprints

Metodología ágil y gestión del proyecto

Visión General de Sprints

📅 Cronograma de Sprints (Diagrama de Gantt)

Sprint 1

1-14 Nov

Configuración + Autenticación

Sprint 2

15-28 Nov

Información SpaceX

Sprint 3

29 Nov-12 Dic

Cápsulas + Mapa + Perfiles

Sprint 4

13-26 Dic

Foros + Chat en Tiempo Real

📝 Asignación de Historias de Usuario

Sprint 1: HU-10, HU-01

Configuración técnica + Registro/login

Sprint 2: HU-02, HU-03, HU-04

Home + Lanzamientos + Cohetes

Sprint 3: HU-05, HU-06, HU-09

Cápsulas + Mapa + Perfiles/Seguidores

Sprint 4: HU-07, HU-08

Foros + Chat global

ℹ️ Metodología Ágil

📅 Duración

Sprints de 2 semanas

⌚ Reuniones

Planificación, Daily Scrum, Review, Retrospective

💻 Herramientas

Trello para Product Backlog y Kanban

👤 Equipo

4 desarrolladores + Product Owner

Sprint 1: Configuración y Autenticación

1 - 14 de Noviembre de 2025 (2 semanas)

⌚ Historias de Usuario

HU-10 (Crítica)

Configurar entorno técnico

HU-01 (Alta)

Registro e inicio de sesión

Justificación: Sin entorno técnico no se puede desarrollar. Sin autenticación, no hay acceso a funciones sociales.

☰ Tareas y Responsables

T1: Proyecto Next.js

Ismael

T2: Tailwind CSS

Ismael

T3: Turso + Conexión

José

T4: Tabla Usuario

José

T5: Formulario Registro

Ismael

T6: API /auth/register

José

T7: API /auth/login

Ismael

🕒 Proceso

Días 1-3: Configuración base

Días 4-7: Formularios y APIs

Días 8-12: Integración

Días 13-14: Testing y demo

⚠ Problemas

Problema: bcrypt incompatible con Next.js App

Router

Solución: Migrar a bcryptjs

Problema: Sesión no persistía

Solución: localStorage temporal

✅ Review

Fecha: 14 Noviembre

Demo: Flujo registro → login → perfil

Feedback PO: "Ver añadido funcional en próxima reunión"

Retrospectiva: Comunicación diaria efectiva

Sprint 2: Información SpaceX

15 - 28 de Noviembre de 2025 (2 semanas)

⌚ Historias de Usuario

HU-02 (Alta)

Página de inicio con próximo lanzamiento

HU-03 (Alta)

Lista de lanzamientos con filtros

HU-04 (Media)

Fichas detalladas de cohetes

Justificación: Con el entorno listo, desarrollamos el núcleo informativo de la plataforma.

☰ Tareas Principales

T8-T10: Página de Inicio

Countdown, accesos rápidos, diseño

T11-T12: Lista Lanzamientos

Grid responsive, filtros avanzados

T13: Fichas Cohetes

Especificaciones técnicas, comparativas

T14-T15: API SpaceX

Endpoints para lanzamientos y cohetes

📅 Proceso

Días 1-3: Investigación API

Días 4-7: Página inicio

Días 8-12: Lista + filtros

Días 13-14: Testing

⚠ Problemas

Problema: Latencia alta en API SpaceX

Solución: Sistema de caché

Problema: URLs de imágenes rotas

Solución: Imagen placeholder

☑ Review

Fecha: 28 Noviembre

Demo: Home, filtros, ficha Falcon 9

Feedback PO: "Excelente. Ver interactividad social en próximo sprint"

Retrospectiva: Caché mejoró rendimiento significativamente

Sprint 3: Cápsulas, Mapa y Perfiles

29 Noviembre - 12 Diciembre 2025 (2 semanas)

⌚ Historias de Usuario

HU-05 (Media)

Información cápsulas Dragon

HU-06 (Media)

Mapa interactivo de bases

HU-09 (Alta)

Perfiles públicos y seguidores

Justificación: Expandimos contenido técnico e iniciamos la transición a la plataforma social.

📅 Proceso

Días 1-4: Endpoints + Leaflet

Días 5-9: Datos reales al mapa

Días 10-14: Sistema seguir

⚠ Problemas

02/12: Mapa no cargaba en móviles → Viewport CSS

07/12: Imágenes faltantes → Fallback visual

11/12: Contador sin actualizar → Estado local React

☰ Tareas Principales

T16-T18: Cápsulas Dragon

Endpoints, listado, historial

T19-T20: Mapa Interactivo

Leaflet, ubicaciones, marcadores

T21: Perfiles Públicos

Diseño, información visible

T22: Sistema Seguir

Solicitudes, estados, contadores

☑ Review

Demo: Mapa interactivo + perfiles

Feedback PO: "Estética de perfiles destacada positivamente"

Retrospectiva: Refactorizar Socket.io antes del chat

Sprint 4: Foros y Chat en Tiempo Real

13 - 26 de Diciembre de 2025 (2 semanas)

◉ Historias de Usuario

HU-07 (Alta)

Foros temáticos (Lanzamientos, Marte, Starship)

HU-08 (Crítica)

Chat global en tiempo real

Justificación: Sprint final que completa el pilar social con comunicación asíncrona (foros) y tiempo real (chat).

☰ Tareas Principales

T23-T24: Servidor WebSockets

Configuración Socket.io, eventos

T25-T26: Esquema Foros

Tablas, posts, comentarios

T27: Layout Chat

Diseño estilo consola espacial

T28-T29: Editor + Notificaciones

Markdown, likes, alertas

☑ Proceso

Días 1-5: Servidor WebSockets

Días 6-10: Desarrollo foros

Días 11-14: Integración + pruebas

⚠ Problemas

16/12: CORS Socket.io → Configuración orígenes

20/12: Necesidad tags → Modificación tabla Foros

24/12: Retraso notificaciones → Simplificar a toasts

☑ Review Final

Fecha: 26 Diciembre

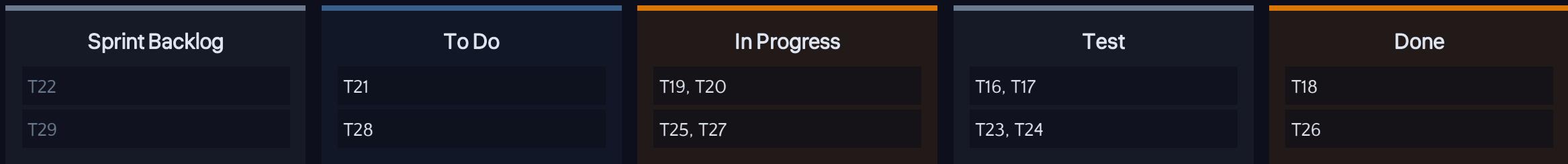
Demo: Registro → Lanzamiento → Foro → Chat

Feedback PO: "Proyecto sobresaliente. Cohesión datos-social única"

Retrospectiva: Planear moderación AI post-sprint

Kanban y Gestión Visual

■ Tablero Kanban por Sprint



⟳ Flujo de Trabajo Ágil

1. Planificación del Sprint

Selección de HU del Product Backlog

2. División en Tareas

Desglose de HU en tareas asignables

3. Daily Scrum

Reuniones diarias de 15 minutos

4. Review + Retrospective

Demo y mejora continua

❖ Herramientas Utilizadas

■ Trello

Product Backlog y tableros Kanban por sprint

❖ GitHub

Control de versiones y Pull Requests

❖ Figma

Diseño UI/UX y prototipos

■ Meet/Discord

Comunicación y Daily Scrums

Conclusión del Proyecto



Sincronización Técnica-Social

Hemos logrado transformar una **API informativa** en una **experiencia comunitaria**, donde el usuario no solo consume datos, sino que participa activamente a través de foros y chats durante eventos de lanzamiento.



Solidez del Stack

El uso de **Next.js 14** junto con **Tailwind CSS** permitió un desarrollo rápido de interfaces tipo "dashboard" profesional. La integración de **Socket.io** fue fundamental para dotar de vida a la plataforma.



Gestión Ágil

La división en **4 sprints** permitió mitigar riesgos técnicos de forma temprana: problemas de latencia de API (Sprint 2), desafíos de persistencia (Sprint 3).

Logros Alcanzados

★ Plataforma Completa

8 semanas de desarrollo, 4 sprints exitosos

★ Tecnologías Modernas

Next.js 14, Turso, Socket.io, Tailwind

★ Integración API SpaceX

Datos en tiempo real con caché optimizado

★ Comunidad Social

Foros, chat en tiempo real, sistema de amigos

Próximos Pasos

→ Transmisiones en Directo

Integración de streams de lanzamientos

→ Gamificación

Sistema de puntos, logros, insignias

→ Moderación AI

Detección automática de contenido inapropiado

→ App Móvil

Versión nativa iOS/Android con React Native

“**Resultado:** Una herramienta robusta y escalable que satisface tanto la necesidad de información técnica precisa como el deseo de pertenencia de la comunidad de entusiastas espaciales.



Gracias

Plataforma Interactiva SpaceX



IES Azarquiel

Toledo, España



Metodologías Ágiles

2025-2026



Equipo

Ismael · Lucilene · Elías · José

✉️ Contacto: equipo@spacex-platform.com

🌐 github.com/spacex-platform