

## INTRODUCCIÓN

Este documento registra el proceso de orquestación con Inteligencia Artificial utilizado para completar el BEX Tech Challenge.

El objetivo principal del reto no fue escribir código manualmente, sino diseñar instrucciones precisas (prompts) para que una IA generara una interfaz funcional, siguiendo un stack y reglas estrictas.

Durante el desarrollo:

- No se escribió lógica de componentes manualmente.
- Todo el código fue generado mediante IA.
- El rol asumido fue el de supervisor y arquitecto, no el de programador tradicional.

El enfoque se centró en:

- Arquitectura clara con React Router v7.
- Uso consistente de shadcn/ui y Tailwind CSS.
- Iteración controlada mediante prompts para corregir errores y mejorar el resultado visual.
- Validación continua mediante build y typecheck.

Este log documenta cómo se interactuó con la IA, qué decisiones se tomaron y cómo se alcanzó el resultado final.

## FLUJO DE TRABAJO DEFINIDO

Antes de comenzar a generar código, se definió un flujo de trabajo por fases, con el fin de:

- Mantener control del proceso.
- Evitar trabajo manual.
- Asegurar cumplimiento estricto de las reglas del challenge.

El flujo se basa en una iteración constante entre:

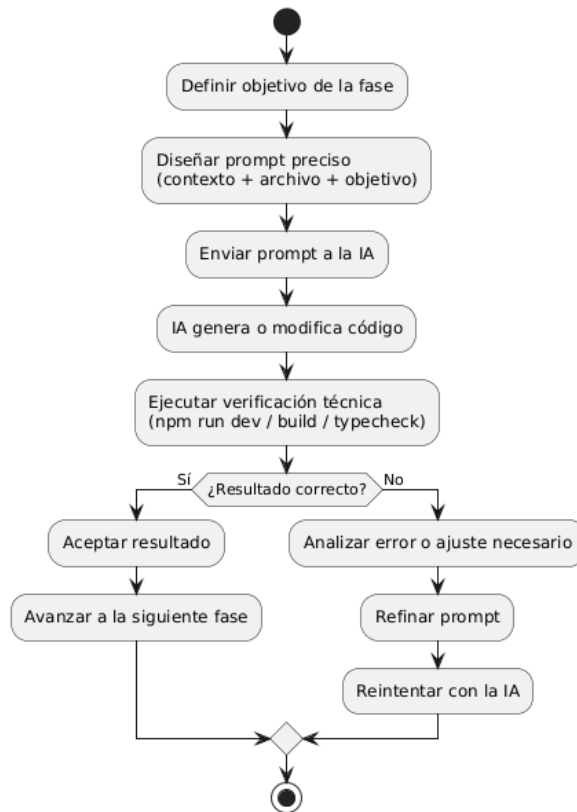
Planificación → Prompt → Generación IA → Verificación → Corrección por Prompt

## DESCRIPCIÓN DEL FLUJO

1. Definición del objetivo
  - Se establece qué parte del sistema se va a construir o modificar (setup, layout, KPIs, tabla, etc.).
2. Diseño del prompt
  - Se redacta una instrucción clara, específica y contextualizada.
  - El prompt indica a la IA qué archivo modificar o qué estructura generar.
3. Generación de código por IA
  - La IA produce el código completo según las instrucciones.
  - No se realizan ediciones manuales.
4. Verificación
  - Se ejecutan comandos como:
    - npm run dev
    - npm run build
    - npm run typecheck
  - Se valida el resultado visual y técnico.
5. Corrección mediante prompt
  - Si hay errores o ajustes necesarios, se corrige únicamente modificando el prompt, nunca el código.
6. Avance a la siguiente fase
  - Una vez validado, se continúa con el siguiente objetivo.

## DIAGRAMA BASE:

**AI-Orchestrated Development Workflow (BEX Challenge)**



## REGISTRO DE PROMPTS UTILIZADOS

Esta sección documenta las interacciones clave con la Inteligencia Artificial durante el desarrollo del Dashboard Inmobiliario.

Cada prompt fue diseñado de forma consciente, con contexto claro, restricciones explícitas y un enfoque iterativo, permitiendo construir la aplicación sin edición manual de código.

### Intento 1 — Inicialización del Proyecto

#### Objetivo

Inicializar el proyecto base utilizando React Router v7, respetando el stack solicitado y limitando la intervención manual al uso de comandos de terminal.

#### Prompt utilizado

Guíame paso a paso usando únicamente comandos de terminal para inicializar un proyecto con React Router v7.

No generes código de componentes ni lógica de negocio.

Concéntrate únicamente en el setup y la estructura inicial del proyecto.

#### Resultado

Proyecto creado correctamente mediante `npm create react-router@latest`, con estructura base funcional y entorno de desarrollo operativo.

### Intento 2 — Definición de Arquitectura y Rutas

#### Objetivo

Definir una arquitectura clara de rutas y layouts antes de comenzar con la generación de componentes visuales.

#### Prompt utilizado

Diseña una arquitectura de rutas limpia y mantenible usando React Router v7.

Incluye un layout público para autenticación y un layout protegido para el dashboard con rutas anidadas.

Explica primero la estructura conceptual antes de generar cualquier código.

#### Resultado

Se definió una jerarquía de rutas clara, separando correctamente autenticación y dashboard, alineada con el enfoque de file-based routing.

### Intento 3 — Layout del Dashboard y Sidebar

#### Objetivo

Generar el layout principal del dashboard con un sidebar persistente y navegación funcional.

#### Prompt utilizado

Genera el layout principal del dashboard usando React Router v7, Tailwind CSS y shadcn/ui. El layout debe incluir un sidebar colapsable y un área de contenido para rutas anidadas. Aplica los cambios directamente sobre el archivo de layout existente. No me pidas copiar o pegar código manualmente.

#### Resultado

Layout del dashboard implementado correctamente, con sidebar funcional y navegación clara sin afectar el sistema de rutas.

### **Intento 4 — Corrección de Errores y Ajustes Estructurales**

#### Objetivo

Corregir errores de importación, referencias de rutas faltantes y problemas de compilación detectados durante el proceso.

#### Prompt utilizado

El proyecto presenta errores de importación y referencias de rutas faltantes durante el build. Analiza la configuración actual de rutas y componentes, y corrige únicamente lo necesario para que el proyecto pase correctamente npm run build y npm run typecheck. No introduces nuevas funcionalidades.

#### Resultado

Errores corregidos exitosamente, logrando una compilación limpia y validación completa de TypeScript.

### **Intento 5 — Tarjetas KPI del Dashboard**

#### Objetivo

Agregar indicadores visuales clave (KPIs) en la vista principal del dashboard.

#### Prompt utilizado

Ubica el archivo de la ruta principal del dashboard y mejóralo agregando una grilla responsiva de tarjetas KPI en la parte superior. Utiliza componentes Card de shadcn/ui y utilidades de Tailwind CSS. Incluye métricas para Ventas Totales, Unidades Disponibles y Visitas Web. Mantén intacto el layout y código existente.

#### Resultado

Se integraron tres tarjetas KPI visualmente claras, alineadas correctamente y sin afectar la estructura existente.

### **Intento 6 — Tabla de Proyectos Inmobiliarios**

#### Objetivo

Completar el reto visual agregando una tabla central con proyectos inmobiliarios ficticios.

#### Prompt utilizado

Modifica el archivo existente de la vista principal del dashboard para agregar una tabla de datos debajo de las tarjetas KPI.

Usa componentes Table y Badge de shaden/ui junto con Tailwind CSS.

Muestra cinco proyectos inmobiliarios ficticios con nombre, ubicación, estado y precio.

No elimines ni reescribas el código de los KPIs existentes.

Resultado

Tabla de proyectos renderizada correctamente debajo de los KPIs, cumpliendo con los requisitos visuales del challenge.

Intento Final — Validación y Cierre

Objetivo

Verificar que el resultado final cumple todas las reglas del challenge antes de la entrega.

Prompt utilizado

Revisa la implementación actual del dashboard y valida que cumpla con todos los requisitos del challenge.

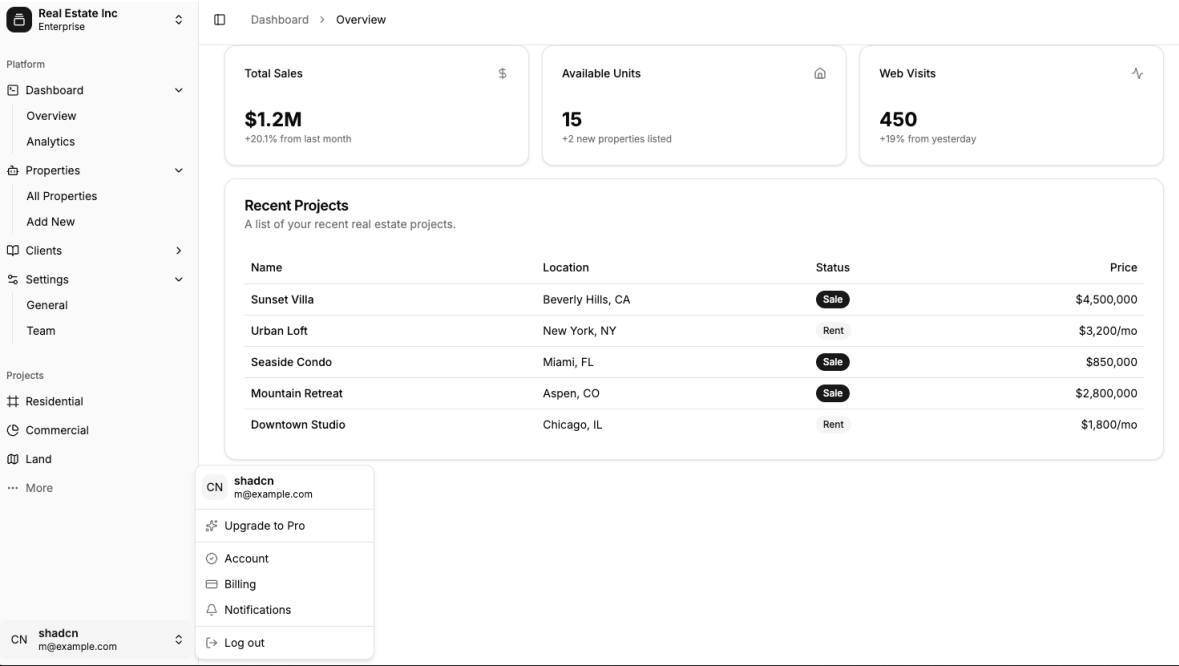
No agregues nuevas funcionalidades.

Enfócate únicamente en estabilidad, estructura y consistencia visual.

Resultado

Dashboard completo, estable y funcional, generado íntegramente mediante orquestación con IA y validado mediante build y typecheck.

CAPTURAS DE PANTALLA Y PRUEBAS:



Total Sales

\$

\$1.2M

+20.1% from last month

Available Units

15

+2 new properties listed

Web Visits

450

+19% from yesterday

Recent Projects

A list of your recent real estate projects.

Name	Location	Status	Price
Sunset Villa	Beverly Hills, CA	Sale	\$4,500,000
Urban Loft	New York, NY	Rent	\$3,200/mo
Seaside Condo	Miami, FL	Sale	\$850,000
Mountain Retreat	Aspen, CO	Sale	\$2,800,000
Downtown Studio	Chicago, IL	Rent	\$1,800/mo

Explorer

bex-tech-challenge

github

react-router

app

components

ui

app-sidebar.tsx

nav-main.tsx

nav-projects.tsx

nav-user.tsx

team-switcher.tsx

hooks

lib

routes

\_auth.login.tsx

\_auth.tsx

\_dashboard.\_index.tsx

\_dashboard.analytics.tsx

\_dashboard.clients.tsx

\_dashboard.properties.tsx

\_dashboard.tsx

welcome

app.css

root.tsx

routes.ts

build

node\_modules

public

.nojekyll

favicon.ico

.dockerignore

.gitignore

components.json

Outline

Timeline

nav-user.tsx

Walkthrough

package.json

.nojekyll

vite.config.ts

deploy.yml

Walkthrough

15 minutes ago

Review

- Added "Recent Projects" data table using `shadcn/ui` Table and Badge components.
- Fixed missing `/analytics` route by creating placeholder and registering in route config.

Verification

- Verified route configuration file `app/routes.ts` `build` - SUCCESS
- Verified type safety (`npm run typecheck`) and resolved missing modules.

Next Steps

- Run `npm run dev` to view the dashboard.
- Implement the actual Property Form and Table.

Problems

Output

Debug Console

Terminal

Ports

ccgarckmac@MacBook-Pro-de-carlos bex-tech-challenge % clear

ccgarckmac@MacBook-Pro-de-carlos bex-tech-challenge % npm run build

> build

> react-router build

vite v7.3.0 building client environment for production...

app/components/ui/separator.tsx (1:0): Error when using sourcemap for reporting an error: Can't resolve original location of error.

app/components/ui/label.tsx (1:0): Error when using sourcemap for reporting an error: Can't resolve original location of error.

app/components/nav-main.tsx (1:0): Error when using sourcemap for reporting an error: Can't resolve original location of error.

app/components/nav-projects.tsx (1:0): Error when using sourcemap for reporting an error: Can't resolve original location of error.

app/components/team-switcher.tsx (1:0): Error when using sourcemap for reporting an error: Can't resolve original location of error.

app/components/nav-user.tsx (1:0): Error when using sourcemap for reporting an error: Can't resolve original location of error.

app/components/ui/dropdown-menu.tsx (1:0): Error when using sourcemap for reporting an error: Can't resolve original location of error.

1862 modules transformed.

build/client/assets/root-BLV\_tIg5.css

3.69 kB | gzip: 0.58 kB

build/client/assets/\_auth-Ds0mtku4.js

54.22 kB | gzip: 9.79 kB

build/client/assets/\_dashboard.clients-C3PbTcca.js

8.78 kB | gzip: 0.22 kB

build/client/assets/\_dashboard.analytics-DfTv0BS9.js

Chat ... ip: 0.36 kB

ip: 0.36 kB