

Carlos Alfonso Hernández Pérez

Sábado 27 de dic. de 25

## 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.

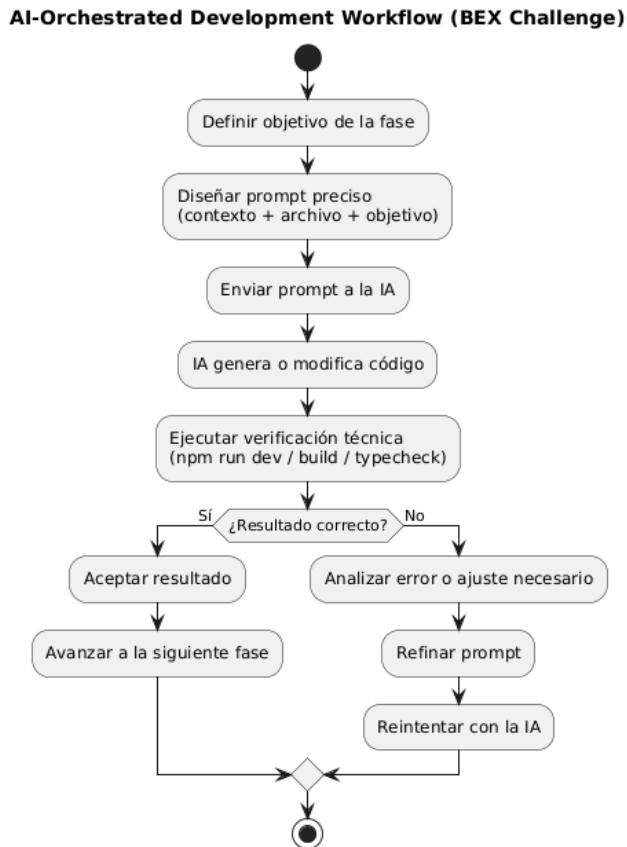
El flujo se basa en una iteración constante entre:

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:



## **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**

Iniciar 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 introduzcas 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 mejoralo 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 shadcn/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:

The screenshot shows a dashboard for 'Real Estate Inc' with a sidebar menu and a main content area. The sidebar includes sections for Platform (Dashboard, Overview, Analytics, Properties, Clients, Settings, General, Team), Projects (Residential, Commercial, Land), and More. The main content area has a header 'Dashboard > Overview'. It features three KPI cards: 'Total Sales' (\$1.2M, +20.1% from last month), 'Available Units' (15, +2 new properties listed), and 'Web Visits' (450, +19% from yesterday). Below these is a section titled 'Recent Projects' with a table of five properties:

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

The bottom right corner shows a user profile for 'shadcn m@example.com' with options to Upgrade to Pro, Account, Billing, Notifications, and Log out.

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

The screenshot shows a developer's environment with multiple windows open:

- Terminal:** Shows a build process for a Vite application. It starts with clearing the build directory and then running `npm run build`. The output shows errors related to sourcemap reporting for various components like `app/components/ui/sePARATOR.tsx` and `app/components/nav-user.tsx` due to unresolved original locations of errors.
- Browser:** A browser window titled "bex-tech-challenge — Walkthrough". It displays a "Walkthrough" page with a timestamp of "15 minutes ago". The page lists two items under "Recent Projects":
  - Added "Recent Projects" data table using `shadcn/ui` Table and Badge components.
  - Fixed missing `/analytics` route by creating placeholder and registering in route config.
 Below this is a "Verification" section with two bullet points:
  - Verified route configuration file `ts app/routes.ts .n build` - **SUCCESS**
  - Verified type safety (`npm run typecheck`) and resolved missing modules.
 Finally, there is a "Next Steps" section with two items:
  - Run `npm run dev` to view the dashboard.
  - Implement the actual Property Form and Table.
- File Explorer:** Shows the project structure for "bex-tech-challenge". It includes files like `.github`, `.react-router`, `app` (containing `components` and `ui`), `hooks`, `lib`, `routes` (with `_auth` and `_dashboard` subfolders), `welcome`, `build`, `node_modules`, `public` (with `.nojekyll`, `favicon.ico`, `.dockerignore`, `.gitignore`, and `components.json`), and `Outline` and `Timeline` sections.