

Prueba Técnica — Lead Web Frontend (Global66)

Objetivo

Evaluar la capacidad del candidato para:

1. **Diseñar en Figma** una interfaz fiel y escalable (auto-layout, componentes, variables/tokens).
 2. **Implementar en Nuxt** una página dinámica de precios con **SSR, SEO, performance y TailwindCSS**.
 3. Demostrar **criterio de growth** (SEO, medición, conversión).
-

Parte A — Diseño (Figma)

Entregable

Replicar **fielmente (imagen adjunta)** la interfaz del hero y secciones inmediatas de la captura (tipo “Valor del dólar hoy…”, banne, y bloque “Sé Global, paga como local”).

Requisitos de Figma

- Auto-layout
- Componentes + Variants
- Variables/Tokens
- Componentization

Entrega: Link a archivo Figma

Criterios: fidelidad visual, escalabilidad (componentes), orden, accesibilidad, limpieza.

Parte B — Frontend (Nuxt + TailwindCSS + SSR + SEO)

Stack

- **Nuxt 2**
- **SSR habilitado.**
- **TailwindCSS** para **todo** el styling (no CSS ad-hoc salvo utilidades mínimas).

- **Sin UI kit externo** (opcionalmente icons SVG inline).

Ruta y comportamiento

- La vista debe quedar en `/precio/*`
 - **Ejemplos válidos:**
 - `/precio/peso-chileno` (USD/CLP)
 - `/precio/sol-peruano` (USD/PEN)
- El wildcard debe mapear el **slug** a una **divisa/currency** con un **diccionario local** (p. ej. `peso-chileno` → CLP, `sol-peruano` → PEN, `dolares` → USD).

Datos y dinámica

- Implementar un **endpoint SSR local** para tasas:

Responder JSON simulado:

```
{
  "base": "USD",
  "rates": { "CLP": 987.62, "PEN": 3.81 },
  "asOf": "2025-10-31T17:51:00Z"
}
```

- Desde la página `/precio/*`, **leer el wildcard**, resolver la **divisa objetivo** y mostrar:
 - **Headline:** “Valor del dólar hoy”
 - **Conversión:** `1 USD = {monto} {divisa}` con formato local (ej: `987,62 CLP`).
 - **Leyenda:** “Tipo de cambio para {fecha} a las {hora} UTC”.
- **SSR SEO:**
 - `<title>` y `<meta name="description">` dinámicos según la divisa.
 - `link rel="canonical"` y `hreflang` mínimos (al menos `es-CL`).

TailwindCSS

- **Obligatorio:** usar utilidades para layout/spacing/typography; configurar `tailwind.config.js` con **tokens** que reflejen los de Figma
- **Responsive** con breakpoints tailwind por defecto (`sm`, `md`, `lg`, `xl`).

- No CSS inline; crear (si hace falta) **component classes** con `@apply`.

Extras (opcionales, suman puntos)

- **i18n** de títulos/leyendas.
- Lighthouse ≥ 90 en Performance y SEO en la ruta
- Estructura limpia del entregable y uso de buenas prácticas como DRY y CLEAN

Entrega: repo con el código + deploy local reproducible.

Parte C — Backend/API mínima

- Endpoints requeridos:
 - Endpoint **GET**: ya descrito en **Parte B**.
 - Endpoint **POST**: que se reciba un campo “name” y “email” y lo guarde en un Google Spreadsheet
- Buscar tasas en API externa o Mockeada.
- Validar query **base** y **target** opcionales (ej. `/api/rates?base=USD&target=CLP`).
- Manejo de errores coherente (400/404/500).
- Log simple de requests (consola).

Parte D — Growth Thinking (máx. 1 página)

1. ¿Qué **metas técnicas y de negocio** monitorearías en `/precio/*`? (ej. CTR al CTA, scroll depth, tiempo en página, SEO por divisa).
2. Caída del **20% orgánico** en “precio dólar” para CLP: ¿cómo la investigarías y qué acciones tomarías en 72h?
3. 3 mejoras de **velocidad** aplicables a esta página sin sacrificar SEO.
4. 2 hipótesis para **mejorar conversión** del hero (y cómo A/B testearlas).
5. ¿Cuánto tiempo te tomó realizar este requerimiento y qué aceleraste con IA?

Entregables finales

1. **Figma**: link
2. **Repositorios**: deben contener README claro de como ejecutarlo y requerimientos (Ej: pasos, versión de node)

- a. Directorio Front: Nuxt + Tailwind
 - b. Directorio Back: API local con Express y link al Spreadsheet (Público)
3. **Documento Growth** (PDF/MD) con respuestas de la Parte D.