

peticion:

quiero desarrollar una aplicacion web inspirada en un juego de mesa "ComoSERnos":

suegra como un proyecto familiar. mi tia hace coaching entonces desarrolló mazos de cartas a modo de juego de mesa pensado para jugar en grupo. por ejemplo en familia, despues de almorzar logica del juego de mesa: un integrante levanta una carta, la lee, esta carta contiene una consigna o una pregunta reflexiva que invita a reflexionar, conectar, conocerse, ser (por eso el nombre del juego) una vez cumplida la consigna, se deja la carta aparte (o tambien se la podría quedar el jugador, o quien la gano) y se pasa al siguiente integrante quien vuelve a levantar otra carta del mazo

cartas de ConoSERnos primera version (ya que despues vienen ReconoSERnos y ReconoSERse):

A lo que soy LEAL en mi vida es a...,1
Cuando algo no me SALE....2
Cuando dejo mi mente en blanco lo primero que APARECE es...,3
Cuando era chiquito/a me gustaba JUGAR con...,4
Cuando estoy TRISTE lo que hago es...,5
Cuando me ENAMORO soy...,6
Cuando me EQUIVOCO con alguien que me importa me siento...,7
Cuando me IGNORAN me siento...,8
Cuando me PIDEN algo que no quiero hacer...,9
Cuando me SIENTO solo/a...,10
Cuando mi SEXTO SENTIDO me dice algo...,11
Cuando MIRO a los ojos me siento...,12
Cuando no PUEDO hacer algo que quiero me siento...,13
Cuando no TENGO un plan me siento...,14
Cuando sea viejito/a me GUSTARÍA...,15
Cuando QUIERO decir lo que siento...,16
El código que jamas ROMPERIA es...,17
El gran ACIERTO de mi vida fue...,18
El GRAN momento de mi vida fue...,19
El PEOR viaje de mi vida fue...,20
En lo que hoy NO CREO ni un poco es en...,21
En lo que me gusta AYUDAR a los demás es en...,22
En lo que nunca PIERDO la esperanza es en...,23
En lo que soy COLGADO/A es...,24
En lo que soy INTOLERANTE es...,25
La mayor LOCURA que hice por amor fue...,26
La mejor película que VI en mi vida es...,27
La noche es IDEAL para...,28
La palabra que menos me REPRESENTA es...,29
La PEOR mentira que dije fue...,30
La persona con que me gustaría REENCONTRARME es...,31
La persona que es un FARO en mi vida es...,32
La persona que mas me hace REIR es...,33
Lo que aun me falta CONCRETAR en mi vida es...,34
Lo que CAMBIARIA de mi es...,35
Lo que DICEN los demás sobre mí...,36
Lo que el DESTINO tiene para mí...,37
Lo que GUIA mi vida es...,38
Lo que hoy ELIJO darle al mundo es...,39
Lo que hoy me cuesta CREER...,40
Lo que hoy me ENAMORA es...,41
Lo que hoy quiero GRITAR es...,42
Lo que jamas COMERÍA es...,43
Lo que jamas SERÉ es...,44
Lo que mas me CUESTA hacer es...,45
Lo que mas me IRRITA es...,46
Lo que mas me REPRESENTA es...,47
Lo que me da CONFIANZA de las personas es...,48
Lo que me da LIBERTAD es...,49
Lo que me da ORGULLO de las personas es...,50
Lo que me da TERROR es...,51
Lo que me da TRISTEZA es...,52
Lo que me FRUSTRA es...,53
Lo que me gustaría CONQUISTAR es...,54
Lo que me gustaría ENSEÑAR es...,55
Lo que me levanta el ANIMO es...,56
Lo que me DEJA sin palabras es...,57
Lo que me pone NERVIOSO/A es...,58
Lo que me saca una SONRISA cada dia es...,59
Lo que menos VALORO de mi es...,60
Lo que NECESITO en este momento es...,61
Lo que NO ME ACUERDO nunca es...,62
Lo que nunca RESIGNARIA es...,63
Lo que nunca voy a ENTENDER es...,64
Lo que PAPA NOEL nunca me trajo fue...,65
Lo que peor me PONE es...,66
Lo que quiero OLVIDAR es...,67
Lo que SUFRIO mucho es...,68
Los días de LLUVIA me...,69
Los TRES DESEOS que hoy pediría son...,70
Mi DIA IDEAL sería en...y con...,71
Mi MAMÁ en mi vida es...,72
Mi mayor APRENDIZAJE de niño/a fue...,73
Mi mayor APRENDIZAJE en la vida fue...,74
Mi mayor DECEPCION fue...,75
Mi mayor DESCUBRIMIENTO fue...,76
Mi mayor DESAFIO es...,77
Mi mejor EDAD hasta ahora fue...,78
Mi meta a CORTO PLAZO es...,79
Mi meta a LARGO PLAZO es...,80
Mi PAPÁ en mi vida es...,81
Mi PENSAMIENTO recurrente es...,82
Mi PREOCUPACIÓN recurrente es...,83
Mi PEOR defecto es...,84
Mi PEOR momento fue...,85
Mi PRIMER AMOR fue...,86
Mi PRIMER PASO para cambiar algo es...,87
Mi PRIORIDAD hoy es...,88
Mis AMIGOS en mi vida son...,89
Mis mejores VACACIONES de niño/a fueron...,90
Si fuera INVISIBLE lo que haría es...,91
Si me ENOJO soy como...,92
Si pudiera CAMBIAR MI NOMBRE me llamaría...,93
Si pudiera VIAJAR EN EL TIEMPO iría a...,94

Si tengo que DEFINIRME con una palabra es....95
Si tengo que elegir el "TITULO DEL LIBRO DE MI VIDA" sería....96
Si pudiera VOLVER a una etapa de mi vida sería a....97
Si tuviera SUPERPODERES me gustaría....98
Tu ALMA GEMELA en esta vida es....99

aplicacion:

la idea es llevar ese juego de mesa a un dispositivo virtual, seguramente se jugaria en teléfonos, pensamos en monetizarlo, por ejemplo: tenes 5 cartas gratuitas por dia y luego cargas credito a traves de mercadopago para obtener mas tarjetas, desarollar incentivos, por ejemplo mientras mas jugas a "ConoSERnos" cada tantas tarjetas compradas obtenes "tarjetas unicas coleccionables" de esa version del juego (tal vez nfts a evaluar)

modo de juego en la aplicacion:
click en el mazo de cartas, se da vuelta una carta, te deja verla durante cierto tiempo, te deja reflexionar, no te permite hacer capturas de pantalla, luego se sugiere pasar al siguiente integrante y se repite
Se me ocurre que ya que es muy facil simplemente anotarse los contenidos de las cartas y luego entonces para que pagar si se puede anotar igual, esta aplicacion podria dar el valor agregado de "arrancar tranqui e ir subiendo la intensidad o la profundidad" esto se podria lograr ponderando las cartas en nivel de profundidad y cuando se empieza una sesion ir de menos a mas aleatoriamente (ya me doy cuenta que hay que hacer muchas mas que 100 cartas asi que despues te las voy a pedir seguramente mas adelante)

sugiereme como hacer la ui/ux, habia pensado en que me des opciones y luego cuando confirmemos como hacerlo te puedo pedir mas adelante un prompt para que google antigravity me ayude con estas tareas, tal vez la pagina principal sea el mazo de cartas y te deje levantar 2 o 3 cartas para luego pedirete registrarte? entonces te da algunas cartas mas y luego te pide cargar dinero? cuantas secciones/botones debe tener la interfaz? mazo de cartas como pantalla principal? luego mi cuenta, mis cartas, cargar credito, cambiar juego, botones de salir, eliminar cuenta, etc. sugiereme lo mejor

las cartas de cada una de las versiones del juego y las cartas unicas en caso de hacerlas, las quiero cargar por csv o similar en la propia api, no quiero un usuario "manager" que administre estas cosas por el momento.

el tema de la administracion de los usuarios sugereme lo mejor. (por si surge algun registro erroneo, tal vez quiera cargarme creditos arbitrariamente o algo) pero me imagino mucho mejor hacer ese tipo de gestiones con codigo yo mismo que integrarlas a la interfaz grafica para que lo haga un manager por el momento (para mantener la simpleza de la app)

los usuarios deberian registrarse con sus datos basicos, validar su email y tal vez su genero / edad / region. (o lo que vos sugieras)

frontend: recomiendame el stack mas conveniente (si fuese una aplicacion android/ios, es muy dificil de llevarla al appstore? no me interesa renegar por el momento. en ese caso conviene que sea una aplicacion web apuntada a mobile y acceder a ella por medio de una url? tal vez de hecho se me ocurre que hoy en dia a los navegadores les encanta empaquetar este tipo de aplicaciones y luego hasta te aparecen con un icono en el telefono como si fueran una app descargada desde el appstore)

backend: node con mongo creo, (recomiendame lo mejor)

en todo lo anterior asesorame, recomendame, decime si estan bien o no esas ideas, sugerime otras, busca el mejor resultado. quiero que funcione, no solo hacerlo y que quede obsoleto
obviamente quiero hacer todo con buenas practicas, seguridad, sesion, validacion, etc. que sea junior no quiere decir que quiera hacer un producto de mala calidad

respuesta:

¡Excelente iniciativa! Convertir un juego de mesa con un concepto tan profundo en una aplicación digital móvil-first es un proyecto con gran potencial. Es una idea que capitaliza el crecimiento de las apps de *wellness* y conexión social.

Tu enfoque en **robustez, seguridad y buenas prácticas** es el correcto. A continuación, te presento una estrategia completa de UX/UI y el stack técnico recomendado para garantizar que la aplicación sea escalable y moderna.



Estrategia UX/UI y Flujo de Usuario

Tu idea de la **intensidad progresiva** de las cartas es el **Diferenciador Principal (USP)**. La interfaz debe reflejar esa simplicidad, conexión y profundidad.

1. Modelo de Interfaz (Frontend)

- **Recomendación:** Desarrolla una Aplicación Web Progresiva (PWA).
 - **Ventajas:** Acceso directo por URL, funciona en navegadores, tiene ícono y apariencia de *app nativa* en el teléfono, evitas las revisiones y comisiones de App Store/Google Play. Es el camino más rápido y eficiente.

- **Diseño:** Mobile-First con un diseño limpio, colores suaves (inspirado en tu `dorsotarjeta.jpg`) y tipografía clara.

2. Flujo de Usuario y Pantallas Clave

Pantalla	Propósito	Elementos Clave
1. Landing / Invitado	Enganche inicial.	Un gran mazo central con el logo (<code>ConoSERnos</code>). Botón: " Jugar Gratis " (o "Empezar Sesión").
2. Sesión / Turno	El corazón del juego.	Mazo Principal (central): animaciones de volteo. Botón: "Pedir Carta". Indicador: Turno actual/Jugador (Opcional si es multijugador).
3. La Carta (Modal / Pantalla Completa)	Reflexión y enfoque.	Contenido de la Carta: Letra grande, fondo tenue. Botón: "Listo" / "Compartir Reflexión" / "Siguiente Jugador". Animación: <i>Efecto humo/neblina</i> al revelar para dar sensación de misterio y profundidad.
4. Menú de Usuario (Sidebar)	Navegación secundaria.	Icono de hamburguesa (o pestaña inferior) que lleva a las secciones de monetización y gestión.

3. Secciones y Botones Clave (Menú Principal)

- **Mazo / Jugar (Pantalla Principal):** Siempre debe ser el Home.
- **Mi Cuenta / Perfil:**
 - Datos de registro (email, género, etc.).
 - Ajustes de la app.
 - Botón "Eliminar Cuenta" (obligatorio por regulaciones).
- **Tienda / Cargar Crédito:**
 - Packs de cartas (ej: 10 cartas, 50 cartas, Ilimitado por 1 día).
 - Integración con Mercado Pago.
- **Mi Colección / Tarjetas Únicas:**
 - Galería visual de las cartas coleccionables obtenidas.
- **Cambiar Juego:**
 - Selector para futuras versiones (ReconoSERnos, ReconoSERse).

4. Sugerencia para la Intensidad Progresiva

Para aplicar el valor agregado, la app debe gestionar las sesiones:

- **Sesión:** Una sesión puede ser definida por un número de cartas (ej: 10 cartas).
- **Ponderación de Cartas:** Clasifica las 100 cartas (y las futuras) con un nivel de profundidad (1 al 5).
- **Lógica de Sorteo:**
 1. Empieza sorteando solo cartas de Nivel 1 y 2.
 2. A medida que avanza la sesión, aumenta la probabilidad de que salga una carta de Nivel 3 y 4.
 3. Las cartas de Nivel 5 (las más profundas) solo salen al final de la sesión.

 Stack Tecnológico Recomendado		
Tu sugerencia de Node.js y Mongo es buena. Para asegurar buenas prácticas, seguridad y escalabilidad , te recomiendo la siguiente arquitectura:		
Frontend (Web App Progresiva)		
Componente	Opción Recomendada	Justificación
Framework	React (o Next.js para mayor estructura)	Líder del mercado, excelente para PWAs y desarrollo móvil-first. Amplia comunidad y fácil de mantener.
Estilos	Tailwind CSS o Styled Components	Modernidad y eficiencia. Permite aplicar estilos siguiendo buenas prácticas sin renegar con CSS nativo.

Backend (API Escalable y Segura)		
Componente	Opción Recomendada	Justificación

Framework	Node.js con NestJS (usando TypeScript)	MUY RECOMENDADO. NestJS fuerza una arquitectura limpia (tipo Angular) con módulos, inyección de dependencias y servicios. Esto evita el "código espagueti" clásico de Node junior y te prepara para la escalabilidad. TypeScript asegura la calidad y tipado del código.
Base de Datos	MongoDB	Excelente para el modelo de datos flexible de las cartas, usuarios, colecciónables y estadísticas de juego. Es rápido y se integra perfectamente con Node.js/NestJS (Mongoose).
Autenticación	JWT (JSON Web Tokens) + Bcrypt	Estándar de la industria. JWT para la sesión y Bcrypt para el <i>hash</i> seguro de las contraseñas.
Pagos	Mercado Pago API	Integración directa con el sistema de pagos que ya tienes pensado. Deberás crear un Módulo de Pagos dedicado en tu backend de NestJS.

🔑 Gestión de Datos y Administración (Tu Solicitud)

Tu idea de **evitar un usuario "manager" en la interfaz** para mantener la simpleza es brillante y ahorra mucho tiempo en el MVP.

1. Carga de Cartas (Data Modeling)

- **Carga Inicial:** Perfecto. Utiliza un script de `seed` o una función específica en tu API para cargar el CSV (o un archivo JSON mejor tipado) a MongoDB la primera vez.
- **Estructura de la Carta (Modelo de Datos):**

```
{
  "id": "17",
  "texto": "El código que jamás ROMPERIA es...",
  "version": "ConoSERnos_v1",
  "profundidad": 4, // Nivel de 1 a 5 (para la lógica progresiva)
```

```
"es_coleccionable": false,  
"categoria": "Ética/Valores"  
}
```



2. Administración de Usuarios/Créditos

- **Acción Recomendada:** Gestiona estas tareas directamente a través de una **consola de base de datos** (ej: MongoDB Compass, o la terminal) o mediante **endpoints de administrador seguros** en tu API.
- **Ejemplo de Gestión con Código:**
 - Tendrías un endpoint `POST /admin/credits` que solo funciona si tienes un Token JWT de super-administrador (que solo usarías tú).
 - Este endpoint podría tener la lógica para `cargarme creditos arbitrariamente` o anular un registro. Es más seguro y rápido de implementar que una interfaz completa.

3. Registro de Usuarios (Datos y Validación)

- **Datos Esenciales:** `email` (único, obligatorio), `password`, `fecha_registro`
- **Validación:** La validación de `email` es crucial para evitar bots y asegurar la cuenta.
- **Datos Demográficos (Sugerencia):** Es excelente que los pidas para el *coaching* futuro. Hazlos **opcionales** en el registro inicial para no crear fricción:
 - `Genero`
 - `Rango de Edad` (ej: 18-25, 26-40, 40+)
 - `Region/País` (útil para la moneda de Mercado Pago).

 **Próximos Pasos (Usando IA)**

Ahora que tienes la estructura y el stack definidos, puedes usar la IA para acelerar. Tu idea de usar **Google Antigravity** para las tareas de desarrollo es muy viable, ya que está diseñado para **generar código y automatizar tareas de backend y frontend**.

Próxima Pregunta Sugerida:

"Quiero usar Google Antigravity para empezar el proyecto 'ConoSERnos' con NestJS y MongoDB. Dame un prompt detallado para la IA que me genere la estructura básica del backend, el modelo de datos de 'Cartas' y el módulo de autenticación con JWT, siguiendo las buenas prácticas."

petición:

sí perfecto, adelante con esto:

"Quiero usar Google Antigravity para empezar el proyecto 'ConoSERnos' con NestJS y MongoDB. Dame un prompt detallado para la IA que me genere la estructura básica del backend, el modelo de datos de 'Cartas' y el módulo de autenticación con JWT, siguiendo las buenas prácticas."

pero ademas quiero que:

- seas bien extenso y específico en los prompts
- me guíes paso a paso en la próxima etapa
- me enseñes a usar antigravity ya que es la primera vez que me dispongo a hacerlo
- me digas todo lo que necesito para avanzar con el proyecto (por ejemplo puedo usar mi propia cuenta de mercadopago para que el dinero ingrese ahí o debo cambiar a modo negocio o algo de eso?)

haceme las preguntas que necesites en caso de ser necesario para comprender mejor lo que se te solicita y espera las respuestas en caso de necesitarlo antes de avanzar

este prompt esta desordenado pero completo. reordena vos las prioridades

respuesta:

¡Me parece genial tu enfoque! Estás buscando construir algo profesional desde el día uno, y esa es la mentalidad correcta para que el proyecto escale.

He reordenado tus prioridades para que el flujo de trabajo sea lógico. Antes de darte los prompts (que ya los tengo listos y bien detallados), necesito aclarar un par de puntos críticos sobre **Mercado Pago** y **Google Antigravity** para que no te trabes al empezar.

Aquí tienes el plan de acción paso a paso.

🛑 Paso 0: Prerrequisitos y Preguntas (Responde esto antes de avanzar)

Para darte la guía final exacta, necesito que me aclares:

1. **Acceso a Google Antigravity:** ¿Ya tienes acceso confirmado y el IDE instalado? (Es un producto muy nuevo). Si aún no lo tienes, ¿estás usando **Project IDX** (el entorno cloud de Google) o **VS Code con Gemini Code Assist**? Los prompts sirven igual, pero la interfaz cambia un poco.
2. **Mercado Pago:**
 - **Tienes cuenta personal o de empresa?** Para empezar a probar en modo "Sandbox" (desarrollo), tu cuenta personal sirve.
 - **Para salir a producción (cobrar de verdad):** Te recomiendo fuertemente convertir tu cuenta a **Cuenta Vendedor / Profesional** (es gratis el cambio). Esto te permite acceder a las credenciales de desarrollador ([Access Token](#) y [Public Key](#)) y separa tus gastos personales de los ingresos de la app (fundamental para impuestos/AFIP).
 - **Lo que necesitas ya:** Entrar a [developers.mercadopago.com.ar](#), loguearte y crear una "Aplicación" nueva para obtener las credenciales de prueba.

🎓 Paso 1: Mini-Tutorial de Google Antigravity (Cómo usarlo)

Como es tu primera vez, olvida cómo usas ChatGPT. En Antigravity no solo "chateas", sino que **gestionas agentes**.

1. **EI "Agent Manager" (Tu Centro de Mando):** No escribas código directo. Abre el panel de agentes (suele estar a la izquierda o con un atajo tipo `Cmd/Ctrl + I` o `Ctrl + K` dependiendo la config).
2. **Contexto es Rey:** Antigravity ve todos tus archivos. No hace falta que copies y pegues código viejo. Solo dile: "En el archivo `auth.service.ts`...".
3. **Iteración:**
 - Tú das el prompt (la orden).
 - El Agente te muestra un "Diff" (lo que va a cambiar en verde/rojo).
 - **TÚ debes revisar y dar "Accept" o "Apply".** Nunca aceptes a ciegas.
4. **Terminal Integrada:** Antigravity puede ejecutar comandos de terminal. Si le dices "Instala las dependencias", él puede correr `npm install` por ti.

🏛️ Paso 2: Los Prompts Maestros (Copia y Pega)

Aquí tienes la estrategia de prompts dividida en módulos. Copia esto tal cual en el chat del agente de Antigravity.

Prompt 1: Configuración Inicial del Proyecto (Scaffolding)

Objetivo: Crear la base sólida con NestJS, seguridad y configuración de entorno.

Plaintext

Actúa como un Arquitecto de Software Senior experto en NestJS y TypeScript.

Quiero inicializar el backend para el proyecto "ConoSERnos" siguiendo estrictamente Clean Architecture y SOLID principles.

CONTEXTO:
Es una API para un juego de cartas digital. Usaremos MongoDB (Mongoose) y autenticación JWT.

TAREAS:

1. Genera un nuevo proyecto NestJS si no existe.
2. Configura la estructura de carpetas para ser modular (modules, controllers, services, schemas, dtos, interfaces).
3. Instala y configura las siguientes dependencias clave:
 - @nestjs/mongoose mongoose (para la DB)
 - @nestjs/config (para variables de entorno .env)
 - class-validator class-transformer (para validación de DTOs)
 - helmet (para seguridad de cabeceras HTTP)
4. Crea un archivo '.env.example' con las variables: PORT, MONGODB_URI, JWT_SECRET, MERCADOPAGO_ACCESS_TOKEN.
5. Configura el 'app.module.ts' para cargar la base de datos de forma asíncrona usando ConfigService.
6. Habilita CORS y un Global ValidationPipe en 'main.ts'.

Por favor, genera el código y los comandos de instalación necesarios.

Prompt 2: Módulo de Usuarios y Autenticación (Auth)

Objetivo: Seguridad robusta sin sobrediseñar.

Plaintext

Ahora vamos a implementar el módulo de Auth y Users.

Requisitos técnicos:

1. Crea un UserSchema (Mongoose) con los campos:
 - email (String, unique, required, trim, lowercase)
 - password (String, required, select: false para seguridad)
 - fullName (String, required)
 - roles (Array de Strings, default ['user'])
 - credits (Number, default: 5) -> Son las cartas gratuitas iniciales.
 - createdAt (Date).
2. Implementa 'AuthService' con métodos para:
 - 'register': Debe hashear la contraseña usando 'bcrypt'.
 - 'login': Debe validar contraseña y retornar un JWT firmado.
 - 'validateUser': Para la estrategia de Passport.
3. Implementa la estrategia JWT usando '@nestjs/passport' y 'passport-jwt'.
4. Crea un decorador personalizado @CurrentUser() para obtener el usuario desde el request.
5. Protege las rutas usando un 'JwtAuthGuard'.

Genera los archivos del módulo (Controller, Service, Schema, Strategy, Guard).

Prompt 3: Modelo de Cartas y Lógica de Negocio (Core)

Objetivo: La lógica del juego y la carga de datos CSV.

Plaintext

Vamos a crear el corazón del juego: el módulo 'Cards'.

1. Schema de Mongoose ('Card'):
Debe tener:
 - content (String, requerido): El texto de la pregunta/consigna.
 - category (String): Ej: "Introspección", "División".
 - depthLevel (Number, 1-5): Para la lógica de intensidad progresiva.
 - version (String): Ej: "ConoSERnos_v1".
 - isCollectible (Boolean, default false).
2. Seed Script (Semilla de datos):
Crea un servicio o script 'CardSeederService' que permita leer un archivo CSV (simulado o real) y poblar la base de datos si está vacía.
Incluye un método para cargar estas 3 cartas de ejemplo para probar:
 - "Cuando estoy TRISTE lo que hago es..." (Nivel 2)
 - "La peor mentira que dije fue..." (Nivel 4)
 - "Mi mayor miedo es..." (Nivel 5)
3. CardsService - Método 'drawCard(user)':
Implementa la lógica de "Intensidad Progresiva".
 - Debo recibir el usuario.
 - Si es el inicio de la sesión, priorizar cartas de 'depthLevel' 1 o 2.
 - Usar 'aggregate' de Mongoose con '\$sample' para obtener una carta aleatoria filtrada por nivel.
 - Descontar 1 crédito al usuario tras entregar la carta (usar transacción o update atómico).

Genera el código completo para este módulo.

Paso 3: ¿Cómo avanzar después de tirar los prompts?

Una vez que Antigravity genere el código:

1. **Prueba de Humo:** Corre `pm run start:dev`. Si levanta en el puerto 3000 (o el que definas) y conecta a Mongo, ¡éxito!
2. **Base de Datos Local:** Necesitas tener MongoDB corriendo en tu PC o usar **Mongo Atlas** (gratis en la nube, recomendado para que no instales nada local). Si usas Atlas, pon la URL en el `.env`.
3. **Carga las Cartas:** Pídele a Antigravity que cree un endpoint temporal `POST /cards/seed` para disparar la carga de tus 100 cartas desde el JSON/CSV que ya tienes.

¿Te parece bien este plan de batalla? ¿Tienes alguna duda sobre la cuenta de Mercado Pago o el acceso al IDE antes de que empieces a ejecutar?

petición:

Paso 0: Prerrequisitos y Preguntas (Responde esto antes de avanzar)
Para darte la guía final exacta, necesito que me aclares:

Acceso a Google Antigravity: ¿Ya tienes acceso confirmado y el IDE instalado? SI LO TENGO
Mercado Pago:
¿Tienes cuenta personal o de empresa? Para empezar a probar en modo "Sandbox" (desarrollo), tu cuenta personal sirve. TENGO CUENTA PERSONAL

Para salir a producción (cobrar de verdad): Te recomiendo fuertemente convertir tu cuenta a Cuenta Vendedor / Profesional (es gratis el cambio). Esto te permite acceder a las credenciales de desarrollador (Access Token y Public Key) y separa tus gastos personales de los ingresos de la app (fundamental para impuestos/AFIP).
Lo que necesitas ya: Entrar a developers.mercadopago.com.ar, loguearte y crear una "Aplicación" nueva para obtener las credenciales de prueba. NO SE COMO SE HACE Y QUE NECESITO (SE BREVE)

respuesta:
¡Excelente! Al tener **Google Antigravity** listo y tu **cuenta personal** de Mercado Pago, ya tienes todo lo necesario para empezar. No necesitas ser empresa para obtener las credenciales de prueba (Sandbox).

Aquí tienes el paso a paso **breve** para obtener tus claves de Mercado Pago:

⚡ Cómo obtener tus Credenciales de Prueba (Sandbox)

1. **Entra:** Ve a developers.mercadopago.com.ar e inicia sesión con tu cuenta normal.
2. **Panel:** Haz clic en "**Tus integraciones**" (arriba a la derecha) o ve al "**Dashboard**".
3. **Crear:** Presiona el botón grande "**Crear aplicación**".
4. **Formulario Rápido:**
 - o **Nombre:** Ponle `monoSERnos Test`.
 - o **Tipo de solución:** Elige "**Pagos Online**" (o Checkout API).
 - o **¿Estás usando una plataforma e-commerce?:** Pon "**No**".
 - o **Producto:** Puedes poner "Otros" o dejarlo genérico.
 - o Acepta los términos y crea la app.
5. **Copiar Claves:**
 - o Entra a la aplicación que acabas de crear.
 - o Busca la sección "**Credenciales de prueba**" (Test credentials).
 - o Copia el **Access Token** (empieza con `TEST-`). **Ese es el que necesitas para tu código.**

✓ Luz Verde: ¡A Programar!

Ya tienes el dato que te faltaba (`MERCADOPAGO_ACCESS_TOKEN`).

Ahora sí, abre **Google Antigravity** y empieza a copiar y pegar los **Prompts Maestros** que te pasé en la respuesta anterior (Paso 2).

Orden sugerido:

1. Pega el **Prompt 1** (Configuración Inicial). Cuando te pida crear el `.env`, pega ahí el Access Token que acabas de copiar de Mercado Pago.
2. Sigue con el **Prompt 2** (Usuarios/Auth).
3. Termina con el **Prompt 3** (Lógica de Cartas).

¡Éxito con ese primer deploy local! Cuéntame si Antigravity te generó bien la estructura.