

Inteligencia Artificial Generación de Prompts

“ORGANIZADOR DE VIAJES CORTOS Y ESCAPADAS”



**Comisión 84180
CENTURIÓN, Lucas Raúl**

2025

Pre-entrega 2 — Fast Prompting en Acción

Alumno: Lucas Centurión

Introducción

Este proyecto tiene como objetivo aplicar la técnica de **Fast Prompting** para el diseño de un sistema de planificación de escapadas de 2–3 días.

El problema planteado es la dificultad que enfrentan muchos usuarios al organizar viajes cortos de forma rápida y optimizada, teniendo en cuenta parámetros como destino, días disponibles, presupuesto y actividades de interés.

La propuesta de solución es implementar una serie de prompts progresivos que evolucionan desde un esquema básico (Prompt 1) hasta una versión optimizada y modular (Prompt 5 Lite). Cada prompt agrega mejoras técnicas y metodológicas que permiten:

- Estructurar la salida en JSON para facilitar la reutilización.
- Incluir supuestos, criterios de planificación y alertas.
- Incorporar medición de costos en tokens.
- Generar materiales visuales (mapa y flyer) de manera automática.

La viabilidad está garantizada ya que el proyecto utiliza herramientas accesibles (Python, librerías oficiales de APIs) y modelos disponibles en la nube, con costos bajos de uso medidos en tokens.

Objetivos

- Aplicar la metodología de **Fast Prompting** mostrando la evolución en 6 etapas (Prompt 1 a Prompt 5 Lite).
 - Desarrollar un sistema que planifique viajes cortos en función de parámetros precargados.
 - Implementar validaciones progresivas para asegurar la calidad de los resultados.
 - Optimizar el consumo de tokens mediante modularización del código.
 - Generar salidas visuales (mapa y flyer) adaptadas al destino.
-

Metodología

El desarrollo se realizó en Python, organizando el flujo en distintos **prompts incrementales**.

Las técnicas aplicadas incluyen:

- **Zero-shot prompting** → en el Prompt 1, salida libre sin estructura.
 - **Instruction prompting** → en el Prompt 2, con más parámetros y guía estructural.
 - **Structured prompting con JSON** → en el Prompt 3, para facilitar parsing y validación.
 - **Role assignment y normalización** → en el Prompt 4, con system prompt y generación de imágenes.
 - **Chain-of-prompts** → en el Prompt 5, separando intake y generación de itinerario.
 - **Cost-aware prompting** → en el Prompt 5 Lite, versión liviana optimizada en consumo de tokens.
-



Herramientas y Tecnologías

- **Lenguaje:** Python 3
- **Librerías:**
 - `openai` → modelos GPT (gpt-4o-mini).
 - `google.genai` → Gemini 2.5 Flash Image para generación de imágenes.
 - `dotenv` → manejo de claves API.
 - `Pillow` → manejo de imágenes PNG.
 - `json` → validación y parsing de salidas estructuradas.
- **Metodología aplicada:** Fast Prompting (iteración y optimización de prompts).
- **Entorno de trabajo:** Jupyter Notebook.

```
In [1]: from openai import OpenAI
import os
from dotenv import load_dotenv

# =====
# Configuración API
# =====
load_dotenv()
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
from IPython.display import display
```

Prompt 1 — Base (mínimo viable)

Descripción técnica

Genera un itinerario simple para un destino y cantidad de días usando un único prompt natural. No hay estructura JSON ni validaciones: el modelo responde en texto libre.

Tecnología aplicada

- Python + `dotenv` para credenciales.
- `openai` Chat Completions con el modelo `gpt-4o-mini`.

Metodología

- *Zero-shot prompting*: instrucciones breves sin formato rígido.
- Salida directa a consola.

Notas

Este es el punto de partida: sin manejo de errores, sin costos/tokenes, sin helpers, sin imágenes.

```
In [2]: # =====
# Datos precargados
# =====
destino = "Mendoza"
cant_dias = 3

# =====
# Prompt 1 - Básico
# =====
prompt1 = f"""
Organizá un viaje de {cant_dias} días a {destino}.
Incluí actividades, comidas y presupuesto.
"""

response1 = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": prompt1}],
    temperature=0.7
)

print("==> Prompt 1 - Respuesta ==>")
print(response1.choices[0].message.content)
```

==== Prompt 1 – Respuesta ===

¡Claro! Aquí tienes una propuesta para un viaje de 3 días a Mendoza, Argentina. Incluye actividades, comidas y un presupuesto estimado.

Día 1: Llegada a Mendoza

Mañana:

- **Llegada a Mendoza**: Si llegas en avión, puedes tomar un taxi o un transfer a tu hotel. Si llegas en autobús, también hay opciones de taxis.
- **Alojamiento**: Reserva un hotel o una hostería en el centro de Mendoza. Presupuesto: \$100 - \$150 por noche.

Tarde:

- **Almuerzo en un restaurante local**: Prueba el asado argentino en un lugar como "El Patio de José" o "La Marchigiana". Presupuesto: \$20 - \$30 por persona.
- **Visita a la Plaza Independencia**: Pasea por la plaza y disfruta de la arquitectura local.

Noche:

- **Cena en un restaurante**: Reserva en "Azafrán" o "María Antonieta", conocidos por su buena comida y vinos. Presupuesto: \$30 - \$50 por persona.
- **Descanso** en el hotel.

Día 2: Tour de bodegas

Mañana:

- **Desayuno en el hotel** o en una cafetería local con medialunas y café. Presupuesto: \$5 - \$10 por persona.
- **Tour de bodegas**: Reserva un tour por las bodegas de Maipú o Luján de Cuyo. Incluye transporte, visitas a 2-3 bodegas y degustaciones. Presupuesto: \$70 - \$100 por persona.

Tarde:

- **Almuerzo en una bodega**: Muchas bodegas ofrecen almuerzos con maridaje de vinos. Disfruta de un menú de 3 pasos. Presupuesto: \$30 - \$50 por persona.
- **Continuación del tour**: Visita otras bodegas y disfruta de las vistas.

Noche:

- **Cena en el barrio de Chacra**: Prueba un restaurante como "Caminos del Vino" para una cena relajada. Presupuesto: \$30 - \$50 por persona.
- **Regreso al hotel**.

Día 3: Aventura y naturaleza

Mañana:

- **Desayuno en el hotel**: Presupuesto: \$5 - \$10 por persona.
- **Excursión a la montaña**: Reserva un tour para hacer rafting, canopy o senderismo en el Parque Provincial Aconcagua. Presupuesto: \$60 - \$100 por persona.

Tarde:

- **Almuerzo en un restaurante en la montaña** (si el tour lo incluye) o en un parador. Presupuesto: \$20 - \$30 por persona.
- **Regreso a Mendoza** y tiempo libre para compras en el centro (artesanías, vinos, etc.).

Noche:

- **Cena de despedida**: Disfruta de una cena en un lugar emblemático como "El Club de la Milanesa" o "La Casa del Vino". Presupuesto: \$30 - \$50 por persona.
- **Regreso al hotel**.

Presupuesto estimado (por persona)

- **Alojamiento (2 noches)**: \$200 - \$300
- **Comidas (3 días)**: \$160 - \$240
- **Tour de bodegas**: \$70 - \$100
- **Excursión a la montaña**: \$60 - \$100
- **Transporte (taxi, transfer)**: \$30 - \$50
- **Otros (souvenirs, snacks)**: \$30 - \$50

Total estimado: \$600 - \$890 por persona.

Notas:

- Los precios pueden variar según la temporada y la disponibilidad.
- Es recomendable hacer reservas con anticipación, especialmente para restaurantes y tours.
- Asegúrate de llevar protector solar y ropa cómoda si planeas actividades al aire libre.

¡Espero que disfrutes tu viaje a Mendoza!

```
In [3]: # =====
# Datos precargados
# =====

destino = "Mendoza"
cant_dias = 3
cant_personas = 2
presupuesto = "200000"
moneda = "ARS"
intereses = "Deportes Invernales"
modo_viaje = "Exprimelo"
```

Prompt 2 — Intermedio (más parámetros, salida guiada)

Descripción técnica

Amplía el contexto con más parámetros (personas, presupuesto, moneda, intereses) y solicita un itinerario día por día con horarios aproximados y un presupuesto dividido por rubros.

Tecnología aplicada

- `openai` (`gpt-4o-mini`) con *temperature* más baja para mejorar coherencia.

Metodología

- *Instruction prompting* con especificación de secciones (actividades, comidas, presupuesto por rubros).

Mejoras respecto a Prompt 1

- Más datos de entrada (personas/presupuesto/moneda/intereses).
- Pedido más estructurado (día por día + desglose de costos).
- Aún **no** hay JSON ni validación programática.

In [4]:

```
# =====
# Prompt 2 - Intermedio
# =====
prompt2 = f"""
Actuá como organizador de escapadas.
Con estos datos: destino={destino}, días={cant_dias}, personas={cant_personas},
Armá un itinerario día por día con horarios aproximados y comidas.
Incluí un presupuesto estimado dividido en transporte, alojamiento y actividades
"""

response2 = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": prompt2}],
    temperature=0.5
)

print("== Prompt 2 - Respuesta ==")
print(response2.choices[0].message.content)
```

==== Prompt 2 – Respuesta ===

¡Claro! Aquí tienes un itinerario de 3 días en Mendoza, enfocado en deportes invernales y ajustado a un presupuesto de 200,000 ARS para dos personas.

Itinerario de Escapada a Mendoza

Día 1: Llegada y Primeras Actividades

Mañana

- **08:00** - Salida desde tu ciudad hacia Mendoza (si viajas en auto, estima el costo de combustible; si tomas un colectivo, el costo promedio es de 15,000 ARS por persona).
- **10:00** - Llegada a Mendoza.
- **10:30** - Check-in en el hotel (sugerencia: un hotel 3 estrellas, costo aproximado 15,000 ARS por noche).

Mediodía

- **12:00** - Almuerzo en un restaurante local (costo aproximado de 4,000 ARS por persona).
- **13:30** - Paseo por el centro de Mendoza, visita a la Plaza Independencia y alrededores.

Tarde

- **15:00** - Visita a una bodega cercana para una cata de vinos (costo aproximado de 2,000 ARS por persona).
- **17:00** - Regreso al hotel para descansar.

Noche

- **20:00** - Cena en un restaurante típico (costo aproximado de 5,000 ARS por persona).
- **22:00** - Regreso al hotel para descansar.

Presupuesto Día 1:

- Transporte: 30,000 ARS (incluyendo combustible o pasajes).
 - Alojamiento: 15,000 ARS.
 - Comidas: 18,000 ARS.
 - Actividades: 4,000 ARS.
- **Total Día 1: 67,000 ARS**

Día 2: Deportes Invernales en Los Andes

Mañana

- **07:00** - Desayuno en el hotel.
- **08:00** - Salida hacia el centro de esquí (ej. Cerro Catedral o Los Penitentes, costo de transporte 10,000 ARS).
- **09:30** - Llegada al centro de esquí. Alquiler de equipo de esquí (costo aproximado de 10,000 ARS por persona).

Mediodía

- **12:00** - Almuerzo en una confitería del centro de esquí (costo aproximado de 4,000 ARS por persona).

Tarde

- **13:00** - Actividades de esquí o snowboard (costo de pase diario 12,000 ARS por persona).
- **17:00** - Regreso al hotel.

Noche

- **20:00** - Cena en un restaurante (costo aproximado de 5,000 ARS por persona).
- **22:00** - Regreso al hotel.

Presupuesto Día 2:

- Transporte: 10,000 ARS.
- Alojamiento: 15,000 ARS.
- Comidas: 18,000 ARS.
- Actividades: 44,000 ARS (alquiler y pase).

Total Día 2: 87,000 ARS

Día 3: Relajación y Regreso

Mañana

- **08:00** - Desayuno en el hotel.
- **09:00** - Check-out del hotel.
- **10:00** - Visita a termas cercanas para relajarse (costo aproximado de 6,000 ARS por persona).

Mediodía

- **12:30** - Almuerzo en un restaurante cercano a las termas (costo aproximado de 4,000 ARS por persona).

Tarde

- **14:00** - Regreso a la ciudad de origen. (costo de transporte 15,000 ARS por persona).
- **16:00** - Llegada a casa.

Presupuesto Día 3:

- Transporte: 30,000 ARS.
- Comidas: 20,000 ARS.
- Actividades: 12,000 ARS.

Total Día 3: 62,000 ARS

Presupuesto Total:

- **Día 1:** 67,000 ARS
- **Día 2:** 87,000 ARS
- **Día 3:** 62,000 ARS

Total General: 216,000 ARS

Notas:

- El presupuesto total excede un poco el límite de 200,000 ARS, pero puedes ajustar actividades o elegir un alojamiento más económico para ajustarte al presupuesto.
- Es recomendable realizar reservas anticipadas para el alojamiento y las actividades, especialmente en temporada alta de esquí.
- Considera llevar ropa adecuada para el frío y actividades de nieve.

Espero que este itinerario te ayude a disfrutar de una escapada emocionante y memorable en Mendoza. ¡Diviértanse!

Prompt 3 — Avanzado (estructura JSON y parsing)

Descripción técnica

Solicita **únicamente JSON** con campos: `parametros`, `supuestos`, `presupuesto` y `itinerario`. Se incluye el helper `safe_json_parse` para limpiar *code fences* y convertir la respuesta a diccionario Python.

Tecnología aplicada

- `openai` (`gpt-4o-mini`) + módulo `json`.

Metodología

- *Structured output prompting*: el modelo debe responder en JSON válido.
- Parsing y `print pretty` de la respuesta.

Mejoras respecto a Prompt 2

- Respuesta estructurada en JSON.
- Helper para tolerar respuestas con `json ...`.
- Base para validaciones y posterior post-proceso.

```
In [5]: import os, json

# =====
# Helper JSON
# =====
def safe_json_parse(raw_text: str):
    cleaned = raw_text.strip()
    if cleaned.startswith("```"):
        cleaned = cleaned.strip("`")
        cleaned = cleaned.replace("json\n", "").replace("json", "", 1).strip()
    try:
        return json.loads(cleaned)
    except:
        return {}

# =====
# Prompt 3 – Avanzado
# =====
prompt3 = """
Actuá como organizador de escapadas de 2-3 días.
Datos: destino={destino}, días={cant_dias}, personas={cant_personas}, presupuest
Si falta información, proponé supuestos y aclarálo en un campo "supuestos".

Devolvé SOLO un JSON con:
- parametros
- supuestos
- presupuesto general
- itinerario simple de {cant_dias} días (con actividades y comidas por día).
"""

response3 = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": prompt3}],
    temperature=0.3
)
```

```
raw3 = response3.choices[0].message.content
itinerario_json3 = safe_json_parse(raw3)

print("==> Prompt 3 – Respuesta JSON ===")
print(json.dumps(itinerario_json3, indent=2, ensure_ascii=False))
```

==== Prompt 3 – Respuesta JSON ===

```
{  
    "parametros": {  
        "destino": "Mendoza",  
        "días": 3,  
        "personas": 2,  
        "presupuesto": 200000,  
        "intereses": "Deportes Invernales",  
        "modo": "Exprímelo"  
    },  
    "supuestos": {  
        "temporada": "invierno",  
        "tipo de alojamiento": "hotel 3 estrellas",  
        "transporte": "auto alquilado",  
        "comidas": "restaurantes locales",  
        "actividades": "esquí y excursiones"  
    },  
    "presupuesto general": {  
        "alojamiento": 60000,  
        "transporte": 30000,  
        "comidas": 50000,  
        "actividades": 70000,  
        "gastos varios": 10000  
    },  
    "itinerario": {  
        "día_1": {  
            "actividades": [  
                "Llegada a Mendoza y alquiler de auto",  
                "Visita a la estación de esquí Los Penitentes",  
                "Esquí (alquiler de equipo incluido)",  
                "Cena en restaurante local"  
            ],  
            "comidas": {  
                "almuerzo": "Comida rápida en la estación de esquí",  
                "cena": "Restaurante 'La Casona' - comida típica"  
            }  
        },  
        "día_2": {  
            "actividades": [  
                "Desayuno en el hotel",  
                "Excursión a la montaña (senderismo)",  
                "Almuerzo picnic en la montaña",  
                "Regreso a la ciudad y cena en un restaurante de parrilla"  
            ],  
            "comidas": {  
                "almuerzo": "Picnic con productos locales",  
                "cena": "Restaurante 'El Asador' - parrilla argentina"  
            }  
        },  
        "día_3": {  
            "actividades": [  
                "Desayuno en el hotel",  
                "Visita a bodegas (tour y degustación)",  
                "Almuerzo en bodega",  
                "Regreso a Mendoza y entrega del auto",  
                "Salida"  
            ],  
            "comidas": {  
                "almuerzo": "Restaurante de bodega con vista a los viñedos"  
            }  
        }  
    }  
}
```

```

        }
    }
}

In [6]: from openai import OpenAI
import os, json
from google import genai
from dotenv import load_dotenv
from PIL import Image
from io import BytesIO

# =====
# Configuración API
# =====
load_dotenv()
openai_client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
gemini_client = genai.Client(api_key=os.getenv("GEMINI_API_KEY"))

# =====
# Datos precargados
# =====
destino = "Mendoza"
cant_dias = 3
cant_personas = 2
presupuesto = "20000"
moneda = "ARS"
intereses = "Deportes Invernales"
modo_viaje = "Exprímelo"
fecha_tentativa = "20/7/2025"

# =====
# Helper JSON
# =====
def safe_json_parse(raw_text: str):
    cleaned = raw_text.strip()
    if cleaned.startswith("```"):
        cleaned = cleaned.strip("`")
        cleaned = cleaned.replace("json\n", "").replace("json", "", 1).strip()
    try:
        return json.loads(cleaned)
    except:
        return {}

```

Prompt 4 — Optimizado (normalización + criterios + imágenes)

Descripción técnica

Pide **normalizar parámetros**, explicitar **supuestos**, definir **criterios de planificación** y **alertas**, y devolver un **itinerario detallado por bloques horarios**. Se fuerza *JSON-only* vía `system`. Además, **genera imágenes** (Mapa, Flyer, Infografía) con **Gemini 2.5 Flash Image** y guarda los PNG.

Tecnología aplicada

- `openai` (`gpt-4o-mini`) para el JSON.
- `google.genai` (Gemini 2.5 Flash Image) + `Pillow` para PNG.

- Cálculo y *print* de **tokens usados**.

Metodología

- Instrucciones estrictas + esquema JSON.
- Post-proceso: extracción de puntos del itinerario para construir prompts de imagen.

Mejoras respecto a Prompt 3

- JSON más rico (criterios, alertas, bloques horarios).
- Visibilidad de tokens/costos.
- Generación de imágenes complementarias.

In [7]:

```
# =====
# Prompt 4 – Optimizado (todo en una salida)
# =====
prompt4 = f"""
Actuá como un organizador de escapadas de 2-3 días.

Usá exclusivamente los siguientes datos del usuario:

- Destino: {destino}
- Días: {cant_dias}
- Personas: {cant_personas}
- Presupuesto: {presupuesto} {moneda}
- Intereses: {intereses}
- Modo de viaje: {modo_viaje}
- Fecha tentativa: {fecha_tentativa}

Normalizá esta información y devolvé SOLO un JSON con:

- parámetros normalizados (destino, días, personas, presupuesto, intereses, modo
- supuestos explícitos.
- criterios de planificación (densidad, descansos, gasto, franja horaria).
- alertas relevantes.
- itinerario DETALLADO de {cant_dias} días (bloques horarios: mañana, mediodía,
"""

response4 = openai_client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": "Sos un planificador de viajes. Respondé solo en JSON."},
        {"role": "user", "content": prompt4}
    ],
    temperature=0.3
)
print("Tokens usados:", response4.usage.total_tokens)

raw4 = response4.choices[0].message.content
itinerario_json4 = safe_json_parse(raw4)
if not isinstance(itinerario_json4, dict):
    print("⚠️ Error: el modelo no devolvió un JSON válido.")

print("== Prompt 4 – Respuesta JSON ==")
print(json.dumps(itinerario_json4, indent=2, ensure_ascii=False))

# =====
# AGREGADO: Generación de imágenes con Gemini 2.5 Flash Image
```

```
# =====
puntos = []
for dia, actividades in itinerario_json4.get("itinerario", {}).items():
    if isinstance(actividades, list):
        puntos.append(f"{dia}: " + ", ".join(actividades))
lista_puntos = " | ".join(puntos)

mapa_prompt = f"Mapa ilustrado minimalista de {destino} para {cant_dias} días. P"
flyer_prompt = f"Flyer turístico 16:9 'Escapada a {destino}', subtítulo {interes}"
infografia_prompt = f"Infografía 16:9 para {destino}, mostrando % de gastos y %"

for nombre, p in [("Mapa", mapa_prompt), ("Flyer", flyer_prompt), ("Infografia",
    response_img = gemini_client.models.generate_content(
        model="gemini-2.5-flash-image-preview",
        contents=[p],
    )
    for part in response_img.candidates[0].content.parts:
        if part.inline_data:
            img = Image.open(BytesIO(part.inline_data.data))
            img.save(f"prompt4_{nombre}.png")
            display(img) # Muestra la imagen en la celda del notebook
            print(f"{nombre} generado → prompt4_{nombre}.png")
```

Tokens usados: 1126
==== Prompt 4 – Respuesta JSON ===

```
{  
    "parametros_normalizados": {  
        "destino": "Mendoza",  
        "dias": 3,  
        "personas": 2,  
        "presupuesto": 200000,  
        "intereses": "Deportes Invernales",  
        "modo": "Exprímelo",  
        "fechas": "20/07/2025"  
    },  
    "supuestos_explicitos": {  
        "clima": "Frío, adecuado para deportes invernales",  
        "disponibilidad": "Alojamiento y actividades disponibles en temporada alta",  
        "transporte": "Acceso a transporte público o privado para traslados"  
    },  
    "criterios_de_planificacion": {  
        "densidad": "Alta, con actividades continuas",  
        "descansos": "Cortos entre actividades para maximizar el tiempo",  
        "gasto": "Moderado, priorizando experiencias sobre lujo",  
        "franja_horaria": "Actividades desde la mañana hasta la noche"  
    },  
    "alertas_relevantes": {  
        "reservas": "Reservar alojamiento y actividades con anticipación",  
        "clima": "Verificar pronóstico del tiempo antes de viajar",  
        "equipamiento": "Asegurarse de tener el equipo adecuado para deportes invernales"  
    },  
    "itinerario": {  
        "dia_1": {  
            "mañana": {  
                "actividad": "Llegada a Mendoza y traslado al alojamiento",  
                "comida": "Desayuno en una cafetería local",  
                "traslado": "Taxi o transporte público al hotel"  
            },  
            "mediodía": {  
                "actividad": "Almuerzo en restaurante de comida regional",  
                "comida": "Asado o empanadas",  
                "traslado": "Caminata o taxi al restaurante"  
            },  
            "tarde": {  
                "actividad": "Visita a una estación de esquí (ej. Los Penitentes)",  
                "comida": "Snack en la estación",  
                "traslado": "Transporte privado o colectivo a la estación"  
            },  
            "noche": {  
                "actividad": "Cena en un restaurante con vista a la montaña",  
                "comida": "Cena gourmet",  
                "traslado": "Taxi de regreso al hotel"  
            }  
        },  
        "dia_2": {  
            "mañana": {  
                "actividad": "Esquí o snowboard en la estación elegida",  
                "comida": "Desayuno en el hotel",  
                "traslado": "Transporte a la estación"  
            },  
            "mediodía": {  
                "actividad": "Almuerzo en la estación de esquí",  
                "comida": "Plato típico de la zona",  
                "traslado": "Caminata o taxi de vuelta al hotel"  
            }  
        }  
    }  
}
```

```
        "comida": "Comida rápida o menú del día",
        "traslado": "Sin traslado, comida en la estación"
    },
    "tarde": {
        "actividad": "Continuación de actividades de esquí",
        "comida": "Snack en la estación",
        "traslado": "Sin traslado, en la estación"
    },
    "noche": {
        "actividad": "Regreso a Mendoza y cena en un bar de tapas",
        "comida": "Tapas y vino local",
        "traslado": "Taxi de regreso al hotel"
    }
},
"dia_3": {
    "mañana": {
        "actividad": "Visita a bodegas locales para degustación de vinos",
        "comida": "Desayuno en el hotel",
        "traslado": "Tour en vehículo a las bodegas"
    },
    "mediodía": {
        "actividad": "Almuerzo en bodega con vista a los viñedos",
        "comida": "Menú de degustación",
        "traslado": "Sin traslado, en la bodega"
    },
    "tarde": {
        "actividad": "Regreso a la ciudad y tiempo libre para compras",
        "comida": "Snack en el centro",
        "traslado": "Transporte al centro"
    },
    "noche": {
        "actividad": "Cena de despedida en un restaurante de lujo",
        "comida": "Cena de varios platos",
        "traslado": "Taxi de regreso al hotel"
    }
}
}
```

MENDOZA - 3 DÍAS



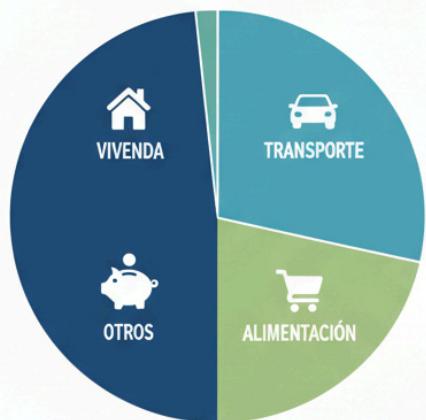
Mapa generado → prompt4_Map.png



Flyer generado → prompt4_Flyer.png

MENDOZA: GASTOS & TIEMPO PROMETIO

DISTRIBUCIÓN DE GASTOS (%)



DISTRIBUCIÓN DE TIEMPO (%)



Fuente: Enceusta Local 2023

Elaboración Propia

Infografia generado → prompt4_Infografia.png

Prompt 5 — Ultra (pipeline en 2 etapas + utilidades)

Descripción técnica

Separa el flujo en **dos prompts**:

1. *Intake* → JSON normalizado
(param/supuestos/ajustes/criterios/presupuesto/checklist/alertas/QA).
2. *Itinerario* → genera un plan detallado en **Markdown** usando el JSON del intake.
Incluye utilidades: cálculo de **tokens por etapa**, guardado a archivo, *builder* de prompts visuales y **función** para generar imágenes con Gemini.

Tecnología aplicada

- `openai` (`gpt-4o-mini`) para intake + itinerario.
- `google.genai` (Gemini 2.5 Flash Image) + `Pillow`.

Metodología

- *Chain-of-prompts*: output estructurado → consumo por segundo prompt.
- Reutilización: función de generación de imágenes para evitar código duplicado.

Mejoras respecto a Prompt 4

- Separación de responsabilidades (intake vs. rendering del itinerario).
- Medición de costos por paso.
- Código más modular y reutilizable.

```
In [8]: # =====
# Helper TOKENS
# =====

def calcular_costo(response, nombre="Prompt"):
    if not hasattr(response, "usage"):
        return
    in_tokens = response.usage.prompt_tokens
    out_tokens = response.usage.completion_tokens
    total_tokens = response.usage.total_tokens
    in_price = in_tokens * 0.00000015
    out_price = out_tokens * 0.00000006
    print(f"*** {nombre} → Tokens usados: {total_tokens} (entrada={in_tokens}, s"

# =====
# Prompt 5 – Intake ultra optimizado
# =====
intake_prompt5 = f"""
Sos un organizador experto en escapadas de 2-3 días.
Devolvé SIEMPRE un JSON válido.

IMPORTANTE:
- Usá EXCLUSIVAMENTE los parámetros precargados.
- No inventes ni cambies destino, fechas, moneda, intereses ni modo.
- Si falta info → en "supuestos". Si es inválida → en "ajustes".

Datos del usuario:
- Destino: {destino}
- Días: {cant_dias}
- Personas: {cant_personas}
- Presupuesto: {presupuesto} {moneda}
- Intereses: {intereses}
- Modo: {modo_viaje}
- Fecha: {fecha_tentativa}

Salida esperada:
{{"
    "parametros": {...},
    "supuestos": [],
    "ajustes": [],
    "criterios": {},
    "presupuesto": {},
    "checklist": [],
    "alertas": [],
    "qa": []
}}
"""

intake_response5 = client.chat.completions.create(
```

```

        model="gpt-4o-mini",
        messages=[
            {"role": "system", "content": "Respondé SOLO con JSON válido."},
            {"role": "user", "content": intake_prompt5}
        ],
        temperature=0.2
    )

calcular_costo(intake_response5, "Prompt 5 - Intake")
raw_intake5 = intake_response5.choices[0].message.content
intake_json5 = safe_json_parse(raw_intake5)

print("== Prompt 5 - Intake JSON ==")
print(json.dumps(intake_json5, indent=2, ensure_ascii=False))

# =====
# Prompt 5 - Itinerario ultra optimizado
# =====
itinerario_prompt5 = f"""
Usá este contexto JSON:

{json.dumps(intake_json5, indent=2, ensure_ascii=False)}"""

Generá un itinerario DETALLADO de {cant_dias} días en Markdown limpio.

Formato:
Día N – Zona / tema principal
09:00-11:00 Actividad (lugar)
11:15-13:00 Actividad (lugar)
13:15-14:30 Almuerzo (opciones)
15:00-17:00 Actividad (lugar)
17:15-19:00 Actividad (lugar)
20:00 Cena (opciones)

Incluir: traslados, resumen, tips, presupuesto bajo/medio/alto ({moneda}), alert
"""

itinerario_response5 = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": "Devolvé solo itinerario en Markdown."},
        {"role": "user", "content": itinerario_prompt5}
    ],
    temperature=0.4
)

calcular_costo(itinerario_response5, "Prompt 5 - Itinerario")
content5 = itinerario_response5.choices[0].message.content

print("\n== Itinerario Ultra generado ==")
print(content5)

with open("itinerario_ultra.txt", "w", encoding="utf-8") as f:
    f.write(content5)

# =====
# Builder para Prompts de Imagen basados en el itinerario
# =====
def build_image_prompts(destino, cant_dias, intereses, itinerario_text):
    puntos = []

```

```

        for line in itinerario_text.splitlines():
            if "Actividad" in line or "Almuerzo" in line or "Cena" in line:
                puntos.append(line.strip("- *"))

        lista_puntos = " | ".join(puntos)

        mapa_prompt = f"""
Mapa ilustrado minimalista de {destino} para una escapada de {cant_dias} días.
Marcar con íconos las paradas clave:
{lista_puntos}
Leyenda: Día 1 a Día {cant_dias}.
Líneas suaves de conexión, tipografías legibles, paleta cálida. Formato 16:9.
Devuelve únicamente la imagen del mapa en PNG, sin texto adicional.
"""

        flyer_prompt = f"""
Flyer turístico 16:9 para {destino}, estilo travel-poster moderno.
Título: 'Escapada a {destino}'.
Subtítulo: {intereses}.
Imagen icónica del lugar, composición limpia, margen seguro para texto.
Devuelve únicamente la imagen en PNG, sin texto adicional.
"""

        infografia_prompt = f"""
Infografía clara para {destino}, formato 16:9.
Incluye dos gráficos simples:
- Gráfico circular de gastos: 30% transporte, 25% alojamiento, 25% comidas, 20%
- Gráfico de barras de tiempo: 40% actividades, 30% traslados, 20% descanso, 10%
Iconos simples y etiquetas legibles.
Devuelve únicamente la imagen en PNG, sin texto adicional.
"""

    return [
        ("Mapa", mapa_prompt),
        ("Flyer", flyer_prompt),
        ("Infografia", infografia_prompt)
    ]

# =====
# Función optimizada para generar imágenes (fix binarios)
# =====
def generar_imagenes(prompts, prefix="output"):
    for nombre, p in prompts:
        try:
            print(f"\n==> Generando {nombre} ==>")
            response_img = gemini_client.models.generate_content(
                model="gemini-2.5-flash-image-preview",
                contents=[p],
            )

            saved = False
            for part in response_img.candidates[0].content.parts:
                if hasattr(part, "inline_data") and part.inline_data:
                    img = Image.open(BytesIO(part.inline_data.data))
                    img.save(f"{prefix}_{nombre}.png")
                    display(img) # Muestra la imagen en la celda del notebook
                    print(f"✓ {nombre} generado → {prefix}_{nombre}.png")
                    saved = True

            if not saved:
        
```

```
        print(f"⚠️ No se generó imagen para {nombre}. Gemini no devolvió nada")
    except Exception as e:
        print(f"❌ Error generando {nombre}: {e}")

# =====
# Uso con el itinerario generado
# =====
prompts_img = build_image_prompts(destino, cant_dias, intereses, contents)
generar_imagenes(prompts_img, prefix="prompt5")
```

```
*** Prompt 5 - Intake → Tokens usados: 332 (entrada=214, salida=118), USD 0.00010
3
==> Prompt 5 – Intake JSON ==
{
    "parametros": {
        "destino": "Mendoza",
        "días": 3,
        "personas": 2,
        "presupuesto": 200000,
        "intereses": "Deportes Invernales",
        "modo": "Exprímelo",
        "fecha": "20/7/2025"
    },
    "supuestos": [],
    "ajustes": [],
    "criterios": {},
    "presupuesto": {},
    "checklist": [],
    "alertas": [],
    "qa": []
}
*** Prompt 5 - Itinerario → Tokens usados: 947 (entrada=269, salida=678), USD 0.0
00447
```

==> Itinerario Ultra generado ==

Itinerario Mendoza - Deportes Invernales (20/07/2025)

Día 1 – Llegada y Exploración de la Ciudad

09:00-11:00 Llegada a Mendoza y traslado al hotel (Hotel X)

11:15-13:00 Visita al Parque General San Martín (paseo y fotos)

13:15-14:30 Almuerzo en "El Patio de Jesús María" (opciones: asado, empanadas)

15:00-17:00 Tour por el centro histórico (Plaza Independencia, Calle Arístides)

17:15-19:00 Degustación de vinos en "Bodega López"

20:00 Cena en "La Lucia" (opciones: parrilla, pastas)

Resumen: Primer día para aclimatarse y disfrutar de la cultura local.

Tips: Llevar ropa cómoda para caminar.

Presupuesto: Bajo (ARS 15,000 por persona)

Alertas QA: Verificar horarios de apertura de las bodegas.

Día 2 – Deportes Invernales en Los Andes

09:00-11:00 Traslado a la estación de esquí "Los Penitentes"

11:15-13:00 Clases de esquí (alquiler de equipo incluido)

13:15-14:30 Almuerzo en el restaurante de la estación (opciones: sándwiches, guisos)

15:00-17:00 Esquí libre (práctica personal)

17:15-19:00 Regreso a Mendoza y descanso en el hotel

20:00 Cena en "Café de la Plaza" (opciones: pizzas, platos locales)

Resumen: Día completo de deportes invernales en la montaña.

Tips: Reservar clases de esquí con anticipación.

Presupuesto: Medio (ARS 40,000 por persona)

Alertas QA: Comprobar el estado de las pistas antes de salir.

Día 3 – Aventura y Relax

09:00-11:00 Traslado a "Potrerillos" para actividades al aire libre

11:15-13:00 Caminata y exploración de la zona (miradores y naturaleza)
13:15-14:30 Almuerzo en "Restaurante El Rincón" (opciones: comida regional)
15:00-17:00 Actividad de rafting o canopy (opciones según disponibilidad)
17:15-19:00 Regreso a Mendoza y tiempo libre para compras
20:00 Cena en "El Club de la Milanesa" (opciones: milanesas, ensaladas)

Resumen: Un día de aventura seguido de tiempo libre para disfrutar de la ciudad.

Tips: Llevar ropa adecuada para actividades al aire libre.

Presupuesto: Alto (ARS 50,000 por persona)

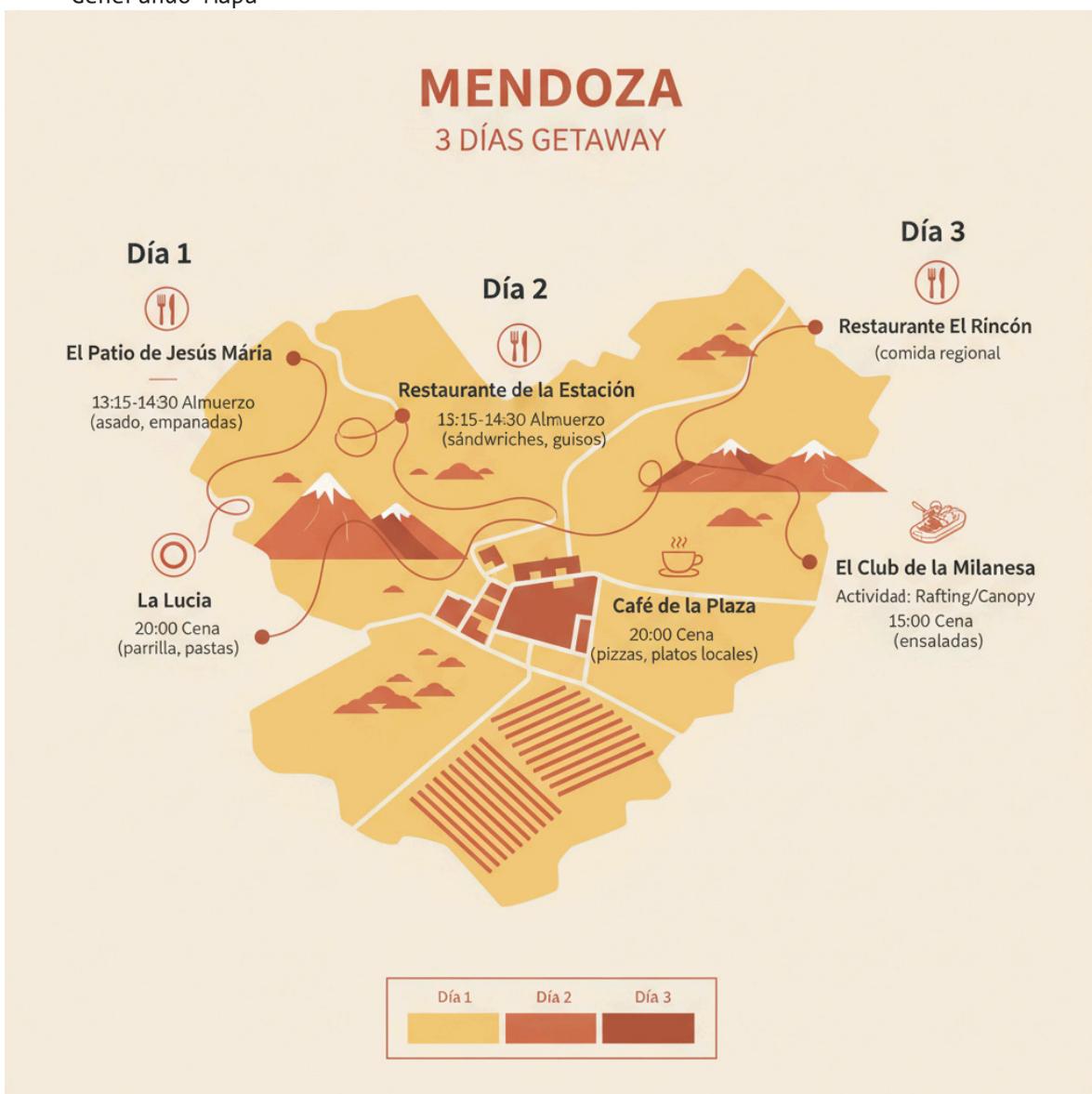
Alertas QA: Verificar condiciones climáticas para actividades de aventura.

Presupuesto Total Estimado: ARS 105,000 por persona

Presupuesto Total para 2 personas: ARS 210,000

Nota: Ajustar actividades y comidas según preferencias y disponibilidad.

== Generando Mapa ==



✓ Mapa generado → prompt5_Mapa.png

== Generando Flyer ==



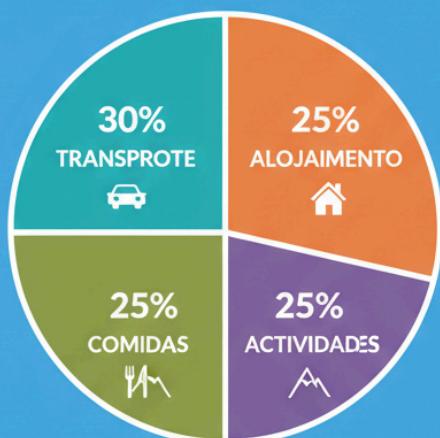
Flyer generado → prompt5_Flyer.png

== Generando Infografia ==

MENDOZA: TU VIAJE IDEAL



GASTOS



TIEMPO



Descubre la aventura, sabor a cada momento.

Infografia generado → prompt5_Infografia.png

Prompt 5 Lite — Minimalista (tokens bajos + mapa/flyer)

Descripción técnica

Versión liviana que mantiene intake en JSON **breve** y un itinerario **simple**. La generación visual se reduce a **Mapa** y **Flyer**. El **Mapa** se adapta dinámicamente al **itinerario** (extrae puntos clave del texto). **Sin infografía** para evitar textos de mala calidad.

Tecnología aplicada

- `openai` (`gpt-4o-mini`) para intake + itinerario.
- `google.genai` (Gemini 2.5 Flash Image) + `Pillow`.
- *Prompts* visuales centralizados en un **diccionario** y función única para exportar PNG.

Metodología

- *Cost-aware prompting*: menos tokens y salidas más simples.
- Dinamización del mapa a partir del itinerario generado.

Mejoras respecto a Prompt 5

- Menor consumo de tokens (prompts y salidas más cortos).
- Eliminación de la infografía.
- Mapa con **fondo de color** y **sin texto**, alimentado por el itinerario.

In [9]:

```
# =====
# Prompt 5 Lite - Intake
# =====
intake_prompt_lite = f"""
Sos un organizador de {cant_dias} días.
Devolvé SOLO un JSON breve.

IMPORTANTE:
- Usá EXCLUSIVAMENTE los parámetros precargados.
- No inventes ni cambies destino, fechas, moneda, intereses ni modo.

JSON esperado:
{{{
    "param": {{
        "dest": "{destino}",
        "dias": {cant_dias},
        "pers": {cant_personas},
        "presup_pp": [{"m": {presupuesto}, "mon": "{moneda}"}],
        "int": ["{intereses}"],
        "modo": "{modo_viaje}",
        "fechas": "{fecha_tentativa}"
    }},
    "sup": [],
    "ajus": [],
    "crit": {{}},
    "presup": {{}},
    "chk": [],
    "alert": [],
    "qa": []
}}
"""

intake_response_lite = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": "Respondé SOLO con JSON válido y breve."},
        {"role": "user", "content": intake_prompt_lite}
    ],
    temperature=0.2
)

calcular_costo(intake_response_lite, "Prompt 5-Lite - Intake")
raw_intake_lite = intake_response_lite.choices[0].message.content
intake_json_lite = safe_json_parse(raw_intake_lite)

print("== Prompt 5 Lite - Intake JSON ==")
print(json.dumps(intake_json_lite, indent=2, ensure_ascii=False))

# =====
# Prompt 5 Lite - Itinerario
# =====
itinerario_prompt_lite = f"""
```

Usá este JSON:

```
{json.dumps(intake_json_lite, indent=2, ensure_ascii=False)}
```

Generá un itinerario simple de {cant_dias} días.

Formato:

Día N - Zona
09:00-11:00 Actividad
11:15-13:00 Actividad
13:15-14:30 Almuerzo (2 opciones)
15:00-17:00 Actividad
17:15-19:00 Actividad
20:00 Cena (2 opciones)

Incluí traslados, resumen, tips, presupuesto ({moneda}), alertas QA.

""",

```
itinerario_response_lite = client.chat.completions.create(  
    model="gpt-4o-mini",  
    messages=[  
        {"role": "system", "content": "Devolvé solo itinerario en texto limpio."  
        {"role": "user", "content": itinerary_prompt_lite}  
    ],  
    temperature=0.4  
)  
  
calcular_costo(itinerario_response_lite, "Prompt 5-Lite - Itinerario")  
content_lite = itinerary_response_lite.choices[0].message.content  
  
print("\n== Itinerario Lite generado ==")  
print(content_lite)  
  
with open("itinerario_lite.txt", "w", encoding="utf-8") as f:  
    f.write(content_lite)  
  
# ======  
# Extraer puntos clave del itinerario  
# ======  
puntos = []  
for line in content_lite.splitlines():  
    if any(keyword in line.lower() for keyword in ["actividad", "almuerzo", "cer"]):  
        puntos.append(line.strip())  
lista_puntos = " | ".join(puntos)  
  
# ======  
# Prompts de Imagen (mapa dinámico + flyer)  
# ======  
prompts_img = {  
    "Mapa": f"""  
Mapa turístico ilustrado de {destino} basado en el itinerario.  
Debe incluir un contorno simple del área de viaje y ubicar íconos en los siguientes puntos:  
{lista_puntos}  
  
Usar solo íconos (vino, montañas, bicicleta, lago, comida, etc.), sin texto ni líneas.  
Fondo con color cálido (arena o terracota), estilo turístico minimalista.  
Formato 16:9, imagen clara y coherente.  
Devuelve solo la imagen en PNG.  
""",
```

```

    "Flyer": f"""
Flyer turístico 16:9 para {destino}, estilo travel-poster moderno.
Título: 'Escapada a {destino}'.
Subtítulo: {intereses}.
Imagen icónica del lugar, composición limpia, margen seguro para texto.
Devuelve únicamente la imagen en PNG.
"""
}

# =====
# Función para generar imágenes
# =====

def generar_imagenes_lite(prompts: dict, prefix="prompt5lite"):
    for nombre, p in prompts.items():
        try:
            print(f"\n==> Generando {nombre} ==>")
            response_img = gemini_client.models.generate_content(
                model="gemini-2.5-flash-image-preview",
                contents=[p],
            )
            saved = False
            for part in response_img.candidates[0].content.parts:
                if hasattr(part, "inline_data") and part.inline_data:
                    img = Image.open(BytesIO(part.inline_data.data))
                    img.save(f"{prefix}_{nombre}.png")
                    display(img) # Muestra la imagen en la celda del notebook
                    print(f"✓ {nombre} generado → {prefix}_{nombre}.png")
                    saved = True
            if not saved:
                print(f"⚠️ No se generó imagen para {nombre}. Gemini no devolvió imágenes")
        except Exception as e:
            print(f"✗ Error generando {nombre}: {e}")

# =====
# Generar imágenes
# =====

generar_imagenes_lite(prompts_img, prefix="prompt5lite")

```

```
*** Prompt 5-Lite - Intake → Tokens usados: 315 (entrada=195, salida=120), USD 0.  
000101  
==> Prompt 5 Lite – Intake JSON ==  
{  
    "param": {  
        "dest": "Mendoza",  
        "dias": 3,  
        "pers": 2,  
        "presup_pp": {  
            "m": 200000,  
            "mon": "ARS"  
        },  
        "int": [  
            "Deportes Invernales"  
        ],  
        "modo": "Exprímelos",  
        "fechas": "20/7/2025"  
    },  
    "sup": [],  
    "ajus": [],  
    "crit": {},  
    "presup": {},  
    "chk": [],  
    "alert": [],  
    "qa": []  
}  
*** Prompt 5-Lite - Itinerario → Tokens usados: 896 (entrada=254, salida=642), US  
D 0.000423
```

==> Itinerario Lite generado ==

Itinerario Mendoza - Deportes Invernales (20/7/2025)

Día 1 - Valle de Uco

09:00-11:00 Llegada y traslado a Valle de Uco
11:15-13:00 Clase de esquí en la estación de esquí
13:15-14:30 Almuerzo
- Opción 1: Restaurante "El Rincón del Esquí"
- Opción 2: "La Cabaña del Montañés"
15:00-17:00 Esquí libre en la pista
17:15-19:00 Descanso en el lodge
20:00 Cena
- Opción 1: "Café del Esquí"
- Opción 2: "Parrilla del Valle"

Día 2 - Los Andes

09:00-11:00 Traslado a Los Andes
11:15-13:00 Actividad de snowboard
13:15-14:30 Almuerzo
- Opción 1: "Restaurante La Nieve"
- Opción 2: "Comedor de Montaña"
15:00-17:00 Paseo en raquetas de nieve
17:15-19:00 Relax en el spa del hotel
20:00 Cena
- Opción 1: "Cocina de la Montaña"
- Opción 2: "Bistro de los Andes"

Día 3 - Parque Provincial Aconcagua

09:00-11:00 Traslado al Parque Provincial Aconcagua
11:15-13:00 Caminata guiada por senderos
13:15-14:30 Almuerzo

- Opción 1: "Picnic en la Montaña"
 - Opción 2: "Restaurante Aconcagua"
- 15:00-17:00 Visita al mirador del Aconcagua
17:15-19:00 Regreso a Mendoza
20:00 Cena
- Opción 1: "Cocina Regional"
 - Opción 2: "Restaurante del Centro"

****Resumen:****

- 3 días de actividades de deportes invernales en Mendoza, incluyendo esquí y snowboard.
- Traslados incluidos a las diferentes zonas.
- Opción de almuerzos y cenas en restaurantes locales.

****Tips:****

- Llevar ropa adecuada para el frío y la nieve.
- Reservar actividades y restaurantes con anticipación.
- Consultar el estado de las pistas antes de ir.

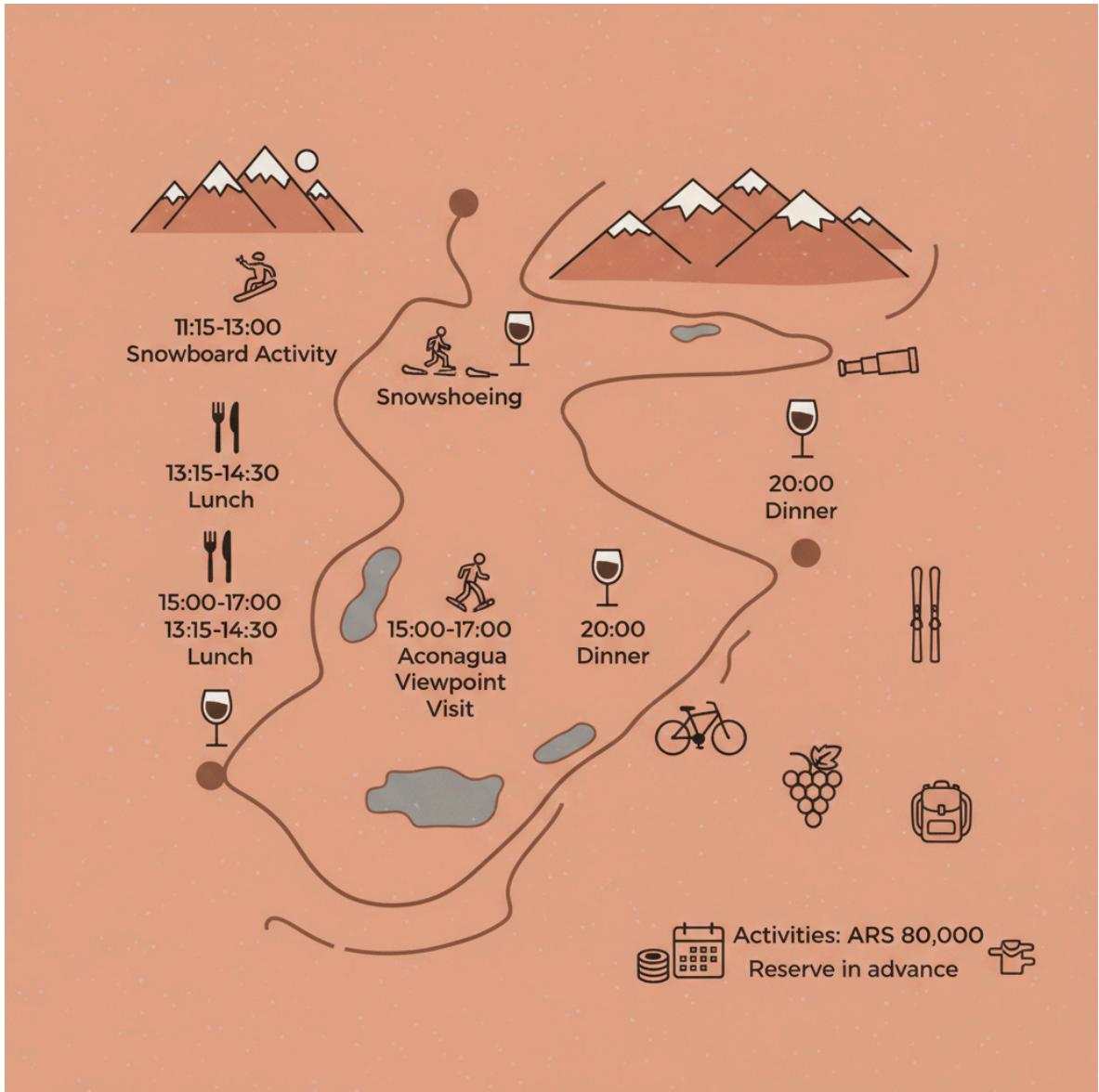
****Presupuesto estimado (por persona):****

- Actividades: ARS 80,000
- Comidas: ARS 50,000
- Traslados: ARS 30,000
- Total: ARS 160,000

****Alertas QA:****

- Verificar disponibilidad de actividades en la fecha seleccionada.
- Comprobar condiciones climáticas antes de realizar deportes al aire libre.
- Asegurarse de tener el equipo adecuado para las actividades.

==== Generando Mapa ===



Mapa generado → prompt5lite_Mapa.png

== Generando Flyer ==



Flyer generado → prompt5lite_Flyer.png

❖ Conclusiones

El desarrollo de este proyecto permitió aplicar de forma práctica la metodología de **Fast Prompting**, evidenciando cómo la iteración y optimización progresiva de prompts mejora tanto la calidad de las respuestas como la eficiencia en el uso de recursos.

Logros principales

- Se diseñaron **6 prompts evolutivos**, mostrando la transición desde un esquema básico hasta una versión optimizada y modular.
- Se incorporó el **uso de JSON estructurado**, lo que facilitó la validación y reutilización de los resultados.
- Se aplicó **role assignment** para controlar la salida y asegurar consistencia.
- Se implementó un esquema de **medición de costos (tokens)**, clave para evaluar la viabilidad económica de la solución.

- Se integraron **herramientas de generación visual** (mapa y flyer), enriqueciendo la experiencia del usuario con recursos gráficos.
- Se redujo el consumo de tokens con la versión **Lite**, demostrando un enfoque de **cost-aware prompting**.

Análisis crítico

- El modelo responde bien a instrucciones estructuradas, pero requiere **validación constante** para evitar salidas inválidas (especialmente en JSON).
- La generación de imágenes mostró limitaciones en cuanto a **calidad de texto y nivel de detalle en mapas**, lo que abre la puerta a futuros ajustes de prompts y/o uso de modelos especializados.
- La evolución de los prompts reflejó claramente la **importancia de la iteración**: cada paso corrigió falencias del anterior y agregó valor al resultado final.

Próximos pasos

- Automatizar la entrada de datos (destino, presupuesto, intereses) a través de formularios o JSON dinámicos.
- Optimizar la integración de imágenes, asegurando mayor coherencia visual con los itinerarios generados.
- Extender la solución hacia una aplicación web o dashboard interactivo que permita a cualquier usuario generar su propia escapada.

En conclusión, este trabajo demuestra cómo el **Fast Prompting** aplicado de manera progresiva no solo mejora los resultados, sino que también constituye una metodología sólida para el diseño de soluciones basadas en IA.