YA-NT2-C | Informe presentación App Promptia

Integrantes:

- Daniel Guzman
- Marcos Elian Wendy
- Mariano Andrés Izarriaga

Índice

Resumen

El presente documento describe la aplicación Promptia, desarrollada como proyecto integrador de dos materias distintas:

- Frontend: realizado en Programación en Nuevas Tecnologías II (PNT2)
- Backend: realizado en Taller de Programación II (TP2)

Promptia es una aplicación web Cliente/Servidor que permite a los usuarios mantener conversaciones con un asistente virtual impulsado por Inteligencia Artificial (Google Gemini).

El sistema incluye autenticación segura, persistencia de conversaciones, manejo de múltiples sesiones y respuestas en tiempo real. Este documento está dirigido a docentes y evaluadores del proyecto y tiene como finalidad explicar el funcionamiento, las funcionalidades y los detalles técnicos tanto del frontend como del backend.

Beneficios

✔ Chat con Inteligencia Artificial (Google Gemini 2.5 Flash)

✔ Respuestas en tiempo real mediante streaming

✔ Autenticación y seguridad mediante JWT

✔ Base de datos (MongoDB)

✔ Historial de chats persistente

✔ Interfaz moderna y responsiva

✔ Frontend separado del backend (arquitectura desacoplada)

✔ Manejo de múltiples conversaciones

✔ Personalización del agente IA (System Prompt y Temperatura)

✔ Herramientas integradas (generación de imágenes y clima)

Funcionalidades Principales

Registro de Usuario

1. Alta de usuario con email y contraseña
2. Contraseña encriptada para seguridad
3. Validación de datos en tiempo real

Login

1. Autenticación mediante email y contraseña
2. Generación de token JWT para sesiones seguras
3. Token almacenado en LocalStorage

Dashboard (Home)

1. Vista inicial con acceso directo al chat
2. Navegación intuitiva

Listado de Sesiones

1. Muestra todos los chats del usuario
2. Sidebar con lista de conversaciones guardadas
3. Indicador visual de sesión activa

Creación de Sesión

1. Botón "Nuevo Chat" para crear conversaciones
2. Generación automática de título
3. Sesiones guardadas automáticamente

Envío de Mensajes

1. El usuario envía mensajes a la IA
2. Interfaz de chat intuitiva
3. Soporte para Enter o botón "Enviar"

Streaming de Respuesta

1. La IA responde en tiempo real (palabra por palabra)
2. Tecnología Server-Sent Events (SSE)
3. Experiencia fluida y dinámica

Persistencia de Historial

1. Cada mensaje se guarda en MongoDB

2. Historial completo disponible al reabrir sesiones
3. Sincronización automática

Personalización del Agente

1. System Prompt: Define el rol del agente (ej: "Eres un experto en programación")
2. Temperatura: Controla creatividad (0.0 = preciso, 2.0 = creativo)

Herramientas del Agente

1. Generación de imágenes: "Genera una imagen de un atardecer"
2. Información del clima: "¿Cómo está el clima en Buenos Aires?"

Logout

1. El usuario cierra sesión de forma segura
2. Eliminación del token JWT

Detalles técnicos

Frontend (Programación en Nuevas Tecnologías 2)

Tecnologías usadas

1. React 19 + TypeScript
2. React Router 7
3. Tailwind CSS
4. Custom Hooks
5. Fetch API para comunicarse con el backend
6. LocalStorage para el token JWT
7. Componentización

Flujo principal

1. Usuario se loguea → el token se guarda en LocalStorage
2. Se renderiza el **ChatPage**
3. Sidebar muestra las sesiones (pedido al backend)
4. Al enviar un mensaje, el frontend hace un fetch a:/api/chat/stream para obtener respuesta en streaming
5. El chat se actualiza en tiempo real

Backend (Taller de Programación 2)

Tecnologías usadas

1. NodeJS y Express
2. MongoDB
3. JWT para autenticación
4. Zod para validaciones
5. Google Gemini API
6. Vitest + Supertest para los tests
7. Server-Sent Events (SSE) para streaming

Arquitectura:

1. Monorepo con pnpm workspaces
2. Separación de capas: Routes / Services / Repositories
3. Código compartido en packages (types, schemas, utils)

Endpoints principales:

1. POST /api/auth/register - Registro de usuarios
2. POST /api/auth/login - Login (devuelve JWT)
3. GET /api/sessions - Listar sesiones del usuario
4. POST /api/sessions - Crear nueva sesión
5. GET /api/sessions/:id - Obtener sesión con historial
6. POST /api/chat/stream - Enviar mensaje (respuesta streaming)

Instrucciones de uso de la App.

Pre-requisitos

1. Navegador web moderno (Chrome, Firefox, Edge, Safari)
2. Conexión a internet estable
3. Correo electrónico válido para registro

Para Desarrollo

1. Node.js (versión 18 o superior)
2. pnpm (gestor de paquetes)
3. MongoDB (local o remoto)
4. API Key de Google Gemini

Guia de uso

Registro

1. Acceder a /register
2. Completar el formulario:
    - Nombre (opcional)
    - Email (obligatorio)
    - Contraseña (mínimo 8 caracteres)
3. Clic en "Sign Up"
4. Redirección automática al chat

Login

1. Acceder a /login
2. Ingresar credenciales
3. Clic en "Sign In"

Credenciales de prueba:
Email: user@gmail.com
Password: password12345

Crear un Chat

1. Clic en "Nuevo Chat" en el sidebar
2. Se crea una nueva sesión automáticamente
3. Comenzar a escribir mensajes

Personalizar el Agente
System Prompt (arriba del chat):

Temperatura:

1. 0.0 - 0.3: Respuestas precisas (código, matemáticas)
2. 0.4 - 0.7: Balance (uso general)
3. 0.8 - 2.0: Respuestas creativas (escritura, ideas)

Usar Herramientas
Generación de imágenes:
Genera una imagen de un gato astronauta en el espacio.

Información del clima:
¿Cómo está el clima en Buenos Aires?

Cambiar entre Sesiones
1. Clic en cualquier sesión del sidebar
2. El historial se carga automáticamente
3. Continuar la conversación

Modo Oscuro
1. Clic en el botón de modo oscuro (navbar superior)
2. Cambio instantáneo de tema

Cerrar Sesión
1. Clic en "Logout" (navbar superior)
2. Token eliminado, redirección a login