**Documento Resumen Final: App BloodConnect SOS**  
*(Para uso interno – Confidencial)*

**Índice**

1. **¿Qué es la app?** (Explicación sencilla)
2. **Para qué sirve** (Objetivo y contexto de crisis climáticas)
3. **Tecnologías clave**
4. **Qué hemos hecho hasta ahora**
5. **Qué falta por hacer** (Pasos prácticos y urgentes)
6. **Cómo ganará dinero** (Modelo de negocio)
7. **Riesgos y cómo evitarlos**
8. **Próximos pasos inmediatos**

**1. ¿Qué es la app?**

Es un **"Uber de la sangre"** que:

* **Conecta donantes con hospitales en segundos** durante emergencias (huracanes, terremotos, brotes de dengue).
* **Predice faltantes de sangre** usando el clima y datos de salud.
* **Premia a los donantes** con descuentos, insignias y reconocimiento.

**2. Para qué sirve**

* **Salvar vidas en desastres**: Cuando un hospital se queda sin sangre, la app encuentra donantes cerca.
* **Combate enfermedades del clima**: Ej.: Si hay un brote de dengue, prioriza donantes con anticuerpos.
* **Incentiva la donación**: Hace que donar sea fácil, rápido y gratificante.

**3. Tecnologías Clave**

| **Función** | **Herramientas** | **¿Por qué?** |
| --- | --- | --- |
| Base de datos | Oracle SQL (ya creada) | Segura, manejas bien PL/SQL. |
| Backend | Java + Spring Boot | Ideal para APIs robustas y escalables. |
| Frontend | Bootstrap + JavaScript | Rápido de hacer y responsive. |
| IA Predictiva | Python (ChatGPT ayuda el código) | Predice faltantes sin gastar en software. |
| Geolocalización | Google Maps API + Leaflet.js | Muestra donantes y emergencias en un mapa. |
| Notificaciones | Firebase Cloud Messaging | Alertas en tiempo real a donantes. |

**4. Qué hemos hecho hasta ahora**

✅ **Base de datos completa** (7 tablas en Oracle):

* Donantes, hospitales, historial de donaciones, clima, etc. **Prototipo de diseño** (UI/UX básico con Bootstrap).
* **Algoritmo predictivo simple** (usando Python y ChatGPT).

**5. Qué falta por hacer**

**Urgente (Próximas 2 semanas)**

1. **Backend (Java Spring Boot)**:
   * Crear 3 APIs básicas:

java

Copy

Download

// Ejemplo API para buscar donantes

@GetMapping("/donantes-cercanos")

public List<Donor> getDonantesCercanos(@RequestParam double lat, @RequestParam double lon) { ... }

* + Conectar con Oracle SQL usando JPA.

1. **Frontend (HTML/JS)**:
   * Página de registro de donantes (con mapa integrado).
   * Panel de emergencias activas (ej.: tarjetas con ubicación y tipo de sangre).
2. **Notificaciones (Firebase)**:
   * Configurar envío de alertas push cuando se active una emergencia.

**A mediano plazo (1-2 meses)**

* Integrar la **IA predictiva** (Python) con el backend Java.
* Desarrollar el sistema de **recompensas** (puntos, insignias).
* Hacer pruebas reales con **2 hospitales aliados**.

**6. Cómo ganará dinero**

* **Suscripciones para hospitales**: 100−100−500/mes por acceso a datos predictivos y alertas.
* **Comisiones por donaciones exitosas**: $2 por donante movilizado.
* **Publicidad contextual**: Ej.: Clínicas privadas patrocinan campañas en la app.

**7. Riesgos y cómo solucionarlos**

| **Riesgo** | **Solución** |
| --- | --- |
| Pocos donantes registrados | Campañas en redes sociales con influencers. |
| Fallos en predicciones de IA | Usar datos reales de hospitales para ajustar modelos. |
| Competencia de apps similares | Focalizarse en crisis climáticas (nicho único). |

**8. Próximos pasos inmediatos**

1. **Comenzar con el backend**:

bash

Copy

Download

# Clona este repositorio base de Spring Boot + Oracle:

git clone https://github.com/ejemplo-spring-oracle.git

1. **Crear la primera API**:
   * Endpoint: GET /api/donantes (lista todos los donantes).
   * Usar ChatGPT para generar código Java si te atoras.
2. **Diseñar la pantalla de mapa**:
   * Integra Leaflet.js con este tutorial: [Leaflet en 5 minutos](https://leafletjs.com/examples/quick-start/" \t "_blank).

**Recursos útiles**:

* [Documentación de Spring Boot](https://spring.io/guides)
* [Ejemplo de API REST con Java](https://www.baeldung.com/rest-with-spring-series)
* [Plantilla de Bootstrap para la app](https://startbootstrap.com/theme/sb-admin-2)

**¡Hoy mismo!**

1. **Ejecuta en ChatGPT**: *"Genera el código Java para conectar Spring Boot con Oracle Database"*.
2. **Comienza con el endpoint**/donantes-cercanos.
3. **Comparte tus avances para resolver dudas técnicas al instante**.

*(Documento actualizado al 15/10/2023 – No compartir sin autorización)* 🔐

¡Perfecto! Crear una **demo interactiva con IA** es más rápido y económico de lo que parece. Aquí tienes un plan paso a paso con las **mejores herramientas de IA** para lograrlo:

**🚀 Paso 1: Definir el User Flow (Flujo de Usuario)**

**Herramienta IA:** ChatGPT o Gemini (para estructurar el flujo).  
**Acciones:**

1. Pídele a ChatGPT:  
   *"Genera un user flow para una app de donación de sangre que incluya: registro de donantes, alertas de emergencia con geolocalización y sistema de recompensas. Usa diagramas de cajas y flechas."*
2. Refina el resultado y elige las pantallas clave (ej: mapa de emergencias, perfil de donante, notificaciones).

**🚀 Paso 2: Generar Assets Visuales (UI/UX)**

**Herramientas IA:**

* **MidJourney** / **DALL-E 3** (para imágenes de fondo, íconos).
* **Galileo AI** (para diseño de interfaces en segundos).
* **Uizard** (prototipado con IA).

**Acciones:**

1. **Diseña las pantallas con Galileo AI**:
   * Ejemplo de prompt: *"Diseña una pantalla de app móvil para donación de sangre con: mapa interactivo, botón de emergencia rojo y perfil de usuario con tipo de sangre O+."*
   * Exporta los diseños en formato PNG o Figma.
2. **Crea íconos con DALL-E 3**:
   * Prompt: *"Ícono minimalista de una gota de sangre con un símbolo de GPS, estilo plano, colores rojo y blanco."*

**🚀 Paso 3: Crear el Prototipo Interactivo**

**Herramientas IA:**

* **Figma + Plugins de IA** (para animaciones y conexiones).
* **ProtoPie** (prototipos interactivos con lógica condicional).
* **Uizard Autodesigner** (convierte texto en prototipos).

**Acciones:**

1. **Usa Uizard Autodesigner**:
   * Sube los diseños de Galileo AI o escribe prompts como:  
     *"Prototipo clickeable de app con 5 pantallas: registro, mapa, alerta de emergencia, perfil y recompensas."*
   * La IA generará un demo navegable automáticamente.
2. **Agrega interacciones con Figma + AI Plugins**:
   * Instala el plugin **"AI Figma Assistant"** para conectar pantallas con gestos (ej: "Al hacer clic en el botón de emergencia, ir al mapa").

**🚀 Paso 4: Simular Datos Reales con IA**

**Herramientas:**

* **ChatGPT** (generar datos falsos realistas).
* **Mockaroo** (para JSONs de donantes ficticios).

**Acciones:**

1. Pídele a ChatGPT:  
   *"Genera una lista de 20 donantes ficticios con: nombre, tipo de sangre, última donación y ubicación (lat/long) en la Ciudad de México."*
2. Usa esos datos en el prototipo para simular donantes cercanos en el mapa.

**🚀 Paso 5: Añadir IA Predictiva (Fake Demo)**

**Herramientas:**

* **Voiceflow** (para simular un chatbot predictivo).
* **MindStudio** (crear flujos de IA sin código).

**Acciones:**

1. Crea un **chatbot de emergencia** con Voiceflow:
   * Ejemplo: *"¿Habrá escasez de sangre tipo A+ en la próxima semana? Considera datos climáticos de lluvias en la zona."*
   * Respuesta simulada: *"Alerta: 80% de probabilidad de escasez por inundaciones. ¡Necesitamos 50 donantes!"*
2. Integra el chatbot en el prototipo (vía enlace o iframe).

**🚀 Paso 6: Demostrar Geolocalización con IA**

**Herramientas:**

* **Mapbox + ChatGPT** (generar coordenadas ficticias).
* **Figma** (simular mapas interactivos).

**Acciones:**

1. Usa **Mapbox Studio** para crear un mapa de una ciudad ficticia con marcadores de emergencia.
2. Pídele a ChatGPT que genere un script de geolocalización:  
   *"Crea un código JSON con 10 ubicaciones de emergencia en Bogotá (lat/long) y tipos de sangre requeridos."*

**🚀 Paso 7: Grabar un Video Demostrativo con IA**

**Herramientas:**

* **ScreenStudio** (grabar el prototipo).
* **Synthesia** (crear un avatar narrador).
* **ElevenLabs** (voz en off realista).

**Acciones:**

1. Graba la demo interactiva con ScreenStudio.
2. Usa ElevenLabs para generar una voz en off:  
   *"En esta pantalla, los donantes reciben alertas en tiempo real durante un huracán."*
3. Combina todo en **CapCut** (gratis) o **Runway** (con IA para edición automática).
4. **Algoritmo de IA predictivo**:
   1. Si desarrollas un modelo único que predice escasez de sangre usando **datos climáticos + epidemiológicos** (ej.: correlación entre huracanes y demanda de plasma).