

# API Endpoints Documentation

## Sistema de Monitoreo de Salud con Wearables

**Versión:** 1.0  
**Base URL:** `http://localhost:8000`  
**Protocolo:** HTTP/HTTPS  
**Formato:** JSON

### Tabla de Contenidos

- 1. [Endpoints Administrativos](#)
- 2. [Endpoints de Autenticación](#)
- 3. [Endpoints de Gestión de Usuarios](#)
- 4. [Endpoints de Datos de Referencia](#)
- 5. [Endpoints de Formularios](#)
- 6. [Endpoints de Datos de Sensores](#)
- 7. [Endpoints de Análisis y Predicciones](#)
- 8. [Endpoints de Dashboard](#)
- 9. [Servicios Externos](#)

### Endpoints Administrativos

1. Admin Panel
  - **URL:** `/admin/`
  - **Método:** GET
  - **Descripción:** Panel de administración de Django para gestión de la base de datos
  - **Autenticación:** Credenciales de superusuario
  - **Uso:** Administración manual de datos, debugging
2. API Documentation (Swagger UI)
  - **URL:** `/api/docs/`
  - **Método:** GET
  - **Descripción:** Interfaz interactiva de documentación de la API con Swagger
  - **Autenticación:** No requiere
  - **Uso:** Documentación y pruebas de endpoints
3. API Documentation (ReDoc)
  - **URL:** `/api/redoc/`
  - **Método:** GET
  - **Descripción:** Documentación alternativa de la API en formato ReDoc
  - **Autenticación:** No requiere

- **Uso:** Documentación de referencia

## 4. OpenAPI Schema

- **URL:** `/api/schema/`
  - **Método:** GET
  - **Descripción:** Esquema OpenAPI 3.0 en formato JSON
  - **Autenticación:** No requiere
  - **Uso:** Generación automática de clientes de API
- 



## Endpoints de Autenticación

### 5. Registro de Usuario

- **URL:** `/api/usuarios/register/`
- **Método:** POST
- **Descripción:** Registra un nuevo usuario (consumidor o administrador)
- **Autenticación:** No requiere
- **Parámetros:**
  - `nombre` (string): Nombre completo del usuario
  - `email` (string): Correo electrónico único
  - `password` (string): Contraseña (mínimo 6 caracteres)
  - `telefono` (string): Número telefónico
  - `rol` (string): "consumidor" o "administrador"
- **Respuesta:** Datos del usuario creado con `user_id`
- **Uso:** Registro inicial de usuarios en la aplicación

### 6. Inicio de Sesión

- **URL:** `/api/usuarios/login/`
  - **Método:** POST
  - **Descripción:** Autentica un usuario y devuelve sus datos
  - **Autenticación:** No requiere (este es el endpoint de autenticación)
  - **Parámetros:**
    - `email` (string): Correo electrónico del usuario
    - `password` (string): Contraseña del usuario
  - **Respuesta:** Datos completos del usuario incluyendo información de perfil
  - **Uso:** Login de usuarios, obtención de datos de sesión
- 



## Endpoints de Gestión de Usuarios

### 7. Listar Usuarios

- **URL:** `/api/usuarios/`
- **Método:** GET
- **Descripción:** Obtiene lista de todos los usuarios registrados
- **Autenticación:** Requerida

- **Respuesta:** Array de usuarios
- **Uso:** Administración de usuarios, búsqueda

## 8. Obtener Usuario Específico

- **URL:** `/api/usuarios/{id}/`
- **Método:** GET
- **Descripción:** Obtiene información detallada de un usuario
- **Autenticación:** Requerida
- **Parámetros:** `id` (integer): ID del usuario
- **Respuesta:** Datos completos del usuario
- **Uso:** Perfil de usuario, edición

## 9. Actualizar Usuario

- **URL:** `/api/usuarios/{id}/`
- **Método:** PUT/PATCH
- **Descripción:** Actualiza información de un usuario
- **Autenticación:** Requerida
- **Parámetros:** Campos a actualizar
- **Respuesta:** Usuario actualizado
- **Uso:** Edición de perfil

## 10. Eliminar Usuario

- **URL:** `/api/usuarios/{id}/`
- **Método:** DELETE
- **Descripción:** Elimina un usuario del sistema
- **Autenticación:** Requerida (admin)
- **Respuesta:** Confirmación de eliminación
- **Uso:** Administración de usuarios

## 11. Listar Administradores

- **URL:** `/api/administradores/`
- **Método:** GET
- **Descripción:** Lista solo usuarios con rol de administrador
- **Autenticación:** Requerida
- **Respuesta:** Array de administradores
- **Uso:** Gestión administrativa

## 12. Listar Consumidores

- **URL:** `/api/consumidores/`
- **Método:** GET
- **Descripción:** Lista solo usuarios con rol de consumidor
- **Autenticación:** Requerida
- **Respuesta:** Array de consumidores con datos de salud
- **Uso:** Gestión de usuarios finales, reportes

---

## Endpoints de Datos de Referencia (Lookup Tables)

### 13. Gestión de Emociones

- **URL:** </api/emociones/>
- **Métodos:** GET, POST, PUT, DELETE
- **Descripción:** Catálogo de emociones disponibles para registro
- **Autenticación:** Requerida
- **Uso:** Selección de emociones en formularios, análisis de estado emocional

### 14. Gestión de Motivos

- **URL:** </api/motivos/>
- **Métodos:** GET, POST, PUT, DELETE
- **Descripción:** Catálogo de motivos de consumo de tabaco
- **Autenticación:** Requerida
- **Uso:** Registro de causas de recaída, análisis de patrones

### 15. Gestión de Soluciones

- **URL:** </api/soluciones/>
- **Métodos:** GET, POST, PUT, DELETE
- **Descripción:** Catálogo de soluciones sugeridas para manejo de deseos
- **Autenticación:** Requerida
- **Uso:** Recomendaciones al usuario, gestión de estrategias

### 16. Gestión de Hábitos

- **URL:** </api/habitos/>
- **Métodos:** GET, POST, PUT, DELETE
- **Descripción:** Catálogo de hábitos monitoreables (ej: cigarrillos)
- **Autenticación:** Requerida
- **Uso:** Configuración de tracking de hábitos

### 17. Gestión de Permisos

- **URL:** </api/permisos/>
- **Métodos:** GET, POST, PUT, DELETE
- **Descripción:** Catálogo de permisos de acceso al sistema
- **Autenticación:** Requerida (admin)
- **Uso:** Control de acceso, gestión de roles

---

## Endpoints de Formularios

### 18. Gestión de Formularios Completos

- **URL:** </api/formularios/>
- **Métodos:** GET, POST, PUT, DELETE

- **Descripción:** Formularios completos registrados por el usuario
- **Autenticación:** Requerida
- **Parámetros de filtro:** `?consumidor_id={id}`
- **Datos incluidos:**
  - Información de salud (edad, peso, altura, género)
  - Hábitos de consumo
  - Estado emocional
  - Lecturas de sensores asociadas
- **Uso:** Registro completo de estado del usuario, historial

## 19. Gestión de Formularios Temporales

- **URL:** `/api/formularios-temporales/`
- **Métodos:** GET, POST
- **Descripción:** Formularios en proceso de llenado (no completados)
- **Autenticación:** Requerida
- **Uso:** Guardado automático, continuación de formularios



## Endpoints de Datos de Sensores

### 20. Gestión de Ventanas de Tiempo

- **URL:** `/api/ventanas/`
- **Métodos:** GET, POST
- **Descripción:** Ventanas de 60 segundos para agrupación de lecturas de sensores
- **Autenticación:** Requerida
- **Datos incluidos:**
  - Timestamp de inicio y fin
  - ID del consumidor
  - Referencia al formulario asociado
- **Uso:** Organización temporal de datos de sensores

### 21. Gestión de Lecturas de Sensores

- **URL:** `/api/lecturas/`
- **Métodos:** GET, POST
- **Descripción:** Lecturas individuales de sensores (frecuencia cardíaca, actividad)
- **Autenticación:** Requerida
- **Datos incluidos:**
  - Frecuencia cardíaca (BPM)
  - Nivel de actividad
  - Timestamp
  - Referencia a ventana
- **Uso:** Captura de datos biométricos en tiempo real



## Endpoints de Análisis y Predicciones

## 22. Gestión de Análisis

- **URL:** `/api/analisis/`
- **Métodos:** GET, POST
- **Descripción:** Resultados de análisis de Machine Learning
- **Autenticación:** Requerida
- **Datos incluidos:**
  - Predicción de riesgo de recaída
  - Nivel de confianza
  - Timestamp del análisis
  - Referencia al formulario analizado
- **Uso:** Almacenamiento de predicciones, historial de análisis

## 23. Predicción de Deseo (ML Endpoint)

- **URL:** `/api/predict/`
- **Método:** POST
- **Descripción:** Ejecuta predicción de Machine Learning en tiempo real
- **Autenticación:** Requerida
- **Parámetros:** Datos de sensores y formulario
- **Respuesta:** Predicción inmediata + task\_id para seguimiento
- **Uso:** Análisis en tiempo real, notificaciones preventivas

## 24. Estado de Tarea Asíncrona

- **URL:** `/api/task-status/{task_id}/`
- **Método:** GET
- **Descripción:** Verifica el estado de una tarea de Celery
- **Autenticación:** Requerida
- **Parámetros:** `task_id` (string): ID de tarea de Celery
- **Respuesta:** Estado (PENDING/SUCCESS/FAILURE) y resultado
- **Uso:** Seguimiento de tareas asíncronas (ML, emails)

## 25. Gestión de Deseos

- **URL:** `/api/deseos/`
- **Métodos:** GET, POST
- **Descripción:** Registro de deseos de consumir tabaco
- **Autenticación:** Requerida
- **Endpoints adicionales:**
  - `POST /api/deseos/{id}/resolve/`: Marca deseo como resuelto
- **Datos incluidos:**
  - Intensidad del deseo
  - Fecha/hora de registro
  - Solución aplicada
  - Estado (resuelto/no resuelto)
- **Uso:** Tracking de crisis, evaluación de efectividad de estrategias

## 26. Gestión de Notificaciones

- **URL:** `/api/notificaciones/`
  - **Métodos:** GET, POST
  - **Descripción:** Notificaciones enviadas al usuario
  - **Autenticación:** Requerida
  - **Endpoints adicionales:**
    - `POST /api/notificaciones/{id}/mark-read/`: Marcar como leída
    - `POST /api/notificaciones/{id}/mark-unread/`: Marcar como no leída
  - **Datos incluidos:**
    - Mensaje de notificación
    - Tipo (alerta/recordatorio/recomendación)
    - Estado de lectura
    - Timestamp
  - **Uso:** Sistema de alertas, comunicación con usuario
- 

## Endpoints de Dashboard (Solo Lectura)

### 27. Tracking de Hábitos

- **URL:** `/api/dashboard/habit-tracking/`
- **Método:** GET
- **Descripción:** Histórico de consumo de cigarrillos por día
- **Autenticación:** Requerida
- **Filtros:** `?consumidor_id={id}`
- **Datos:** Fecha, cantidad de cigarrillos, motivo, emoción
- **Uso:** Gráficas de consumo diario

### 28. Estadísticas de Hábitos

- **URL:** `/api/dashboard/habit-stats/`
- **Método:** GET
- **Descripción:** Estadísticas agregadas de hábitos de consumo
- **Autenticación:** Requerida
- **Datos:** Total, promedio diario, tendencia
- **Uso:** KPIs de progreso del usuario

### 29. Timeline de Frecuencia Cardíaca

- **URL:** `/api/dashboard/heart-rate/`
- **Método:** GET
- **Descripción:** Historial de frecuencia cardíaca promedio por ventana
- **Autenticación:** Requerida
- **Filtros:** `?consumidor_id={id}&fecha_inicio={date}&fecha_fin={date}`
- **Datos:** Timestamp, BPM promedio
- **Uso:** Gráficas de frecuencia cardíaca en el tiempo

### 30. Estadísticas de Frecuencia Cardíaca

- **URL:** </api/dashboard/heart-rate-stats/>
- **Método:** GET
- **Descripción:** Estadísticas de frecuencia cardíaca (promedio, min, max)
- **Autenticación:** Requerida
- **Datos:** Promedio, mínimo, máximo, desviación estándar
- **Uso:** Resumen de salud cardiovascular

### 31. Timeline de Predicciones

- **URL:** </api/dashboard/predictions/>
- **Método:** GET
- **Descripción:** Historial de predicciones de ML por fecha
- **Autenticación:** Requerida
- **Datos:** Fecha, predicción, confianza
- **Uso:** Análisis de patrones de riesgo

### 32. Resumen de Predicciones

- **URL:** </api/dashboard/prediction-summary/>
- **Método:** GET
- **Descripción:** Resumen estadístico de predicciones
- **Autenticación:** Requerida
- **Datos:** Total predicciones, tasa de riesgo alto, tendencia
- **Uso:** Indicadores de progreso

### 33. Tracking de Deseos

- **URL:** </api/dashboard/desires/>
- **Método:** GET
- **Descripción:** Historial de deseos registrados
- **Autenticación:** Requerida
- **Datos:** Fecha, intensidad, solución aplicada, resuelto
- **Uso:** Gráficas de frecuencia e intensidad de deseos

### 34. Estadísticas de Deseos

- **URL:** </api/dashboard/desires-stats/>
- **Método:** GET
- **Descripción:** Métricas de deseos (total, resueltos, promedio intensidad)
- **Autenticación:** Requerida
- **Datos:** Total, % resueltos, intensidad promedio
- **Uso:** Efectividad de estrategias de manejo

### 35. Resumen Diario

- **URL:** </api/dashboard/daily-summary/>
- **Método:** GET
- **Descripción:** KPIs diarios consolidados
- **Autenticación:** Requerida



- **Datos:**
  - Cigarrillos consumidos hoy
  - Deseos registrados
  - Frecuencia cardíaca promedio
  - Predicciones de riesgo
- **Uso:** Dashboard principal del usuario

## 36. Comparación Semanal

- **URL:** `/api/dashboard/weekly-comparison/`
  - **Método:** GET
  - **Descripción:** Comparación de métricas entre semanas
  - **Autenticación:** Requerida
  - **Datos:** Semana actual vs anterior (consumo, deseos, predicciones)
  - **Uso:** Análisis de progreso semanal
- 

## Servicios Externos

### 37. Redis (Cache y Message Broker)

- **URL:** `redis://localhost:6379/0`
- **Tipo:** Servicio interno
- **Descripción:** Sistema de caché y cola de mensajes para Celery
- **Puerto:** 6379
- **Uso:**
  - Cache de datos frecuentes
  - Cola de tareas asíncronas
  - Sesiones de usuario
- **Configuración:** `.env` → `CELERY_BROKER_URL`

### 38. PostgreSQL (Base de Datos)

- **URL:** `localhost:5432`
- **Tipo:** Servicio interno
- **Descripción:** Base de datos relacional principal
- **Base de datos:** `wearable`
- **Usuario:** `postgres`
- **Uso:**
  - Almacenamiento persistente de datos
  - Relaciones entre entidades
  - Vistas materializadas para dashboard
- **Configuración:** `.env` → Variables `POSTGRES_*`

### 39. Celery Worker

- **Comando:** `celery -A WearableApi worker`
- **Tipo:** Servicio interno
- **Descripción:** Procesador de tareas asíncronas

- **Puerto:** N/A (usa Redis)
- **Uso:**
  - Predicciones de Machine Learning
  - Envío de emails
  - Tareas en segundo plano
- **Configuración:** `WearableApi/celery.py`

## 40. Celery Beat

- **Comando:** `celery -A WearableApi beat`
- **Tipo:** Servicio interno
- **Descripción:** Programador de tareas periódicas
- **Puerto:** N/A (usa Redis)
- **Uso:**
  - Análisis periódicos
  - Notificaciones programadas
  - Limpieza de datos antiguos
- **Configuración:** `django_celery_beat` (DB)

## 41. SendGrid (Email Service)

- **URL:** `https://api.sendgrid.com/v3/`
- **Tipo:** Servicio externo (API)
- **Descripción:** Servicio de envío de emails transaccionales
- **Autenticación:** API Key
- **Uso:**
  - Confirmación de registro
  - Notificaciones críticas por email
  - Alertas de riesgo alto
- **Configuración:** `.env` → `SENDGRID_API_KEY`

## 42. Sentry (Error Tracking)

- **URL:** `https://sentry.io/`
- **Tipo:** Servicio externo (monitoring)
- **Descripción:** Monitoreo de errores y performance
- **Autenticación:** DSN
- **Uso:**
  - Tracking de excepciones
  - Monitoreo de performance
  - Alertas de errores en producción
- **Configuración:** `.env` → `SENTRY_DSN`

---

## Resumen de Endpoints

Por Categoría:

- **Administrativos:** 4 endpoints

- **Autenticación:** 2 endpoints
- **Gestión de Usuarios:** 6 endpoints
- **Datos de Referencia:** 5 endpoints
- **Formularios:** 2 endpoints
- **Datos de Sensores:** 2 endpoints
- **Análisis y Predicciones:** 5 endpoints
- **Dashboard:** 10 endpoints (solo lectura)

**Total Endpoints API:** 36 endpoints principales

**Servicios Externos:** 6 servicios

---

## Seguridad

### Autenticación

- Sistema basado en credenciales (email/password)
- Contraseñas hasheadas con Django's `make_password`
- Validación de email único en registro

### Autorización

- Endpoints protegidos requieren autenticación
- Separación de roles (consumidor/administrador)
- Permisos granulares por endpoint

### Datos Sensibles

- API Keys almacenadas en variables de entorno (`.env`)
  - Nunca expuestas en código o repositorio
  - HTTPS en producción
- 

## Consideraciones de Despliegue

### Desarrollo Local

- Django dev server: `python manage.py runserver`
- `DEBUG=True`
- HTTP permitido
- Base URL: `http://localhost:8000`

### Producción

- WSGI server (Gunicorn)
- `DEBUG=False`
- HTTPS forzado
- Cache habilitado con Redis
- Celery workers escalables
- Base URL: `https://yourdomain.com`



## Notas Adicionales

1. **Paginación:** Todos los endpoints de lista soportan paginación (50 items por página)
2. **Filtrado:** Endpoints de dashboard soportan filtrado por `consumidor_id` y fechas
3. **Ordenamiento:** Datos ordenados por timestamp descendente por defecto
4. **Formato de Fecha:** ISO 8601 (`YYYY-MM-DDTHH:MM:SS`)
5. **Respuestas de Error:** Formato consistente con código HTTP y mensaje
6. **CORS:** Habilitado para desarrollo, configurar origins específicos en producción