

# Documento de Requisitos del Sistema

## Inventario Emocional - Al-Anon

### 1. Introducción

#### 1.1 Propósito

Este documento especifica los requisitos funcionales y no funcionales del sistema **Inventario Emocional**, una aplicación móvil diseñada para miembros de Al-Anon que permite registrar y monitorear estados emocionales diarios, facilitando el proceso de autoconocimiento y recuperación emocional.

#### 1.2 Alcance

El sistema proporcionará:

- Registro y autenticación de usuarios (miembros y administradores)
- Formulario estructurado para el inventario emocional diario
- Consulta de historial de inventarios
- Visualización de estadísticas y patrones emocionales
- Sistema de chat en tiempo real para soporte comunitario
- Panel administrativo para gestión de usuarios y moderación

#### 1.3 Definiciones y Acrónimos

- Al-Anon:** Organización de apoyo a familiares y amigos de personas con problemas de alcoholismo
- Inventario Emocional:** Registro estructurado de emociones, resentimientos, miedos y defectos de carácter
- CRUD:** Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar)
- MVC:** Model-View-Controller (Modelo-Vista-Controlador)
- API:** Application Programming Interface

## 2. Descripción General

---

### 2.1 Perspectiva del Producto

Inventario Emocional es una aplicación móvil independiente que utiliza:

- Framework Kivy para desarrollo multiplataforma (Android/iOS)
- Base de datos MongoDB para almacenamiento persistente
- Arquitectura MVC para separación de responsabilidades

### 2.2 Funciones del Producto

1. **Gestión de Usuarios:** Registro, autenticación y administración
2. **Inventario Emocional:** Captura estructurada de datos emocionales
3. **Análisis y Visualización:** Estadísticas y gráficos de patrones emocionales
4. **Comunicación:** Chat en tiempo real entre miembros
5. **Administración:** Gestión de usuarios y moderación de contenido

### 2.3 Características de los Usuarios

#### Miembro

- **Descripción:** Usuario principal del sistema
- **Nivel técnico:** Básico a intermedio
- **Actividades principales:**
  - Registrar inventario emocional diario
  - Consultar historial personal
  - Visualizar estadísticas
  - Participar en chat grupal

#### Administrador

- **Descripción:** Usuario con privilegios de gestión
- **Nivel técnico:** Intermedio a avanzado
- **Actividades principales:**
  - Gestionar cuentas de usuarios
  - Moderar chat
  - Supervisar uso del sistema

### 3. Requisitos Funcionales

---

#### RF-01: Registro de Usuario

**Prioridad:** Alta

**Descripción:** El sistema debe permitir a nuevos usuarios crear una cuenta en la aplicación.

**Entradas:**

- Nombre completo
- Correo electrónico
- Contraseña (mínimo 8 caracteres)
- Confirmación de contraseña

**Proceso:**

1. Validar que el correo electrónico no esté registrado
2. Validar formato de correo electrónico
3. Verificar que las contraseñas coincidan
4. Encriptar contraseña
5. Crear registro en base de datos
6. Asignar rol de "miembro" por defecto

**Salidas:**

- Confirmación de registro exitoso
- Redirección a pantalla de login

**Restricciones:**

- El correo electrónico debe ser único
- La contraseña debe cumplir requisitos de seguridad

---

#### RF-02: Iniciar Sesión

**Prioridad:** Alta

**Descripción:** El sistema debe autenticar usuarios mediante credenciales y diferenciar entre miembros y administradores.

**Entradas:**

- Correo electrónico
- Contraseña

- Tipo de usuario (miembro/administrador)

**Proceso:**

1. Verificar existencia del usuario
2. Validar contraseña encriptada
3. Verificar rol del usuario
4. Crear sesión activa
5. Redirigir según rol

**Salidas:**

- Acceso al dashboard correspondiente
- Token de sesión

**Restricciones:**

- Máximo 3 intentos fallidos antes de bloqueo temporal
  - Sesión expira después de 24 horas de inactividad
- 

## RF-03: Registrar Inventario Emocional

**Prioridad:** Alta

**Descripción:** Los miembros deben poder completar un formulario estructurado de inventario emocional.

**Entradas:**

- Fecha del inventario
- Resentimientos (persona/situación, causa, afecta a)
- Defectos de carácter identificados
- Miedos reconocidos
- Parte de responsabilidad propia

**Proceso:**

1. Presentar formulario con campos estructurados
2. Validar que todos los campos requeridos estén completos
3. Asociar inventario al usuario autenticado
4. Registrar fecha y hora de creación
5. Guardar en base de datos

**Salidas:**

- Confirmación de registro guardado
- Actualización de historial
- Disponibilidad para estadísticas

**Restricciones:**

- Solo usuarios autenticados pueden registrar
  - Un inventario por día por usuario
- 

## RF-04: Consultar Historial

**Prioridad:** Media

**Descripción:** Los miembros deben poder visualizar sus inventarios emocionales anteriores.

**Entradas:**

- Usuario autenticado
- Filtros opcionales (rango de fechas)

**Proceso:**

1. Recuperar inventarios del usuario desde la base de datos
2. Ordenar por fecha (más reciente primero)
3. Aplicar filtros si se especifican
4. Presentar lista con resúmenes
5. Permitir selección para ver detalle completo

**Salidas:**

- Lista de inventarios
- Vista detallada al seleccionar

**Restricciones:**

- Solo se muestran inventarios del usuario actual
  - Máximo 100 registros por consulta
- 

## RF-05: Ver Estadísticas

**Prioridad:** Media

**Descripción:** El sistema debe generar visualizaciones gráficas de los patrones emocionales del usuario.

**Entradas:**

- Usuario autenticado
- Rango de fechas (opcional)

**Proceso:**

1. Recuperar inventarios del período seleccionado
2. Analizar frecuencia de emociones
3. Identificar patrones recurrentes
4. Generar gráficos (líneas, barras, circular)
5. Calcular tendencias

**Salidas:**

- Gráfico de emociones en el tiempo
- Estadísticas de defectos de carácter más frecuentes
- Análisis de miedos recurrentes
- Resumen de progreso

**Restricciones:**

- Requiere mínimo 3 inventarios para generar estadísticas
  - Gráficos limitados a últimos 90 días
- 

## RF-06: Chat en Tiempo Real

**Prioridad:** Alta

**Descripción:** Los miembros deben poder comunicarse mediante un chat general.

**Entradas:**

- Mensaje de texto (máximo 500 caracteres)
- Usuario emisor

**Proceso:**

1. Validar usuario autenticado
2. Validar longitud y contenido del mensaje
3. Registrar mensaje en base de datos
4. Transmitir a todos los usuarios conectados
5. Actualizar interfaz en tiempo real

**Salidas:**

- Mensaje visible para todos los usuarios
- Notificación de nuevo mensaje

**Restricciones:**

- Solo miembros autenticados pueden enviar mensajes
  - Máximo 10 mensajes por minuto por usuario
  - Mensajes ofensivos serán moderados
- 

## RF-07: Gestionar Usuarios (Administrador)

**Prioridad:** Media

**Descripción:** Los administradores deben poder gestionar cuentas de usuarios.

**Entradas:**

- Acción (listar, editar, eliminar, bloquear)
- ID de usuario objetivo

**Proceso:**

1. Verificar permisos de administrador
2. Ejecutar acción solicitada
3. Registrar cambio en log de auditoría
4. Actualizar base de datos

**Salidas:**

- Confirmación de acción realizada
- Lista actualizada de usuarios

**Restricciones:**

- Solo administradores pueden ejecutar estas acciones
  - No se puede eliminar la propia cuenta de administrador
- 

## RF-08: Moderar Chat (Administrador)

**Prioridad:** Media

**Descripción:** Los administradores deben poder moderar el contenido del chat.

**Entradas:**

- ID del mensaje

- Acción (eliminar, advertir usuario)

**Proceso:**

1. Verificar permisos de administrador
2. Ejecutar acción de moderación
3. Notificar al usuario si corresponde
4. Registrar en log de moderación

**Salidas:**

- Mensaje eliminado o usuario advertido
- Registro de acción de moderación

**Restricciones:**

- Solo administradores pueden moderar
  - Todas las acciones quedan registradas
- 

## 4. Requisitos No Funcionales

---

### RNF-01: Usabilidad

**Descripción:** La interfaz debe ser intuitiva y fácil de usar.

**Criterios de aceptación:**

- Usuarios nuevos pueden registrarse en menos de 3 minutos
  - Completar un inventario toma menos de 10 minutos
  - Navegación clara con máximo 3 clics para cualquier función
  - Diseño adaptable a diferentes tamaños de pantalla
- 

### RNF-02: Rendimiento

**Descripción:** El sistema debe responder de manera ágil.

**Criterios de aceptación:**

- Tiempo de carga inicial: < 3 segundos
- Tiempo de respuesta para operaciones comunes: < 2 segundos
- Carga de historial: < 1 segundo
- Actualización de chat en tiempo real: < 500ms

---

## RNF-03: Seguridad

**Descripción:** Protección de datos sensibles y privacidad del usuario.

**Criterios de aceptación:**

- Contrasenñas encriptadas con bcrypt o similar
  - Comunicación mediante HTTPS
  - Sesiones con tokens seguros
  - Datos personales no compartidos con terceros
  - Cumplimiento de normativas de privacidad
- 

## RNF-04: Disponibilidad

**Descripción:** El sistema debe estar disponible cuando los usuarios lo necesiten.

**Criterios de aceptación:**

- Disponibilidad: 99% uptime
  - Mantenimientos programados en horarios de baja demanda
  - Sistema de respaldo automático diario
  - Recuperación ante fallos en menos de 1 hora
- 

## RNF-05: Escalabilidad

**Descripción:** Capacidad de crecer según aumente el número de usuarios.

**Criterios de aceptación:**

- Soportar hasta 10,000 usuarios simultáneos
  - Base de datos escalable horizontalmente
  - Arquitectura modular para agregar funcionalidades
  - Optimización de consultas para grandes volúmenes
- 

## RNF-06: Compatibilidad

**Descripción:** Funcionalidad en diferentes plataformas.

**Criterios de aceptación:**

- Compatible con Android 8.0 o superior

- Compatible con iOS 12 o superior
  - Diseño responsive para tablets
  - Funcionamiento con conexiones 3G/4G/WiFi
- 

## RNF-07: Mantenibilidad

**Descripción:** Facilidad para mantener y actualizar el sistema.

**Criterios de aceptación:**

- Código documentado con comentarios claros
  - Arquitectura MVC bien definida
  - Pruebas unitarias con cobertura > 70%
  - Versionado con Git
  - Logs detallados de errores
- 

## RNF-08: Portabilidad

**Descripción:** Facilidad para migrar o desplegar en diferentes entornos.

**Criterios de aceptación:**

- Configuración mediante variables de entorno
  - Base de datos independiente de la lógica de negocio
  - Despliegue mediante contenedores Docker
  - Documentación de instalación completa
- 

## 5. Restricciones del Sistema

---

### Restricciones Tecnológicas

- Desarrollo en Python 3.8+
- Framework Kivy para la interfaz
- Base de datos MongoDB
- Requiere conexión a Internet para funcionalidades en tiempo real

### Restricciones de Negocio

- Aplicación gratuita para miembros de Al-Anon
- No se almacenará información médica sensible
- Cumplimiento con lineamientos de Al-Anon

## Restricciones de Diseño

- Arquitectura MVC obligatoria
  - Separación clara entre capas
  - Interfaz minimalista y accesible
- 

## 6. Criterios de Aceptación

Para que el sistema sea aceptado, debe cumplir:

1.  Todos los requisitos funcionales de prioridad Alta implementados
  2.  Al menos 80% de requisitos funcionales de prioridad Media
  3.  Cumplimiento de todos los requisitos no funcionales críticos (seguridad, rendimiento)
  4.  Pruebas de usabilidad exitosas con usuarios reales
  5.  Documentación completa (técnica y usuario)
  6.  Sin errores críticos en producción
  7.  Tiempo de respuesta dentro de parámetros establecidos
- 

## 7. Apéndices

### A. Casos de Uso Relacionados

- CU-01: Registro de Usuario
- CU-02: Iniciar Sesión
- CU-03: Registrar Inventario Emocional
- CU-04: Consultar Historial
- CU-05: Ver Estadísticas
- CU-06: Participar en Chat
- CU-07: Gestionar Usuarios
- CU-08: Moderar Chat

## B. Referencias

- Documentación oficial de Kivy: <https://kivy.org/doc/stable/>
  - MongoDB Documentation: <https://docs.mongodb.com/>
  - AI-Anon Guidelines: <https://al-anon.org/>
- 

**Versión:** 1.0

**Fecha:** Diciembre 2024

**Estado:** Aprobado