Especificaciones Técnicas - Triqueta Digital

Proyecto: Triqueta Digital
Versión: 1.0
Fecha: Octubre 2025
Estado: Aprobado para desarrollo

Tabla de Contenidos

---

1. Diagrama de Casos de Uso

1.1 Descripción

El diagrama de casos de uso representa visualmente las interacciones principales entre los actores del sistema (Usuario y Administrador) y las funcionalidades que ofrece la plataforma Triqueta Digital. Este diagrama proporciona una vista de alto nivel del sistema, identificando:

- **Actores principales:**
  - **Usuario:** Persona que utiliza la plataforma para explorar, buscar y guardar actividades culturales, deportivas y recreativas.
  - **Administrador:** Usuario con permisos especiales para gestionar el contenido, usuarios y procesos de ingesta de datos.
- **Módulos funcionales:**
  - **Exploración:** Permite ver detalles de actividades, recibir recomendaciones, buscar y explorar el catálogo completo.
  - **Autenticación:** Gestiona el registro, inicio de sesión y cierre de sesión de usuarios.
  - **Perfil:** Permite visualizar y gestionar favoritos, así como administrar el perfil personal.
  - **Administración:** Funcionalidades exclusivas para monitorear el dashboard, crear, editar, eliminar e importar actividades.
- **Relaciones:**
  - Las relaciones `<<extends>>` indican funcionalidades que extienden el comportamiento de un caso de uso base.
  - Las relaciones `<<include>>` muestran dependencias entre casos de uso, donde uno incluye necesariamente a otro.

El diagrama completo se muestra en la **Figura 1** (ver más adelante).

---

2. Historias de Usuario

Total de Historias: 35

Índice de Módulos

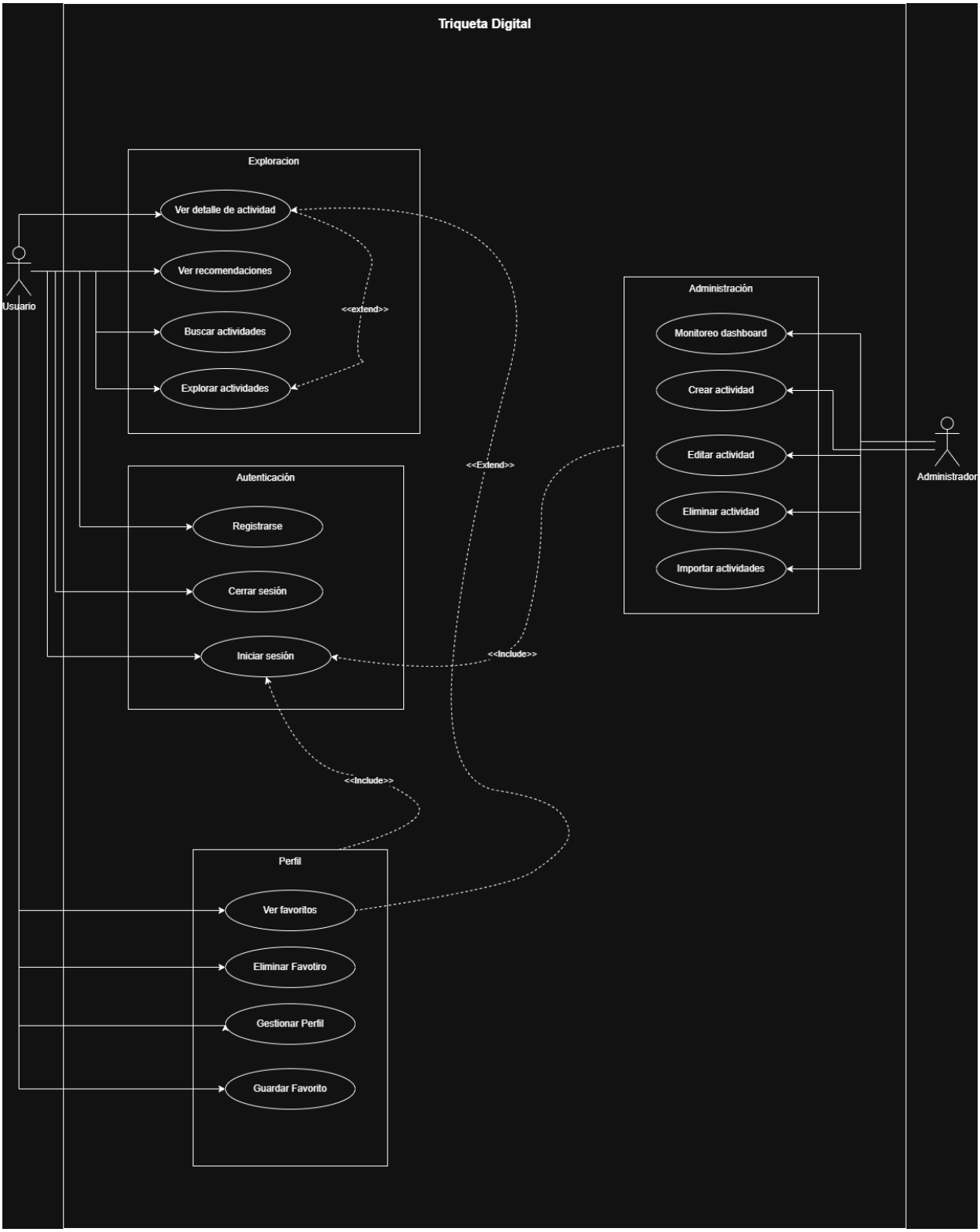

Figure 1: Diagrama de Casos de Uso - Triqueta Digital

5. Perfil de Usuario - 2 historias
6. Administración - 9 historias
7. Gestión de Ingesta (ETL) - 4 historias
8. Dispositivos IoT - 2 historias

2.1 Autenticación y Gestión de Usuarios

HU-001: Registro de Usuario **Como:** Usuario no autenticado
Quiero: Registrarme en la plataforma con mi email y contraseña
Para: Crear una cuenta y acceder a las funcionalidades de la plataforma

Prioridad: Alta

Criterios de aceptación: - El formulario solicita: email, contraseña, confirmación de contraseña y nombre completo - El email debe tener formato válido y no estar registrado - La contraseña debe cumplir requisitos: mínimo 8 caracteres, 1 mayúscula, 1 número, 1 carácter especial - Se debe aceptar términos y condiciones - Al registrarse se generan access token y refresh token automáticamente - Se redirige a la página principal con mensaje de confirmación - Errores se muestran de forma descriptiva

---

HU-002: Inicio de Sesión **Como:** Usuario registrado
Quiero: Iniciar sesión con mi email y contraseña
Para: Acceder a mi cuenta y utilizar la plataforma

Prioridad: Alta

Criterios de aceptación: - El formulario solicita email y contraseña - Si las credenciales son correctas, se generan tokens (access 30 min, refresh 30 días) - Se redirige a la página principal con mensaje de bienvenida - Credenciales incorrectas muestran mensaje de error genérico - Rate limiting: máximo 5 intentos cada 15 minutos - Después de 5 intentos, se bloquea temporalmente el acceso

---

HU-003: Cierre de Sesión **Como:** Usuario registrado
Quiero: Cerrar mi sesión de forma segura
Para: Proteger mi cuenta cuando termine de usar la plataforma

Prioridad: Media

Criterios de aceptación: - Botón “Cerrar sesión” visible en header/menú - Al hacer clic, se revoca el refresh token - Se elimina información de sesión del cliente - Usuario es redirigido a página de inicio con mensaje de confirmación - No se puede acceder a páginas protegidas después del logout

---

HU-004: Renovación Automática de Sesión **Como:** Usuario registrado
Quiero: Que mi sesión se renueve automáticamente
Para: No tener que iniciar sesión constantemente

Prioridad: Alta

Criterios de aceptación: - Cuando el access token está por expirar (<5 min), se renueva automáticamente - El sistema usa el refresh token para obtener nuevo access token - El proceso es transparente (sin interrumpir navegación) - Si refresh token ha expirado, se redirige al login - Las peticiones en curso no fallan durante renovación

---

HU-005: Recuperación de Contraseña **Como:** Usuario registrado
Quiero: Recuperar mi contraseña si la olvido
Para: Poder acceder nuevamente a mi cuenta

Prioridad: Media

Criterios de aceptación: - Enlace "¿Olvidaste tu contraseña?" en página de login - Usuario ingresa su email registrado - Sistema envía email con enlace de recuperación (válido 1 hora) - El enlace redirige a página para establecer nueva contraseña - Nueva contraseña debe cumplir requisitos de seguridad - Al cambiar contraseña, se invalidan todos los refresh tokens previos - Se muestra mensaje de confirmación

---

2.2 Actividades

HU-006: Explorar Actividades **Como:** Usuario no autenticado
Quiero: Ver un listado de actividades culturales, recreativas y deportivas
Para: Conocer la oferta disponible en mi localidad

Prioridad: Alta

Criterios de aceptación: - Página muestra listado en formato de tarjetas (grid responsive) - Cada tarjeta: título, imagen, localidad, tipo, fecha, precio, etiquetas - Paginación de 20 actividades por página - Se muestra total de actividades encontradas - Skeleton loaders mientras carga - Lazy loading de imágenes - Mensaje informativo si no hay actividades

---

HU-007: Filtrar Actividades **Como:** Usuario no autenticado
Quiero: Filtrar actividades por tipo, localidad, fecha y precio
Para: Encontrar actividades que se ajusten a mis preferencias

Prioridad: Alta

Criterios de aceptación: - Panel lateral/sheet con filtros: tipo, localidad, rango de fechas, precio, actividades gratuitas, nivel, etiquetas - Los filtros se aplican individualmente o en combinación - Listado se actualiza automáticamente al aplicar filtros - Se muestra número de resultados encontrados - Filtros aplicados son visibles (badges/tags) - Botón "Limpiar filtros" para resetear - Filtros persisten al navegar entre páginas - En móvil, filtros en drawer/sheet

---

HU-008: Buscar Actividades **Como:** Usuario no autenticado
Quiero: Buscar actividades por palabras clave
Para: Encontrar rápidamente actividades específicas

Prioridad: Alta

Criterios de aceptación: - Barra de búsqueda visible en página de actividades - Búsqueda en: título, descripción y etiquetas - Resultados ordenados por relevancia - Búsqueda case-insensitive y sin acentos - Coincidencias parciales permitidas - Se puede combinar búsqueda con filtros - Mensaje si no hay resultados

---

HU-009: Ver Detalle de Actividad **Como:** Usuario no autenticado
Quiero: Ver información completa de una actividad
Para: Conocer todos los detalles antes de participar

Prioridad: Alta

Criterios de aceptación: - Página muestra: título, imagen, descripción completa, tipo, fecha/hora, ubicación, localidad, precio, nivel, etiquetas, contacto, enlace externo, fuente - Botón "Volver" a la lista - Responsive en todos los dispositivos - Se registra la vista (para popularidad)

HU-010: Crear Actividad (Admin) **Como:** Administrador
Quiero: Crear una nueva actividad manualmente
Para: Agregar ofertas al catálogo

Prioridad: Alta

Criterios de aceptación: - Solo rol "administrador" puede acceder - Botón "Nueva Actividad" en panel admin - Formulario con campos obligatorios: título, descripción, tipo, fecha, ubicación, localidad, precio, nivel, etiquetas (mín 1) - Campos opcionales: fecha fin, contacto, enlace - Validación en tiempo real - Vista previa antes de guardar - Actividad se crea con estado "activa" - Mensaje de confirmación y redirección a listado

---

HU-011: Editar Actividad (Admin) **Como:** Administrador
Quiero: Editar información de una actividad existente
Para: Corregir errores o actualizar datos

Prioridad: Alta

Criterios de aceptación: - Solo rol "administrador" puede acceder - Botón "Editar" en cada actividad del listado admin - Formulario precargado con datos actuales - Se pueden modificar todos los campos - Validaciones aplican igual que al crear - Botón "Cancelar" descarta cambios - Al guardar, se actualiza "updated_at" - Mensaje de confirmación y cambios reflejados inmediatamente

---

HU-012: Eliminar Actividad (Admin) **Como:** Administrador
Quiero: Eliminar actividades obsoletas o incorrectas
Para: Mantener catálogo actualizado

Prioridad: Media

Criterios de aceptación: - Solo rol "administrador" puede acceder - Botón "Eliminar" en cada actividad - Dialog de confirmación explicando soft delete - Al confirmar, actividad cambia estado a "inactiva" - NO se elimina físicamente de BD - Actividades inactivas no aparecen en búsquedas públicas - Se pueden ver en panel admin con filtro especial - Opción para "restaurar" actividades inactivas

---

HU-013: Importar Actividades CSV/JSON (Admin) **Como:** Administrador
Quiero: Importar actividades masivamente desde archivo
Para: Agregar múltiples actividades eficientemente

Prioridad: Media

Criterios de aceptación: - Solo rol "administrador" puede acceder - Página /admin/actividades/importar con documentación de formato - Drag & drop o selección de archivo (.csv, .json) - Barra de progreso durante procesamiento - Sistema valida campos, formatos y detecta duplicados - Resumen al finalizar: total, exitosos, duplicados, errores (con detalle) - Actividades importadas con estado "pendiente_validacion" - Opción para descargar archivo de errores

---

2.3 Favoritos

HU-014: Guardar Actividad como Favorita **Como:** Usuario registrado
Quiero: Marcar una actividad como favorita
Para: Guardarla y acceder fácilmente después

Prioridad: Alta

Criterios de aceptación: - Solo usuarios autenticados pueden guardar - Botón de "corazón" en cada tarjeta - Al hacer clic, se guarda como favorita - Ícono cambia visualmente (relleno/color) - Notificación breve "Añadido a favoritos" - Si no está autenticado, pide iniciar sesión - No se permiten duplicados - Acción reversible

---

HU-015: Ver Mis Favoritos **Como:** Usuario registrado
Quiero: Ver listado de mis actividades favoritas
Para: Revisar actividades que me interesan

Prioridad: Alta

Criterios de aceptación: - Solo usuarios autenticados acceden a /favoritos - Muestra todas las actividades guardadas - Formato de tarjetas igual que listado general - Ordenadas por fecha de guardado (recientes primero) - Incluye fecha en que se guardó - Total de favoritos visible - Mensaje si no hay favoritos invitando a explorar - Actividades inactivas no se muestran - Botón "Quitar de favoritos" en cada tarjeta - Responsive

---

HU-016: Quitar Actividad de Favoritos **Como:** Usuario registrado
Quiero: Eliminar actividad de mis favoritos
Para: Mantener lista actualizada

Prioridad: Media

Criterios de aceptación: - Botón de favorito permite quitar favoritos - Al hacer clic en actividad favorita, se elimina - Ícono cambia visualmente (vacío/color diferente) - Notificación "Eliminado de favoritos" - Cambio inmediato en todas las vistas - En página de favoritos, botón "Quitar" disponible - Tarjeta desaparece con animación - No requiere confirmación (es reversible)

2.4 Recomendaciones (IA)

HU-017: Recibir Recomendaciones Personalizadas **Como:** Usuario registrado
Quiero: Recibir recomendaciones basadas en mis intereses
Para: Descubrir actividades relevantes automáticamente

Prioridad: Alta

Criterios de aceptación: - Solo usuarios autenticados acceden a /recomendaciones - Sistema usa algoritmo híbrido: popularidad, etiquetas, localidad, disponibilidad - Se muestran 10 recomendaciones por defecto - Actividades favoritas NO se recomiendan - Cada recomendación: información de actividad, score (opcional), explicación breve - Se pueden refrescar recomendaciones - Si usuario sin etiquetas, recomendaciones por popularidad - Responsive y cacheadas para performance

---

HU-018: Ver Explicación de Recomendación **Como:** Usuario registrado
Quiero: Entender por qué se recomienda una actividad
Para: Confiar en las recomendaciones

Prioridad: Media

Criterios de aceptación: - Cada tarjeta incluye explicación breve - Explicación menciona: etiquetas coincidentes, localidad, popularidad, disponibilidad - Lenguaje claro y natural (no técnico) - Se pueden ver múltiples razones si aplican - Opcional: ícono de información para expandir/colapsar

---

2.5 Perfil de Usuario

HU-019: Ver Mi Perfil **Como:** Usuario registrado
Quiero: Ver mi información de perfil
Para: Revisar mis datos y preferencias

Prioridad: Media

Criterios de aceptación: - Solo usuarios autenticados acceden a /profile - Muestra: nombre, email, foto, etiquetas de interés, localidad preferida, disponibilidad horaria, nivel de actividad - Botón "Editar perfil" - Estadísticas: total de favoritos, actividades vistas - Responsive

---

HU-020: Editar Mi Perfil **Como:** Usuario registrado
Quiero: Actualizar mi información de perfil
Para: Personalizar mi experiencia en la plataforma

Prioridad: Alta

Criterios de aceptación: - Formulario editable con: nombre, foto URL, etiquetas de interés, localidad preferida, disponibilidad horaria, nivel de actividad - Validación en tiempo real - Etiquetas validadas contra lista permitida - Botón "Guardar" y "Cancelar" - Al guardar, perfil actualizado inmediatamente - Mensaje de confirmación - Solo usuario propietario puede editar su perfil

---

2.6 Administración

HU-021: Ver Dashboard de Administración **Como:** Administrador
Quiero: Ver métricas clave en un dashboard
Para: Monitorear estado y uso de la plataforma

Prioridad: Media

Criterios de aceptación: - Solo rol "administrador" accede a /admin/dashboard - Métricas: usuarios (total, activos 7d/30d, nuevos), actividades (total, por localidad, por tipo, pendientes), interacción (favoritos, top 10 actividades, top 10 etiquetas), sistema (última ejecución ETL, búsquedas recientes) - Gráficos visuales (torta, barras) - Actualización cada 5 minutos (caché) - Botón refrescar manual - Responsive con iconos descriptivos

---

HU-022: Gestionar Usuarios (Admin) **Como:** Administrador
Quiero: Ver y gestionar usuarios
Para: Administrar roles y estados de cuentas

Prioridad: Baja

Criterios de aceptación: - Solo rol "administrador" accede a /admin/usuarios - Listado con: email, nombre, rol, estado, fecha registro, último acceso - Paginado y filtrable por rol y estado - Búsqueda por email o nombre - Acciones: ver detalle, cambiar rol, activar/desactivar - Acciones requieren confirmación - Admin NO puede autodesactivarse ni quitarse su rol - Logs de auditoría de acciones - Mensaje de confirmación

---

HU-023: Ver Detalle de Usuario (Admin) **Como:** Administrador
Quiero: Ver información detallada de un usuario
Para: Entender su actividad

Prioridad: Baja

Criterios de aceptación: - Al hacer clic en usuario, se abre perfil detallado - Muestra: info básica, perfil, estadísticas (favoritos, actividades vistas), historial reciente, fechas clave - Botón volver al listado - Se pueden realizar acciones desde el detalle

---

HU-024: Validar Actividades Importadas (Admin) **Como:** Administrador
Quiero: Revisar y validar actividades importadas
Para: Asegurar calidad de datos antes de publicar

Prioridad: Media

Criterios de aceptación: - Solo rol "administrador" accede a /admin/validacion - Muestra actividades "pendiente_validacion" - Información completa en formato expandible - Indica fuente de datos - Acciones: aprobar, editar y aprobar, rechazar (con nota) - Acciones en lote (selección múltiple) - Filtro por fuente - Conteo de pendientes - Actividades procesadas desaparecen del listado - Actividades rechazadas guardadas para auditoría

---

HU-025: Ver Estado de Procesos ETL (Admin) **Como:** Administrador
Quiero: Ver estado de procesos ETL
Para: Monitorear ingesta de datos

Prioridad: Media

Criterios de aceptación: - Solo rol "administrador" accede a /admin/etl - Info última ejecución: fecha/hora inicio/fin, duración, estado, totales (procesadas, exitosas, errores), lista de errores - Lista de fuentes de datos con estado - Próxima ejecución programada - Historial de ejecuciones (tabla) - Si proceso corriendo, se muestra en tiempo real - Botón ver logs detallados

---

HU-026: Ejecutar Proceso ETL Manual (Admin) **Como:** Administrador
Quiero: Ejecutar manualmente proceso ETL
Para: Actualizar catálogo bajo demanda

Prioridad: Media

Criterios de aceptación: - Solo rol "administrador" puede ejecutar - Botón "Ejecutar ETL" en página gestión ETL - Verifica que no haya otro proceso corriendo - Al confirmar, inicia proceso en background - Notificación "Proceso iniciado" - Redirección a vista de monitoreo en tiempo real - Muestra: estado actual, progreso, logs - Timeout 1 hora máximo - Al finalizar, resumen completo - Registro en tabla de ejecuciones

---

HU-027: Ver Logs de Ejecución ETL (Admin) **Como:** Administrador
Quiero: Ver logs detallados de ejecución ETL
Para: Diagnosticar problemas

Prioridad: Baja

Criterios de aceptación: - Desde historial, clic para ver logs - Formato texto estructurado con: timestamp, nivel, mensaje, detalles (expandible) - Filtro por nivel - Búsqueda de mensajes específicos - Errores destacados visualmente - Descarga de log completo (.txt) - Solo lectura

---

HU-028: Configurar Fuentes de Datos ETL (Admin) **Como:** Administrador
Quiero: Configurar fuentes de datos ETL
Para: Definir de dónde se obtienen actividades

Prioridad: Baja

Criterios de aceptación: - Solo rol "administrador" accede - Página configuración fuentes ETL - Se agregan: API REST, CSV público, JSON público - Por fuente se configura: nombre, tipo, URL, credenciales, mapeo de campos, estado - Prueba de conexión antes de guardar - Fuentes editables o desactivables - Solo fuentes activas se procesan - Validación de URL accesible

---

HU-029: Programar Ejecuciones ETL Automáticas (Admin) **Como:** Administrador
Quiero: Programar ejecuciones automáticas ETL
Para: Mantener catálogo actualizado sin intervención

Prioridad: Baja

Criterios de aceptación: - Solo rol "administrador" accede - Sección de programación en configuración ETL - Configurable: frecuencia (diaria, semanal, mensual, cron), hora, días semana, zona horaria - Muestra próxima ejecución calculada - Activar/desactivar programación - Editable - Validación de conflictos - Ejecuciones programadas en logs

---

2.7 Gestión de Ingesta (ETL)

HU-030: Extractor de API IDRD **Como:** Sistema
Quiero: Extraer datos de actividades desde API IDRD
Para: Incorporar oferta deportiva automáticamente

Prioridad: Media

Criterios de aceptación: - Script ETL se conecta a API IDRD - Extrae actividades deportivas disponibles - Maneja errores de conexión y timeout - Logs detallados del proceso - Datos extraídos pasan a transformer

---

HU-031: Transformer de Datos ETL **Como:** Sistema
Quiero: Normalizar y transformar datos extraídos
Para: Adaptarlos al modelo de datos de la plataforma

Prioridad: Media

Criterios de aceptación: - Transforma campos al formato interno - Valida tipos de datos y formatos - Normaliza fechas, coordenadas, texto - Detecta y marca duplicados - Genera logs de transformaciones - Datos transformados pasan a loader

---

HU-032: Loader de Datos ETL **Como:** Sistema
Quiero: Cargar datos transformados en la base de datos
Para: Incorporarlos al catálogo de actividades

Prioridad: Media

Criterios de aceptación: - Inserta actividades en PostgreSQL - Actividades insertadas con estado "pendiente_validacion" - Maneja errores de inserción - Previene duplicados - Actualiza contadores (procesadas, exitosas, errores) - Registra ejecución en tabla etl_executions

---

HU-033: Reporte de Ejecución ETL **Como:** Sistema
Quiero: Generar reporte detallado de cada ejecución ETL
Para: Proveer información a administradores

Prioridad: Baja

Criterios de aceptación: - Genera resumen con totales - Lista detallada de errores - Timestamp inicio/fin - Estado final - Logs completos guardados - Reporte accesible desde panel admin

---

2.8 Dispositivos IoT

HU-034: Vincular Dispositivo IoT **Como:** Usuario registrado
Quiero: Vincular un dispositivo IoT a mi cuenta
Para: Sincronizar recomendaciones con el dispositivo físico

Prioridad: Baja

Criterios de aceptación: - Solo usuarios autenticados acceden - Página /dispositivos con formulario de vinculación - Usuario ingresa código único del dispositivo (ej: TRIQ-ABC123) - Sistema valida que código existe y no está vinculado - Se crea relación en tabla usuarios_dispositivos - Se genera token JWT de dispositivo (válido 1 año) - Se muestra información del dispositivo vinculado - Botón para desvincular dispositivo - Un dispositivo solo vinculado a un usuario a la vez

---

HU-035: API de Sincronización para Dispositivo IoT **Como:** Dispositivo IoT
Quiero: Obtener recomendaciones del usuario vinculado
Para: Mostrarlas en el dispositivo físico

Prioridad: Baja

Criterios de aceptación: - Endpoint GET /api/v1/iot/recommendations - Autenticación con device_token (JWT) - Sistema obtiene user_id asociado al dispositivo - Genera recomendaciones usando algoritmo estándar - Retorna top 3-5 recomendaciones - Respuesta optimizada para display limitado: título (max 50 chars), fecha, localidad, tipo - Incluye timestamp de última actualización - Solo dispositivos vinculados y autenticados acceden - Caché de 1 hora

---

Resumen de Historias de Usuario por Prioridad

Prioridad	Cantidad	IDs
Alta	16 historias	HU-001, HU-002, HU-004, HU-006, HU-007, HU-008, HU-009, HU-010, HU-011, HU-014, HU-015, HU-017, HU-020
Media	14 historias	HU-003, HU-005, HU-012, HU-013, HU-016, HU-018, HU-019, HU-021, HU-024, HU-025, HU-026, HU-030, HU-031, HU-032
Baja	5 historias	HU-022, HU-023, HU-027, HU-028, HU-029, HU-033, HU-034, HU-035

Resumen de Historias de Usuario por Módulo

Módulo	Historias	Prioridad Alta	Prioridad Media	Prioridad Baja
Autenticación	5	3	2	0
Actividades	8	6	2	0
Favoritos	3	2	1	0
Recomendaciones	2	1	1	0
Perfil Usuario	2	1	1	0
Administración	9	0	4	5
ETL	4	0	3	1
IoT	2	0	0	2

---

3. Requisitos Funcionales

3.1 Módulo de Autenticación y Gestión de Usuarios

RF-001: Registro de Usuario **Prioridad:** Alta
Descripción: El sistema debe permitir el registro de nuevos usuarios mediante email y contraseña.

Entradas: - Email válido - Contraseña (mínimo 8 caracteres, al menos 1 mayúscula, 1 número, 1 carácter especial) - Nombre completo - Aceptación de términos y condiciones

Proceso: 1. Validar formato de email 2. Verificar que el email no esté registrado 3. Hash de contraseña (bcrypt o Argon2) 4. Crear usuario con rol "usuario" 5. Generar tokens JWT (access + refresh)

Salidas: - Usuario creado en base de datos - Access token (válido 30 min) - Refresh token (válido 30 días) - Mensaje de confirmación

Criterios de aceptación: - El email debe ser único en el sistema - La contraseña debe cumplir política de seguridad - Los tokens generados deben ser válidos

---

RF-002: Login de Usuario Prioridad: Alta
Descripción: El sistema debe permitir la autenticación mediante OAuth2 con flujo password grant.

Entradas: - Email - Contraseña

Proceso: 1. Buscar usuario por email 2. Verificar hash de contraseña 3. Generar access token (30 minutos) 4. Generar refresh token (30 días) 5. Registrar último acceso

Salidas: - Access token (JWT) - Refresh token (JWT) - Información básica del usuario (id, nombre, rol)

Criterios de aceptación: - Credenciales válidas permiten acceso - Credenciales inválidas retornan error 401 - Tokens incluyen claims: user_id, rol, exp

---

RF-003: Refresh Token Prioridad: Alta
Descripción: El sistema debe permitir renovar el access token usando un refresh token válido.

Entradas: - Refresh token válido

Proceso: 1. Validar refresh token 2. Verificar que no esté revocado 3. Generar nuevo access token 4. Opcionalmente rotar refresh token

Salidas: - Nuevo access token - Nuevo refresh token (si rotación habilitada)

Criterios de aceptación: - Solo tokens válidos y no revocados permiten refresh - Tokens expirados retornan error 401

RF-004: Logout de Usuario Prioridad: Media
Descripción: El sistema debe permitir cerrar sesión revocando los tokens.

Entradas: - Refresh token

Proceso: 1. Marcar refresh token como revocado en BD 2. Registrar evento de logout

Salidas: - Confirmación de logout

Criterios de aceptación: - El refresh token queda invalidado - No se puede renovar access token con ese refresh token

---

RF-005: Gestión de Perfil de Usuario Prioridad: Alta
Descripción: El usuario debe poder ver y actualizar su información de perfil.

Campos editables: - Nombre completo - Foto de perfil (URL) - Etiquetas de interés (array de strings) - Disponibilidad horaria preferida - Localidad de preferencia - Nivel de actividad física preferido

Proceso: 1. Autenticar usuario 2. Validar campos enviados 3. Actualizar registro en BD 4. Retornar perfil actualizado

Criterios de aceptación: - Solo el usuario autenticado puede editar su propio perfil - Las etiquetas deben ser validadas contra lista permitida

---

3.2 Módulo de Actividades

RF-006: Listar Actividades **Prioridad:** Alta
Descripción: El sistema debe permitir listar actividades con paginación y filtros.

Filtros disponibles: - `tipo`: cultura, deporte, recreación - `localidad`: Chapinero, Santa Fe, La Candelaria - `fecha_desde`: fecha ISO 8601 - `fecha_hasta`: fecha ISO 8601 - `precio_min`: número - `precio_max`: número - `es_gratis`: boolean - `nivel_actividad`: bajo, medio, alto - `etiquetas`: array de strings

Parámetros de paginación: - `page`: número de página (default: 1) - `page_size`: tamaño de página (default: 20, max: 100)

Proceso: 1. Validar filtros y parámetros 2. Construir query SQL con filtros aplicados 3. Ejecutar query con LIMIT/OFFSET para paginación 4. Calcular metadata de paginación 5. Retornar respuesta

Salidas:

```
{
  "data": [
    {
      "id": "uuid",
      "titulo": "string",
      "descripcion_corta": "string",
      "imagen_url": "string",
      "fecha_inicio": "datetime",
      "localidad": "string",
      "precio": "number",
      "es_gratis": "boolean",
      "etiquetas": ["string"],
      "popularidad": "number"
    }
  ],
  "pagination": {
    "total": 100,
    "page": 1,
    "page_size": 20,
    "total_pages": 5
  }
}
```

Criterios de aceptación: - Filtros se aplican correctamente en combinación - Paginación funciona correctamente - Tiempo de respuesta <2s para queries normales

---

RF-007: Ver Detalle de Actividad **Prioridad:** Alta
Descripción: El sistema debe mostrar información completa de una actividad.

Entradas: - ID de actividad (UUID)

Salidas: - Título - Descripción completa (markdown o HTML) - Tipo de actividad - Fecha y hora inicio/fin - Ubicación (dirección, coordenadas GPS) - Localidad - Precio - Es gratis (boolean) - Nivel de actividad física - Etiquetas - Información de contacto - Enlace externo - Fuente de datos - Popularidad (score) - Fecha de creación - Fecha de última actualización

Proceso: 1. Validar UUID 2. Query a BD 3. Registrar vista (para popularidad) 4. Retornar actividad completa

Criterios de aceptación: - Actividad existente retorna información completa - Actividad inexistente retorna error 404 - Vista se registra máximo 1 vez por usuario por día

---

RF-008: Búsqueda de Actividades **Prioridad:** Alta
Descripción: El sistema debe permitir búsqueda por palabras clave.

Entradas: - `q`: query de búsqueda (texto libre) - Filtros opcionales (localidad, tipo, fecha, etc.) - Parámetros de paginación

Proceso: 1. Sanitizar query de búsqueda 2. Aplicar búsqueda full-text en campos: - `titulo` (peso 3) - `descripcion` (peso 2) - `etiquetas` (peso 4) 3. Aplicar filtros adicionales 4. Ordenar por relevancia (score de búsqueda) 5. Paginar resultados

Salidas: - Lista de actividades coincidentes (mismo formato que RF-006) - Cada resultado incluye score de relevancia

Criterios de aceptación: - Búsqueda encuentra coincidencias parciales - Búsqueda es case-insensitive - Acentos son ignorados (búsqueda normalizada) - Tiempo de respuesta <2s

---

RF-009: CRUD de Actividades (Administrador) **Prioridad:** Alta
Descripción: Los administradores deben poder crear, editar y eliminar actividades.

Operaciones:

Crear Actividad:

```
POST /api/v1/actividades
Headers: Authorization: Bearer <admin_token>
Body: {
  "titulo": "string" (required),
  "descripcion": "string" (required),
  "tipo": "cultura|deporte|recreacion" (required),
  "fecha_inicio": "datetime" (required),
  "fecha_fin": "datetime" (optional),
  "ubicacion_direccion": "string" (required),
  "ubicacion_lat": "float" (required),
  "ubicacion_lng": "float" (required),
  "localidad": "Chapinero|Santa Fe|La Candelaria" (required),
  "precio": "float" (default: 0),
  "es_gratis": "boolean" (default: true),
  "nivel_actividad": "bajo|medio|alto" (optional),
  "etiquetas": ["string"] (required),
  "contacto": "string" (optional),
  "enlace_externo": "url" (optional),
  "fuente": "string" (default: "manual")
}
```

Actualizar Actividad:

```
PUT /api/v1/actividades/{id}
Headers: Authorization: Bearer <admin_token>
Body: { campos a actualizar }
```

Eliminar Actividad (Soft Delete):

```
DELETE /api/v1/actividades/{id}
Headers: Authorization: Bearer <admin_token>
```

Criterios de aceptación: - Solo usuarios con rol "administrador" pueden realizar operaciones - Fechas de actividades deben ser futuras al crear - Actividades eliminadas tienen estado "inactiva" pero no se borran de BD - Actividades inactivas no aparecen en búsquedas públicas

---

RF-010: Importación Manual de Actividades **Prioridad:** Media
Descripción: Los administradores deben poder importar actividades desde archivo CSV/JSON.

Entradas: - Archivo CSV o JSON - Formato definido en documentación

Formato CSV esperado:

```
titulo,descripcion,tipo,fecha_inicio,fecha_fin,ubicacion_direccion,ubicacion_lat,ubicacion_lng,localidad
```

Proceso: 1. Validar formato de archivo 2. Parsear contenido 3. Validar cada registro: - Campos obligatorios presentes - Formatos correctos - Detectar duplicados (titulo + fecha + ubicación) 4. Insertar actividades válidas con estado "pendiente_validacion" 5. Generar reporte detallado

Salidas:

```
{
  "resumen": {
    "total_procesados": 100,
    "exitosos": 85,
    "duplicados": 10,
    "errores": 5
  },
  "errores_detalle": [
    {
      "fila": 15,
      "error": "Campo 'fecha_inicio' inválido"
    }
  ]
}
```

Criterios de aceptación: - Solo administradores pueden importar - Registros inválidos se reportan sin detener proceso - Duplicados se detectan y reportan

---

3.3 Módulo de Favoritos

RF-011: Guardar Actividad como Favorito **Prioridad:** Alta
Descripción: Los usuarios deben poder marcar actividades como favoritas.

Entradas: - ID de actividad

Proceso: 1. Autenticar usuario 2. Verificar que actividad existe y está activa 3. Crear registro en tabla `favoritos` (user_id, actividad_id, fecha_guardado) 4. Incrementar campo `popularidad_favoritos` en actividad

Salidas: - Confirmación: `{"message": "Actividad guardada como favorita"}`

Criterios de aceptación: - Usuario autenticado puede guardar favoritos - No se permiten duplicados (constraint unique en BD) - Guardar favorito duplicado retorna 409 Conflict

---

RF-012: Listar Favoritos del Usuario **Prioridad:** Alta
Descripción: El usuario debe poder ver sus actividades favoritas.

Proceso: 1. Autenticar usuario 2. Query JOIN entre favoritos y actividades 3. Filtrar por user_id y actividades activas 4. Ordenar por fecha_guardado DESC 5. Paginar resultados

Salidas: - Lista de actividades favoritas (mismo formato que RF-006) - Incluye campo adicional: `fecha_guardado`

Criterios de aceptación: - Solo se muestran favoritos del usuario autenticado - Actividades eliminadas/inactivas no aparecen

---

RF-013: Eliminar Favorito **Prioridad:** Media
Descripción: El usuario debe poder quitar una actividad de favoritos.

Entradas: - ID de actividad

Proceso: 1. Autenticar usuario 2. Eliminar registro de tabla `favoritos` WHERE user_id = X AND actividad_id = Y 3. Decrementar campo `popularidad_favoritos` en actividad

Salidas: - Confirmación: `{"message": "Favorito eliminado"}`

Criterios de aceptación: - Solo el usuario propietario puede eliminar su favorito - Eliminar favorito inexistente retorna 404

3.4 Módulo de Recomendaciones (IA)

RF-014: Obtener Recomendaciones Personalizadas **Prioridad:** Alta
Descripción: El sistema debe generar recomendaciones basadas en etiquetas del usuario y popularidad.

Entradas: - ID de usuario (implícito por token JWT) - `limit`: cantidad de recomendaciones (default: 10, max: 50)

Algoritmo MVP:

```
def calcular_score_recomendacion(actividad, usuario):
    score = 0

    # Base: Popularidad (0-100)
    score += actividad.popularidad_normalizada * 100

    # Bonus por etiquetas coincidentes
    etiquetas_usuario = set(usuario.etiquetas_interes)
    etiquetas_actividad = set(actividad.etiquetas)
    coincidencias = etiquetas_usuario.intersection(etiquetas_actividad)
    score += len(coincidencias) * 10

    # Bonus por localidad preferida
    if actividad.localidad == usuario.localidad_preferida:
        score += 5

    # Bonus por disponibilidad horaria
    if hora_actividad in usuario.disponibilidad_horaria:
        score += 3

    # Penalización si ya es favorito
```

```
    if actividad in usuario.favoritos:
        score = 0  # Excluir

    return score
```

Proceso: 1. Obtener perfil completo del usuario 2. Query actividades activas con fecha >= hoy 3. Calcular score para cada actividad 4. Ordenar por score DESC 5. Filtrar actividades con score > 0 6. Tomar top N 7. Generar explicación para cada recomendación

Salidas:

```
{
  "recomendaciones": [
    {
      "actividad": { /* datos completos */ },
      "score": 125.5,
      "explicacion": "Recomendado porque te gusta 'arte' y 'música'"
    }
  ]
}
```

Criterios de aceptación: - Usuarios sin etiquetas reciben recomendaciones basadas solo en popularidad - Explicación debe ser comprensible y específica - Tiempo de generación <1s para 10 recomendaciones

---

RF-015: Actualizar Scoring de Popularidad **Prioridad:** Media
Descripción: El sistema debe actualizar automáticamente el score de popularidad.

Componentes del score: - `popularidad_favoritos`: contador de favoritos - `popularidad_vistas`: contador de vistas (limitado 1/usuario/día)

Score normalizado:

```
popularidad_normalizada = (
    (favoritos * 1.0) + (vistas * 0.1)
) / max_popularidad_sistema
```

Eventos que actualizan: - Usuario guarda favorito: `popularidad_favoritos += 1` - Usuario elimina favorito: `popularidad_favoritos -= 1` - Vista de detalle: `popularidad_vistas += 0.1` (máx 1/user/día)

Proceso: - Actualizar en tiempo real al ocurrir evento - Recalcular `popularidad_normalizada` diariamente (batch job)

Criterios de aceptación: - Score refleja interacción real - No se puede inflar artificialmente (límites por usuario)

3.5 Módulo de Gestión de Ingesta (ETL Manager)

RF-016: Dashboard de Estado de Ingesta **Prioridad:** Media
Descripción: Los administradores deben poder ver el estado de procesos ETL.

Información mostrada: - Última ejecución: fecha, hora, duración - Estado: éxito, error, corriendo - Actividades importadas: total - Actividades pendientes validación - Errores reportados: lista - Fuentes de datos: lista con estado - Próxima ejecución programada (si aplica)

Endpoint: `GET /api/v1/admin/etl/status`

Criterios de aceptación: - Solo administradores acceden - Información en tiempo real si hay proceso corriendo

---

RF-017: Activar Proceso de Ingesta Manual **Prioridad:** Media
Descripción: Los administradores deben poder ejecutar manualmente el script ETL.

Endpoint: `POST /api/v1/admin/etl/run`

Proceso: 1. Verificar permisos de admin 2. Verificar que no hay otro proceso ETL corriendo 3. Ejecutar script ETL Docker: `bash docker run --network=triqueta_network \ -e DATABASE_URL=$DB_URL \ triqueta-etl:latest` 4. Registrar inicio en tabla `etl_executions` 5. Retornar ID de ejecución

Salidas:

```
{
  "execution_id": "uuid",
  "status": "running",
  "started_at": "datetime"
}
```

Monitoreo: - Frontend puede consultar logs vía SSE o polling - Endpoint: `GET /api/v1/admin/etl/executions/{id}/logs`

Criterios de aceptación: - Solo administradores pueden activar - No se permiten ejecuciones concurrentes - Timeout de ejecución: 1 hora

---

RF-018: Validar Actividades Importadas **Prioridad:** Media
Descripción: Los administradores deben poder revisar y validar actividades importadas.

Endpoint: `GET /api/v1/admin/actividades?estado=pendiente_validacion`

Acciones disponibles: - **Aprobar:** `POST /api/v1/admin/actividades/{id}/aprobar` - Cambiar estado a "activa" - **Editar y Aprobar:** `PUT /api/v1/admin/actividades/{id}` + aprobar - Editar campos + cambiar estado a "activa" - **Rechazar:** `POST /api/v1/admin/actividades/{id}/rechazar` - Cambiar estado a "rechazada" - Opcionalmente agregar nota de rechazo

Criterios de aceptación: - Solo actividades con estado "activa" aparecen en búsquedas públicas - Administradores pueden ver actividades en cualquier estado - Actividades rechazadas se mantienen en BD para auditoría

---

3.6 Módulo de Dispositivos IoT

RF-019: Vincular Dispositivo IoT **Prioridad:** Baja (post-MVP)
Descripción: Los usuarios deben poder vincular un dispositivo IoT a su cuenta.

Entradas: - `device_code`: código único del dispositivo (ej: "TRIQ-ABC123")

Proceso: 1. Autenticar usuario 2. Validar que código existe en tabla `dispositivos` 3. Verificar que dispositivo no esté ya vinculado 4. Crear relación en tabla `usuarios_dispositivos` 5. Generar token JWT de dispositivo (válido 1 año)

Salidas:

```
{
  "device_id": "uuid",
  "device_token": "jwt_token",
  "vinculado_at": "datetime"
}
```

Criterios de aceptación: - Un dispositivo solo puede estar vinculado a un usuario a la vez - Usuario puede desvincular dispositivo - Token de dispositivo tiene claims: device_id, user_id, exp

---

RF-020: API de Sincronización para Dispositivo **Prioridad:** Baja (post-MVP)
Descripción: El dispositivo IoT debe poder obtener recomendaciones del usuario vinculado.

Endpoint: `GET /api/v1/iot/recommendations`

Headers: - `Authorization: Bearer <device_token>`

Proceso: 1. Validar token de dispositivo 2. Obtener user_id asociado 3. Generar recomendaciones (algoritmo RF-014) 4. Filtrar top 3-5 5. Retornar información simplificada para display

Salidas:

```
{
  "recommendations": [
    {
      "titulo": "string (max 50 chars)",
      "fecha": "datetime",
      "localidad": "string",
      "tipo": "string"
    }
  ],
  "last_updated": "datetime"
}
```

Criterios de aceptación: - Solo dispositivos vinculados y autenticados acceden - Respuesta optimizada para display limitado - Actualización cada 1 hora o bajo demanda

---

3.7 Módulo de Administración

RF-021: Dashboard de Administración **Prioridad:** Media
Descripción: Los administradores deben tener un dashboard con métricas clave.

Métricas mostradas:

Usuarios: - Total de usuarios registrados - Usuarios activos (últimos 7/30 días) - Nuevos usuarios (última semana)

Actividades: - Total de actividades activas - Actividades por localidad (gráfico de torta) - Actividades por tipo (gráfico de barras) - Actividades pendientes de validación

Interacción: - Top 10 actividades más populares - Top 10 etiquetas más usadas - Total de favoritos guardados

Sistema: - Última ejecución ETL - Búsquedas recientes (últimas 20)

Endpoint: `GET /api/v1/admin/dashboard`

Criterios de aceptación: - Solo administradores acceden - Datos actualizados cada 5 minutos (caché) - Gráficos simples y claros en frontend

---

RF-022: Gestión de Usuarios (Administrador) **Prioridad:** Baja
Descripción: Los administradores deben poder gestionar usuarios.

Operaciones:

Listar Usuarios:

```
GET /api/v1/admin/usuarios
Parámetros: page, page_size, rol, estado
```

Ver Detalle:

```
GET /api/v1/admin/usuarios/{id}
Respuesta: perfil completo + actividad reciente
```

Cambiar Rol:

```
PATCH /api/v1/admin/usuarios/{id}/rol
Body: {"rol": "administrador" | "usuario"}
```

Desactivar/Activar Cuenta:

```
PATCH /api/v1/admin/usuarios/{id}/estado
Body: {"activo": true | false}
```

Criterios de aceptación: - Solo administradores acceden - Admin no puede autodesactivarse - Admin no puede quitarse rol de administrador (requiere otro admin) - Se registra auditoría de cambios en tabla `audit_logs`

Resumen de Requisitos Funcionales por Prioridad

Alta Prioridad (MVP Core):

- RF-001 a RF-005: Autenticación
- RF-006 a RF-009: Actividades
- RF-011 a RF-014: Favoritos y Recomendaciones

Media Prioridad (MVP Extendido):

- RF-010: Importación manual
- RF-015 a RF-018: Gestión ETL
- RF-021: Dashboard admin

Baja Prioridad (Post-MVP):

- RF-019 a RF-020: Dispositivos IoT
- RF-022: Gestión avanzada de usuarios

---

4. Requisitos No Funcionales

Basado en ISO/IEC 25010:2023

4.1 Usabilidad

RNF-001: Accesibilidad **Estándar:** WCAG 2.1 Level AA

Requisitos específicos: - Contraste de colores mínimo 4.5:1 para texto normal - Contraste de colores mínimo 3:1 para texto grande (18pt+) - Navegación completa por teclado (tab, enter, esc, flechas) - Textos alternativos (alt) en todas las imágenes significativas - Labels asociados a todos los inputs de formularios

- Orden lógico de foco (tab order) - Indicadores visuales de foco - No uso de color como único medio de transmitir información - Subtítulos en videos (si aplica)

Herramientas de validación: - axe DevTools - WAVE - Lighthouse accessibility audit

Métrica de aceptación: - Score de accesibilidad Lighthouse >= 95 - 0 errores críticos en WAVE

RNF-002: Aprendizaje y Usabilidad **Descripción:** La interfaz debe ser intuitiva y fácil de aprender.

Requisitos: - Usuario nuevo debe poder buscar actividades en <5 minutos sin instrucciones - Formularios con validación inline y mensajes de error claros - Máximo 3 clics para acciones principales - Feedback visual inmediato en todas las acciones (loading, success, error) - Tooltips en funciones no evidentes - Mensajes de error en lenguaje claro (no códigos técnicos)

Métricas de aceptación: - System Usability Scale (SUS) score >70 - Tasa de abandono en registro <30% - Tasa de completitud en onboarding >80%

RNF-003: Diseño Responsive **Descripción:** La interfaz debe adaptarse a diferentes tamaños de pantalla.

Breakpoints: - Mobile: 375px - 767px - Tablet: 768px - 1023px - Desktop: 1024px - 1920px - Large Desktop: 1921px+

Requisitos: - Diseño mobile-first - Touch targets mínimo 44x44px en móvil - No scroll horizontal - Imágenes responsive con lazy loading - Tipografía escalable (rem/em)

Criterios de aceptación: - 100% funcionalidad en todos los breakpoints - Lighthouse Mobile score >90

4.2 Eficiencia de Desempeño

RNF-004: Tiempo de Respuesta API **Descripción:** La API debe responder rápidamente a las peticiones.

Objetivos: - P50 (mediana): <500ms - P90: <2s - P95: <3s - P99: <5s

Por tipo de endpoint: | Tipo | P90 Target | |——|————| | Autenticación | <1s | | Listados con paginación | <2s | | Búsqueda | <2s | | Detalle de actividad | <500ms | | Recomendaciones | <1s | | Operaciones CRUD | <1s |

Herramientas de medición: - Prometheus + Grafana - APM (Application Performance Monitoring) - Logs de latencia

Criterios de aceptación: - 90% de requests cumplen targets - No timeout errors bajo carga normal

---

RNF-005: Tiempo de Carga Frontend **Descripción:** La aplicación web debe cargar rápidamente.

Métricas Core Web Vitals: - **LCP (Largest Contentful Paint):** <2.5s - **FID (First Input Delay):** <100ms - **CLS (Cumulative Layout Shift):** <0.1

Métricas adicionales: - **FCP (First Contentful Paint):** <1.8s - **TTI (Time to Interactive):** <3.8s - **Speed Index:** <3.4s

Optimizaciones requeridas: - Code splitting por rutas - Lazy loading de componentes pesados - Lazy loading de imágenes - Compresión gzip/brotli - Minificación de JS/CSS - Caché de assets estáticos (1 año) - CDN para assets estáticos

Criterios de aceptación: - Lighthouse Performance score >90 (desktop) - Lighthouse Performance score >80 (mobile)

RNF-006: Capacidad y Escalabilidad **Descripción:** El sistema debe soportar la carga esperada y poder escalar.

Capacidad mínima MVP: - 200 usuarios concurrentes - 1,000 actividades en catálogo - 100 requests/segundo - 10,000 usuarios registrados

Escalabilidad: - Arquitectura stateless (horizontal scaling) - Base de datos con replicas de lectura - Caché distribuida (Redis) - Load balancer

Criterios de aceptación: - Tests de carga con 300 usuarios concurrentes sin degradación - CPU usage <70% bajo carga normal - Memory usage <80% bajo carga normal

---

RNF-007: Optimización de Consultas **Descripción:** Las consultas a base de datos deben estar optimizadas.

Requisitos: - Índices en campos de búsqueda frecuente: - `usuarios.email` (unique) - `actividades.localidad` - `actividades.tipo` - `actividades.fecha_inicio` - `actividades.estado` - Índice full-text en: - `actividades.titulo` - `actividades.descripcion` - `actividades.etiquetas` - Índice GiST para coordenadas GPS (PostGIS) - Queries complejas con EXPLAIN ANALYZE - Paginación obligatoria en listados

Criterios de aceptación: - No queries con tiempo de ejecución >500ms - Todas las queries usan índices (no seq scans en tablas grandes)

4.3 Compatibilidad

RNF-008: Compatibilidad de Navegadores **Descripción:** La aplicación debe funcionar en navegadores modernos.

Navegadores soportados: | Navegador | Versión Mínima | |———|————-| | Chrome | 100+ | | Firefox | 100+ | | Safari | 15+ | | Edge | 100+ |

No soportados: - Internet Explorer (cualquier versión) - Navegadores sin soporte de ES2020

Criterios de aceptación: - 100% funcionalidad en navegadores soportados - Mensaje de advertencia en navegadores no soportados

---

RNF-009: Compatibilidad de Dispositivos **Descripción:** La aplicación debe funcionar en diferentes dispositivos.

Dispositivos soportados: - Desktop: Windows, macOS, Linux - Móvil: iOS 14+, Android 10+ - Tablet: iPad, Android tablets

Resoluciones mínimas: - Desktop: 1366x768 - Tablet: 768x1024 - Móvil: 375x667

Criterios de aceptación: - Interfaz completamente funcional en todos los dispositivos - Touch gestures funcionales en dispositivos táctiles

---

RNF-010: API RESTful **Descripción:** La API debe seguir principios REST.

Requisitos: - Versionado en URL: `/api/v1/` - Métodos HTTP semánticos: - GET: lectura - POST: creación - PUT/PATCH: actualización - DELETE: eliminación - Códigos de estado HTTP correctos: - 200: OK - 201: Created - 204: No Content - 400: Bad Request - 401: Unauthorized - 403: Forbidden - 404: Not Found - 409: Conflict - 422: Unprocessable Entity - 500: Internal Server Error - Formato JSON para requests y responses - CORS configurado correctamente - Content-Type: application/json

Estructura de respuesta estándar:

```
{
  "data": {},
  "message": "string",
  "errors": []
}
```

Criterios de aceptación: - 100% endpoints siguen convenciones REST - Documentación OpenAPI/Swagger disponible

---

4.4 Fiabilidad

RNF-011: Disponibilidad **Descripción:** El sistema debe estar disponible la mayor parte del tiempo.

Objetivo: 99% uptime mensual

Cálculo: - 99% uptime = máximo 7.2 horas de downtime por mes - 99.5% uptime = máximo 3.6 horas de downtime por mes

Estrategias: - Health checks en todos los servicios - Auto-restart de contenedores caídos - Monitoreo 24/7 con alertas - Backups automáticos diarios - Plan de recuperación ante desastres

Downtime planificado: - Mantenimientos programados fuera de horas pico - Notificación con 48 horas de anticipación

Criterios de aceptación: - Uptime mensual >= 99% - RTO (Recovery Time Objective) <4 horas - RPO (Recovery Point Objective) <24 horas

---

RNF-012: Tolerancia a Fallos **Descripción:** El sistema debe manejar errores gracefully.

Requisitos: - Manejo de excepciones en todos los endpoints - Logs detallados de errores - Mensajes de error amigables al usuario - Fallback para servicios externos (ej: APIs públicas) - Circuit breaker para servicios dependientes - Retry logic con exponential backoff

Ejemplo:

```
try:
    result = external_api.fetch_data()
except ExternalAPIError:
    # Log error
    logger.error("External API failed", exc_info=True)
    # Return cached data or graceful degradation
    result = get_cached_data()
```

Criterios de aceptación: - 0% errores no manejados - Ningún error expone detalles técnicos al usuario final - Sistema sigue funcionando (degradado) si API externa falla

---

RNF-013: Recuperación y Backups **Descripción:** Los datos deben estar respaldados y recuperables.

Estrategia de backups: - **Base de datos:** - Backup completo diario (3:00 AM) - Backup incremental cada 6 horas - Retención: 30 días - Almacenamiento: S3/Cloud Storage - Encriptación en reposo - **Archivos cargados:** - Almacenamiento redundante (S3 con versionado) - Replicación cross-region

Plan de recuperación: 1. Identificar tipo de fallo 2. Notificar stakeholders 3. Restaurar desde backup más reciente 4. Validar integridad de datos 5. Retomar operaciones 6. Post-mortem

Criterios de aceptación: - Backups exitosos 100% del tiempo - Test de restauración mensual - RTO <4 horas, RPO <24 horas

---

4.5 Seguridad

RNF-014: Autenticación y Autorización **Descripción:** El sistema debe implementar autenticación y autorización seguras.

Requisitos de autenticación: - OAuth2 con Password Grant Type - JWT con algoritmo RS256 (asimétrico) - Access token: 30 minutos de vigencia - Refresh token: 30 días de vigencia - Refresh token rotation habilitada - Revocación de tokens - Rate limiting en login: 5 intentos / 15 min

Requisitos de contraseña: - Mínimo 8 caracteres - Al menos 1 mayúscula - Al menos 1 número - Al menos 1 carácter especial - Hash con bcrypt (cost factor 12+) o Argon2id

Estructura JWT:

```
{
  "sub": "user_id",
  "rol": "usuario|administrador",
  "exp": 1234567890,
  "iat": 1234567890
}
```

Autorización: - RBAC (Role-Based Access Control) - Roles: usuario, administrador - Decoradores/middlewares para proteger endpoints - Verificación de permisos en cada request

Criterios de aceptación: - OWASP ASVS v5.0 Nivel 2 cumplimiento en autenticación - Tokens no almacenados en localStorage (httpOnly cookies preferred)

RNF-015: Protección contra Ataques Comunes **Descripción:** El sistema debe estar protegido contra ataques OWASP Top 10.

Protecciones requeridas:

A01 - Broken Access Control: - Validación de autorización en cada endpoint - Usuarios solo acceden a sus propios recursos

A02 - Cryptographic Failures: - HTTPS obligatorio en producción - Contraseñas hasheadas (bcrypt/Argon2) - Secrets en variables de entorno - No datos sensibles en logs

A03 - Injection: - ORM (SQLAlchemy) con prepared statements - Validación de inputs con Pydantic - Sanitización de queries de búsqueda

A04 - Insecure Design: - Threat modeling STRIDE - Security by design

A05 - Security Misconfiguration: - Headers de seguridad: `X-Content-Type-Options: nosniff` `X-Frame-Options: DENY` `X-XSS-Protection: 1; mode=block` `Strict-Transport-Security: max-age=31536000` `Content-Security-Policy: default-src 'self'` - Deshabilitar debug en producción - No exponer stack traces

A06 - Vulnerable Components: - Dependencias actualizadas - Dependabot/Renovate habilitado - SAST (Bandit, Semgrep)

A07 - Authentication Failures: - Rate limiting - MFA ready (preparado para futuro)

A08 - Software and Data Integrity Failures: - Firma de JWT - Validación de integridad en uploads

A09 - Logging Failures: - Logs centralizados - Alertas en eventos críticos

A10 - SSRF: - Whitelist de URLs externas - Validación de inputs

Criterios de aceptación: - 0 vulnerabilidades críticas en SAST/DAST - Compliance con OWASP ASVS Nivel 2

RNF-016: Protección de Datos Personales **Descripción:** El sistema debe cumplir con Ley 1581 de 2012 (Colombia).

Requisitos: - Aviso de privacidad claro y accesible - Consentimiento explícito en registro - Derecho de acceso: usuario puede ver sus datos - Derecho de rectificación: usuario puede editar sus datos - Derecho de supresión: usuario puede solicitar borrado de cuenta - Datos mínimos necesarios (data minimization) - Encriptación de datos sensibles en BD (opcional para MVP) - Logs de acceso a datos personales

Datos personales recopilados: - Email (obligatorio) - Nombre completo (obligatorio) - Preferencias (opcional) - Ubicación preferida (opcional)

No se recopilan: - Documento de identidad - Información financiera (no hay pagos en MVP) - Datos biométricos

Criterios de aceptación: - Aviso de privacidad publicado - Funcionalidad de descarga de datos personales - Funcionalidad de eliminación de cuenta

---

RNF-017: Rate Limiting **Descripción:** El sistema debe limitar la tasa de requests para prevenir abuso.

Límites: | Endpoint | Límite | |———-|——| | POST /auth/register | 3 / hora / IP | | POST /auth/login | 5 / 15 min / IP | | POST /auth/refresh | 10 / hora / usuario | | GET /actividades/* | 100 / min / usuario | | POST /favoritos | 20 / min / usuario | | Endpoints admin | 200 / min / admin |

Implementación: - Redis para almacenar contadores - Sliding window algorithm - Headers de respuesta: `X-RateLimit-Limit: 100` `X-RateLimit-Remaining: 87` `X-RateLimit-Reset: 1640000000` - Status code 429 Too Many Requests cuando se excede

Criterios de aceptación: - Rate limiting activo en producción - Ataques de fuerza bruta mitigados

---

4.6 Mantenibilidad

RNF-018: Código Limpio y Estándares **Descripción:** El código debe ser legible, mantenible y seguir estándares.

Backend (Python): - PEP 8 (linting con ruff o flake8) - Type hints en todas las funciones - Docstrings en módulos, clases y funciones públicas - Complejidad ciclomática <10 por función - Cobertura de tests >80%

Frontend (TypeScript): - ESLint con reglas estrictas - Prettier para formateo - Componentes funcionales con hooks - Props con TypeScript interfaces - Cobertura de tests >70%

Estructura de proyecto:

```
backend/
  app/
      api/          # Routers
      core/         # Config, security
      models/       # SQLAlchemy models
      schemas/      # Pydantic schemas
      services/     # Business logic
      db/           # Database setup
      main.py
  tests/
  requirements.txt
  Dockerfile

frontend/
  src/
      components/   # React components
      pages/        # Page components
      services/     # API calls
      hooks/        # Custom hooks
      types/        # TypeScript types
      utils/        # Utilities
      main.tsx
  tests/
  package.json
  Dockerfile
```

Criterios de aceptación: - CI/CD ejecuta linters y falla si hay errores - Pull requests requieren revisión de código - Documentación actualizada

---

RNF-019: Pruebas Automatizadas **Descripción:** El sistema debe tener cobertura de pruebas automatizadas.

Tipos de pruebas:

Backend: - **Unit tests:** pytest - Cobertura >80% - Tests de servicios y helpers - **Integration tests:** - Tests de endpoints con TestClient - Base de datos de prueba (SQLite o PostgreSQL) - **E2E tests (opcional MVP):** - Playwright para flujos críticos

Frontend: - **Unit tests:** Vitest - Cobertura >70% - Tests de componentes con React Testing Library - **E2E tests (opcional MVP):** - Playwright para flujos de usuario

CI/CD: - Tests ejecutados en cada push - Tests bloqueantes para merge - Reporte de cobertura automático

Criterios de aceptación: - Cobertura backend >80% - Cobertura frontend >70% - 0 tests failing en main branch

---

RNF-020: Documentación **Descripción:** El sistema debe estar bien documentado.

Documentación requerida: - **README.md:** Setup, instalación, ejecución - **API Documentation:** OpenAPI/Swagger generado automáticamente - **docs/:** - SRS.md (este documento) - ARCHITECTURE.md (diagrama de componentes) - DEPLOYMENT.md (guía de despliegue) - CONTRIBUTING.md (guía para colaboradores) - **Docstrings:** En código Python - **Comentarios JSDoc:** En código TypeScript (funciones complejas)

Criterios de aceptación: - Swagger UI accesible en `/docs` - README con instrucciones claras - Nuevo desarrollador puede levantar proyecto en <30 min

---

4.7 Portabilidad

RNF-021: Contenedorización **Descripción:** El sistema debe estar completamente dockerizado.

Requisitos: - **Backend:** Dockerfile multi-stage - **Frontend:** Dockerfile multi-stage con Nginx - **ETL:** Dockerfile para script - **docker-compose.yml:** Para desarrollo local - Backend - Frontend - PostgreSQL - Redis - Adminer (opcional)

Ejemplo docker-compose.yml:

```yaml
version: '3.8'
services:
  backend:
    build: ./backend
    ports:
      - "8000:8000"
    environment:
      - DATABASE_URL=postgresql://user:pass@db:5432/triqueta
      - REDIS_URL=redis://redis:6379
    depends_on:
      - db
      - redis

  frontend:
    build: ./frontend
    ports:
      - "3000:80"
    depends_on:
      - backend

  db:
    image: postgres:15-alpine
    environment:
      - POSTGRES_DB=triqueta
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=pass
    volumes:
      - postgres_data:/var/lib/postgresql/data

  redis:
    image: redis:7-alpine

volumes:
  postgres_data:
```

Criterios de aceptación: - `docker-compose up` levanta todo el stack - Configuración mediante variables de entorno - Imágenes optimizadas (<500MB backend, <50MB frontend)

RNF-022: CI/CD **Descripción:** El sistema debe tener pipeline de CI/CD automatizado.

Pipeline stages: 1. **Lint:** Ejecutar linters (ruff, eslint) 2. **Test:** Ejecutar tests con cobertura 3. **Build:** Construir imágenes Docker 4. **Security Scan:** SAST con Bandit/Semgrep, dependency check 5. **Deploy (manual trigger):** Desplegar a staging/producción

Herramientas: - GitHub Actions (preferred) - GitLab CI - Jenkins

Criterios de aceptación: - Pipeline ejecutado en cada push - Merge bloqueado si pipeline falla - Deploy automatizado a staging en merge a develop - Deploy manual a producción desde main

---

4.8 Satisfacción del Usuario

RNF-023: Net Promoter Score (NPS) **Descripción:** El sistema debe generar satisfacción en usuarios.

Objetivo: NPS >70

Método de medición: - Encuesta post-MVP (después de 2 semanas de uso) - Pregunta: "¿Qué tan probable es que recomiendes Triqueta Digital? (0-10)" - Cálculo: % Promoters (9-10) - % Detractors (0-6)

Criterios de aceptación: - NPS >= 70 en encuesta piloto (n>=20)

RNF-024: Tasa de Adopción **Descripción:** El sistema debe lograr adopción de usuarios.

Métricas: - Usuarios registrados: >= 200 en primer mes post-lanzamiento - Usuarios activos mensuales (MAU): >= 100 - Tasa de retención D7: >= 40% - Tasa de retención D30: >= 20%

Criterios de aceptación: - Cumplir métricas en evaluación post-lanzamiento

---

Resumen de Requisitos No Funcionales por Categoría

Usabilidad

- RNF-001: Accesibilidad WCAG 2.1 AA
- RNF-002: Aprendizaje <5 min
- RNF-003: Responsive design

Eficiencia

- RNF-004: Tiempo respuesta API <2s (P90)
- RNF-005: Tiempo carga frontend (Core Web Vitals)
- RNF-006: Capacidad 200 usuarios concurrentes
- RNF-007: Queries optimizadas

Compatibilidad

- RNF-008: Navegadores Chrome/Firefox/Safari/Edge 100+
- RNF-009: Dispositivos desktop/móvil/tablet
- RNF-010: API RESTful estándar

Fiabilidad

- RNF-011: Disponibilidad 99%
- RNF-012: Tolerancia a fallos
- RNF-013: Backups diarios, RTO <4h

Seguridad

- RNF-014: OAuth2 + JWT
- RNF-015: Protección OWASP Top 10
- RNF-016: Cumplimiento Ley 1581/2012
- RNF-017: Rate limiting

Mantenibilidad

- RNF-018: Código limpio, linters
- RNF-019: Tests >80% backend, >70% frontend
- RNF-020: Documentación completa

Portabilidad

- RNF-021: Docker Compose
- RNF-022: CI/CD automatizado

Satisfacción

- RNF-023: NPS >70
- RNF-024: Adopción >= 200 usuarios

Matriz de Trazabilidad: Requisitos No Funcionales → ISO/IEC 25010:2023

ISO/IEC 25010	Característica	RNF
4.1	Functional Suitability	RF-001 a RF-022
4.2	Performance Efficiency	RNF-004, RNF-005, RNF-006, RNF-007
4.3	Compatibility	RNF-008, RNF-009, RNF-010
4.4	Usability	RNF-001, RNF-002, RNF-003
4.5	Reliability	RNF-011, RNF-012, RNF-013
4.6	Security	RNF-014, RNF-015, RNF-016, RNF-017
4.7	Maintainability	RNF-018, RNF-019, RNF-020
4.8	Portability	RNF-021, RNF-022
