**Contenido**

[**1. Introducción** 1](#_Toc208670362)

[**1.1 Propósito** 1](#_Toc208670363)

[**1.2 Alcance del sistema** 2](#_Toc208670364)

[**1.3 Definiciones, acrónimos y abreviaturas** 2](#_Toc208670365)

[**1.4 Visión general del documento** 2](#_Toc208670366)

[**2. Descripción general** 3](#_Toc208670367)

[**2.1 Perspectiva del producto** 3](#_Toc208670368)

[**2.2 Funcionalidad del producto (visión general de módulos)** 3](#_Toc208670369)

[**2.3 Características de los usuarios** 4](#_Toc208670370)

[**2.4 Restricciones** 4](#_Toc208670371)

[**2.5 Suposiciones y dependencias** 4](#_Toc208670372)

[**3. Requisitos específicos** 5](#_Toc208670373)

[**3.1 Requisitos funcionales** 5](#_Toc208670374)

[**3.2 Requisitos no funcionales** 6](#_Toc208670375)

[**3.3 Requisitos de interfaz** 6](#_Toc208670376)

[**4. Modelos del sistema** 7](#_Toc208670377)

[**4.1 Diagramas de casos de uso** 7](#_Toc208670378)

[**4.2 Diagramas de secuencia (interacciones clave)** 7](#_Toc208670379)

[**4.3 Diagrama de clases (vista lógica)** 8](#_Toc208670380)

[**4.4 Modelo de datos (ER y relacional)** 8](#_Toc208670381)

[**5. Apéndices** 9](#_Toc208670382)

[**5.1 Glosario** 9](#_Toc208670383)

[**5.2 Futuras mejoras** 9](#_Toc208670384)

# 

# **1. Introducción**

## **1.1 Propósito**

El propósito de este documento es especificar de manera detallada los requisitos funcionales y no funcionales del sistema de intercambio de habilidades, con el fin de garantizar que el desarrollo cumpla con las necesidades de los usuarios y de los interesados en el proyecto.  
Este documento servirá como referencia para:

* El equipo de desarrollo (para implementar las funcionalidades).
* Los diseñadores de interfaces (para alinear las pantallas a los requisitos).
* Los interesados/propietarios del sistema (para validar que se satisfacen los objetivos de negocio).

## **1.2 Alcance del sistema**

El sistema será una **plataforma digital de intercambio de habilidades**. Los usuarios podrán registrarse, crear un perfil con las habilidades que ofrecen y las que buscan, y conectarse con otros usuarios compatibles.  
El intercambio se regulará a través de un sistema de **créditos**:

* Los usuarios ganan créditos al completar exitosamente un intercambio de sus habilidades.
* Los usan para solicitar servicios o conocimientos de otros.

El sistema incluirá:

* Registro, autenticación y gestión de sus perfiles.
* Gestión de habilidades.
* Sistema de créditos.
* Algoritmo de emparejamiento (matches).
* Módulo de mensajería/contacto entre usuarios.
* Panel de administración básico para que un rol (Administrador) pueda supervisar la plataforma, gestionar cuentas problemáticas y atender reportes.

## **1.3 Definiciones, acrónimos y abreviaturas**

* **SRS**: Software Requirements Specification (Especificación de Requisitos de Software).
* **API**: Application Programming Interface.
* **Match**: Emparejamiento automático entre usuarios basado en intereses/habilidades.
* **Crédito**: Unidad de valor digital utilizada dentro del sistema para balancear el intercambio de servicios.
* **peer-to-peer:**Modelo donde los participantes comparten recursos directamente entre sí sin un intermediario centralizado

## **1.4 Visión general del documento**

El presente documento describe primero el contexto general del seguido de los requisitos funcionales y no funcionales. Posteriormente se incluyen modelos gráficos del sistema y un conjunto de apéndices con glosario.

# **2. Descripción general**

## **2.1 Perspectiva del producto**

El sistema será una plataforma web accesible desde navegadores en computadoras. Funcionará como intermediario entre personas que **ofrecen habilidades** y aquellas que **buscan adquirirlas**.  
El producto se basa en un modelo **peer-to-peer**, donde no hay transacciones monetarias directas, sino un sistema de **créditos internos** que regulan el intercambio.  
Se compondrá de:

* **Frontend**: interfaz de usuario.
* **Backend**: lógica del sistema, gestión de datos, API REST.
* **Base de datos**: persistencia de usuarios, habilidades, créditos y matches.
* **Panel de administración básico**: supervisión y control mínimo de cuentas.

## **2.2 Funcionalidad del producto (visión general de módulos)**

El sistema ofrecerá las siguientes funcionalidades principales:

1. **Registro y autenticación de usuarios** (crear cuenta, login, recuperación de contraseña).
2. **Gestión de perfiles** (editar información personal, agregar habilidades ofrecidas y buscadas).
3. **Sistema de créditos** (ganar créditos al ofrecer habilidades, consumirlos al solicitar).
4. **Búsqueda y filtrado de usuarios** (por habilidad, ubicación, nivel de experiencia, etc.).
5. **Algoritmo de matches** (emparejamiento automático según compatibilidad de oferta y demanda).
6. **Mensajería o contacto** (para que los usuarios acuerden intercambios).
7. **Panel de administración básico** (gestión de cuentas problemáticas, moderación de reportes).

## **2.3 Características de los usuarios**

Se identifican los siguientes tipos de usuarios:

* **Usuario registrado**: persona que crea un perfil, ofrece habilidades y busca otras.
* **Usuario invitado**: visitante que navega parcialmente la plataforma, pero debe registrarse para participar.
* **Administrador**: rol especial encargado de moderar usuarios, gestionar reportes y mantener la integridad de la comunidad.

## **2.4 Restricciones**

* **Tecnológicas**:
  + Desarrollo backend con API REST (Node.js).
  + Frontend web responsivo (React).
  + Base de datos relacional (MySQL).
* **De seguridad**:
  + Autenticación segura.
  + Cifrado de contraseñas (bcrypt ).
* **De tiempo**:
  + Las primeras versiones deben priorizar las funcionalidades núcleo (usuarios, habilidades, créditos, matches).

## **2.5 Suposiciones y dependencias**

* Se asume que los usuarios cuentan con conexión estable a Internet.
* El sistema depende de un proveedor de hosting en la nube
* El sistema puede integrarse en el futuro con servicios externos (ej. notificaciones por correo, pagos si se extiende a modelo híbrido).
* El éxito del sistema depende de la **adopción por parte de una comunidad activa** de usuarios.

# **3. Requisitos específicos**

## **3.1 Requisitos funcionales**

**RF-01 Registro y autenticación de usuarios**

* El sistema debe permitir el registro de nuevos usuarios mediante correo electrónico y contraseña.
* El sistema debe permitir autenticación con credenciales válidas.
* El sistema debe ofrecer recuperación de contraseña vía correo electrónico.

**RF-02 Gestión de perfil de usuario**

* El usuario podrá editar sus datos personales (nombre, foto, descripción, etc.).
* El usuario podrá agregar, modificar o eliminar habilidades ofrecidas.
* El usuario podrá agregar, modificar o eliminar habilidades buscadas.

**RF-03 Sistema de créditos**

* El usuario gana créditos al ofrecer una habilidad o completar un intercambio.
* El usuario consume créditos únicamente cuando un intercambio se confirma como realizado.
* El sistema debe registrar todas las transacciones de créditos.

**RF-04 Búsqueda y filtrado de usuarios**

* El sistema permitirá buscar usuarios por nombre o habilidad.
* El sistema permitirá filtrar resultados por ubicación, nivel de experiencia u otras categorías.

**RF-05 Algoritmo de emparejamiento (matches)**

* El sistema debe sugerir posibles conexiones entre usuarios basadas en compatibilidad de habilidades ofrecidas/buscadas.
* El sistema debe actualizar los matches dinámicamente según cambios en el perfil de los usuarios.

**RF-06 Mensajería / contacto entre usuarios**

* El sistema permitirá a los usuarios iniciar conversaciones con matches confirmados.
* El sistema debe notificar a los usuarios cuando reciban un nuevo mensaje.

**RF-07 Panel de administración básico**

* El administrador podrá visualizar una lista de usuarios registrados.
* El administrador podrá bloquear o suspender cuentas problemáticas.
* El administrador podrá revisar y gestionar reportes enviados por usuarios.

**RF-08 API REST**

* El backend debe exponer endpoints para todas las funcionalidades principales (usuarios, habilidades, créditos, matches, mensajes).
* La API debe estar documentada con Swagger u OpenAPI.

## **3.2 Requisitos no funcionales**

* **Rendimiento:** el sistema debe responder en menos de 2 segundos para el 90% de las solicitudes.
* **Escalabilidad:** el sistema debe soportar al menos 1.000 usuarios concurrentes en la primera versión escalable.
* **Seguridad:**
  + Las contraseñas deben almacenarse en formato cifrado.
  + El sistema debe usar tokens JWT para la autenticación de sesiones.
* **Usabilidad:**
  + La interfaz debe ser responsiva y accesible desde móviles y computadores.
  + La navegación debe ser clara y consistente.
* **Mantenibilidad:**
  + El código debe estar documentado.
  + La API debe seguir convenciones RESTful.
* **Disponibilidad:** el sistema debe estar disponible un 99% del tiempo mensual.

## **3.3 Requisitos de interfaz**

* **Interfaz de usuario (UI/UX):**
  + Pantalla de registro e inicio de sesión.
  + Pantalla de perfil con habilidades ofrecidas y buscadas.
  + Pantalla de resultados de búsqueda y matches.
  + Pantalla de chat/mensajería.
  + Panel básico de administración (tabla de usuarios, acciones de bloqueo).
* **Interfaz con software externo:**
  + Enviar correos electrónicos para recuperación de contraseña.
  + Posible integración futura con notificaciones push.
* **Interfaz con la base de datos:**
  + El backend debe interactuar con una base de datos relacional para almacenar usuarios, habilidades, créditos, matches y mensajes.

# **4. Modelos del sistema**

## **4.1 Diagramas de casos de uso**

Principales actores:

* **Usuario registrado**
* **Administrador**
* **Sistema** (interno, como motor de matches y créditos)

Casos de uso principales del **Usuario registrado**:

1. Registrarse e iniciar sesión.
2. Gestionar perfil (datos, habilidades ofrecidas, habilidades buscadas).
3. Buscar y filtrar usuarios.
4. Visualizar matches sugeridos.
5. Contactar a otro usuario (mensajería).
6. Participar en intercambio de habilidades (ganar o gastar créditos).

Casos de uso principales del **Administrador**:

1. Acceder al panel de administración.
2. Consultar lista de usuarios.
3. Bloquear o suspender usuarios.
4. Revisar reportes enviados por usuarios.

## **4.2 Diagramas de secuencia (interacciones clave)**

Ejemplos de flujos:

* **Flujo de registro e inicio de sesión**
  + Usuario → Sistema: Registro con datos
  + Sistema → BD: Guarda usuario
  + Usuario → Sistema: Login
  + Sistema → Usuario: Token de sesión
* **Flujo de match entre usuarios**
  + Usuario A → Sistema: Publica habilidades ofrecidas y buscadas
  + Usuario B → Sistema: Publica habilidades ofrecidas y buscadas
  + Sistema → Algoritmo: Evalúa compatibilidad
  + Algoritmo → Sistema: Retorna match
  + Sistema → Usuario A/B: Notificación de match
* **Flujo de intercambio confirmado**
  + Usuario A → Sistema: Solicita habilidad de Usuario B
  + Usuario B → Sistema: Acepta solicitud
  + Ambos realizan la sesión fuera o dentro de la plataforma
  + Usuario A/B → Sistema: Confirman intercambio
  + Sistema → Créditos: Descuenta de A, acredita a B

## **4.3 Diagrama de clases (vista lógica)**

Clases principales (a nivel conceptual):

* **Usuario** (id, nombre, email, contraseña cifrada, rol, créditos, estado)
* **Habilidad** (id, nombre, descripción, nivel)
* **PerfilHabilidad** (relación Usuario ↔ Habilidad, como ofrecida o buscada)
* **Match** (id, usuarioA, usuarioB, estado)
* **Mensaje** (id, remitente, receptor, contenido, fecha)
* **Reporte** (id, usuarioReportado, motivo, estado)

## **4.4 Modelo de datos (ER y relacional)**

Entidades principales y relaciones:

* **Usuario** (1) ↔ (N) **PerfilHabilidad**
* **Habilidad** (1) ↔ (N) **PerfilHabilidad**
* **Usuario** (1) ↔ (N) **Mensaje** (enviado/recibido)
* **Usuario** (N) ↔ (N) **Match** (mediante tabla intermedia)
* **Usuario** (1) ↔ (N) **Reporte**

# **5. Apéndices**

## **5.1 Glosario**

* **Créditos**: moneda interna usada para balancear intercambios.
* **Match**: emparejamiento automático entre usuarios compatible en habilidades ofrecidas y buscadas.
* **Perfil de usuario**: información personal y profesional registrada por un usuario.
* **Intercambio**: actividad confirmada entre dos usuarios donde se transfieren habilidades y créditos.

## **5.2 Futuras mejoras**

* **Panel de administración avanzado**: estadísticas, gestión de créditos, configuración global del sistema.
* **Sistema de reputación o calificaciones**: usuarios podrán dejar reseñas tras un intercambio.
* **Integración con notificaciones push o correo**: para avisos más dinámicos.
* **Aplicación móvil nativa**: iOS/Android.
* **Pagos externos** (si se evoluciona a modelo híbrido con dinero real).