Universidad Da Vinci De Guatemala

Facultad de Ingeniería

Ingeniería en Sistemas

“Evaluación final”

Kimberly Dayana Vasquez Mayén

202404641

Programación Web

Guatemala 05 de diciembre de 2025

Contenido

1. Introducción

El presente documento técnico tiene como propósito exponer la propuesta de desarrollo del proyecto **SkillTrade**, una plataforma web diseñada para facilitar el intercambio de habilidades entre usuarios sin la necesidad de utilizar dinero como medio de transacción. Este proyecto forma parte de la evaluación final del curso de Desarrollo Web y busca demostrar la capacidad para conceptualizar, estructurar y planificar una solución tecnológica completa, tal como lo haría un ingeniero de software en un entorno profesional.

A lo largo del documento se presentan la identificación del problema y la solución planteada, la justificación técnica del stack seleccionado, el diseño del modelo de datos, la definición de la API principal y la estimación de horas y costos de desarrollo. Cada sección responde a los requerimientos establecidos por el curso y sigue lineamientos formales para garantizar claridad, coherencia y un enfoque orientado a la ingeniería.

SkillTrade se concibe como un prototipo funcional (v0) que aborda una necesidad real dentro del contexto educativo y social: permitir que personas con habilidades útiles puedan intercambiarlas de manera organizada, segura y accesible. Este documento constituye la base técnica sobre la cual se desarrolla dicha solución.

2. SkillTrade

En esta sección se describe el problema que aborda SkillTrade, la solución propuesta y la arquitectura técnica seleccionada para su implementación, con un enfoque profesional y orientado a un proyecto web.

2.1. Definición del problema y solución

2.1.1. El problema

En la actualidad, muchas personas poseen habilidades valiosas como tocar un instrumento, programar, diseñar, hablar otro idioma, brindar tutorías académicas o realizar trabajos creativos pero no siempre cuentan con los recursos económicos para pagar cursos formales o servicios profesionales que les permitan aprender otras habilidades que necesitan.

Al mismo tiempo, existe un amplio grupo de personas que desea aprender algo nuevo o mejorar en áreas específicas (por ejemplo, reforzar matemáticas, practicar conversación en otro idioma, aprender herramientas digitales, etc.), pero se encuentra con varias dificultades:

- Desconoce a quién acudir o cómo encontrar a alguien confiable.
- Las plataformas de aprendizaje existentes suelen centrarse en cursos estructurados y de pago.
- Los intercambios informales de ayuda (“yo te enseño esto y tú me ayudas con aquello”) se organizan de forma desordenada, por mensajes sueltos en redes sociales o chats personales, sin registro de acuerdos ni mecanismos de reputación.

Como consecuencia, se pierden muchas oportunidades de aprendizaje colaborativo. Personas que podrían enseñar algo útil no logran conectar con quienes quieren aprenderlo, y quienes buscan apoyo en un tema concreto no disponen de una plataforma clara, estructurada y accesible para encontrar a alguien dispuesto a ayudar a cambio de otra habilidad.

El problema central identificado es la **ausencia de un espacio digital organizado, accesible y confiable para facilitar el intercambio de habilidades sin necesidad de dinero**, donde se gestione de forma clara quién ofrece, quién solicita, qué se intercambia y cómo se coordinan esos acuerdos.

2.1.2. La solución: funcionalidad principal de la aplicación web

Como respuesta a este problema se propone **SkillTrade**, una plataforma web orientada al **trueque de habilidades** entre usuarios, utilizando el conocimiento como principal moneda de intercambio. En lugar de pagar por un curso o servicio, cada persona puede **ofrecer lo que sabe** y, a cambio, **recibir ayuda o clases** en otra área donde desea aprender o mejorar.

La funcionalidad principal de SkillTrade se puede resumir en los siguientes puntos:

- **Perfiles de usuario:** Cada usuario dispone de un perfil con su información básica (nombre, breve descripción, ubicación general) y un resumen de las habilidades que ofrece y aquellas que desea aprender.
- **Gestión de habilidades ofrecidas y buscadas:**
    - Registro de habilidades ofrecidas (por ejemplo: “Clases básicas de guitarra”, “Asesoría en matemáticas”, “Diseño de logotipos”).
    - Publicación de habilidades o temas que el usuario está buscando aprender o reforzar.
- **Búsqueda y descubrimiento de habilidades:**
    - Filtros por categoría (Idiomas, Tecnología, Música, Arte y Diseño, Asesoría Académica, Servicios Creativos, etc.).
    - Búsqueda por palabras clave (por ejemplo: “guitarra”, “inglés”, “programación web”).
- **Solicitudes de intercambio:** Un usuario puede enviar una solicitud a otro, indicando:

- La habilidad que desea recibir.
- La habilidad que ofrece a cambio (cuando aplique).
- Un mensaje inicial con la explicación de su interés y posibles horarios.

- **Gestión del estado de la solicitud:** Cada solicitud de intercambio tiene estados definidos (pendiente, aceptada, rechazada, completada), lo que permite llevar un **historial básico de los intercambios realizados**.
- **Mensajería interna por solicitud:** Los participantes pueden comunicarse mediante mensajes asociados a cada solicitud, para coordinar modalidad (en línea o presencial), horarios, materiales y otros detalles.
- **Reseñas y calificaciones:** Una vez completado el intercambio, los usuarios pueden evaluarse mutuamente mediante calificaciones (por ejemplo, de 1 a 5 estrellas) y comentarios breves sobre la experiencia, construyendo un sistema de reputación dentro de la plataforma.

2.1.3. Justificación: innovación y aporte de valor

SkillTrade se considera una propuesta innovadora y necesaria por las siguientes razones:

1. **Cambio de modelo: del pago al trueque de conocimiento**

   Mientras que la mayoría de plataformas educativas y de servicios profesionales se basa en pagos monetarios, SkillTrade utiliza las habilidades como moneda de intercambio. Esto:

   - Reduce la barrera económica para acceder al aprendizaje.
   - Permite que estudiantes y personas con recursos limitados reciban apoyo a cambio de lo que ellos mismos pueden enseñar.

2. **Estructuración de intercambios que actualmente son informales**

Los intercambios de favores y habilidades ya existen, pero se dan de forma improvisada, sin estructura ni seguimiento. SkillTrade aporta:

- Perfiles claros y organizados.
- Registro de habilidades ofrecidas y solicitadas.
- Historial de solicitudes e intercambios.
- Sistema de reseñas y calificaciones que genera confianza.

3. **Enfoque en comunidad y aprendizaje colaborativo**

   La plataforma no se limita a listar anuncios, sino que crea un entorno donde cada usuario puede ser **docente y estudiante al mismo tiempo**, fomentando:

   - Aprendizaje horizontal entre pares.
   - Desarrollo de habilidades blandas como la comunicación y la enseñanza.
   - Redes de apoyo entre personas con intereses comunes.

4. **Alineación con la realidad de estudiantes y autodidactas**

   Para muchas personas, los cursos formales resultan costosos. SkillTrade ofrece una alternativa donde:

   - Un usuario puede dar asesorías en un tema que domina y, a cambio, recibir clases en otra área.
   - Se democratiza el acceso al conocimiento, especialmente en contextos con limitaciones económicas.

En conjunto, SkillTrade aporta valor al **facilitar el acceso al aprendizaje, estructurar el intercambio de habilidades y generar confianza entre los participantes** mediante mecanismos de reputación y organización.

2.2. Propuesta técnica (arquitectura)

La propuesta técnica de SkillTrade define las tecnologías a utilizar en el frontend, el backend y la capa de datos, justificando las decisiones tomadas a partir de comparaciones entre alternativas. Además, integra el uso de **v0** para la construcción del prototipo visual del proyecto.

2.2.1. Frontend

Para el desarrollo del frontend se consideran tres opciones ampliamente utilizadas: **React**, **Vue** y **Angular**.

React

- Biblioteca orientada a componentes.
- Utiliza Virtual DOM para actualizar únicamente las partes necesarias de la interfaz, mejorando la experiencia de uso.
- Amplia comunidad, ecosistema maduro y abundante documentación.
- Excelente integración con APIs REST y librerías de enrutamiento, manejo de estado y componentes de interfaz.

Vue

- Framework progresivo, también basado en componentes.
- Curva de aprendizaje amigable para desarrolladores que parten de HTML, CSS y JavaScript.
- Buen rendimiento y documentación clara.

- Comunidad en crecimiento, aunque con menor presencia en algunos entornos que React.

Angular

- Framework completo y estructurado, con muchas funcionalidades integradas (formularios, enrutamiento, inyección de dependencias, etc.).
- Uso intensivo de TypeScript y una curva de aprendizaje más pronunciada.
- Muy adecuado para proyectos grandes con equipos organizados, pero puede resultar pesado para un prototipo de tamaño medio.

Elección de frontend: React

Para SkillTrade se selecciona **React** debido a que:

- Permite construir una interfaz modular basada en componentes reutilizables (tarjetas de habilidades, formularios de registro, listados de solicitudes, etc.).
- Se integra de manera natural con una API REST construida en el backend.
- Cuenta con una comunidad muy amplia y recursos abundantes, lo que facilita la solución de problemas y la evolución del proyecto.
- Es una tecnología ampliamente utilizada en la industria, por lo que su uso aporta valor formativo y profesional.

Uso de v0 en el frontend

Como apoyo al diseño de la interfaz, se empleará **v0** para generar el **prototipo visual (v0) de las pantallas principales** de SkillTrade. La estrategia es:

- Utilizar v0 para diseñar vistas como la página principal, el listado de habilidades, el perfil de usuario y la sección de solicitudes de intercambio.
- Tomar estos diseños como base para la implementación en React, adaptando componentes y estilos según las necesidades funcionales reales.

El prototipo generado con v0 formará parte de la **DEMO que se presentará en el video explicativo**, integrándose explícitamente en la arquitectura del proyecto.

2.2.2. Backend y publicación

Para el backend se evalúan las siguientes opciones:

- **Node.js con Express (JavaScript)**
- **Django (Python)**
- **Laravel (PHP)**

Node.js + Express

- Permite utilizar JavaScript tanto en el frontend como en el backend.
- Diseñado para aplicaciones asíncronas y escalables.
- Express proporciona una forma sencilla y flexible de construir APIs REST.
- Integración natural con bases de datos como MongoDB mediante librerías especializadas.

Django

- Framework robusto y maduro en Python.
- Ofrece una estructura clara y muchas utilidades integradas.

- Introduce un segundo lenguaje (Python), lo que incrementa ligeramente la complejidad del proyecto al coexistir con JavaScript en el frontend.

Laravel

- Framework completo en PHP, con herramientas para migraciones, autenticación y más.
- Requiere trabajar en PHP para la capa de servidor, mientras el frontend se mantiene en JavaScript.

Elección de backend: Node.js + Express

Para SkillTrade se selecciona **Node.js con Express** como backend por las siguientes razones:

- Unifica el lenguaje del proyecto (JavaScript) en frontend y backend.
- Facilita la construcción de una **API REST** clara y modular para gestionar usuarios, habilidades, solicitudes, reseñas y mensajes.
- Ofrece un ecosistema muy amplio de librerías para seguridad, validación y autenticación.
- Se integra de forma directa con MongoDB, la base de datos elegida para la persistencia de datos.

Publicación del proyecto

El plan de publicación contempla:

- **Frontend (React):** despliegue en una plataforma como **Vercel** (o alternativa similar), permitiendo integración continua a partir del repositorio en GitHub.

- **Backend (Node.js + Express):** despliegue en un servicio como **Render**, Railway u otro proveedor que permita hospedar aplicaciones Node.js en un entorno accesible.
- **Base de datos:** uso de **MongoDB Atlas** en su modalidad gratuita (Free Tier), accesible desde el backend.

Este esquema permite contar con una aplicación web desplegada de extremo a extremo, con costos muy bajos o nulos en un entorno de prototipo.

2.2.3. Persistencia de datos: SQL vs NoSQL

La capa de datos puede implementarse sobre dos enfoques principales:

- **Bases de datos relacionales (SQL)**, como MySQL o PostgreSQL.
- **Bases de datos NoSQL orientadas a documentos**, como MongoDB.

Enfoque SQL

Ventajas principales:

- Modelo relacional estructurado, adecuado para datos muy tabulares.
- Integridad referencial mediante llaves foráneas.
- Ideal para sistemas con reglas rígidas y relaciones muy estrictas.

Enfoque NoSQL (MongoDB)

Ventajas principales:

- Esquema flexible, que permite que los documentos evolucionen en su estructura sin requerir migraciones complejas.
- Integración sencilla con Node.js utilizando librerías como Mongoose.

- Buena capacidad de escalado horizontal.
- Adecuado para datos semi estructurados y modelos que pueden crecer o variar con el tiempo.

Naturaleza de los datos en SkillTrade

SkillTrade maneja:

- Información de **usuarios** (perfiles, descripciones, ubicación, fecha de registro).
- **Habilidades** con descripciones, categorías y niveles que podrían ampliarse con campos opcionales en el futuro.
- **Solicitudes de intercambio** que vinculan usuarios y habilidades, con estados y mensajes asociados.
- **Reseñas** y **mensajes**, que se relacionan con solicitudes y usuarios.

Dado que la estructura puede evolucionar y que la relación entre entidades se presta bien al modelo de documentos, el enfoque NoSQL resulta especialmente conveniente.

Elección de base de datos: MongoDB

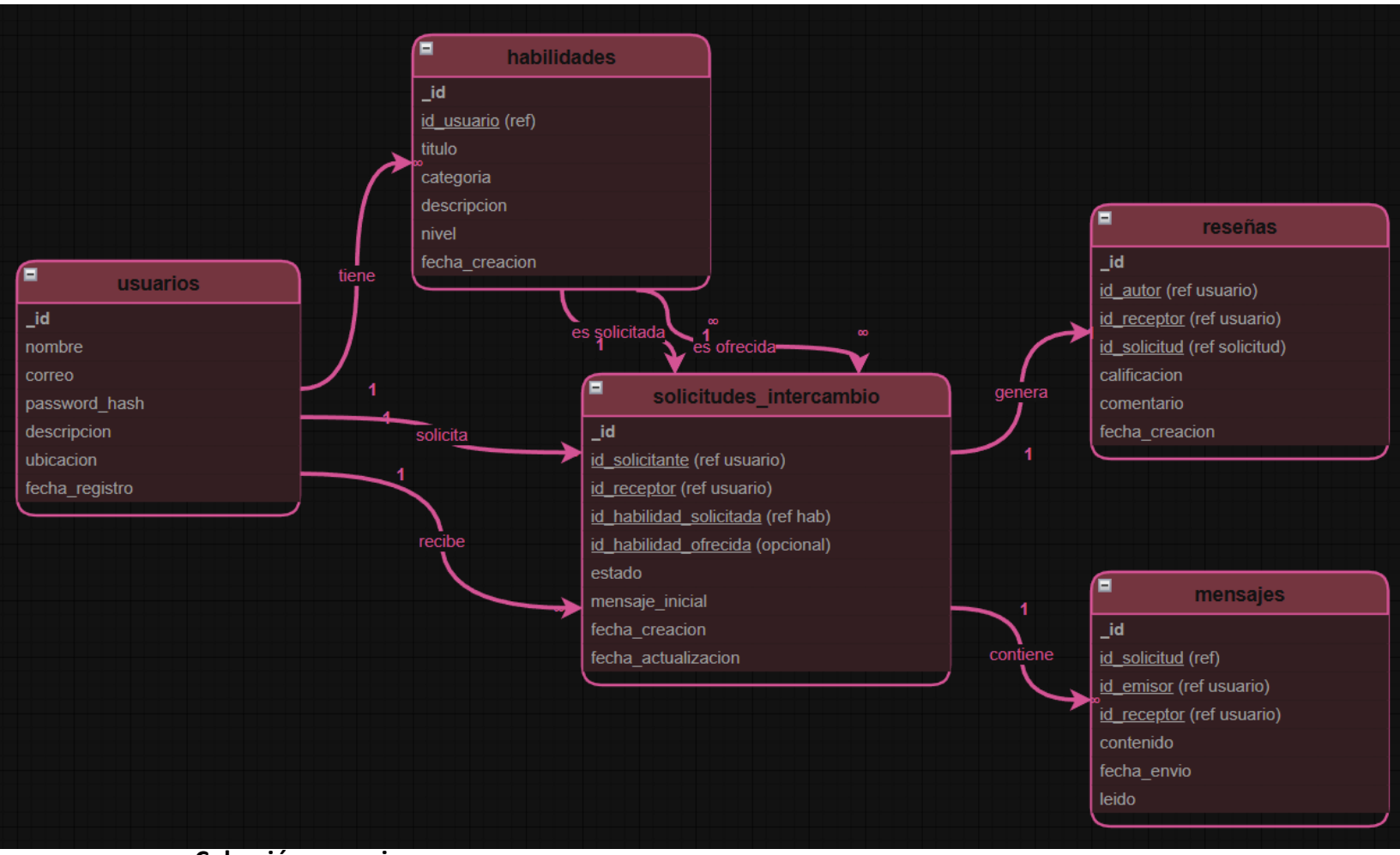
Se elige **MongoDB** como sistema de persistencia de datos porque:

- Su modelo orientado a documentos se adapta de forma natural a las entidades de SkillTrade.
- La flexibilidad del esquema facilita la introducción de nuevos campos y atributos a futuro.
- La integración con Node.js mediante Mongoose simplifica las operaciones CRUD y la validación de datos.

- MongoDB Atlas facilita el despliegue en la nube con planes gratuitos adecuados para un prototipo.

2.2.4. Esquema de base de datos

El modelo de datos de SkillTrade se organiza en varias colecciones principales dentro de MongoDB. A nivel conceptual, se propone el siguiente esquema:



Colección: usuarios

Almacena la información básica de cada persona registrada.

- _id (ObjectId)
- nombre (string)
- correo (string, único)

- password_hash (string)
- descripcion (string)
- ubicacion (string, opcional)
- fecha_registro (date)

Colección: habilidades

Registra las habilidades ofrecidas por los usuarios.

- _id (ObjectId)
- id_usuario (ObjectId) – referencia al usuario que ofrece la habilidad
- titulo (string)
- categoria (string)
- descripcion (string)
- nivel (string: básico, intermedio, avanzado)
- fecha_creacion (date)

Colección: solicitudes_intercambio

Representa los acuerdos formales de intercambio entre usuarios.

- _id (ObjectId)
- id_solicitante (ObjectId) – usuario que envía la solicitud
- id_receptor (ObjectId) – usuario que recibe la solicitud
- id_habilidad_solicitada (ObjectId) – habilidad que se desea recibir
- id_habilidad_ofrecida (ObjectId, opcional) – habilidad ofrecida a cambio

- estado (string: pendiente, aceptada, rechazada, completada)
- mensaje_inicial (string)
- fecha_creacion (date)
- fecha_actualizacion (date)

Colección: reseñas

Permite la evaluación de la experiencia de intercambio.

- _id (ObjectId)
- id_autor (ObjectId) – usuario que escribe la reseña
- id_receptor (ObjectId) – usuario evaluado
- id_solicitud (ObjectId) – referencia a la solicitud de intercambio
- calificacion (number, 1–5)
- comentario (string)
- fecha_creacion (date)

Colección: mensajes

Contiene los mensajes enviados dentro de una solicitud de intercambio.

- _id (ObjectId)
- id_solicitud (ObjectId) – referencia a solicitudes_intercambio
- id_emisor (ObjectId)
- id_receptor (ObjectId)
- contenido (string)

- fecha_envio (date)
- leido (bool)

Relaciones conceptuales principales

- Un **usuario** puede tener múltiples **habilidades** registradas.
- Un **usuario** puede enviar y recibir múltiples **solicitudes_intercambio**.
- Cada **solicitud_intercambio** puede tener varios **mensajes** asociados.
- Después de completar una solicitud, los usuarios pueden generar **reseñas** entre sí.

El **esquema de base de datos** se representa mediante un **diagrama de colecciones y relaciones** (similar a un ERD adaptado a NoSQL), que muestra gráficamente las entidades descritas y sus vínculos lógicos.

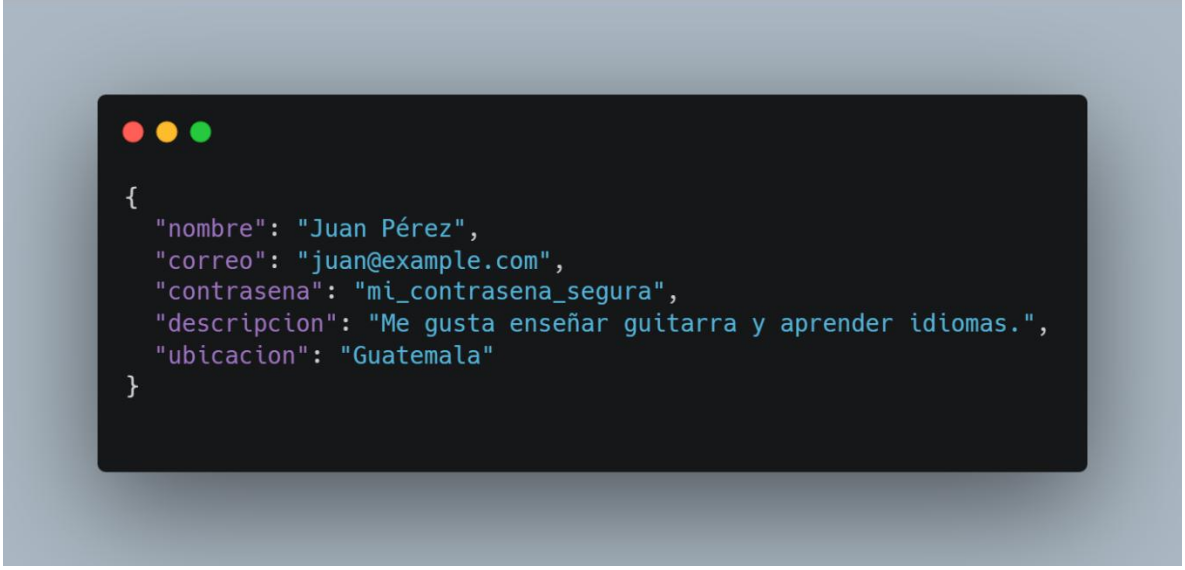
3. Diseño de Interfaz de Programación (API)

La comunicación entre el frontend (React) y el backend (Node.js + Express) se realizará mediante una API REST que intercambia datos en formato JSON sobre HTTPS.
A continuación se documentan **cuatro endpoints principales** que cubren el flujo básico de SkillTrade: registro de usuarios, publicación de habilidades, búsqueda de habilidades y creación de solicitudes de intercambio.

Método: POST
Ruta: /api/usuarios/registro

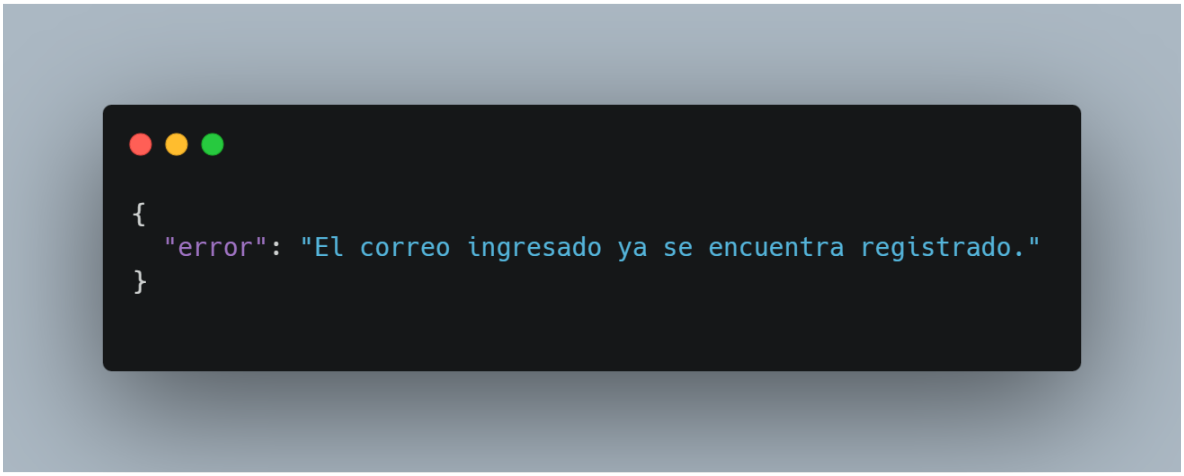
Descripción:
Registra un nuevo usuario en la plataforma SkillTrade.

Cuerpo de la petición (JSON):



Respuesta 201 – Created (éxito):

Respuesta 400 – Bad Request (correo ya registrado):



3.2. Endpoint 2 – Publicación de una habilidad

Método: POST

Ruta: /api/habilidades

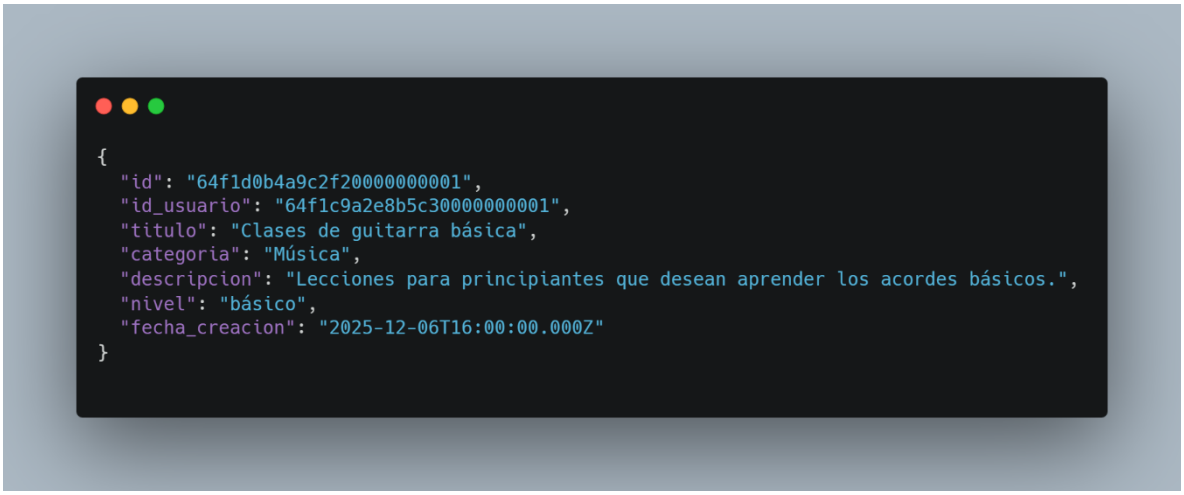
Descripción:

Registra una nueva habilidad ofrecida por un usuario autenticado.

Cuerpo de la petición (JSON):

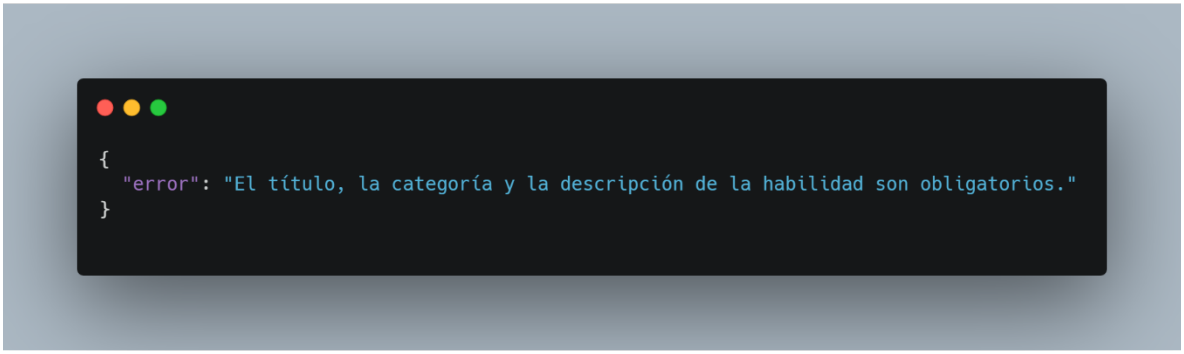
```
{
  "id_usuario": "64f1c9a2e8b5c30000000001",
  "titulo": "Clases de guitarra básica",
  "categoria": "Música",
  "descripcion": "Lecciones para principiantes que desean aprender los acordes básicos.",
  "nivel": "básico"
}
```

Respuesta 201 – Created (éxito):



```
{
  "id": "64f1d0b4a9c2f20000000001",
  "id_usuario": "64f1c9a2e8b5c30000000001",
  "titulo": "Clases de guitarra básica",
  "categoria": "Música",
  "descripcion": "Lecciones para principiantes que desean aprender los acordes básicos.",
  "nivel": "básico",
  "fecha_creacion": "2025-12-06T16:00:00.000Z"
}
```

Respuesta 400 – Bad Request (datos incompletos):



3.3. Endpoint 3 – Búsqueda de habilidades

Método: GET

Ruta: /api/habilidades

Descripción:

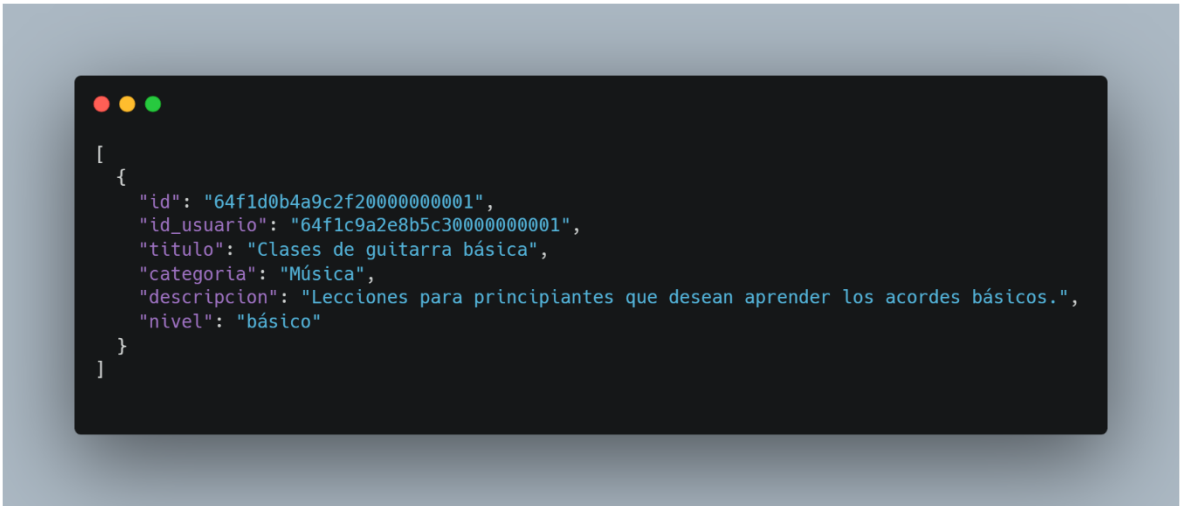
Devuelve una lista de habilidades registradas, permitiendo aplicar filtros opcionales por categoría y texto de búsqueda.

Parámetros de consulta (query params):

- categoria (opcional): filtra por categoría, por ejemplo: ?categoria=Musica
- q (opcional): texto a buscar en título o descripción, por ejemplo: ?q=guitarra

Ejemplo de petición: GET /api/habilidades?categoria=Musica&q=guitarra

Respuesta 200 – OK (éxito):



3.4. Endpoint 4 – Creación de solicitud de intercambio

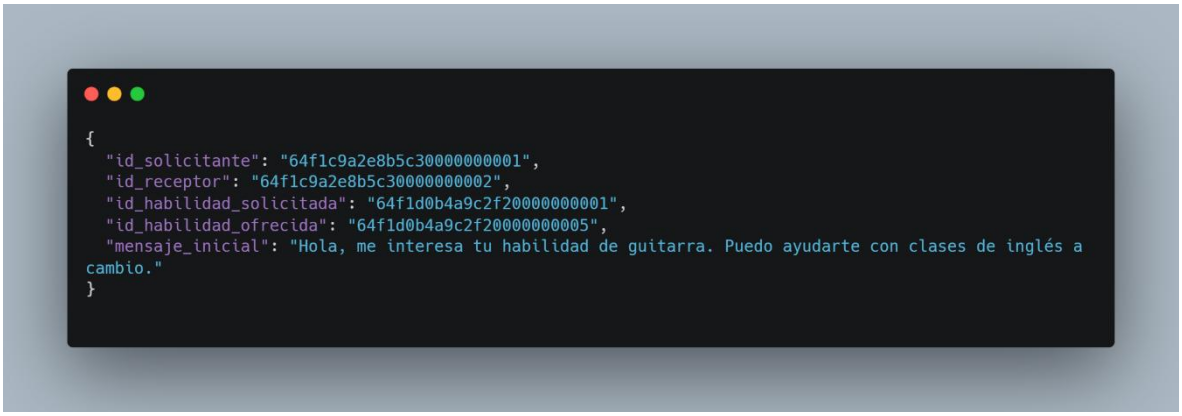
Método: POST

Ruta: /api/solicitudes

Descripción:

Crea una nueva solicitud de intercambio entre dos usuarios.

Cuerpo de la petición (JSON):

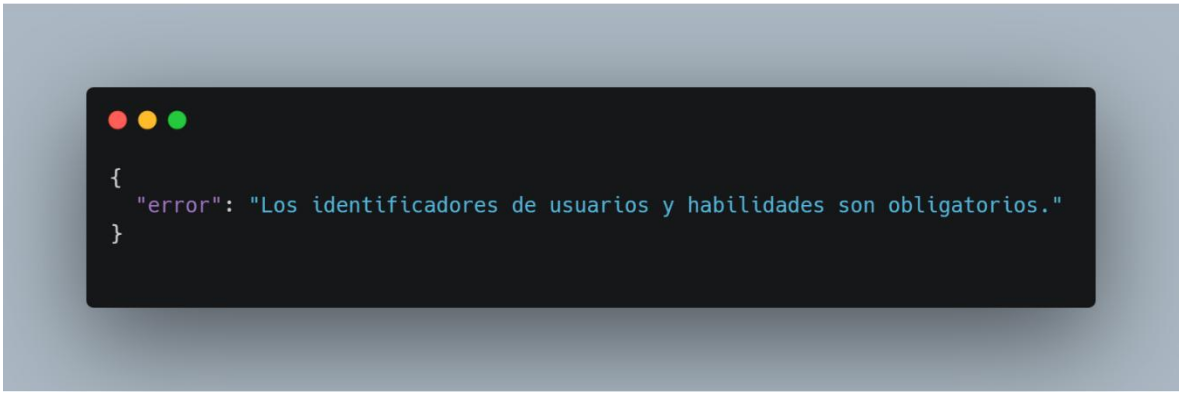


```
{
  "id_solicitante": "64f1c9a2e8b5c30000000001",
  "id_receptor": "64f1c9a2e8b5c30000000002",
  "id_habilidad_solicitada": "64f1d0b4a9c2f20000000001",
  "id_habilidad_ofrecida": "64f1d0b4a9c2f20000000005",
  "mensaje_inicial": "Hola, me interesa tu habilidad de guitarra. Puedo ayudarte con clases de inglés a
cambio."
}
```

Respuesta 201 – Created (éxito):

```
{
  "id": "64f1e2c5b7d3a00000000001",
  "id_solicitante": "64f1c9a2e8b5c30000000001",
  "id_receptor": "64f1c9a2e8b5c30000000002",
  "id_habilidad_solicitada": "64f1d0b4a9c2f20000000001",
  "id_habilidad_ofrecida": "64f1d0b4a9c2f20000000005",
  "estado": "pendiente",
  "mensaje_inicial": "Hola, me interesa tu habilidad de guitarra. Puedo ayudarte con clases de inglés a
cambio.",
  "fecha_creacion": "2025-12-06T16:30:00.000Z",
  "fecha_actualizacion": "2025-12-06T16:30:00.000Z"
}
```

Respuesta 400 – Bad Request (datos inválidos):



4. Planificación y Costos

Esta sección presenta la estimación del esfuerzo necesario para desarrollar el prototipo v0 de SkillTrade y el cálculo del costo teórico del proyecto, considerando una tarifa por hora definida para el rol de desarrolladora.

4.1. Estimación de esfuerzo

Se estima que el desarrollo del prototipo v0 de SkillTrade requiere aproximadamente **54 horas de trabajo**, distribuidas en las siguientes tareas principales:

A. Diseño y planificación – 13 horas

- Análisis de requerimientos funcionales y no funcionales: **4 h**
- Diseño del modelo de datos (colecciones y relaciones en MongoDB): **3 h**
- Diseño de la API (definición de endpoints principales): **2 h**
- Diseño de interfaces: wireframes y estructura de pantallas clave: **4 h**

B. Backend – Node.js + Express + MongoDB – 21 horas

- Configuración inicial del proyecto (estructura, dependencias, entorno): **2 h**
- Implementación de modelo de usuarios y registro/autenticación básica: **3 h**
- Implementación de modelo de habilidades y endpoints asociados: **3 h**

- Implementación de solicitudes de intercambio (creación, listado básico, cambio de estado): **4 h**
- Implementación de sistema de reseñas: **2 h**
- Implementación de mensajería básica por solicitud: **3 h**
- Pruebas de endpoints y ajustes menores: **4 h**

C. Frontend – React – 16 horas

- Configuración inicial del proyecto (Vite/CRA, estructura de carpetas): **2 h**
- Desarrollo de vistas principales: home, login y registro: **4 h**
- Vista de listado y detalle de habilidades: **4 h**
- Vista de solicitudes de intercambio e historial básico: **3 h**
- Integración con la API (llamadas HTTP, manejo de respuestas y errores): **3 h**

D. Pruebas integrales y documentación – 4 horas

- Pruebas funcionales del flujo completo (registro → habilidades → solicitudes): **2 h**
- Elaboración y ajuste de la documentación final (PDF, capturas, README): **2 h**

Total estimado: 13 h + 21 h + 16 h + 4 h = **54 horas**

Esta estimación incluye un pequeño margen de seguridad para imprevistos técnicos y correcciones menores.

4.2. Presupuesto

Para calcular el costo teórico del desarrollo, se define una **tarifa por hora** y se multiplica por el total de horas estimadas.

- **Tarifa por hora:** Q 50.00
- **Horas totales estimadas:** 54 h

Cálculo del costo total:

$$\text{Costo total} = 54\ \text{h} \times Q50.00/\text{h} = Q2{,}700.00$$

Por lo tanto, el costo estimado de desarrollo del prototipo v0 de SkillTrade sería de **Q 2,700.00** si se contratara a una desarrolladora con una tarifa de Q 50.00 por hora.

En el contexto de este proyecto académico:

- No se contratará a terceros.
- Se utilizarán servicios en su versión gratuita (Vercel, Render/Railway, MongoDB Atlas Free Tier).
- El presupuesto se considera **teórico**, con fines de práctica en estimación y planificación de proyectos de software.

5. Conclusión

El desarrollo del proyecto **SkillTrade** permitió integrar los principios fundamentales de la ingeniería de software aplicados al desarrollo web, consolidando un proceso que inicia en la identificación clara de un problema y culmina en el diseño estructurado de una solución viable y técnicamente fundamentada. La plataforma propuesta ofrece un espacio innovador para el intercambio de habilidades sin necesidad de transacciones monetarias, promoviendo el aprendizaje colaborativo y accesible para diversos perfiles de usuarios.

La arquitectura seleccionada —compuesta por React en el frontend, Node.js con Express en el backend y MongoDB como sistema de persistencia— responde a criterios de flexibilidad, escalabilidad y coherencia en el uso de tecnologías modernas. Asimismo, el modelo de datos, los endpoints definidos y las estimaciones de tiempo y costo demuestran la capacidad para planificar un proyecto realista y funcional.

En conjunto, este documento refleja una propuesta sólida, alineada con los objetivos académicos del curso y con estándares profesionales del desarrollo web. SkillTrade se consolida así como un prototipo factible y escalable, con potencial para evolucionar hacia un producto más completo en fases posteriores.