

## Historias de Usuario.

<b>Nº: 01</b>	Creación de Repositorios para el Frontend y Backend
<b>Descripción</b>	Yo como desarrollador, quiero crear dos repositorios separados para el frontend y el backend del proyecto, para poder organizar de manera clara y eficiente el código fuente, facilitar el trabajo colaborativo y permitir despliegues independientes.
<b>Tipo</b>	Tarea
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"><li>• Se debe crear un repositorio para el frontend que contenga la estructura básica del frontend (es decir, el proyecto de React creado con Vite).</li><li>• Se debe crear un repositorio para el backend que contenga la estructura básica del backend (es decir, el archivo <code>main.py</code> para ejecutar el servidor FastAPI).</li><li>• Ambos repositorios deben tener un archivo <code>README.md</code> inicial que describa brevemente su propósito.</li><li>• Ambos repositorios deben contar con un archivo <code>.gitignore</code>.</li><li>• Los repositorios deben estar alojados en la plataforma de control de versiones GitHub.</li></ul>

Table 1: Historia de usuario 1  
Fuente: Elaboración propia.

<b>N°: 02</b>	Creacion y Conexión a las bases de datos
<b>Descripción</b>	Yo como desarrollador quiero que el servidor de FastAPI y el worker de Celery establezcan conexiones simultáneas a las bases de datos PostgreSQL y Redis para asegurar que tanto la API como Celery puedan acceder y manipular los datos necesarios de forma eficiente y coherente.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe de crear un contenedor para la base de datos de redis.</li> <li>• Se debe de crear la base de datos en neonDB.</li> <li>• Se debe tener una conexión a Redis y a NeonDB en el servidor de FastAPI.</li> <li>• Se debe tener una conexión a Redis y a NeonDB en el worker de Celery.</li> </ul>

Table 2: Historia de usuario 2  
Fuente: Elaboración propia.

<b>N°: 03</b>	Registro de usuario
<b>Descripción</b>	Yo como persona quiero registrarme en la plataforma proporcionando nombre, correo electrónico y contraseña para poder acceder al prototipo.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe de tener un formulario de registro en el frontend con los campos: <ul style="list-style-type: none"> <li>– Nombre</li> <li>– Nombre de usuario</li> <li>– Correo electrónico</li> <li>– Contraseña</li> <li>– Confirmar contraseña</li> </ul> </li> <li>• El formulario valida en cliente: <ul style="list-style-type: none"> <li>– Que el correo electrónico tenga formato válido.</li> <li>– Que la contraseña cumpla con un mínimo de 8 caracteres.</li> </ul> </li> </ul>

Table 3: Historia de usuario 3  
Fuente: Elaboración propia.

<b>Nº: 04</b>	Inicio de sesión
<b>Descripción</b>	Como usuario registrado quiero iniciar sesión con mi correo electrónico y contraseña para acceder al prototipo.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Existe un formulario como una nueva página en el frontend con los campos: <ul style="list-style-type: none"> <li>– Email</li> <li>– Contraseña</li> </ul> </li> <li>• Al hacer clic en “Iniciar sesión”, el frontend envía una petición.</li> </ul>

Table 4: Historia de usuario 4  
Fuente: Elaboración propia.

<b>Nº: 05</b>	Recuperación de contraseña
<b>Descripción</b>	Como usuario que olvidó su contraseña quiero recibir un enlace de recuperación en mi correo electrónico para poder restablecer mi contraseña y volver a ingresar al prototipo.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• En la interfaz de login hay un enlace “¿Olvidaste tu contraseña?” que lleva a un formulario de recuperación.</li> <li>• El formulario sólo solicita: <ul style="list-style-type: none"> <li>– Correo electrónico</li> <li>– Confirmar correo electrónico</li> </ul> </li> </ul>

Table 5: Historia de usuario 5  
Fuente: Elaboración propia.

<b>N°: 06</b>	Cambiar contraseña
<b>Descripción</b>	Yo como usuario quiero poder cambiar mi contraseña para mantener segura mi cuenta.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• En la interfaz de login hay un enlace “¿Olvidaste tu contraseña?” que lleva a un formulario de recuperación.</li> <li>• El formulario sólo solicita: <ul style="list-style-type: none"> <li>– Correo electrónico</li> <li>– Confirmar correo electrónico</li> </ul> </li> <li>• El formulario debe solicitar: <ul style="list-style-type: none"> <li>– Nueva contraseña</li> <li>– Confirmación de la nueva contraseña</li> </ul> </li> <li>• El formulario debe validar que la nueva contraseña sea válida (mínimo 8 caracteres, un número, una letra minúscula y otra mayúscula).</li> </ul>

Table 6: Historia de usuario 6  
Fuente: Elaboración propia.

<b>N°: 07</b>	Creación del Home
<b>Descripción</b>	Como usuario registrado en la plataforma, quiero acceder a una página de inicio que contenga una barra de navegación, las lecciones disponibles, un botón para acceder al pool de respuestas incorrectas, un botón para ver mis insignias para tener una visión clara de mi progreso y acceder rápidamente a las funcionalidades principales de la aplicación.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• La página de inicio debe mostrar una barra de navegación fija con un botón el cual debe de ir a las secciones: <ul style="list-style-type: none"> <li>– Mi perfil</li> <li>– Cerrar sesión</li> </ul> </li> <li>• Debe mostrarse una sección central con las lecciones disponibles que el usuario puede comenzar.</li> <li>• Debe haber un botón visible para acceder al “pool de respuestas incorrectas”.</li> <li>• Debe haber un botón para ver las insignias disponibles.</li> <li>• El usuario debe ver su nombre y puntaje actual en la página principal.</li> </ul>

Table 7: Historia de usuario 7  
Fuente: Elaboración propia.

<b>N°: 08</b>	Creación de insignias
<b>Descripción</b>	Yo como usuario, quiero visualizar una sección con las insignias disponibles del prototipo para motivarme a seguir aprendiendo y ver todas las insignias disponibles.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• La sección debe mostrar todas las insignias disponibles en el sistema, incluyendo las bloqueadas y desbloqueadas.</li> <li>• Cada insignia debe mostrar: <ul style="list-style-type: none"> <li>– Nombre</li> <li>– Descripción breve</li> <li>– Requisito de experiencia para desbloquearla</li> </ul> </li> </ul>

Table 8: Historia de usuario 8  
Fuente: Elaboración propia.

<b>N°: 09</b>	Creación de mi perfil
<b>Descripción</b>	Como usuario registrado quiero acceder a una sección “Mi perfil” donde se muestren estadísticas como experiencia acumulada, días consecutivos (Racha de días) y posición del raking para motivarme a seguir utilizando el prototipo.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe mostrar nombre, nombre de usuario y la insignia a la que pertenece según los puntos de experiencia totales obtenidos.</li> <li>• Se debe mostrar la racha de días.</li> <li>• Se debe mostrar la experiencia total acumulada.</li> <li>• Se debe mostrar la posición en el ranking global.</li> </ul>

Table 9: Historia de usuario 9  
Fuente: Elaboración propia.

<b>N°: 10</b>	Creación del pie de pagina
<b>Descripción</b>	Yo como usuario, quiero que la aplicación tenga un pie de pagina visible en la parte inferior de todas las páginas, para poder acceder fácilmente a información como contacto y redes.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• El footer debe mostrarse en todas las vistas de la aplicación.</li> <li>• Debe contener los siguientes elementos: <ul style="list-style-type: none"> <li>– Nombre del prototipo (English with Ogloc).</li> <li>– Enlaces a redes sociales del desarrollador.</li> <li>– Información de contacto.</li> <li>– Año actual.</li> </ul> </li> </ul>

Table 10: Historia de usuario 10  
Fuente: Elaboración propia.

<b>Nº: 11</b>	Creación vista de Preguntas
<b>Descripción</b>	Yo como usuario, quiero acceder a una vista que muestre el título, el texto y las preguntas correspondientes a cada lección, para poder comprender el contenido y responder las preguntas dentro del contexto de aprendizaje.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• La vista debe mostrar el título de la lección.</li> <li>• Debe incluir el texto principal relacionado con la lección.</li> <li>• Debe mostrar las preguntas de la lección una por una junto con sus dos respuestas.</li> <li>• Debe incluir un botón de siguiente para pasar a la siguiente pregunta de la lección.</li> <li>• Debe haber un botón para enviar respuestas y un botón para habilitar el micrófono.</li> </ul>

Table 11: Historia de usuario 11  
Fuente: Elaboración propia.

<b>Nº: 12</b>	Creación de rutas usuario
<b>Descripción</b>	Yo como desarrollador, quiero definir y estructurar las rutas (end-points) del servidor que manejen información sobre el usuario, para asegurar un acceso claro y consistente a la lógica del back-end como el retorno de la información del usuario.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se deben implementar las siguientes rutas: <ul style="list-style-type: none"> <li>– Ruta para obtener la información del usuario.</li> <li>– Ruta para obtener el ranking global.</li> <li>– Ruta para editar información del perfil del usuario.</li> </ul> </li> <li>• Cada ruta debe contener la lógica requerida para que la misma sea funcional.</li> <li>• Cada ruta debe estar protegida por el token JWT.</li> </ul>

Table 12: Historia de usuario 12  
Fuente: Elaboración propia.

<b>N°: 13</b>	Creación de rutas de autenticación
<b>Descripción</b>	Yo como desarrollador quiero definir y estructurar las rutas (end-points) del servidor que realicen la tarea de autenticar el usuario para asegurar un acceso claro al prototipo.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se deben implementar las siguientes rutas: <ul style="list-style-type: none"> <li>– Ruta para logIn.</li> <li>– Ruta para registro de usuario.</li> <li>– Ruta de recuperación de la contraseña (envío del email).</li> <li>– Ruta para cambiar la contraseña (cambio real de la contraseña).</li> <li>– Ruta para verificar el token en caso de cargar el logIn o el Registro en el frontend estando ya autenticado.</li> </ul> </li> <li>• Cada ruta debe contener la lógica suficiente para ser funcional.</li> <li>• Cada ruta debe estar protegida por el token JWT.</li> </ul>

Table 13: Historia de usuario 13  
Fuente: Elaboración propia.

<b>N°: 14</b>	Creación de rutas lecciones
<b>Descripción</b>	Yo como desarrollador quiero definir y estructurar las rutas (end-points) del servidor que retornen las lecciones disponibles a el usuario para asegurar el acceso y uso de las lecciones dentro del prototipo.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se deben implementar las siguientes rutas: <ul style="list-style-type: none"> <li>– Ruta para obtener la información general de las lecciones.</li> <li>– Ruta para obtener la información de una lección en específico.</li> </ul> </li> <li>• Cada ruta debe contener la lógica suficiente para ser funcional.</li> <li>• Cada ruta debe estar protegida por el token JWT.</li> </ul>

Table 14: Historia de usuario 14  
Fuente: Elaboración propia.



<b>Nº: 15</b>	Creación de ruta comparación de respuestas
<b>Descripción</b>	Yo como desarrollador quiero definir y estructurar las rutas (end-points) del servidor que comparan la respuesta que brinda el usuario con la respuesta correcta que da el modelo para asegurar que la respuesta del usuario sea valida.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se deben implementar las siguientes rutas: <ul style="list-style-type: none"> <li>– Ruta para comparar la respuesta del usuario con la del modelo.</li> </ul> </li> <li>• Cada ruta debe contener la lógica suficiente para ser funcional.</li> <li>• Cada ruta debe estar protegida por el token JWT.</li> </ul>

Table 15: Historia de usuario 15  
Fuente: Elaboración propia.

<b>Nº: 16</b>	Protección de paginas
<b>Descripción</b>	Yo como usuario quiero que el prototipo tenga autenticación de las paginas privadas, para asegurarme de que mi información y el acceso al prototipo estén protegidos.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe añadir la lógica para asegurarse de que se tiene y se envía el token JWT en cada petición al backend.</li> <li>• Se debe añadir la lógica para asegurarse que, en caso de que no esté el token en el local storage o que no sea válido, redirija al logIn.</li> </ul>

Table 16: Historia de usuario 16  
Fuente: Elaboración propia.

<b>Nº: 17</b>	Integración de Celery
<b>Descripción</b>	Yo como desarrollador, quiero poder generar nuevas lecciones de forma periódica para que el prototipo presente nuevas lecciones cada cierto periodo de tiempo.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe implementar el gestor de tareas Celery dentro del backend como un servicio independiente del servidor de FastAPI.</li> <li>• Se debe implementar la función de Celery Beat con el objetivo de que envíe la señal de inicio para que Celery genere las lecciones en un momento determinado.</li> </ul>

Table 17: Historia de usuario 17  
Fuente: Elaboración propia.

<b>Nº: 18</b>	Generación de texto
<b>Descripción</b>	Yo como desarrollador quiero integrar el modelo de generador de texto seleccionado (gpt2) para generar el texto de las lecciones.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe descargar localmente el modelo GPT-2 small alojado en Hugging Face.</li> <li>• Se debe crear una clase que utilice el modelo mencionado para generar texto.</li> <li>• Se debe implementar el patrón singleton en la clase del modelo, con el objetivo de evitar que se cargue múltiples veces en memoria.</li> </ul>

Table 18: Historia de usuario 18  
Fuente: Elaboración propia.

<b>N°: 19</b>	Generación de preguntas y respuestas
<b>Descripción</b>	Yo como desarrollador quiero integrar los modelos de generador de preguntas y respuestas seleccionados para generar las preguntas y respuestas correspondientes al texto de las lecciones.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe descargar localmente los modelos de la plataforma de Hugging Face: <ul style="list-style-type: none"> <li>– t5-large-generation-race-QuestionAnswer</li> <li>– t5-large-generation-squad-QuestionAnswer</li> </ul> </li> <li>• Se debe crear una clase por cada modelo.</li> <li>• Se debe implementar el patrón singleton en ambas clases, con el objetivo de evitar que se carguen múltiples veces en memoria.</li> </ul>

Table 19: Historia de usuario 19  
Fuente: Elaboración propia.

<b>N°: 20</b>	Generación de respuestas incorrectas
<b>Descripción</b>	Yo como desarrollador quiero integrar el modelo de generador de respuestas incorrectas seleccionado para generar las respuestas distractoras correspondientes a cada pregunta de cada lección.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe descargar localmente el modelo de la plataforma de Hugging Face: <ul style="list-style-type: none"> <li>– t5-large-generation-race-Distractor</li> </ul> </li> <li>• Se debe crear una clase para el modelo.</li> <li>• Se debe implementar el patrón singleton en la clase, con el objetivo de evitar que se cargue múltiples veces en memoria.</li> </ul>

Table 20: Historia de usuario 20  
Fuente: Elaboración propia.

<b>N°: 21</b>	Comparación de respuestas
<b>Descripción</b>	Yo como desarrollador quiero integrar el modelo de comparación de respuestas seleccionado para comparar la respuesta brindada por el usuario y la generada por los demás modelos y saber si son iguales o no.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Yo como desarrollador quiero integrar el modelo de comparación de respuestas seleccionado para: <ul style="list-style-type: none"> <li>– Comparar la respuesta brindada por el usuario y la generada por los demás modelos.</li> <li>– Saber si ambas respuestas son iguales o no.</li> </ul> </li> </ul>

Table 21: Historia de usuario 21  
Fuente: Elaboración propia.

<b>N°: 22</b>	Generación de lecciones
<b>Descripción</b>	Yo como desarrollador quiero generar una lección completa utilizando los modelos ya implementados para que el contenido (texto, preguntas, respuestas correctas e incorrectas) se produzca de forma automática y coherente.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe de seguir un flujo coherente de generación: <ul style="list-style-type: none"> <li>– Primero texto.</li> <li>– Luego preguntas y respuestas.</li> <li>– Finalmente respuesta incorrecta.</li> </ul> </li> </ul>

Table 22: Historia de usuario 22  
Fuente: Elaboración propia.

<b>N°: 23</b>	Guardar las lecciones
<b>Descripción</b>	Yo como desarrollador quiero guardar las lecciones que fueron generadas en la base de datos PostgreSQL y en Redis para que el usuario pueda verlas e interactuar con ellas dentro del prototipo.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Se debe guardar la lección y las preguntas generadas en las tablas correspondientes en PostgreSQL.</li> <li>• Se debe guardar la información general de cada lección dentro de Redis en los sets y hash correspondientes.</li> </ul>

Table 23: Historia de usuario 23  
Fuente: Elaboración propia.

<b>N°: 24</b>	Creación de rutas preguntas Incorrectas
<b>Descripción</b>	Yo como desarrollador quiero definir y estructurar las rutas (end-points) del servidor que brinden las preguntas que ha fallado el usuario para asegurar su disponibilidad para que el usuario las resuelva nuevamente.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Contenedor para el servidor de FastAPI.</li> <li>• Contenedor para el worker de Celery.</li> <li>• Contenedor para Celery Beat.</li> <li>• Se debe generar un archivo <code>docker-compose</code> para la correcta orquestación de los contenedores.</li> </ul>

Table 24: Historia de usuario 24  
Fuente: Elaboración propia.

<b>N°: 25</b>	Creación de contenedores
<b>Descripción</b>	Yo como desarrollador quiero crear contenedores Docker para el servidor FastAPI y el worker de Celery para aprovechar una base Linux en los contenedores y asegurar un mejor rendimiento de Celery, el servidor y de los modelos de PLN.
<b>Tipo</b>	Backend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• Contenedor para el servidor de FastAPI.</li> <li>• Contenedor para el worker de Celery.</li> <li>• Contenedor para Celery Beat.</li> <li>• Se debe generar un archivo <b>docker-compose</b> para la correcta orquestación de los contenedores.</li> </ul>

Table 25: Historia de usuario 25  
Fuente: Elaboración propia.

<b>N°: 26</b>	Diseño responsive
<b>Descripción</b>	Yo como usuario quiero que la aplicación tenga un diseño responsive para que pueda usarla cómodamente desde cualquier dispositivo, ya sea escritorio, tablet o móvil.
<b>Tipo</b>	Frontend
<b>Criterios de Aceptación</b>	<ul style="list-style-type: none"> <li>• La interfaz se adapta correctamente a diferentes tamaños de pantalla (escritorio, tablet y móvil).</li> <li>• Los elementos importantes son visibles y accesibles en todos los dispositivos.</li> <li>• No hay desbordamientos horizontales ni contenido cortado.</li> </ul>

Table 26: Historia de usuario 26  
Fuente: Elaboración propia.