

Historias Técnicas

Nombre: Alejandro Hidalgo

Código: 6823

Historia Técnica #1: Configuración de TensorFlow y Keras

Descripción:

El sistema requiere TensorFlow y Keras para implementar redes neuronales convolucionales (CNN), **para poder** procesar imágenes y detectar emociones faciales en tiempo real.

Validación:

- TensorFlow y Keras configurados correctamente en el entorno.
- Librerías funcionales para entrenamiento y evaluación de modelos de detección emocional.
- TensorFlow y Keras deben ser instalados correctamente en el entorno de desarrollo.
- Comprobar que las versiones de TensorFlow y Keras sean compatibles entre sí y con las bibliotecas de Python utilizadas.
- Ejecutar una prueba de "hello world" (como un modelo simple de regresión o clasificación) para verificar que TensorFlow y Keras están funcionando correctamente en el entorno.
- Configurar el entorno para que pueda ser utilizado para el entrenamiento de modelos de detección facial, sin necesidad de entrenar el modelo en este sprint.
- Incluir una guía de instalación detallada para que otros miembros del equipo puedan configurar fácilmente TensorFlow y Keras en sus entornos locales.

Valor: 400

Prioridad: Alta

Estimación: 12H

Historia Técnica #2: Configuración de Librosa para Análisis de Audio

Descripción:

El sistema requiere la librería Librosa, **para poder** analizar características de audio y entrenar un modelo de clasificación emocional basado en voz.

Validación:

- Librosa instalada y funcional en el entorno.
- Extracción de características de audio probada con muestras iniciales.
- Asegurar que el entorno esté configurado correctamente para procesar archivos de audio, con soporte para formatos comunes como WAV, MP3, y FLAC.

Valor: 350

Prioridad: Alta

Estimación: 10H

Historia Técnica #3: Evaluación Automatizada del Modelo CNN

Descripción:

Implementar una función que evalúe automáticamente el desempeño del modelo CNN con métricas estándar (accuracy, precision, recall, F1-score), para validar su rendimiento de forma objetiva.

Validación:

- Ejecución automática de evaluaciones al finalizar cada ciclo de entrenamiento.
- Visualización clara de métricas en consola o archivo log.
- Exportación de métricas en formato CSV.

Valor: 350

Prioridad: Media

Estimación: 4H

Historia Técnica #4: Diseño de Base de Datos

Descripción:

El sistema necesita una base de datos PostgreSQL, **para poder** almacenar emociones detectadas, sesiones y registros de interacción.

Validación:

- Diseño del esquema relacional de la base de datos.
- Base de datos exportada y funcional en PostgreSQL.
- Verificar que las relaciones entre las tablas estén correctamente implementadas, por ejemplo, claves foráneas entre usuarios y sesiones, o entre sesiones y emociones_detectadas.

- Configurar la conexión entre el backend y PostgreSQL, asegurando que el sistema pueda almacenar y recuperar datos de manera eficiente.
- Asegurar que la base de datos pueda manejar una cantidad razonable de registros sin experimentar caídas de rendimiento.

Valor: 300

Prioridad: Alta

Estimación: 4H

Historia Técnica #5: Implementación de Módulo de Visualización de Resultados

Descripción:

Desarrollar el módulo que visualiza los resultados de las emociones detectadas en tiempo real, utilizando gráficos y notificaciones visuales, para facilitar el análisis por parte del usuario.

Validación:

- Gráficos generados en tiempo real usando datos del sistema.
- Uso de librerías como Chart.js, D3.js o Matplotlib.
- Componente React para mostrar estos gráficos en el dashboard.

Valor: 300

Prioridad: Alta

Estimación: 6H

Historia Técnica #6: Implementación de Seguridad con Encriptación

Descripción:

El sistema necesita implementar encriptación, **para poder** garantizar la privacidad de los datos emocionales capturados.

Validación:

- Transferencia de datos segura validada con pruebas.
- Realizar pruebas básicas de seguridad para asegurarse de que no haya vulnerabilidades en la transmisión de datos.
- Verificar que la base de datos esté protegida mediante encriptación (si es necesario) y que los datos sensibles no puedan ser accesibles sin autorización.

Valor: 200

Prioridad: Alta

Estimación: 6H

Historia Técnica #7: Configuración de Matplotlib

Descripción:

El sistema requiere Matplotlib, **para poder** generar gráficos interactivos que muestren tendencias emocionales.

Validación:

- Gráficos generados correctamente con filtros funcionales.
- Validar que se pueden generar gráficos básicos como barras, líneas o dispersión utilizando datos emocionales.
- Verificar que los gráficos de Matplotlib se integren correctamente con el sistema de visualización de emociones y se actualicen en tiempo real.
- Incluir ejemplos básicos de cómo generar gráficos utilizando Matplotlib, y cómo integrar esos gráficos en el frontend.

Valor: 200

Prioridad: Media

Estimación: 6H

Historia Técnica #8: Configuración del Backend en Python

Descripción:

Configurar el backend en Python para manejar la lógica central del sistema, exponer APIs REST y generar gráficos de interacción.

Criterios de Aceptación:

- El backend en Flask recibe solicitudes y devuelve respuestas.
- Se implementa una ruta /api para procesar las solicitudes .
- Se implementa /api/graph para generar gráficos basados en interacciones y devolverlos como imágenes.
- Conexión exitosa con la base de datos PostgreSQL.

- Pruebas unitarias de las rutas con Postman o cURL.

Valor: 400

Prioridad: Alta

Estimación: 10H

Historia Técnica #9: Configuración Inicial del entorno de React para frontend principal

Descripción:

Configurar React como framework frontend para establecer el entorno de desarrollo, inicializar el proyecto, definir una estructura organizada y crear componentes reutilizables base que serán usados en todo el sistema.

Criterios de Aceptación:

- React debe ser configurado correctamente utilizando herramientas estándar como create-react-app o configuraciones personalizadas.
- Crear carpetas organizadas para componentes, estilos, servicios y utilidades.
- Implementar componentes genéricos como botones, formularios, y contenedores reutilizables.
- Validar que los componentes sean responsivos y estilizados
- Verificar que el proyecto se ejecute correctamente en navegadores modernos como Chrome, Firefox y Edge.
- Asegurar tiempos de carga inicial razonables (menos de 5 segundos bajo condiciones normales).

Valor: 300

Prioridad: Alta

Estimación: 6H

Historia Técnica #10: Implementación de Pantallas de Inicio de Sesión y Registro

Descripción:

Diseñar e implementar las pantallas de inicio de sesión y registro, asegurando usabilidad y responsividad.

Validación:

- Asegurarse de que el framework de diseño (Material-UI o TailwindCSS) esté instalado correctamente a través de npm install o yarn add y que esté correctamente configurado en el proyecto.

- Crear y verificar componentes básicos (por ejemplo, un botón, un campo de texto, etc)
- Campos de formulario validados (email, contraseña, etc.).
- Verificar que los componentes interactivos (como botones, formularios) funcionen correctamente, con respuestas visuales adecuadas
- Integración completa con la API del backend para gestionar usuarios.

Valor: 250

Prioridad: Media

Estimación: 8H

Historia Técnica #11: Conexión del Frontend con el Backend

Descripción:

El sistema requiere conectar el frontend (React) con el backend python, **para poder** enviar y recibir datos eficientemente.

Validación:

- Flujo de inicio de sesión y registro probado y funcional.
- Comunicación con el microservicio establecida y probada para registrar interacciones.
- Pruebas exitosas de intercambio de datos entre frontend y backend.
- Realizar pruebas de integración en las que se envíen y reciban datos entre el frontend y el backend
- Validar que los datos enviados por el frontend se reflejen correctamente en el backend.

Valor: 350

Prioridad: Alta

Estimación: 8H

Historia Técnica #12: Creación del Prototipo de la Interfaz de Usuario

Descripción:

Como equipo de diseño y desarrollo, queremos crear un prototipo de la interfaz de usuario. El prototipo debe reflejar las principales pantallas de la aplicación (dashboard,

visualización de emociones, configuración de sesiones, etc.) y ser completamente interactivo para simular las interacciones del usuario. El objetivo es validar los flujos de usuario, la disposición de los elementos y la navegación antes de comenzar el desarrollo real con React.

Validación:

- El prototipo debe incluir las pantallas principales del sistema, como el dashboard, visualización de emociones, y la configuración de sesiones.
- El prototipo debe ser interactivo, permitiendo simular clics, desplazamientos y cambios de pantalla como si fuera la aplicación real.
- Los stakeholders y usuarios finales deben poder navegar el prototipo para evaluar la experiencia de usuario (UX) y realizar comentarios.
- El diseño del prototipo debe estar alineado con los requisitos del sistema, como los gráficos de emociones y las funcionalidades de filtrado.
- Debe incluir el flujo completo de la interfaz, permitiendo que se puedan realizar pruebas de usabilidad con usuarios reales.
- El prototipo debe ser entregado en un formato que pueda ser probado y validado
- Comentarios y ajustes deben ser posibles antes de pasar a la fase de desarrollo de la interfaz.

Valor: 300

Prioridad: Alta

Estimación: 8 horas

Historia Técnica #13: Configuración Avanzada del Frontend en React.js

Descripción:

Configurar y estructurar el entorno del frontend utilizando React.js, asegurando una arquitectura escalable, componentes reutilizables y una base sólida para el desarrollo de la interfaz gráfica del sistema web.

Criterios de Aceptación:

- Inicialización del proyecto con create-react-app o configuración manual con Webpack.
- Estructura organizada por carpetas: components/, pages/, services/, assets/, utils/.
- Integración de TailwindCSS o Material UI para el diseño visual.
- Implementación de un layout base con barra lateral, cabecera y espacio de contenido.

- Comprobación de compatibilidad con navegadores modernos.
- Verificación de tiempos de carga inicial inferiores a 5 segundos bajo condiciones normales.

Valor: 350

Prioridad: Alta

Estimación: 4H

Historia Técnica #14: Entrenamiento de Modelos de Reconocimiento Emocional

Descripción: Entrenar modelos de reconocimiento de emociones faciales, asegurando precisión, y evaluar su desempeño.

Criterios de Aceptación:

- Los modelos de reconocimiento facial y de voz son entrenados con datasets específicos que incluyen al menos 6 emociones principales (felicidad, tristeza, enojo, sorpresa, miedo y neutralidad).
- Métricas de desempeño calculadas, incluyendo F1-Score y Accuracy.
- Ajustar hiperparámetros (como tasa de aprendizaje y número de épocas) para mejorar la precisión del modelo.
- Exportar los modelos entrenados en un formato compatible (por ejemplo, .h5 para TensorFlow o .pt para PyTorch) y probar la carga de los modelos en el sistema.
- Validar que los modelos entrenados pueden procesar imágenes y audios en tiempo real.

Valor: 400

Prioridad: Alta

Estimación: 12 horas

Historia Técnica #15: Implementación de Redirección Basada en Roles

Descripción:

El sistema necesita implementar la lógica para redirigir automáticamente al menú correspondiente según el rol del usuario (administrador o usuario regular) al iniciar sesión.

Validación:

- El sistema debe identificar el rol del usuario al iniciar sesión y almacenarlo en localStorage o token.
- Los usuarios deben ser redirigidos automáticamente a:

- Administradores: menu_admin.js.
- Usuarios: menu_usuario.js.
- Proteger las rutas para evitar que usuarios no autenticados accedan a menús sin permiso.
- Validar que los usuarios no autenticados sean redirigidos al inicio de sesión.

Valor: 300

Prioridad: Alta

Estimación: 8H

Historia Técnica #16: Desarrollo de Menús Dinámicos para Usuarios y Administradores

Descripción:

Implementar menús para usuarios y administradores. Cada menú debe ofrecer las opciones correspondientes según el rol del usuario al iniciar sesión. Los usuarios regulares tendrán acceso al sistema de reconocimiento, mientras que los administradores tendrán un menú básico con cierre de sesión y otras funcionalidades que serán agregadas en futuros sprints.

Validación:

1. Implementar un componente React que gestione los menús dinámicamente según el rol:
 - **Usuarios:**
 - Botón para redirigir al reconocimiento emocional.
 - Botón para cerrar sesión, eliminando los datos de sesión en localStorage y redirigiendo al inicio de sesión.
 - **Administradores:**
 - Botón para cerrar sesión, eliminando los datos de sesión en localStorage y redirigiendo al inicio de sesión.
 - Mensaje indicando que las funcionalidades administrativas estarán disponibles en futuros sprints.
2. Asegurar que cada menú sea accesible únicamente para usuarios con el rol correspondiente:
 - Usuarios con rol usuario acceden al menú de usuario.
 - Usuarios con rol administrador acceden al menú de administrador.

3. Proteger las rutas para evitar que usuarios no autenticados accedan a los menús.
4. Validar que las opciones de redirección funcionen correctamente durante pruebas funcionales.

Valor: 300

Prioridad: Alta

Estimación: 4H

Historia Técnica #17: Desacoplamiento de la Base de Datos

Descripción:

El sistema debe ser independiente del motor de base de datos, permitiendo la compatibilidad con distintos gestores de bases de datos (ejemplo: MySQL, PostgreSQL, SQL Server, etc.), sin necesidad de realizar cambios significativos en el código.

Criterios de Aceptación:

- Utilizar JPA (Java Persistence API) con Hibernate o Spring Data JPA para la capa de acceso a datos, evitando consultas SQL directas específicas de un motor.
- Configurar perfiles de conexión en application.properties o application.yml para que el sistema pueda cambiar de base de datos fácilmente.
- Utilizar un pool de conexiones (HikariCP, Apache DBCP, etc.) para manejar eficientemente la comunicación con distintos motores de base de datos.
- Implementar migraciones con Flyway o Liquibase, para que la estructura de la base de datos pueda crearse y actualizarse independientemente del motor utilizado.
- Probar el sistema con al menos dos motores de base de datos diferentes (ejemplo: MySQL y PostgreSQL) para validar la compatibilidad.
- Evitar consultas SQL nativas específicas de un motor y, en su lugar, utilizar JPQL (Java Persistence Query Language) o consultas con Criteria API.
- Documentar el proceso de configuración para cambiar de un motor de base de datos a otro en el entorno de desarrollo y producción.

Valor: 350

Prioridad: Alta

Estimación: 10H

Historia Técnica #18: Evaluación Automatizada del Modelo CNN

Descripción:

Implementar una función que evalúe automáticamente el desempeño del modelo CNN con métricas estándar (accuracy, precision, recall, F1-score).

Validación:

- Ejecución automática de evaluaciones al finalizar cada ciclo de entrenamiento.
- Visualización clara de métricas en consola o archivo log.
- Exportación de métricas en formato CSV.

Valor: 350

Prioridad: Alta

Estimación: 8H

Historia Técnica #19: Integración con Power BI para Explotación de Datos Emocionales**Descripción:**

Como administrador, quiero que el sistema permita conectar los datos emocionales recolectados con una herramienta externa como Power BI, para realizar análisis avanzados y visualizaciones personalizadas que mejoren la toma de decisiones.

Criterios de Aceptación:

- Los datos exportables deben incluir: fecha, emoción detectada, duración, intensidad, ID de sesión y usuario (cuando aplique).
- El sistema debe garantizar la seguridad de los datos durante la exportación.
- La integración debe estar documentada para permitir que otros usuarios configuren su conexión desde Power BI Desktop.

Valor: 300

Prioridad: Alta

Estimación: 8H

Historia Técnica #20: Bloqueo Temporal por Múltiples Intentos Fallidos de Inicio de Sesión**Descripción:**

Implementar una funcionalidad de seguridad que bloquee temporalmente la cuenta de

usuario tras varios intentos fallidos de inicio de sesión, para prevenir accesos no autorizados por fuerza bruta.

Criterios de Aceptación:

- El sistema debe permitir un máximo de 5 intentos fallidos consecutivos.
- Si se superan los intentos, la cuenta se bloquea durante 15 minutos.
- El mensaje debe indicar claramente que la cuenta está temporalmente bloqueada.
- Cada intento debe registrarse en la base de datos con fecha, IP y usuario.
- Un administrador debe poder desbloquear manualmente una cuenta si es necesario.
- El contador debe reiniciarse automáticamente tras un inicio de sesión exitoso.

Valor: 300

Prioridad: Alta

Estimación: 6H