**Historias de usuario y Historias Tecnicas:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Tipo** | **Descripción** | **Estimación (horas)** | **Prioridad** | **Criterios de Validación** |
| HT-001 | Técnica | Como desarrollador, quiero configurar el entorno de desarrollo para el modelo SVM, para garantizar un proceso de entrenamiento eficiente. | 8 | 1 | - El entorno debe incluir herramientas para el entrenamiento y validación. - Incluir GPU si es necesario para SVM. |
| HU-001 | Usuario | Como administrador, quiero entrenar y validar el modelo SVM con datos nutricionales, para que el sistema ofrezca recomendaciones precisas. | 16 | 2 | - El modelo debe alcanzar al menos un 85% de precisión (RNF-003). - El tiempo de respuesta del modelo debe ser menor a 2 segundos (RNF-001). |
| HT-002 | Técnica | Como ingeniero de datos, quiero diseñar y normalizar la base de datos nutricional, para que los datos sean consistentes y procesables. | 12 | 3 | - La base de datos debe permitir consultas eficientes. - Soporte para escalabilidad horizontal (RNF-014). |
| HU-002 | Usuario | Como administrador, quiero configurar un sistema de aprendizaje continuo para actualizar el modelo SVM, para mejorar las recomendaciones con el tiempo. | 14 | 4 | - El modelo debe aceptar nuevos datos y actualizarse sin interrumpir el servicio. - Las actualizaciones deben respetar los estándares de privacidad (RNF-004). |
| HT-003 | Técnica | Como administrador, quiero implementar encriptación AES-256 para todos los datos sensibles, para garantizar la privacidad de los estudiantes. | 10 | 5 | - Los datos en tránsito y en reposo deben estar encriptados. - Cumplir con RNF-004 y políticas de seguridad establecidas. |
| HU-003 | Usuario | Como administrador, quiero garantizar la calidad y consistencia de los datos nutricionales, para que el modelo SVM pueda entrenarse correctamente. | 10 | 6 | - Los datos deben ser verificados automáticamente antes de ser utilizados. - Cumplir con estándares de seguridad y encriptación (RNF-004). |
| HT-004 | Técnica | Como desarrollador, quiero implementar y probar la API que conecta el modelo SVM con la interfaz web, para que los estudiantes reciban recomendaciones. | 16 | 7 | - La API debe responder en menos de 2 segundos (RNF-001). - Documentación generada automáticamente (RNF-011). |
| HU-004 | Usuario | Como nutricionista, quiero analizar patrones alimenticios de los estudiantes mediante el modelo SVM, para optimizar sus hábitos según sus actividades. | 12 | 8 | - El sistema debe procesar datos históricos y ofrecer análisis predictivos. - Resultados exportables en CSV o JSON. - Precisión del análisis >= 85%. |
| HT-005 | Técnica | Como ingeniero, quiero establecer pruebas automatizadas para evaluar la precisión del modelo SVM, para garantizar su efectividad. | 12 | 9 | - La precisión del modelo debe ser mayor al 85% (RNF-003). - Implementar pruebas unitarias y de integración para garantizar resultados consistentes. |
| HU-005 | Usuario | Como estudiante, quiero recibir recomendaciones alimenticias personalizadas basadas en el modelo SVM, para poder mejorar mi dieta según mis necesidades académicas. | 16 | 10 | - Las recomendaciones deben basarse en patrones de consumo y preferencias. - Tiempo de respuesta menor a 2 segundos. - Cumplir con el 85% de precisión (RNF-003). |
| HT-006 | Técnica | Como desarrollador, quiero crear un sistema de autenticación basado en roles, para controlar el acceso a los recursos del sistema. | 10 | 11 | - Implementar roles básicos (estudiante, nutricionista, administrador). - Asegurar que las contraseñas cumplan con RNF-006. |
| HU-006 | Usuario | Como administrador, quiero que el sistema permita probar las recomendaciones generadas por el modelo SVM, para garantizar su calidad antes de desplegar la interfaz. | 8 | 12 | - Crear un módulo de pruebas para validar las recomendaciones. - Registro automático de problemas y resultados incorrectos. |
| HT-007 | Técnica | Como administrador, quiero configurar el sistema de backups automáticos, para evitar pérdida de datos en caso de fallos. | 8 | 13 | - Backups diarios configurados (RNF-005). - Validar integridad de los datos recuperados de los backups. |
| HU-007 | Usuario | Como estudiante, quiero acceder a una interfaz adaptativa para recibir recomendaciones alimenticias, para que sea fácil de usar desde cualquier dispositivo. | 10 | 14 | - La interfaz debe ser responsive (RNF-007). - Compatible con principales navegadores (RNF-017). - Diseño centrado en el usuario. |
| HT-008 | Técnica | Como desarrollador, quiero optimizar el tiempo de respuesta de las consultas a la base de datos, para cumplir con los tiempos requeridos. | 10 | 15 | - Las consultas deben ejecutarse en menos de 100 ms en promedio. - Uso eficiente de índices en las tablas más consultadas. |
| HU-008 | Usuario | Como nutricionista, quiero que el sistema optimice automáticamente los menús recomendados usando SVM, para ofrecer mejores opciones a los estudiantes. | 14 | 16 | - El sistema debe incluir un módulo de optimización. - Resultados ajustados a los objetivos del usuario según el análisis predictivo. |
| HT-009 | Técnica | Como administrador, quiero configurar un entorno de pruebas para simular 1000 usuarios concurrentes, para validar la escalabilidad del sistema. | 10 | 17 | - El entorno debe simular cargas de tráfico reales. - Cumplir con el requisito de soporte para 1000 usuarios concurrentes (RNF-002). |
| HU-009 | Usuario | Como estudiante, quiero ver un dashboard de progreso nutricional, para monitorear mis hábitos alimenticios y mis avances. | 12 | 18 | - El dashboard debe mostrar gráficos claros y fáciles de entender. - Compatible con dispositivos móviles. - El diseño debe ser intuitivo para usuarios nuevos (RNF-008). |
| HT-010 | Técnica | Como desarrollador, quiero integrar la exportación de datos en formatos CSV y JSON, para que los datos sean compatibles con otros sistemas. | 6 | 19 | - Exportación disponible desde la interfaz web. - Cumplir con los estándares definidos (RNF-020). |

**Propuesta de tecnología q voy a utilizar**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterio** | **Opción Elegida** | **Opción No Elegida 1 (JavaScript/Node.js)** | **Opción No Elegida 2 (Java)** |
| **Lenguaje de Programación** | **Python** | **JavaScript (Node.js)** | **Java** |
| **Por qué se eligió/no se eligió** | Python es el líder en Machine Learning, con una amplia gama de bibliotecas especializadas y optimizadas (como Scikit-learn, TensorFlow). Es ideal para tareas de análisis predictivo y modelos SVM. | Aunque JavaScript es muy versátil y utilizado para aplicaciones web completas, no tiene el mismo nivel de soporte ni eficiencia en el área de Machine Learning como Python. | Java es robusto y adecuado para aplicaciones empresariales, pero no tiene tantas bibliotecas optimizadas para Machine Learning como Python. Además, tiene una curva de aprendizaje más alta. |
| **Frameworks** | **Django** | **React + Express.js** | **Spring Boot** |
| **Por qué se eligió/no se eligió** | Django proporciona una estructura robusta para el backend, con seguridad integrada y facilidad para conectar bases de datos relacionales, lo cual es ideal para la gestión de usuarios y datos nutricionales. | React + Express.js es excelente para aplicaciones web, pero Express no ofrece tantas herramientas integradas para la gestión de backend como Django. Además, Node.js no es el mejor para tareas de Machine Learning. | Spring Boot es adecuado para aplicaciones grandes y complejas, pero su configuración es más compleja y no está tan optimizado para trabajar con Machine Learning, lo que lo hace menos eficiente para tu proyecto. |
| **Base de Datos** | **PostgreSQL** | **MongoDB** | **MySQL** |
| **Por qué se eligió/no se eligió** | PostgreSQL es una base de datos relacional avanzada que maneja de forma eficiente las consultas SQL complejas, ideal para datos estructurados como los nutricionales y las preferencias de los usuarios. | MongoDB es una base de datos NoSQL flexible y escalable, pero no es ideal para manejar datos estructurados ni realizar consultas complejas, lo cual es necesario para este proyecto. | MySQL es confiable, pero menos avanzado que PostgreSQL para manejar consultas complejas y menos flexible que PostgreSQL en términos de rendimiento con datos estructurados. |
| **IDE** | **Visual Studio Code** | **PyCharm** | **IntelliJ IDEA** |
| **Por qué se eligió/no se eligió** | Visual Studio Code es un IDE ligero y versátil que soporta múltiples lenguajes y tecnologías, ideal para proyectos que involucren Python, JavaScript y Machine Learning. Además, tiene muchas extensiones útiles. | PyCharm es excelente para Python, pero Visual Studio Code es más versátil y consume menos recursos, permitiendo trabajar con más tecnologías y lenguajes a la vez sin perder rendimiento. | IntelliJ IDEA es ideal para proyectos en Java, pero consume más recursos y no es tan adecuado para proyectos que involucren una combinación de tecnologías como Python y JavaScript. |