|  |  |
| --- | --- |
| Épica | **N/A** |
| Historia de usuario | **HU001\_ServicioNotificacionesPush** |
| Fecha de elaboración | **20/09/2024** |
| Versión de documento | **V1** |
| Elaborado por | **Santiago Montañez** |

Control de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Autor** | **Cambios realizados** |
| 1.0 | 20/09/2024 |  | V1 |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **DEFINICIÓN** | | |
| **1.Prioridad del Negocio:** | | **2.Beneficio que se obtiene con esta HU: Fomentar el aprendizaje constante de los usuarios de las aplicaciones de CREA.** |
| **3. Descripción de la Historia:**  Yo como cliente, requiero que se habiliten notificaciones del progreso o avance de mi formación en un periodo de tiempo determinado (diariamente). | | |
| **4. Dependencia:** | Accesos bases de datos para conexión | |
| **5. Descripción del flujo**:  Actualmente las aplicaciones de CREA muestran el progreso de los usuarios como un valor en porcentaje, por lo cual se requiere que luego de un periodo de tiempo especificado en el que el progreso no cambie de valor se envíe una notificación dentro de la aplicación a los usuarios (por roles) y de esta forma se haga un recordatorio de la necesidad de continuar avanzando en su aprendizaje.  Opciones de tecnologías (Node.JS, Python y .NET)  NodeJS:   * Buen soporte para MongoDB. * Librerías oficiales de Firebase. * APIs REST.   Python:   * APIs REST con Django REST Framework. * El soporte para bases de datos NoSQL en Django no es nativo como en otros frameworks.   .NET:   * Buen soporte para APIs REST. * Configuración compleja.   SPRINT 1:   * Definir los requerimientos del servicio * Configurar el entorno de desarrollo (elegir tecnología) * Crear y estructurar el proyecto (API REST) * Implementar conexión a ambas bases de datos (Firebase y MongoDB)   Entrega: conexión a ambas bases de datos  SPRINT 2:   * Implementación de endpoints necesarios (estilo crud) * Crear y probarlas operaciones básicas crud requeridas * Incluirla lógica de autenticación   Entrega: Operaciones crud requeridas  SPRINT 3:   * Diseñar lógica para envío de notificaciones cada 24 horas * Implementar los endpoints que activan las notificaciones * Integrar el servicio de notificaciones push   Entrega: Notificaciones automáticas cada 24 horas  SPRINT 4:   * Realizar pruebas de las notificaciones * Configurar y probar el entorno de producción * Desplegar el servicio web   Entrega: Servicio Funcional desplegado | | |
| **6. Anexo del Flujo de la HU: N/A** | | |
| **7. Criterios de Aceptación: Recepción de notificaciones push en la aplicación por parte del usuario** | | |
| **Glosario de Términos:**   * CRUD (Create, Read, Update, Delete) | | |
| **DEFINICIÓN TÉCNICA** | | |
|  | | |

Aws

Azure

Google cloud mas complicado fire base

Digital ocean (Droplets, web app, contenedores Docker, se pueden funciones estáticas), averiguar si realmente la ia esta hecha en Python si es algo de machine learning o Deep learning en cuyo caso seria un pipeline, o si es una url de consumo a otra ia externa hecha en Python,