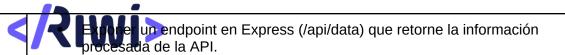




HISTORIAS DE USUARIOS

Nombre de la HU:		Integración y creación de APIs con Node				
Objetivo de la HU		Como inversionista, quiero crear una aplicación que consuma APIs externas de cualquier tema que deseen implementar, usando axios y que ejecute cronjobs para enviar emails de notificación de interaciones, con el fin de aprender a integrar servicios externos y automatizar procesos. Aceptando ideas creativas e innovadoras.				
	Configurar el entorno de desarrollo (Node.js + editor).					
TASK1	 Crear un proyecto base con npm init -y. Agregar el tsconfig con tsc –ini Modificar el tsconfig para uso de node Instalar dependencias iniciales: express, axios, node-cron, nodemailer Instalar dependencias de desarrollo con @types/<librería></librería> 					
TASK2	Crear la estructura base de la app con Express.					
	 Crear un archivo principal app.js o index.js. Configurar el servidor Express en un puerto (ej. 3000). Crear la carpeta /routes para manejar endpoints. Probar el servidor con un endpoint básico (GET /ping). 					
	Integrar consumo de APIs externas con axios.					
TASK 3	 Implementar 	vicio en /services/apiService.js. un método que realice una petición GET a una API pública (ej. nonedas, noticias).				



Configurar cronjobs para envío automático de emails

TAS

- Configurar node-cron en el proyecto.
- Implementar una tarea que se ejecute cada día a una hora definida (ej. 8:00 AM).
- Usar nodemailer para enviar un correo con el resumen de los datos obtenidos desde la API externa.
- Colocar la lógica de envío en /services/emailService.js.

Manejo de errores y logs.

TAS

- Usar try/catch en llamadas axios y envío de emails.
- Mostrar mensajes amigables cuando ocurra un error.
- Agregar logs con console.log o una librería (ej. winston) para registrar las ejecuciones de cronjobs.

Pruebas básicas y validación.

TAS

- Probar el endpoint /api/data en Postman o el navegador.
- Verificar que el cronjob dispare el envío de emails correctamente.
- Validar que la aplicación sigue corriendo sin interrumpirse ante errores de red o fallos de correo

Criterios de aceptación.

- El servidor Express se levanta sin errores.
- El endpoint /api/data devuelve información real desde una API externa.
- El cronjob envía un correo automáticamente en el horario configurado.
- La aplicación maneja errores sin detenerse y muestra mensajes claros.
- La estructura está organizada en rutas y servicios, siguiendo buenas prácticas.

Story Points: 20



Al finalizar esta historia de usuario, serás capaz de construir una aplicación con Express que consume APIs externas y automatiza procesos con cronjobs de envío de emails. Esto te permitirá dominar integraciones en Node.js y sentar bases sólidas para proyectos más complejos que incluyan automatización, notificaciones y consumo de servicios externos.





P-J-H-B	В	F	
S1 - TASK	S1 - TASK	S1 – TASK	
S2 – SPIKE - RETORSPECTIVE	S2 - TASK	S2 – TASK	
S3 – HU - CALIFICABLE	S3 – SPIKE RETRO	S3 – SPIKE RETRO	
	S4 - TASK	S4 – TASK	
	S5 – HU - NOTA	S5 – SPIKE RETRO	
		S5 – TASK	
		S7 – HU - NOTA	

criterio	recuerda	comprende	practica	analiza	Evalua
Tarea 1	2	3	7	8	10
Tarea 2					