Logotipo

Descripción generada automáticamente

**Informe Final Desarrollo Proyecto APT**

**Fase 2**

**SRNexus**

Alumno: Elias Miguel Retamales Enrique

Docente: Viviana Soto

**CONTENIDO**

**Abstract**

1.1. Abstract en Español

1.2. Abstract en Inglés

**Conclusiones Individuales**

2.1. Conclusión

**Descripción del Proyecto**

3.1. Breve descripción y justificación de su relevancia

3.2. Relación con intereses profesionales

**Argumento sobre la factibilidad del proyecto.**

4.1. Factibilidad dentro de la asignatura

**Objetivos**

5.1. Objetivo general

5.2 Objetivos específicos

**Metodología de Trabajo**

6.1. Descripción de la metodología ágil

6.2. Propuesta metodológica en función de los requerimientos

6.3. Fases de la metodología ágil

**Plan de Trabajo y Sprints**

7.1 Sprints.

**Indicadores de Calidad**

8.1 Puntos de control

**Arquitectura de sistema**

9.1 Diagrama de sistema

**Estructura de datos relacionales**

10.1 Estructura DB

**Evidencias**

11.1. Adjunta.

**Plan de Trabajo del Equipo**

12.1. Recursos y herramientas utilizadas

**Abstract.**

1. **1.1. Abstract en español**
2. El proyecto consiste en una plataforma de integración IoT/DAQ que permitirá gestionar la información obtenida de sensores especializados (strain gage, PT100, acelerómetros, inclinómetros MEMS y magnetómetros, entre otros) conectados a dispositivos ESP32 y otras fuentes DAQ, almacenando estos datos en una base de datos de series temporales (InfluxDB). La plataforma, desarrollada en Laravel, contará con un ORM personalizado para manejar esta integración y ofrecerá tanto una API como una interfaz web para la visualización y análisis de los datos recopilados.
3. **1.2. Abstract en Inglés**
4. The project consists of an IoT/DAQ integration platform that will allow managing information obtained from specialized sensors (strain gage, PT100, accelerometers, MEMS inclinometers and magnetometers, among others) connected to ESP32 devices and other DAQ sources, storing this data in a time series database (InfluxDB). The platform, developed in Laravel, will feature a custom ORM to handle this integration and will offer both an API and a web interface for the visualization and analysis of the collected data.

**Conclusiones individuales**

**2.1. Conclusión**

Este proyecto es una nueva oportunidad para desarrollar una solución práctica para las problemáticas de manejo de información en el área de monitoreo de sistemas y automatización en la recopilación de datos. Es un reto práctico, fruto de varios años de estudios y pruebas que han llevado a la detección de esta necesidad en mi experiencia laboral de control de salud estructural.

**Descripción del Proyecto**

**3.1 Descripción breve del proyecto APT, justificando su relevancia.**

La plataforma permitirá gestionar información proveniente de sensores IoT especializados y otros dispositivos DAQ (dispositivos de adquisición de datos), facilitando el análisis en tiempo real y almacenando los datos de manera eficiente en una base de datos de series temporales. Esto es crucial en industrias que dependen del monitoreo estructural, como la construcción entre otras áreas de la ingeniería, ya que proporciona una herramienta integral para la toma de decisiones basada en datos confiables. Además, mejora la eficiencia de los procesos industriales al permitir la recopilación automatizada de datos en tiempo real.

**3.2 Relación del proyecto APT con tus intereses profesionales.**

Este proyecto refleja mis intereses profesionales en el desarrollo de sistemas embebidos, la integración de sensores IoT y la creación de plataformas web. La experiencia en el manejo de grandes volúmenes de datos en tiempo real a través de InfluxDB, y su visualización en una interfaz accesible, es clave para mi crecimiento en el campo de la ingeniería informática y electrónica, así como en la automatización de sistemas industriales.

**Argumento sobre la factibilidad del proyecto.**

**4.1. Factibilidad dentro de la asignatura:**

Debido a mis años de experiencia en el uso de herramientas digitales y electrónicas, tengo la capacidad de manejar eficientemente las tecnologías involucradas en el proyecto. Las herramientas como Laravel, InfluxDB, ESP32, y los módulos de comunicación ya son parte de mi dominio técnico, lo que garantiza que pueda cumplir con los plazos establecidos. Además, existe una gran cantidad de documentación asociada a estas herramientas, lo que facilita su integración. Aunque el proyecto en su totalidad no estará listo para ser un producto comercial al finalizar la asignatura, como desarrollo conceptual puede cumplir con las expectativas y demostrar su viabilidad técnica.

**Objetivos.**

**5.1. Objetivo general:**

Desarrollar una plataforma de integración que permita gestionar y visualizar los datos obtenidos de sensores IoT en tiempo real, utilizando una API en Laravel que se conecte de manera eficiente con una base de datos de series temporales (InfluxDB).

**5.2 Objetivos específicos:**

1. Implementar una API que gestione la recolección, almacenamiento y consulta de datos provenientes de sensores IoT.
2. Crear un ORM en Laravel que facilite la comunicación eficiente con la base de datos InfluxDB.
3. Desarrollar gráficos interactivos que permitan visualizar los registros de los sensores en tiempo real desde una interfaz web.
4. Probar la plataforma en diferentes condiciones para garantizar la estabilidad y rendimiento en el manejo de grandes volúmenes de datos.

**Metodología de trabajo.**

**6.1. Descripción de la metodología ágil:**

El proyecto seguirá una metodología ágil, permitiendo un desarrollo iterativo en sprints semanales. Cada sprint se enfocará en la entrega de componentes funcionales, comenzando con la configuración de sensores y finalizando con la implementación de la API y la interfaz web. Esta metodología garantiza flexibilidad en la planificación, lo que permite realizar ajustes en función de la retroalimentación y los requerimientos emergentes.

**6.2. Propuesta metodológica en función de los requerimientos:**

La metodología se adapta a las necesidades del proyecto al proporcionar entregas incrementales de cada componente crítico. Durante el desarrollo, se realizarán reuniones diarias (conmigo mismo…) para revisar el progreso y solucionar obstáculos rápidamente. Esto permitirá cumplir con los objetivos propuestos de manera eficiente, asegurando que se alcancen los resultados esperados en los tiempos establecidos.

**6.3. Fases de la metodología ágil:**

La metodología ágil se organiza en iteraciones semanales (sprints). Cada sprint se enfocará en una parte específica del proyecto, permitiendo una entrega continua de valor y ajustando el desarrollo conforme se reciba retroalimentación. Para la ejecución del proyecto se considerará la siguiente organización inicial:

**Sprint 1:** Configuración de la base de datos InfluxDB y el software/hardware para la recolección y almacenamiento de datos.  
**Sprint 2:** Creación del proyecto en Laravel y desarrollo del ORM para la integración con InfluxDB.  
**Sprint 3:** Desarrollo de la estructura de datos relacional para almacenar estadísticas derivadas de los datos de sensores.  
**Sprint 4:** Implementación del sistema de autenticación para la plataforma web y API.  
**Sprint 5:** Desarrollo de modelos, migraciones y controladores básicos para la gestión de datos.  
**Sprint 6:** Implementación de gráficos interactivos para la visualización de datos en tiempo real.  
**Sprint 7:** Optimización de la plataforma, pruebas de rendimiento y ajustes en la base de datos.  
**Sprint 8:** Pruebas finales de seguridad y estabilidad del sistema antes de la entrega.  
**Sprint 9:** Posibles solicitudes adicionales.  
**Sprint 10:** Cierre de documentación.  
**Sprint 11:** Puesta en marcha.

Estas fases permiten un desarrollo flexible y controlado, asegurando que cada componente del sistema esté completamente funcional antes de avanzar al siguiente. Las iteraciones posteriores llevan un enfoque ejecutivo y no relacionado directamente con el desarrollo de la solución.

**Plan de Trabajo y Sprints.**

**7.1 Sprints.**

**Sprint 1 (Semana 1 - 1.5)** Creación de la base de datos InfluxDB y configuración.

1. Configurar la base de datos InfluxDB.
2. Configuración del hardware de prueba (ESP32) para enviar datos a InfluxDB.
3. Realizar pruebas iniciales de comunicación entre los dispositivos IoT y la base de datos.

**Sprint 2 (Semana 1.5 - 2)** Creación del proyecto Laravel y programación del ORM para InfluxDB:

1. Crear el proyecto base en Laravel.
2. Desarrollar el ORM para gestionar la comunicación eficiente con InfluxDB.

**Sprint 3 (Semana 2 - 3)** Desarrollo de la estructura de datos relacional para almacenamiento de estadísticas:

1. Crear una base de datos relacional para datos sintetizados y estadísticas (vacía).

**Sprint 4 (Semana 3 - 3.5)** Creación de la autenticación para la plataforma web y API:

1. Configuración de roles y permisos de acceso.
2. Implementar la autenticación basada en tokens para la API.

**Sprint 5 (Semana 3.5 - 4.5)** Creación de modelos, migraciones, controladores y vistas básicas (CRUD):

1. Desarrollo de los modelos y migraciones para la base de datos relacional.
2. Crear los controladores y vistas CRUD para proyectos, sensores, y registros.

**Sprint 6 (Semana 4.5 - 5.5)** Desarrollo de vistas con gráficos:

1. Implementar gráficos interactivos para visualizar datos en tiempo real.

**Sprint 7 (Semana 5.5 - 6):**

1. Optimización de la plataforma y pruebas de rendimiento:
2. Mejorar la eficiencia en la consulta de datos entre Laravel e InfluxDB.

**Sprint 8 (Semana 6 - 6.5):**

1. Pruebas finales de seguridad y estabilidad:
2. Realizar pruebas de seguridad en la autenticación y acceso a la API.

**Sprint 9 (Semana 6.5 - 7) (Condicional):**

1. Integración de funcionalidades adicionales:
2. Revisar cualquier funcionalidad adicional solicitada o identificada durante el desarrollo.

**Sprint 10 (Semana 7.5 - 8) (Documentación):**

1. Revisión final y documentación completa:
2. Revisar todo el código y realizar la depuración final.

**Sprint 11 (Semana 8 - 8.5) (Presentación):**

1. Entrega final y retroalimentación:
2. Realizar los últimos ajustes y pruebas tras la retroalimentación recibida.

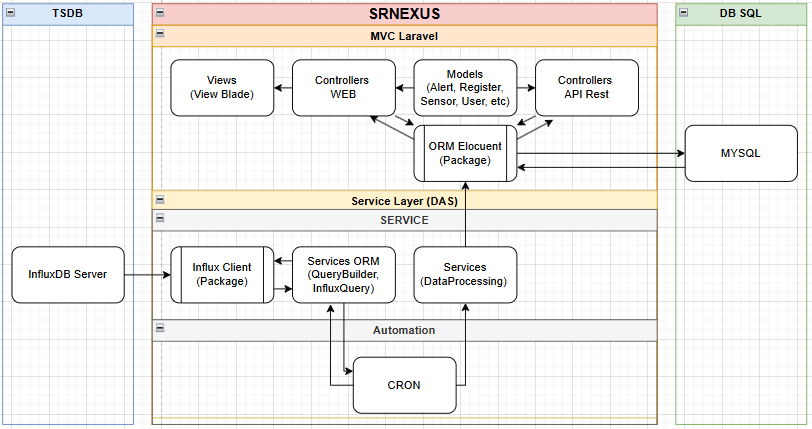
**Indicadores de Calidad**

**8.1 Puntos de control:**

1. Eficiencia en la integración de sensores IoT: La plataforma debe ser capaz de recibir y procesar datos de los sensores de manera continua y sin interrupciones.
2. Precisión en el almacenamiento y visualización de series temporales: Los datos deben ser almacenados de manera precisa, reflejando adecuadamente la información capturada por los sensores en tiempo real.
3. Interfaz web amigable y funcional: La visualización de los datos debe ser clara y accesible a través de una interfaz web intuitiva.

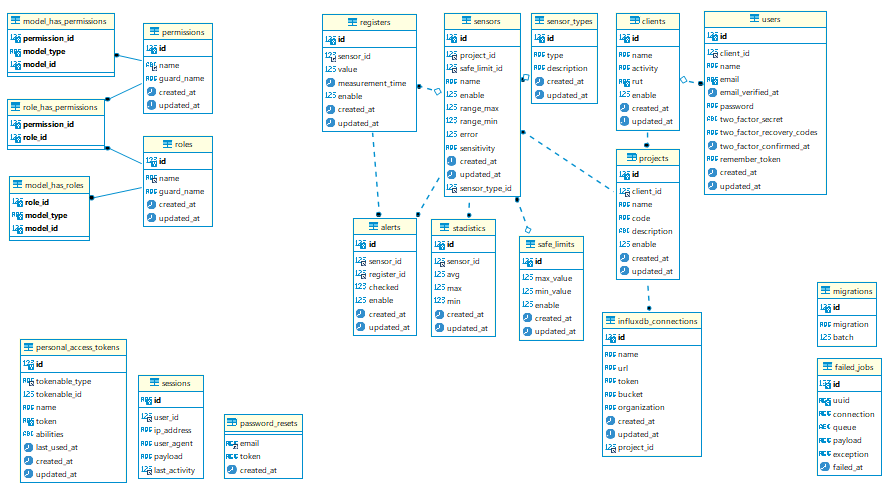
**9.1 Arquitectura del sistema:**

Estructura seleccionada en base al framework y los requerimientos de comunicación con los sistemas TSDB. Se ha determinado unas estructura tipo MVC+Service.



**10.1 Estructura de DB:**

La estructura de base de datos está definida de forma jerarquizada para enlazar elementos segun su posición en el proceso de adquisición de datos relacionada a cada proyecto por cada cliente. Se definen tablas tipo para el soporte de esta estructura con el fin de aportar mayor orden al proceso de detección de alertas. En conjunto se consideran las tablas creadas para el control de permisos y roles de usuario, y por último se conservan las tablas de control de laravel para funciones integradas como control de migraciones.

  
Puede observar el archivo en detalle bajo el nombre “Estructura DB”

**Evidencias (Adjuntas al informe)**

1. Configuración de InfluxDB (en script python) y su conexión con el hardware ESP32 (script C)
2. Capturas de pantalla de la API en funcionamiento, mostrando la integración con InfluxDB.
3. Pruebas de visualización de datos en la interfaz web, incluyendo gráficos en tiempo real.
4. Base de datos relacional

**Plan de Trabajo del Equipo.**

**10.1. Recursos y herramientas utilizadas:**

**FrontEnd:**

1. Adminlte3
2. JS/JQuery

**BackEnd:**

1. Laravel Framework
2. Composer
3. MySQL
4. InfluxDB

**IoT/QA TSDB:**

1. EPS32
2. Arduino IDE

**QA:**

1. Xdebug
2. Postman

**Control de Versiones:**

1. Github