

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**INGENIERÍA DEL SOFTWARE**

**Ejercicio 1 - Diseño de una Arquitectura Software**

**Diseño y Arquitectura del Software**

**Integrantes Portavoz**

SAMUEL RUSU [s.rusu.2019@alumnos.urjc.es](mailto:s.rusu.2019@alumnos.urjc.es)

MARÍA ESTEBAN SÁNCHEZ

SERGIO VILLAGARCÍA SÁNCHEZ

JESÚS ORTIZ LOP

CARLOS HERNÁNDEZ HERNÁNDEZ

MARIO RECIO MONTERO

**ÍNDICE**

**1.ROLES**

**ASS:**

**-** SAMUEL RUSU

**-** SERGIO VILLAGARCÍA SÁNCHEZ

**ASC:**

**-** CARLOS HERNÁNDEZ HERNÁNDEZ

**-** MARÍA ESTEBAN SÁNCHEZ

**ASJ:**

**-** MARIO RECIO MONTERO

**-** JESÚS ORTIZ LOPO

**2.ADMENTOR Y REQUISITOS FUNCIONALES**

|  |  |  |
| --- | --- | --- |
|  | **Nombre** | **Descripción** |
| **RF1** | **Centro de notificaciones** | Centro de notificaciones para el recibo de datos de los sensores y visualización de las analíticas. Desde este módulo se pueden gestionar todas las funcionalidades del software. |
| **RF2** | **Almacenamiento de inventario** | Añadir una base de datos SQL, que almacenará tanto las órdenes de trabajo, como el inventario de todo el material existente. |
| **RF2.1** | **Almacenamiento de datos de sensores** | Añadir una base de datos SQL, que almacenará todos los datos de los sensores |
| **RF3** | **Sistema de mensajería** | los operarios de la factoría 4.0 debe estar permanente notificados a través de un sistema de mensajería interno. |
| **RF3.1** | **Suscripción de los operadores** | Poderse suscribir a diferentes eventos y notificaciones como actualizaciones de la producción, fallos en los sensores o sobrecarga en la producción. |
| **RF4** | **Módulo de ordenes de trabajo** | Incluir un módulo de asignación de órdenes para operarios y máquinas que van a fabricar cada componente. |
| **RF5** | **Módulo de selección de algoritmos** | Dependiendo del momento de la producción elige el algoritmo que le corresponde. |
| **RF5.1** | **Algoritmo de optimización de volumen de trabajo** | Ya que se enviarán múltiples ordenes de trabajo, se requiere implementar un algoritmo que gestione el volumen y la gestión de dichas órdenes. |
| **RF5.2** | **Algoritmo de predicción de fallo** | Ya que es posible que se produzcan incidencias en las líneas de trabajo, es necesario incluir un algoritmo para detectarlos, y asignar los recursos necesarios desde otras líneas. |
| **RF6** | **Componente visual** | Incluir un componente de visualización para mostrar los datos en tiempo real del proceso productivo y las órdenes de trabajo. |
| **RF7** | **Medidas de seguridad** | Se requiere implementar medidas de seguridad para gestionar el acceso de los usuarios con el software. |
| **RF7.1** | **Seguridad en los mensajes** | A la hora de mandar y recibir mensajes, se deberá tener en cuenta que sea un sistema fiable, utilizando alternativas como Apache Kafka o MQTT. |
| **RF7.2** | **Límite de intentos de conexión** | Si el número de intentos supera los permitidos, se deberá suspender el intento de acceso al software y se considerará al dispositivo como fuera de servicio. |
| **RF8** | **Implementación de 3 familias de sensores** | Ya que los sensores IoT se clasifican en tres familias, cada una con ciertas funcionalidades características, se debe dar soporte a cada una de estas variantes. |

COSAS DE CLASE:

**LOS EVENTOS. Construir un sistema distribuido que recoja eventos mediante sensores y los notifique a alguien (un usuario, una base de datos) de forma distribuida.**

**Hay que capturar requisitos por jerarquía. Se puede meter un requisito dentro de otro si lo engloba (bajo o alto nivel).**

**3.RESULTADOS DE LAS TAREAS**

-Incluir resultados intermedios

**3.DECISIONES TOMADAS Y ARQUITECTURAS RESULTANTES**

**Iteración 1:**

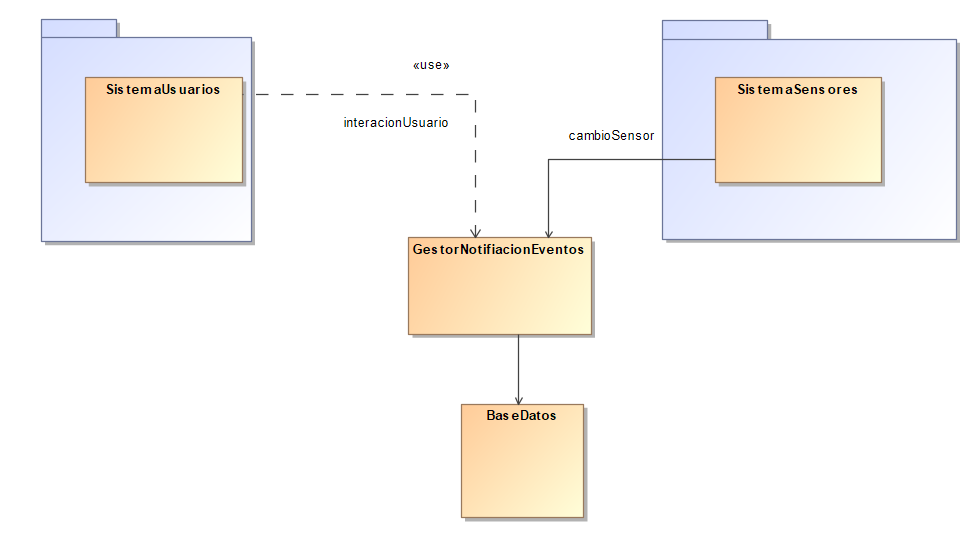
-Decisión 1: Se necesita un sistema de gestión y envío de notificaciones, que reaccione a los sensores, enviando las señales pertinentes.

**Solución:** Arquitectura con emisores de eventos, consumidores y canales para transmitir los eventos, donde los consumidores son responsables de reaccionar a los eventos.

-Decisión 2: Se requiere una pasarela a través de la cual los operarios que quieran consumir ciertos eventos o notificaciones se den de alta en el sistema de notificaciones.

**Solución:** Arquitectura cliente servidor para soportar peticiones y poder persistir los cambios o actualizaciones de datos en el sistema.

**Arquitectura resultante:**



**Iteración 2:**

-Decisión 1: Es necesaria una base de datos SQL para gestionar todo el espacio de almacenamiento de los datos que producen los sensores.

**Solución:** Se utilizará el patrón Singleton para asegurar que la base de datos tenga una única instancia.

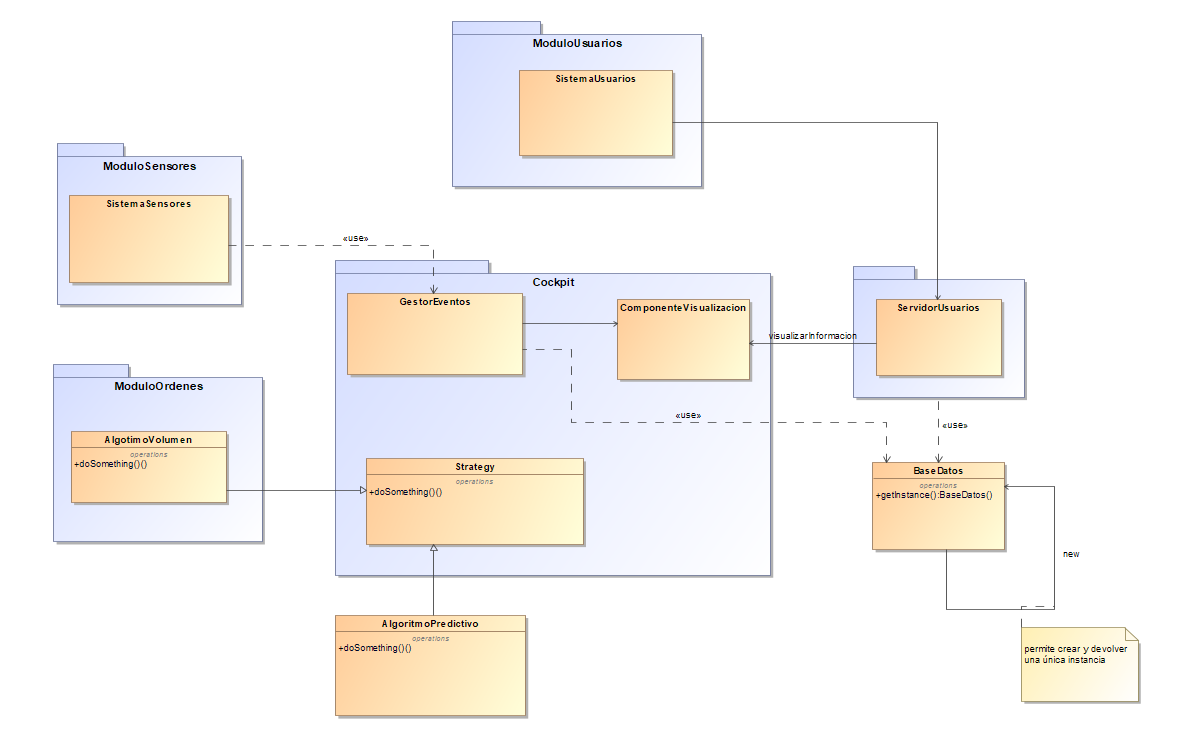
-Decisión 2: Se requiere un módulo de órdenes de trabajo para gestionar toda la operativa de cada trabajador y máquina.

**Solución:** Dentro de los usuarios que consumen las notificaciones enviadas por el sistema interno habrá un apartado específico para gestionar las necesidades de cada operador.

-Decisión 3: Se requiere un patrón para seleccionar el algoritmo inteligente predictivo óptimo en cada situación entre los dos que dispone nuestro software: uno para optimizar el volumen de órdenes de trabajo y otro para predecir el fallo una línea de trabajo y asignar recursos de otras líneas.

**Solución:** El patrón Strategy permitirá cambiar el algoritmo seleccionado dependiendo de las necesidades de nuestro sistema.

**Arquitectura resultante:**



**Iteración 3:**

-Decisión 1:

**Solución:**

-Decisión 2:

**Solución:**

**Arquitectura resultante:**

**Iteración 4:**

-Decisión 1:

**Solución:**

-Decisión 2:

**Solución:**

**Arquitectura resultante:**

**4.CONCLUSIONES**

-Problemas encontrados

-Incluir si alguna decisión ha sido muy discutida etc.

**5.BIBLIOGRAFÍA**

**6.TABLA DE TIEMPOS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Iteration | Time in ADD (ASS) | Reflection time (ASS-ASC) | Time in refined ADD (ASS) | Design ADD time (ASJ) |
| **1** | **1** | 120’ | 90’ | 30’ | 30’ |
|  | **1.1** | 60’ | 45’ | 20’ | 20’ |
| **2** | **2** | 70’ | 60’ | 20’ |  |
| **3** | **3** |  |  |  |  |
|  | **3.1** |  |  |  |  |
|  | **3.2** |  |  |  |  |
|  | **3.3** |  |  |  |  |
| **4** | **4** |  |  |  |  |
|  | **4.1** |  |  |  |  |
|  | **4.2** |  |  |  |  |
|  | **4.3** |  |  |  |  |
| **5** | **5** |  |  |  |  |
|  | **5.1** |  |  |  |  |
|  | **5.2** |  |  |  |  |
|  | **5.3** |  |  |  |  |

*Tabla 1: Tabla de tiempos*