



UCA

Universidad
de Cádiz

GRADO EN INGENIERÍA INFORMÁTICA

INGENIERÍA DE SISTEMAS DE LA INFORMACIÓN

Hito 1

Autores:

Alejandro Cuevas Casado
Juan Mariano Centeno Ariza

Fecha: 18/10/2023

Edificio Inteligente

Información y dominio

Nuestro proyecto se centra en la automatización y la gestión eficiente de un edificio de oficinas. El objetivo principal es mejorar la comodidad de los empleados, optimizar la utilización de recursos y aumentar la eficiencia operativa del edificio.

Componentes claves:

1. Sensores y dispositivos IoT: Estos dispositivos incluyen sensores de temperatura, humedad, luz, ocupación, medidores de energía, sistemas de seguridad, sistemas de acceso y más. Recopilan datos en tiempo real sobre el entorno y la actividad en el edificio.
2. Red de comunicación: Se necesita una red de comunicación confiable para conectar todos los sensores y dispositivos IoT al sistema central de gestión.
3. Sistema de Procesamiento de Eventos Complejos: El CEP analiza los datos generados por los sensores y dispositivos en tiempo real. Detecta patrones, correlaciones y eventos significativos, y toma decisiones basadas en estos datos.
4. Sistemas de control y automatización: Estos sistemas, controlados por el CEP, permiten realizar ajustes automáticos en el edificio, como la climatización, la iluminación, la seguridad y el acceso.

Motivación y necesidad:

1. Eficiencia energética: La necesidad de reducir el consumo de energía y los costos asociados es un factor importante en la motivación de los edificios inteligentes. La monitorización en tiempo real de la ocupación y las condiciones ambientales permite la optimización de sistemas como la climatización y la iluminación, lo que resulta en un menor gasto energético.
2. Comodidad y productividad de los empleados: La comodidad y la satisfacción de los empleados es fundamental. La capacidad de ajustar automáticamente la iluminación, la temperatura y otros servicios en función de las preferencias y las necesidades de las personas mejora la calidad de vida y la productividad en el edificio.

3. Seguridad y gestión de riesgos: Los sistemas de seguridad y acceso permiten una respuesta más rápida a situaciones de emergencia, como intrusiones o incendios. La detección en tiempo real de eventos sospechosos o peligrosos es esencial para garantizar la seguridad de los empleados.

4. Reducción de costos operativos: La automatización y la toma de decisiones basadas en datos ayudan a reducir los costos operativos. Los edificios inteligentes pueden programar el mantenimiento preventivo, ahorrar energía y optimizar el uso de personal de mantenimiento.

5. Sostenibilidad y cumplimiento de regulaciones: Los edificios inteligentes pueden contribuir a la sostenibilidad al reducir el desperdicio de recursos y cumplir con regulaciones y estándares ambientales.

En resumen, el escenario de un edificio inteligente con IoT y CEP responde a la necesidad de mejorar la eficiencia operativa, la comodidad de los empleados y la seguridad, al tiempo que se reduce el consumo de energía y los costos operativos. Esta combinación de tecnologías permite una gestión más inteligente de los edificios, lo que beneficia tanto a los propietarios como a los empleados.

Esquemas

Puerta

```
CREATE SCHEMA PuertaEvent(  
    cod_empleado INT,  
    Nombre STRING,  
    Apellidos STRING,  
    Fecha STRING,  
    Tipo STRING  
);
```

Habitación

```
CREATE SCHEMA HabitacionEvent(  
    cod_sala INT,  
    temperatura DOUBLE,  
    humedad DOUBLE,  
    luz INT,  
    num_empleados INT,  
    fecha_ult_mant STRING,  
    fecha_sistema STRING  
);
```

Producto

```
CREATE SCHEMA ProductEvent (  
  cod_producto INT,  
  ubicacion STRING,  
  stock_actual INT,  
  stock_minimo INT  
);
```

Odenador

```
CREATE SCHEMA OrdenadorEvent(  
  cod_ordenador INT,  
  cod_sala INT,  
  status STRING  
);
```

Patrones

- ***Patrón que detecta entrada y salida en el edificio***

```
SELECT * FROM PuertaEvent  
WHERE Tipo = 'Entrada';
```

```
SELECT * FROM PuertaEvent  
WHERE Tipo = 'Salida';
```

- ***Patrón para detectar cambios bruscos de temperatura***

```
SELECT * FROM HabitacionEvent  
WHERE temperatura > (umbral);
```

- ***Patrón para detectar bajos niveles de stock***

```
SELECT * FROM ProductoEvent  
WHERE stock_actual < stock_minimo;
```

- ***Patrón para detectar ordenadores que necesitan mantenimiento***

```
SELECT * FROM OrdenadorEvent  
WHERE cod_sala = (cod_sala) AND status = 'Avería';
```

- ***Patrón para detectar la temperatura y humedad media en habitaciones con intervalo de 10 minutos***

```
SELECT cod_sala, AVG(temperatura), AVG(humedad) as temperatura_promedio
FROM HabitacionEvent.win:time(10 minutes)
GROUP BY cod_sala;
```

- ***Patrón con ventana deslizante para detectar aumento o disminución de empleados en una habitación***

```
SELECT cod_sala, COUNT(*) as numIngresos
FROM PuertaEvent[Tipo = 'Entrada'].win:length(5)
GROUP BY cod_sala;
```

Habitación

- ***Patrón selección todo***

```
@Name('Patron1')
SELECT * FROM HabitacionEvent WHERE temperatura > 20;
```

- ***Patrón dependiente de la bajada de temperatura de una sala***

```
@Name('PatronAlertaBajadaTemp')
insert into alerta
SELECT h1.cod_sala as cod_sala, h1.num_empleados as emp1, h2.num_empleados
as emp2, h2.temperatura as temp, 'Bajada' as tipo
FROM PATTERN [every h1=HabitacionEvent ->
h2=HabitacionEvent((h2.num_empleados - h1.num_empleados) >= 10 AND
h1.cod_sala = h2.cod_sala) where timer:within (5 minutes)];
```

- ***Patrón dependiente de la subida de temperatura de una sala***

```
@Name('PatronAlertaSubidaTemp')
insert into alerta
SELECT h1.cod_sala as cod_sala, h1.num_empleados as emp1, h2.num_empleados
as emp2, h2.temperatura as temp, 'Subida' as tipo
FROM PATTERN [every h1=HabitacionEvent ->
h2=HabitacionEvent((h2.num_empleados - h1.num_empleados) <= 10 AND
h1.cod_sala = h2.cod_sala) where timer:within (5 minutes)];
```

- ***Patrón selección de la tabla alertas***

```
@Name('Patron Alerta Temperatura')
SELECT cod_sala, temp, tipo FROM alerta;
```

Producto

- ***Patrón dependiente de la diferencia de stock de un producto***

@Name('PatronAlertaProducto')

insert into alertaProducto

SELECT p1.cod_producto as cod_producto, p1.ubicacion as ubicacion,
p1.stock_actual as actual, p1.stock_minimo as minimo
FROM PATTERN [every p1=ProductoEvent(p1.stock_actual < p1.stock_minimo)];

- ***Patrón selección de la tabla de alerta***

@Name('Patron Alerta Stock')

SELECT * FROM alertaProducto;

Ordenador

- ***Patrón selección de la tabla de eventos***

@Name('Patron1')

SELECT * FROM OrdenadorEvent

WHERE status = 'Averia';

- ***Patrón dependiente de avería***

@Name('PatronAlertaOrdenador')

insert into alertaOrdenador

SELECT o1.cod_ordenador as cod_ordenador, o1.cod_sala as cod_sala, o1.status as
status
FROM PATTERN [every o1=OrdenadorEvent(o1.status = 'Averia')];

- ***Patrón selección de la tabla de alerta***

@Name('Patron Alerta Averia')

SELECT * FROM alertaOrdenador;

Puerta

- ***Patrón selección de la tabla de eventos entrada***

@Name('PatronEntrada')

SELECT * FROM PuertaEvent

WHERE Tipo = 'Entrada';

- ***Patrón selección de la tabla de eventos salida***

@Name('PatronSalida')

SELECT * FROM PuertaEvent

WHERE Tipo = 'Salida';

- ***Patrón dependiente alerta de entrada o salida***

@Name('PatronAlertaEntradaSalida')

insert into alertaPuerta

SELECT pu1.cod_empleado as empleado, pu1.Nombre as nombre, pu1.Apellidos as
apellidos, pu1.Fecha as fecha, pu1.Tipo as tipo

FROM PATTERN [every pu1=PuertaEvent];

- ***Patrón selección de la tabla de alerta***

@Name('Patron Alerta Puerta')

SELECT * FROM alertaPuerta;