

Intelligent City Vision

Héctor Julián cruz navas

Unidades Tecnológicas de Santander

Planeación de Sistemas informáticos

PEDRO ANTONIO RAMIREZ GARCIA

02/10/2024

**Misión:**

Desarrollar un sistema predictivo inteligente que optimice la gestión de recursos en ciudades inteligentes, utilizando inteligencia artificial, machine learning y big data para anticipar la demanda de servicios clave (energía, agua, transporte, seguridad), mejorando la sostenibilidad, eficiencia y calidad de vida de los ciudadanos.

Visión:

Ser una solución líder en la gestión eficiente de recursos en Smart Cities, permitiendo que con el paso del tiempo las ciudades de todo el mundo se vuelvan más sostenibles, proactivas y resilientes, mediante la integración de tecnologías avanzadas y la toma de decisiones basada en datos.

En este caso se ha tomado la decisión de usar a Santander, España, por los siguientes motivos:

Tamaño adecuado: Es una ciudad relativamente pequeña, con una población de alrededor de 172,000 habitantes, lo que facilita la implementación inicial de un sistema como el que propones sin los desafíos logísticos de una gran metrópoli.

Potencial de éxito: Santander ya ha demostrado interés en la tecnología, y ha sido pionera en la implementación de proyectos de Smart City a pequeña escala, como sensores para monitorear el tráfico y el uso de energía, lo que la convierte en una excelente candidata para expandir un sistema predictivo más avanzado.

Infraestructura disponible: La ciudad cuenta con una infraestructura tecnológica básica y programas municipales orientados hacia la sostenibilidad y la innovación, por lo que es más probable que un proyecto de este tipo sea bien recibido y apoyado.

Santander tiene el tamaño perfecto y el ecosistema adecuado para llevar a cabo este proyecto de optimización de recursos con un enfoque en tecnologías emergentes.

Funcionalidades del Sistema Predictivo Inteligente para Smart Cities

1. Predicción de Demanda Energética:

El sistema analizará el consumo energético en tiempo real y predecirá los picos de demanda, permitiendo ajustar el suministro para evitar sobrecargas o apagones.

2. Optimización del Tráfico:

A través de sensores IoT y análisis de datos históricos, el sistema preverá embotellamientos y ajustará los semáforos o sugerirá rutas alternativas a los ciudadanos.

3. Gestión Eficiente de Recursos Hídricos:

El sistema predecirá el uso del agua basándose en factores como el clima, el horario y el comportamiento histórico, permitiendo gestionar el suministro de manera más eficiente.

4. Seguridad Proactiva:

Monitoreo de cámaras y sensores para detectar patrones sospechosos o situaciones de riesgo antes de que se conviertan en incidentes graves, enviando alertas a los equipos de seguridad.

5. Gestión de Alertas en Tiempo Real:

Los gestores de la ciudad recibirán alertas inmediatas ante cualquier desviación significativa en los patrones de consumo, tráfico o seguridad, lo que les permitirá tomar acciones preventivas.

6. Interfaz de Monitoreo Centralizado:

Una plataforma intuitiva para que los operadores puedan visualizar el estado de los recursos en tiempo real, revisar predicciones, ajustar configuraciones y monitorear la ciudad de forma integral.

Riesgos del Sistema y Cómo Gestionarlos:

1. Fallas en la Seguridad de los Datos (Ciberataques):

Riesgo: El sistema manejará datos sensibles de ciudadanos y servicios críticos, lo que lo hace vulnerable a ataques cibernéticos.

Manejo: Implementar medidas de ciberseguridad robustas, como cifrado de datos, autenticación multifactor, y monitoreo constante de las redes. Realizar auditorías de seguridad y simulacros de ataque (penetration tests) de forma periódica para mejorar la protección.

2. Errores en Predicciones:

Riesgo: Los modelos de machine learning no son infalibles, lo que puede llevar a predicciones incorrectas o decisiones subóptimas, como un mal ajuste en el suministro de recursos.

Manejo: Utilizar un enfoque de mejora continua para los modelos de IA, alimentándolos con nuevos datos constantemente y ajustándolos según sea necesario. Además, se debe implementar un sistema de revisión manual de las predicciones más críticas para evitar grandes fallos.

3. Dependencia de la Infraestructura Tecnológica:

Riesgo: Si los sensores IoT, las redes o los servidores fallan, el sistema podría quedar inoperativo, afectando los servicios críticos de la ciudad.

Manejo: Implementar infraestructura redundante (backup de servidores y redes) para que el sistema pueda seguir funcionando incluso si un componente falla. Utilizar redes de emergencia y sistemas de respaldo locales para asegurar la continuidad del servicio.

4. Problemas de Privacidad de los Ciudadanos:

Riesgo: La recolección de datos a gran escala (tráfico, seguridad, consumo) puede generar preocupación por la privacidad de los ciudadanos.

Manejo: Asegurarse de que todos los datos personales se mantengan anonimizados y que el sistema cumpla con las regulaciones de protección de datos, como el Reglamento General de Protección de Datos (GDPR) en Europa. Comunicar claramente a los ciudadanos el uso de sus datos y ofrecer transparencia en el proceso.

5. Altos Costos de Implementación y Mantenimiento:

Riesgo: La implementación inicial puede ser costosa, y si no se gestiona adecuadamente, el mantenimiento a largo plazo también puede ser un desafío.

Manejo: Desarrollar un plan de implementación por fases, comenzando con un piloto en áreas específicas de la ciudad para evaluar la efectividad del sistema antes de expandirlo. Buscar financiación pública y privada, y establecer alianzas estratégicas con empresas de tecnología para compartir costos.

6. Aceptación Social y Resistencia al Cambio:

Riesgo: Los ciudadanos y funcionarios pueden ser reacios a adoptar nuevas tecnologías, lo que ralentizaría la implementación del sistema.

Manejo: Crear campañas de concientización pública para educar a los ciudadanos sobre los beneficios del sistema. Involucrar a la comunidad en la planificación y hacer que el sistema sea accesible y transparente, permitiendo una adopción más fluida.

Requerimientos Funcionales

Los requerimientos funcionales describen lo que el sistema debe hacer, las funcionalidades principales que el sistema proporcionará a los usuarios y cómo interactuará con otros sistemas.

1. Predicción de Uso de Recursos (Energía, Agua, Transporte):

El sistema debe prever la demanda de recursos como energía, agua y transporte en Purityville. Usando sensores IoT, el sistema recopilará datos en tiempo real y ejecutará algoritmos predictivos que ajusten el suministro en función de la demanda proyectada.

Funcionalidad: Capturar datos en tiempo real desde sensores IoT, procesarlos a través de modelos de machine learning y ajustar el suministro de recursos.

2. Sistema de Gestión de Tráfico Inteligente:

El sistema debe poder monitorear y ajustar el flujo de tráfico en Purityville. A partir de los datos de tráfico en tiempo real, el sistema ajustará semáforos y sugerirá rutas alternativas a los ciudadanos para evitar congestiones.

Funcionalidad: Monitorear el tráfico, ajustar semáforos y sugerir rutas alternativas.

3. Sistema de Seguridad Proactiva:

El sistema debe poder predecir y monitorear riesgos de seguridad, como incidentes o comportamientos inusuales en ciertas áreas. Basado en patrones históricos y datos en tiempo real, enviará alertas automáticas a las autoridades si detecta algo sospechoso.

Funcionalidad: Monitorear cámaras de seguridad y sensores para detectar patrones de riesgo y emitir alertas automáticas.

4. Optimización del Consumo de Agua:

Utilizando datos de consumo de agua y previsiones meteorológicas, el sistema debe prever el uso del agua en Purityville y ajustar la distribución en función de la demanda.

Funcionalidad: Predecir la demanda de agua y ajustar la distribución en consecuencia.

5. Interfaz de Monitoreo Centralizado:

El sistema debe ofrecer a los gestores de Purityville una plataforma para visualizar todos los datos en tiempo real y acceder a predicciones, alertas y estadísticas clave sobre el uso de los recursos y la seguridad.

Funcionalidad: Proveer una plataforma de monitoreo donde los usuarios puedan visualizar gráficos, datos y alertas en tiempo real.

6. Gestión de Alertas y Respuesta:

El sistema debe emitir alertas automáticas a los operadores cuando los recursos excedan un umbral crítico o cuando se prevean problemas inminentes (como un pico de demanda de energía o tráfico). Las alertas deben ser visibles en la interfaz y enviarse por otros medios (SMS, correo electrónico, etc.).

Funcionalidad: Generar y enviar alertas automáticas ante situaciones críticas.

Requerimientos No Funcionales:

Los requerimientos no funcionales se refieren a los criterios de calidad y restricciones del sistema, garantizando que funcione correctamente bajo todas las condiciones.

1. Escalabilidad:

El sistema debe ser capaz de gestionar un aumento en el número de sensores IoT y volúmenes de datos a medida que Purityville crece. Debe soportar tanto un aumento en la población como en el tráfico de datos.

Criterio: El sistema debe escalar horizontalmente, añadiendo más servidores o procesadores en la nube si la carga de trabajo aumenta.

2. Seguridad de los Datos:

El sistema debe garantizar la seguridad de los datos capturados por los sensores IoT, protegiendo la privacidad de los ciudadanos de Purityville. Los datos deben ser encriptados y acceder a ellos solo debe ser permitido a usuarios autorizados.

Criterio: Implementar cifrado de extremo a extremo y autenticación multifactor para el acceso a los datos.

3. Fiabilidad y Disponibilidad:

El sistema debe estar disponible y operativo el 99.9% del tiempo, asegurando que no haya interrupciones en los servicios críticos como la predicción de energía y el monitoreo de seguridad.

Criterio: Proveer servidores redundantes y una arquitectura de alta disponibilidad (HA) para minimizar el tiempo de inactividad.

4. Latencia Baja:

Las predicciones y ajustes en el suministro de recursos deben realizarse en tiempo real o con una latencia mínima para asegurar que el sistema pueda reaccionar a cambios imprevistos rápidamente.

Criterio: El sistema debe procesar y enviar alertas en menos de 5 segundos tras la detección de un evento crítico.

5. Compatibilidad:

El sistema debe ser compatible con los dispositivos IoT existentes y futuros, permitiendo la integración de nuevos tipos de sensores sin requerir grandes modificaciones.

Criterio: Uso de protocolos estándares como MQTT, HTTP, o LoRaWAN para la comunicación con dispositivos IoT.

6. Usabilidad:

La interfaz de usuario debe ser intuitiva y fácil de utilizar para los gestores de la ciudad, con un diseño que permita acceder rápidamente a los datos importantes y emitir decisiones basadas en las predicciones.

Criterio: La interfaz debe cumplir con normas de usabilidad como ISO 9241 para asegurar que sea accesible y eficiente para los usuarios.

7. Mantenimiento y Actualización:

El sistema debe permitir la implementación de parches y actualizaciones sin interrumpir el funcionamiento normal. Además, debe ser fácil de mantener y modificar.

Criterio: Actualizaciones automáticas o programadas, con mantenimiento fácil y documentación clara.

8. Rendimiento:

El sistema debe ser capaz de manejar grandes volúmenes de datos sin degradación del rendimiento, y debe procesar las predicciones y ajustes de recursos en tiempo real.

Criterio: El sistema debe ser capaz de procesar al menos 10,000 eventos de sensores por segundo sin afectar el rendimiento.

Diseño del sistema:

1. Arquitectura General del Sistema

La arquitectura del sistema se puede dividir en varias capas o componentes:

1-Sensores y Dispositivos IoT

Sensores de Energía: Monitorean el consumo de energía en tiempo real.

Sensores de Tráfico: Detectan el flujo vehicular y peatonal (cámaras, sensores de movimiento).

Sensores de Agua: Miden el uso y la calidad del agua en diferentes puntos.

Cámaras de Seguridad: Proporcionan vigilancia y monitoreo de áreas clave.

2-Red de Comunicación:

Redes Inalámbricas: 5G, LoRaWAN o Wi-Fi para la transmisión de datos desde los sensores al sistema central.

Pasarelas IoT: Dispositivos que agrupan datos de varios sensores y los envían al servidor central.

3-Procesamiento de Datos:

Servidor Central (Cloud): Recibe, almacena y procesa los datos provenientes de los sensores. Este servidor puede estar en la nube (AWS, Google Cloud) para asegurar escalabilidad y disponibilidad.

Módulo de Machine Learning: Procesa los datos para realizar predicciones sobre consumo de recursos, tráfico y seguridad, usando algoritmos de IA.

4-Interfaz de Usuario

Aplicación Web/ Móvil: Interfaz que permite a los gestores de la ciudad y a los ciudadanos acceder a la información en tiempo real, visualizar datos históricos, recibir alertas y gestionar recursos.

Dashboards: Visualizaciones que muestran el estado actual de la ciudad, incluyendo gráficos y estadísticas sobre el uso de recursos y la seguridad.

5- Gestión de Alertas y Respuesta

Módulo de Alertas: Genera alertas automáticas en caso de eventos críticos (picos de consumo, congestiones, incidentes de seguridad) y las envía a los responsables.

Sistema de Respuesta: Facilita la toma de decisiones rápidas y efectivas por parte de los gestores de la ciudad.

2. Flujo de Datos del Sistema

El flujo de datos en el sistema sigue este proceso:

Captura de Datos: Los sensores recolectan información en tiempo real sobre el tráfico, el consumo de energía, el uso del agua, etc.

Transmisión de Datos: Los datos son enviados a través de la red de comunicación a la pasarela IoT, que los transmite al servidor central.

Procesamiento de Datos: El servidor central almacena y analiza los datos. Los modelos de machine learning procesan la información para prever patrones futuros.

Generación de Reportes: Se generan reportes y visualizaciones para que los gestores de la ciudad puedan ver el estado actual y las predicciones.

Alertas y Respuesta: Si se detecta un evento crítico, el sistema emite alertas y proporciona recomendaciones para la respuesta, que son enviadas a los operadores a través de la interfaz de usuario.

3. Consideraciones Técnicas

Tecnologías Utilizadas:

Lenguajes de Programación: Python (para backend y machine learning), JavaScript (para frontend), HTML/CSS (para diseño web).

Bases de Datos: MongoDB o PostgreSQL para almacenar datos.

Frameworks de Machine Learning: TensorFlow, Keras o Scikit-learn para construir modelos predictivos.

Plataformas en la Nube: AWS, Google Cloud o Azure para el procesamiento y almacenamiento de datos.

4. Seguridad y Protección de Datos

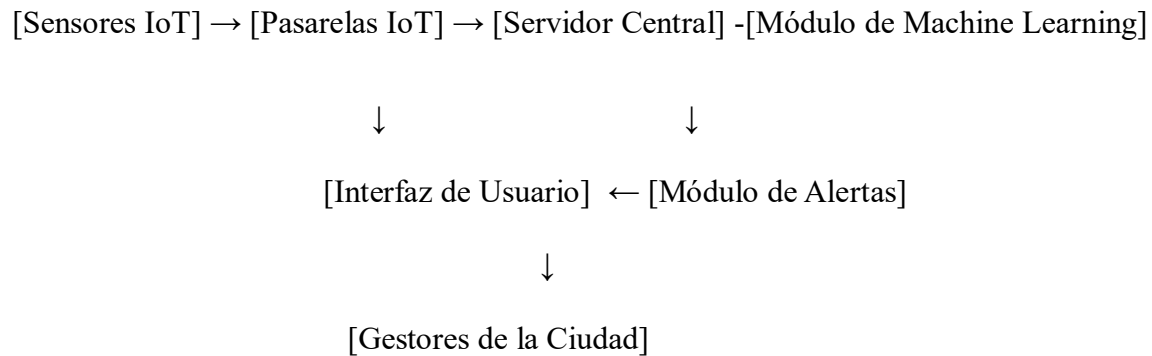
Cifrado: Implementar cifrado de datos en tránsito y en reposo para proteger la información sensible de los ciudadanos.

Autenticación: Usar autenticación multifactor y control de acceso basado en roles para la interfaz de usuario.

Auditoría y Monitoreo: Mantener un registro de acceso y monitorear el sistema para detectar y prevenir accesos no autorizados.

5. Diagrama de Flujo del Sistema

A continuación, se puede representar un diagrama de flujo simplificado que ilustre cómo interactúan los diferentes componentes del sistema.



1. Sensores y Dispositivos IoT (Internet de las Cosas)

Estos dispositivos recolectan datos en tiempo real sobre diferentes aspectos de la ciudad.

- **Sensores de Energía:** Miden el consumo de electricidad en áreas específicas de la ciudad.
- **Sensores de Tráfico:** Detectan el flujo de vehículos y peatones mediante cámaras o detectores de movimiento.
- **Sensores de Agua:** Monitorean el consumo y la calidad del agua, proporcionando información sobre su uso en tiempo real.

- **Cámaras de Seguridad:** Proveen datos visuales para monitorear incidentes y detectar comportamientos inusuales en áreas específicas.

2. Red de Comunicación

Permite la transmisión de datos desde los sensores IoT hacia el sistema central.

- **Redes Inalámbricas:** Uso de tecnologías como **5G**, **LoRaWAN** o **Wi-Fi** para garantizar una comunicación rápida y eficiente entre los dispositivos IoT y los servidores.

- **Pasarelas IoT:** Dispositivos intermedios que agrupan los datos de varios sensores y los transmiten al servidor central de forma eficiente.

3. Servidor Central y Almacenamiento

La infraestructura central que recibe, almacena y procesa los datos recopilados por los sensores.

- **Servidor en la Nube:** Se utiliza una plataforma en la nube (AWS, Google Cloud, Azure) para procesar y almacenar grandes volúmenes de datos en tiempo real.
- **Bases de Datos:** Las bases de datos NoSQL (por ejemplo, **MongoDB**) permiten almacenar grandes cantidades de datos no estructurados, como los capturados por los sensores IoT.

4. Módulo de Machine Learning e Inteligencia Artificial

Este componente es responsable de analizar los datos en tiempo real y generar predicciones.

- **Algoritmos de Machine Learning:** Procesan los datos y hacen predicciones sobre la demanda de energía, tráfico, uso de agua y seguridad. Por ejemplo, predecir congestiones o picos de consumo energético.
- **Entrenamiento de Modelos Predictivos:** A medida que el sistema recolecta más datos, los modelos se actualizan para mejorar la precisión de las predicciones.

5. Sistema de Gestión de Alertas

Este módulo se encarga de generar y distribuir alertas automáticas a los gestores de la ciudad cuando se detectan anomalías o eventos críticos.

- **Módulo de Detección de Incidentes:** Analiza los datos en tiempo real para detectar situaciones fuera de lo normal (picos de energía, tráfico o seguridad) y genera alertas.
- **Distribución de Alertas:** Las alertas se envían a través de diferentes canales (interfaz de usuario, SMS, correo electrónico) a los responsables de la ciudad.

6. Interfaz de Usuario (Aplicación Web/Móvil)

Proporciona un panel de control intuitivo para que los gestores de la ciudad puedan monitorear el estado actual, visualizar datos históricos, y actuar sobre las alertas.

- **Dashboard de Visualización:** Permite a los usuarios ver el estado en tiempo real del tráfico, el consumo de energía, agua, y seguridad, junto con gráficos y estadísticas.
- **Control de Alertas:** Los usuarios pueden gestionar y responder a las alertas generadas por el sistema, y ajustar parámetros del sistema si es necesario.
- **Accesibilidad:** La interfaz es accesible tanto desde dispositivos móviles como desde computadoras, facilitando la gestión remota de la ciudad.

7. Mantenimiento y Actualización del Sistema

Este componente se enfoca en la facilidad de mantenimiento y la capacidad de realizar actualizaciones del sistema sin interrumpir su funcionamiento.

- **Actualización de Software:** Se deben aplicar actualizaciones periódicas de software y seguridad sin afectar el servicio.
- **Monitorización de Estado:** Se deben monitorear continuamente los servidores, sensores y sistemas para detectar fallos y resolverlos rápidamente.