

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Arquitectura de Computadores y Ensambladores 2

Catedrático: Ing. Gabriel Díaz

Auxiliares: Sergio André Lima Corado, Estuardo Sebastián Valle Bances

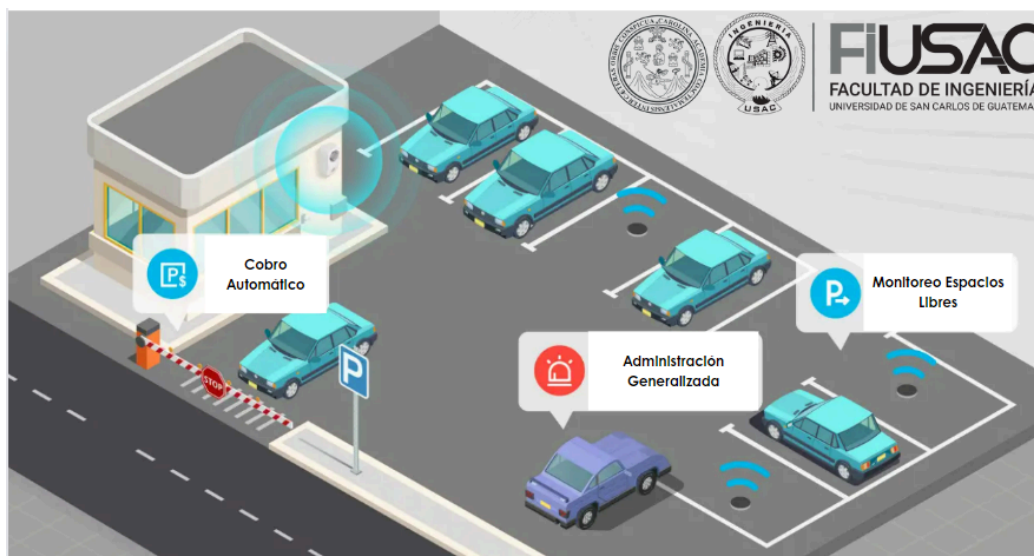


Proyecto 1

Monitoreo de tráfico vehicular en parqueo de la Facultad de Ingeniería, Fase 2 (Parking 4.0 y Smart Cities)

Objetivos:

- Implementar un sistema de comunicación IIoT para la gestión eficiente del parqueo.
- Aprender a enfrentar problemas y desarrollarlos bajo el IoT Stack Connected Framework.
- Implementar una solución Inteligente haciendo uso de protocolos MQTT, Dashboards y Bases de Datos.



Descripción:

El Departamento de Asuntos Económicos y Sociales de las **Naciones Unidas** ha pronosticado que todo el crecimiento de la población mundial se concentrará en las áreas urbanas, y se anticipa que la tasa de urbanización **alcanzará el 68% para el año 2050.**

En la Universidad de San Carlos de Guatemala, esto es evidente en horas pico, donde la demanda supera la capacidad disponible.

1. **Sobrepago:** Los usuarios a menudo no saben cuánto tiempo estarán en el campus, lo que los lleva a pagar más de lo necesario por el estacionamiento.
2. **Impacto ambiental:** Los estacionamientos acumulan contaminantes que, con la lluvia, se filtran hacia cuerpos de agua cercanos, afectando el entorno del campus.
3. **Falta de espacios adecuados:** La insuficiencia de espacios en áreas designadas obliga a los conductores a estacionar en lugares no autorizados, creando más congestión.
4. **Uso ineficiente:** Los conductores no siempre estacionan correctamente, lo que reduce la cantidad de espacios disponibles para otros vehículos en la universidad.

En respuesta a la necesidad de optimizar la gestión del parqueo en la Facultad de Ingeniería, se propone una segunda fase para la implementación de una talanquera inteligente. Esta fase incluye funcionalidades avanzadas, **como el manejo de saldos en tarjetas RFID**, una **plataforma web** para la administración del sistema, y la adopción de un sistema de comunicación **IoT basado en el protocolo MQTT.**

Dispositivo y Software

Investigadores de la Escuela de Ciencias y Sistemas, han determinado que la industria actual dentro del IoT, Smart Cities y 4.0 Parking lleva a las siguientes conclusiones sobre el desarrollo del dispositivo:

1. Gestión de la demanda de estacionamiento y optimización de espacios
2. Guía personalizada de estacionamiento
3. Sistemas de reserva de estacionamiento
4. Precios dinámicos de estacionamiento y optimización de políticas
5. Detección de zonas de estacionamiento, tarifas y violaciones por exceder el tiempo permitido

Por lo que se requiere implementar los siguientes requerimientos en el sistema:

1. Manejo de Saldos en RFID:

- **Estudiantes:** Cada tarjeta RFID asociada a un estudiante contará con un saldo predefinido. El costo por entrada al parqueo será de Q3. El sistema verificará el saldo antes de permitir el acceso; si el saldo es insuficiente, el acceso será denegado.
- **Administrativos:** Las tarjetas RFID asociadas al personal administrativo tendrán saldo ilimitado, permitiéndoles ingresar al parqueo sin costo.
- **Base de Datos (DB):** El sistema deberá contar con una base de datos que registre los saldos de los usuarios y sus accesos al parqueo. (Modelo Relacional Proporcionado en Anexos)

Además, almacenará en tiempo real la cantidad de vehículos dentro del parqueo y el total de espacios disponibles, o los datos que considere necesarios para el *Panel de Monitoreo del Parqueo*.

Las tarjetas RFID pueden ser agregadas al sistema de manera manual por este proyecto, es decir grabando los datos de los usuarios en la base de datos directamente. Al ingresar cada tarjeta también se debe de definir si es **Estudiante o Administrativo**. Previo a la calificación, el auxiliar proveerá los datos para añadir dos tags RFID extra a la base de datos. Y que se puedan utilizar para la calificación.

2. Comunicación MQTT:

- **Protocolo MQTT:** Se utilizará el protocolo MQTT para la comunicación entre los dispositivos IoT y el servidor central. En cuanto al broker de MQTT los estudiantes serán libres de poder utilizar:

Broker	Link
Eclipse Mosquitto -	Mosquitto
EMQX Community Edition -	EMQX

- **Topics MQTT a Implementar:**
 - **/talanquera:** Control de apertura y cierre de la talanquera.
 - **/clima:** Monitoreo de las condiciones climáticas (temperatura y humedad).
 - **/luz:** Gestión de la iluminación en el parqueo.
 - **/rfid:** Verificación y actualización de saldos, así como la validación de accesos.

Nota: El estudiante puede utilizar cualquier otro topic, en la calificación el topic a elección del grupo será analizado por medio de un script para verificar su funcionamiento.

3. Página Web de Administración:

- **Autenticación:** La página web incluirá un sistema de login para el administrador, con almacenamiento de las credenciales dentro de la base de datos utilizando contraseñas con hash (md5).
- **Panel de Control:** Una vez autenticado, el administrador tendrá acceso a:
 - **Panel de Clima:** Visualización **animada** en tiempo real de los datos de temperatura y humedad del parqueo. (p5.js)

- **Panel de Usuarios:** Listado de todos los usuarios registrados en el sistema, se deberá mostrar su saldo disponible, estado (dentro/fuera del parqueo), último ingreso y último egreso, y si es usuario o administrador. Al seleccionar un usuario, se mostrará todos sus datos personales, los datos de su RFID, todo su historial de ingresos y egresos en la talanquera, y la posibilidad de modificar el saldo (sumar/restar).

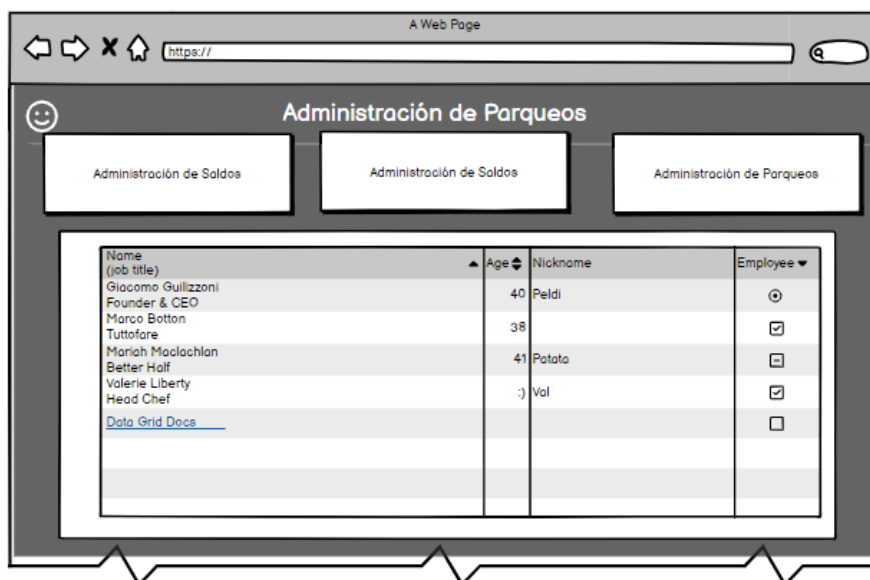


- **Panel de Monitoreo del Parqueo:** Estadísticas en tiempo real del parqueo, incluyendo:
 - Número de vehículos dentro del parqueo.
 - Número de espacios disponibles.
 - Porcentaje de ocupación del parqueo.
 - Porcentaje de usuarios externos (usuarios que pagaron con monedas).
 - Número de vehículos que han ingresado y salido en todo el día.

Dichas gráficas pueden ser realizadas haciendo uso de [Grafana](#), una potente herramienta para realizar dashboards. Para verificar su funcionamiento, se le solicitará a dos estudiantes que muestren este dashboard. (Solamente para 1 variable), el dashboard de Grafana puede ser desplegado tanto en Cloud como de manera local.

4. Automatización del Prototipo y Uso de Broker MQTT:

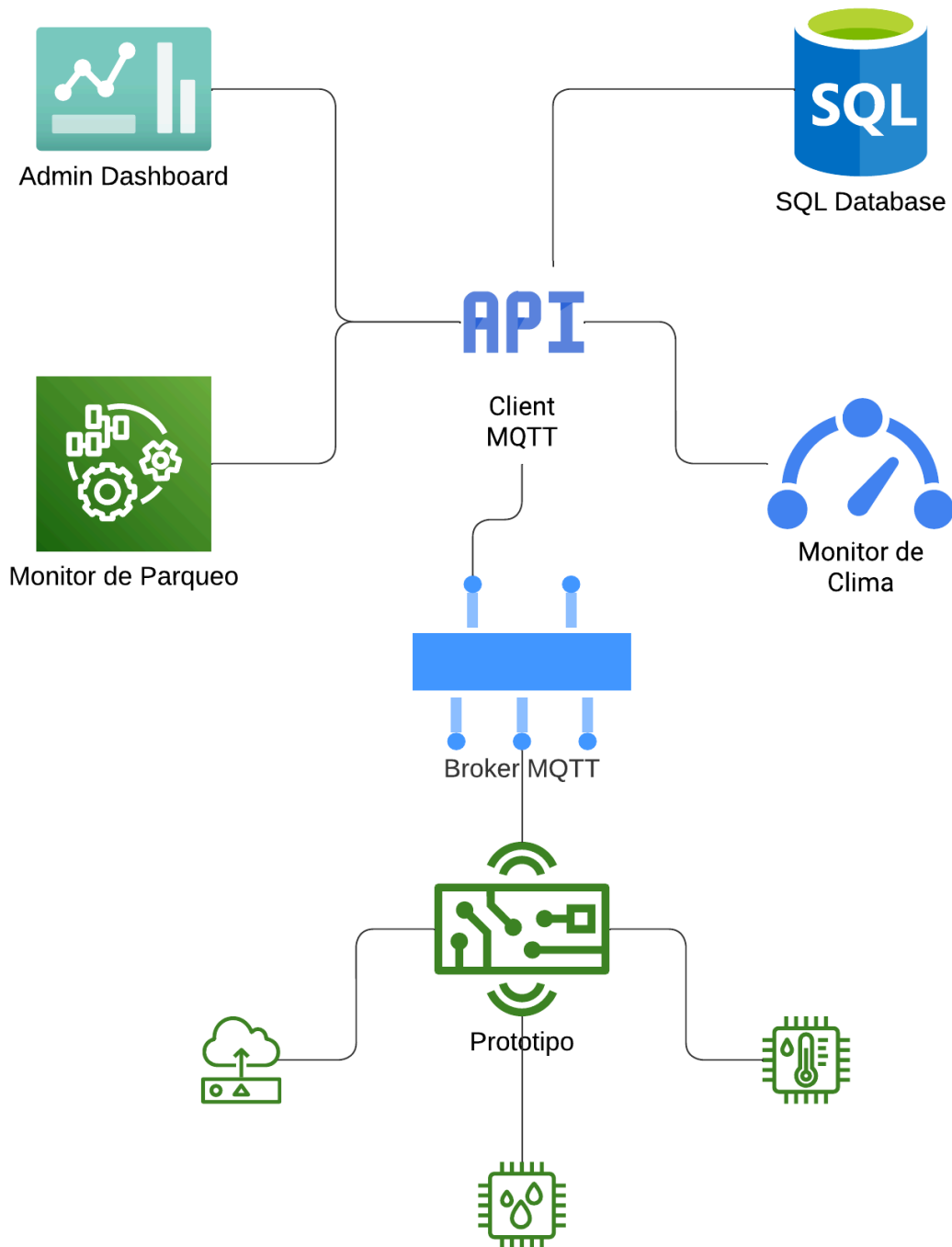
- **Cobro y Verificación de Saldos:** Cada entrada al parqueo tendrá un costo de Q3. El sistema verificará el saldo del usuario antes de permitir el acceso.
- **Control de Capacidad:** El sistema denegará el acceso si el parqueo ha alcanzado su capacidad máxima (100% de ocupación).
- **Validaciones de Acceso:** Se mantendrán las validaciones de la fase anterior, incluyendo la verificación de si un RFID ya ha sido utilizado para ingresar o salir del parqueo.
- **Interrupciones:** En esta fase no es obligatorio, aunque sigue siendo una opción válida implementar interrupciones para el control de la talanquera.



Materiales Requeridos:

1. **Sensores y Actuadores:** Sensores de movimiento, lectores RFID, y motores servo para la talanquera, similares a los utilizados en la fase anterior.
2. **Broker MQTT:** Implementación de un broker MQTT, como Mosquitto.
3. **Base de Datos:** Necesaria para la gestión de usuarios, saldos, y estadísticas del parqueo.
4. **Página Web:** Desarrollo de una aplicación web con un backend en Flask, Django o Node.js, y un frontend en React.js o Angular.

Arquitectura Sugerida de la Solución:



Repositorio de GitHub:

Todo el código utilizado y la documentación deberá ser subido a un repositorio de GitHub. En la entrega sólo deberá contener el link del repositorio, esto con el fin de evitar inconvenientes por el tamaño de los archivos al momento de la entrega. Se utilizará el mismo repositorio, solo se deberá de almacenar en la nueva Carpeta de Proyecto_1.

- Agregar al usuario del auxiliar como colaborador a su repositorio de github:
 - **Grupos 1 - 4:** andre-corado
 - **Grupos 5 - 9:** vallit0
- Para distribuir la carga de trabajo cada integrante del grupo debe de realizar al menos **3 commits** durante el desarrollo de la práctica.
- Todo código o documento que no se encuentre en el repositorio no será tomado en cuenta para la calificación.

Restricciones:

1. **Broker MQTT:** No se podrá calificar sin uso de un broker. **(-100%)**
2. **Base de Datos:** Los usuarios y las métricas necesarias deben de ser almacenados en Base de Datos. **(-50%)**
3. **Entrega Tarde:** Se deberán de mandar todos los entregables y commits en la fecha establecida, de no ser así se tendrá una penalización del 50%.

Documentación:

La documentación deberá de realizarse en formato Markdown en el README del repositorio de la práctica, dicho lo anterior se solicita:

- Descripción de la Solución
- Capas del Framework IoT
- Modelo de la Base de Datos
- Detallar Broker MQTT utilizado
- Topics y su nomenclatura
- Explicación de métodos y consultas importantes

Fecha de entrega: 28 de septiembre del 2024