**(Avance final) Etapa 3**

**integración de competencias**

Integrantes:

Kameron Rivera,

Marcelo Contreras,

Fabian Araya,

Diego Rojo,

Kevin Araya,

Andrés Pasten.

**Caso: MazturCars**

**“MazturCars”**, ubicada en el centro de Ovalle, tiene una capacidad de 50 vehículos y enfrenta graves problemas operativos por su sistema manual de control vehicular. Actualmente, los ingresos y salidas se registran en libreta, generando errores, filas, pérdida de tiempo y mala experiencia para el usuario.

**Integración de tecnologías 4.0:**

* La industria 4.0 promueve la automatización y digitalización de procesos mediante tecnologías como sensores inteligentes, internet de las Cosas (IoT), big data y sistemas ciberfísicos. En el contexto de un estacionamiento, la aplicación de estas tecnologías permite optimizar la gestión de accesos vehiculares, mejorar la seguridad, reducir errores humanos y proporcionar datos útiles en tiempo real para la toma de decisiones

**Objetivo general:**

* Diseñar e implementar un sistema automatizado para gestionar la entrada y salida de vehículos, asignación de espacios, cálculo de tarifas y monitoreo ambiental mediante tecnologías IoT.

**Objetivos Específicos:**

* Detectar vehículos al entrar/salir mediante sensores IoT.
* Capturar y registrar automáticamente las matrículas (LPR).
* Asignar dinámicamente espacios de estacionamiento.
* Calcular tarifas basadas en tiempo de uso.
* Medir temperatura ambiente en el acceso.
* Validad pago antes de autorizar la salida del vehículo.
* Enviar notificaciones al encargado en tiempo real.

**Requerimientos Funcionales:**

**Detección de Vehículos con IoT**  
El sistema debe detectar automáticamente la entrada y salida de vehículos mediante sensores IoT instalados en los puntos de acceso.

**Lectura de Matrículas (LPR)**  
El sistema debe capturar y registrar automáticamente la matrícula del vehículo utilizando tecnología de reconocimiento de placas (LPR) al momento de entrada y salida.

**Asignación Dinámica de Estacionamientos**  
El sistema debe asignar de forma automática y dinámica un espacio de estacionamiento disponible al vehículo ingresado.

**Cálculo de Tarifas por Tiempo de Uso**  
El sistema debe calcular el monto a pagar por cada vehículo en función del tiempo total de permanencia en el estacionamiento.

**Medición de Temperatura Ambiente**  
El sistema debe medir y registrar la temperatura ambiente en los accesos utilizando sensores de temperatura.

**Validación de Pago para Autorizar Salida**  
El sistema debe verificar que el pago correspondiente haya sido realizado antes de autorizar la salida del vehículo del estacionamiento.

**Notificaciones en Tiempo Real al Encargado**  
El sistema debe enviar notificaciones en tiempo real al encargado del estacionamiento ante eventos importantes (ej. entrada/salida, fallos de sensores, intentos de salida sin pago).

**tecnologías Utilizadas (industria 4.0)**

|  |  |
| --- | --- |
| Tecnología | Función |
| Sensor ultrasónico HC-SR04 | Detectar entrada/salida de vehículos |
| Sensor de temperatura DHT22 | Medir temperatura ambiental en tiempo real |
| Cámara LPR (Reconocimiento de matrículas) | Capturar automáticamente la placa del vehículo |
| Microcontrolador ESP32 | Procesar datos y comunicarse con la base de datos vía Wi-Fi |
| Base de datos relacional | Registrar entradas, salidas, temperatura, estado de espacios y pagos |
| Sistema web/app | Gestión de usuarios, pagos, notificaciones y reportes |

**Estructura de Clases (Adaptación del Diagrama)**

**SistemaEstacionamiento**

Contiene sensores, cámaras, espacios, registros y envía notificaciones.

* +registrarEntrada()
* +registrarSalida()
* +notificarEncargado(mensaje: string)

**SensorIOT**

Detecta la presencia del vehículo y mide temperatura ambiental.

* +detectarVehiculo(): boolean
* +enviarDatos(): void

**CamaraLPR**

Captura automáticamente la matrícula del vehículo.

* +capturarMatricula(): string

**EspacioEstacionamiento**

Asigna o libera un espacio a un vehículo.

* +asignarVehiculo(vehiculo): void
* +liberarEspacio(): void

**Registro**

Vincula vehículo, espacio, duración y tarifa.

* +emitirRegistro(): void
* +registrarPago(Pago):void

**Vehiculos**

Guarda datos de tiempo y permite cálculo de tarifa.

* +calcularTiempoUso(): int
* +calcularTarifa(): int

**Encargado**

Recibe notificaciones cuando ocurren eventos importantes.

* +recibirNotificacion(mensaje: string): void

**Pago**

Procesamos el pago del cliente en base a la tarifa y tiempo de uso.

* +procesarPago(): boolean

**Base de datos Propuesta:**

|  |
| --- |
| Tabla: vehiculos |
| Campo Tipo |
| patente TEXT |
| horaEntrada DATETIME |
| horaSalida DATETIME |

|  |
| --- |
| Tabla: registros |
| Campo Tipo |
| id INTEGER |
| patente TEXT |
| espacio INT |
| temperatura REAL |
| tiempoUso INT |
| tarifa REAL |
| estadoPago TEXT |

|  |
| --- |
| Tabla: espacios |
| Campo Tipo |
| id INTEGER |
| ocupado BOOLEAN |

|  |
| --- |
| Tabla: notificaciones |
| Campo Tipo |
| id INTEGER |
| mensaje TEXT |
| fechaHora DATETIME |

**Mapa de colores

El contenido generado por IA puede ser incorrecto. Diagrama de clases generales para el proyecto**

**Relaciones entre clases:**

**CLASE INICIO CLASE DESTINO RELACION PORQUE**

**SistemaEstacionamiento à EspacioEstacionamiento Agregación Contiene muchos espacios**

**SistemaEstacionamiento à SensorIOT Agregación Contiene muchos sensores**

**SistemaEstacionamiento à CamaraLPR Agregación Contiene muchas cámaras**

**SistemaEstacionamiento à Registro Composición Implementa registros de e/s**

**Registro à Vehiculos Asociación Registramos un vehículo**

**Registro à EspacioEstacionamiento Asociación Registramos el espacio usado**

**EspacioEstacionamiento à Vehiculos Asociación Se asigna un vehículo a un espacio**

**SistemaEstacionamiento à Encargado Asociación Se le envían notificaciones**

**Pago à Registro Agregacion se utiliza una instancia del pago**

**Prototipo del Sistema:**

**Interfaz de página de gestión de “MazturCars”:**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Descripción General del Prototipo Web "MazturCars"**

* Este prototipo web simula el funcionamiento del sistema automatizado de gestión de estacionamientos para "MazturCars", mostrando funciones clave como ingreso, salida y monitoreo en tiempo real. Desarrollado con HTML, CSS y JavaScript, busca demostrar cómo la digitalización y las tecnologías 4.0 pueden mejorar la eficiencia y experiencia del usuario.

**Secciones del Prototipo:** (Usamos de ejemplo la patente **PL3820**)

**Acceso de vehículos:**

* Esta sección simula el ingreso de un vehículo al estacionamiento. Al ingresar la patente, el sistema verifica disponibilidad, asigna un espacio si hay cupo y registra la hora de entrada. Digitaliza el proceso manual, reduce errores y simula la detección de matrículas (LPR) de forma conceptual.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Salida de vehículos:**

* En esta sección se gestiona la salida de vehículos, calculando automáticamente el tiempo de estadía y la tarifa según la patente ingresada. El sistema simula la validación del pago y autoriza la salida, liberando el espacio ocupado, lo que optimiza el cobro y mejora la experiencia del cliente.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Monitoreo y Notificaciones:**

Esta área multifuncional del prototipo cumple dos roles clave

* **Monitoreo Ambiental:** Presenta una simulación de la medición de la temperatura ambiente en el acceso al estacionamiento, demostrando la capacidad del sistema para integrar sensores IoT para recolección de datos ambientales.
* **Notificaciones al Encargado:** Muestra un flujo de mensajes en tiempo real dirigidos al personal. Estas notificaciones informan sobre eventos importantes como ingresos de vehículos, asignación de espacios, cálculos de tarifas, salidas confirmadas e intentos fallidos de ingreso por falta de cupo. Esto permite una supervisión constante y facilita la toma de decisiones operativas.

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Estado del Estacionamiento:**

* Esta sección muestra en tiempo real la ocupación de los 50 espacios del estacionamiento "MazturCars", indicando visualmente si están libres u ocupados (con patente). Incluye un contador de espacios disponibles y facilita una gestión eficiente y asignación dinámica.

**Imagen de la pantalla de un video juego

El contenido generado por IA puede ser incorrecto.Imagen que contiene Forma

El contenido generado por IA puede ser incorrecto.**

Ocupado

Estado de pago pendiente

**Imagen que contiene agua, grupo

El contenido generado por IA puede ser incorrecto.**

Estado libre

**Explicación de como funcionaria el código del sistema:**

**Ingreso de Vehículos:**

1. El sistema registrara e ingresara la patente del auto.
2. El sistema busca un espacio libre entre los 50 disponibles.
3. Si hay espacio:
   * Se registra la patente en ese espacio.
   * Se guarda la hora exacta de entrada.
   * Se muestra una notificación indicando que el auto entró y cuál espacio ocupa.
4. Si no hay espacio:
   * Se muestra una notificación de “estacionamiento lleno”.

**Salida de Vehículos:**

1. El sistema ingresa la patente de un vehículo que quiere salir.
2. El sistema:
   * Busca el registro de entrada del vehículo.
   * Calcula cuántos minutos estuvo estacionado.
   * Usa ese tiempo para calcular el precio que debe pagar (por ejemplo, $50 por minuto, mínimo $1000).
3. Luego:
   * Se muestra una notificación con el tiempo y el precio.
   * Se libera el espacio que ocupaba ese vehículo.
   * Se muestra otra notificación que confirma que el auto salió.

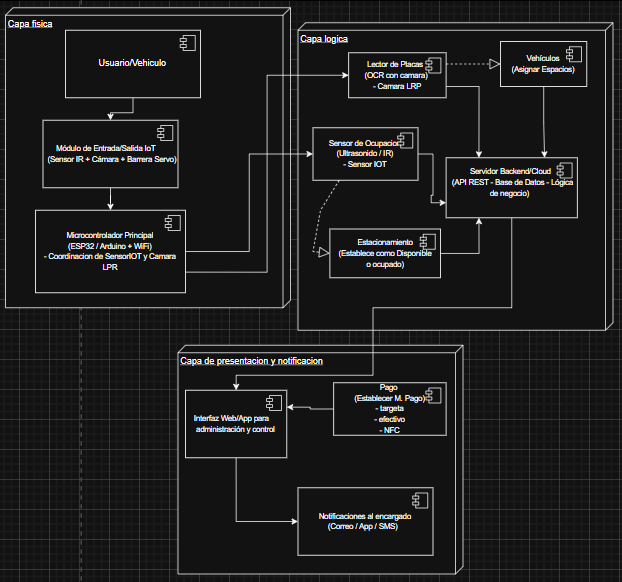
**Monitoreo Ambiental:**

* El sistema simula un sensor de temperatura en la entrada.
* Cada 5 segundos, cambia la temperatura simulada (por ejemplo, entre 20 y 30 °C).
* Esa temperatura se muestra en la interfaz gráfica.

**Notificaciones al Encargado:**

* Cada vez que ocurre algo importante (entrada, salida, tarifa, error, etc.), se escribe un mensaje en un área de notificaciones.
* Esto permite que el encargado vea en tiempo real lo que está ocurriendo en el sistema

**Diagrama de arquitectura desde un ámbito general del proyecto:**



**Funcionamiento y Flujo del Sistema**

**Capa Física**

* Usuario/Vehículo llega al estacionamiento.
* Es detectado por el Módulo de Entrada/Salida IoT, compuesto por:
* Sensor IR o láser (detecta presencia).
* Cámara (captura placa).
* Barrera servo (abre/cierra).
* El Microcontrolador (ESP32) gestiona la entrada de datos, toma decisiones locales, y se conecta a la nube vía Wifi.

**Capa Lógica**

* El Lector de Placas (OCR) analiza la imagen enviada desde la cámara y toma el número de matrícula.
* El Sensor de Ocupación informa al microcontrolador sobre los espacios disponibles.
* Ambos módulos envían datos al Servidor Backend/Cloud, el cual:
  + Recibe datos en tiempo real (vía API REST).
  + Administra la base de datos (registro de entradas, tiempos, placas).
  + Aplica lógica de negocio (tarifas, asignación de espacio).
* El módulo de Estacionamiento permite detectar si un espacio está ocupado o disponible y envía esta información al servidor
* El módulo de Vehículo posee los atributos que serán enviados al sistema para almacenar su información. Detectada por el lector de placas

**Capa de Presentación y Notificación**

* Los datos procesados se reflejan en la Interfaz Web/App para control administrativo.
* Esta información es calculada y procesada para emitir el comprobante de pago.
* Permite establecer el método de pago para la ejecución.
* El sistema también envía Notificaciones al encargado (por App, SMS o correo) cuando:
* Se detectan eventos como ingreso, salida, anomalías o espacio lleno.

**Mapa físico del estacionamiento**

**simbología:**

* **Sensor IR/CamaraLPR (Detecta si el espacio está ocupado y obtiene la patente del vehículo que lo ocupe, cambiando el estado a ocupado, impidiendo que se asigne el espacio)**

****

**Servidor/microcontrolador principal ESP32/Arduino + wifi (El microcontrolador, esta conectado a cada sensor IR y CamaraLPR a través de wifi, evitando así el uso de cables que complique la instalación y mantención del sistema, el microcontrolador esta físicamente ubicado junto al servidor, que es el encargado de almacenar todos los datos y asigna los espacios disponibles)**

**Un conjunto de letras blancas en un fondo blanco

El contenido generado por IA puede ser incorrecto.**

**Icono

El contenido generado por IA puede ser incorrecto.[Puntos de mejora: importante avance 2]**

* Mapa físico del sistema donde muestra todos los dispositivos, como se interconectar y ubican en él espacio.

**Icono

El contenido generado por IA puede ser incorrecto.**

* Agregar las funcionalidades de pago a los diagramas.

**Icono

El contenido generado por IA puede ser incorrecto.**

* Proponer ideas de escalabilidad y optimización del uso de sensores.

**Conclusión y reflexión.**

El desarrollo de Mazturcars se encargado de colocarnos en una posición muy ventajosa en cuanto a nuestra capacidad de innovar. En el día a día podemos observar el uso de diferentes tecnologías y, nosotros que nos encontramos en esta área, somos capaces de proponer verdaderas soluciones a diferentes problemáticas hacia este tipo de eventos, por lo que el caso de estacionamiento, combinando automatización IoT y procesamiento inteligente de datos para optimizar operaciones y mejorar la experiencia del usuario, ha sido una verdadera muestra de como podemos proponer ideas increíbles con el uso de las tecnologías (4.0) y como podemos pasar de una implementación conocida a una innovadora.

**pero…**

Aunque la implementación actual ha logrado resolver problemas operativos clave, como la gestión manual de ingresos y salidas, la asignación dinámica de espacios y la validación de pagos en tiempo real.

aún con eso, se podrían implementar algunas cuantas mejoras para mejorar la experiencia de usuario en otras áreas de implementación y escalabilidad del uso de los sensores, tales como:

**Enfoque técnico**

**optimización de sensores IoT:**

* Se podría considerar el trabajar con sensores mas avanzados para así mejorar la precisión de la detección de vehículos y minimizar errores en espacios ocupados.

**análisis de patrones con IA:**

* Se podrían aplicar algoritmos de IA para predecir la demanda de estacionamiento según horarios y eventos locales.

**integración de pagos electrónicos [mejorado]:**

* Implementar métodos de pago adicionales como QR, NFC o billeteras digitales para agilizar la salida y mejorar la experiencia de usuario.

**Escalabilidad**

**Interconexión con otros estacionamientos:**

* Se podría permitir que MazturCars comparta datos con otros sistemas similares dentro de Ovalle, creando así, una red inteligente de estacionamientos capaces de optimizar la movilidad urbana.

**Integración de aplicación/web y módulo de reservas anticipada:**

* Tomando en cuenta la simulación de prototipo de interfaz web, se podría tomar en cuenta la funcionalidad para integrarlo no solo de manera administrativa, sino que también para el usuario de cualquier usuario. La sección de usuario podría dar la opción de reservar un espacio mediante el sitio web o aplicación móvil, reduciendo el estrés de encontrar estacionamiento en horarios de alta demanda.

**Innovación enfocada en la calidad de servicio**

**Sistema de recomendaciones:**

* Analizar diferentes datos históricos para sugerir a los usuarios mejores horarios para estacionar su vehiculo y mejorar la gestión de espacios en hora punta.

**Monitoreo de seguridad:**

* Se podrían agregar detecciones de anomalías con IA para identificar comportamientos sospechosos o posibles intentos de fraude.

**energía sostenible [falta de conocimiento aun esta implementación]:**

* Se podrían implementar paneles solares para alimentar a los sensores y microcontroladores, reduciendo costos operativos y generando un impacto ecológico positivo. :3



**Preguntas de cierre**

**1. ¿Qué aprendimos de la actividad realizada?**

* Como grupo, aprendimos cómo se puede automatizar un estacionamiento usando tecnologías como sensores, microcontroladores y cámaras. También entendimos cómo organizar el sistema en capas (física, lógica y de presentación) y cómo representarlo con diagramas para explicar mejor su funcionamiento, además, fuimos capaces de responder al llamado de innovación, al proponer nuevas soluciones e ideas para implementar a futuro.

**2. ¿En qué ámbitos se puede aplicar lo que realizamos en la actividad?**

* Este tipo de sistema se puede aplicar en estacionamientos de empresas, centros comerciales, edificios públicos o privados, y cualquier lugar donde se quiera controlar mejor el acceso de vehículos y también en otras áreas ya que nos sirve como base para otros proyectos que usen sensores e IoT.

**3. ¿Hubo algún término, definición o parte del proceso que necesitemos reforzar? ¿Cuál?**

* Sí, como grupo creemos que sería útil reforzar cómo funciona el reconocimiento de patentes y la conexión del sistema a la nube. También sería bueno repasar cómo se crean y usan las APIs para que los sensores se comuniquen con la base de datos y mas conocimiento de cómo podemos aplicar las diferentes mejoras a futuro, asegurando así la creación de un proyecto sostenible y escalable, de fácil uso y enfocado en la comodidad humana.