**UNIVERSIDAD CATOLICA DE HONDURAS**

**“CAMPUS JESUS SACRAMENTADO”**



**ASIGNATURA:** Programación de Negocios

**CATEDRATICO:** Ing. Héctor Sabillon

**ALUMNO:** Ramón Humberto Ventura Rodríguez

**CUENTA:** 0319-1998-00321

**FECHA:** 15/07/2019

**MANUAL DE DESARROLLO**

**Roles**

En este caso todos los roles y responsabilidades fueron realizadas por una misma persona.

**Plan de Trabajo**

El plan de trabajo básicamente se compuso de las siguientes actividades:

* Determinación de los requerimientos del sistema
* Diseño de la base de datos
* Desarrollo del código c#
* Diseño de pantallas

En primer lugar se necesitaba definir los requerimientos del proyecto solicitado, se determinó que el estacionamiento requería tres tablas diferentes:

Tipo\_Vehiculos: Esta tabla se encarga de registrar el tipo de vehículo de cada uno de los vehículos que fueran ingresando al estacionamiento.

Vehículos: Encargada de registrar el código de la placa del vehículo.

Cobro: Probablemente la más compleja, ya que almacena registro del pago de acuerdo al tipo de vehículo y a su vez registrar la hora de entrada, hora de salida y el tiempo que estuvo en el estacionamiento. Todo esto con el objetivo de poder generar un reporte de los ingresos totales.

Una vez finalizado el diseño de las entidades correspondientes, se decidió que la arquitectura de la aplicación estaría basada en tres capas: Capa de datos, capa de negocio y capa de presentación. Esto con la finalidad de poder afianzar conocimientos sobre distintas prácticas de la programación orientada objetos.

La arquitectura en capas nos permite desarrollar de una forma más clara aunque costosa en tiempo, sin embargo en caso de obtener errores o si decidimos actualizar un bloque de código será mucho más sencillo.

La capa de datos es la encargada de comunicarse con la base de datos, por lo tanto para cada operación es recomendable realizar un procedimiento almacenado y de esta forma llamar dichos procedimientos desde el código c# y al mismo tiempo enviar los parámetros requeridos.

La capa negocio únicamente fue utilizada como una conexión entre capa de datos y la capa presentación.

Por último la capa de presentación es encargada de almacenar los diseños y enviar parámetros a las capas superiores.

En cuanto a diseño se decidió seguir los lineamientos del material design, instalando la extensión material design themes desde el visual studio. La selección de colores se hizo en base a la paleta definida por los mismos lineamientos.

Cada una de las ventanas es llamada dentro de un grid contenedor ubicado en un single page el cual permanece estático.

Las opciones dentro de la aplicación son las siguientes:

Entrada: Controla el ingreso de vehículos al parqueo, en caso de que el vehículo no este registrado lo hace automáticamente y finalmente se almacena la hora de entrada obteniendo la hora del sistema.

Salida: Controla la salida de los vehículos. Genera el cobro de acuerdo al número de placa que se ingrese, por ultimo verifica si la placa se encuentra en la lista de estacionamiento para generar un factura a dicho vehículo por salir.

Estacionamiento: Muestra una lista de los vehículos que están en el estacionamiento.

Vehículos: Muestra una lista de todos los vehículos que han estado en el estacionamiento.

Reportes: Muestra una lista de las horas de entrada y de salida, tiempo en el estacionamiento y pago para cada vez que un vehículo fue al estacionamiento.

# **BIBLIOGRAFIA**

1. *Ecured*. (15 de 07 de 2019). Obtenido de https://www.ecured.cu/Arquitectura\_en\_Capas
2. *Material Design*. (15 de 07 de 2019). Obtenido de https://material.io/
3. *Microsoft Developer Network*. (15 de 07 de 2019). Obtenido de https://msdn.microsoft.com/es-es