**PROYECTO FINAL**

**DEL CURSO SQL**

BASE DE DATOS DE UN ESTACIONAMIENTO



Alumna: Camila Larroca

Tutor: Sergio Occhipinti

Profesora: Lucia Soledad Blanc

1. DESCRIPCION DE LA TEMATICA

Introducción

En primer lugar, nos encontramos frente a un proyecto de Base de Datos, cuya temática e idea principal surge a partir de la realización de una página web en JavaScript, realizado en dicha materia de la carrera de Backend; por lo que quise continuar con mi proyecto, dando lugar a la Base de Datos necesaria para un estacionamiento de autos.

Objetivos

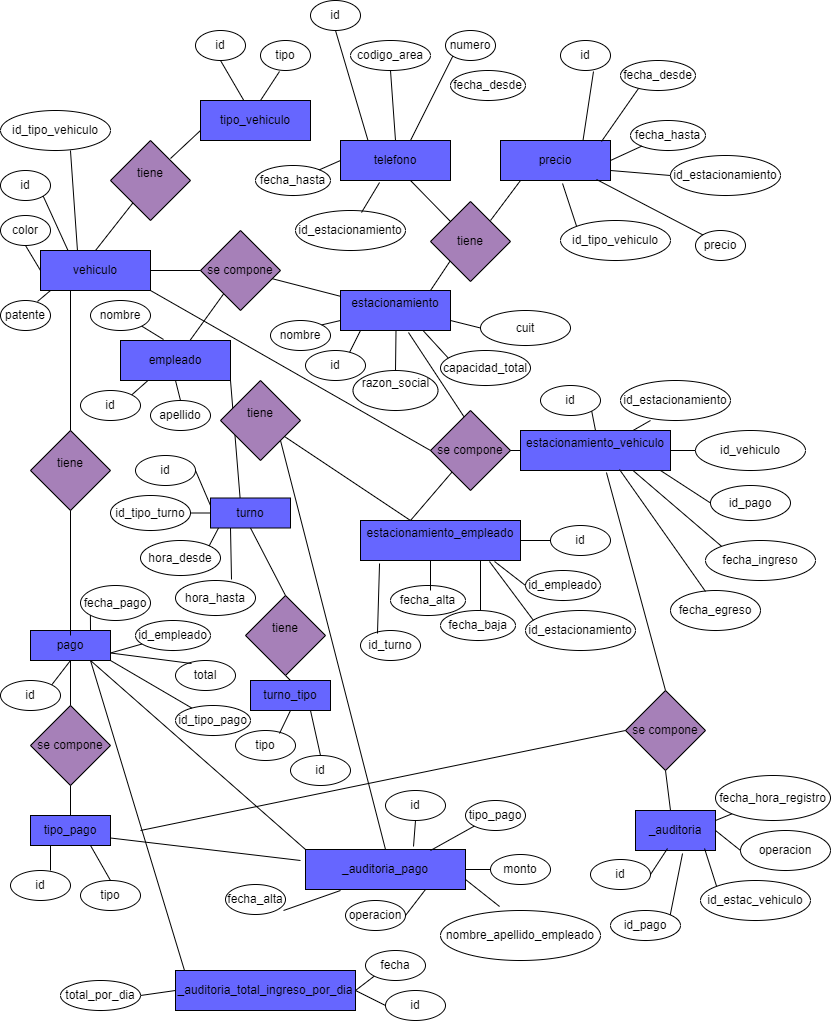
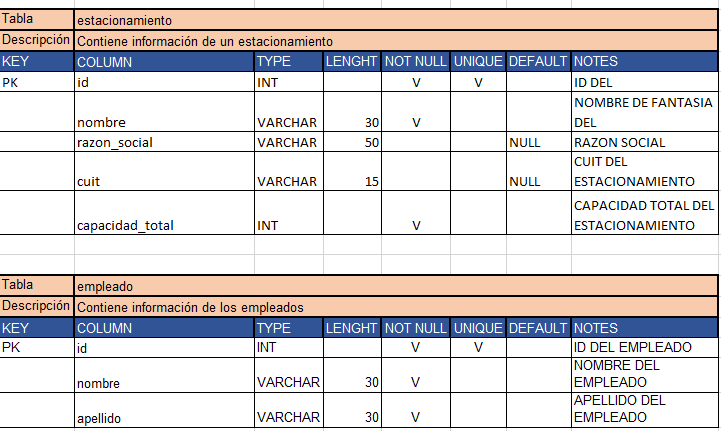
Con este proyecto busco plasmar lo aprendido en el transcurso de la materia, tanto en lo conceptual como en la parte práctica, y hacer una base de datos que cualquier estacionamiento de autos pueda utilizar en su negocio.

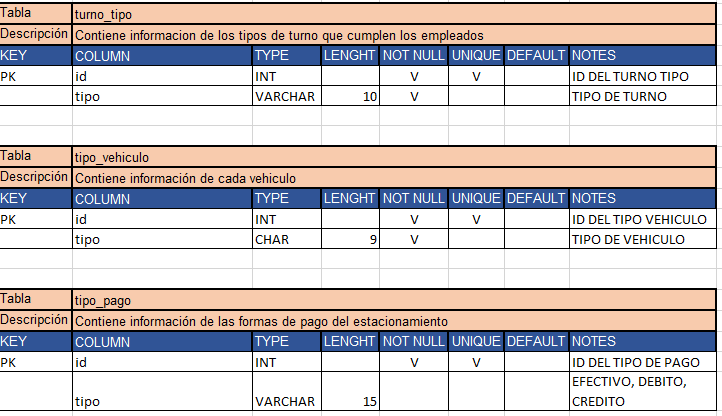
Situación problemática

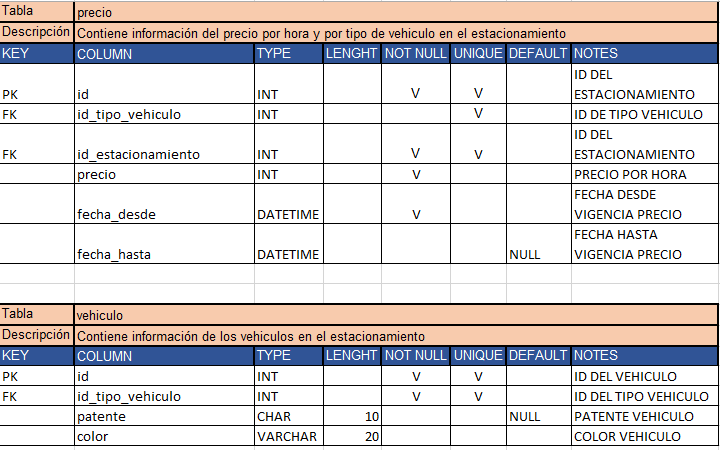
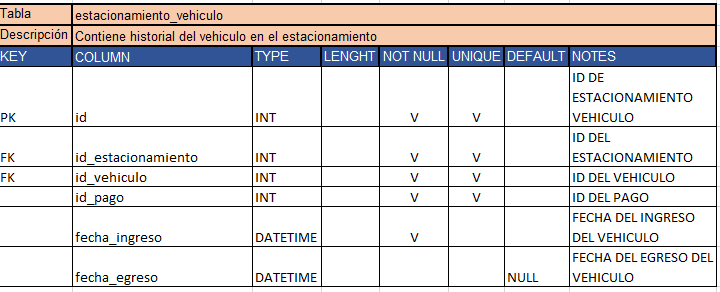
Creo que el crecimiento poblacional, las distancias entre distintos destinos, la economía, los robos de vehículos o a partes de los mismos, los coches debido al gran volumen de aquellos en un mismo sitio, se fue tornando necesario que cada vez hayan más estacionamientos de autos, por lo que dicho crecimiento requiere de un sistema que brinde información necesaria en relación a un estacionamiento en sí, como de sus empleados y los vehículos que utilizan dicho espacio. He aquí donde vengo a brindar mi base de datos.

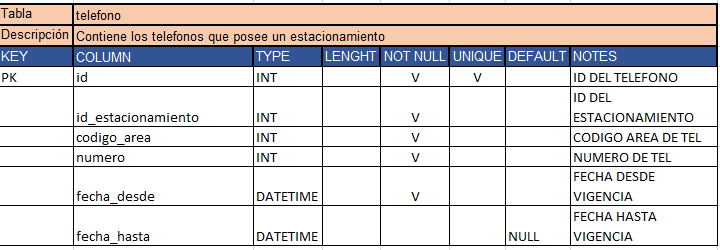
Modelo de negocio

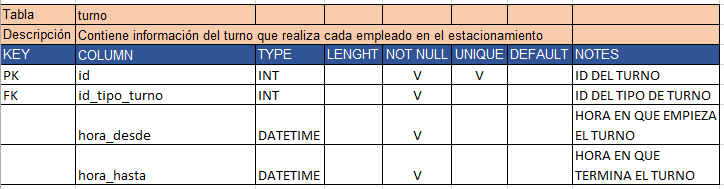
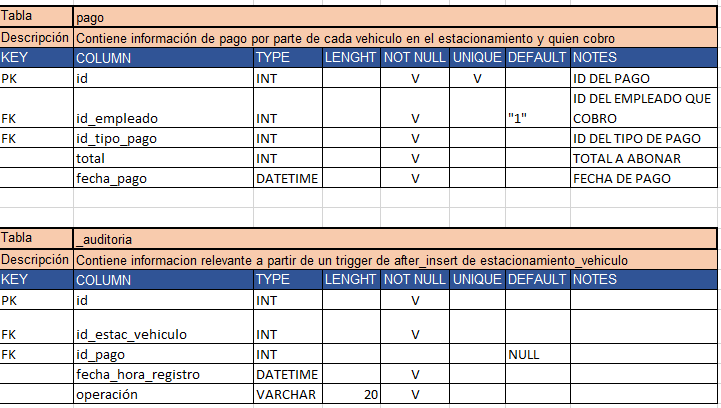
Mi idea es que todo aquel estacionamiento de vehículos (auto, moto, camioneta) que necesite usar una base de datos, pueda utilizar la creada por mí; es decir que no está formulada para un estacionamiento en particular, si no para un ESTACIONAMIENTO EN GENERAL, sea en Córdoba o en cualquier parte del país, y por qué no de cualquier país. Ya que el objeto de dicha base de datos es siempre el mismo, UN VEHICULO.

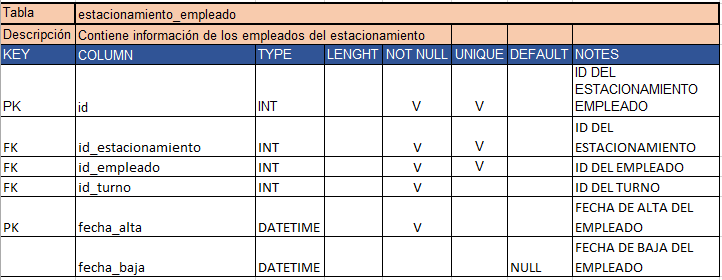
1. MODELO ENTIDAD-RELACION
2. ****TABLAS

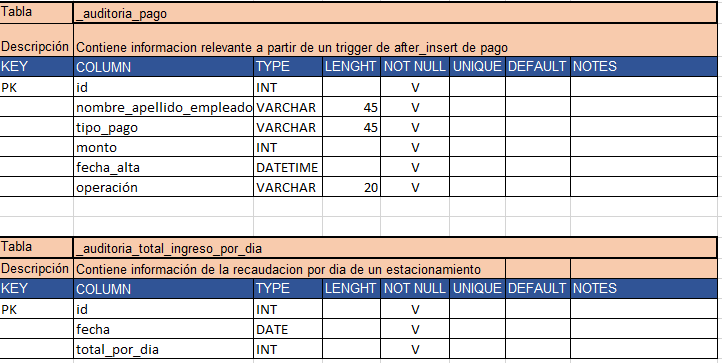
****

****

****

****

****

****

1. VISTAS

Vista 1: autos\_en\_estacionamiento

Descripción: es una tabla que me brinda un listado de los autos que se encuentran en el estacionamiento.

Objetivo: visualizar de forma rápida y sencilla la cantidad de vehículos que se encuentran en el estacionamiento, saber la capacidad del mismo, dándome información de hora de entrada, patente y color de los mismos.

Tablas/Datos: utilizo la tabla ‘estacionamiento\_vehiculo’, y extrajo de ella la fecha de ingreso. Hago un JOIN con la tabla vehículos, para utilizar la patente y color de cada vehículo. Incorporo un condicional WHERE, en el que indico que tanto los datos de fecha de egreso y de id de pago (de estacionamiento\_vehiculo), sean nulos; de esta forma, solo tiene en cuenta registros que ESTEN en el estacionamiento.

Vista 2: movimiento\_vehiculo.

Descripción: creo una vista que me detalla el movimiento de los vehículos y el total a abonar de cada uno.

Objetivo: tener un conocimiento del movimiento en el estacionamiento, para saber capacidad y cuánto dinero va ingresando. Vista para controlar el estacionamiento.

Tablas/Datos: utilizo la tabla ‘estacionamiento\_vehiculo’, tomo de ella los datos id de vehículo, fecha de ingreso y fecha de egreso, y hago un JOIN con la tabla ‘pago’, insertando en el WHERE la condición de que la fecha de egreso no sea nula, de esta forma, toma los vehículos que ya egresaron del estacionamiento.

Vista3: empleados\_activos.

Descripción: en esta vista, detallo los empleados activos en la nómina de mi estacionamiento y a qué turno pertenecen.

Objetivo: tener un conocimiento de la nómina de empleados.

Tablas/Datos: utilizo la tabla ‘empleado’, tomo el id, nombre, apellido, hago un JOIN con la tabla ‘estacionamiento\_empleado’ para obtener la fecha de alta de los mismos, y vuelvo a hacer dos JOIN mas, con la tabla ‘turno’ y con la ‘turno\_tipo’, para saber a qué turno corresponde cada empleado; utilizo en el WHERE la condición de que la fecha de baja del empleado sea nula, de esta forma, me mostrará solamente empleados activos.

Vista 4: precio\_actualizado\_por\_vehiculo

Descripción: en mi estacionamiento, utilizo un precio dependiendo del vehículo (auto, moto o camioneta), de esta forma creo una vista que me informe el precio por hora vigente de cada vehículo.

Objetivo: tener una actualización constante del precio de cada vehículo.

Tablas/Datos: utilizo para esta vista, la tabla ‘precio’ y con ella los datos de id de estacionamiento, el precio propiamente dicho, y la fecha desde cuando está en vigencia. A su vez, hago un JOIN con la tabla ‘tipo\_vehiculo’, así puedo obtener el id y tipo de vehículos; y establezco en el WHERE la cláusula de que la fecha hasta del precio, sea nula, obteniendo solamente precios actuales

Vista 5: total\_facturado\_por\_turno

Descripción: utilizo una vista para saber cuál es el momento del día que se factura más en el estacionamiento.

Objetivo: quiero saber en qué turno se recauda más para saber en qué momento del día tener más empleados.

Tablas/Datos: en esta vista utilizo en primer lugar la tabla ‘estacionamiento\_vehiculo’, y hago un JOIN con la tabla pago, haciendo una suma del total de cada pago; a su vez hago un JOIN con la tabla ‘empleado’, utilizando su id, y haciendo un JOIN nuevamente con la tabla ‘estacionamiento\_empleado’, para que por medio de aquel, pueda tener el dato del id del turno, hacer un JOIN con la tabla ‘turno’, y con la tabla ‘tipo\_turno’, agrupando por tipo de la tabla ‘tipo\_turno’ y ordenándolos también por tipo de la misma tabla.

1. FUNCIONES

Función 1: `fc\_insert\_auditoria\_pago`

Descripción: creo una función que es llamada desde un trigger (`tr\_after\_insert\_pago`), y reciba ciertos parámetros y realice la inserción en otra tabla de ciertos valores, basados en los primeros

Objetivo: tener un control del pago y del empleado que estuvo a cargo de la transacción.

Tablas/Datos: esta función recibe como parámetro, nombre\_apellido\_empleado, de quien realizó la transacción, siendo un VARCHAR (45), recibe un tipo\_pago (haciendo referencia a aquél de la tabla pago) siendo un VARCHAR (45), recibe a su vez un monto (total a abonar por hora de estacionamiento) siendo un INT, y recibe también una fecha alta, siendo el momento del pago, de tipo DATETIME, y por ultimo recibe como parámetro también operación, la cuál será un insert. Retorno un valor booleano.

Función 2: `fc\_calcular\_total\_por\_dia`

Descripción: realizo una función que recibe como parámetro una fecha y me retorna un INT, ya que quiero saber cuánto se recauda en una fecha determinada. Por lo tanto es una función NOT DETERMINISTIC, y de READS SQL DATA.

Objetivo: saber cuánto se recauda en un día determinado

Tablas/Datos: utilizo la tabla ‘pago’ para crear mi función, y dentro de esta utilizo la columna ‘total’. Hago un declare de ‘total\_por\_dia’, para que guarde la suma del total de la tabla ‘pago’. Y como solamente quiero saber la fecha y no la hora, determino en el WHERE que únicamente me muestre el dato tipo DATE, y que sea igual a lo que le paso por parámetro. Retorna un int de ‘total\_por\_dia’.

1. STORE PROCEDURE Y TRIGGERS

Procedure1: `sp\_total\_ingreso\_por\_dia`

Descripción: creo un store procedure que llama a una función en el que calculo el total de dinero ingresado por día.

Objetivo: control de la recaudación

Tablas/Datos: utilizo la tabla ‘pago’ y creo una tabla llamada `\_auditoria\_total\_ingreso\_por\_dia`. El store procedure llama a la función `fc\_calcular\_total\_por\_dia` a la que le paso por parámetro una fecha determinada. VER FUNCION 2.

Procedure2: `sp\_empleados`

Descripción: creo un store procedure que me muestre los empleados activos del estacionamiento.

Objetivo: llevar un control de los mismos.

Tablas/Datos: utilizo todos los datos de la tabla ‘empleado’, siendo los mismos id, nombre, apellido; hago un COUNT del id, para saber cuántos empleados son; y un JOIN con la tabla estacionamiento\_empleado, ya que necesito saber la fecha de alta y baja, y establezco en el WHERE la condición de que la fecha de baja sea NULL. Mostrándome así, solo empleados activos.

Trigger1: `tr\_insert\_vehiculo`

Descripción: creo un trigger de tipo after insert de la tabla ‘estacionamiento\_vehiculo’.

Objetivo: necesito que se inserte de forma automática en la tabla ‘\_auditoria’, cada vez que ingresa un vehículo al estacionamiento, y así ir teniendo un mayor control de dicho movimiento.

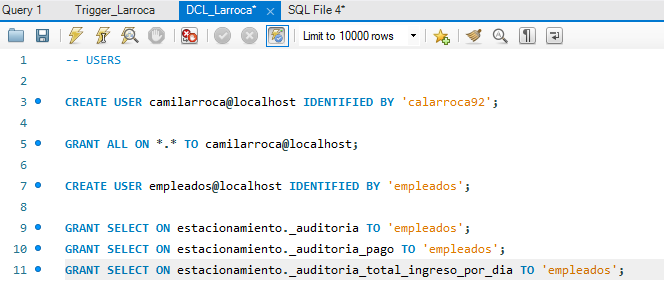
Tablas/Datos: a partir de la tabla ‘estacionamiento\_vehiculo’, se dispara un trigger que luego de insertar un dato en aquella tabla, es decir, un nuevo ingreso de vehículo, se inserta en la tabla ‘\_auditoria’, en cada celda los datos de id\_estac\_vehiculo, fecha\_hora\_registro, operación, ingresando los valores: NEW.id, NEW.id\_pago, NEW.fecha\_ingreso 'insert\_update')

Trigger2: `tr\_insert\_update\_vehiculo`

Descripción: Creo un Trigger a partir de la tabla estacionamiento\_vehiculo, hago un after update insert, luego de actualizar un registro, se inserta en la tabla \_auditoria, con fecha actual.

Objetivo: cuando se actualiza un registro en una tabla, se inserta luego la fecha y hora actual de dicha inserción. Tengo un mayor control ya que la fecha está siempre actualizada

Tablas/Datos: utilizo la tabla ‘estacionamiento\_vehiculo’ y con un before insert, antes de que se inserte un nuevo registro en aquella, por cada celda, inserto la hora y la fecha actual.

1. USUARIOS Y PERMISOS

Script creación de objetos:

Link a github con cada script: <https://github.com/cami-larroca/estacionamiento.git>

Script inserción de datos:

Link a github con script: <https://github.com/cami-larroca/estacionamiento.git>

EER DIAGRAM (MySQL Workbench):

Link github:

1. INFORMES

De acuerdo a la vista creada ‘total\_facturado\_por\_turno’, podemos ver en el siguiente gráfico además del monto total por turno (hasta la fecha), cuál es el turno que genera más ingresos en el estacionamiento; pudiendo decir que se trata del turno tarde, que en nuestra base de datos abarca el horario de 18:00:00 hs hasta las 00:00:00 hs. Respecto a la vista de ‘empleados\_activos’, podemos ver un mayor porcentaje que corresponden al turno noche. Y en el caso de ‘precio\_actualizado\_por\_vehiculo’, podemos ver el valor de cada uno y gráficamente qué monto es el mayor.

1. USO DE HERRAMIENTAS

MySQL Workbench: para la creación completa de la base de datos y la utilización de Lenguajes DDL, MDL, DCL y TCL.

Microsoft Excel: para realizar el listado de tablas y para realizar los gráficos para informes