



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION**

<b>AÑO:</b>	2018	<b>PERIODO:</b>	SEGUNDO TÉRMINO
<b>MATERIA:</b>	SISTEMA DE BASES DE DATOS I	<b>PROFESORES:</b>	GUSTAVO CALI, IRENE CHEUNG FRANK MALO, GINGER SALTOS, JOSÉ VILLA
<b>EVALUACIÓN:</b>	TERCERA	<b>FECHA:</b>	12/02/2019

**NOMBRE:** \_\_\_\_\_ **MATRICULA:** \_\_\_\_\_

**Sección 1 (30%):**

**Desarrolle el modelo conceptual del siguiente modelo de negocios.**

Se le pide diseñar la base de datos para la aplicación Polimatch que le permita encontrar una cita por el día de San Valentín.

Polimatch es una aplicación que utiliza la información personal de los usuarios para mostrar una lista de personas acorde a sus gustos. Para comenzar a utilizar la aplicación el usuario debe registrarse usando una red social como Facebook o Instagram, para garantizar que los perfiles sean auténticos, además de obtener información personal como los nombres completos, la edad, el género, teléfonos, foto de perfil, ocupación profesional, ciudad donde vive y los correos electrónicos.

El siguiente paso es elegir algunos intereses como: deportes, comida, películas, etc. Finalmente, para completar su perfil se pide que elija opciones para una cita perfecta para San Valentín como por ejemplo: cena romántica, karaoke, playa, vista al atardecer, picnic, etc.

Polimatch le mostrará los perfiles que tienen afinidad con los gustos registrados por el usuario. La mecánica es simple, el usuario va revisando uno por uno los perfiles candidatos pero tiene que votar, si le gusta (like) o sino le gusta (pass) antes de ver el siguiente perfil, todos los likes y pass se guardan. Si le da like a una persona éste se registra anónimamente, y si los dos usuarios se gustaron mutuamente se crea un match entre ellos. Al comenzar el match tendrá un estado 'INICIAL' donde se habilitará un chat por 24 horas. La idea es que en 24 horas puedan chatear e interactuar de tal forma que puedan decidir si continuar o no el contacto con esa persona.

Si en esas 24 horas uno de los dos no quiere continuar con el contacto, se le presentará la opción de culminar con el chat y el match cambiará a estado 'TERMINADO'. También se puede dar el caso de que uno de los dos no escriban, por lo tanto después de 24 horas el match cambiará a estado "FALLIDO".

Si por el contrario en esas 24 horas eligen continuar el contacto, el match cambiará de estado a "CITA", lo cual significa que el chat no tendría límite de tiempo y los usuarios cambiarán a un estado de "NO DISPONIBLE" eso significa que no aparecerán en la búsqueda de otros usuarios.

En estado 'TERMINADO' y 'FALLIDO' se perderá el match y el chat entre esos usuarios. El usuario podrá realizar un máximo de 10 likes por día e ilimitado de pass.

**NOTA:** No olvide que en su modelo conceptual deben constar las entidades, atributos, cardinalidades de atributos y relaciones, relaciones y nombres de las relaciones.

**Sección 2 (20%):**

Tomando en cuenta el modelo conceptual de la Sección 1, grafique el modelo lógico completamente normalizado.

**Sección 3 (20%):**

Utilizando el esquema de la base de datos db\_parqueos, escriba los queries solicitados:

- Calcule el promedio de días de las reservas tipo 'Estándar' del mes de febrero 2019 realizadas en el parqueo 'Parqueo Central'.
- Encuentre el listado de personas (sin repetir) que reservaron un parqueo durante diciembre de 2018 y que contrataron servicio de transporte hacia el 'Aeropuerto Internacional'.
- Liste todos los centros de parqueo (Parking) indicando el nombre y el total de reservas canceladas (estado) durante el año 2018 ordenadas de forma descendente.
- Encuentre el centro de parqueo (Parking) más utilizado por los usuarios que tienen nivel de acceso '5' y que tienen como destino 'Estación Central'.

**Sección 4 (30%):**

Utilizando el esquema de la base de datos db\_parqueos, realice lo solicitado:

- a) Crear un procedimiento llamado sp\_nueva\_reserva, el cual va a recibir los datos requeridos para crear una nueva reserva. Usted NO deberá hacer las validaciones al inicio del procedimiento, debe realizarlas después de haber hecho la inserción en la tabla correspondiente, es decir **para corregir un error debe usar transaccionalidad en el procedimiento.**

Las validaciones que debe realizar són:

- Verificar que el parqueo no esté ocupado
- Verificar que el tiempo de inicio no sea mayor al tiempo de fin.

Los datos de salida con la descripción del error queda a su criterio.

- b) Considerando el schema de la siguiente tabla:

```
Create Table ReservaRegular(  
    IDReservaRegular int(11) AUTO_INCREMENT primary key,  
    IDLugarParking int(11) not null,  
    IDParking int(11) default 1,  
    Inicia timestamp default current_timestamp,  
    Fin timestamp default '0000-00-00 00:00:00',  
    Longitud int(11) not null check (longitud >= 0),  
    foreign key (IDLugarParking) references LugarParking(IDLugarParking),  
    foreign key (IDParking) references Parking(IDParking)  
);
```

Datos existentes:

IDReserva Regular	IDLugar Parking	IDParking	Inicia	Fin	Longitud
1	1	1	'2019-02-11 00:09:00'	'2019-02-011 00:11:00'	2
2	2	3	'2019-02-11 00:19:00'	'2019-02-011 00:22:30'	3

Escriba cuales serán los registros finales de la tabla ReservaRegular de acuerdo a las siguientes sentencias de sql:

- **INSERT INTO** ReservaRegular (**IDReservaRegular ,IDLugarParking, Inicia, Longitud** )  
**VALUES(0, 3, '2019-02-13 00:19:00' , 0);**
- **INSERT INTO** ReservaRegular **VALUES(0, 3, '2019-02-11 00:19:00' , 0);**
- **INSERT INTO** ReservaRegular **VALUES(0, 3, 1, '2019-02-21 00:14:00' , '2019-02-21 00:15:00' , 1);**
- **DELETE FROM** ReservaRegular **WHERE IDParking = 1;**
- **UPDATE** ReservaRegular **SET longitud = -2 WHERE IDReservaRegular = 2;**

c) A usted le enviaron el siguiente cuadro con queries y sus respectivas frecuencias de uso anual, **de ser necesario** escriba la(s) sentencia(s) que permita optimizar los queries. Justifique su respuesta.

	<b>QUERY</b>	<b>FRECUENCIA ANUAL</b>
<b>A</b>	CREATE TABLE TEMP ( ID INT PRIMARY KEY, Usuario VARCHAR(1024), CantReservas INT(11), Minutos DECIMAL(10,0));	720
<b>B</b>	UPDATE TipoReserva SET precio = precio + 9 WHERE IDTipoReserva IN ( SELECT IDTipoReserva FROM LugarParking WHERE IDEstado = ( SELECT IDEstado FROM Estado WHERE Estado = 'Vacio'));	720
<b>C</b>	SELECT u.Nombre, count(O.idReservaRegular) as CantReservas FROM Usuario u JOIN Operacion O USING(IDUsuario) JOIN ReservaRegular USING(IDReservaRegular) GROUP BY IDUsuario;	720

**ESQUEMA DE BASE DE DATOS DE PARQUEADEROS EN CENTROS COMERCIALES**  
Nombre de la base de datos: *db\_parqueos*

