

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA



Proyecto de base de datos

Asignatura: Estructura de Datos

Profesor: Delia Ibacache Bastías

Integrantes: Dorian Canales Gutierrez
Vicente Herrera Meza
Sebastian Santos Ramirez
Ignacio Villalobos Jeria

Índice

1. Definición de la problemática	3
1.1 Elección de una problemática o necesidad	3
1.2. Identificación de las necesidades	4
Imagen 1: Prototipo interfaz	5
2. Diseño conceptual	6
2.1. Lista Entidades y atributos	6
Tabla 1: Descripción de entidades del modelo de datos	10
2.2. Descripción de las asociaciones	11
Tabla 2: Asociaciones	13
2.3. Modelo Entidad-Relación	14
Imagen 2: Modelo Mer en pdf	14

1. Definición de la problemática

1.1 Elección de una problemática o necesidad

Como equipo de trabajo hemos optado por elegir como problemática para encontrarle una solución óptima al manejo de una base de datos a “**SaborViña**”, ya que relata un problema que puede estar afectando de manera real a muchos pequeños y medianos restaurantes, no tan son en Viña del Mar sino que también a lo largo de todo Chile.

Teniendo en consideración que la problemática enseñada por “SaborViña” tiene como principal objetivo poder que pequeños restaurantes se conecten con los usuarios de la aplicación, debemos tener en consideración que vamos a necesitar información de ambos, como nombre, rut, etc. Además, el restaurante debe ser administrado por su dueño, el cual puede manejar su restaurante a través de la aplicación, pudiendo controlar a través de un menú de opciones que tendrá en su panel cosas como promociones, platos, eventos, etc.

El usuario podrá calificar y contar su experiencia con el restaurante tanto dentro como fuera de la aplicación. Cabe destacar que la aplicación busca también aumentar el consumo de los clientes dentro de la aplicación otorgándoles descuentos únicos para canjear en los restaurantes. También los usuarios podrán ordenar sus comidas dentro del restaurante a través de la aplicación y solicitar retirarlo en el local o una entrega a domicilio. Por último, el usuario podrá elegir reservar una mesa dentro del restaurante.

Es muy importante saber que el dueño de un restaurante tiene su nombre y su RUT para poder identificarlo y que pueda ingresar a su panel con las opciones que puede hacer el dueño dentro de la app en su restaurante. Es muy importante tener un panel de opciones para que el ambiente sea amigable para crear, modificar, actualizar y eliminar cosas dentro del restaurante de una manera efectiva.

Restaurante es uno de nuestros dos pilares más robustos de nuestra aplicación, necesitamos saber su ubicación para poder mostrarles a los usuarios sus locales más cercanos, como también su RUT para identificar que sea único, su fono para que los usuarios puedan llamar y realizar sus consultas al local y también su nombre para que se pueda mostrarle al usuario dentro de la aplicación y sepa como se llama su restaurante que va a visitar. Además, el restaurante podrá tener promociones de sus platos con una descripción donde diga el porqué de la promoción y el descuento que se le aplicará al plato en sí. También tendrán un menú con un listado de todos los platos que vengan en ese menú y nos interesa tener identificado el menú para saber qué platos incluyen de una manera rápida. El restaurante tendrá las mesas que tienen en el local que nos interesa guardar con un número incremental para poder saber cuántas mesas tiene y poder asignarles mesas a sus usuarios. Por último, el restaurante tendrá eventos que podrán agregar, en donde nos interesa saber el artista que irá al restaurante para que el usuario pueda saber que artista irá al evento, además de saber la fecha y hora en que será el evento para que el usuario se informe bien de cuándo será el suceso.

Usuario es nuestro primer pilar más fuerte ya que requerimos a usuarios para que tanto la aplicación como los restaurantes puedan existir, por ende requerimos identificar al usuario con su nombre, RUT, su celular para que se pueda llamar al usuario cuando se le necesite en algo (entregas de domicilio, error en la orden, etc); status dentro de la aplicación

para que los restaurantes logren detectar con prontitud qué tipo de usuario está en su local o solicitando una orden y puntos que se van agregando a través de las compras que va realizando dentro de la app. Cabe destacar que el usuario puede agregar y contar sus experiencias en las redes sociales, dónde nos facilita llegar a más público, detectar qué aplicación es la que más se utiliza en el momento para poder publicitar nuestra aplicación en la red social HIT del momento. Además podrá calificar al restaurante dando una feedback para así los demás usuarios puedan elegir de una mejor manera su destino y compra. También el usuario podrá ordenar platos en un restaurante y solicitar el retiro en el local o domicilio, ésta orden tendrá una PK única incremental para que podamos encontrar de una manera oportuna cualquier orden en caso de que ocurra alguna circunstancia en donde la necesitemos observar, además de tener el valor total a pagar, los platos que incluye y la fecha en que se pidió. También tendrá descuentos en los restaurantes que ha visitado, en donde tendrá fecha límite para aplicar el cupón y tendrá también una identificación para saber qué cupón gastan los usuarios. Por último, el usuario podrá reservar mesas de un restaurante dentro de la aplicación, en donde estará la fecha en la que asistirá, su respectiva hora en la que acudirá y una id para poder detectar de una manera oportuna e informarle al usuario ante algún error que pueda existir en el momento.

Platos tendrá una descripción para poder saber qué incluye el plato, su respectivo precio para que el usuario sepa lo que va a consumir, una id para identificar el plato y una lista de fotos del plato para que el usuario pueda observar de una mejor manera qué es lo que va a consumir.

1.2. Identificación de las necesidades

1. Un usuario al ingresar a la aplicación podrá visualizar a los restaurantes cercanos y mejor valorados según los usuarios que visitaron el restaurante anteriormente otorgándoles a los restaurantes una mejor visibilidad.
2. El usuario podrá realizar pedidos a los restaurantes desde la aplicación, visualizando los platos con sus fotos y descripciones, podrá realizar el pedido con retiro en local o solicitar la entrega a domicilio.
3. El usuario podrá dar un feedback al restaurante visitado calificandolo y realizando comentarios según su experiencia.
4. El dueño o administrador del restaurante podrá manejar un panel de control, con el cual podrá modificar la información de su restaurante, como por ejemplo: los menús, promociones, reservas, etc.
5. Los usuarios podrán reservar mesas en los restaurantes y además podrán obtener información sobre eventos, promociones y artistas que estarán presentes el día de su reservación.



Imagen 1: *Prototipo interfaz*

[Prototipo de APK en PDF](#)



2. Diseño conceptual

2.1. Lista Entidades y atributos

Entidad	Atributo	Metadato descriptivo	Metadato estructural
Restaurante: Entidad que almacena los datos del restaurante.	Rut	Identificador único.	Char (10) con estructura: dddddddd-{d/k}
	Nombre	Nombre del restaurante.	varchar(50) Sólo caracteres alfabéticos
	Fono	Número el cual sirve de contacto.	varchar(15) Solo dígitos con el formato: país+área+nro
	Url	Dirección web para acceder a la página del local.	varchar(255) String separado por el subdominio y dominio
Dueño: Entidad que almacena los datos del dueño que administra el restaurante.	Rut	Identificador único del dueño.	Char (10) con estructura: dddddddd-{d/k}
	Nombres	Nombre del dueño.	varchar(35) Sólo caracteres alfabéticos
	Apellidos	Apellidos del dueño	varchar(35) Sólo caracteres alfabéticos

Ubicación: Entidad que almacena la ubicación o ubicaciones del restaurante.	Latitud	Ubicación de norte o sur.	decimal(8,6)
	Longitud	Distancia en grados de la ubicación.	decimal(9,6)
Usuario: Entidad que almacena los datos del usuario que hará uso de la aplicación “SaborViña”.	Rut	Identificador único para cada ciudadano residente en Chile.	Char (10) con estructura: dddddddd-{d/k}
	Nombres	Apellidos de usuario.	varchar(35) Sólo caracteres alfabéticos
	Apellidos	Nombres de usuario.	varchar(35) Sólo caracteres alfabéticos
	Fono	Número telefónico para comunicarse con este.	varchar(15) Solo dígitos con el formato: país+área+nro
	Status	Historial de seriedad en las compras concretadas.	varchar(15) Sólo caracteres alfabéticos string
	PuntosLealtad	Valor el cual nos indica los puntos que tiene para canjear descuentos.	int(6)
Plato: Entidad que almacena los datos y descripción del plato de comida que ofrece el restaurante.	Descripción	Cadena de caracteres para describir alimentos a la venta. Incluye la lista de ingredientes que contiene el plato	varchar(200)

	Nombre	Nombre representativo del plato	varchar(25)
	Precio	Valor que tendrá el plato de comida	int(6)
	ID	Identificador único que tendrá el plato	int(4)
Foto: Entidad que almacena la imagen del plato de comida.	Imagen	Archivo el cual contiene.	Blob
Evento: Entidad que contiene los datos del evento a realizar en el restaurante. (En caso que esté programado algún evento).	Artista	Persona que viene a prender la fiesta.	varchar(20)
	Fecha	Día y hora en el cual se realiza el evento.	datetime(0)
Reserva: Entidad que almacena los datos de la reserva realizada por el usuario.	IDreserva	Id único de reserva hecha por cliente.	int(5)
	FechaReservada	Fecha y hora en la cual el usuario asistirá al restaurante.	datetime(0)
	FechaSolicitada	Fecha y hora en la cual se realizó la solicitud de reserva.	datetime(0)
Mesa: Entidad que almacena los datos de la mesa en el restaurante que fue reservada por el usuario.	Número	Número identificador de la mesa.	int(3)
	Ubicación	Posición dentro del restaurante donde se encontrará la mesa (Terraza, en el interior, cerca de la ventana, etc)	varchar(25)
	Capacidad	Máximo de usuarios que pueden ocupar la mesa.	int (2)

Calificación: Entidad que almacena los datos de la calificación dada por el usuario que asistió al restaurante.	Estrellas	Métrica usada para determinar la calidad del restaurante.	varchar(1)
	Comentario	Describe la experiencia que tuvo el usuario.	varchar(50)
	NombreRedSocial	Nombre de la red social en la cual se realizará la calificación.	varchar(25)
Orden: Entidad que almacena los datos de la orden realizada por el usuario.	IDorden	Identificador único de la orden de los cliente.	int(8)
	LocalDomicilio	Permite al cliente seleccionar si quiere su orden enviada a domicilio, o retirarla en el restaurante.	int(1) 1:Local y 2:Domicilio.
	Estado	Estado de la orden (entregado, en proceso, cancelado,etc)	varchar(45)
	Fecha	Día y hora en el que se realiza la orden.	datetime(0)
	Valor	Costo total de la orden.	int(7)
Promoción: Entidad que contiene los datos de la promoción del restaurante (en caso de que exista), la cual se aplica a un plato.	Descuento	Valor el cual será descontado con este beneficio.	int(6)
	Descripción	Explica para qué es aplicable la promoción, con sus respectivas restricciones.	varchar(100)

	ID	Número identificador para la promoción	int(5)
Descuento: Entidad que almacena los datos del descuento que puede o no tener un usuario para el restaurante.	Cupón	Código el cual da una rebaja al producto aplicable	varchar(15)
	FechaVencimiento	Valor numérico el cual dice la fecha límite antes de que este ya no pueda surtir efecto en la compra.	date(0)
Orden_tiene_Platos:	Cantidad	Número de platos de un mismo tipo en la orden	int(3)
	Rebaja	Precio total de rebaja aplicada a el/los plato/s del mismo tipo.	int(5)

Tabla 1: *Descripción de entidades del modelo de datos*

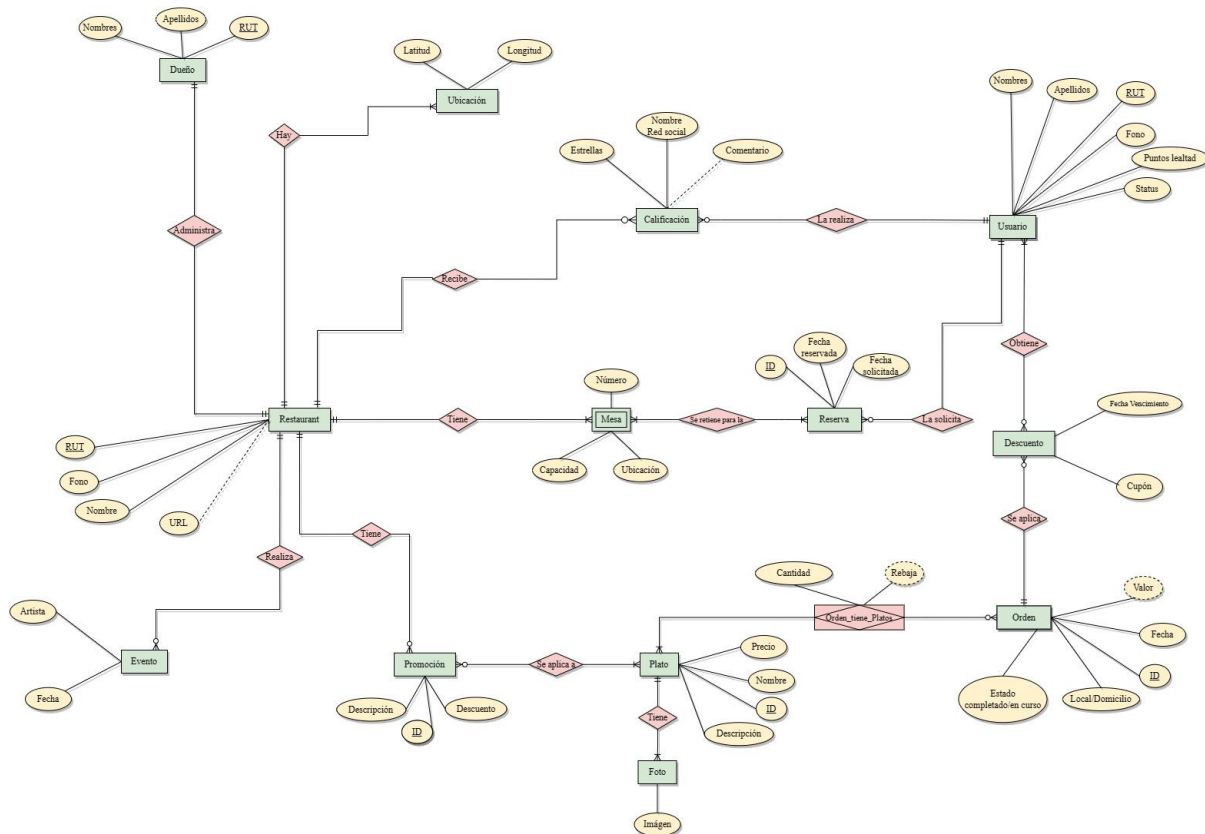
2.2. Descripción de las asociaciones

Nº	Nombre	Descripción
1.	Dueño_Restaurant	Asociación de uno obligatorio a uno obligatorio, un dueño es el propietario de un restaurante y el local puede tener un único dueño.
2.	Restaurant_Evento	Asociación de uno obligatorio a muchos opcional, un restaurante puede realizar muchos o ningún evento y que ese mismo evento no se puede realizar en muchos restaurantes a la vez.
3.	Ubicación_Restaurant	Asociación de uno obligatorio a uno mandatario, indica que en esa ubicación solo se encontrará un restaurante y el restaurante solo tendrá una ubicación posible.
4.	Restaurant_Promoción	Asociación de uno obligatorio a muchos opcional, esto indica que un restaurante puede tener ninguna o muchas promociones y ésta sólo será de ese restaurante.
5.	Restaurant_Calificación	Asociación de uno obligatorio a muchos opcional, un restaurante puede tener muchas o ninguna calificación y esa calificación es para un único restaurante.
6.	Restaurant_Mesa	Asociación de uno obligatorio a muchos obligatorio, un restaurante puede tener una o muchas mesas y esas mesas corresponden a un único restaurante.
7.	Promocion_Plato	Asociación de muchos opcional a muchos obligatorio, la/las promoción/nes puede/n aplicarse en uno o muchos platos y a su vez, muchos platos pueden estar en una o muchas promociones.

9.	Calificación_Usuario	Asociación de muchos opcional a uno obligatorio, una calificación la realiza un usuario, pero un usuario puede realizar o no muchas calificaciones.
10.	Usuario_Descuento	Asociación de muchos obligatorio a muchos opcional, un descuento puede ser para cero o muchos usuarios, por otro lado, un usuario puede tener cero o muchos descuentos.
11.	Mesa_Reserva	Asociación de muchos obligatorio a uno obligatorio, una mesa se podrá reservar una vez por fecha, también, una reserva puede reservar una o muchas mesas.
12.	Reserva_Usuario	Asociación de muchos opcional a uno obligatorio, una reserva la solicita un usuario pero este usuario puede o no reservar muchas veces.
13.	Descuento_Orden	Asociación de muchos opcional a uno obligatorio, un descuento se puede aplicar a solo una orden y una orden puede tener ninguno o muchos descuentos.
14.	Plato_Orden	Asociación de muchos obligatorio a muchos opcional, un plato puede estar en muchas o ninguna orden realizada, y una orden puede tener uno o muchos platos.
15.	Plato_Foto	Asociación de uno obligatorio a muchos obligatorio, un plato puede tener una o muchas fotos y una foto le pertenece a un único plato.

Tabla 2: Asociaciones

2.3. Modelo Entidad-Relación



Para mayor información adjuntamos un github en donde se encontrarán pdfs con las imágenes presentadas para que puedan visualizarlas con una mejor definición, se recomienda que los descarguen para poder realizar aumento en éstos pdfs y ver con claridad las características que tienen las imágenes.

https://github.com/DorianICG/BD_1S_2023.git