

Gestión de Datos:

Trabajo Práctico Delivery

Alumno	Legajo
Camila Albite	1642168
Juan Ignacio Ferro	1677743

Trabajo Práctico Cuatrimestral

-2023-

DER

“LocalRegistrado” esta relacionado con muchos “Pedido” ya que un mismo local puede atender varios pedidos, sin embargo cada “Pedido” recibe una FK del “id_local” para saber a que local pertenece.

“Pedido” esta relacionado uno a muchos con “EstadoPedido”, ya que de los estados que puede adoptar son varios para un mismo pedido (Aceptado, Recibido, etc.), a su vez tiene una Foreign Key que proviene de “Pedido” ya que tiene un Id del pedido asociado al estado. Lo mismo sucede con “Envio” y “EstadoEnvio”.

Un “Pedido” esta relacionado con un unico “Envio” ya que un mismo pedido no puede tener varios envios, a su vez “Pedido” tiene una FK perteneciente de “Envio” para indicar que nro de envio esta relacionado con que pedido.

Un “Envio” esta relacionado con un unico “Repartidor” que se encargara de entregar el pedido, a su vez “Envio” tiene una Foreign Key (FK) asociada a “Repartidor” para saber quien realizara el envio.

Un “Envio” puede ser un “ServicioMensajeria”, por esto estan relacionados uno a uno, a su vez “ServicioMensajeria” tiene una FK asociada al envio que es el “nro_envio” que esta vinculado con el envio a realizar.

El “ServicioMensajeria” esta relacionado con un unico “Paquete” que es obligatorio que este para este servicio.

Varios “ServicioMensajeria” pueden ser pagados con un mismo “MedioDePago” ya que se pueden solicitar varios de estos servicios y abonarlos de la misma manera.

Un “MedioDePago” puede tener varias “Tarjeta” asociadas.

Un “MedioDePago” puede estar vinculado con varios “Pedido”, el cual tiene una FK que define el medio de pago a utilizar en el pedido.

Un “Usuario” puede tener muchos “ServicioMensajeria” asociados, ya que puede solicitar esta servicio muchas veces.

“Usuario” esta relacionado con muchos “MedioDePago” ya que este puede optar por pagar con diferentes medios de pago.

Un “Usuario” esta relacionado con muchos “Pedido” ya que este puede solicitar muchos pedidos, a su vez “Pedido” posee una FK asociada al usuario que permite saber que usuario solicito cada pedido.

Un “Usuario” esta relacionado con una “DireccionUsuario” ya que este posee una unica direccion, a su vez “Usuario” recibe una FK de “DireccionUsuario” que permite saber la direccion del usuario.

“Usuario” esta relacionado con muchos “Reclamo” ya que puede realizar muchos reclamos, a su vez “Reclamo” posee una FK proveniente de “Usuario” para saber que usuario realizo el reclamo.

“Usuario” puede estar vinculado a muchos “Cupon” ya que puede haber muchos cupones que pertenezcan a un mismo usuario, a su vez “Cupon” recibe una FK de “Usuario” el cual vincula el cupon con el usuario.

“Reclamo” puede tener muchos “OperadorReclamo” ya que puede ir rotando quien atiende un mismo reclamo.

Un “Reclamo” puede estar relacionado un “Cupon” en el caso de que se haga un reclamo de algun cupon.

“Cupon” esta relaciondado con “CuponReclamo”, que se efectua cuando hay un reclamo por un cupon, tambien recibe una FK de “Cupon” para saber a que usuario esta vinculado el cupon.

Un “Reclamo” esta relacionado con un “CuponReclamo” y verifica que el usuario que posee el cupon sea el mismo que era el que lo reclamo, se verifica la FK de “usuario_dni” que estan tanto en “Reclamo” como en “CuponReclamo”.

Un “Pedido” puede estar relacionado con muchos “CuponPedido”, el cual recibe una FK de “Pedido” en donde esta el “id_pedido”

Un “Cupon” esta puede estar relacionado con un “CuponPedido” en el caso que se necesite.

Un “Pedido” puede estar relacionado con un “Reclamo” en el caso que se necesite.

Decisiones script

En el procedimiento “migrar_usuario” habia que especificar que el DNI, nombre, apellido y email no sean “null” ya que definimos en la tabla que no podia ser de esta manera porque seria un usuario sin Primary Key (PK) en el caso del DNI y porque si posee null el resto de campos consideramos que no seria un usuario valido. a su vez, en ese mismo procedimiento decidimos ordenar a los usuario por su DNI.

Algo similar pasa con el procedimiento “migrar_repartidor” con “migrar_usuario” ya que consideramos que si no tiene DNI, nombre, apellido y mail no iba a ser un Repartidor valido.

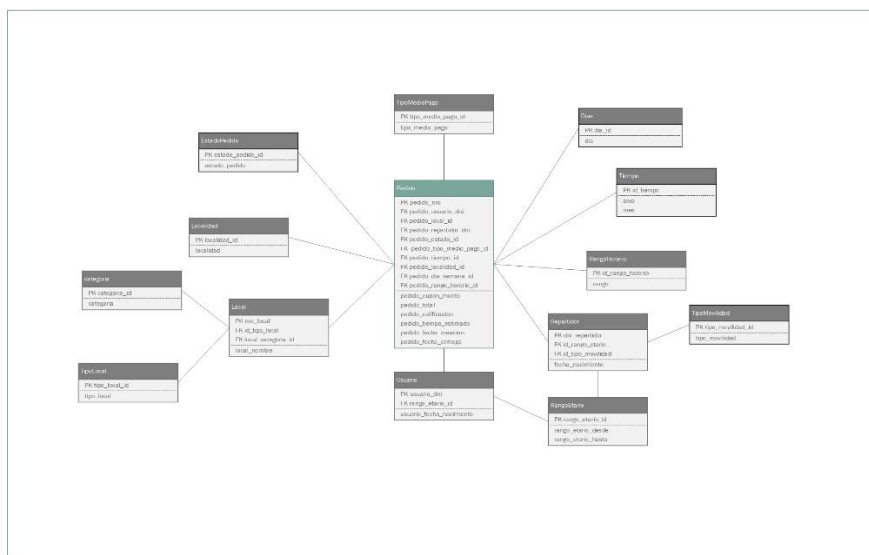
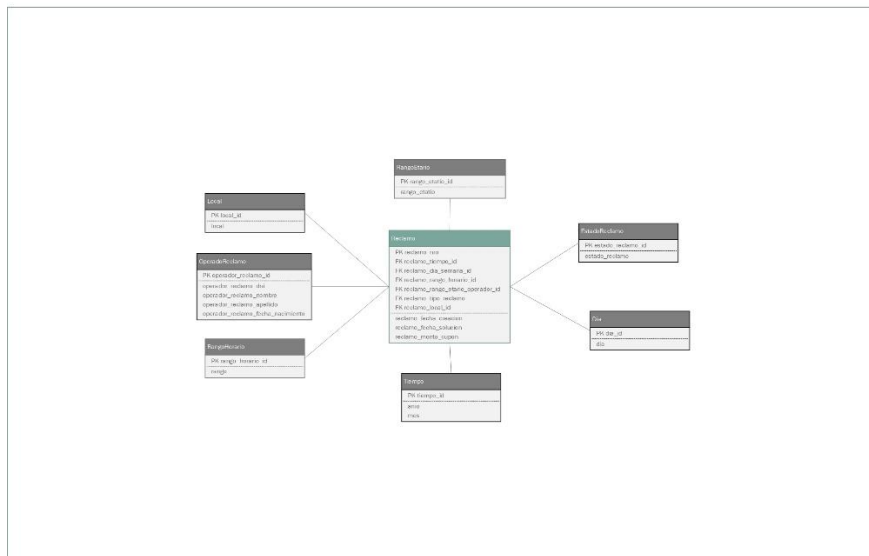
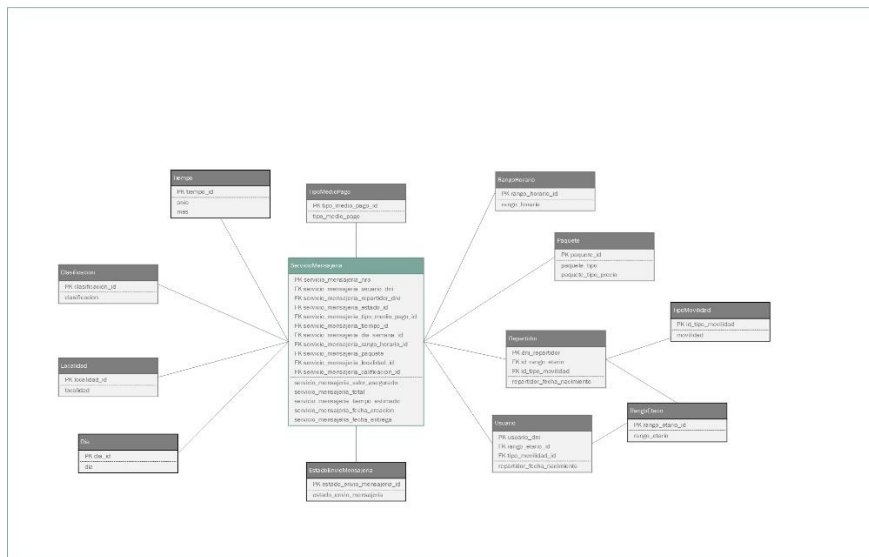
En el procedimiento “migrar_local” tuvimos que indicar que “local_nombre” no sea null ya que era la PK. Tambien utilizamos select distinct para obtener las tablas porque consideramos que no podia haber locales con un mismo id.

En el procedimiento “migrar_direccion_local” sucede lo mismo con “migrar_local” donde “local_direccion” es la PK.

En el procedimiento "migrar_atencion_local" sucede lo mismo con "migrar_local" donde "horario_local_dia" es la PK.

El resto de procedures son simplemente hacer un insert into de las tablas de la base de datos dada (Tabla Maestra) a las tablas que creamos nosotros.

Decisiones Modelo BI



Las tablas de hechos que decidimos hacer son “Reclamo”, “Pedido” y “ServicioMensajería” que son las que consideramos necesarias para satisfacer las vistas requeridas en el TP.

Por un lado, en la tabla Pedido están contenidas las FK necesarias para poder acceder a la información del usuario que realiza el pedido, el local al que se hizo el pedido, el repartidor encargado del envío, el estado del pedido, el medio de pago utilizado, la fecha en que se registró el pedido precisando el día de la semana y el rango horario para poder utilizar esa información en las vistas implementadas posteriormente.

Por otro lado, en la tabla Reclamo se encuentran las FK requeridas para tener acceso a información sobre el tipo de reclamo, el local al cual se realizó el pedido que presento problemas, el repartidor asignado al mismo, el día de la semana y el rango horario en el que se realizó el reclamo para poder agrupar los reclamos usando esa información como criterio.

Además, agregamos la información necesaria relacionada con la fecha y la hora en la que se realizó el reclamo para poder utilizarla en la creación de las vistas.

Por último, en la tabla ServicioMensajería consideramos necesarias las FK relacionadas con el usuario que solicita el envío, el repartidor asignado, el paquete involucrado en el envío, el estado del envío, el día de la semana y la franja horaria en la que fue realizado el envío, la localidad dentro de la cual se hizo el envío y la calificación que el usuario le asignó a ese envío para poder contar con la información suficiente para constituir las vistas.

Además, agregamos información sobre los el tiempo de entrega del paquete y los montos arancelados por el usuario ya que era información que necesitábamos utilizar en las vistas.

Para la vista que tiene como objetivo listar el porcentaje de pedidos y mensajería entregados mensualmente según el rango etario de los repartidores y la localidad consideramos necesaria la creación de 3 vistas, enumerando los pedidos entregados en una, los totales en otra y por último la vista que consigue cumplir el objetivo. Esta separación favorece la performance, y hace la consulta final mucho más rápida.

Teniendo en cuenta que no existían categorías en la tabla maestra optamos por insertar algunas de ejemplo para simular el funcionamiento.

Se tomó la decisión de crear dos funciones necesarias para resolver rangos horarios y rangos etarios: `getRangoHorario()` y `getRangoEtario()`. Les encontramos mucha utilidad ya que en varios puntos esto se pide como dimensión de análisis.

Ambas reciben un parámetro de tipo `datetime` y devuelven un rango horario o etario respectivamente.

En la vista `BI_V_MAYOR_CANTIDAD_PEDIDOS` hicimos uso de particiones para poder determinar los rangos y días de la semana con mayor cantidad de pedidos. No es la única forma de resolverlo, pero consideramos que fue la más performante.