

Estrategia

TP Gestión de Datos 1C 2023

Grupo: MargeCreoQueOdioGDD

Número: 24

Curso: K3051

Integrantes

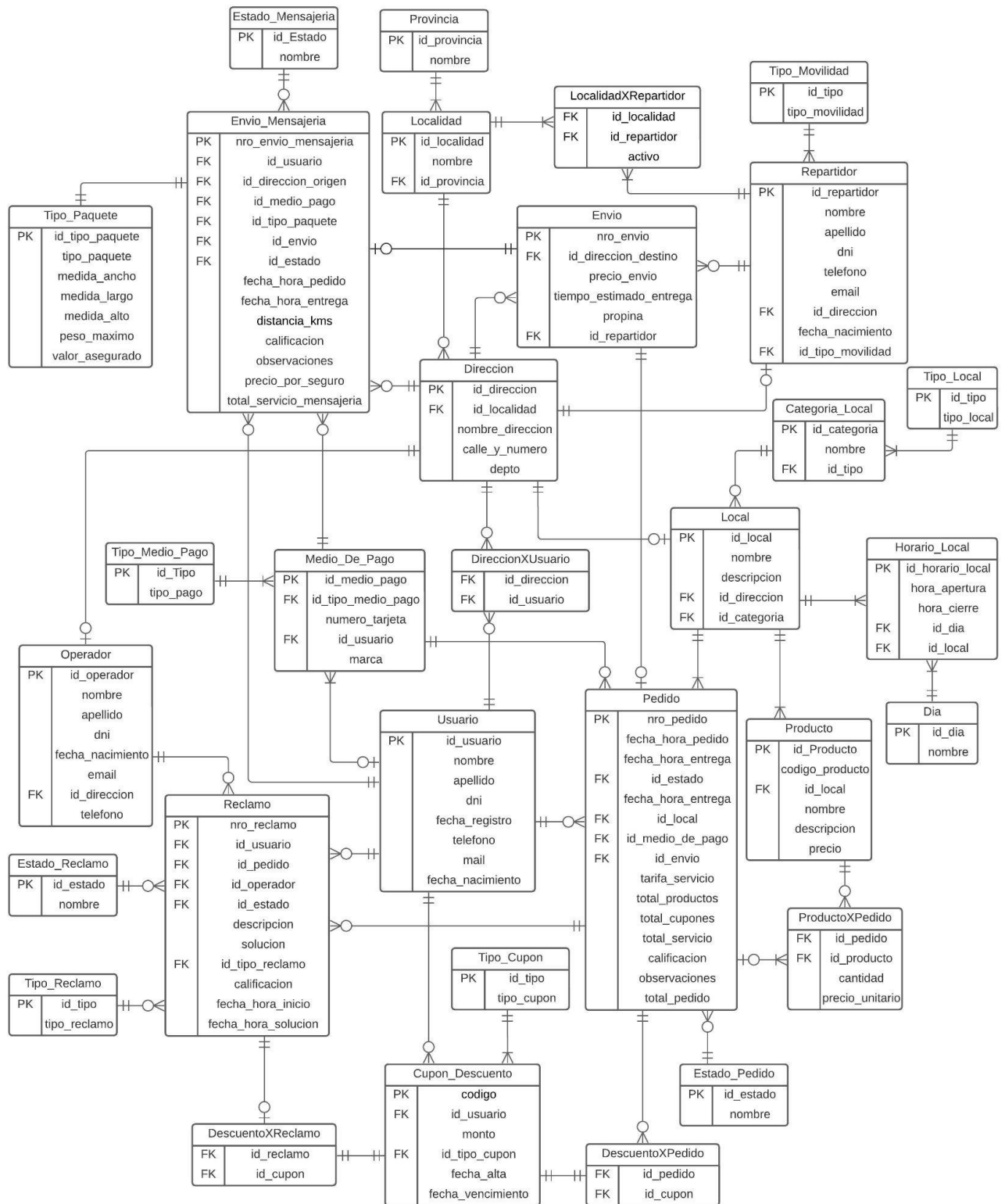
- Antonella Bevilacqua (163.197-4)
- Camila Garcia Federico (168.258-1)

Índice

Modelo de datos.....	4
Estrategias.....	5
Tablas.....	7
1. Usuario.....	7
2. Local.....	7
3. Pedido.....	8
4. Cupón de Descuento.....	9
5. Medio de Pago.....	10
6. Envío.....	10
7. Repartidor.....	10
8. Dirección.....	11
9. Envio Mensajería.....	11
10. Reclamo.....	12
Migración.....	13
Modelo BI.....	14
Estrategias.....	15
Tablas.....	15
1. Estado Mensajería, Estado Reclamo, Estado Pedido.....	15
2. Tipo Movilidad, Tipo Medio Pago.....	16
3. Categoría_Local.....	16
4. Local.....	16
5. Envío.....	16
6. Localidad.....	16
7. Tipo_Paquete.....	17
8. Tiempo.....	17
9. Día.....	17
10. Rango Etario, Rango Horario.....	17
11. Cupon_Descuento.....	17
12. Envio_Mensajería.....	18
13. Pedido.....	18
14. Reclamo.....	19
Funciones.....	20

Entrega 1: DER del modelo migrado

Modelo de datos



Estrategias

Se crearon Primary Keys para entidades que no disponían de una clave candidata.

En las siguientes entidades, se crearon con un IDENTITY de tipo INT seteadas en 1 y sus saltos en 1 también:

- Pedido
- Reclamo
- Envio_Mensajeria
- Usuario
- Operador
- Repartidor
- Cupon_Descuento
- Medio_De_Pago
- Dirección
- Localidad
- Provincia
- Paquete
- Tipo_Paquete
- Local
- Tipo_Local
- Categoría_Local
- Horario_Local
- Dia
- Estado_Pedido
- Estado_Reclamo
- Estado_Mensajeria
- Tipo_Medio_Pago
- Tipo_Reclamo
- Tipo_Movilidad
- Tipo_Cupon
- Producto

En otras entidades, se consideraron como claves primarias campos que ya eran parte de las mismas:

- Envio_Mensajeria -> número de envío
- Pedido -> número de pedido
- Envío -> número de envío

- Reclamo -> número de reclamo
- Cupon_Descuento -> código

Tablas

Nuestro DER está compuesto por las siguientes tablas:

1. *Usuario*

Se creó esta tabla para almacenar información sobre los clientes que utilizan la app de Delivery Online, la misma contiene información personal del usuario, como su nombre, apellido, DNI, fecha de nacimiento, teléfono, correo electrónico, fecha de registro y dirección, así como información adicional como cupones de descuento que se hayan otorgado al usuario.

Esta tabla es necesaria para la realización de diversas operaciones y transacciones que los clientes pueden realizar, como hacer pedidos, aplicar descuentos disponibles en su cuenta, realizar reclamos sobre los pedidos, solicitar envíos de mensajería, etc. Por ejemplo, cuando un usuario realiza un pedido, la tabla **Pedido** se vincula con la tabla **Usuario** a través de una FK para que se pueda relacionar el pedido con el cliente que lo realizó. Además, si el usuario tiene un cupón de descuento disponible, se relaciona con la tabla **Cupon_Descuento** a través de una FK para aplicar el descuento al pedido que se requiera.

DireccionXUsuario: En esta tabla se va a guardar una referencia a la tabla dirección y otra referencia al usuario. Esto es debido a que consideramos que además de que un usuario puede tener muchas direcciones guardadas, éstas direcciones podrían ser las mismas que otros usuarios tengan (por ejemplo, 2 usuarios o más que viven juntos), entonces para resolver la relación muchos a muchos entre Usuario y Dirección creamos esta tabla intermedia. Además, la tabla dirección no podría guardar una referencia al usuario ya que usamos la misma tabla para otros tipos de direcciones como las de repartidores, operadores, locales, etc.

2. *Local*

Será la tabla donde estarán todos los locales dados de alta en el sistema. El campo id_Direccion representa la dirección física del local la cual será una FK a la tabla Dirección.

Horario Local: En esta tabla se van a guardar los días y horarios de apertura y de cierre.

Dia: Se crea esta tabla para guardar el nombre del día.

HorarioXDia: Esta tabla se crea para romper la relación muchos a muchos de la tabla Dia y la tabla Horario_Local, de esta forma obtenemos que día está abierto y a qué horario.

Tipo: Esta tabla se creó para separar los dos tipos de locales que existen en el sistema, en este caso serán comida y mercado. Tiene una relación 1 a N con Local, ya que cada tipo de local puede tener varios locales asociados, pero cada local sólo pertenece a un único tipo de local.

Cada uno de estos tipos cuenta con determinadas categorías asociadas, es decir, estos dos tipos no necesariamente comparten categorías, por eso también se creó la tabla *Categoría*. Con este modelo, se garantiza que un local de tipo "mercado" sólo pueda tener categorías asociadas que estén en la tabla Categoría y que tengan un tipo de local asociado que sea "mercado" y lo mismo ocurre para los locales de tipo "comida". De esta forma se evita la posibilidad de que un local tenga categorías que no correspondan a su tipo.

Categoría: Guardará el nombre de la categoría que se dé de alta.

3. *Pedido*

Esta tabla almacena información sobre los pedidos realizados por los clientes, desde la fecha y hora de creación hasta el estado actual del pedido, la ubicación del local, el método de envío y el costo del servicio de entrega. El campo *estado* se creó para almacenar el estado actual del pedido, que puede ser "pendiente", "en preparación", "en camino", "entregado", "cancelado", etc.

ProductoXPedido: Dado que un pedido puede contener varios productos y un producto puede estar presente en varios pedidos, se creó esta tabla intermedia que contiene información específica de cada producto dentro de un pedido. Contiene una clave primaria compuesta por las claves foráneas de las tablas **Pedido** y "Producto", lo que asegura la unicidad de cada registro en la misma. Además, se agregaron los atributos *cantidad*, *precio_unitario* y *total_producto* para almacenar información relevante de la cantidad de cada producto en un pedido, el precio unitario y el total de ese producto en ese pedido.

Producto: Es una tabla que almacena información sobre los productos disponibles para la venta en el local, incluyendo campos como su nombre, descripción y precio. El campo precio define el precio por el producto específico. Como un local tiene muchos productos se estableció la relación uno a muchos desde Local a Producto.

Estado_Pedido: Esta tabla guarda información sobre los distintos estados en que puede estar un pedido desde que se realiza hasta que se entrega (realizado, confirmado por el local, en preparación, en camino, entregado, cancelado, etc).

EstadoXPedido: Dado que un pedido en particular puede tener varios estados desde que se realiza hasta que se entrega, y un estado en particular puede estar en muchos pedidos, se creó esta tabla. Esta tabla guarda información específica sobre el estado actual de un pedido en particular.

⇒ **IMPORTANTE**: Decidimos no cumplir con la Primera Forma Normal que indica que no se deben almacenar campos calculables en el caso del campo *total_pedido* ya que consideramos que ante un gran volúmen de datos calcularlo es poco performante.

4. *Cupón de Descuento*

Esta tabla se creó para almacenar la información sobre los cupones de descuento que la empresa le da a los usuarios, ya sean debido a un reclamo o por un simple regalo. Los atributos de esta tabla incluyen: *código* que es la PK de esta tabla, el *monto*, el *id_usuario* del usuario al que le pertenece, el *id_pedido* que guarda el número de pedido en el cual se usó, el *id_tipo_cupon* que es una referencia a la tabla Tipo_Cupon. Por otro lado, tiene *fecha_alta* que es la fecha en que se da de alta este cupón y *fecha_vencimiento* que es la fecha límite hasta que el usuario puede usarlo.

Tipo_Cupon: Esta tabla guarda información sobre el tipo de cupón que puede ser “Bienvenida”, “Cliente”, “Frecuente”, “Referido”, “Inactividad”, “Devolución”, etc.

DescuentoXPedido: Esta tabla guarda información sobre un descuento aplicado a un pedido debido a un cupón de descuento.

DescuentoXReclamo: Esta tabla guarda información sobre un cupón de descuento que se generó a raíz de un reclamo.

5. Medio de Pago

Esta tabla se creó para almacenar información sobre los distintos medios de pago que se pueden utilizar para realizar los pagos de los pedidos. Los atributos de esta tabla incluyen una PK *id_medio_pago*, *id_tipo_medio_pago* que es una referencia a la tabla Tipo_Medio_Pago que indica el tipo de medio de pago utilizado (tarjeta de crédito, de débito o efectivo), *nro_tarjeta* que almacena el número de tarjeta de crédito o débito y *marca_tarjeta* que se utilice si corresponde. La creación de esta tabla permite que los datos de pago se almacenen en una tabla separada, en lugar de estar incluidos en la tabla de Pedido o Envio Mensajería.

6. Envío

Esta tabla se creó para almacenar información tanto de los envíos que se realizan para pedidos como para los que corresponden al servicio de mensajería. Los atributos que hay en esta tabla son: *nro_envio* que es la PK, *id_direccion_destino* que guarda una referencia a la tabla Direccion y es la dirección a la cual se debe enviar el pedido y que eligió el usuario. Por otro lado, *precio_envio* que es el precio del envío que el sistema calcula en base a la distancia, *tiempo_estimado_entrega* que representan los minutos que tarda en entregarse el pedido, *propina* que haya dejado el usuario, y *id_repartidor* que guarda una referencia a la tabla Repartidor.

⇒ **IMPORTANTE**: Decidimos que tanto Pedido como Envio_Mensajería tengan una referencia a una tabla Envio para normalizar.

⇒ *precio_envio* es calculable por el sistema, nosotros no tenemos que calcularlos, por lo tanto, no cuenta como campo calculable.

7. Repartidor

Esta tabla tendrá a los repartidores que están dados de alta en el sistema. Un campo importante de esta tabla es *id_localidad*, se

establece una relación uno a muchos entre **Localidad** y **Repartidor** porque cada repartidor puede estar activo en una sola localidad y a su vez, cada localidad puede contar con varios repartidores asignados. De esta manera, se puede tener un control más preciso sobre la asignación de repartidores a envíos, ya que se puede limitar la asignación de envíos a aquellos repartidores que estén activos en la localidad donde se encuentra el envío.

Tipo Movilidad: Esta tabla guarda información sobre el tipo de movilidad que usa el repartidor.

8. **Dirección**

Se creó esta tabla para almacenar información sobre las direcciones, la misma incluye los siguientes atributos: una PK *id* que identifica de manera única cada registro, una FK *id_localidad* que indica la localidad de la dirección, el *nombre* de la dirección (casa, departamento, etc.), la *calle*, el *número*, el *código postal*, el *departamento* (en caso de que sea necesario, podría ser null) y observaciones adicionales.

Localidad: Se crea esta tabla para separar la localidad de una dirección ya que una localidad posee muchas direcciones que le pertenecen.

Provincia: Se crea esta tabla para separar la provincia de una localidad ya que una provincia posee muchas localidades que le pertenecen.

9. **Envio Mensajería**

Esta tabla almacena la información relacionada al Servicio de Mensajería que ofrece la aplicación.

Tipo Paquete: Se crea ya que en el enunciado se menciona que los tipos de paquetes están disponibles con sus medidas y capacidades máximas y están previamente configurados en el sistema.

Estado Mensajería: Esta tabla guarda información sobre los distintos tipos de estados en que puede estar el servicio de envío de mensajería (realizado, confirmado, en camino origen, en camino destino, entregado, cancelado, etc).

EstadoXMensajería: Dado que un servicio de envío de mensajería en particular puede tener varios estados desde que se realiza hasta que se entrega, y un estado en particular puede estar en muchos servicios de mensajería, se creó esta tabla. Esta tabla guarda información específica sobre el estado actual de un servicio de mensajería en particular.

⇒ **IMPORTANTE**: Decidimos no cumplir con la Primera Forma Normal que indica que no se deben almacenar campos calculables en el caso del campo *total_servicio_mensajería* ya que consideramos que ante un gran volumen de datos calcularlo es poco performante.

⇒ *precio_por_seguro* y *distancia_kms* son calculables por el sistema, nosotros no tenemos que calcularlos, por lo tanto, no cuenta como campo calculable.

10. **Reclamo**

Esta tabla almacena la información relacionada con los reclamos que pueden iniciar los usuarios con respecto a un pedido. Su PK va a ser el campo *nro_reclamo* y va a tener *id_pedido* que es el pedido por el cual se originó el reclamo, el *id_usuario* del Usuario que inició el reclamo, el *id_operador* que se asignó para atender el reclamo, el *codigo_cupon* que el operador da de alta como retribución para el usuario, entre otros.

Operador: Se crea esta tabla ya que un Reclamo tiene asignado un único Operador, pero un Operador puede atender muchos Reclamos.

Estado_Reclamo: Esta tabla guarda información sobre los distintos tipos de estados en que puede estar el reclamo desde que se inicia hasta que se finaliza (iniciado, en revisión, solucionado, etc.).

EstadoXReclamo: Dado que un reclamo en particular puede tener varios estados desde que se realiza hasta que se finaliza, y un estado en particular puede estar en muchos reclamos, se creó esta tabla. Esta tabla guarda información específica sobre el estado actual de un reclamo en particular.

Entrega 2: Migración de datos

Migración

En esta instancia se creó el script de la migración primeramente utilizando DROP para eliminar todos los datos sólo en caso de existan, se realizó DROP por cada tabla, cada stored procedure y el esquema, en nuestro caso llamado MargeCreoQueOdioGDD. Una vez realizado esto se realiza la creación del esquema y de cada tabla.

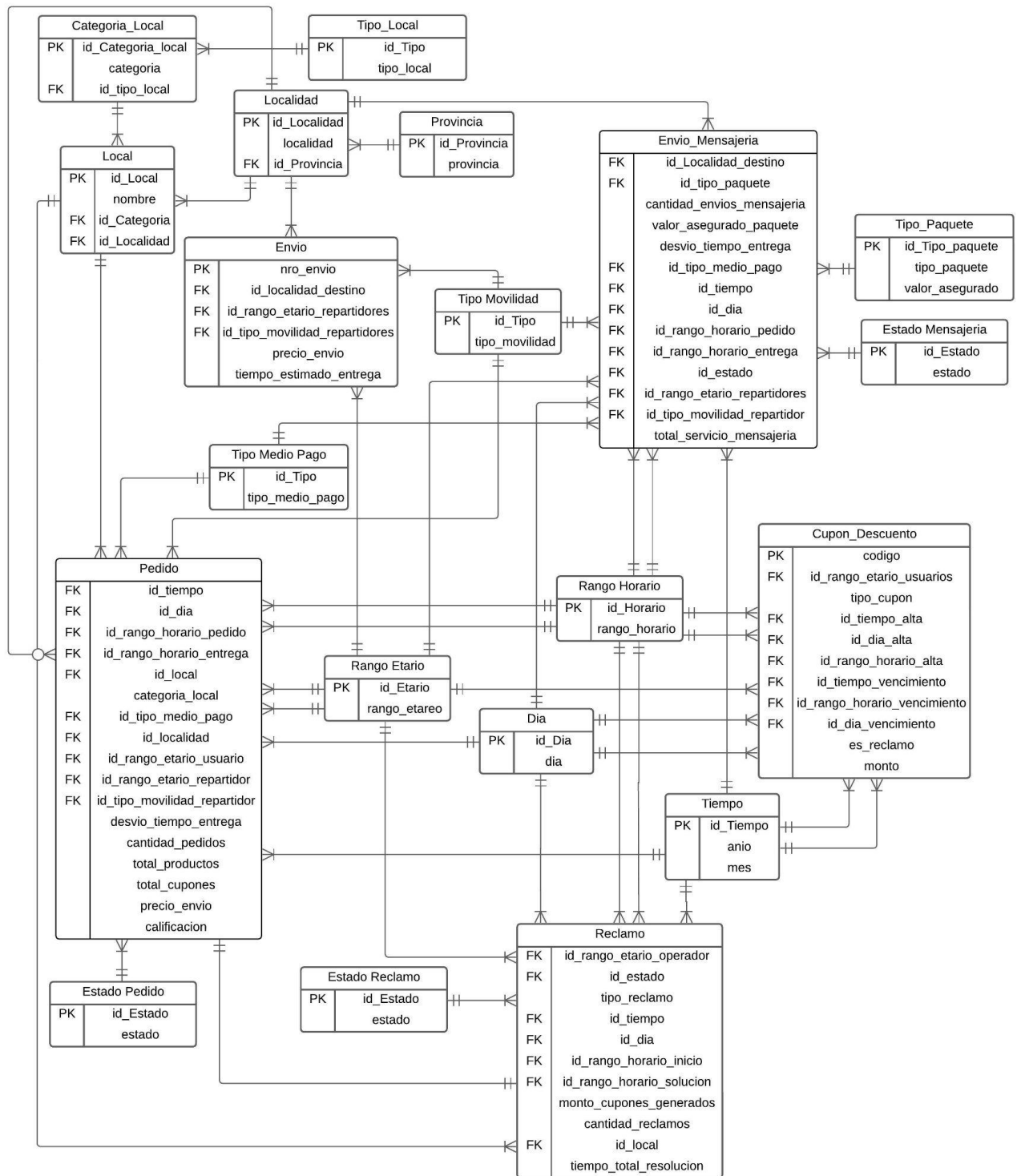
Creamos las tablas necesarias y tuvimos en cuenta respetar los tipos de datos que se presentan en la tabla maestra al realizar su migración. La mayoría de las tablas tendrá PK autoincremental salvo algunas las cuales consideramos que serían ideales para utilizarlas como PK ya que no se repetían entre sí y podría identificar cada registro de forma única. En el caso de las FK realizamos CONSTRAINTS luego de crear las tablas para poder referenciar a las tablas correspondientes. Se realizó un constraint por cada FK que se necesitó utilizando ALTER TABLE.

Luego creamos los stored procedures los cuales realizamos uno por cada tabla a migrar. Se tuvo en cuenta no migrar los NULLs y tampoco tener FK o PK en NULL. **En el caso de la categoría de los locales, se le asignó una categoría a cada local, habiendo previamente creado las categorías ya que desde la tabla maestra no se encontraron datos de esto.**

Por último se ejecutan todos los stored procedures imprimiendo un mensaje como “Se comienza a migrar ...” para poder tener claro cuando se está migrando alguna tabla.

Entrega 3: Modelo BI

DER del modelo BI



Estrategias

Para esta entrega, creamos tanto las tablas dimensionales como las tablas de hechos. Las **tablas de hechos** que nos quedaron son **BI_Pedido**, **BI_Envio_Mensajeria** y **BI_Reclamo** ya que son las tablas que contienen datos calculados o datos importantes para obtener las métricas necesarias como `tiempo_total_entrega`, `total_pedido`, `total_servicio_mensajeria`, y otros totales.

⇒ **IMPORTANTE:** Agregamos una nueva tabla `DireccionXUsuario` en el Modelo de Datos de la entrega n° 2 que es una tabla intermedia entre Usuario y Dirección ya que nos dimos cuenta que Usuario puede tener muchas direcciones guardadas, y éstas direcciones podrían ser las mismas que otros usuarios tengan (por ejemplo, 2 usuarios o más que viven juntos), entonces para resolver la relación muchos a muchos entre Usuario y Dirección creamos esta tabla intermedia.

Además, **agregamos el campo ACTIVO en la entidad**

LocalidadXRepartidor para representar la localidad en la que está activo ese repartidor, asumimos que la localidad en la que está activo es en la que el repartidor hizo su última entrega. Esto lo hacemos colocando un 1 si la localidad es la activa, y de lo contrario se coloca un 0.

Tablas

En esta instancia se crearon tablas para guardar solo los datos que necesitamos para nuestro modelo BI, a continuación se explican las tablas creadas.

Tablas dimensionales

1. Estado Mensajeria, Estado Reclamo, Estado Pedido

Estas tres tablas las utilizamos para almacenar la información sobre los estados de las tres tablas de hechos que se mencionan más adelante en este documento. Las tres guardan información acerca del estado del pedido/reclamo/envio mensajeria.

2. *Tipo Movilidad, Tipo Medio Pago*

Estas dos tablas se crean para almacenar los tipos de movilidad para los repartidores y los tipos de medio de pago con los cuales se hayan abonado los pedidos y los envíos mensajería.

3. *Categoria_Local*

Se crea esta tabla para almacenar las categorías de los locales que tenemos en nuestro sistema, teniendo esta tabla una FK que representa el id del tipo de local al cual corresponde esta categoría, por ello también creamos la tabla *Tipo_Local*, para almacenar los tipos de locales que figuran en nuestro sistema.

4. *Local*

Esta tabla se crea para almacenar los locales que figuran en nuestro sistema, tendrá dos FK: una será representativa de la categoría del local y la otra de la localidad del local.

5. *Envío*

La tabla BI_Envio se utiliza para almacenar la información relacionada con los envíos, como la localidad de destino, el repartidor asignado, el costo del envío y el tiempo estimado de entrega. El campo ID_LOCALIDAD_DESTINO representa el destino a donde se dirige el envío, ID_RANGO_ETARIO_REPARTIDORES Y ID_TIPO_MOVILIDAD_REPARTIDORES dos FK que representan: el rango etario de los repartidores y la movilidad del repartidor responsable del mismo envío respectivamente. En este enfoque, optamos por excluir el campo propina de nuestro modelo debido a que no tenemos la intención de utilizarlo para ningún propósito específico.

6. *Localidad*

Se crea esta tabla para almacenar las localidades que tenemos en nuestro sistema, teniendo esta tabla una FK que representa el id de la provincia a la cual corresponde esta localidad, por ello

también creamos la tabla *Provincia*, para almacenar las provincias que figuran en nuestro sistema.

7. *Tipo_Paquete*

Esta tabla se crea para almacenar la información que se relaciona con los paquetes de los envíos mensajería de nuestro sistema, para este modelo BI **decidimos solo guardar lo correspondiente a los campos tipo_paquete y valor_asegurado** ya que son dos campos que utilizaremos luego en nuestras vistas para mostrar la información que requiere el enunciado.

8. *Tiempo*

En esta tabla se almacenarán todos los meses y años que figuren en nuestro sistema, para migrar los datos a esta tabla se tuvo en cuenta toda fecha encontrada en nuestro modelo inicial.

9. *Dia*

Esta tabla dimensional almacena todos los días de la semana

10. *Rango Etario, Rango Horario*

En estas tablas se almacenará el rango etario mencionado en el enunciado y los rangos horarios cada dos horas respectivamente.

11. *Cupon_Descuento*

Esta tabla se crea para almacenar la información relacionada con todos los cupones de descuentos, ya sean cupones dados de alta a partir de un reclamo o no. Además, tenemos FKs a las tablas Rango_Horario, Tiempo y Dia que corresponden al tiempo de vencimiento y al tiempo de alta, por eso hay dos relaciones por cada tabla. Por otro lado, **decidimos tener un campo Reclamo de tipo NVARCHAR que nos indica mediante un string “Si” si el cupón se creó a partir de un reclamo** para poder realizar una de las vistas pedidas.

Tablas de hechos

12. Envio_Mensajeria

Esta tabla se crea para almacenar y gestionar información específica de los envíos relacionados con el servicio de mensajería, permitiendo un tratamiento diferenciado de estos envíos en comparación con los envíos de pedidos regulares. Los campos importantes de esta tabla son los que tienen que ver con el año, mes, día y rango horario los cuales son FKs a la tablas Dia, Tiempo y Rango_Horario ya que en las vistas se requiere de los mismos para consultar sobre cuando se pidió y cuando se entregó. También están los campos como las FKs a los rangos etarios de los repartidores y a los tipos de movilidad de los repartidores. Por último, los campos como **cantidad envíos mensajeria**, **valor asegurado paquete**, **desvío en tiempo de entrega** y **total servicio mensajeria** se guardan en esta tabla haciendo pequeños cálculos para luego usarlos en las vistas.

13. Pedido

La tabla contiene campos relacionados con el año, mes, día y rango horario, los cuales son claves foráneas (FK) que se vinculan con las tablas Dia, Tiempo y Rango_Horario. Estos campos permiten conocer cuándo se realizó el pedido y cuándo se realizó la entrega. También se incluyen campos que hacen referencia a los rangos etarios de los repartidores y a los tipos de movilidad utilizados por ellos. Estas son claves foráneas (FK) que se relacionan con las tablas correspondientes y permiten tener información sobre las características de los repartidores involucrados en los pedidos. También tenemos el campo del rango etario de los usuarios que nos indicará a que rango pertenecen los usuarios que realizan los pedidos. Por otro lado, tenemos campos que calculamos como **desvío en tiempo de entrega** (este es el desvío de entrega según tipo de movilidad, día y rango horario), **cantidad pedidos** (esta es la cantidad de pedidos hechos según día, rango horario, localidad, categoría local, mes y año), **total productos** (que lo calculamos sumando todos los valores de los productos) y **total cupones** (total del monto utilizado por cupones).

14. Reclamo

Esta tabla se crea para almacenar la información relacionada con los reclamos. Guardamos todos estos campos Anio, Mes, Dia y Rango_Horario tanto para la fecha en la que se inició el reclamo como para la fecha en que se solucionó, estos datos se guardan con FKs a la tabla Tiempo, a la tabla Dia y a la tabla Rango_Horario. También tenemos los campos con FK al rango etario de los operadores de los reclamos, el estado del reclamo y una última FK a la tabla Local para saber sobre cual local se debe hacer el reclamo. Por otro lado, **creamos un campo TIEMPO_TOTAL_RESOLUCION en la tabla Reclamo que guarda la diferencia en minutos entre que se inició el reclamo y entre que se solucionó** ya que era un dato que íbamos a necesitar para las métricas, el **monto cupones generados** (este es el monto total generado en cupones según mes) y la **cantidad de reclamos** realizados.

Funciones

En esta instancia se crearon distintas funciones que nos servirán para guardar los datos en las tablas con el formato requerido para las vistas, a continuación se explican las funciones creadas.

calcularDiferenciaMinutos: Toma dos fechas con formato DATETIME y nos devuelve cuantos minutos pasaron desde la fecha de inicio hasta la fecha final, esto nos es útil cuando necesitamos saber los minutos que pasaron desde que se realizó el pedido hasta que se entregó.

edadActual: Toma una fecha de nacimiento con formato DATETIME y nos devuelve la edad en formato INT, esta función sola no se utiliza, se utiliza en conjunto con la del rango etario para, luego de obtener la edad, obtenemos el rango etario al que pertenece el usuario/repartidor/operador.

rangoEtario: Toma una edad en formato INT y nos devuelve el rango etario al que pertenece, lo decidimos poner al rango como un VARCHAR que nos muestre a qué rango etario pertenece según la edad que nos pasan por parámetro.

obtenerHora: Toma una fecha en formato DATETIME y nos devuelve solo la hora en formato INT.

rangoHorario: Toma una hora en formato INT y en base a esto nos devuelve el rango horario al que pertenece en formato VARCHAR.

diaSemana: Toma un día de la semana en inglés y lo traduce al español. Esta función nos fue necesaria para saber qué día de la semana le pertenece a una fecha (DATETIME), encontramos la función DATENAME que nos devuelve que día de la semana fué pero en inglés, por eso fue necesaria crear esta función y usarla en conjunto con esa.

mesDelAnio: Toma un INT que nos indica que mes es y devuelve en formato NVARCHAR el nombre del mes correspondiente.

restar: Toma dos valores FLOAT y devuelve en formato FLOAT el resultado positivo de la resta. Se hizo esta función para facilitar el cálculo de la diferencia de minutos entre 2 campos y no devolver ningún valor negativo ya que verifica cual es el mayor y a ese le resta el otro valor.