

Estrategia

TP Gestión de Datos 1C 2023

Grupo: MargeCreoQueOdioGDD

Número: 24

Curso: K3051

Integrantes

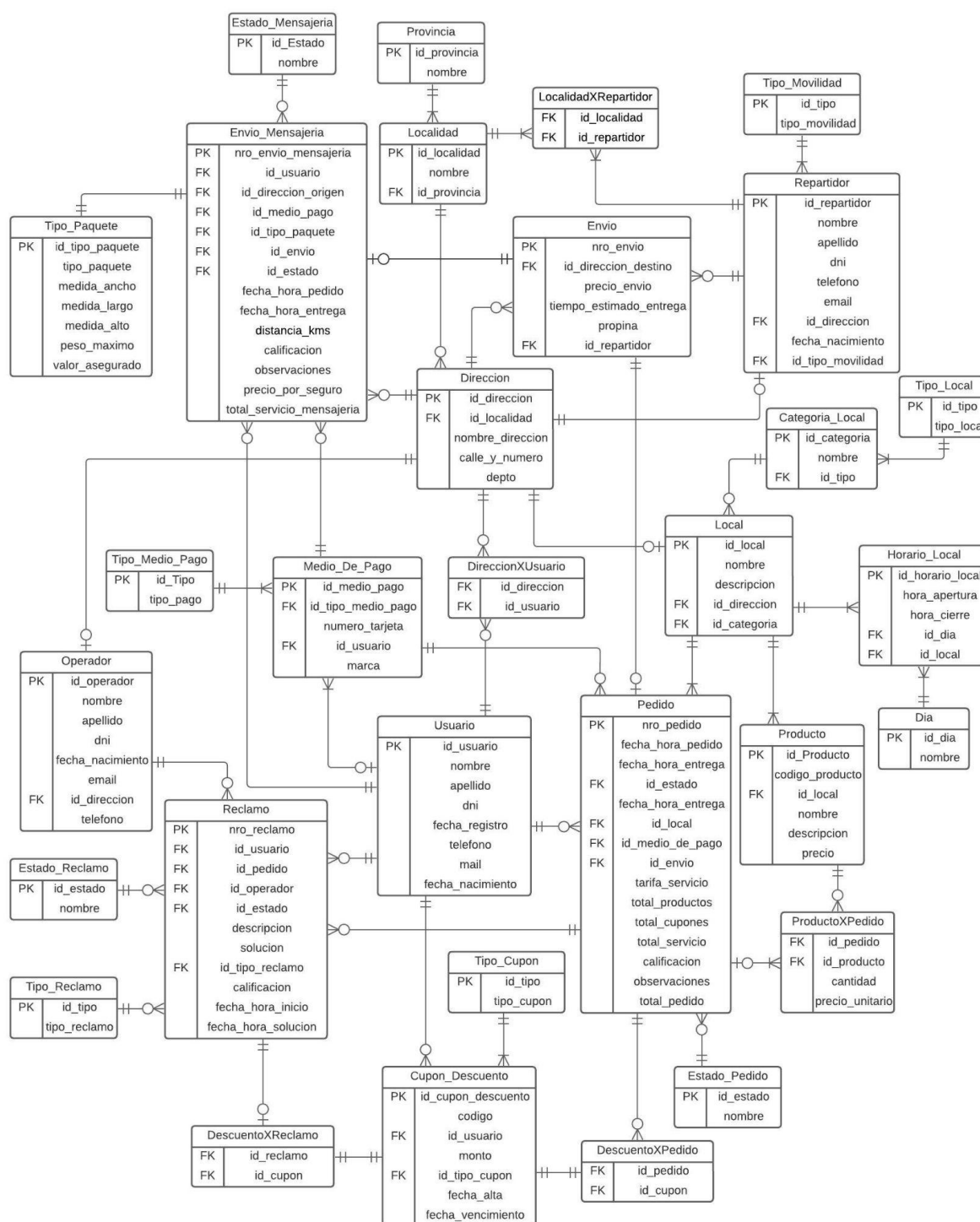
- Antonella Bevilacqua (163.197-4)
- Camila Garcia Federico (168.258-1)

Índice

Modelo de datos.....	3
Estrategias.....	4
Tablas.....	6
1. Usuario.....	6
2. Local.....	6
3. Pedido.....	7
4. Cupón de Descuento.....	8
5. Medio de Pago.....	9
6. Envío.....	9
7. Repartidor.....	10
8. Dirección.....	10
9. Envio Mensajeria.....	10
10. Reclamo.....	11
Migración.....	13
DER del modelo BI.....	14
Estrategias.....	15
Tablas.....	15
1. Envio.....	15
2. Local.....	16
3. Localidad.....	16
4. Tipo_Paquete.....	16
5. Categoria_Local.....	16
6. Repartidor, Usuario Y Operador.....	17
7. Cupon_Descuento.....	17
8. Envio_Mensajeria.....	17
9. Pedido.....	18
10. Reclamo.....	18
Funciones.....	20
Vistas.....	21

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 Envío 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 Envío_Mensajería tengan una referencia a una tabla Envío 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_Mensajeria: 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).

EstadoXMensajeria: 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_mensajeria* ya que consideramos que ante un gran volúmen 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.

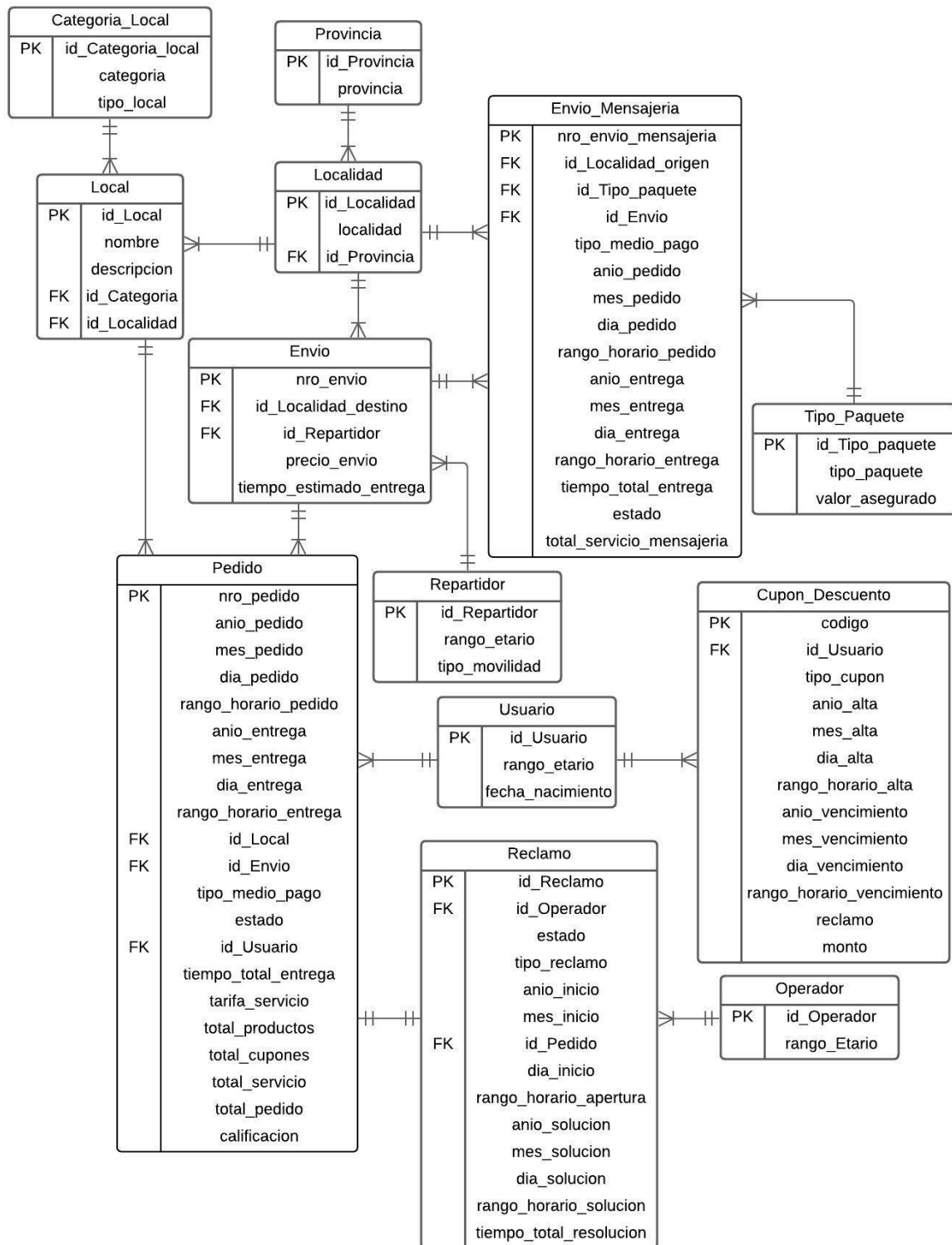
Creamos las tablas necesarias y tuvimos en cuenta respetar los tipos de datos que se presentan en la tabla maestra al realizar su migración. La mayoría de las tablas tendrá PK autoincremental salvo algunas las cuales consideramos que serían ideales para utilizarlas como PK ya que no se repetían entre 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. Algunas tablas dimensionales, como Tiempo, Día, Rango Horario, Rango Etario, Tipo Medio de Pago, Tipo Local, tipo Movilidad, Estado Pedido, Estado Reclamo y Estado Mensajería decidimos **desnormalizarlas** para un mejor rendimiento.

Por otro lado, las **tablas de hechos** que nos quedaron son **BI_Pedido**, **BI_Envio_Mensajería** 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_mensajería, y otros totales.

⇒ **IMPORTANTE:** Agregamos una nueva tabla DirecciónXUsuario 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.

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. Envío

La tabla BI_Envío 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. Los campos ID_LOCALIDAD_DESTINO y ID_REPARTIDOR son dos FK que representan: la localidad destino a donde se dirige el envío y el id 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

2. 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. **Decidimos desnormalizar la tabla que almacena los datos sobre el tipo de local y colocar directamente un campo en la tabla de *Categoria_Local* que almacene este valor (el tipo de local que corresponde) como un NVARCHAR.**

3. 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.

4. 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.

5. Categoria_Local

Esta tabla se crea para almacenar las categorías de locales. Además, para un mejor rendimiento **se decidió desnormalizar la tabla Tipo_Local y almacenar en esta tabla el campo tipo_local.**

6. Repartidor, Usuario Y Operador

Se crean estas 3 tablas únicamente para guardar, en el caso del repartidor **se decidió desnormalizar la tabla del tipo de**

movilidad para un mejor rendimiento y almacenarlo directamente como un campo en la tabla Repartidor, además de contener un campo con su rango etario. En el caso del usuario almacenamos el rango etario y la fecha de nacimiento y en el caso del operador solo el rango etario.

7. 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, para un mejor rendimiento **se decidió desnormalizar las tablas dimensionales Tiempo que contenía los campos Año y Mes, Día que contenía el campo Día y la tabla Rango_Horario que contenía el campo de tipo NVARCHAR Rango_Horario**. Por lo que guardamos todos estos campos en Cupon_Descuento tanto para la fecha de alta como para la de vencimiento. 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

8. 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 pertenecían a las tablas **Tiempo, Día y Rango_Horario que fueron desnormalizadas para un mejor rendimiento**), ya que en las vistas se requiere de los mismos para consultar sobre cuando se pidió y cuando se entregó, tener en cuenta que el año se guarda como INT y los meses y días se guardan como NVARCHAR para de esa forma mostrar que mes es y qué día de la semana corresponde según la fecha. También están los campos como los rangos horarios que nos muestran en formato NVARCHAR el

rango horario en el que se pidió un envío mensajería y en el que fue entregado, y el estado del envío mensajería (la tabla **Estado_Mensajería** también fue desnormalizada para un mejor rendimiento). Por último, se relaciona con las tablas Tipo_Paquete, Localidad y Envío con FKs, teniendo también los campos del tipo_medio_pago en el cual solo se guarda qué tipo de medio de pago se utilizó como NVARCHAR, el **tiempo total de entrega** que se obtiene haciendo un pequeño cálculo y el monto total del servicio.

9. Pedido

Esta tabla se crea para almacenar la información relacionada con los pedidos. Además, para un mejor rendimiento **se decidió desnormalizar las tablas dimensionales Tiempo que contenía los campos Anio y Mes, Dia que contenía el campo Dia, la tabla Rango_Horario que contenía el campo de tipo NVARCHAR Rango_Horario y la tabla Estado_Pedido que contenía el campo Estado**. Guardamos todos estos campos Anio, Mes, Dia y Rango_Horario tanto para la fecha en la que se hizo el pedido como para la fecha en que se entregó, y el campo estado para saber el estado del pedido. Por otro lado, **creamos un campo TIEMPO_TOTAL_ENTREGA en la tabla Pedido que guarda la diferencia en minutos entre que se realizó el pedido y entre que se entregó** ya que era un dato que íbamos a necesitar para las métricas.

10. Reclamo

Esta tabla se crea para almacenar la información relacionada con los reclamos. Además, para un mejor rendimiento **se decidió desnormalizar las tablas dimensionales Tiempo que contenía los campos Anio y Mes, Dia que contenía el campo Dia, la tabla Rango_Horario que contenía el campo de tipo NVARCHAR Rango_Horario, la tabla Estado_Reclamo que contenía el campo Estado y la tabla Tipo_Reclamo que contenía el campo de tipo NVARCHAR Tipo_Reclamo**. 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, el campo estado para saber el estado del mismo y el campo tipo_reclamo para saber de qué tipo es el reclamo que se inició. 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.

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.

Vistas

A continuación damos explicación de las vistas creadas

V_MayorCantidadPedidos

Esta vista muestra el día de la semana y la franja horaria con la mayor cantidad de pedidos, agrupados por localidad y categoría del local, para cada mes de cada año. Utiliza una consulta CTE para calcular el recuento de pedidos y asignar un número de fila basado en ese recuento. Luego, filtra los resultados para obtener solo la fila con el número de fila igual a 1, que representa la mayor cantidad de pedidos. La vista proporciona información sobre los momentos más concurridos para cada localidad y categoría de local.

V_MontoTotalXPedidosCancelados

Esta vista muestra el monto total que no cobró cada local de los pedidos que fueron cancelados según el día de la semana y la franja horaria. En este caso **asumí que la tarifa de servicio, el precio de envío, la propina y el total descontado por cupones no cuentan**, es decir, el local sólo recibe el valor total por los productos que se habían pedido.

V_ValorEnvioPromedioMensualXLocalidad

Esta vista muestra el promedio mensual del valor de envío para cada localidad. Filtra los pedidos cancelados y agrupa los resultados por mes y localidad de destino. Calcula el promedio del precio de envío para cada grupo y muestra el resultado final con el mes, la localidad y el promedio mensual. Brinda información sobre el promedio mensual de los valores de envío por localidad.

V_DesvioPromedioEnTiempoDeEntrega

Esta vista muestra el desvío promedio en tiempo de entrega según el tipo de movilidad, el día de la semana y la franja horaria, y se calcula en minutos. Para realizar esta vista hicimos un UNION ALL en el FROM entre las tablas BI_Pedido y BI_Envio_Mensajeria, a las cuales se les hizo un INNER JOIN a cada una con las tablas BI_Envio y BI_Repartidor para así obtener el tipo de movilidad, el tiempo estimado de entrega. Con esto, obtuvimos una nueva tabla con los campos Tipo Movilidad, Dia, Franja Horaria, Tiempo Total Entrega y Tiempo Estimado Entrega

desde la cual hicimos la consulta final que nos proporciona información sobre el promedio de la diferencia de minutos entre el tiempo estimado y el tiempo real en que se realizó la entrega para cada franja horaria, día y tipo de movilidad.

V_MontoTotalDeCuponesUsadosXMes

Esta vista muestra el monto total de cupones utilizados por mes y rango etario de los usuarios. Se realiza un inner join entre las tablas BI_Pedido y BI_Usuario para obtener información sobre los pedidos y los usuarios correspondientes. Luego, se agrupan los resultados por mes de pedido y rango etario del usuario. Se suma el total de cupones utilizados en cada grupo y se muestra el resultado final con el mes, el rango etario y el monto total de cupones utilizados. Esta vista proporciona información sobre el monto total de cupones utilizados por mes y rango etario de los usuarios.

V_PorcentajeEntregasMensuales

Esta vista calcula el porcentaje de pedidos y envíos de mensajería entregados mensualmente según el rango etario de los repartidores y la localidad. Se tienen en cuenta tanto los pedidos entregados como los envíos de mensajería entregados. La vista utiliza expresiones de tabla común (CTE) para recopilar los totales y calcular los porcentajes de entregas mensuales. Proporciona una visión general del desempeño de las entregas en cada categoría, considerando el total general de entregas mensuales. El primer CTE TotalEntregados recopila la cantidad total de pedidos y envíos de mensajería entregados agrupados por año, mes, rango etario y localidad. Para esto utilizamos una combinación con UNION ALL de las tablas de pedidos y envíos de mensajería para filtrar solo los registros que fueron entregados. Al contar estos registros se almacena el resultado en TOTAL_ENTREGADOS. El segundo CTE llamado PorcentajeEntregas calcula el porcentaje de entregas mensuales para cada combinación de año, mes, rango etario y localidad. Utilizamos SUM(TOTAL_ENTREGADOS) OVER (PARTITION BY ANIO, MES) para obtener la suma total de entregas mensuales y luego dividir la cantidad de entregas en cada categoría por esta suma total. El resultado se almacena en la columna PORCENTAJE_ENTREGAS. Finalmente, la vista selecciona las

columnas de interés (año, mes, rango etario, localidad y porcentaje de entregas) de la tabla PorcentajeEntregas y las devuelve como resultado.

V_ValorAseguradoPromedioMensual

Esta vista muestra el promedio mensual del valor asegurado de los paquetes enviados a través del servicio de mensajería, según el tipo de paquete. Se calcula el promedio agrupando los datos por mes de entrega y tipo de paquete, proporcionando así una medida del valor promedio asegurado para cada tipo de paquete en cada mes.

V_PromedioCalificacionMensual

Esta vista muestra el promedio mensual de calificación por local, calculando el promedio de las calificaciones asignadas a los pedidos entregados en cada local durante cada mes. Se agrupan los resultados por ID de local, año y mes, y se calcula el promedio de calificación para cada grupo. Es útil para analizar el rendimiento y la satisfacción de los locales según las calificaciones recibidas en cada mes.

V_CantidadReclamosMensuales

Esta vista muestra la cantidad de reclamos mensuales en cada local, desglosados por año, mes, día y franja horaria de apertura del local. Proporciona una visión general de la frecuencia de reclamos en cada local a lo largo del tiempo.

V_TiempoPromedioResolucion

Esta vista muestra el tiempo promedio de resolución de reclamos según el año, mes, tipo de reclamo y rango de edad del operador. Nos proporciona una visión general del tiempo promedio que se tarda en resolver los reclamos, agrupados por diferentes categorías.

V_MontoMensualGenerado

Esta vista muestra el monto que se generó a partir de los cupones pero solo los que fueron por reclamo, para esto nos facilita el campo RECLAMO, cuando figura 'Si' es porque fue un cupón generado a partir de un reclamo, sino es NULL en cualquier otro caso. Si fue un cupón generado por un reclamo entonces sumamos el monto para luego mostrar el total por cada mes de cada año.