



Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

# Gestión de Datos

Trabajo Práctico

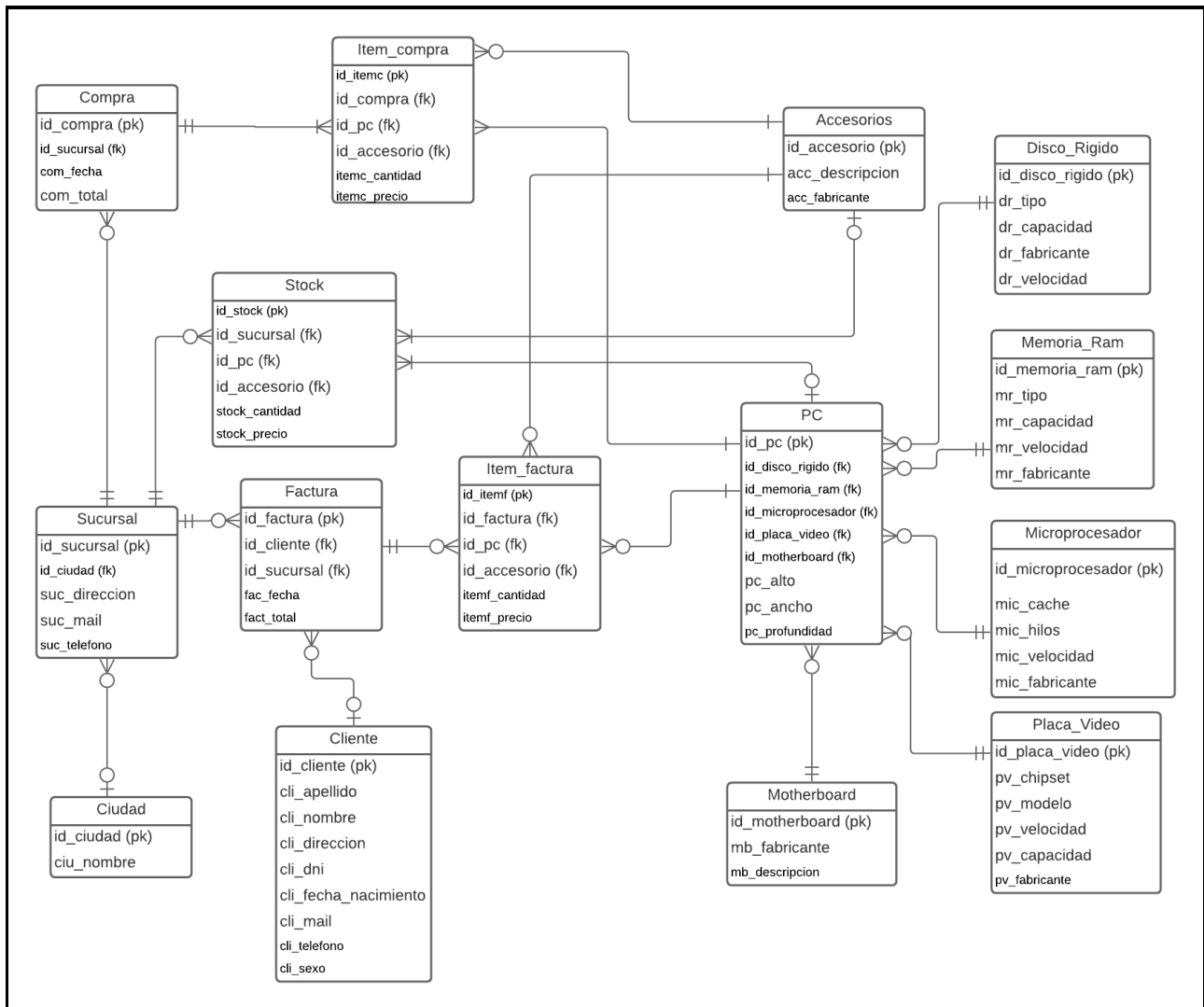
1º Cuatrimestre 2021

FRBA - Computación

# Índice

DER Modelo Relacional.....	3
Decisiones estratégicas.....	4
Migración de datos.....	4
Motherboards.....	4
Placa de video.....	4
PC.....	4
Accesorios.....	4
Sucursales.....	4
Clientes.....	4
Compras.....	4
Ítem compra.....	5
Facturas.....	5
Ítem factura.....	5
Stock.....	5
Triggers.....	5
Actualizar stock al comprar.....	5
Actualizar stock al facturar.....	5

# DER Modelo Relacional



# Decisiones estratégicas

## Migración de datos

### Motherboards

Al no haber datos sobre las motherboards en la tabla maestra provista se decidió modelar la tabla dejando las foreign keys correspondientes (id\_motherboard en la tabla PC) con valor *NULL*.

### Placa de video

Como no hay códigos de las placas de video dejamos que se genere un id automáticamente utilizando la keyword *IDENTITY*.

Ademas al migrar los datos, dado que no sabemos si todas las columnas tienen valores no nulos lo mejor es filtrar en el *WHERE* todas aquellas filas que tengan al menos un campo de la placa de video no nulo.

### PC

Como no hay datos sobre las motherboards se deja la foreign key con valor *NULL*.

### Accesorios

Siendo que no hay datos sobre el fabricante de los accesorios se decidió dejar este campo con valor *NULL*

### Sucursales

De igual manera que con las placas de video, al no haber un código de identificación se decidió que el motor los genere automáticamente.

Asimismo, se decidió tomar por válida cualquier sucursal que tenga al menos un campo no nulo (Dirección, mail o teléfono)

### Clientes

Ya que no hay datos sobre el sexo de los clientes por lo que por ahora se optó por dejar el valor en *NULL*.

De igual manera, al no haber un código de identificación de los clientes vamos a tomar por valido cualquier cliente que tenga al menos un campo que no sea *NULL*.

### Compras

Para poder mantener el *IDENTITY* en el campo id, y que las próximas compras que se generen obtengan un id autonumerico al insertar, y a la vez poder conservar los códigos de identificación de la tabla maestra es necesario permitir insertar *IDENTITYs* en la tabla con el comando *SET IDENTITY\_INSERT [ALTA\_DATA].[Compra] ON*. Luego se deshabilita para impedir que se ingresen IDs inválidos durante el funcionamiento normal del aplicativo.

## Ítem compra

Al no haber un id para cada ítem de una compra se deja que el *IDENTITY* genere uno automáticamente.

## Facturas

Ídem caso de los ids con la tabla de Compra, se permite insertar *IDENTITYs* durante la migración y luego se deshabilita.

Dado que los precios de venta son calculados en base a los precios de compra de cada producto. Por ahora se deja en *NULL* el precio total de la factura y luego se actualizará cuando se migren los ítems correspondientes a cada factura.

## Ítem factura

Similar al caso de los ítems de las compras, al no haber un id para cada ítem vendido en una factura se deja que el *IDENTITY* lo genere automáticamente.

Como no se especifica la cantidad de ítems vendidos se asume que se vende una sola unidad de cada producto.

Los precios se calculan como el promedio del precio de compra del producto mas el 20% (Incremento indicado en el enunciado)

Una vez migrados estos datos se actualizan los precios totales de las facturas.

## Stock

La tabla de stock se actualiza al final de la migración dado que es necesario calcular el stock actual a partir de la cantidad de productos comprados y vendidos.

En este punto, además se calcula el precio del producto en stock como el promedio del precio de compra más un 20%.

Adicionalmente, se agrega una *CONSTRAINT* que impide que el stock de un producto sea inferior a 0.

## Triggers

### Actualizar stock al comprar

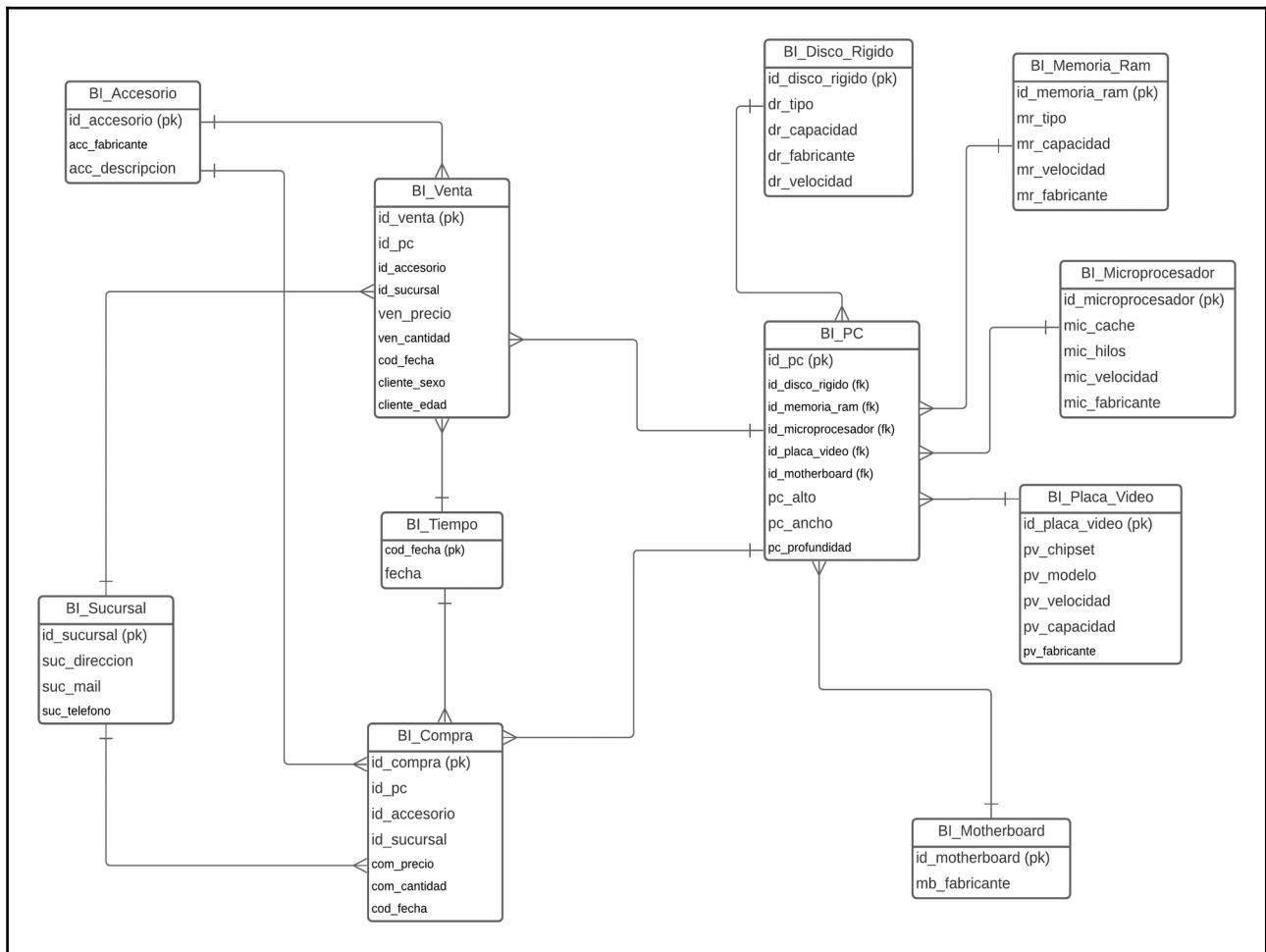
Cuando se inserta una fila en la tabla de item\_compras se actualiza el precio en stock del producto comprado al precio de compra mas un 20%. De esta manera se puede mantener un precio actualizado ante los cambios del mercado.

Asimismo, se actualiza la cantidad disponible del producto en la tabla Stock.

### Actualizar stock al facturar

Al insertarse una fila en la tabla de item\_factura se actualiza la cantidad disponible del producto en la tabla Stock.

# DER Modelo BI



# Decisiones estratégicas

## Tablas

### Hechos

Se decidió crear dos tablas de hechos: Compras y Ventas. En cada una de estas tablas se guardan las primary keys de las tablas de dimensiones y también los datos intrínsecos de estas entidades (Cantidades y precios)

### Dimensiones

Las dimensiones a tener en cuenta son: Cliente, Tiempo, Sucursal, Accesorio y PC. A su vez la dimensión PC depende de cinco tablas que representan los componentes de la PC, esto se hizo de esta manera para mantener la similitud con el modelo relacional y además, para poder soportar varios modelos de PC que tengan algún componente igual sin necesidad de tener datos redundantes.

### Dimensión Cliente

Dado que no hay casos de uso que requieran filtrar por los datos de los clientes (Sexo y rango de edad) y ya que estos datos solo existen para las ventas, se decidió des-normalizar estos datos y almacenarlos directamente sobre la tabla de hechos de Ventas.

### Dimensión Tiempo

Se decidió conformar la tabla de la dimensión tiempo como una primary key (Autogenerada) y un campo fecha de tipo DATETIME2 con el fin de poder utilizar de manera conveniente las funciones propias de este tipo de dato.

Al solo ser necesario evaluar el mes en el que se realizaron las operaciones se decidió utilizar la función *EOMONTH*, provista por el motor, de manera que resulte mas simple comparar dos fechas dadas y evitar así tener muchas entradas en esta tabla que se correspondan al mismo periodo. Como consecuencia, es necesario utilizar las funciones de formato en las vistas que resuelven los casos de uso para dar claridad a lo que se representa con este dato.

## Migración

### General

A excepción de la dimensión tiempo, el resto de dimensiones fueron migradas a partir de sus contrapartes en el modelo transaccional. Dado que nuestro modelo relacional ya es consistente, no es necesario hacer consideraciones sobre los campos que contienen para determinar si se trata de un dato valido.

Se decidió crear tablas nuevas para el modelo de BI en lugar de reutilizar las del modelo transaccional (Por ejemplo, mediante vistas), ya que se requiere la disponibilidad total del modelo transaccional para atender las necesidades de negocio. Por este motivo, se opto por realizar una migración de los datos a un modelo paralelo. Esta migración se puede programar de manera tal de no interrumpir el proceso de negocio.

## Tiempo

Las entradas de la dimensión tiempo se obtienen a partir de obtener todas las fechas existentes en las tablas Factura y Compra del modelo transaccional. Una vez obtenidas todas las fechas se insertan en la tabla Tiempo aquellas que se correspondan a meses diferentes, de esta manera evitando la duplicación de periodos.

## Ventas

Los datos de las ventas se obtienen a partir de las tablas Factura e Item\_Factura del modelo transaccional. Los IDs se generan automáticamente por el objeto *IDENTITY*

## Compras

Los datos de las compras se obtienen a partir de las tablas Compra e Item\_Compra del modelo transaccional. Los IDs se generan automáticamente por el objeto *IDENTITY*

## Casos de uso

### Promedio de tiempo en stock

Este caso de uso esta resuelto en las vistas [ALTA\_DATA].[pc\_promedio\_tiempo\_en\_stock] y [ALTA\_DATA].[acc\_promedio\_tiempo\_en\_stock].

Se calcula el promedio de tiempo en stock de un producto como los días del mes dividido la cantidad de ventas. De esta manera se obtiene cada cuantos días se hace una venta de cada producto, obteniendo efectivamente, para cada mes y cada sucursal el promedio de tiempo entre dos ventas consecutivas de ese producto. Esta magnitud se encuentra medida en días.

### Precio promedio de compra y venta

Este caso de uso esta resuelto en las vistas [ALTA\_DATA].[pcPrecioPromedioCompraVenta] y [ALTA\_DATA].[accPrecioPromedioCompraVenta].

Estas vistas calculan el precio promedio de venta como el promedio del precio unitario de cada ítem de la tabla Ventas. En el caso de las compras se calcula como el promedio de los precios de compra de la tabla Compras

### Cantidad de artículos comprados y vendidos por sucursal y por mes

Este caso de uso esta resuelto en las vistas [ALTA\_DATA].[pcCantidadCompraVenta] y [ALTA\_DATA].[accCantidadCompraVenta].

Para obtener la cantidad de ítems vendidos se suman las cantidades de los productos de la tabla Ventas. De forma similar se hace para los productos comprados con la tabla Compras.

## Ganancias

Este caso de uso esta resuelto en las vistas [ALTA\_DATA].[pcGanancias] y [ALTA\_DATA].[accGanancias].



Las ganancias se calculan obteniendo la suma de los precios individuales de cada ítem en la tabla de Ventas multiplicado por la cantidad correspondiente menos la suma de los precios individuales de cada ítem en la tabla de Compras multiplicado por la cantidad comprada.