

**UNIVERSIDAD DE ORIENTE**

**NÚCLEO NUEVA ESPARTA**

**ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS**

**LICENCIATURA EN INFORMÀTICA**

**ADMINISTRACIÓN DE BASE DE DATOS**

**Base de datos Black Tech Store planificada e implementada con el SGBD Oracle   
11g Express Edition**

**Profesora: Integrantes:**

Ing. Eyamir Ugueto Andrés Luna. C.I. 25.999.071

Eduardo Rodríguez. C.I. 26.082.457

Hilary Alzate. C.I. 25.794.960

Inés Natera. C.I. 26.243.417

Javier Rodríguez. C.I. 25.877.576

**Pasos para el diseño físico**

1. Traducir el modelo lógico al SGBD seleccionado

* Diseñar las relaciones bases
* Diseñar la representación de los datos variados
* Diseñar las restricciones generales

1. Diseñar la organización de los archivos y los índices

* Analizar las transacciones
* Seleccionar la organización de los archivos
* Seleccionar los índices
* Estimar los requisitos de espacio de disco

1. Diseñar las vistas de usuario
2. Diseñar los mecanismos de seguridad
3. Considerar la introducción de una cantidad controlada de redundancia
4. Monitorizar y ajustar el sistema final

**Desarrollo e implementación del diseño físico en el Oracle**

1. Traducir el modelo lógico al SGBD seleccionado.

* Diseñar las relaciones bases.

A continuación, puede apreciarse un ejemplo de la tabla Cliente como parte del desarrollo del diseño físico (Ver Figura 1). El resto puede apreciarse en la carpeta ***BlackTechStore-DB/relationships.sql.***

CREATE TABLE cliente(

doc\_identidad NUMBER(10) NOT NULL ENABLE,

nombre\_completo VARCHAR(50) NOT NULL ENABLE,

telefono CHAR(11) NOT NULL ENABLE,

direccion VARCHAR(120) NOT NULL ENABLE,

email VARCHAR(50),

estado CHAR(1) DEFAULT 'A' NOT NULL ENABLE,

CONSTRAINT CLIENTE\_PK PRIMARY KEY (doc\_identidad),

CONSTRAINT CLIENTE\_CH\_ESTADO CHECK (estado IN('A','I')) ENABLE,

CONSTRAINT CLIENTE\_CH\_TELEFONO CHECK (regexp\_like (telefono, '^[[:digit:]]+$') AND length(telefono)='11') ENABLE

);

Figura 1. Ejemplo de creación de la tabla cliente.

* Diseñar la representación de los datos variados.

El equipo DBA decidió que resulta viable mantener actualizados los datos derivados manteniendo la coherencia de los datos operativos con los que *se* calculan.

* Diseñar las restricciones generales,

A continuación, puede apreciarse un del disparador que se encarga de ajustar los precios de venta de los artículos que se adquieran a un margen de ganancia del 30%, solo en el caso de que el costo del mismo sea mayor que el anterior (Ver Figura 2). El resto de los disparadores pueden apreciarse en la carpeta ***BlackTechStore-DB/triggers.sql.***

CREATE OR REPLACE TRIGGER detalles\_fc\_art\_trigger

AFTER INSERT ON actualiza\_articulo

FOR EACH ROW

DECLARE

costo\_actual NUMBER(10,2);

costo\_nuevo NUMBER(10,2);

BEGIN

UPDATE factura\_compra SET total = total + :NEW.subtotal WHERE numero = :NEW.numero\_factura\_compra;

UPDATE articulo SET cant\_existencia = cant\_existencia + :NEW.cantidad WHERE codigo = :NEW.codigo\_articulo;

SELECT costo INTO costo\_actual FROM articulo WHERE codigo = :NEW.codigo\_articulo;

costo\_nuevo := :NEW.costo \* 1.3;

IF costo\_nuevo > costo\_actual THEN

UPDATE articulo SET costo = costo\_nuevo WHERE codigo = :NEW.codigo\_articulo;

END IF;

END;

/

Figura 2. Ejemplo de creación de un Disparador.