**Universidad Tecnológica de Panamá**

**Facultad de Ingeniería de Sistemas Computacionales**

**BASE DE DATOS I**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DENOMINACIÓN DE LA ASIGNATURA: | | | | **BASES DE DATOS I** | | | | GRUPO: | | **VIL332** |
| PROFESORA: | | **ING. MAYLIN CHÉRIGO** | **LABORATORIO No: 7** | | | | Crear, insertar, borrar tablas | | | |
| NOMBRE: | **JOY NELATON** | | | | CÉDULA: | **8-902-1282** | | FECHA: | **9-07-2021** | |

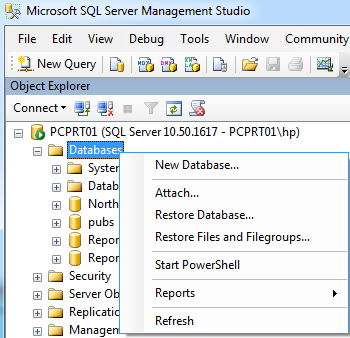
1. **INSTRUCCIONES:**

* Entrega individual.
* No coloque una página de presentación al documento. Utilizar el documento actual como base.
* Coloque la referencia utilizadas.
* Transforme a formato PDF, con el nombre: Nombre.Apellido - Laboratorio #
* El trabajo debe ser entregado a través de la plataforma Campus Virtual UTP, en la sección asignada, en la hora y fecha asignada.

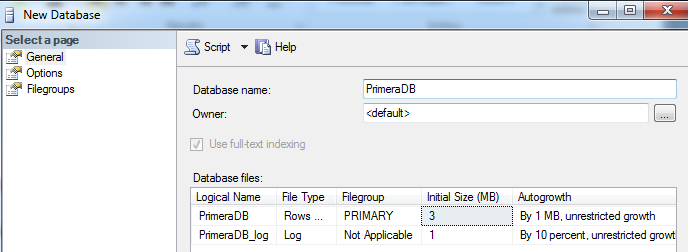
1. **METODOLOGÍA:**

Para presentar el informe de los resultados obtenidos, haga captura de pantalla desde el SQL Server mostrando la instrucción y el resultado de la consulta generada por la misma.

Copie estas capturas de pantalla a medida que va desarrollando y obteniendo resultados, colocándolas seguidas del enunciado, identificándolas como sus resultados obtenidos; con Corte y sólo presente el área de trabajo donde aparece la instrucción y el resultado obtenido, no incluya el explorador de objetos, el menú de opciones ni la barra de herramientas estándar en su respuesta final.

1.  **CREACIÓN DE UNA BASE DE DATOS**

**A.1 A través del modo gráfico:**

* Seleccione DataBases en el explorador,
* Pulsar botón derecho y desplegar menú
* Escoja New Database
* Se despliega una nueva pantalla
* Colocar en esta el nombre de la Base de Datos; en este caso **PrimeraBD.**
* Note que se crean automáticamente dos archivos de base de datos con el mismo nombre. En el primero se guardarán todos nuestros datos y el segundo (el log), almacenará todas las operaciones sobre la base de datos y que no han sido guardadas. Esto permite hacer rollback o garantiza la recuperación de información en caso, por ejemplo, de alguna falla eléctrica.
* Estos archivos tienen definido un tamaño que puede alterar de acuerdo con el tamaño que estime usted ocuparán los datos que almacenarán. También encontrará la ruta donde se creará y almacenarán dichos archivos.
* Tome la opción **ACEPTAR.** Note que ahora su base de datos debe aparecer listada en DataBases.

Resultados Obtenidos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

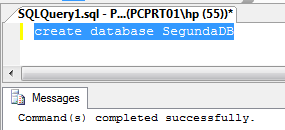
Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**A.2 A través de sentencias SQL**

Escriba la sentencia

**Create database SegundaDB**

Donde:

**Create** especifica que estoy creando un objeto

**Database** indica que creo una Base de datos

**SegundaBD** es el nombre de la Base de Datos

Resultados Obtenidos

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. **CREACIÓN DE LAS TABLAS**

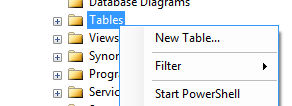
**Departamento ( No\_depto, Nombre\_depto)**

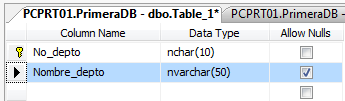
**PK**

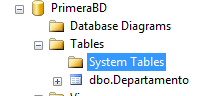
**Empleado (id\_empleado, nombre, Depto\_labora)**

**PK FK**

**B.1 A través del modo gráfico en Base de datos Primera DB**:

**Creación de Tabla Departamento que sólo tiene llave primaria**:

* Abrir la base de datos PrimeraDB, y dé click sobre **Tables**.
* Pulsar botón derecho y desplegar menú. Seleccionar **New Table**
* Se despliega una nueva pantalla que le permite insertar las columnas, los tipos de datos y definir los tipos de llave (posicionare sobre el campo y botón derecho).
* Se va a crear la tabla DEPARTAMENTO, para ello defina No\_depto como llave primaria, tomando la opción de Primary Key que aparce al desplegar menú con botón derecho. Deje sin activar la columna **Allow nulls** (permita valores nulos); pues NO se desea que permita nulos.
* Cierre la ventana X; indique que quiere salvarla como parte de la base de datos llamada PrimeraBD (Yes), dé nombre a la tabla y salve la tabla con el nombre **Departamento** (note que se creó esta tabla como parte de la bd PrimeraBD)



Resultados Obtenidos

Escala de tiempo

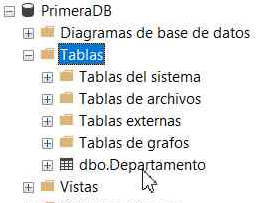
Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

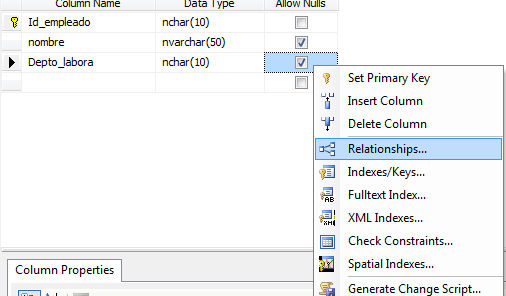
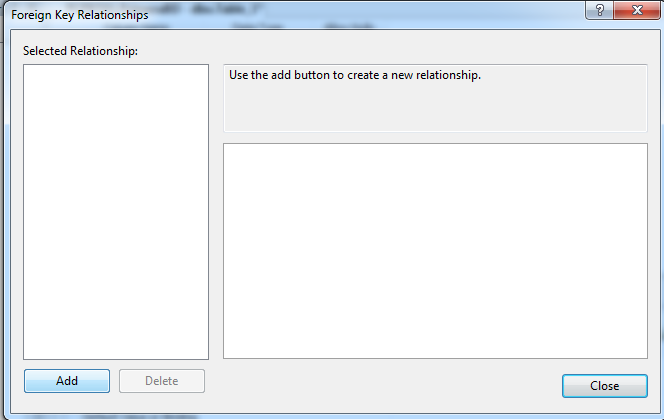
Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

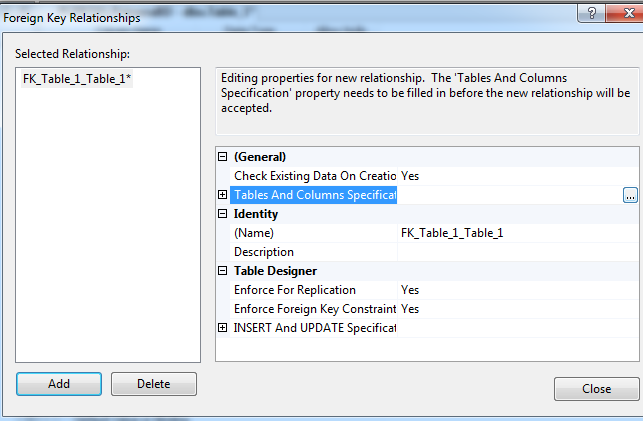
Descripción generada automáticamente

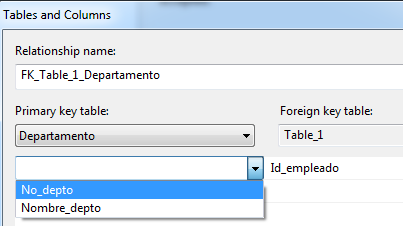
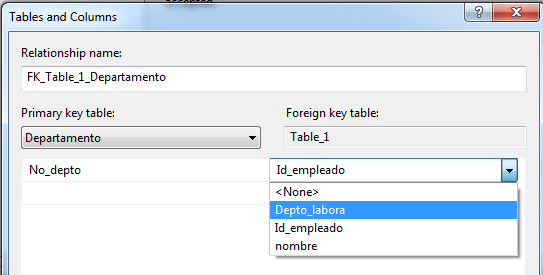
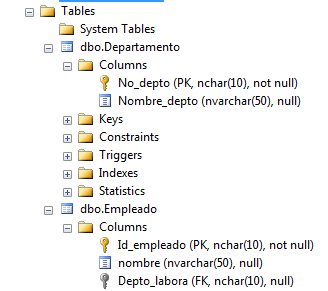
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Creación de Tabla Empleado que tiene su llave primaria y un FK que hace referencia a una llave PK de la tabla Departamento.**

* Ahora creará la tabla empleado, y definina ID\_empleado como llave primaria (pulse botón derecho, seleccione primary key).
* Defino **nombre** con su tipo de dato y not null.
* El siguiente campo es muy importante, ya que en éste se define una llave foránea. Por lo tanto debe corresponderse en tipo y cantidad con la definida en la tabla padre como llave primaria. Una vez definido su tipo y largo, pulse botón derecho y seleccione Relationships.
* Aparece la siguiente pantalla. Tome en esta la opción **Add.**
* Se despliega esta información sobre la pantalla, tome los puntos suspensivos que aparece al hacer click sobre **Tables and Columns Specifications.**



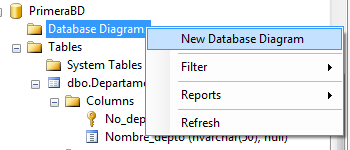
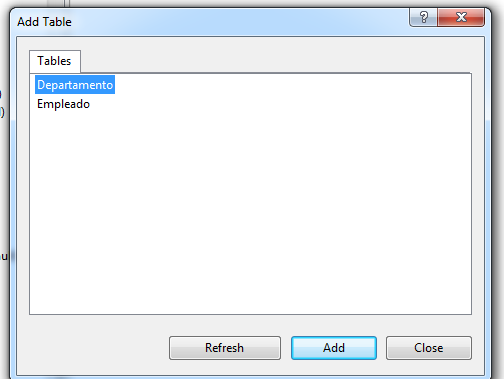
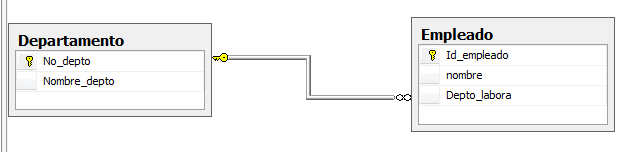
* ****Seleccione la tabla donde tenemos la primary key (Departamento) con el cual se relaciona el campo que estamos creando. Automáticamente sale otro renglón abajo para definir qué atributo de esta tabla es el que desea establecer como enlace. Seleccionamos No\_depto.
* Seleccione el campo de la tabla donde va a ser foreing Key, es decir (Depto\_labora). La tabla Empleado hasta este momento no le ha dado el nombre por lo que es Table\_1 (nombre puesto por el sistema).
* Escoja Ok y Close.
* Ahora salve la tabla cerrando la pantalla de creación (X), sale mensaje si quiere que sea parte de PrimeraDB, diga Yes y luego le coloque el nombre Empleado y salve (YES y YES). Este último Yes relacionó las dos tablas.
* Note que se tiene las dos tablas creadas. Observe que, la tabla departamento tiene una llave primaria: No\_depto, en tanto que la Tabla Empleado, tiene una llave primaria Id\_empleado y como llave foránea al Depto\_labora

Resultados Obtenidos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Cómo ver el Diagrama Relacional de lo que se ha creado**

* Para poder ver el diagrama Relacional de lo que ha realizado, ubíquese sobre Database Diagram , pulsando el botón derecho, tomando New Database Diagram.
* Sale esta pantalla con el nombre de las tablas. 
* Pulse Add e irá adicionándolas una a una al área de trabajo.
* Al final cierre y arrastre su Modelo relacional al lugar donde lo vea mejor. Debe salir las relaciones de llaves primarias y foráneas definidas. Usted puede mover, quebrar las líneas de conexión según lo desee.
* Del lado de la Tabla Padre aparece una llave pequeña indicando la llave primaria. Del lado de la llave foránea (tabla hijo) le sale un símbolo de una cadena o enlace. Esto denota que la llave primaria de esta tabla Departamento, se relaciona con un Fk en Empleado.

Resultados Obtenidos

**Interfaz de usuario gráfica, Texto, Aplicación, Tabla

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente**

**Formato General para Creación de Tablas**

**CREATE TABLE NombreTabla**

**(**

**Nombre\_columna1 tipoDato [PRIMARY KEY] [NOT NULL] [UNIQUE][DEFAULT defaultOption]**

**[REFERENCES ParentTableName] [CHECK searchCondition] [,...]**

**[PRIMARY KEY (listOfColumns),]**

**[UNIQUE (listOfColumns),] […,]**

**[FOREIGN KEY (listOfFKColumns) REFERENCES ParentTableName [(listOfCKColumns)],**

**[CHECK (searchCondition)] [,…] ,**

**Nombre\_columna2….**

**)**

Las restricciones vistas en clases (not null, unique, primary key, foreign key, check y default) se pueden hacer a nivel de columnas o a nivel de tablas.

* **A nivel de columnas**: Hace referencia a una columna y se define dentro de la especificación de la misma

Nombre\_columna [CONSTRAINT nombre\_restricción] tipo\_restricción

* **A nivel de Tablas**: Hace referencia a una o más columnas y es definido separadamente de la definición de las columnas de la tabla. Se puede definir cualquier restricción excepto NOT NULL

(ver anexo al final para mayor detalle)

NOTA: Se recomienda que los nombres de los constraint sean:

NOMBRE-DE-TABLA\_ NOMBRE-CAMPO\_TIPO-RESTRICCION NN, PK,FK

**Procederemos a crear estas tablas en la BD SegundaDB**

Departamento1 ( **Num\_departamento**, Nombre\_departamento)

**PK**

Empleado1 (**Identif\_Empl**; Nombre\_Emp, **Departamento\_lab**)

**PK** **FK**

**Creación de tabla Departamento1.**

**A nivel de columnas**:

use SegundaDB

create table Departamento1

**(**

Num\_departamento nchar(10)

Constraint Departamento1\_num\_depto\_pk primary key,

Nombre\_departamanto nchar(50) not null

**)**

Crea la tabla Departamento1 como parte de la base de datos SegundaDB. Por eso la activa con el comando **use SegundaDB**.

Defina la primera columna de esta tabla ***Num\_departamento*** y seguido le indica a través del **constraint** que la llave primaria es el atributo Num\_departamento, luego defina la siguiente columna con su tipo de dato e indique que no debe ser nula (Como no ha colocado nombre a esta restricción not null, en caso de querer o necesitar cambiarla, tendrá que ver qué nombre le puso el sistema).

Si la llave primaria es compuesta, esta no puede ser definida a nivel de columna; por lo que es recomendable hacerlo una vez definidas todas las columnas de la tabla, para que la restricción sea a nivel de tabla y poder colocar las columnas que necesite, ya que han sido definidas previamente.

Resultados Obtenidos

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**A nivel de Tablas**

use SegundaDB

create table Departamento1

**(**

Num\_departamento nchar(10),

Nombre\_departamento nchar(50) not null,

Constraint Departamento1\_Nombre\_departamento\_pk

primary key (Num\_departamento),

**)**

**Creación de tabla Empleado1.**

**OBSERVE; Esta es otra forma de poder definir las restricciones, sin definir nombre a las restriccciones (constraint); pero no es lo recomendado, sin embargo lo puede encontrar de esta forma.**

use SegundaDB

create table Empleado1

(

Identif\_Empl nchar(10),

Nombre\_Empl nchar(50) not null,

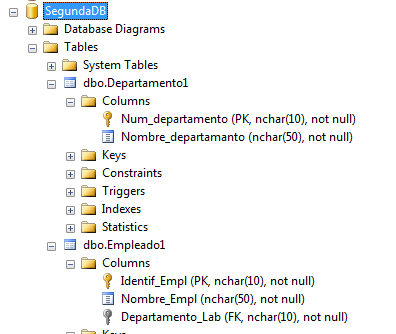
Departamento\_Lab nchar(10) not null ,

primary key (Identif\_Empl),

foreign key (Departamento\_Lab)

References Departamento1 (Num\_departamento)

)



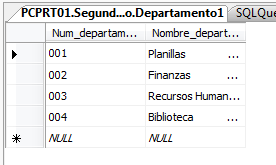
* Note las tablas creadas y las restricciones con las que se crearon las mismas,

Resultados Obtenidos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **INSERTANDO VALORES A LAS TABLAS**

**C.1 A través del modo gráfico**

A través del modo gráfico es sumamente sencillo insertar valores a las tablas.

* Solo hay que ubicarse sobre la tabla a la que desea insertar datos,
* Botón derecho
* Tomar Edit top 200 row
* Trae una tabla para ir adicionando valores a las columnas.
* **Innserte algunos valores usando esta opción.**

Resultados Obtenidos

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**C.2 A través de sentencias SQL**

insert into NombreTabla

values (valorcolum1,valorcolum2….,valorcolumN)

insert into Departamento1

values ('005','UTP') Se colocan entre comilla simple porqu son tipo Char

Resultados Obtenidos

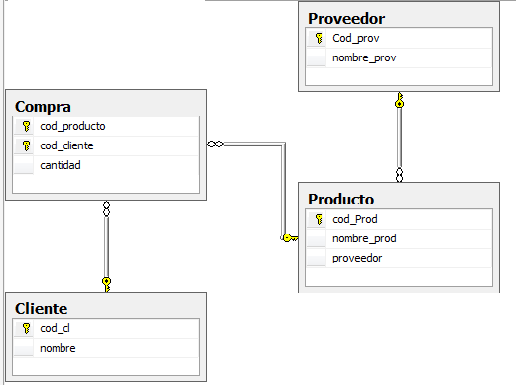
**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Tabla

Descripción generada automáticamente**

1. **Ahora creará una tabla con llave compuesta, en donde cada parte de la llave compuesta es a su vez un foreign key**



Create Database VentaDB

use VENTADB

CREATE TABLE Cliente

( cod\_cl nchar(4)

Constraint **Cliente\_cod\_cl\_pk** primary key,

nombre nvarchar(**3**0) not null

)

CREATE TABLE Proveedor

( Cod\_prov char(5)

Constraint **Proveedor\_cod\_prov\_pk** primary key,

nombre\_prov varchar(**3**0) not null

)

CREATE TABLE Producto

( cod\_Prod char(4)

Constraint **Producto\_cod\_prod\_pk** primary key,

nombre\_prod varchar(20),

proveedor char(5),

constraint **Producto\_cod\_prod\_fk** foreign key (proveedor)

references Proveedor(Cod\_prov)

)

Tabla donde es llave primaria y el atributo Llave Primaria

Atributo que es foreign key en la tabla que estamos creando

create table Compra

(

cod\_producto char(4),

cod\_cliente nchar(4),

cantidad smallint,

constraint Compra\_cod\_producto\_cod\_cliente\_PK

**primary key (cod\_producto, cod\_cliente),**

constraint Compra\_cod\_producto\_fk

**foreign key (cod\_producto) references Producto (cod\_Prod),**

constraint Compra\_cod\_cliente\_fK

**foreign key (cod\_cliente) references Cliente(cod\_cl)**

**)**

Resultados Obtenidos

**Tabla

Descripción generada automáticamente**

**Diagrama

Descripción generada automáticamente**

**ANEXOS**

**RESTRICIONES DE INTEGRIDAD: CONSTRAINT**

|  |  |
| --- | --- |
| **NOT NULL:** | Especifica que la columna no puede contener valores nulos |
| **UNIQUE:** | Esta restricción obliga a que todos los valores de una determinada columna no estén repetidos en otros registros. Si tenemos varias restricciones UNIQUE en una misma tabla, todas deben ser cumplidas a la vez para cada registro. |
| **PRIMARY KEY:** | Identifica la llave primaria de la tabla, que permite identificar cada fila de esta |
| **FOREIGN KEY:** | Establece una relación entre una columna de una tabla principal y la columna de la tabla foránea. Para que esta relación se cumpla, la clave principal que enlaza con la externa (foránea) debe cumplir obligatoriamente que las dos columnas sean del mismo tipo. |
| **CHECK:** | Evalúa por medio de expresiones los valores que se insertan en una columna. Especifica una condición que debe ser CUMPLIRSE PARA QUE PUEDA REALIZARSE LA INSERSION |
| **DEFAULT** | Como su propio nombre indica, esta restricción introduce un valor por defecto en una columna cuando no se índica ningún valor para insertar. Con esta restricción aseguramos la integridad de dominio, ya que aseguramos valores válidos para nuevos registros que se inserten |

**Eliminar una tabla creada**

**Drop table nombreTabla**

*Ejemplo: Drop table Cliente eliminará la tabla cliente*

**Colocar Comentarios**

- - Esto es un comentario que cabe en una línea y su identificador son dos guiones seguidos

/\* Esto es un comentario que abarca más de una línea.

Inicia con una barra y luego un asterisco y termina con un asterisco y luego una

barra

\*/

Resultados Obtenidos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**

1. **CONSIDERACIONES FINALES:**

*Indique en esta sección si considera o no que el laboratorio cumplió su objetivo.*

El laboratorio ha sobrepasado las expectativas previas, garantizando la obtención de conocimiento significativo.

1. **BIBLIOGRAFÍA:**

* *A fondo SQL Server, Kalen Delaney, Serie de programación Microsoft, McGraw Hill profesional*
* *http://www.aulaclic.es/sqlserver/t\_1\_1.htm*

1. **RÚBRICA:**

Esta actividad de aprendizaje tendrá una puntuación totalde100, donde la evaluación se basada en los aspectos de ***excelente***, ***bueno***, ***regular***, ***deficiente***.

Se evaluará la existencia del código como texto y las evidencias de los resultados a través de las capturas de pantalla.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N.** | **ASPECTOS QUE EVALUAR** |  |  |  |  |
| **1** | **CONTENIDO DE ACUERDO CON LO SOLICITADO EN EL ENUNCIADO** | **Excelente (90)** | **Bueno (60)** | **Regular (30)** | **Deficiente (5)** |
| Presentación de las imágenes de evidencias de los resultados – *(90 pts.)* | Presentó todos los códigos y resultados claramente. | Colocó algunos código o resultados incompletos | Hicieron falta códigos o resultados. | No colocó ningún código ni resultado. |
| **3** | **ENTREGA DE TRABAJO EN LA PLATAFORMA** – (*10 puntos)* | **Excelente (10)** | **Bueno (7)** | **Regular (5)** | **Deficiente (0)** |
| Entrega a tiempo en la plataforma. | Entregó a tiempo. | No entregó a tiempo, con excusa. | No entregó a tiempo, sin excusa. | No entregó. |