

**EL TINTO BENDITO**

JESUS ANIBAL BAENA ARCILA  
EDWIN ARMANDO ESPINOSA ZAMBRANO  
JOHANN SANTIAGO NIÑO MEDINA  
OSCAR DANIEL PIÑEROS HERNÁNDEZ  
JUAN DIEGO VASQUEZ SOSSA  
JORGE HERNÁN VICQUE SÁNCHEZ

SERGIO ARTURO MEDINA CASTILLO  
FORMADOR

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA  
MISION TIC 2022  
DESARROLLO DE SOFTWARE  
2021

## TABLA DE CONTENIDO

<b>1. FORMALIZACIÓN.....</b>	<b>3</b>
1.1. Diccionario de Datos.....	3
1.2. Primera forma normal. ....	4
1.3. Segunda forma normal. ....	5
1.4. Tercera forma normal. ....	5
1.5. Cuarta forma normal.....	6
1.6. Modelo conceptual de la base de datos. ....	7
<b>2. SCRIPTS PARA CREACIÓN DE LA BASE DE DATOS.....</b>	<b>8</b>
<b>3. SCRIPTS PARA POBLAR LA BASE DE DATOS. ....</b>	<b>10</b>
<b>4. MANEJO DEL DOM CON JAVASCRIPT. ....</b>	<b>11</b>

## **EL TINTO BENDITO**

### **SPRINT 4**

#### **1. FORMALIZACIÓN.**

##### **1.1. Diccionario de Datos.**

Elementos de información del sistema.

#### **PRODUCTOS**

- Código Producto
- Nombre Producto
- Descripción Producto
- Categoría
- URL Imagen
- Precio
- Estado (Activo, Inactivo)
- Disponible (Si, No)

#### **EMPLEADOS**

- Cédula Empleado
- Nombre
- Apellido
- Móvil
- Correo
- Cargo
- Estado (Activo, Inactivo)
- Fecha Ingreso
- Fecha Retiro

### LOCALES (o Puntos de Venta)

- Código Local
- Nombre
- Dirección
- Teléfono
- Cédula Encargado

### CLIENTES

- Cédula Cliente
- Nombre Cliente
- Apellido Cliente
- Móvil Cliente
- Correo Cliente

### PEDIDOS

- Código Pedido
- Código Producto (Puede ser más de un producto)
- Cliente (quien hace el pedido)
- Código Local (donde se entregará el pedido)
- Fecha
- Estado (Entregado, Cancelado, Abierto)

### 1.2. Primera forma normal.

Identificación de tablas y de campos.

TABLA	CAMPOS
PRODUCTOS	ID_Prod, Nombre_Prod, Desc_Prod, Cat_Prod, URL_ImgProd, Precio_Prod, Disp_Prod, Estado_Prod
EMPLEADOS	ID_Empleado, Cedula_Emp, Nombre_Emp, Apell_Emp, Movil_Emp, Correo_Emp, Cargo_Emp, Estado_Emp, Fec_IngEmp, Fec_RetEmp
LOCALES	ID_Local, Nombre_Local, Dir_Local, Tel_Local, ID_Encargado
CLIENTES	ID_Cliente, Ced_Cliente, Nom_Cliente, Apell_Cliente, Movil_Cliente, Correo_Cliente
PEDIDOS	ID_Pedido, ID_Producto, ID_Cliente, ID_Local, Fec_Pedido, Est_Pedido

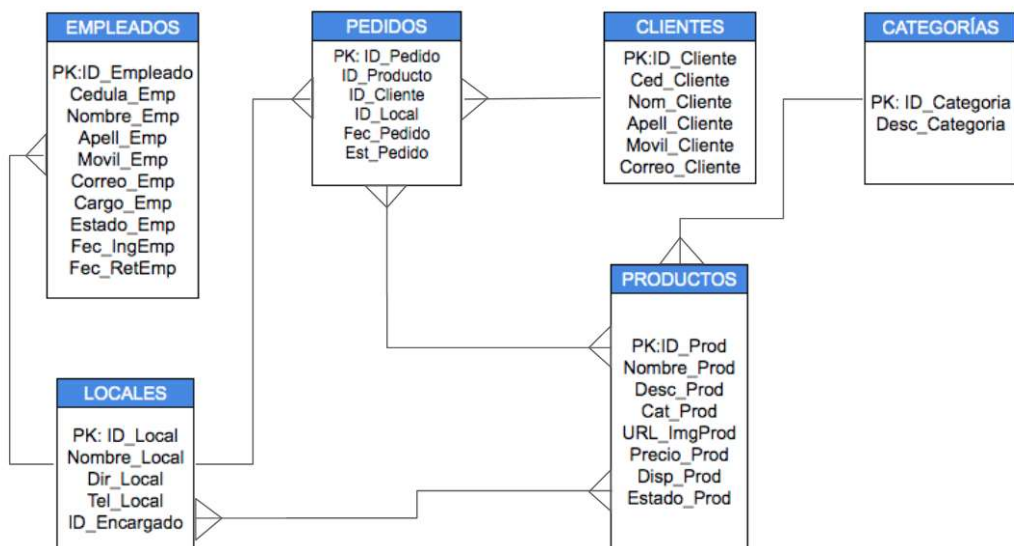
### 1.3. Segunda forma normal.

Identificación de dependencias funcionales directas (DFD) y dependencias funcionales transitivas (DFT)

TABLA	DFD (Llave Primaria)	DFT
PRODUCTOS	ID_Prod	Cat_Prod
EMPLEADOS	ID_Empleado	
LOCALES	ID_Local	ID_Encargado
CLIENTES	ID_Cliente	
PEDIDOS	ID_Pedido	ID_Producto, ID_Cliente, ID_Local,

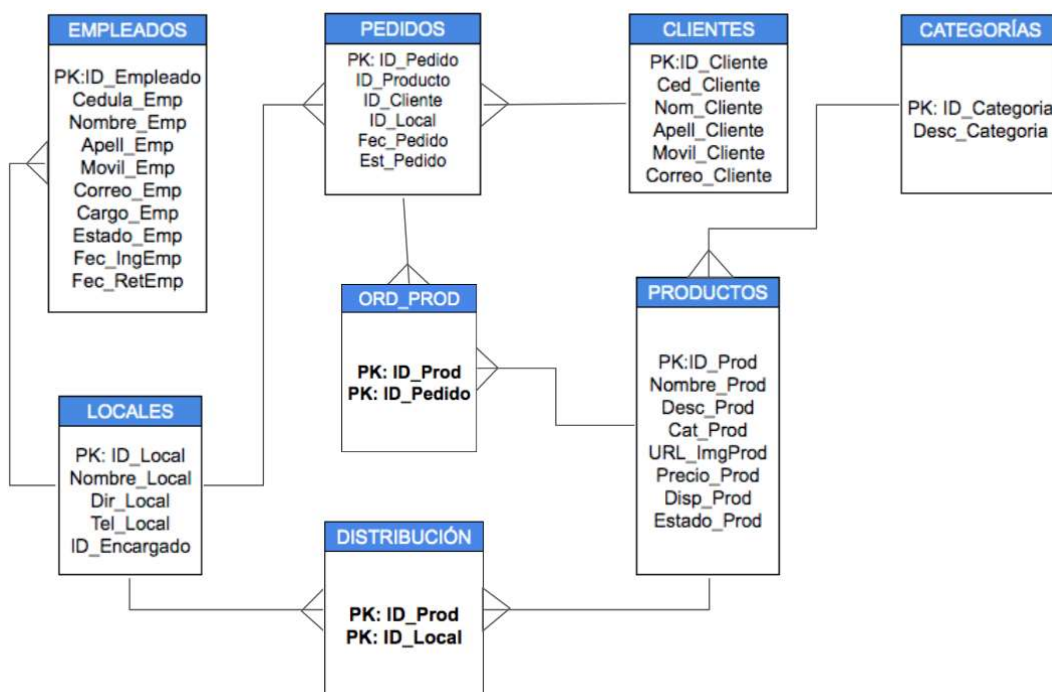
### 1.4. Tercera forma normal.

Eliminar las dependencias funcionales y establecer relaciones.

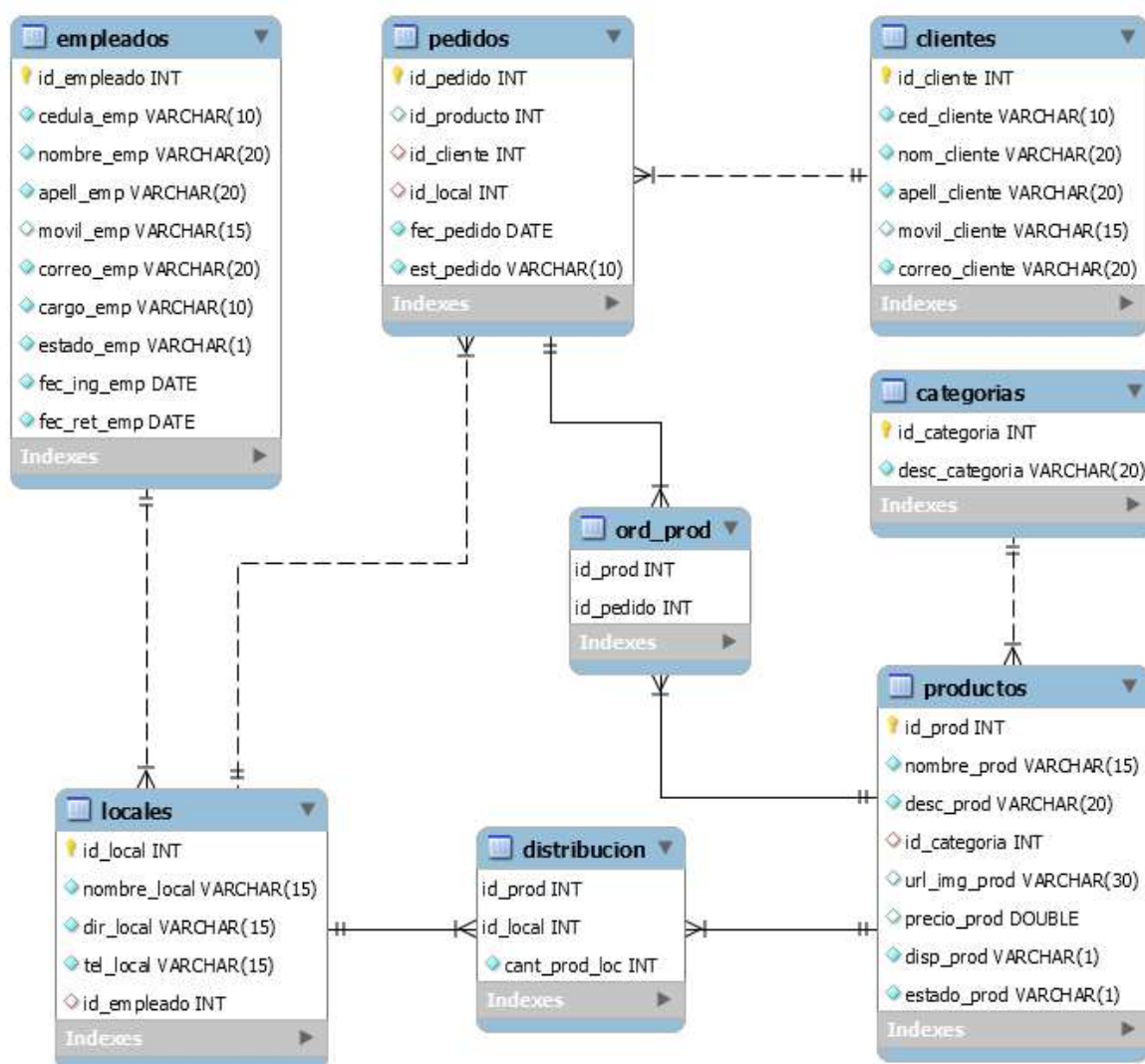


### 1.5. Cuarta forma normal.

Eliminar relaciones Muchos a Muchos.



### 1.6. Modelo conceptual de la base de datos.



## 2. SCRIPTS PARA CREACIÓN DE LA BASE DE DATOS.

```
create database el_tinto_bendito;

use el_tinto_bendito;

create table empleados
(
    id_empleado int auto_increment,
    cedula_emp varchar(10) not null,
    nombre_emp varchar(20) not null,
    apell_emp varchar(20) not null,
    movil_emp varchar(15),
    correo_emp varchar(20) not null,
    cargo_emp varchar(10) not null,
    estado_emp varchar(1) not null,
    fec_ing_emp date not null,
    fec_ret_emp date not null,
    constraint empleados_pk primary key(id_empleado)
);

create table locales
(
    id_local int auto_increment,
    nombre_local varchar(15) not null,
    dir_local varchar(15) not null,
    tel_local varchar(15) not null,
    id_empleado int,
    constraint locales_pk primary key(id_local),
    constraint locales_id_empleado_fk foreign key(id_empleado) references empleados(id_empleado)
);

create table clientes
(
    id_cliente int auto_increment,
    ced_cliente varchar(10) not null,
    nom_cliente varchar(20) not null,
    apell_cliente varchar(20) not null,
    movil_cliente varchar(15),
    correo_cliente varchar(20) not null,
    constraint clientes_pk primary key(id_cliente)
);

create table categorias
(
    id_categoria int auto_increment,
    desc_categoria varchar(20) not null,
    constraint categorias_pk primary key(id_categoria)
);

create table productos
(
    id_prod int auto_increment,
```



```
nombre_prod varchar(15) not null,  
desc_prod varchar(20) not null,  
id_categoria int,  
url_img_prod varchar(30),  
precio_prod double,  
disp_prod varchar(1) not null,  
estado_prod varchar(1) not null,  
constraint productos_pk primary key(id_prod),  
constraint productos_id_categoria_fk foreign key(id_categoria) references categorias(id_categoria)  
);  
  
create table pedidos  
(  
    id_pedido int auto_increment,  
    id_producto int,  
    id_cliente int,  
    id_local int,  
    fec_pedido date not null,  
    est_pedido varchar(10) not null,  
    constraint pedidos_pk primary key(id_pedido),  
    constraint pedidos_id_cliente_fk foreign key(id_cliente) references clientes(id_cliente),  
    constraint pedidos_id_local_fk foreign key(id_local) references locales(id_local)  
);  
  
create table ord_prod  
(  
    id_prod int,  
    id_pedido int,  
    constraint ord_prod_pk primary key(id_prod,id_pedido),  
    constraint ord_prod_id_prod_fk foreign key(id_prod) references productos(id_prod),  
    constraint ord_prod_id_pedido_fk foreign key(id_pedido) references pedidos(id_pedido)  
);  
  
create table distribucion  
(  
    id_prod int,  
    id_local int,  
    cant_prod_loc int not null,  
    constraint distribucion primary key(id_prod,id_local),  
    constraint distribucion_id_prod_fk foreign key(id_prod) references productos(id_prod),  
    constraint distribucion_id_local_fk foreign key(id_local) references locales(id_local)  
);
```

### 3. SCRIPTS PARA POBLAR LA BASE DE DATOS.

```
INSERT INTO el_tinto_bendito.empleados
(cedula_emp, nombre_emp, apell_emp, movil_emp, correo_emp, cargo_emp, estado_emp, fec_ing_emp)
VALUES ('1010189765',
'JINNETH', 'MELO', '145678988', 'jinneth.melo@tintobendito.com', 'ADMINISTRADORA', 'A', '2021-08-13');

INSERT INTO el_tinto_bendito.empleados
(cedula_emp, nombre_emp, apell_emp, movil_emp, correo_emp, cargo_emp, estado_emp, fec_ing_emp)
VALUES ('1010189765',
'LAURA', 'CORREA', '3116894122', 'laura.correa@tintobendito.com', 'ADMINISTRADORA', 'A', '2021-08-18');

INSERT INTO el_tinto_bendito.empleados
(cedula_emp, nombre_emp, apell_emp, movil_emp, correo_emp, cargo_emp, estado_emp, fec_ing_emp)
VALUES ('55891255',
'NICOLE', 'ARIAS', '3162122143', 'nick.arias@tintobendito.com', 'ADMINISTRADORA', 'A', '2021-08-21');

INSERT INTO el_tinto_bendito.empleados
(cedula_emp, nombre_emp, apell_emp, movil_emp, correo_emp, cargo_emp, estado_emp, fec_ing_emp)
VALUES ('79685233',
'RICADO', 'FRANKLIN', '3015624655', 'ricardo.franklin@tintobendito.com', 'VENDEDOR', 'A', '2021-09-01');

INSERT INTO el_tinto_bendito.locales
(nombre_local, dir_local, tel_local, id_empleado)
VALUES ('PRINCIPAL', 'CL 85 # 9 -12 LOCAL 3-32', '3108594263', 3);

INSERT INTO el_tinto_bendito.locales
(nombre_local, dir_local, tel_local, id_empleado)
VALUES ('GRAN ESTACION', 'Ac. 26 # 62 - 47 LOCAL 2-21', '3108594263', 1);

INSERT INTO el_tinto_bendito.categorias (desc_categoria) VALUES ('BEBIDAS CON CAFE');

INSERT INTO el_tinto_bendito.categorias (desc_categoria) VALUES ('OTRAS BEBIDAS');

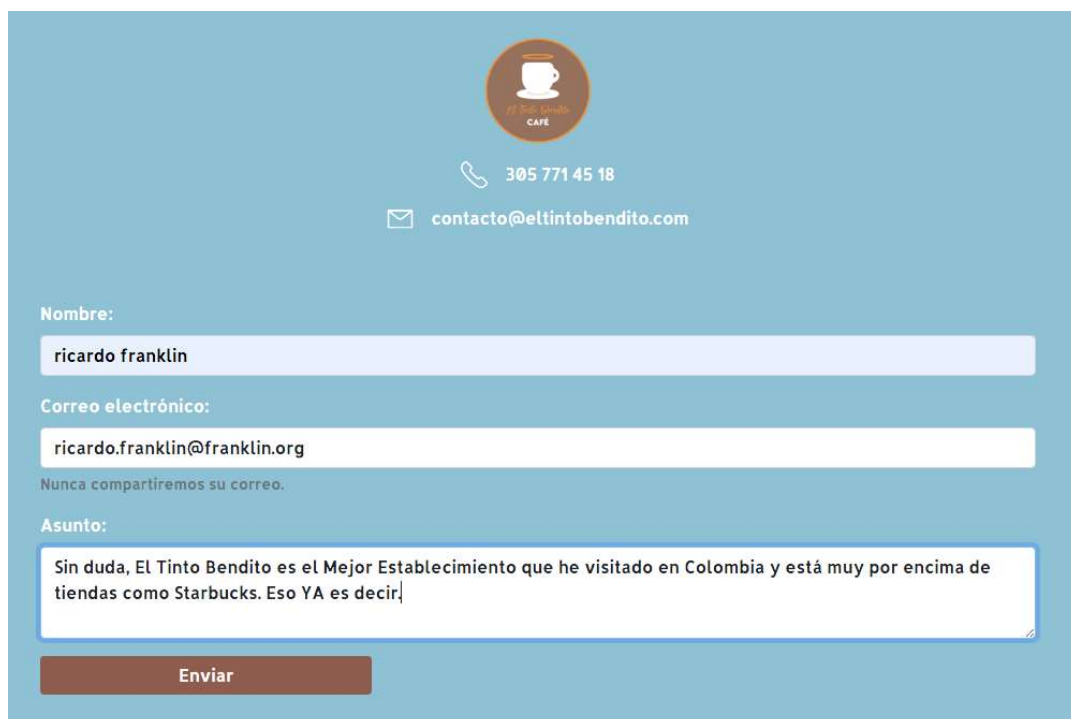
INSERT INTO el_tinto_bendito.categorias (desc_categoria) VALUES ('PASTELERIA');

INSERT INTO el_tinto_bendito.categorias (desc_categoria) VALUES ('CAFE GRANO');

INSERT INTO el_tinto_bendito.categorias (desc_categoria) VALUES ('CAFE MOLIDO');
```

#### 4. MANEJO DEL DOM CON JAVASCRIPT.

El sitio web de El Tinto Bendito ofrece a sus clientes un formulario de contacto de propósito general. Se muestra a continuación, el formulario diligenciado con algunos datos de prueba:



Nombre:

ricardo franklin

Correo electrónico:

ricardo.franklin@franklin.org

Nunca compartiremos su correo.

Asunto:

Sin duda, El Tinto Bendito es el Mejor Establecimiento que he visitado en Colombia y está muy por encima de tiendas como Starbucks. Eso YA es decir|

Enviar

Cuando el usuario hace clic sobre el botón Enviar entra en funcionamiento el script *index.js* capturando los datos ingresados por el usuario y mostrando estos como contenido de un elemento modal que es activado por el mismo script, tal y como se muestra en la siguiente imagen:

### Datos Recibidos

Nos pondremos en contacto con usted a la mayor brevedad.

**Nombre:** RICARDO FRANKLIN

**Correo:** ricardo.franklin@franklin.org

**Asunto:** Sin duda, El Tinto Bendito es el Mejor Establecimiento que he visitado en Colombia y está muy por encima de tiendas como Starbucks. Eso YA es decir.

Nombre:

ricardo franklin

Correo electrónico:

ricardo.franklin@franklin.org

Nunca compartiremos su correo.

Asunto:

Sin duda, El Tinto Bendito es el Mejor Establecimiento que he visitado en Colombia y está muy por encima de tiendas como Starbucks. Eso YA es decir.

Enviar

©2021 El Tinto Bendito

Cerrar

Cuando el usuario hace clic ahora sobre el botón Cerrar, el modal desaparece y los campos del formulario quedan limpios. Este es el código del archivo *index.js* con el que se implementa la funcionalidad mencionada:

```
const getElement = (selector) => {
  const element = document.querySelector(selector);
  if (element) return element
  throw Error(
    `No se encontró el elemento con la clase o ID ${selector}.`
  );
};

const txtFecha = getElement('#txtFecha');
const fechaActual = new Date().getFullYear();
const txtNombre = getElement('#txtNombre');
const txtEmail = getElement('#txtEmail');
const txtAsunto = getElement('#txtAsunto');
const btnEnviar = getElement('#btnEnviar');
const btnClose = getElement('#btnClose');
const modNom = getElement('#mod-nom');
const modMail = getElement('#mod-mail');
const modAsun = getElement('#mod-asun');
const txtFechaMod = getElement('#txtFechaMod');
const winModal = document.getElementById('modal')

btnClose.addEventListener('click', () => {
```

```
winModal.classList.toggle('visible');  
txtNombre.value = '';  
txtEmail.value = '';  
txtAsunto.value = '';  
}))  
  
btnEnviar.addEventListener('click', (ev)=> {  
  ev.preventDefault();  
  winModal.classList.toggle('visible');  
  txtFechaMod.textContent = fechaActual;  
  modNom.textContent = txtNombre.value.toUpperCase();  
  modMail.textContent = txtEmail.value;  
  modAsun.textContent = txtAsunto.value;  
})  
  
txtFecha.textContent = fechaActual;
```