

**Universidad Mariano Gálvez**

**Base de datos 2**

**Juan Carlos Monterroso**



**Proyecto: Sistema aduanas**

**Integrantes:**

**Jorge Raúl de León Campos 5090-17-3495**

**Eduardo Benjamín Lucero Schoenfeld 5090-17-15740**

**Luisa Fernanda González Morataya 5090-17-677**

**José Luis Adrián Orellana Zavala 5090-17-8991**

## Índice

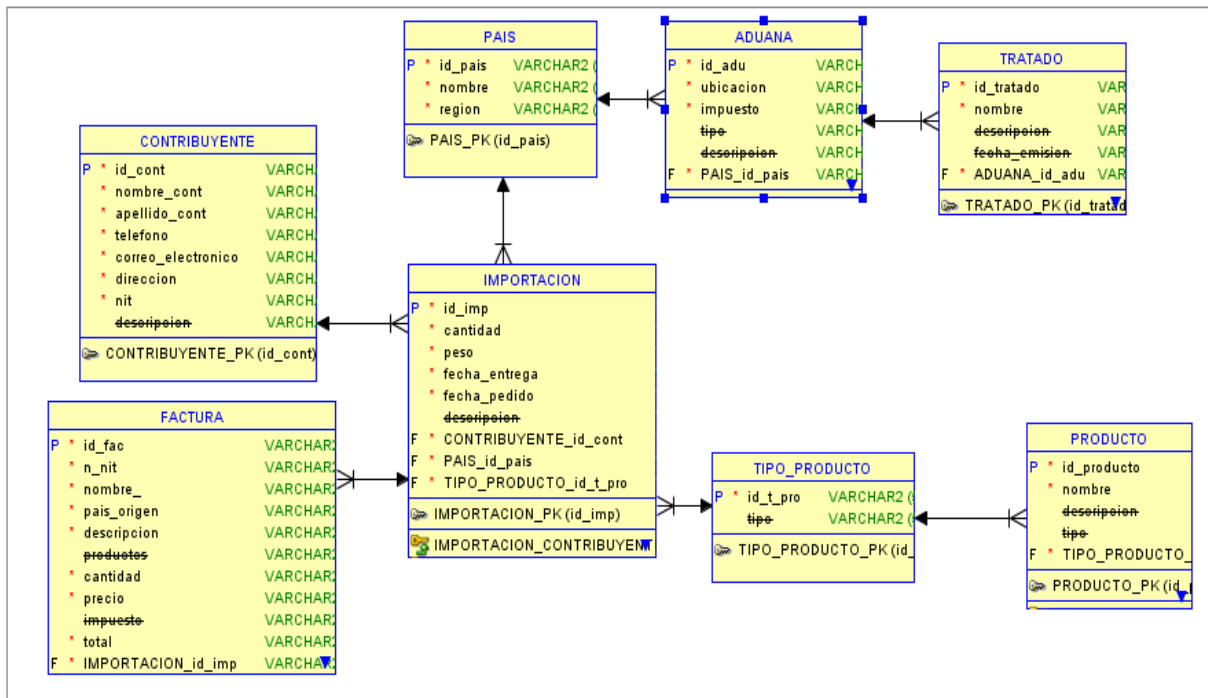
Contenido:	paginas
Introducción.....	3
Procedimiento de la base de datos.....	4-5
Puntos importantes de la aplicación asp-net.....	5-6
Códigos del proyecto.....	12-13
Conclusión.....	14

## **Introducción**

El programa consiste en un sistema de ingreso de mercancías, donde realiza un control de la información del contribuyente, mercancías que importan los contribuyentes, el país donde proviene las mercancías, la cantidad o el monto total de la importación y el impuesto que importa, por medio de los datos ingresados se estimara si la realización del ingreso se realizó correctamente, procurando que no se realizó ningún error, para luego indicando su acceso o la negación de la mercancía.

Se llevará un registro de transacciones que se ha realizado correctamente, para la observación de los datos de los contribuyentes o mercancías previamente ingresados.

## Procedimiento de la base de datos



### 1. Entidad de relación.

La base de datos consiste con 8 entidades las cuales poseen atributos y cada entidad tiene una llave principal. Cada entidad esta relacionada para que los códigos tengan una llave foránea y poder realizar consultas. Estas sirven para que un dato tenga un identificador único y de esa manera no poder crear un dato existente o que se copie alguno.

Algunos atributos tienen una línea ----- eso significa que son anulables sin embargo los que tienen " \* " no pueden ser nulos.

Indicadores:

P\* Llave principal.

F\* Llave foránea.

### 2.Base de datos

Esta base de datos permite almacenar información acerca de un contribuyente o usuario, el cual el realiza importaciones en el cual tiene un país, aduana y tratado. Al mismo tiempo se puede corroborar que tiene productos para hacer su encargo, finalmente se tomó referencia a la factura para poder realizar el pedido establecido siguiendo las normas de las aduanas y también de los precios con su respectivo cargo de impuesto, ya sea que lo tenga o no. La validación estará diseñada

con un semáforo que indicara con color verde si la transacción fue emitida y rojo si esta no cumple con los requisitos implementados. La aplicación se realizó en asp.net c#.

### **3.Funcionamiento de los triggers**

Se realizo 8 triggers y 8 secuencias, cada trigger contiene su propia secuencia para poder generar el id de cada entidad, esto para evitar que los usuarios que usen la pagina tengan acceso a dichos códigos. De esa manera se limitan a poder interactuar con ellos, al mismo tiempo sea más estético a la vista del usuario. Lo importante es facilitar la aplicación, no complicárselo más.

### **Puntos importantes de la aplicación asp-net.**

#### **1. Botones insert, update, delete, select.**

Cada formulario de la pagina web contiene todos los botones para que el trabajador pueda tener acceso para eliminar, insertar, modificar y eliminar cualquier registro siempre y cuando este seguro de que acción pueda tomar. Además, para la eliminación y modificación antes de ese procedimiento se realiza la consulta del código que se desea modificar o eliminar, todo eso para que tengan una certeza del dato que esta a punto de manipular.

#### **2. Validar letras y números en los textbox.**

Cada campo tiene su sentencia de validaciones, eso nos permite impedir que exista un nombre con números y números con letras en donde no debe existir, esto nos garantiza tener una base de datos y aplicación consistente.

#### **3. Semáforo**

El semáforo nos ayudara a evaluar si nuestro proceso se efectúa correctamente, si está en verde es porque los datos se subieron con éxito ahora si fuese el contrario y marca rojo, hay un error a la hora de validar un dato o alguno puede estar mal escrito y en el peor de los casos el código no le pertenezca al contribuyente que esta realizando su importación.

#### 4. Consistencia con la base de datos.

Una de las ventajas es que la pagina siempre estará conectada con la base de datos eso quiere decir que los datos se mantienen en tiempo real y las 24 horas del día para poder realizar cualquiera de los procesos mencionados en el objetivo numero uno.

#### 5. ¿Cómo obtener las llaves foráneas sin necesidad de escribir código?

Se utilizo un combobox el cual nos permite desplegar el nombre de la tabla a evaluar, posteriormente evalúa el nombre para mandar a imprimir el código principal al código secundario para que coincidan los datos que se están ingresando, de esa manera evitamos que la persona que escriba en la aplicación tenga libertad de colocar cualquier código. De esa manera los datos tienen sentido y orden a la hora de realizar consultas.

## Códigos de nuestro proyecto

### 1. Código de base de datos.

---

```
CREATE TABLE ADUANA
```

```
(  
  id_adu    VARCHAR2 (50 CHAR) NOT NULL ,  
  ubicacion VARCHAR2 (50 CHAR) NOT NULL ,  
  impuesto  VARCHAR2 (50 CHAR) NOT NULL ,  
  tipo      VARCHAR2 (50 CHAR) ,  
  descripcion VARCHAR2 (50 CHAR) ,  
  PAIS_id_pais VARCHAR2 (50 CHAR) NOT NULL  
);
```

```
ALTER TABLE ADUANA ADD CONSTRAINT ADUANA_PK PRIMARY KEY ( id_adu );
```

```
CREATE TABLE CONTRIBUYENTE
```

```
(  
  id_cont      VARCHAR2 (50 CHAR) NOT NULL ,  
  nombre_cont  VARCHAR2 (50 CHAR) NOT NULL ,  
  apellido_cont VARCHAR2 (50 CHAR) NOT NULL ,  
  telefono     VARCHAR2 (50 CHAR) NOT NULL ,  
  correo_electronico VARCHAR2 (50 CHAR) NOT NULL ,  
  direccion    VARCHAR2 (50 CHAR) NOT NULL ,  
  nit          VARCHAR2 (50 CHAR) NOT NULL ,  
  descripcion  VARCHAR2 (50 CHAR)  
);
```

```
ALTER TABLE CONTRIBUYENTE ADD CONSTRAINT CONTRIBUYENTE_PK PRIMARY KEY (  
id_cont );
```

```
CREATE TABLE FACTURA
```

```
(  
  id_fac      VARCHAR2 (50 CHAR) NOT NULL ,  
  n_nit       VARCHAR2 (50 CHAR) NOT NULL ,  
  nombre_     VARCHAR2 (50 CHAR) NOT NULL ,  
  pais_origen VARCHAR2 (50 CHAR) NOT NULL ,  
  descripcion VARCHAR2 (200 CHAR) NOT NULL ,  
  productos   VARCHAR2 (200 CHAR) ,  
  cantidad    VARCHAR2 (50 CHAR) NOT NULL ,  
  precio      VARCHAR2 (50 CHAR) NOT NULL ,  
  impuesto    VARCHAR2 (50 CHAR) ,  
  total       VARCHAR2 (50 CHAR) NOT NULL ,  
  IMPORTACION_id_imp VARCHAR2 (50 CHAR) NOT NULL  
);
```

```
ALTER TABLE FACTURA ADD CONSTRAINT FACTURA_PK PRIMARY KEY ( id_fac );
```

```
CREATE TABLE IMPORTACION
```

```
(  
  id_imp          VARCHAR2 (50 CHAR) NOT NULL ,  
  cantidad        VARCHAR2 (50 CHAR) NOT NULL ,  
  peso            VARCHAR2 (50 CHAR) NOT NULL ,  
  fecha_entrega   VARCHAR2 (50 CHAR) NOT NULL ,  
  fecha_pedido    VARCHAR2 (50 CHAR) NOT NULL ,  
  descripcion     VARCHAR2 (50 CHAR) ,  
  CONTRIBUYENTE_id_cont VARCHAR2 (50 CHAR) NOT NULL ,  
  PAIS_id_pais    VARCHAR2 (50 CHAR) NOT NULL ,  
  TIPO_PRODUCTO_id_t_pro VARCHAR2 (50 CHAR) NOT NULL  
);
```

```
ALTER TABLE IMPORTACION ADD CONSTRAINT IMPORTACION_PK PRIMARY KEY ( id_imp )  
;
```

```
CREATE TABLE PAIS
```

```
(  
  id_pais VARCHAR2 (50 CHAR) NOT NULL ,  
  nombre  VARCHAR2 (50 CHAR) NOT NULL ,  
  region  VARCHAR2 (50 CHAR) NOT NULL  
);
```

```
ALTER TABLE PAIS ADD CONSTRAINT PAIS_PK PRIMARY KEY ( id_pais );
```

```
CREATE TABLE PRODUCTO
```

```
(  
  id_producto    VARCHAR2 (50 CHAR) NOT NULL ,  
  nombre          VARCHAR2 (50 CHAR) NOT NULL ,  
  descripcion     VARCHAR2 (50 CHAR) ,  
  tipo            VARCHAR2 (50 CHAR) ,  
  TIPO_PRODUCTO_id_t_pro VARCHAR2 (50 CHAR) NOT NULL  
);
```

```
ALTER TABLE PRODUCTO ADD CONSTRAINT PRODUCTO_PK PRIMARY KEY ( id_producto );
```

```
CREATE TABLE TIPO_PRODUCTO
```

```
(  
  id_t_pro VARCHAR2 (50 CHAR) NOT NULL ,  
  tipo     VARCHAR2 (50 CHAR)  
);
```



```
ALTER TABLE TIPO_PRODUCTO ADD CONSTRAINT TIPO_PRODUCTO_PK PRIMARY KEY (
id_t_pro );
```

```
CREATE TABLE TRATADO
```

```
(
  id_tratado  VARCHAR2 (50 CHAR) NOT NULL ,
  nombre     VARCHAR2 (50 CHAR) NOT NULL ,
  descripcion VARCHAR2 (50 CHAR) ,
  fecha_emision VARCHAR2 (50 CHAR) ,
  ADUANA_id_adu VARCHAR2 (50 CHAR) NOT NULL
);
```

```
ALTER TABLE TRATADO ADD CONSTRAINT TRATADO_PK PRIMARY KEY ( id_tratado );
```

```
ALTER TABLE ADUANA ADD CONSTRAINT ADUANA_PAIS_FK FOREIGN KEY ( PAIS_id_pais )
REFERENCES PAIS ( id_pais );
```

```
ALTER TABLE FACTURA ADD CONSTRAINT FACTURA_IMPORTACION_FK FOREIGN KEY (
IMPORTACION_id_imp ) REFERENCES IMPORTACION ( id_imp );
```

```
ALTER TABLE IMPORTACION ADD CONSTRAINT IMPORTACION_CONTRIBUYENTE_FK
FOREIGN KEY ( CONTRIBUYENTE_id_cont ) REFERENCES CONTRIBUYENTE ( id_cont );
```

```
ALTER TABLE IMPORTACION ADD CONSTRAINT IMPORTACION_PAIS_FK FOREIGN KEY (
PAIS_id_pais ) REFERENCES PAIS ( id_pais );
```

```
ALTER TABLE IMPORTACION ADD CONSTRAINT IMPORTACION_TIPO_PRODUCTO_FK
FOREIGN KEY ( TIPO_PRODUCTO_id_t_pro ) REFERENCES TIPO_PRODUCTO ( id_t_pro );
```

```
ALTER TABLE PRODUCTO ADD CONSTRAINT PRODUCTO_TIPO_PRODUCTO_FK FOREIGN
KEY ( TIPO_PRODUCTO_id_t_pro ) REFERENCES TIPO_PRODUCTO ( id_t_pro );
```

```
ALTER TABLE TRATADO ADD CONSTRAINT TRATADO_ADUANA_FK FOREIGN KEY (
ADUANA_id_adu ) REFERENCES ADUANA ( id_adu );
```

---

## 2.Procedimientos dentro de la base de datos.

---

```
CREATE SEQUENCE uno
START WITH 509017000
INCREMENT BY 1;
```

```
CREATE TRIGGER tigre1
BEFORE INSERT ON CONTRIBUYENTE
FOR EACH ROW
BEGIN
    SELECT uno.NEXTVAL INTO :NEW.ID_CONT FROM DUAL;
END;
```

-----

```
CREATE SEQUENCE dos
START WITH 802017000
INCREMENT BY 1;
```

```
CREATE TRIGGER tigre2
BEFORE INSERT ON IMPORTACION
FOR EACH ROW
BEGIN
    SELECT dos.NEXTVAL INTO :NEW.ID_IMP FROM DUAL;
END;
```

-----

```
CREATE SEQUENCE tres
START WITH 1
INCREMENT BY 1;
```

```
CREATE TRIGGER tigre3
BEFORE INSERT ON PAIS
FOR EACH ROW
BEGIN
    SELECT tres.NEXTVAL INTO :NEW.ID_PAIS FROM DUAL;
END;
```

-----

```
CREATE SEQUENCE cuatro
START WITH 802017000
INCREMENT BY 1;
```

```
CREATE TRIGGER tigre4
BEFORE INSERT ON ADUANA
FOR EACH ROW
BEGIN
    SELECT cuatro.NEXTVAL INTO :NEW.ID_ADU FROM DUAL;
```

END;

-----  
CREATE SEQUENCE cinco  
START WITH 608017000  
INCREMENT BY 1;

CREATE TRIGGER tigre5  
BEFORE INSERT ON TRATADO  
FOR EACH ROW  
BEGIN  
SELECT cinco.NEXTVAL INTO :NEW.ID\_TRATADO FROM DUAL;  
END;

-----  
CREATE SEQUENCE seis  
START WITH 456817000  
INCREMENT BY 1;

CREATE TRIGGER tigre6  
BEFORE INSERT ON TIPO\_PRODUCTO  
FOR EACH ROW  
BEGIN  
SELECT seis.NEXTVAL INTO :NEW.ID\_P\_PRO FROM DUAL;  
END;

-----  
CREATE SEQUENCE siete  
START WITH 802517000  
INCREMENT BY 1;

CREATE TRIGGER tigre7  
BEFORE INSERT ON PRODUCTO  
FOR EACH ROW  
BEGIN  
SELECT siete.NEXTVAL INTO :NEW.ID\_PRODUCTO FROM DUAL;  
END;

-----  
  
CREATE SEQUENCE ocho  
START WITH 10001000  
INCREMENT BY 1;

```

CREATE TRIGGER tigre4
BEFORE INSERT ON FACTURA
FOR EACH ROW
    BEGIN
        SELECT ocho.NEXTVAL INTO :NEW.ID_FAC FROM DUAL;
    END;

create or replace procedure seleccionar (registro out sys_refcursor)
as
begin
open registro for select *from CONTRIBUYENTE;

```

---

### 3.Condiciones de letras, números.

AGREGAR EN LA PAGINA MAESTRA, EN LA ETIQUETA DE <script>

```

-----
function onKeyDecimal(e, thix) {

    var keynum = window.event ? window.event.keyCode : e.which;
    if (document.getElementById(thix.id).value.indexOf('.') != -1 && keynum == 46)
        return false;
    if ((keynum == 8 || keynum == 48 || keynum == 46))
        return true;
    if (keynum <= 47 || keynum >= 58) return false;
    return /\d/.test(String.fromCharCode(keynum));
}

function justNumbers(e) {
    var keynum = window.event ? window.event.keyCode : e.which;
    if ((keynum == 8 || keynum == 48))
        return true;
    if (keynum <= 47 || keynum >= 58) return false;
    return /\d/.test(String.fromCharCode(keynum));
}

function soloLetras(evt) {
    var charCode = (evt.which) ? evt.which : event.keyCode
    if (charCode > 31 && (charCode < 48 || charCode > 57))
        return true;
}

```

```

    return false;
}

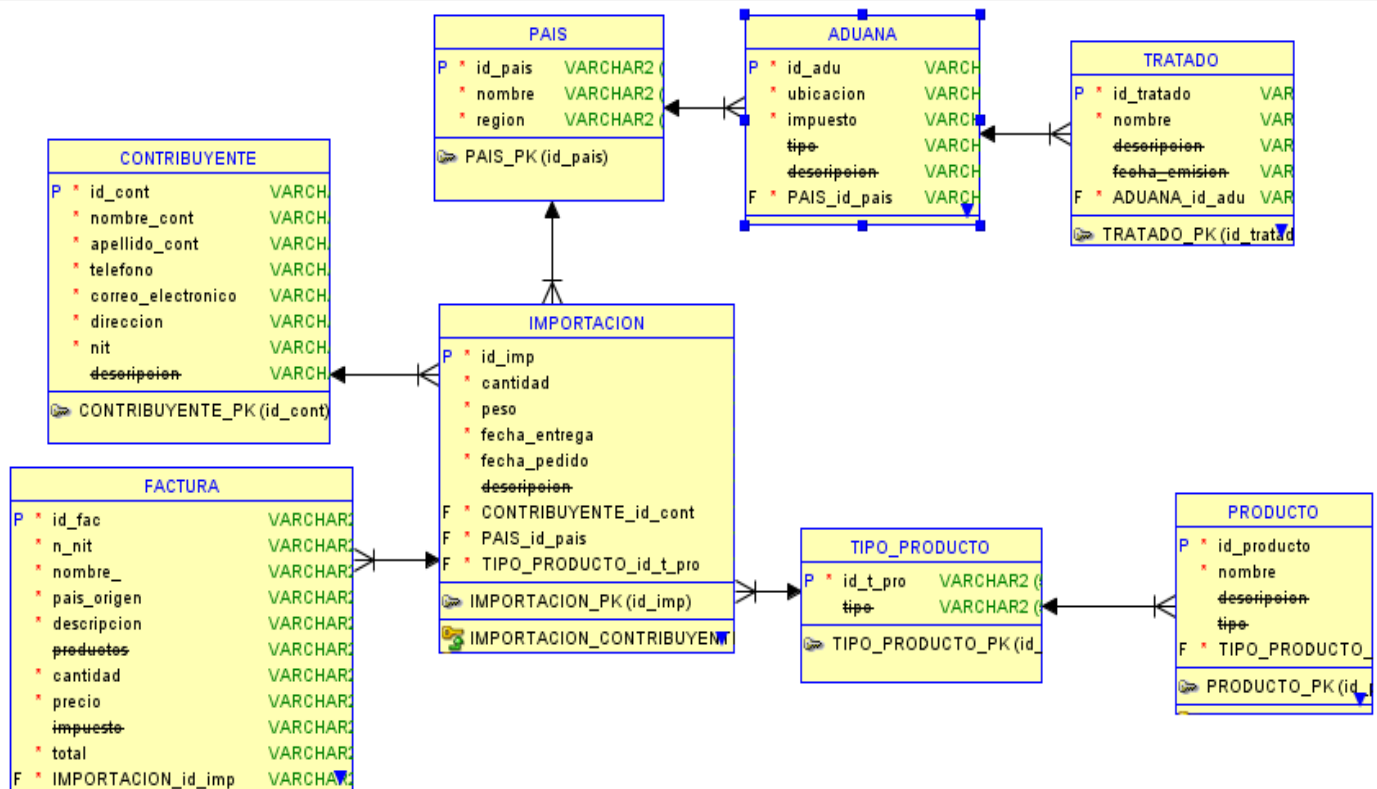
```

EN EL ASPX, EN EL CODIGO DEL TEXTBOX, COLOCAR ESTO DENTRO

PARA VALIDAR SOLO NUMEROS  
 onkeypress="return justNumbers(event);"

PARA VALIDAR SOLO LETRAS  
 onkeypress="return soloLetras(event);"

## 4.Data Modeler



## **Conclusión**

El sistema nos permitirá proporcionar un registro de mercancías, contribuyente y países con mayor importación, para la delimitación o la admisión de entrada. Facilitando la detección de errores en la ingesión de datos, evitando el ingreso incorrecto de productos no revisados o información del contribuyente no valida o detallada