**UTN-FRGP/TUP FINAL LABORATORIO DE COMPUTACIÓN III**

|  |  |  |
| --- | --- | --- |
| Nombre y Apellido: | | |
| DNI: | | Legajo: |
| Fecha: | | EXAMEN FINAL Tomo: Folio: |
| CALIFICACIÓN |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COMPETENCIAS ESPECIFICAS  EVALUADAS | **0**  **No lo**  **hace** | **2**  **Lo hace con errores sustanciales** | **4**  **Lo hace con errores que deben mejorarse** | **6**  **Lo hace correctamente**  **(Aceptable)** | **8**  **Lo hace muy bien** | **10**  **Lo hace excelente mente** |
| Aplicar vocabulario y terminología propia de la materia. |  |  |  |  |  |  |
| Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas. |  |  |  |  |  |  |
| Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores. |  |  |  |  |  |  |
| Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma |  |  |  |  |  |  |
| Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje. |  |  |  |  |  |  |

**UTN-FRGP/TUP FINAL LABORATORIO DE COMPUTACIÓN III**

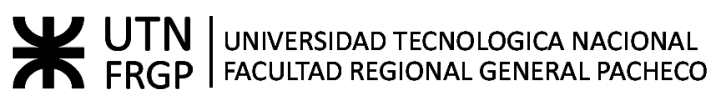
|  |  |  |
| --- | --- | --- |
| Nombre y Apellido: | | |
| DNI: | | Legajo: |
| Fecha: | | EXAMEN FINAL Tomo: Folio: |
| CALIFICACIÓN |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COMPETENCIAS ESPECIFICAS  EVALUADAS | **0**  **No lo**  **hace** | **2**  **Lo hace con errores sustanciales** | **4**  **Lo hace con errores que deben mejorarse** | **6**  **Lo hace correctamente**  **(Aceptable)** | **8**  **Lo hace muy bien** | **10**  **Lo hace excelente mente** |
| Aplicar vocabulario y terminología propia de la materia. |  |  |  |  |  |  |
| Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas. |  |  |  |  |  |  |
| Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores. |  |  |  |  |  |  |
| Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma |  |  |  |  |  |  |
| Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje. |  |  |  |  |  |  |

**UTN-FRGP/TUP FINAL LABORATORIO DE COMPUTACIÓN III**

|  |  |  |
| --- | --- | --- |
| Nombre y Apellido: | | |
| DNI: | | Legajo: |
| Fecha: | | EXAMEN FINAL Tomo: Folio: |
| CALIFICACIÓN |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COMPETENCIAS ESPECIFICAS  EVALUADAS | **0**  **No lo**  **hace** | **2**  **Lo hace con errores sustanciales** | **4**  **Lo hace con errores que deben mejorarse** | **6**  **Lo hace correctamente**  **(Aceptable)** | **8**  **Lo hace muy bien** | **10**  **Lo hace excelente mente** |
| Aplicar vocabulario y terminología propia de la materia. |  |  |  |  |  |  |
| Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas. |  |  |  |  |  |  |
| Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores. |  |  |  |  |  |  |
| Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma |  |  |  |  |  |  |
| Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje. |  |  |  |  |  |  |



**INFORME DE PROYECTO**

**Revestimientos A&T**



**Integrantes:**

Roberto Jeremías Galeano.

Federico Ivan Parrotta.

27/06/2020

Primer Cuatrimestre.

UTN FRGP - Laboratorio de Computación III

Profesores  Ariel Herrera , Diego Español.

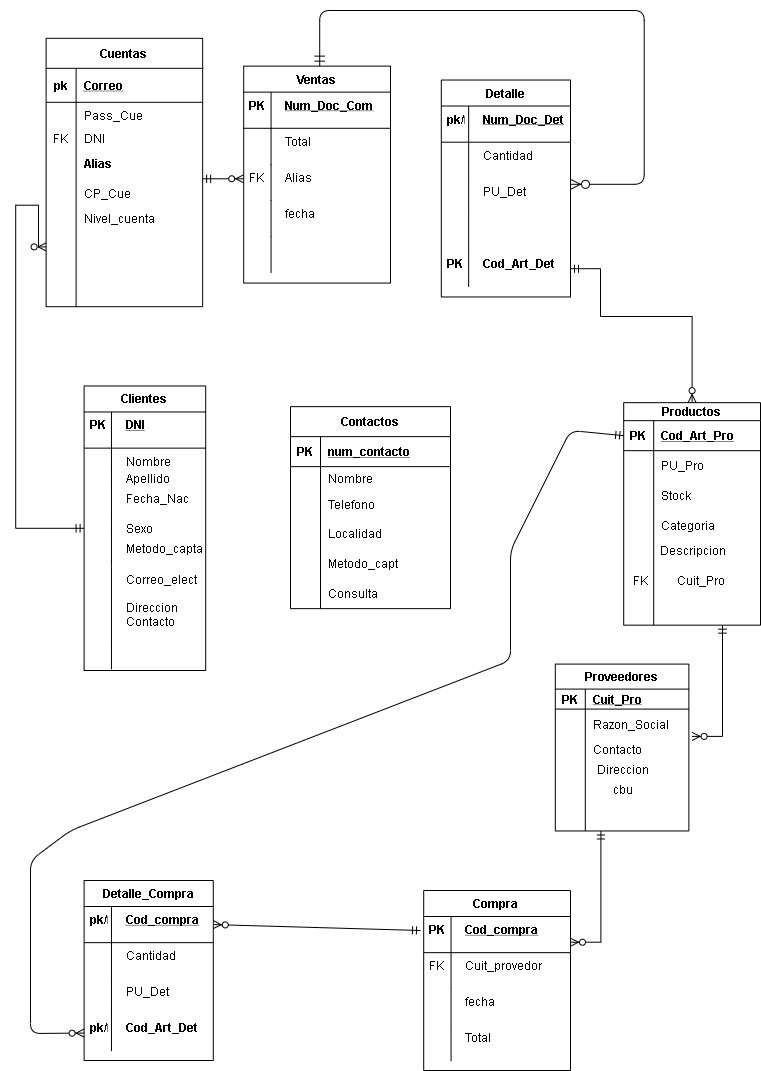
**Índice**

**Introducción.**

El proyecto consta de la creación de una base de datos requerida para el funcionamiento del área de ventas online del negocio “A&TI revestimientos”.

En la base de datos se almacenará información relacionada a los clientes , los presupuestos (solicitados en caso de necesitar atención personalizada), las ventas, las cuentas de los clientes (necesarias para poder efectuar compras en la plataforma online), los distintos productos a la venta además de información adicional de los mismos para que el cliente pueda navegar por el catálogo de “productos” del sistema (por ejemplo un enlace con el que adjunte una imagen , precio por metro entre otros datos), un apartado de proveedores para poder facilitar el acceso a la información necesaria para efectuar pedidos en caso de necesitar mayor stock en el negocio.

**DER**



**Creación de base de datos:**

--delete

drop database Revestimientoayt

create database Revestimientoayt

GO

USE  Revestimientoayt

GO

--EN CLIENTES NO APARECE LA OPCION DE CARGAR FECHA DE NACIMIENTO RECOMIENDO BORRARLO

--METODO CAPTACION ME PARECE INECESARIO PERO NO DEJO PARA QUE PUEDA SER NULO

 CREATE TABLE Clientes (

Correo\_Cli  VARCHAR (20) NOT NULL,

Nombre char(12) NOT NULL,

 Apellido char(20) NOT NULL,

Fecha\_Nac  smalldatetime  default getdate(),

Sexo CHAR(8)NOT NULL,

Metodo\_captacion VARCHAR(20),

Dni CHAR(8)NOT NULL,

Direccion VARCHAR (20) NOT NULL,

Usuario\_Cli CHAR(8)NOT NULL,

CONSTRAINT PK\_Clientes PRIMARY KEY  (Dni)

  )

  --EN CUENTAS  PUEDE SER NULO EL CP Y EL NIVEL CUENTA

create TABLE Cuentas(

Correo VARCHAR (20) NOT NULL,

Pass\_Cue VARCHAR(20)NOT NULL,

Dni CHAR (8) NOT NULL,

Alias CHAR(20)NOT NULL,

CP\_Cue CHAR(8),

Nivel\_cuenta  int ,

CONSTRAINT PK\_Cuentas PRIMARY KEY  (Correo),

CONSTRAINT FK\_Cuentas\_clientes foreign KEY  (Dni) references  Clientes (dni)

)

   CREATE TABLE Ventas (

  Num\_Doc\_Com  int IDENTITY (1,1) NOT NULL,

Total  DECIMAL(8,2) DEFAULT '0,00',

 Alias char(8) NOT NULL,

Fecha  smalldatetime  default getdate(),

   CONSTRAINT PK\_Ventas PRIMARY KEY  (Num\_Doc\_Com),

    )

 create table Detalle (

  Num\_Doc\_Det int not null,

Cantidad char(8) NOT NULL,

 PU\_Det DECIMAL(8,2) DEFAULT '0,00',

   Cod\_Art\_Det char(8) NOT NULL,

  CONSTRAINT PK\_Detalle PRIMARY KEY  ( Cod\_Art\_Det,Num\_Doc\_Det)

    )

   CREATE TABLE Productos (

   Cod\_Art\_Pro char(8) NOT NULL,

 PU\_Pro DECIMAL(8,2) DEFAULT '0,00',

 Stock int  NOT NULL,

  Categoria char(30),

  Descripcion varchar(100) ,

  Url\_imagen varchar(100)

     CONSTRAINT PK\_Productos PRIMARY KEY  ( Cod\_Art\_Pro)

    )

       CREATE TABLE Proveedores (

   Cuit\_Pro char(8) NOT NULL,

 Razon\_Social char(8) NOT NULL,

 Contacto char(8) NOT NULL,

 Direccion VARCHAR (20) NOT NULL,

 cbu char(8) NOT NULL,

    CONSTRAINT PK\_Proveedores PRIMARY KEY  ( Cuit\_Pro)

    )

       CREATE TABLE Compra  (

  Cod\_compra int IDENTITY (1,1) NOT NULL,

Total  DECIMAL(8,2) DEFAULT '0,00',

 cuit\_proveedor char(8) NOT NULL,

Fecha  smalldatetime  default getdate(),

   CONSTRAINT PK\_Compra PRIMARY KEY  (Cod\_compra),

   CONSTRAINT FK\_Compra\_Proveedores foreign KEY  (cuit\_proveedor) references  Proveedores (Cuit\_Pro)

    )

     create table Detalle\_compra (

  Num\_Doc\_Det  int NOT NULL,

Cantidad char(8) NOT NULL,

 PU\_Det DECIMAL(8,2) DEFAULT '0,00',

  Cod\_Art\_Det char(8) NOT NULL,

  CONSTRAINT PK\_Detalle\_compra PRIMARY KEY  ( Cod\_Art\_Det,Num\_Doc\_Det)

  )

   create table Franquicias(

     Id\_Sucursal char(8)not null,

      NombreSucursal  varchar (50) not null,

      DescripcionSucursal varchar (50) not null,

      HorarioSucursal varchar(50) not null,

      ProvinciaSucursal varchar (50) not null,

      DireccionSucursal varchar (50) not null,

      CONSTRAINT PK\_Franquicia PRIMARY KEY  ( Id\_Sucursal)

    )

 go

 ;

**Creacion de procedimientos almacenados:**

create procedure  spActualizarProductos

(

@Cod\_art\_pro char (8),

@PU\_Pro Decimal,

@stock int,

@Url\_imagen varchar (100)

)

AS

update Productos

set

PU\_Pro=@PU\_Pro,

Stock=@stock,

 Url\_imagen=@Url\_imagen

 where  Productos.Cod\_Art\_Pro=@Cod\_art\_pro

RETURN

--------------------------------------------------------------------------------------------------------------------------

--procedimineto eliminar producto

 Create procedure  speliminarproc

(@Cod\_art\_pro char(8))

as

delete  from  productos

where Cod\_Art\_Pro = @Cod\_art\_pro

--------------------------------------------------------------------------------------------------------------------------

---procedimiento agregar producto

create  PROCEDURE spinsertarprod

(

@Cod\_art\_pro char (8),

@PU\_Pro Decimal,

@stock int,

@categoria char(30),

@descripcion varchar(30),

@Url\_imagen varchar (100)

)

AS

INSERT INTO Productos

(

PU\_Pro,

Cod\_Art\_Pro,

Stock,

Categoria,

Descripcion,

 Url\_imagen

)

VALUES

(

@Cod\_art\_pro,

@PU\_Pro,

@stock ,

@categoria,

@descripcion,

@Url\_imagen

)RETURN

--------------------------------------------------------------------------------------------------------------------------

--procedimiento modificar cuenta

create  PROCEDURE spinsertarprod

(

@pass\_cue CHAR(8),

@Alias char(8)

)

AS

update Cuentas

 Set Pass\_Cue =@pass\_cue

 where  Cuentas.Alias=@Alias

RETURN

---------------------------------------------------------------------------------------------------------------------------

create  PROCEDURE spcomprar(

@alias char(8),

@pu decimal(8,2)

)

as

begin

begin try

begin transaction

insert into ventas values(@pu, @alias, getdate())

--SI NO ACTUALIZO NINGUNA FILA

IF @@ROWCOUNT = 0 BEGIN

RAISERROR('OCURRIO UN ERROR', 16, 1)

END

commit transaction

end try

BEGIN CATCH

PRINT ERROR\_MESSAGE()

END CATCH

end

---------------------------------------------------------------------------------------------------------------------------

create  PROCEDURE spinsertarcliente(

(

@correo\_cli VARCHAR(20),

@nombre VARCHAR(20),

@apellido CHAR(8),

@sexo char,

@Dni CHAR(8),

@Direccion VARCHAR (20) ,

@Usuario\_Cli CHAR(8)

)

AS

INSERT INTO Clientes

(

Correo\_Cli,

Nombre,

Apellido,

Sexo,

Dni,

Direccion,

Usuario\_Cli

)

VALUES

(

@correo\_cli,

@nombre,

@apellido,

@sexo,

@Dni,

@Direccion,

@Usuario\_Cli

)RETURN

**Trigger 1**

create TRIGGER ACTUALIZASTOCK

ON Detalle\_compra

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

UPDATE productos set Stock = Stock + (SELECT Cantidad FROM

INSERTED)

where Cod\_Art\_Pro = (SELECT Cod\_Art\_Det FROM INSERTED)

end

**Trigger 2**

create TRIGGER ACTUALIZASTOCK2

ON Detalle

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

UPDATE productos set Stock = Stock - (SELECT Cantidad FROM

INSERTED)

where Cod\_Art\_Pro = (SELECT Cod\_Art\_Det FROM INSERTED)

end

**Consulta**

**--**

Obtiene nombre y apellidos para poder obtener por ejemplo un promedio de edad

select Nombre , Apellido , Fecha\_Nac from Clientes

inner join Cuentas on Clientes.Dni = Cuentas.Dni

inner join Ventas on Cuentas.Alias = Ventas.Alias

---

obtiene nombre , apellido y fecha de nacimiento de un determinado usuario que halla realizado una compra

select distinct Nombre , Apellido , Fecha\_Nac from Clientes

inner join Cuentas on Clientes.Dni = Cuentas.Dni

inner join Ventas on Cuentas.Alias = Ventas.Alias

where Ventas.Alias = 'juan5'

--consigue toda la informacion de un determinado alias.

select \* from Cuentas where Alias = 'juan5'

--consigue toda la informacion de una determinada contraseña

select \* from Cuentas where Pass\_Cue = 'pass1'

las dos combinadas se utilizan para validar que la contraseña y la cuenta

al momento de login

--obtiene todos los productos que contengan una decripcion general a la ingresada

select \* from Productos where Descripcion like '%pin%'

--obtiene toda la informacion de un cliente mediante su dni

select \* from Clientes where Dni = '40333333'

--obtiene toda la informacion de un producto mediante su codigo de articulo

select \* from Productos where Cod\_Art\_Pro = 'cie2'