**Instituto Tecnológico de León**

**Integrantes:**

Luis Adrian Casillas Ornelas

Leonardo Missael Hernandez Calvillo

Luis Brandon Torres Rocha

**Profesor:**

Luz del Carmen Ruiz Gaytán

**Taller de Base de Datos**

**Proyecto Final: Conectividad a la BD**

**7/07/2021**

Contenido

[**Enunciado semántico:** 3](#_Toc76454005)

[**Modelo entidad relación:** 3](#_Toc76454006)

[**Modelo relacional:** 4](#_Toc76454007)

[**Script:** 4](#_Toc76454008)

[**Conexión a la BD con un lenguaje de programación:** 15](#_Toc76454009)

[**Pagina de inicio:** 15](#_Toc76454010)

[**Consulta:** 16](#_Toc76454011)

[**Enunciado:** 16](#_Toc76454012)

[**Modelo entidad relación** 17](#_Toc76454013)

[**Modelo relacional:** 17](#_Toc76454014)

[**Código de la conexión a la BD:** 18](#_Toc76454015)

# **Enunciado semántico:**

# **Modelo entidad relación:**

# **Modelo relacional:**

# **Script:**

use master

create database residencias;

use residencias;

-----------------------**Creacion de la BD**--------------------------------

--Se crea la tabla alumno

create table alumno(

No\_control char(8) not null,

Nombre\_alum varchar(30),

Telefono char(10),

Correo varchar(50),

Semestre char(2) not null,

Creditos int not null

)

--Se establece la llave primaria

alter table alumno add constraint pk\_alumno primary key(No\_control)

--Se llena la tabla alumno

select \* from alumno

insert into alumno values

('17240666','Adrian', '4778964523', 'a@gmail.com','10',190)

insert into alumno values

('17425464','Andrea', '4778568887', 'b@gmail.com','9',200)

insert into alumno values

('19795486','Eduardo', '4774865403', 'c@gmail.com','11',200)

insert into alumno values

('16856489','Paola', '4778963245', 'd@gmail.com','10',195)

insert into alumno values

('17986536','Miguel', '4798546326', 'e@gmail.com','12',205)

--Se crea la tabla empresa

create table empresa(

Clave\_E char(5) not null,

Nombre\_emp varchar(30),

Direccion varchar(30))

--Se llena la tabla Empresa

select \* from empresa

insert into empresa values

('E-896', 'INDESYTH', 'Av. Miguel Hidalgo #2')

insert into empresa values

('E-892', 'INDEX', 'Av. Republica #452')

insert into empresa values

('E-452', 'SEIQ S. DE R.L. DE C.V', 'CIRCUITO EL MARQUÉS SUR # 12')

insert into empresa values

('E-798', 'IHFES', 'ADMINISTRACIÓN DE FINANC.')

insert into empresa values

('E-895', 'AURIMÉDICA S DE SR DE CV', 'ARMANDO BIRLAIN 2001')

--Se establece su llave primaria

alter table empresa add constraint pk\_empresa primary key(Clave\_E)

--Se crea la tabla proyecto

create table proyecto(

No\_proy char(5) not null,

Nombre\_proy varchar(30),

area varchar(30),

clave\_emp char(5))

--Se llena la tabla proyecto

select \* from proyecto

insert into proyecto values

('Q-546', 'MINI DRONE DE USO RECREATIVO', 'Software', 'E-896')

insert into proyecto values

('Q-547', 'PROGRAMACION IOS PARA DRONES', 'Software', 'E-892')

insert into proyecto values

('Q-548', 'ELABORACION SITIO WEB CASE IH', 'Sitio Web', 'E-452')

insert into proyecto values

('Q-549', 'PRESUPUESTO AUTORIZADO', 'Software', 'E-798')

insert into proyecto values

('P-256', 'CONSULTA ON LINE', 'Software', 'E-895')

--Se estableve la llave primaria

alter table proyecto add constraint pk\_proyecto primary key(No\_proy)

--Se establece la llave foranea

alter table proyecto add constraint fk\_proyecto

foreign key (clave\_emp) references empresa(Clave\_E)

--Se crea la tabla telefono

create table telefono(

Clave\_E char(5)not null,

No\_tel char(10) not null,

tipo varchar(20)

)

--Se llena la tabla telefono

select \* from telefono

insert into telefono values ('E-896', '4778965423','Oficina')

insert into telefono values ('E-892', '4778964789','Oficina')

insert into telefono values ('E-452', '4785463259','Oficina')

insert into telefono values ('E-798', '4795969872','Oficina')

insert into telefono values ('E-895', '4775786423','Oficina')

--Se estableve la llave primaria

alter table telefono add constraint pk\_telefono primary key(No\_tel)

--Se establece la llave foranea

alter table telefono add constraint fk\_telefono

foreign key (Clave\_E) references empresa(Clave\_E)

--Se crea la tabla realiza

create table realiza(

No\_control char(8) not null,

No\_proy char(5) not null,

periodo varchar(20)

)

--Se estableve la llave foranea

alter table realiza add constraint fk\_alumno

foreign key(No\_control) references alumno(No\_control)

alter table realiza add constraint fk\_realiza2

foreign key(No\_proy) references proyecto(No\_proy)

--Se llena la tabla telefono

select \* from realiza

insert into realiza values ('17240666', 'Q-546','Enero-Junio')

insert into realiza values ('17425464', 'Q-547','Enero-Junio')

insert into realiza values ('19795486', 'Q-548','Agosto-Diciembre')

insert into realiza values ('16856489', 'Q-549','Agosto-Diciembre')

insert into realiza values ('17986536', 'P-256','Enero-Junio')

--Se crea la tabla asesor

create table asesor(

Clave\_A char(5) not null,

Nombre\_A char(40) not null,

)

--Se estableve la llave primaria

alter table asesor add constraint pk\_asesor primary key(Clave\_A)

--Se llena la tabla asesor

select \* from asesor

insert into asesor values ('A-001', 'Laura Juarez Guerra')

insert into asesor values ('A-002', 'Gerardo Carpio')

insert into asesor values ('A-003', 'Ruth Saenz')

insert into asesor values ('A-004', 'Cirino Silva')

insert into asesor values ('A-005', 'Edna Militza')

insert into asesor values ('E-789', 'Moises Elias Perez')

insert into asesor values ('E-790', 'Pedro Gonzalez')

insert into asesor values ('E-791', 'Omar Palomares')

insert into asesor values ('E-792', 'Martin Mora')

insert into asesor values ('E-793', 'Jorge Cosileon')

--Se crea la tabla interno

create table interno(

Clave\_A char(5) not null,

Horas\_asig char(3) not null,

Grado varchar(20)

)

--Llenar la tabla interno

select \* from interno

insert into interno values ('A-001', '100','Maestria')

insert into interno values ('A-002', '120','Doctorado')

insert into interno values ('A-003', '70','Maestria')

insert into interno values ('A-004', '100','Maestria')

insert into interno values ('A-005', '105','Maestria')

--Se estableve la llave primaria

alter table interno add constraint fk\_interno

foreign key(Clave\_A) references asesor(Clave\_A)

--Se crea la tabla externo

create table externo(

Clave\_A char(5) not null,

cargo varchar(30) not null,

Clave\_E char(5)

)

--Llenar la tabla externo

select \* from externo

insert into externo values ('E-789', 'Director','E-896')

insert into externo values ('E-790', 'Jefe de Sistemas','E-892')

insert into externo values ('E-791', 'Asesor Web','E-452')

insert into externo values ('E-792', 'Encargado Sistemas','E-798')

insert into externo values ('E-793', 'Jefe de Sistemas','E-895')

--Se crea llave foranea

alter table externo add constraint fk\_externo

foreign key(Clave\_A) references asesor(Clave\_A)

alter table externo add constraint fk\_externo2

foreign key(Clave\_E) references empresa(Clave\_E)

--Se crea la tabla tiene\_asesor

create table tiene\_asesor(

No\_control char(8) not null,

Clave\_A char(5) not null

)

--Se llena tabla tiene\_asesor

select \* from tiene\_asesor

insert into tiene\_asesor values ('17240666', 'A-001')

insert into tiene\_asesor values ('17425464', 'A-002')

insert into tiene\_asesor values ('19795486', 'A-003')

insert into tiene\_asesor values ('16856489', 'A-004')

insert into tiene\_asesor values ('17986536', 'A-005')

insert into tiene\_asesor values ('17240666', 'E-789')

insert into tiene\_asesor values ('17425464', 'E-790')

insert into tiene\_asesor values ('19795486', 'E-791')

insert into tiene\_asesor values ('16856489', 'E-792')

insert into tiene\_asesor values ('17986536', 'E-793')

--Se crea llave foranea

alter table tiene\_asesor add constraint fk\_tiene\_asesor1

foreign key(No\_control) references alumno(No\_control)

alter table tiene\_asesor add constraint fk\_tiene\_asesor2

foreign key(Clave\_A) references asesor(Clave\_A)

----------------------**Creacion de Vistas**-----------------------

--Muestra el numero de control, el nombre del alumno y el nombre proyecto

CREATE VIEW vwAlumno\_Proyecto

WITH ENCRYPTION

AS

SELECT RE.No\_control, Nombre\_alum, Nombre\_proy

FROM (alumno AS AL JOIN realiza AS RE ON AL.No\_control = RE.No\_control)

JOIN proyecto AS PO ON RE.No\_proy = PO.No\_proy

--Consulta

select \* from vwAlumno\_Proyecto

--Muestra el numero de control, el nombre del proyecto, y quien asesora el proyecto

CREATE VIEW vwAsesor\_Proyecto

WITH ENCRYPTION

AS

SELECT RE.No\_control, AE.Nombre\_A AS 'Nombre Asesor', PO.Nombre\_proy

FROM realiza AS RE

JOIN tiene\_asesor AS TA ON TA.No\_control = RE.No\_control

JOIN asesor AS AE ON TA.Clave\_A = AE.Clave\_A

JOIN proyecto AS PO ON PO.No\_proy = RE.No\_proy

--Consulta

SELECT \* FROM vwAsesor\_Proyecto

--Muestra empresa, proyecto y asesor de la empresa

CREATE VIEW vwEmpresa\_Proyecto

WITH ENCRYPTION

AS

SELECT Nombre\_emp, Nombre\_proy, Nombre\_A

FROM (empresa AS EM JOIN proyecto AS PO ON EM.Clave\_E = PO.clave\_emp)

JOIN externo AS EX ON EM.Clave\_E = EX.Clave\_E

JOIN asesor AS AE ON EX.Clave\_A = AE.Clave\_A

--Consulta

SELECT \* FROM vwEmpresa\_Proyecto

--Muestra el numero de control, nombre del proyecto y el periodo en el que se realiza

CREATE VIEW vwProyecto\_Periodo

WITH ENCRYPTION

AS

SELECT RE.No\_control, PO.Nombre\_proy, RE.periodo

FROM realiza AS RE JOIN proyecto AS PO ON RE.No\_proy = PO.No\_proy

--Consulta

SELECT \* FROM vwProyecto\_Periodo

--Muestra nombre del alumno, el numero de creditos y el semestre que cursa

CREATE VIEW vwInfo\_Alumno

WITH ENCRYPTION

AS

SELECT Nombre\_alum, Creditos, Semestre

FROM alumno

--Consulta

SELECT \* FROM vwInfo\_Alumno

--Muestra nombre de la empresa, direccion y numero de contacto

CREATE VIEW vwInfo\_Empresa

WITH ENCRYPTION

AS

SELECT EM.Nombre\_emp, EM.Direccion, TE.No\_tel

FROM empresa AS EM JOIN telefono AS TE ON EM.Clave\_E = TE.Clave\_E

--Consulta

SELECT \* FROM vwInfo\_Empresa

--Muestra la informacion de contacto del alumno

CREATE VIEW vwContacto\_Alumno

WITH ENCRYPTION

AS

SELECT Nombre\_alum, Telefono, Correo

FROM alumno

--Consulta

SELECT \* FROM vwContacto\_Alumno

--Muestra los asesores internos y el grado academico que tienen

CREATE VIEW vwAsesorInt\_Info

WITH ENCRYPTION

AS

SELECT AR.Nombre\_A, IR.Grado

FROM interno AS IR JOIN asesor AS AR ON IR.Clave\_A = AR.Clave\_A

--Consulta

SELECT \* FROM vwAsesorInt\_Info

-----------------**Creacion de Usuarios**--------------------------

--Se crean los usuarios de la base de datos

CREATE LOGIN escuela WITH PASSWORD = '1234';

CREATE LOGIN docente WITH PASSWORD = '1234';

CREATE LOGIN alumno WITH PASSWORD = '1234';

--Se crean los usuarios correspondientes a la BD

CREATE USER institucion FOR LOGIN escuela;

CREATE USER profesor FOR LOGIN docente;

CREATE USER estudiante FOR LOGIN alumno;

--Se crea un rol dueño para el usuario 'institucion'

CREATE ROLE escuela AUTHORIZATION db\_owner;

--Se agrega el usuario al rol que le corresponde

EXEC sp\_addrolemember 'escuela','institucion';

--Se crea un rol escritor para el usuario 'profesor'

CREATE ROLE docente AUTHORIZATION db\_datawriter;

--Se crea un rol lector para el usuario 'profesor'

CREATE ROLE docenteL AUTHORIZATION db\_datareader;

--Se agrega el usuario al rol que le corresponde

EXEC sp\_addrolemember 'docente','profesor';

EXEC sp\_addrolemember 'docenteL','profesor';

--Se establecen los permisos

DENY ALTER ON alumno to docente;

DENY ALTER ON asesor to docente;

DENY ALTER ON empresa to docente;

DENY ALTER ON externo to docente;

DENY ALTER ON interno to docente;

DENY ALTER ON proyecto to docente;

DENY ALTER ON realiza to docente;

DENY ALTER ON telefono to docente;

DENY ALTER ON tiene\_asesor to docente;

--Se crea un rol dueño para el usuario 'profesor'

CREATE ROLE alumno AUTHORIZATION db\_datareader;

--Se agrega el usuario al rol que le corresponde

EXEC sp\_addrolemember 'alumno','estudiante';

-------------------**TEST DE PRIVILEGIOS**----------------------------

USE residencias;

--El Usuario3 (estudiante intenta cambiar datos de la BD)

SELECT \* FROM alumno;

UPDATE alumno SET Nombre\_alum = 'Luis' WHERE Nombre\_alum = 'Adrian'

--El Usuario1 (institucion intenta cambiar datos de la BD)

SELECT \* FROM alumno;

UPDATE alumno SET Nombre\_alum = 'Luis' WHERE Nombre\_alum = 'Adrian'

--El Usuario2 (profesor intenta borrar o cambiar una base de datos)

DROP TABLE alumno;

ALTER TABLE alumno ADD Email2 varchar(255);

-------------------------**Transacciones**----------------------------

/\*Se maneja una transaccón simple debido a que en la base de datos

no se encontró una situación donde se necesitara una transacción.\*/

BEGIN TRANSACTION;

USE residencias

SELECT \* FROM alumno

SELECT \* FROM realiza

COMMIT;

-------------------**Procedimientos almacenados**----------------------

--Procedimiento ingresar registros a la tabla alumno

CREATE PROCEDURE ingresarAlumno

@No\_control char(8),

@Nombre\_alum varchar(30),

@Telefono char(10),

@Correo varchar(50),

@Semestre char(2),

@Creditos int

AS

insert into alumno values (@No\_control,@Nombre\_alum,@Telefono,

@Correo,@Semestre, @Creditos)

GO

--Procedimiento ingresar registros a la tabla realiza

CREATE PROCEDURE ingresarRealiza

@No\_control char(8),

@No\_proy char(5),

@periodo varchar(20)

AS

insert into realiza values (@No\_control,@No\_proy,@periodo)

GO

--Procedimiento ingresar registros a la tabla proyecto

CREATE PROCEDURE ingresarProyecto

@No\_proy char(5),

@Nombre\_proy varchar(30),

@area varchar(30),

@clave\_emp char(5)

AS

insert into proyecto values (@No\_proy,@Nombre\_proy,@area, @clave\_emp)

GO

--Procedimiento ingresar registros a la tabla empresa

CREATE PROCEDURE ingresarEmpresa

@Clave\_E char(5),

@Nombre\_emp varchar(30),

@Direccion varchar(30)

AS

insert into empresa values (@Clave\_E,@Nombre\_emp,@Direccion)

GO

--Procedimiento ingresar registros a la tabla telefono

CREATE PROCEDURE ingresarTelefono

@Clave\_E char(5),

@No\_tel char(10),

@tipo varchar(20)

AS

insert into telefono values (@Clave\_E,@No\_tel,@tipo)

GO

--Procedimiento ingresar registros a la tabla tiene\_asesor

CREATE PROCEDURE ingresarTieneA

@No\_control char(8),

@Clave\_A char(5)

AS

insert into tiene\_asesor values (@No\_control, @Clave\_A)

GO

--Procedimiento ingresar registros a la tabla asesor

CREATE PROCEDURE ingresarAsesor

@Clave\_A char(5),

@Nombre\_A char(40)

AS

insert into asesor values (@Clave\_A,@Nombre\_A)

GO

--Procedimiento ingresar registros a la tabla externo

CREATE PROCEDURE ingresarExterno

@Clave\_A char(5),

@cargo varchar(30),

@Clave\_E char(5)

AS

insert into externo values (@Clave\_A,@cargo,@Clave\_E)

GO

--Procedimiento ingresar registros a la tabla interno

CREATE PROCEDURE ingresarInterno

@Clave\_A char(5),

@Horas\_asig char(3),

@Grado varchar(20)

AS

insert into interno values (@Clave\_A,@Horas\_asig,@Grado)

GO

--PROCEDIMIENTOS PARA ELIMINAR--

--Procedimiento eliminar registros a la tabla alumno

CREATE PROCEDURE EliminarRegistroAlumno

@No\_control char(8)

AS

DELETE FROM alumno

WHERE No\_control = @No\_control

GO

--Procedimiento eliminar registros a la tabla empresa

CREATE PROCEDURE EliminarRegistroEmpresa

@Clave\_E char(5)

AS

DELETE FROM empresa

WHERE Clave\_E = @Clave\_E

GO

--Procedimiento eliminar registros a la tabla proyecto

CREATE PROCEDURE EliminarRegistroProyecto

@No\_proy char(5)

AS

DELETE FROM proyecto

WHERE No\_proy = @No\_proy

GO

--Procedimiento eliminar registros a la tabla telefono

CREATE PROCEDURE EliminarRegistroTelefono

@No\_tel char(10)

AS

DELETE FROM telefono

WHERE No\_tel = @No\_tel

GO

--Procedimiento eliminar registros a la tabla realiza

CREATE PROCEDURE EliminarRegistroRealiza

@No\_control char(8),

@No\_proy char(5)

AS

DELETE FROM realiza

WHERE No\_control = @No\_control AND No\_proy = @No\_proy

GO

--Procedimiento eliminar registros a la tabla asesor

CREATE PROCEDURE EliminarRegistroAsesor

@Clave\_A char(5)

AS

DELETE FROM asesor

WHERE Clave\_A = @Clave\_A

GO

--Procedimiento eliminar registros a la tabla asesor interno

CREATE PROCEDURE EliminarRegistroAInterno

@Clave\_A char(5)

AS

DELETE FROM interno

WHERE Clave\_A = @Clave\_A

GO

--Procedimiento eliminar registros a la tabla asesor externo

CREATE PROCEDURE EliminarRegistroAExterno

@Clave\_A char(5)

AS

DELETE FROM externo

WHERE Clave\_A = @Clave\_A

GO

--Procedimiento eliminar registros a la tabla tiene Asesor

CREATE PROCEDURE EliminarRegistroTAsesor

@No\_control char(8),

@Clave\_A char (5)

AS

DELETE FROM tiene\_asesor

WHERE No\_control = @No\_control AND Clave\_A = @Clave\_A

GO

------------------------**Creacion de trigger**-------------------------

create trigger CheckAlumnoInsert on alumno for insert

as

if (select Creditos from inserted) < 0

begin

print 'Los créditos del alumno no pueden ser negativos.'

rollback

end

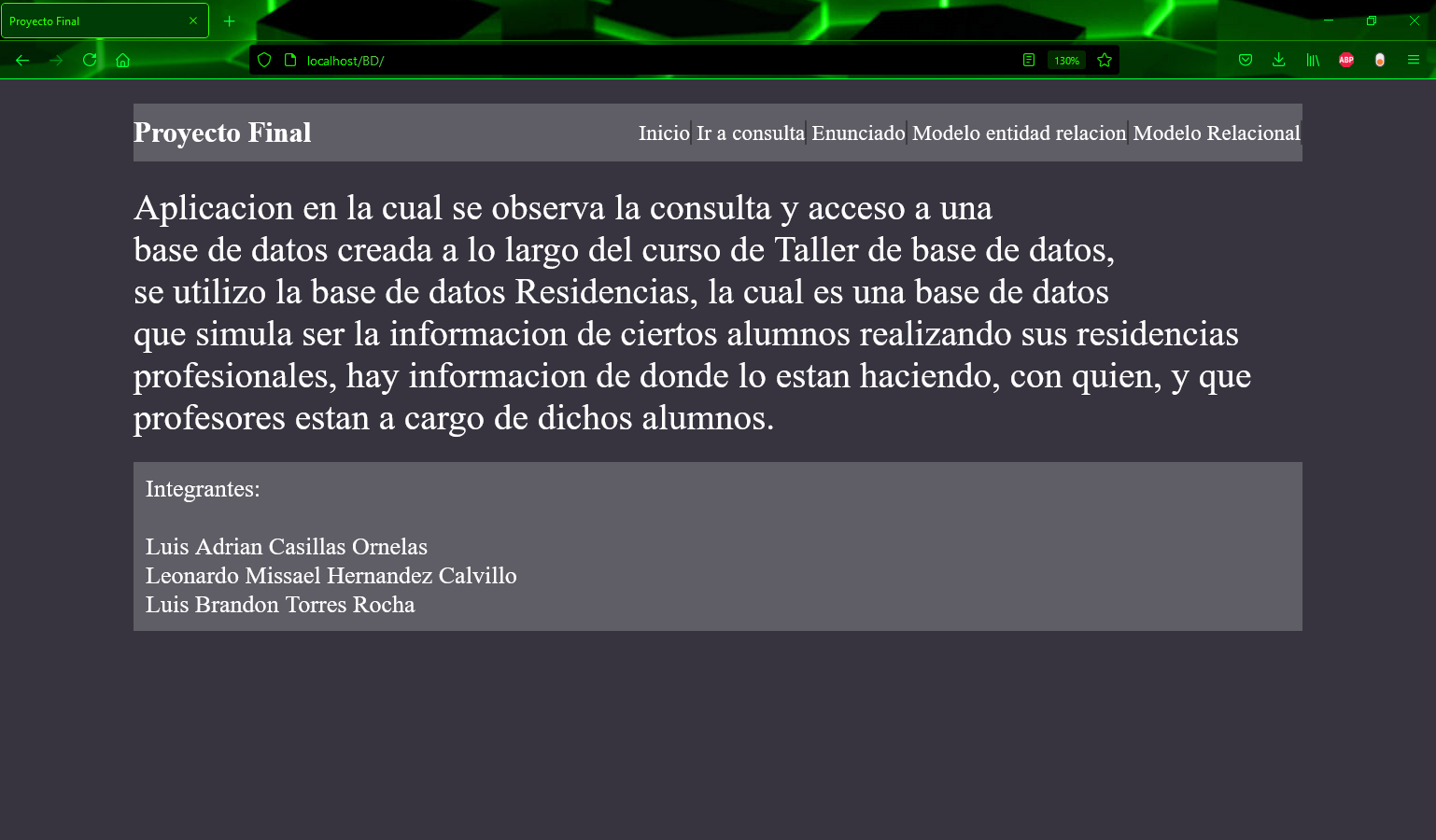
go

# **Conexión a la BD con un lenguaje de programación:**

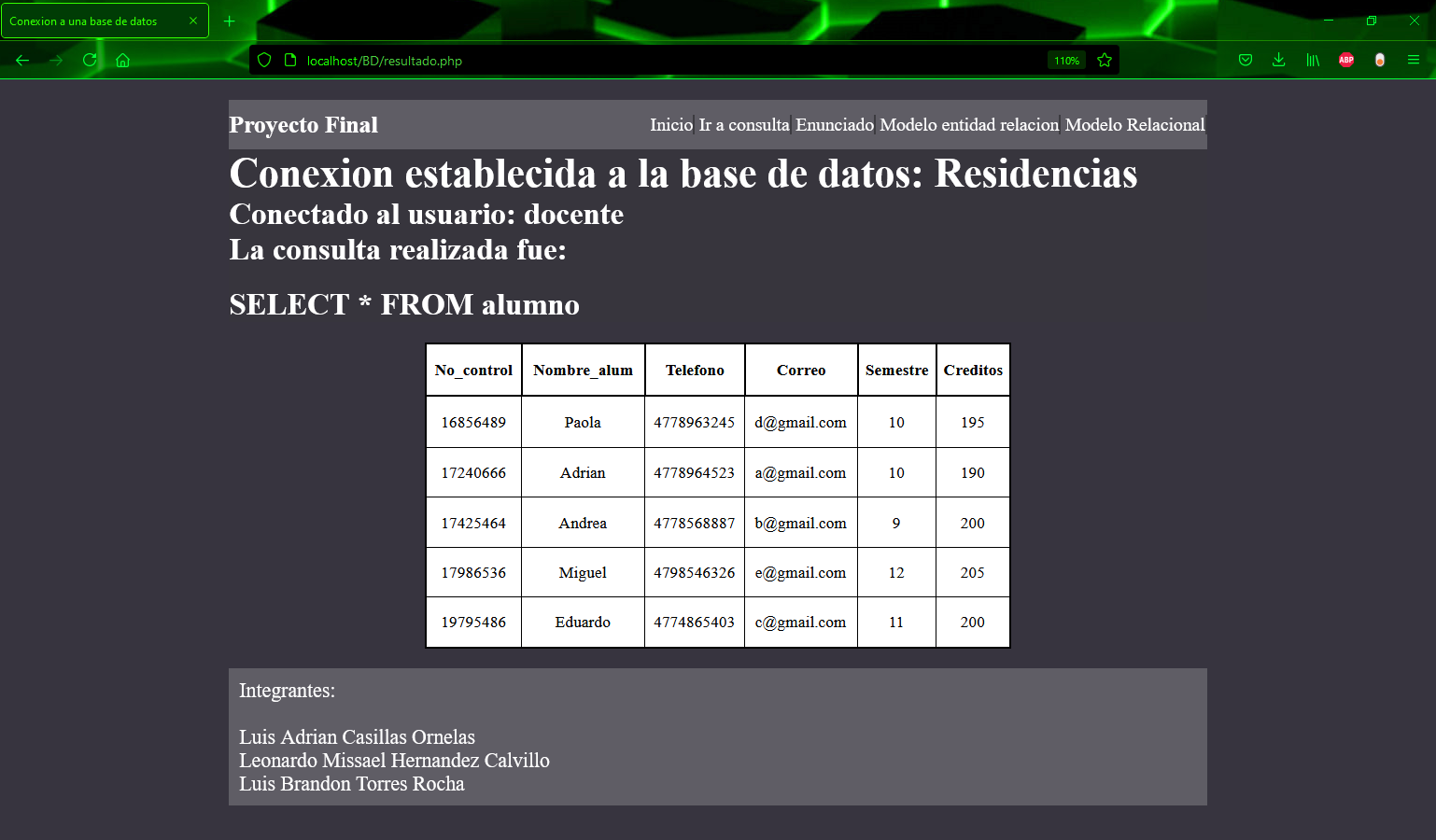
Se realizó la conexión a la base de datos en PHP en el cual se tiene una pagina de inicio donde se explica de que trata el proyecto, así como contar con pestañas para ver una consulta previamente establecida en el lenguaje de programación, otra pestaña para ver el enunciado. El modelo entidad relación y una última para ver el modelo relacional.

En la pestaña de la consulta se puede observar a qué base de datos está conectado, con cual usuario se ingresó (establecido en el lenguaje), cuál fue la consulta que se ralizó y la tabla resultante.

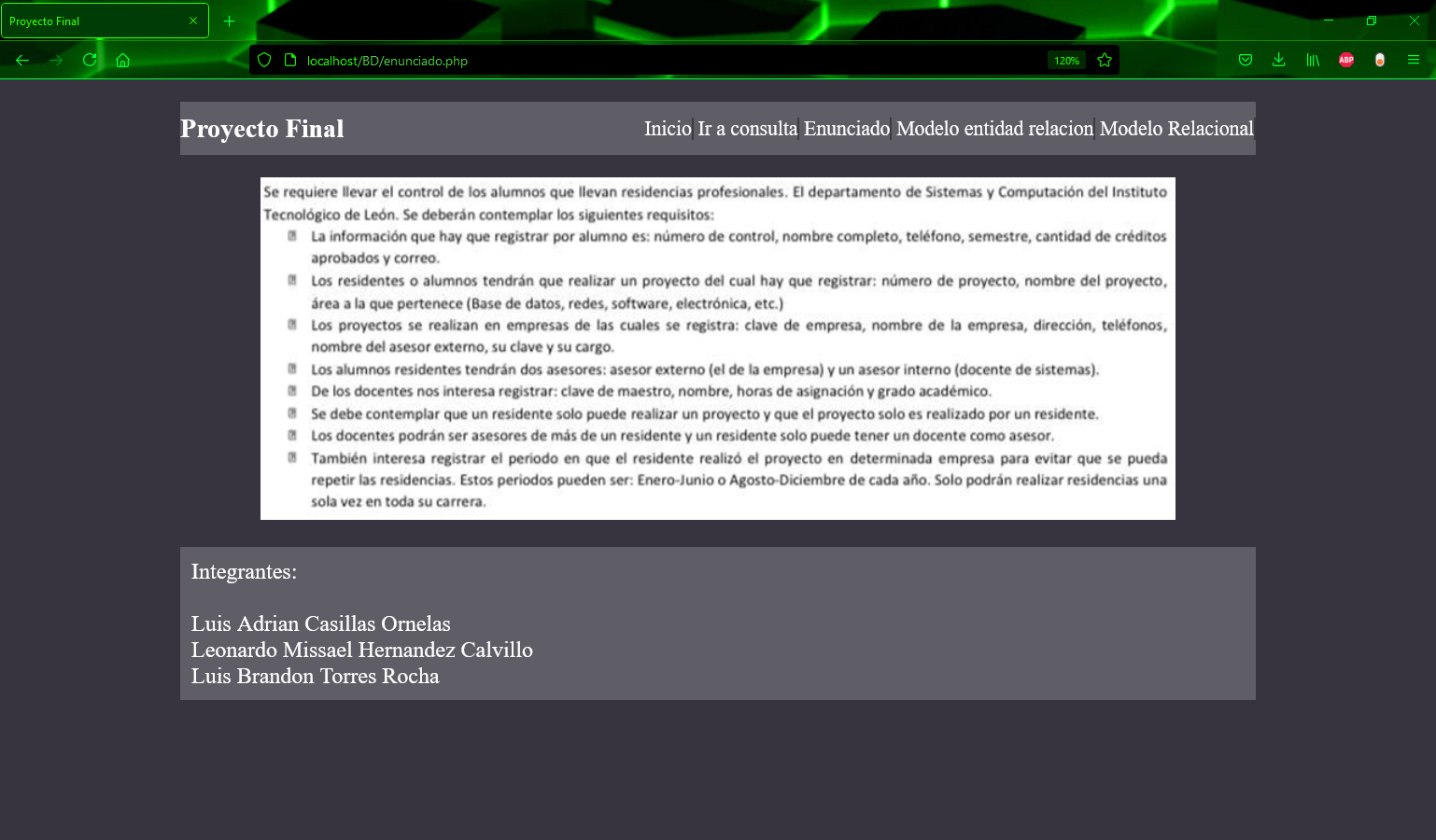
## **Pagina de inicio:**



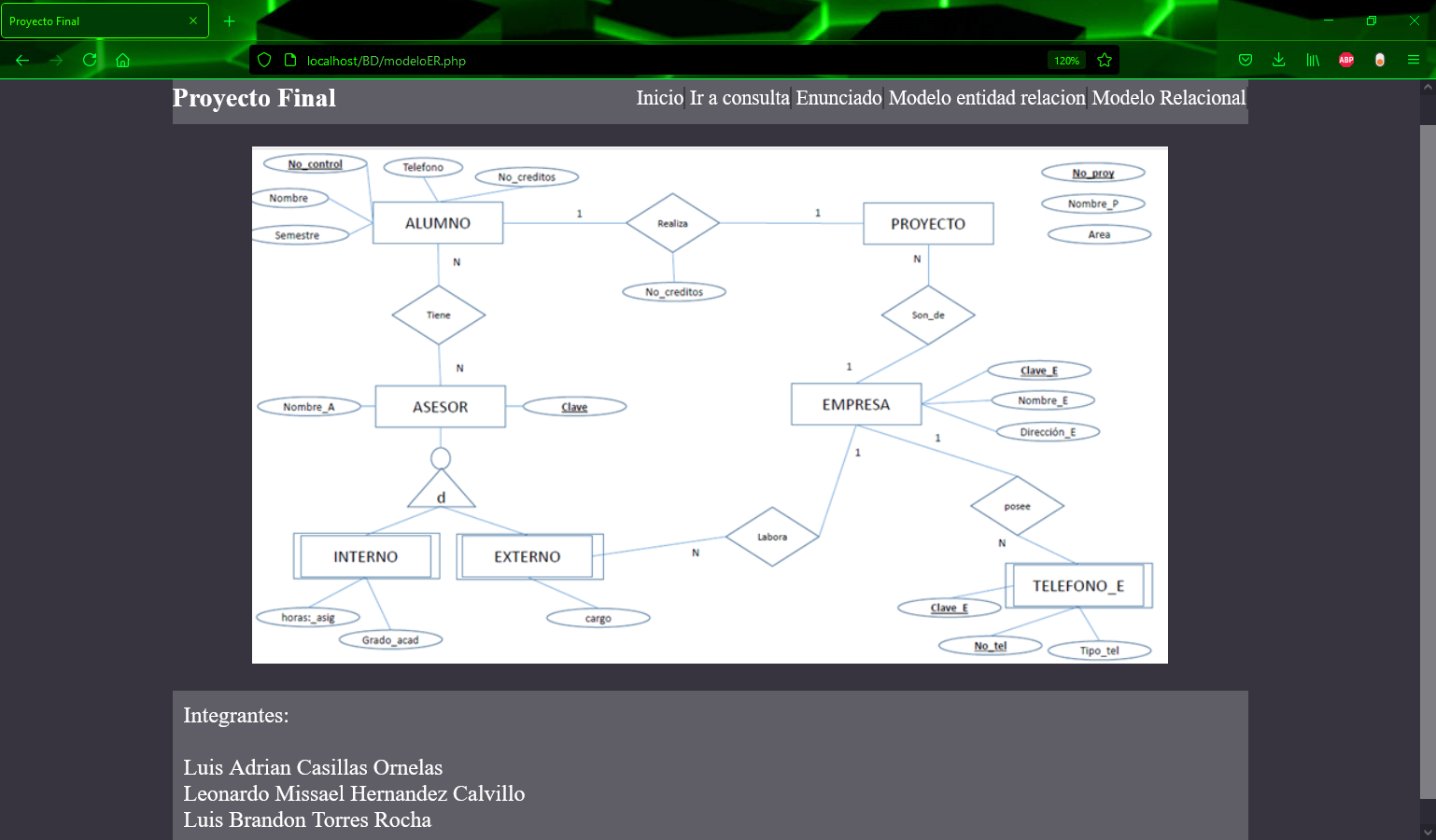
## **Consulta:**



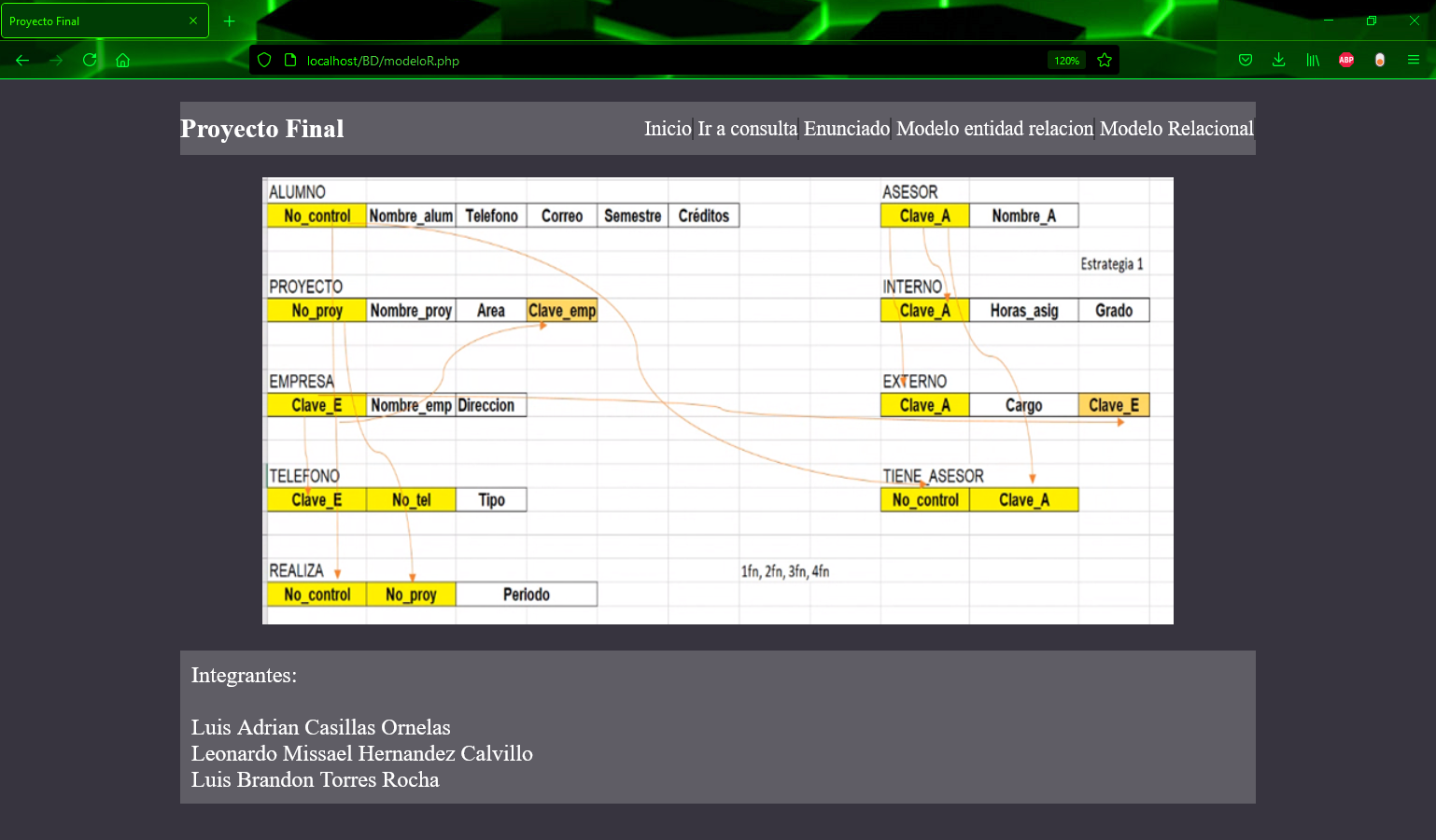
## **Enunciado:**



## **Modelo entidad relación**



## **Modelo relacional:**



## **Código de la conexión a la BD:**

<?php

$nombreServer = 'DESKTOP-L4BS0N0\SQLEXPRESS';

$infoDeConexion = array("Database" => "Residencias", "UID"=>"docente","PWD"=>"1234");

$conexion = sqlsrv\_connect($nombreServer, $infoDeConexion);

$sql = "";

  function validacion($conexion,$infoDeConexion){

*if* ($conexion) *return* true;

*else* {

*die*('No se pudo conectar a la base de datos, revise las credenciales');

*return* false;

    }

  }

  function consulta($conexion){

    global $sql;

    $sql= "SELECT \* FROM alumno";

    $stmt = sqlsrv\_query($conexion, $sql);

*return* $stmt;

  }

?>