UNIVERSIDAD DE CÓRDOBA
Departamento de Informática y Análisis Numérico
Ingeniería del Software, Conocimiento y Bases de Datos

Bases de Datos Avanzadas

Curso 21/22 - Luis Aneri Delgado - Universidad de Córdoba

Fase 1: Diseños de Bases de Datos

OBJETIVO

El objetivo de esta práctica es afianzar los conocimientos sobre el proceso de diseño de bases de datos impartidos en la parte teórica de la asignatura. Para ello, se deberá proponer un problema con información suficiente para la evaluación por parte del profesor del conocimiento de los alumnos del proceso y técnicas del diseño de bases de datos.

TRABAJO A REALIZAR

1. *Descripción del problema, requisitos y reglas de negocio.*
2. *Construcción del modelo conceptual y validación (análisis)*
3. *Construcción del modelo relacional y validación.*
4. *Generar el script de creación de la base de datos, consistente en la creación de las tablas y la inserción de una extensión representativa de las tablas correspondientes.*
5. *Realizar una carga de información lo suficientemente completa para que le sirva como prueba y validación para el diseño realizado, lo que llevará a cabo mediante consultas a las correspondientes bases de datos y generando informes de salida que le permita comprobar la coherencia e integridad de la información.*

TRABAJO REALIZADO

1. Descripción del problema, requisitos y reglas de negocio

Nombre: Rutas turísticas y guías.

Dominio: El problema administra una aplicación que permite definir **rutas turísticas** o de **senderismo** por usuarios de tipo **guía** para que usuarios de tipo **turista** puedan usarlas por un costo de alquiler. Las rutas aparecerán en un mapa con los **puntos de interés**. Los usuarios podrán añadir una **valoración** a la ruta. Las rutas compradas se almacenarán en la **biblioteca** del usuario.

Requisitos:

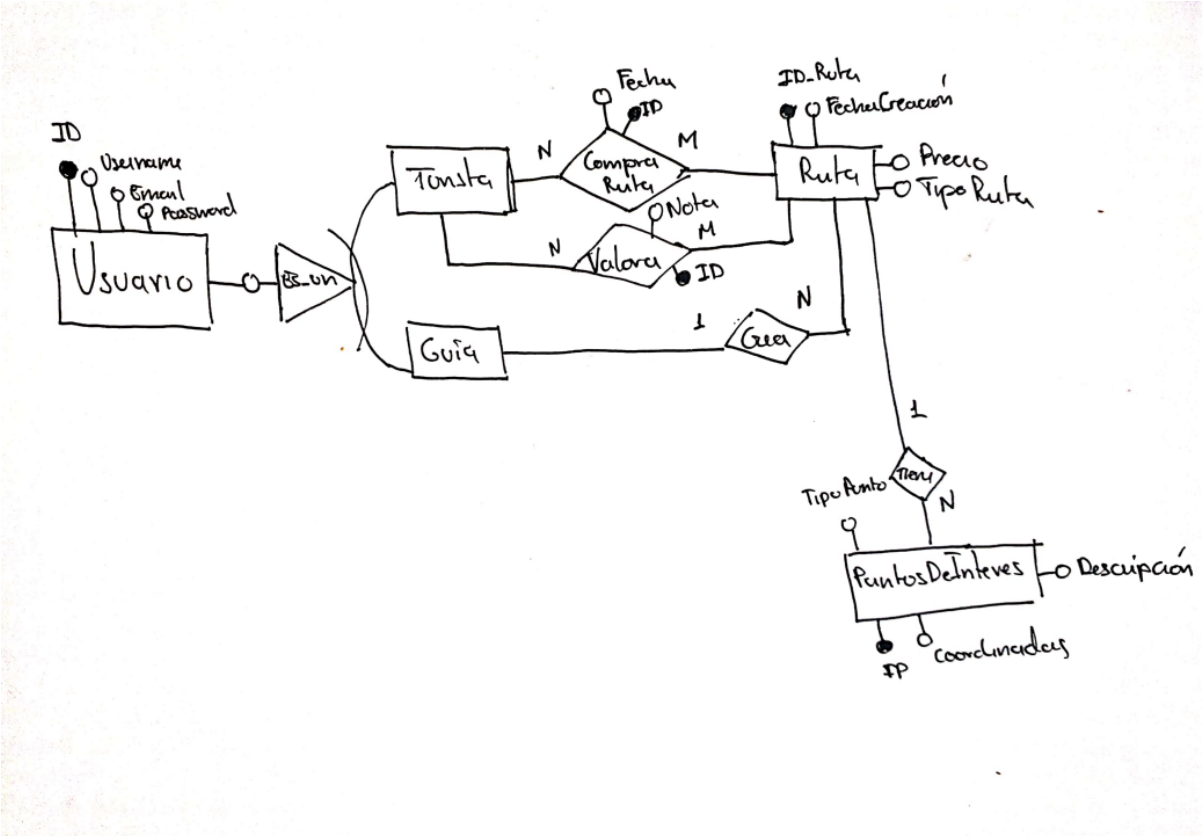
- El *turista* tiene un nombre de un **ID**,usuario (**username**), un **email** y un **password.**
- Un *guía* posee los mismos campos.
- Una *ruta* posee un un **ID**, **nombre**, una **fecha de creación**, un **precio**, el **guía** y el **tipo de ruta(** turística, senderismo).
- Un *punto de interés* posee un **ID**, una **descripción**, una **foto**, unas **coordenadas,** la **ruta**, y el **tipo de punto**(*inicio, medio, final).*
- Una *valoración* posee un **ID**, una **nota**, el **usuario**, la **ruta**.
- Una *compra* posee el un **ID**,**usuario**, la **ruta** y la **fecha de adquisición**

Reglas de Negocio

- Un usuario no puede hacer una misma crítica sobre una ruta
- Un usuario debe haber comprado la ruta para valorarla
- Pasados tres meses la compra se borra

2. *Construcción del modelo conceptual y validación (análisis)*

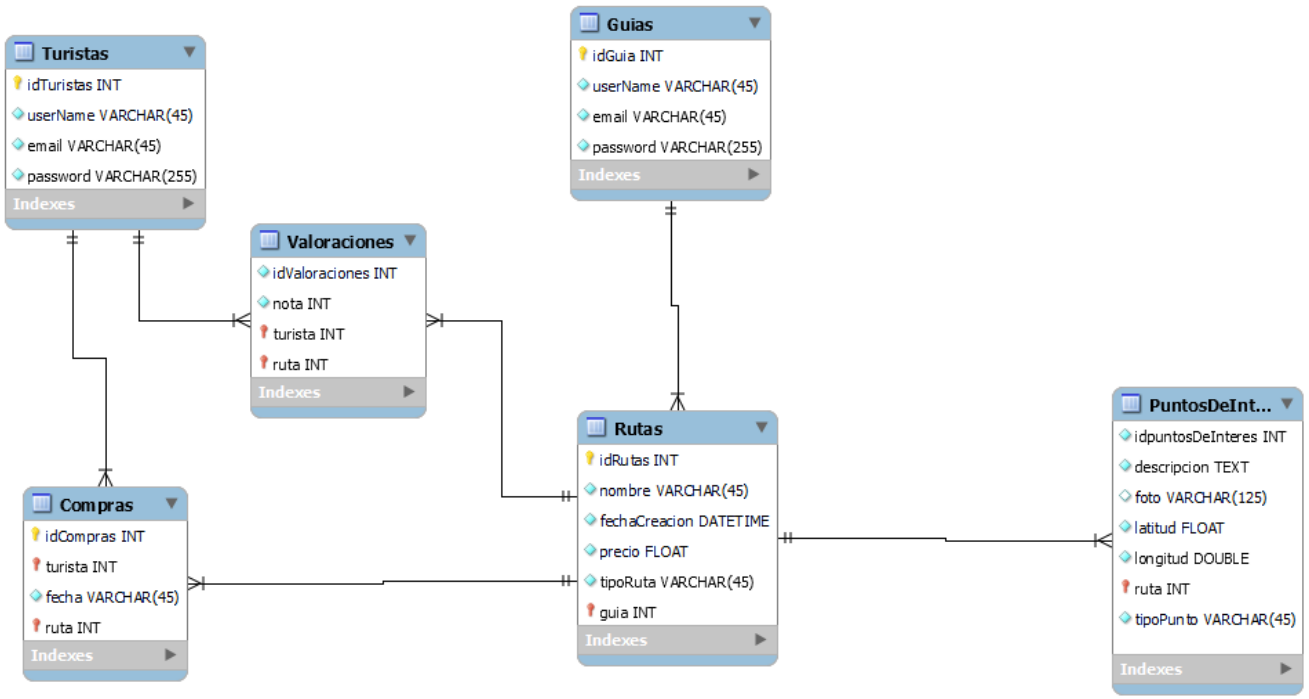
Modelo Conceptual



3. *Construcción del modelo relacional y validación.*

Turistas (idTurista, userName, Email, Password)
Guias (idGuia, userName, email, password)
Compras (idCompras, fechaCompra, **turista**, **ruta**)
Valoraciones (idValoraciones, nota, **turista**, **ruta**)
Rutas (idRuta, nombre, fechaCreacion, precio, tipoRuta, **guia**)
PuntosDeInteres (idPuntosInteres, descripcion, foto, latitud, longitud, tipoPunto, **ruta**)

Modelo Entidad Relación




```
CREATE TABLE Turistas(
  idTuristas INT NOT NULL PRIMARY KEY,
  userName VARCHAR(45) NOT NULL,
  email VARCHAR(45) NOT NULL,
  password VARCHAR(255) NOT NULL);
  constraint fk_evento foreign key (evento) references
eventos(idEvento),
  constraint fk_votante foreign key (votante) references
votantes(dni));


CREATE TABLE Guias (
  idGuia INT NOT NULL PRIMARY KEY,
  userName VARCHAR(45) NOT NULL,
  email VARCHAR(45) NOT NULL,
  password VARCHAR(255) NOT NULL);

CREATE TABLE Rutas (
  idRutas INT NOT NULL PRIMARY KEY,
  nombre VARCHAR(45) NOT NULL,
  fechaCreacion DATE NOT NULL,
  precio FLOAT NOT NULL,
  tipoRuta VARCHAR(45) NOT NULL,
  guia INT NOT NULL);

CREATE TABLE Compras (
  idCompras INT NOT NULL PRIMARY KEY,
  turista INT NOT NULL,
  fecha VARCHAR(45) NOT NULL,
  rut INT NOT NULL);


CREATE TABLE  PuntosDeInteres (
  idpuntosDeInteres INT NOT NULL PRIMARY KEY,
descripcion  VARCHAR(255) NOT NULL,
  foto VARCHAR(125) NULL,
  latitud FLOAT NOT NULL,
  longitud FLOAT NOT NULL,
  ruta INT NOT NULL,
  tipoPunto VARCHAR(45) NOT NULL);


CREATE TABLE Valoraciones (
  idValoraciones INT NOT NULL PRIMARY KEY,
  nota INT NOT NULL,
  turista INT NOT NULL,
  ruta INT NOT NULL);
```

5. Realizar una carga de información lo suficientemente completa para que le sirva como prueba y validación para el diseño realizado, lo que llevará a cabo mediante consultas a las correspondientes bases de datos y generando informes de salida que le permita comprobar la coherencia e integridad de la información.

	IDRUTAS	NOMBRE	FECHACREACION	PRECIO	TIPORUTA	GUIA
1	1	Gujo	03-NOV-21	89.87	4	2
2	2	Dasoguz	13-JAN-21	83.79	1	10
3	3	Khirbat Tin Nur	20-FEB-21	68.41	4	3
4	4	Sovetskaya	03-OCT-21	5.38	4	5
5	5	Kuala Lumpur	14-NOV-21	76.77	1	9
6	6	Qu¿ng Ngãi	30-DEC-21	8.78	3	9
7	7	Piedade	13-MAR-21	41.71	3	3
8	8	Cilated	03-SEP-21	38.16	4	1
9	9	Gamping Lor	07-JUL-21	66.18	4	14
10	10	Lobão	03-JUL-21	43.9	3	14
11	11	Port Saint Lucie	04-APR-21	29.61	2	4
12	12	Mandeman Daya	16-NOV-21	83.35	5	18
13	13	Ogawa	16-MAY-21	45.64	5	5
14	14	Cimanggu	05-JUN-21	5.41	2	17
15	15	Ambanja	02-AUG-21	42.3	1	18
16	16	Malhão	23-JAN-21	92.66	4	13
17	17	Pershotravneve	15-JAN-21	61.1	5	7
18	18	Bokhan	15-SEP-21	8.39	5	12
19	19	Nkpor	14-DEC-21	85.95	1	2
20	20	Huacao	22-MAR-21	72.27	2	15

tablas de turistas

	IDTURISTAS	USERNAME	EMAIL	PASSWORD
1	1	sgaraghan0	mcouvert0@opensource.org	Q0lo08YPu
2	2	vcrowthe1	cdarby1@topsy.com	TZ6tzW8Ek
3	3	adavenport2	wgosart2@usnews.com	mxFY1PRGb1
4	4	nberrey3	emuneely3@illinois.edu	FFQM2NAV
5	5	sarkcol14	sbanck4@trellian.com	xE0olI
6	6	bfabb5	spipping5@businessinsider.com	eRUJrI5L9bD
7	7	pmcpheat6	fdurn6@nature.com	nYbSybu
8	8	bpumphrey7	dleyburn7@feedburner.com	EGgkJamJfAk
9	9	gswine8	eburburough8@dailymail.co.uk	kDlbubIkq3
10	10	bpache9	mgeydon9@walmart.com	uNk5hORNSL
11	11	kzanittia	rmansera@devhub.com	QwMVx411
12	12	fsparwellb	mlerouxb@china.com.cn	Ma76zNL2J
13	13	lrollingc	cjordinc@delicious.com	KoBKgsSJZI
14	14	jcastaignetd	jhutchcraftd@guardian.co.uk	Fuyk4L81eVzV
15	15	mchestneye	vwollene@360.cn	q33qraEMk
16	16	omitchelsonf	kbecomf@prlog.org	UEzm7ZA
17	17	psollomg	bsciacovellig@weibo.com	cleWB9XdVPC4
18	18	tciccottoh	pvarnalsh@t.co	y7IEfdBfURnA
19	19	jlanti	pbeamesi@xing.com	kyOJhNe
20	20	btrewhelaj	rjerischj@histats.com	dq26NUobmv

tabla de rutas

1. Conseguir toda la información de los turistas

```
SELECT * FROM turistas;
```

ery Result ×

SQL | All Rows Fetched: 20 in 0.515 seconds

	IDTUR...	USERNAME	EMAIL	PASSWORD
1	1	sgaraghan0	mcouvert0@opensource.org	Q0lo08YPu
2	2	vcrowthe1	cdarby1@topsy.com	TZ6tzW8Ek
3	3	adavenport2	wgosart2@usnews.com	mxFY1PRGb1
4	4	nberrey3	emuneely3@illinois.edu	FFQM2NAV
5	5	sarkcoll4	sbanck4@trellian.com	xE0olI
6	6	bfabb5	spipping5@businessinsider.com	eRUJrI5L9bD
7	7	pmcpheat6	fdurn6@nature.com	nYbSybu
8	8	bpumphrey7	dleyburn7@feedburner.com	EGgkJamJfAk
9	9	gswine8	eburburough8@dailymail.co.uk	kDlbubIkq3
10	10	bpache9	mgeydon9@walmart.com	uNk5hORNSL
11	11	kzanittia	rmansera@devhub.com	QwMVx411
12	12	fsparwellb	mlerouxb@china.com.cn	Ma76zNL2J
13	13	lrollingc	cjordinc@delicious.com	KoBKgsSJZI
14	14	jcastaignetd	jhutchcraftd@guardian.co.uk	Fuyk4L8leVzV
15	15	mchestneye	vwollene@360.cn	q33qraEMk

2. Conseguir la información de un guia y las rutas que tienen asignadas.

```
SELECT t.idguia, t.username, t.email, t.password, r.nombre, r.precio,r.tiporuta FROM guias t, rutas r WHERE t.idguia = r.guia and t.idguia LIKE 2;
```

y Result ×

SQL | All Rows Fetched: 2 in 0.046 seconds

IDGUIA	USERNAME	EMAIL	PASSWORD	NOMBRE	PRECIO	TIPORUTA
2	zwalpole1	abrenstuhl1@icq.com	8ACtJD5	Nkpor	85.95	1
2	zwalpole1	abrenstuhl1@icq.com	8ACtJD5	Gujo	89.87	4

Fase 2: Uso de Disparadores

OBJETIVO

El objetivo de esta práctica es el uso de los disparadores (“triggers”) en el desarrollo de procedimientos de control de la integridad de la base de datos, alertas y construcción de bases de datos activas. Se deberá incluir en el esquema de la base de datos la definición de una serie de disparadores y desarrollar los procedimientos para que: a) se activen y b) no se activen.

TRABAJO A REALIZAR

Diseñar y programar una serie de disparadores que abarquen aspectos relacionados con las siguientes funcionalidades:

1. *Un disparador de auditoría que informe de las modificaciones de uno o varios atributos de una tabla.*
2. *Un disparador de seguridad que impida realizar actualizaciones de la base de datos en base a algún criterio relacionado con la fecha, usuario, etc.*
3. *Un disparador que sustituya a una restricción de dominio existente en la base de datos.*
4. *Un disparador que sustituya a alguna restricción de integridad de referencia existente en la base de datos.*
5. *Un disparador sobre alguna tabla cuya condición se satisfaga en función de la extensión de alguna otra tabla.*

1.Un disparador de auditoría que informe de las modificaciones de uno o varios atributos de una tabla.

```
CREATE TABLE AuditoriaCompra
(
    idCompras INT NOT NULL,
    turista INT NOT NULL,
    fecha VARCHAR2(35),
    ruta INT NOT NULL,
    fecha_modificacion DATE,
    tipo_modificacion VARCHAR(15)
);
```

Para este apartado, he creado una tabla llamada AuditoriaCompra, que controlará todas las modificaciones que se hagan sobre la tabla Compras

Para ello, he creado un trigger, que controla inserciones de datos, eliminaciones de datos, y por último, actualizaciones de datos.

Es el siguiente:

```
create or replace trigger audit_trigger_compras
AFTER insert or update or delete on compras
for each row
    begin
        IF UPDATING THEN
            INSERT INTO AuditoriaCompra (idCompras,turista,fecha, ruta, fecha_modificacion,tipo_modificacion)
            values(:new.idCompras, :new.turista, :new.fecha ,:new.ruta, sysdate,'Actualizacion');
        ELSIF DELETING THEN
            INSERT INTO AuditoriaCompra (idCompras,turista,fecha, ruta, fecha_modificacion,tipo_modificacion)
            values(:old.idCompras, :old.turista, :old.fecha ,:old.ruta, sysdate,'Eliminacion');
        ELSE
            INSERT INTO AuditoriaCompra (idCompras,turista,fecha, ruta, fecha_modificacion,tipo_modificacion)
            values(:new.idCompras, :new.turista, :new.fecha ,:new.ruta, sysdate,'Insercion');
        END IF;
    end;

alter trigger audit_trigger_compras disable;
```

Y realizando las siguientes modificaciones:

```
INSERT INTO Compras (idCompras, turista, fecha, ruta) values (50, 16, '12/9/2019', 17);
DELETE FROM compras WHERE idCompras=50;
```

resultado

	IDCOMPRAS	TURISTA	FECHA	RUTA	FECHA_MODIFICACION	TIPO_MODIFICACION
1	50	16	12/9/2019	17	15-JAN-22	Insercion
2	50	16	12/9/2019	17	15-JAN-22	Eliminacion

2. Un disparador de seguridad que impida realizar actualizaciones de la base de datos en base a algún criterio relacionado con la fecha, usuario, etc.

El disparador creado impide realizar actualizaciones en la tabla producto si precio<0, ya que un precio no puede ser negativo.

Es el siguiente:

```
create or replace trigger trigger_seguridad_precio
  BEFORE insert or update on RUTA
  for each row
   begin
   IF :new.PRECIO<0 THEN
    RAISE_APPLICATION_ERROR(-20001,'******El precio de una ruta no puede ser negativo.******');
    END IF;
   end;

INSERT INTO PRODUCTO(ID_PRODUCTO,STOCK,PRECIO,DEPARTAMENTO_NUM) VALUES(21905,10,-6,87);
```

Y como se puede observar en la siguiente imagen, si intentamos insertar un producto con un precio<0, nos dará el error generado por nuestro disparador:

```
Error starting at line : 42 in command -
insert into Rutas (idRutas, nombre, fechaCreacion, precio, tipoRuta, guia) values (101, 'Guj?', to_date('3/11/2021', 'DD-MM-YYYY'), -10, 4, 2)
Error report -
ORA-20001: El precio de una ruta no puede ser negativo.
ORA-06512: at "I82ANDEL.TRIGGER_SEGURIDAD_PRECIO", line 3
ORA-04088: error during execution of trigger 'I82ANDEL.TRIGGER_SEGURIDAD_PRECIO'
```

3. Un disparador que sustituya a una restricción de dominio existente en la base de datos.

El disparador creado obliga a que el número de departamento esté entre 0 y 100, ya que 100 son todos los departamentos que hay.

Es el siguiente:

```
create or replace trigger trigger_num_nota
  BEFORE insert or update on valoraciones
  for each row
   begin
   IF :new.nota<1 OR :new.nota>10 THEN
    RAISE_APPLICATION_ERROR(-20002,'La nota debe de ser entre 1 y 10');
    END IF;
   end;
```

Lo probamos

```
Error starting at line : 54 in command -
   insert into Valoraciones (idValoraciones, nota, turista, ruta) values (30, 11, 20, 14)
Error report -
ORA-20002: La nota debe de ser entre 1 y 10
ORA-06512: at "I82ANDEL.TRIGGER_NUM_NOTA", line 3
ORA-04088: error during execution of trigger 'I82ANDEL.TRIGGER_NUM_NOTA'
```

4. Un disparador que sustituya a alguna restricción de integridad de referencia existente en la base de datos.

El disparador creado controla que el campo email es del tipo email

Es el siguiente:

```
CREATE OR REPLACE TRIGGER email_validate_insert
    BEFORE INSERT OR UPDATE
    on TURISTAS
    FOR EACH ROW
BEGIN
    IF :new.email NOT LIKE '%_@%_.__%' THEN
        RAISE_APPLICATION_ERROR(-20003,'El email debe ser valido');
    END IF;
END;
```

lo comprobamos con un email no valido

```
Error starting at line : 66 in command -
insert into Turistas (idTuristas, userName, email, password) values (20, 'btrewhelaj', 'rjerischj', 'dq26NUobmv')
Error report -
ORA-20003: El email debe ser valido
ORA-06512: at "I82ANDEL.EMAIL_VALIDATE_INSERT", line 3
ORA-04088: error during execution of trigger 'I82ANDEL.EMAIL_VALIDATE_INSERT'
```

5. Un disparador sobre alguna tabla cuya condición se satisfaga en función de la extensión de alguna otra tabla.

Fase 3: BBDD OBJETO-RELACIONAL

OBJETIVO

El objetivo de esta práctica es el uso de objetos con Oracle mediante la definición y manipulación de tablas de objetos y de tablas anidadas.

TRABAJO A REALIZAR

Se debe modificar el esquema de la base de datos con el propósito de utilizar objetos, definidos por el alumno, y la definición y manipulación de tablas basadas en los objetos previamente definidos. Los objetos y tablas que defina el alumno serán libremente determinados por el alumno en función de su esquema, estando sujeta a los siguientes requisitos:

1. *Al menos deberá definirse un objeto el cual será utilizado para la definición de la estructura de una tabla.*
2. *Al menos deberá existir en el nuevo esquema una tabla en la que en su definición exista un atributo que a su vez sea una tabla.*
3. *Al menos deberá existir una tabla en la que en su definición exista un atributo que sea un array.*

Creamos el objeto informacion de visita

```
CREATE OR REPLACE TYPE INFORMACION_VISITA as object
(
    precio FLOAT,
    fecha_visita DATE,
    nombre VARCHAR(50)
)
```

Creamos la nested table

```
CREATE OR REPLACE TYPE PUNTOS_INTERES as table of VARCHAR2(150);
```

Creamos la tabla

```
CREATE OR REPLACE TYPE VALORACIONES as VARRAY(50)of INT;
```

```
CREATE TABLE VISITAS
(
    id INT PRIMARY KEY,
    guia INT,
    informacion INFORMACION_VISITA,
    puntosInteres PUNTOS_INTERES,
    valoraciones VALORACIONES_VISITA
)
    NESTED TABLE puntosInteres STORE AS NESTED_PUNTOS_INTERES;
```

La llenamos

```
DECLARE
    info INFORMACION_VISITA;
    vals VALORACIONES_VISITA;
BEGIN
    info := INFORMACION_VISITA(25.99, to_date('3/11/2021', 'DD-MM-YYYY'),'Cordoba misteriosa');
    vals := VALORACIONES_VISITA(8,9,8,7,9,7,7);
    INSERT INTO VISITAS VALUES(2,5,info,PUNTOS_INTERES('CASA ENCANTADA','MEZQUITA','FACULTAD DE DERECHO', 'CORREDE
end;
```

SELECT * FROM VISITAS;

1	2	5	[I82ANDEL.INFORMACION_VISITA]	I82ANDEL.PUNTOS_INTERES('CASA ENCANTADA', 'MEZQUITA', 'FACULTAD DE DERECHO'

4. *Al menos deberá definir un método para un objeto y utilizarlo en alguna operación de manipulación de la base de datos. Por ejemplo, un método que obtenga el consumo medio de cada vehículo, la media de productos vendidos, el total de facturas emitidas, etc.*

```
CREATE OR REPLACE TYPE valoracion_metod AS OBJECT
(
id_ruta INT,
email VARCHAR2(55),
puntuacion_guia FLOAT,
puntuacion_visita FLOAT,
MEMBER FUNCTION media RETURN FLOAT
);


CREATE OR REPLACE TYPE BODY valoracion_metod AS
    MEMBER FUNCTION media RETURN FLOAT IS
    BEGIN
        RETURN (puntuacion_vendedor+puntuacion_producto)/2;
    END media;
END;


CREATE TABLE VALORACION of valoracion_metod;

INSERT INTO VALORACION VALUES(valoracion_metod(12,'paco@uco.es',8,6));
INSERT INTO VALORACION VALUES(valoracion_metod(8,'pepe#uco.es',9.5,10));
INSERT INTO VALORACION VALUES(valoracion_metod(3,'jose@uco.es',4,8.5));

select * from VALORACION;
select v.media() from VALORACION v;
```

select * from VALORACION;

	ID_RUTA	EMAIL	PUNTUACION_GUIA	PUNTUACION_VISITA
1	12	paco@uco.es	8	6
2	8	pepe#uco.es	9.5	10
3	3	jose@uco.es	4	8.5

select v.media() from VALORACION ;

	V.MEDIA()
1	7
2	9.75
3	6.25

Fase 4: USO DEL MIDDLEWARE DE BASES DE DATOS

OBJETIVO

El objetivo de esta práctica es que los alumnos se familiaricen con el acceso a bases de datos desde entornos externos a las mismas, es decir, hagan uso del middleware para la conexión y manipulación de las bases de datos. No se pretende en esta práctica que los alumnos aprendan ningún lenguaje de programación en particular, sino que hagan uso de los mínimos recursos de los lenguajes de programación para acceder y manipular la base de datos construida.

TRABAJO A REALIZAR

El alumno deberá realizar esta práctica para la de datos creada realizando procedimientos para manipular esta información (acceso, inserción y modificación). El alumno seleccionará sobre qué tablas de la base de datos implementa la funcionalidad requerida.

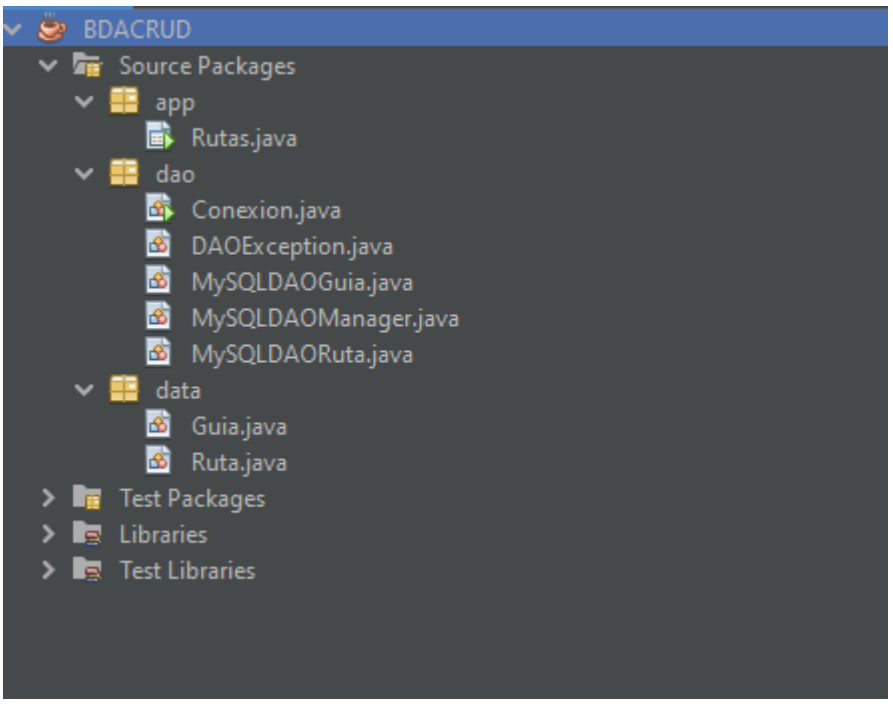
Esta actividad la podrá realizar de alguna de las siguientes formas:

***HTML/pHp:** Construirá una página HTML desde la que se pueda llevar a cabo la conexión a la base de datos mediante una cadena de conexión introducida por el usuario, y construirá una o varias páginas HTML desde las que se realicen alguna operación de inserción, consulta y modificación de la base de datos.*

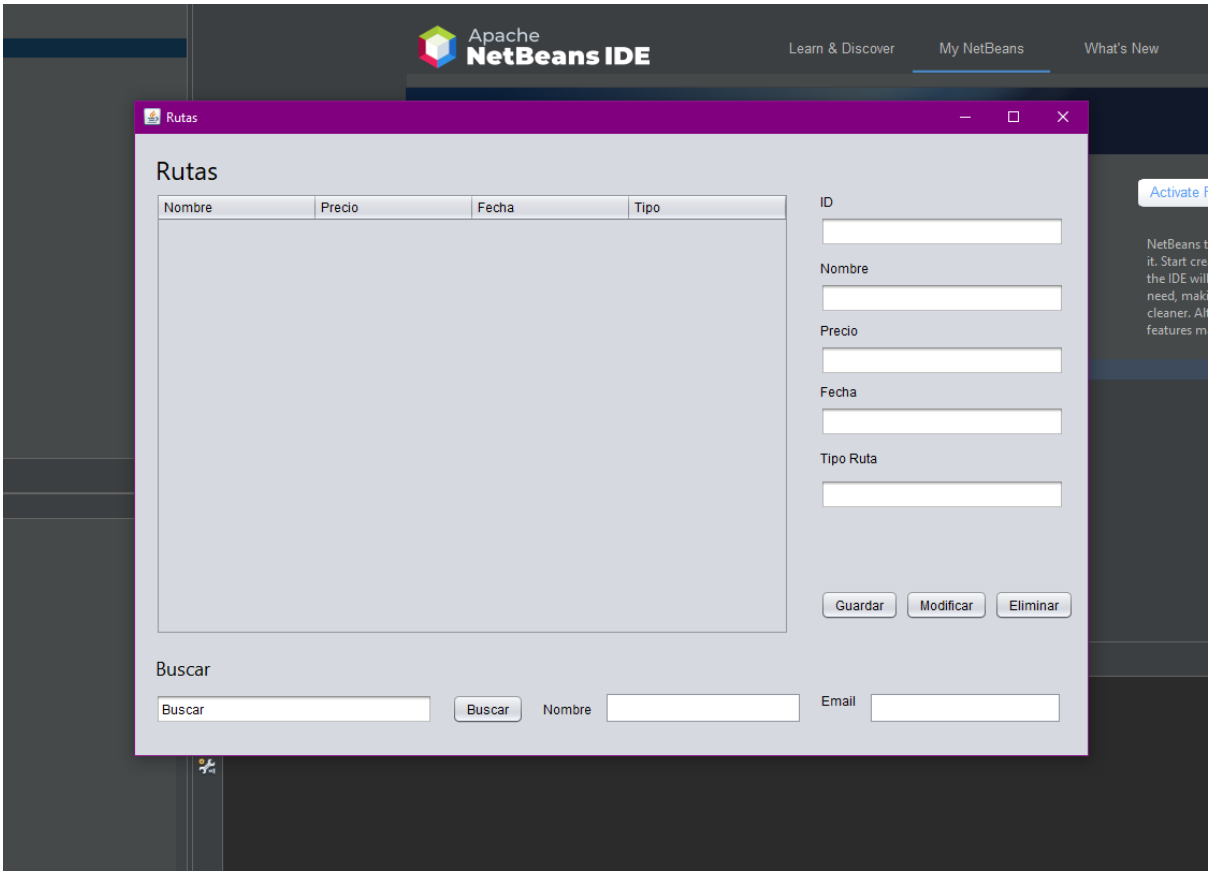
***Java:** Construirá un programa/Web en lenguaje Java que permita realizar la funcionalidad requerida.*

En mi caso he elegido utilizar java para desarrollar la app ya que estoy más acostumbrado al lenguaje y en concreto una aplicación de escritorio.

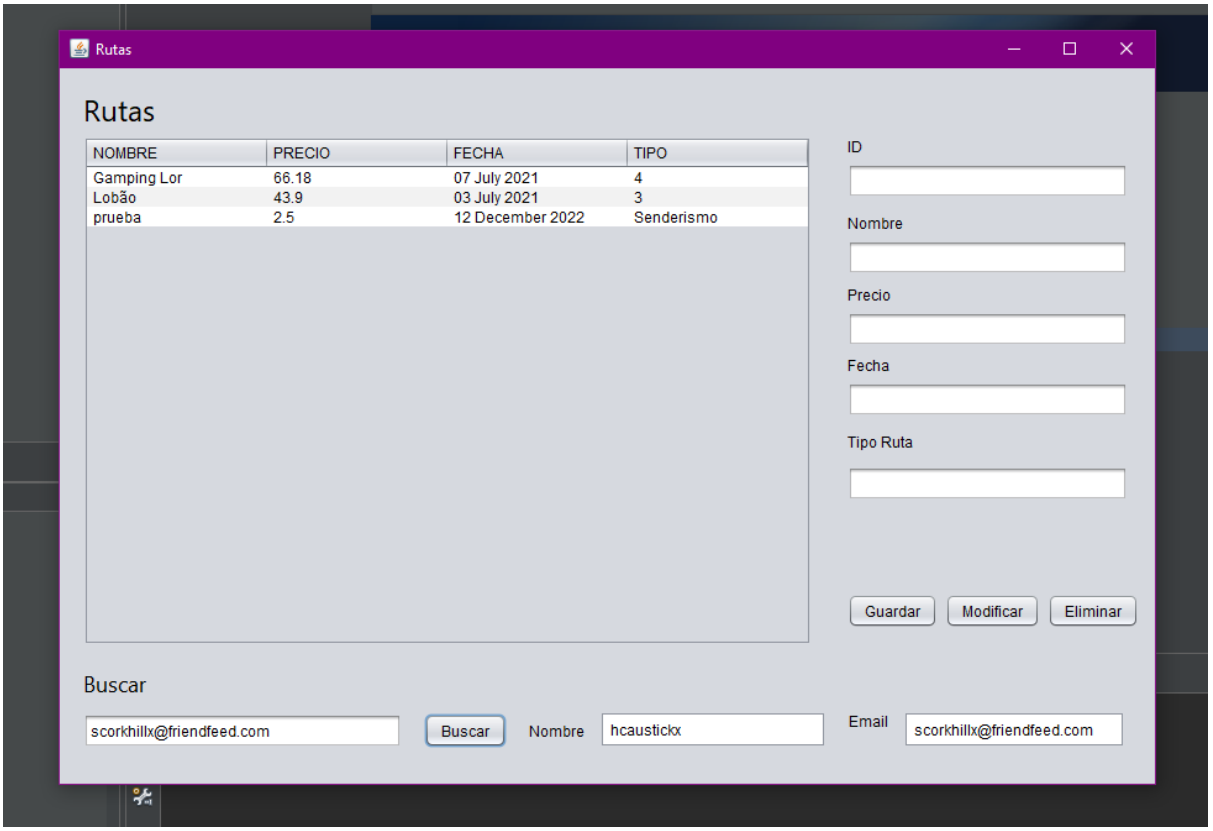
La aplicación ha sido desarrollada en la IDE **Netbeans.**



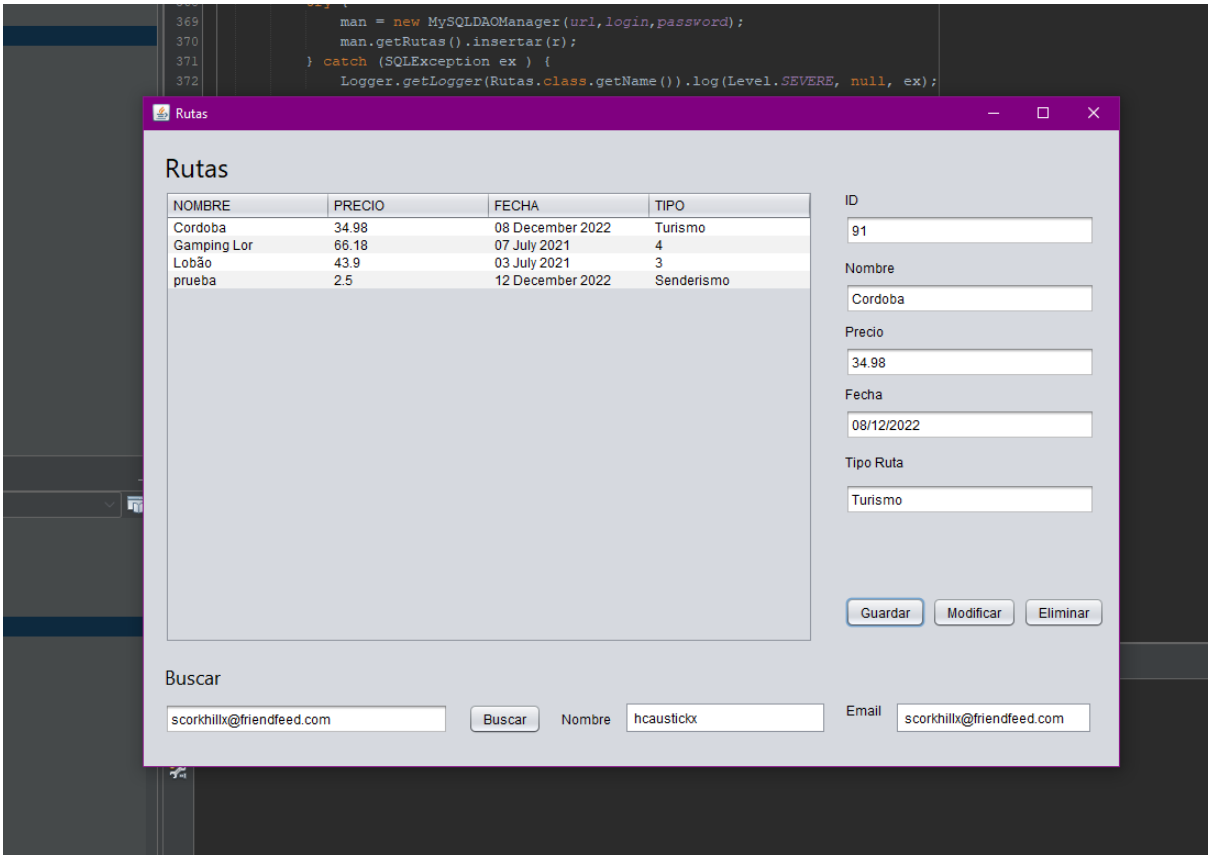
Aplicación sencilla con tres paquetes. Las clases usadas Ruta y Guia. Los Dao y la ventana que utilizaremos.



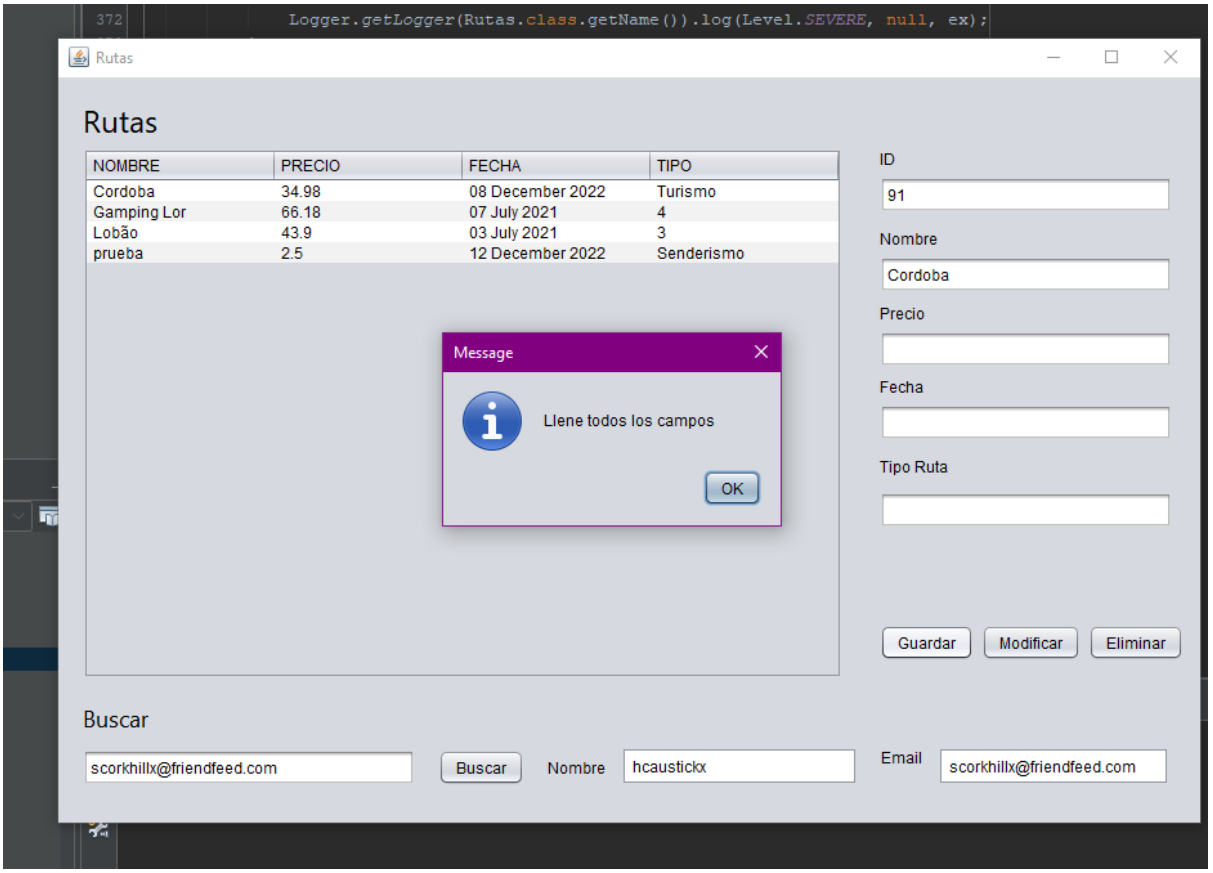
Introducimos un email y muestra las rutas del guía junto el nombre.



podemos añadir, modificar y eliminar .



Si no introducimos algun dato



Saldra mensaje de error

Aqui acabamos de borrar la ruta con la ide 91(Cordoba)