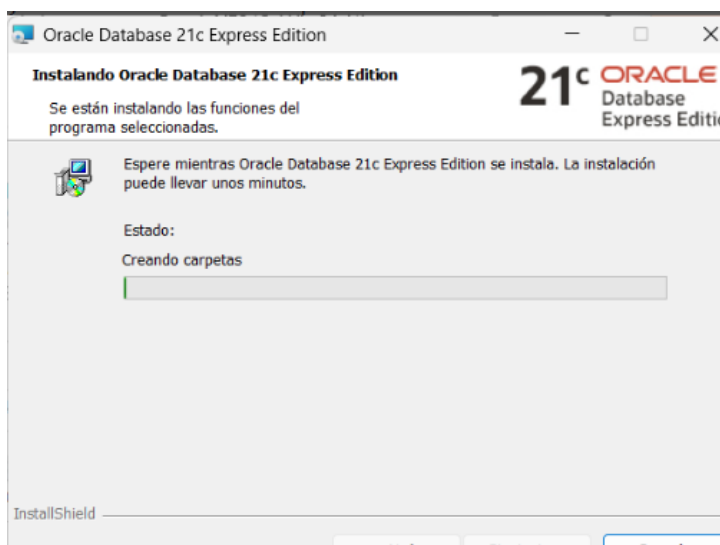
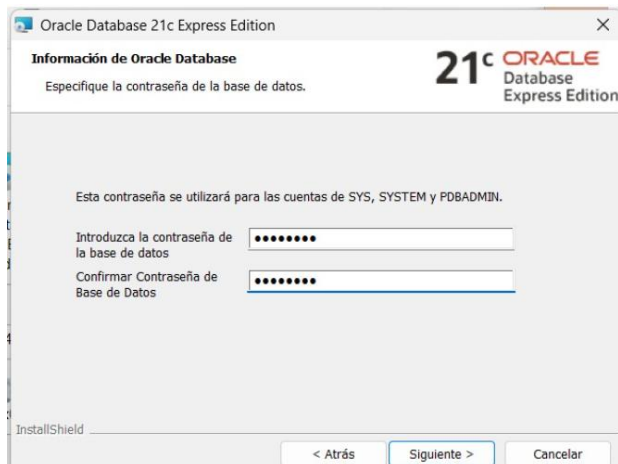


Instalación de oracle 11g (pasos y configuración)

Aunque el trabajo original solicitaba la instalación de Oracle Database 11g Express Edition, esta versión ya no está disponible para descarga directa desde el sitio oficial de Oracle. Por lo tanto, se optó por utilizar Oracle Database 21c Express Edition, que es la versión actual y gratuita de Oracle XE.



Conexión a Oracle Database 21c XE mediante SQL*Plus

```
SQL*Plus: Release 21.0.0.0.0 - Production on Sßb Nov 1 21:12:22 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Introduzca el nombre de usuario: system
Introduzca la contraseña:
Hora de Última Conexión Correcta: Sßb Nov 01 2025 21:03:28 -05:00

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
```

```
SQL> SELECT * FROM dual;

D
-
X

SQL> |
```

En estas capturas se muestran la conexión exitosa a la base de datos Oracle Database 21c Express Edition utilizando la herramienta SQL*Plus. El usuario system se autentica correctamente, y se verifica la conexión mediante la ejecución de la consulta SELECT * FROM dual;, que devuelve la fila X. Este paso confirma que la instalación de Oracle está funcional y que el entorno de desarrollo está listo para crear tablas, vistas, índices y otros objetos de esquema.

Diseño de esquema básico (diagrama entidad-relación)

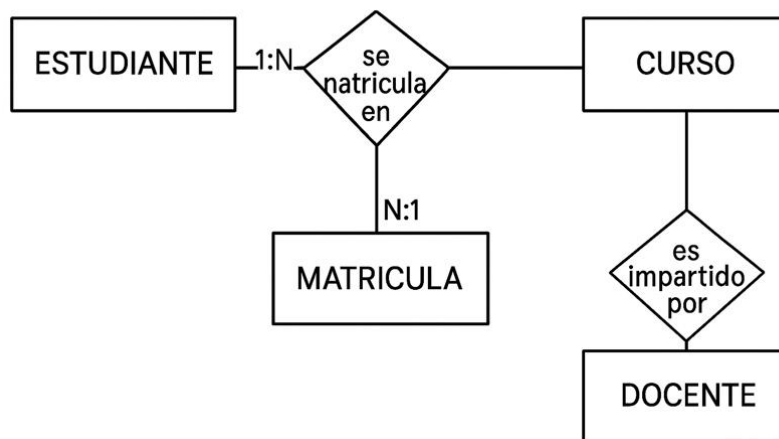
1. Entidades principales

Entidad	Atributos
ESTUDIANTE	id_estudiante (nombre, apellido, dni, email)
CURSO	id_curso (nombre, duración_semanas, modalidad)
DOCENTE	id_docente (nombre, especialidad, email)
MATRICULA	id_matricula (fecha, id_estudiante, id_curso)
ASIGNACIÓN	id_asignacion, id_docente, id_curso

2. Relación y cardinalidades

Relación	Cardinalidad	Descripción
ESTUDIANTE ---- MATRICULA	1 : N	Un estudiante puede tener muchas matrículas
CURSO ---- MATRICULA	1 : N	Un curso puede tener muchos estudiantes matriculados
DOCENTE ---- ASIGNACION	1 : N	Un docente puede impartir varios cursos
CURSO ---- ASIGNACION	1 : 1	Un curso solo puede tener un docente asignado

3. Diagrama ENTIDAD ---- RELACIÓN (DER)



El modelo está compuesto por cinco entidades. Estudiante, Curso y Docente son entidades fuertes, ya que poseen identificadores propios y existen de forma independiente en el sistema. Por otro lado, Matrícula funciona como una entidad asociativa que representa la relación entre Estudiante y Curso, permitiendo que un estudiante pueda inscribirse en múltiples cursos.

La relación entre Curso y Docente se establece mediante la relación “es impartido por”, asegurando que cada curso tenga un único docente responsable. Esta relación se representa explícitamente en el diagrama mediante un rombo, lo cual refleja claramente la dependencia entre ambas entidades.

Creación del esquema de Base de Datos

1. Creación de secuencias

```
SQL> CREATE SEQUENCE seq_estudiante START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;
Secuencia creada.

SQL> CREATE SEQUENCE seq_curso START WITH 100 INCREMENT BY 1 NOCACHE NOCYCLE;
Secuencia creada.

SQL> CREATE SEQUENCE seq_docente START WITH 200 INCREMENT BY 1 NOCACHE NOCYCLE;
Secuencia creada.

SQL> CREATE SEQUENCE seq_matricula START WITH 1000 INCREMENT BY 1 NOCACHE NOCYCLE;
Secuencia creada.
```

Se crean secuencias para generar valores únicos para las claves primarias de las tablas Estudiante, Curso, Docente y Matricula. Las secuencias se configuran sin caché ni ciclo (NOCACHE NOCYCLE) para evitar saltos en los números en un entorno de desarrollo.

2. Creación de tablas con restricciones

```
SQL> CREATE TABLE Estudiante (
2   id_estudiante NUMBER PRIMARY KEY,
3   nombre VARCHAR2(50) NOT NULL,
4   apellido VARCHAR2(50) NOT NULL,
5   dni VARCHAR2(10) UNIQUE,
6   email VARCHAR2(100)
7 );
Tabla creada.
```

```
SQL> CREATE TABLE Docente (
2   id_docente NUMBER PRIMARY KEY,
3   nombre VARCHAR2(100) NOT NULL,
4   especialidad VARCHAR2(50),
5   email VARCHAR2(100)
6 );
Tabla creada.
```

```
SQL> CREATE TABLE Curso (
2   id_curso NUMBER PRIMARY KEY,
3   nombre VARCHAR2(100) NOT NULL,
4   duracion_semanas NUMBER CHECK (duracion_semanas > 0),
5   modalidad VARCHAR2(20) CHECK (modalidad IN ('Presencial', 'Online')),
6   id_docente NUMBER,
7   CONSTRAINT fk_curso_docente FOREIGN KEY (id_docente) REFERENCES Docente(id_docente)
8 );
Tabla creada.
```

```
SQL> CREATE TABLE Matricula (
2   id_matricula NUMBER PRIMARY KEY,
3   fecha DATE DEFAULT SYSDATE,
4   id_estudiante NUMBER,
5   id_curso NUMBER,
6   CONSTRAINT fk_mat_est FOREIGN KEY (id_estudiante) REFERENCES Estudiante(id_estudiante),
7   CONSTRAINT fk_mat_curso FOREIGN KEY (id_curso) REFERENCES Curso(id_curso)
8 );
Tabla creada.
```

Se crean las tablas principales del esquema: Estudiante, Docente, Curso y Matricula. Se definen claves primarias, restricciones NOT NULL, UNIQUE y FOREIGN KEY para garantizar la integridad referencial.

3. Verificación de datos insertados

```
SQL> SELECT * FROM Estudiante;
```

ID_ESTUDIANTE	NOMBRE	
APELLIDO		DNI
EMAIL		
1	Grecia	
López		75123456
grecia@gmail.com		
2	Carlos	
Mendoza		75234567
carlos.mendoza@gmail.com		
ID_ESTUDIANTE	NOMBRE	
APELLIDO		DNI
EMAIL		

```
SQL> SELECT * FROM Docente;
```

ID_DOCENTE	
NOMBRE	
ESPECIALIDAD	
EMAIL	
200	
Dr. Ana Ruiz	
Desarrollo	
ana.ruiz@gmail.com	
ID_DOCENTE	
NOMBRE	
ESPECIALIDAD	
EMAIL	
201	
Ing. Luis Torres	
Bases de Datos	
luis.torres.bd@gmail.com	

```
SQL> SELECT * FROM Curso;
```

ID_CURSO		
NOMBRE		
DURACION_SEMANAS	MODALIDAD	ID_DOCENTE
100		
Programación en Python		
12	Online	200
101		
Bases de Datos		
10	Presencial	201
ID_CURSO		
NOMBRE		
DURACION_SEMANAS	MODALIDAD	ID_DOCENTE

```
SQL> SELECT * FROM Matricula;
```

ID_MATRICULA	FECHA	ID_ESTUDIANTE	ID_CURSO
1000	02/11/25	1	100
1001	02/11/25	2	101

Se ejecutan consultas `SELECT * FROM...` para verificar que los datos de prueba se hayan insertado correctamente en todas las tablas. Esto confirma que las relaciones entre tablas funcionan según el diseño del modelo.

Conexión básica desde lenguaje elegido:

Conexión desde Python a Oracle Database 21c XE

```
Microsoft Windows [Versión 10.0.26100.4351]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USER>cd C:\Users\USER\Downloads\1BD_proyecto

C:\Users\USER\Downloads\1BD_proyecto>python conexion_oracle.py
Conexión exitosa

Grecia López
Carlos Mendoza

Conexión cerrada.
```

1. Creación de la vista

```
SQL> CREATE VIEW vista_estudiantes AS
2  SELECT id_estudiante, nombre, apellido, email
3  FROM ESTUDIANTE;

Vista creada.
```

Se crea una vista simple llamada `vista_estudiantes` que selecciona todas las columnas de la tabla `ESTUDIANTE`. Las vistas permiten simplificar consultas complejas y restringir el acceso a datos.

2. Creación del índice

```
SQL> CREATE INDEX idx_est_apellido ON ESTUDIANTE(apellido);

Índice creado.
```

Se crea un índice no único sobre la columna `apellido` de la tabla `ESTUDIANTE`. Los índices mejoran el rendimiento de las consultas que filtran o ordenan por esa columna.

3. Creación del sinónimo

```
SQL> CREATE SYNONYM est FOR ESTUDIANTE;  
Sinónimo creado.
```

Se crea un sinónimo privado est como alias para la tabla ESTUDIANTE, facilitando su uso en consultas futuras y abreviando el nombre del objeto.

4. Verificación de los objetos creados: vista y sinónimo.

```
SQL> SELECT * FROM vista_estudiantes;  
ID_ESTUDIANTE NOMBRE  
-----  
APELLIDO  
-----  
EMAIL  
-----  
1 Grecia  
López  
greCIA@gmail.com  
2 Carlos  
Mendoza  
carlos.mendoza@gmail.com  
ID_ESTUDIANTE NOMBRE  
-----  
APELLIDO  
-----  
EMAIL  
-----  
  
SQL> SELECT * FROM est;  
ID_ESTUDIANTE NOMBRE  
-----  
APELLIDO DNI  
-----  
EMAIL  
-----  
1 Grecia  
López 75123456  
greCIA@gmail.com  
2 Carlos  
Mendoza 75234567  
carlos.mendoza@gmail.com  
ID_ESTUDIANTE NOMBRE  
-----  
APELLIDO DNI  
-----  
EMAIL  
-----
```