

El alumno/a ha de realizar una aplicación en Java que utilizando Hibernate lleve la gestión de una Academia de Idiomas. La base de datos se almacena en un SGBD Oracle, cuyo script se incluye más abajo.

La pantalla principal de la aplicación será la siguiente:

Codigo Alumno	Nombre
001	Antonio
002	Maria

Codigo Curso	Nombre Curso	N° Exámenes
I001	Ingles Basico	5
I002	Ingles Intermedio	8
I003	Ingles Avanzado	10
F001	Frances Basico	3
C002	Chino Intermedio	9

Matricular Alumno en Curso

Codigo Alumno	Nombre Alumno	Codigo Curso	Nombre Curso	Nota Media
001	Antonio	I001	Ingles Basico	7
001	Antonio	F001	Frances Basico	0

Numero Examen	Fecha Examen	Nota
1	01/10/2019	6
2	15/10/2019	8
3		0
4		0
5		0

Fecha Examen: 30/10/2019

Nota:

Actualizar

Boletin JSON

Listado Matricula XML

- Tendremos un JTable donde se mostrarán nuestros Alumnos (**JTableAlumnos**).
- Tendremos un JTable donde se mostrarán nuestros Cursos (**JTableCursos**).
- Tendremos un JTable donde se mostrarán los Cursos en los que está matriculado cada Alumno (**JTableMatriculas**). Al seleccionar un alumno del JTableAlumnos, se actualizara el JTable JTableMatriculas. Al seleccionar un curso del JTableCursos, se actualizara el JTable JTableMatriculas.
- Tendremos un JTable donde se mostrarán los Exámenes realizados por un Alumno en un Curso determinado (**JTableExámenes**). Al seleccionar una matrícula del JTableMatriculas, se actualizara el JTable JTableExámenes.
- Al seleccionar un examen, se mostrará en los JTextField correspondientes la fecha y la nota que se obtuvo en ese examen.

TAREAS A REALIZAR:

- ✓ 1. La carga de datos de los JTable iniciales. (1.5 puntos)
- ✓ 2. Gestión de Alumnos : alta, baja, consulta y modificación. Se incluirá un botón en la pantalla principal que abrirá un JDialog para realizar las opciones de CRUD de alumnos. (1.5 puntos)
 - ✓ a. La consulta se podrá realizar mediante el código del alumno o filtrando aquellos alumnos que cumplan una condición (a elección del alumno/a).
 - ✓ b. No se puede borrar ningún alumno que esté matriculado en algún curso.
- ✓ 3. Gestión de Cursos : alta, baja, consulta y modificación. Se incluirá un botón en la pantalla principal que abrirá un JDialog para realizar las opciones de CRUD de cursos. (1.5 puntos)
 - ✓ a. La consulta se podrá realizar mediante el código del curso o filtrando aquellos cursos que cumplan una condición (a elección del alumno/a).
 - ✓ b. No se puede borrar ningún curso que tenga alumnos matriculados.
- ✓ 4. Al presionar el botón de **Matricular Alumno en Curso** se ejecutará un procedimiento almacenado en Oracle que generará un registro en la tabla Matriculas para el alumno seleccionado en el Jtable de alumnos y el curso seleccionado en el Jtable de Cursos y tantos exámenes como indique el curso. Controlar los posibles errores de la ejecución del procedimiento. (1.5 puntos)
- ✓ 5. Al presionar el botón **Actualizar** se actualizará la fecha y nota en que un alumno realizó el examen. Así como la nota media de la matrícula del alumno en ese curso. (2 puntos)
- ✓ 6. Al presionar el botón **Boletín JSON** se generará un fichero de texto con formato JSON donde se mostrarán los datos de los exámenes de la matrícula seleccionada. (1 punto)
- ✓ 7. Al presionar el botón **Listado Matriculas XML** se generará un fichero de texto con formato XML donde se mostrarán los datos de TODOS los alumnos con los cursos en los que está matriculado cada uno de ellos. (1 punto)

NOTA: Los apartados 1, 2, 3 y 5 se realizarán exclusivamente con Hibernate
Fecha máxima de entrega : 07/12/2022 a la 15:00

ENTREGAR AL PROFESOR UN FICHERO RAR CON :

CARPETA CON EL NOMBRE DEL ALUMNO, DONDE SE ALMACENARA EL PROYECTO JAVA, UN EJECUTABLE JAR (OBLIGATORIO Y DE EJECUCION EN CUALQUIER PC) Y UN FICHERO DE TEXTO CON LAS OBSERVACIONES PERTINENTES.

TODOS LOS FICHEROS DEBEN TENER RUTA RELATIVA

```
CREATE USER AD_ACADEMIA IDENTIFIED BY AD_ACADEMIA;  
GRANT DBA TO AD_ACADEMIA;
```

```
CREATE TABLE ALUMNOS(  
    cCodAlu VARCHAR2(6) CONSTRAINT PK_ALUMNOS PRIMARY KEY,  
    cNomAlu VARCHAR2(100) NOT NULL  
);
```

```
CREATE TABLE CURSOS(  
    cCodCurso VARCHAR2(6) CONSTRAINT PK_CURSOS PRIMARY KEY,  
    cNomCurso VARCHAR2(100) NOT NULL,  
    nNumExa  NUMBER(3) DEFAULT 1 NOT NULL  
);
```

```
CREATE TABLE MATRICULAS(  
    cCodAlu  VARCHAR2(6) NOT NULL,  
    cCodCurso VARCHAR2(6) NOT NULL,  
    nNotaMedia  NUMBER(3) DEFAULT 0 NOT NULL,  
    CONSTRAINT PK_MATRICULAS PRIMARY KEY (cCodAlu,cCodCurso)  
);
```

```
ALTER TABLE MATRICULAS  
ADD CONSTRAINT FK_MATRICULAS_ALUMNO FOREIGN KEY (cCodAlu)  
REFERENCES ALUMNOS(cCodAlu);
```

```
ALTER TABLE MATRICULAS  
ADD CONSTRAINT FK_MATRICULAS_CURSOS FOREIGN KEY (cCodCurso)  
REFERENCES CURSOS(cCodCurso);
```

```
CREATE TABLE EXAMENES(  
    cCodAlu  VARCHAR2(6) NOT NULL,  
    cCodCurso VARCHAR2(6) NOT NULL,  
    nNumExam  NUMBER(3) DEFAULT 1 NOT NULL,  
    dFecExam  DATE,  
    nNotaExam  NUMBER(6,2) DEFAULT 0 NOT NULL,  
    CONSTRAINT PK_EXAMENES PRIMARY KEY (cCodAlu,cCodCurso,nNumExam)  
);
```

```
ALTER TABLE EXAMENES  
ADD CONSTRAINT FK_EXAMENES_MATR FOREIGN KEY (cCodAlu,cCodCurso)  
REFERENCES MATRICULAS(cCodAlu,cCodCurso);
```

```
INSERT INTO ALUMNOS VALUES ('001','Antonio');  
INSERT INTO ALUMNOS VALUES ('002','Maria');
```

```
INSERT INTO CURSOS VALUES ('I001','Ingles Basico',5);  
INSERT INTO CURSOS VALUES ('I002','Ingles Intermedio',8);  
INSERT INTO CURSOS VALUES ('I003','Ingles Avanzado',10);  
INSERT INTO CURSOS VALUES ('F001','Frances Basico',3);
```

```
INSERT INTO CURSOS VALUES ('C002','Chino Intermedio',9);  
COMMIT;
```

```
CREATE OR REPLACE PROCEDURE sp_AltaMatricula (xcodAlu VARCHAR2, xcodCurso VARCHAR2, xError OUT  
NUMBER)  
IS  
    xNR NUMBER;  
BEGIN  
    xError := 0;  
    INSERT INTO MATRICULAS VALUES (xcodAlu, xcodCurso,0);  
    SELECT nNumExa INTO xNR FROM CURSOS WHERE cCodCurso = xcodCurso;  
    FOR i IN 1..xNR LOOP  
        INSERT INTO EXAMENES (cCodAlu,cCodCurso,nNumExam) VALUES (xcodAlu, xcodCurso,i);  
    END LOOP;  
    xError := xNR;  
EXCEPTION  
    WHEN OTHERS THEN xError := -1;  
END;
```