

PAC DESARROLLO

CFGS Desarrollo de Aplicaciones

Módulo 06:



Acceso a Datos

1S 2022/2023

INFORMACIÓN IMPORTANTE

Para la correcta realización de la PAC el alumno deberá consultar los contenidos recogidos en el **Tema 1, Tema 2 de la UF2 y los temas de la UF3** del material didáctico.

Requisitos que deben cumplirse en vuestros trabajos:

- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- La PAC debe entregarse en formato ZIP.
- Este ZIP, contendrá el proyecto realizado en Java
- En el caso de no realizarse la entrega en dicho formato el alumno se hace responsable de posibles incompatibilidades en la visualización de su entrega y por ende afectará a su calificación.
- Para que el ejercicio sea corregido, este debe **entregarse obligatoriamente en la tarea**. No se corregirán proyectos entregados por otros medios, como los adjuntos de los comentarios de la tarea o los mensajes privados.

CRITERIOS DE CORRECCIÓN

1. Todos los programas realizados en la PAC deben realizarse con IDE con que se pueda trabajar con el lenguaje Java
2. Para la realización de esta PAC es necesario que se utilicen las estructuras de control y las estructuras repetitivas siempre que sea posible.
3. Se deben poner comentarios para su mejor comprensión. Estos comentarios explicarán la funcionalidad del código. Se valorarán los comentarios en la parte de presentación.
4. Si se detecta la entrega de una PAC copiada, ya sea de una fuente externa o con un contenido idéntico al de otro alumno/a, serán evaluadas como suspenso, con una calificación de 0.

En esta PAC se van a valorar vuestros conocimientos sobre Hibernate siguiendo unas especificaciones.

Realizaremos un ejercicio en el que tendréis que crear la configuración e ingresar los datos necesarios.

¡Es muy importante seguir todas las puntuaciones que realizo sobre la PAC, deberéis de poner los mismos nombres de atributos y métodos!

Estructura de la PAC.

Tendrás que realizar el ejercicio usando Maven.

Todas las clases Java deberán estar incluidas dentro del package pac.

Por lo que la estructura del proyecto en Maven A LA HORA DE REALIZARLO sería (recuerda que es **obligatoria**):

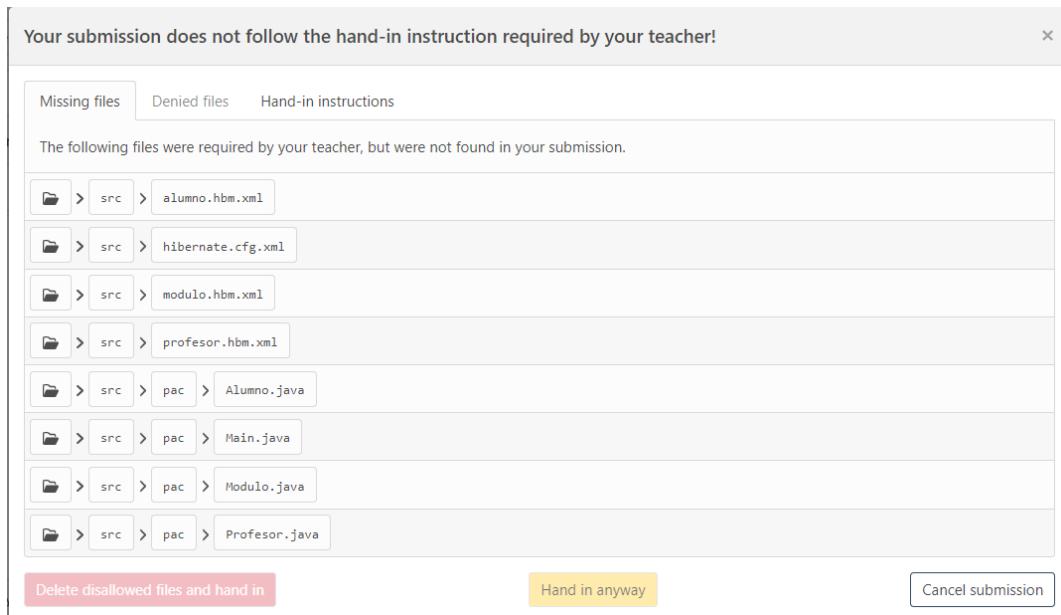
- src
 - main
 - java
 - pac
 - *.java
 - resources
 - *.xml

Y la ESTRUCTURA DE LAS CARPETAS DE ENTREGA OBLIGATORIA:

Archivo: PAC_UF3_ApellidoNombre.zip y dentro directamente la carpeta src; es decir /PAC_UF3_ApellidoNombre.zip/src/

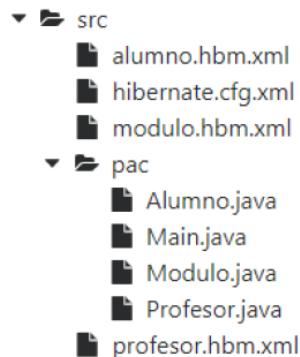
COMPROBAR QUE DENTRO DEL ARCHIVO ZIP APARECE DIRECTAMENTE la carpeta src; es decir que la ruta sea /PAC_UF3_ApellidoNombre.zip/src

Si no es así; el codegrade no aceptará vuestra entrega, y aparecerá el siguiente mensaje: que lo que viene a decir es que no se están entregando los archivos requeridos. Revisa la videotutoría de la PAC de desarrollo, para ver como se crea el archivo zip correctamente.



CONTENIDO ARCHIVO A SUBIR EN LA PAC

- La estructura que debéis comprimir para enviarla a la plataforma es:



* ESTRUCTURA DE LAS CARPETAS DE ENTREGA OBLIGATORIA, no más archivos ni menos.

* Especial énfasis en los nombres de las Clases y de los archivos de configuración.xml;

*Se suspenderá la pac de desarrollo, si no se pueden pasar las pruebas unitarias; es decir, si no se siguen las especificaciones del enunciado. No se trata solo que funcione, se trata que funciones siguiendo las especificaciones del enunciado.

Ejercicios:

1. Crear los ficheros de configuración para Hibernate con los datos:
 - a. Connexión a 127.0.0.1, puerto 3306
 - b. Username: ilerna
 - c. Password: Ilerna_1234
 - d. Base de datos: m06
2. Crea las clases con los atributos expuestos a continuación, también deberás crear las funciones necesarias para que funcione con hibernate y el método `toString()` de cada clase.
 - a. Modulo:
 - Long id nombre de la columna idModulo
 - String nombre
 - String codigo
 - b. Alumno:
 - Long id nombre de la columna idAlumno
 - String nombre
 - String nacionalidad
 - int edad
 - String sexo
 - Set<Modulo> modulos
 - 1. La relación entre Alumno y Modulo será llamada alumno_modulo
 - c. Profesor:
 - Long id nombre de la columna idProfesor
 - String nombre
 - String sexo
3. Crea los archivos de configuración hibernate para dichas clases (Alumno, Profesor y Modulo).
Todos los ID se deberán establecer de forma autonumérica.
 - a. Las tablas resultantes tendrán el nombre:
 - alumnos
 - modulos
 - alumno_modulo
 - profesores
4. Creación de funciones que realizaran los inserts en la BD con hibernate, después de cada inserción debe mostrar un mensaje por pantalla.
 - a. Modulo (nombre, codigo):
 - programación B, M03B
 - Acceso a Datos, M06
 - Desarrollo de aplicaciones móviles, M08
 - Servicios y procesos, M09Mensaje por pantalla:
`Insert into modulo, nombre: {nombre}, codigo {codigo}`
Ejemplo:
`Insert into módulo, nombre: Programación B, código: M03B`
 - b. Profesor (nombre, sexo)
 - Alvaro, HombreMensaje por pantalla:
`Insert into profesor, nombre: {nombre}, sexo: {sexo}`
Ejemplo:
`Insert into profesor, nombre: Alvaro, sexo: Hombre`

- c. Alumno (nombre, nacionalidad, edad, sexo, set<módulos>)
- Juan, Espaniola, 26, Hombre, Modulos (1,2,3,4)
 - Pedro, Andorrana, 21, Hombre, Modulos (1,2,4)
 - Marta, Espaniola, 19, Mujer, Modulos (3,4)
 - Carla, Francesa, 35, Mujer, Modulos (2,3,4)

Mensaje por pantalla:

Insert into alumno, nombre: {nombre}, nacionalidad:
{nacionalidad}, edad: {edad}, sexo: {sexo}, módulos: {nº módulos}

Ejemplo:

Insert into alumno, nombre: Juan, nacionalidad: Espaniola, edad
26, sexo: Hombre, módulos: 4

5. Creación de una clase con nombre Main donde se realizará la conexión con Hibernate, se ejecutarán las funciones de insert anteriores.

Debido a discrepancias entre idiomas al programar, es recomendable no establecer caracteres especiales como: (` ~ ¸ ñ)

Es por ello qué:

- nacionalidad Española = Espaniola.
- Módulo = modulo
- Programación = programacion

¡Ánimo con la pac!

