

DAM. UNIDAD 4. ACCESO MEDIANTE COMPONENTES. TAREA EVALUABLE 4

DAM. Acceso a Datos (ADA) (a
distancia en inglés)

Unidad 4. ACCESO MEDIANTE COMPONENTES

Tarea evaluable 4

Abelardo Martínez

Año 2023-2024

Aspectos a tener en cuenta

Importante

Si buscas las soluciones navegando por Internet o preguntando al oráculo de ChatGPT te estarás engañando a ti mismo. Ten en cuenta que ChatGPT no es infalible ni todopoderoso.

Es una gran herramienta para agilizar tu trabajo una vez que dominas un tema, pero utilizarla como atajo a la hora de adquirir habilidades y conocimientos básicos perjudica seriamente tu aprendizaje. Si lo utilizas para obtener soluciones o consejos por tu cuenta, comprueba también cuidadosamente las soluciones propuestas. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación ampliada que encontrarás en el "Aula Virtual".

Consejos para programar

Aconsejamos seguir las siguientes normas de codificación:

- Una instrucción por línea.
- Añade comentarios para que tu código sea más claro y legible.
- Utiliza la notación húngara para reconocer el tipo de variables a primera vista.
- Recuerde que hay varias formas de aplicar una solución, así que elija la que más le guste. Recomendamos encarecidamente utilizar `soluciones` basadas en `tampones`.

A. Instrucciones y directrices

El proyecto DEBE realizarse en Java. Otras tecnologías -como Spring Boot- no serán soportadas. Para su desarrollo se puede utilizar cualquiera de los IDEs propuestos en la unidad 1, aunque se recomienda encarecidamente Eclipse.

1. VISIÓN GENERAL

Deberá crear por su cuenta una aplicación Java que utilice los conceptos impartidos durante la UNIDAD 4 para cumplir una especificación proporcionada.

2. CALENDARIO Y EXPECTATIVAS

- Porcentajes dentro del TERM: 50% del total del TERM (AT3 haría el otro 50%)
- Porcentajes dentro de la TAREA: 100% competencias ADA (las competencias en inglés deben ser APROBADAS).
- Plazo: 23:59 horas del domingo 18 de febrero de 2024 (3 SEMANAS)

3. CALIFICACIÓN

Debes obtener 5 puntos sobre 10 en ADA y un COMPETENTE en inglés para aprobar esta TAREA EVALUABLE.

Junto con este documento se proporcionará una escala de calificaciones detallada (consulte la RÚBRICA DE APRENDIZAJE).

4. RECURSOS

Deberás hacer una lectura exhaustiva de todos los materiales proporcionados por tu profesor, así como de las tareas no evaluables, pero también bucear en Internet para encontrar ejemplos que proporcionen resultados similares a los requeridos por esta tarea.

Siéntete libre de copiar y pegar código de CUALQUIER recurso siempre que entiendas cada parte de l mismo ya que se te pedirá que defiendas tu trabajo en una reunión individual.

5. PLAGIARISMO

No debes permitir que otros estudiantes copien tu trabajo y debes tener cuidado para evitar que esto ocurra.

En caso de sospecha de plagio, podría exigirse una entrevista oral adicional.

6. ENTREGA Y RETROALIMENTACIÓN

- La tarea se entregará SÓLO en un archivo con formato ZIP, comprimiendo la carpeta del proyecto desde su IDE (es decir, Eclipse).
- Posteriormente, ESTARÁ OBLIGADO a asistir a una entrevista oral con su profesor para discutir ciertos aspectos de su tarea en inglés durante un máximo de 15 minutos.

- Recibirá sus notas desglosadas por cada criterio y el total, junto con cualquier comentario con sugerencias sobre cómo podría haberlo hecho mejor.

B. Detalles de la evaluación

SÓLO SE PERMITE EL INGLÉS para la realización de la tarea evaluable, tanto los comentarios como los textos explicativos/aclaratorios.

1. CADA MÉTODO DEBE ESTAR DEBIDAMENTE DESCRITO CON SUS PROPIAS PALABRAS. En el principio de cada método debes añadir comentarios para explicar con tus propias palabras cómo funciona.
2. ADEMÁS, DEBES AÑADIR UN TEXTO EXPLICANDO CON TUS PROPIAS PALABRAS, TU EXPERIENCIA IMPLEMENTANDO ESTA SOLUCIÓN.

Cree un archivo de texto y cópielo en la carpeta del proyecto o cree el archivo de texto dentro del propio proyecto en el IDE de Eclipse.

- PÁRRAFO 1. Describa brevemente la solución aportada.
- PÁRRAFO 2. Describa brevemente las dificultades encontradas.
- PÁRRAFO 3. Describa brevemente varias posibles ampliaciones que haya recomendado.

B.1. Características obligatorias

Actividad (EVALUABLE)

Crear un programa en Java para gestionar operaciones básicas de una base de datos MySQL llamada DBChessGames con la estructura proporcionada vía JDBC, para obtener una Salida como la que también se proporciona, importando una librería jar que contiene varios JavaBeans (entre otras clases auxiliares). Ver apéndices para información detallada.

Por favor, siga estas **ESPECIFICACIONES TÉCNICAS**:

- RDBMS: MySQL
- Idioma: Java
- Marco: Ninguno
- ORM: Ninguno
- DAO: JavaBeans (archivo jar)

La aplicación (método main) irá así:

1. Lee un ajedrecista aleatorio y un torneo de la base de datos (pl+t).
2. Simular que el jugador está jugando una partida de ajedrez en ese torneo (Partida).
3. Simular lo mismo (pl+t) solicita un aplazamiento de una partida de ajedrez (Aplazamiento).

Deberías crear una clase JavaBean llamada PlayerBean, actuando como SOURCE, que estará interactuando con otras 3 clases JavaBean llamadas MessageBean, GameBean y DeferralBean, actuando como LISTENERS.

Además, utilizará otras 2 clases AUXILIARES llamadas Tournament y DBBean.

a) Cuando el jugador de ajedrez juega un partido en un torneo, una propiedad llamada nextMatchDate cambiará de NULL a la fecha especificada y los oyentes reaccionarán a este cambio ...

- MessageBean insertará un mensaje en la tabla de mensajes y lo imprimirá (ver captura de pantalla de salida).

- GameBean insertará un partido en la tabla Game y actualizará la tabla players indicando que tiene un partido (ver captura de pantalla de salida).

b) Cuando el ajedrecista pide un aplazamiento en un torneo, una propiedad llamada nextDeferralDate cambiará de NULL a la fecha especificada y los oyentes reaccionarán a este cambio ...

- MessageBean insertará un mensaje en la tabla de mensajes y lo imprimirá (ver captura de pantalla de salida).
- DeferralBean insertará un aplazamiento en la tabla Deferral y actualizará la tabla de jugadores indicando que tiene un aplazamiento (ver captura de pantalla de salida).

c) Cuando el jugador de ajedrez termina un partido, una propiedad llamada nextMatchDate cambiará de la fecha especificada a NULL y los oyentes reaccionarán a este cambio ...

- MessageBean insertará un mensaje en la tabla de mensajes y lo imprimirá (ver captura de pantalla de salida).
- GameBean actualizará el partido en la tabla Game y actualizará la tabla players indicando que ya no tiene partido (ver captura de pantalla de salida).

d) Cuando el ajedrecista recibe el veredicto de un aplazamiento, una propiedad llamada nextDeferralDate cambiará de la fecha especificada a NULL y los oyentes reaccionarán a este cambio ...

- MessageBean insertará un mensaje en la tabla de mensajes y lo imprimirá (ver captura de pantalla de salida).
- GameBean actualizará el aplazamiento en la tabla Aplazamiento y actualizará la tabla de jugadores indicando que ya no tiene aplazamiento (ver captura de pantalla de salida).

B2 Funciones opcionales

Actividad (EVALUABLE)

Para obtener la máxima puntuación, debe añadir el código adicional para evitar que cualquier jugador solicite un aplazamiento si no hay ningún JUEGO en estado PENDIENTE relacionado con ese jugador.

C. Rúbrica de aprendizaje

C.1. ADA skills

Se requiere un mínimo de 5 sobre 10 para esta parte.

Estas notas se invalidarán (nota 4) si no defiende su trabajo en una entrevista oral.

ELEMENTOS DE EVALUACIÓN	DETALLES DEL ELEMENTO DE EVALUACIÓN	PUNTUACIÓN (PUNTOS)
Archivo JAR	Se importa el archivo jar.	0.5
Judías	Utilización e interacción entre los JavaBeans.	3.5
Clases	Uso de clases auxiliares.	1
Base de datos	Interacción con la base de datos.	3
[Opcional] Evitar solicitar un aplazamiento si no hay ningún partido en estado pendiente.		2

C2. Inglés skills

Obligatorio ser COMPETENTE para aprobar esta parte.

ELEMENTOS DE EVALUACIÓN	DETALLES DEL ELEMENTO DE EVALUACIÓN	PUNTUACIÓN
Capacidad de redacción	Cada método se describe correctamente. Se proporciona un texto adecuado (dentro del código o en un archivo de texto) para describir la TA utilizando TRES PÁRRAFOS .	COMPETENTE/NO COMPETENTE
Competencias orales	Utiliza un vocabulario adecuado al propósito. Muestra fluencia y confianza.	COMPETENTE/NO COMPETENTE
Capacidad de comprensión		Cumplido ya que todos los materiales están en inglés
Capacidad de lectura		Cumplido ya que todos los materiales están en inglés

D. Apéndices

D.1. Base de datos

```
DROP DATABASE IF EXISTS DBChessGames;
CREAR BASE DE DATOS DBChessGames CHARACTER SET utf8 COLLATE utf8_spanish_ci;

-- CREATE USER 'mavenuser'@'localhost' IDENTIFIED WITH mysql_native_password
-- GRANT ALL PRIVILEGES ON DBChessGames.* to 'mavenuser'@'localhost'; USE

DBChessGames;

DROP TABLE IF EXISTS Jugador;
DROP TABLE IF EXISTS Torneo; DROP
TABLE IF EXISTS Juego;
DROP TABLE IF EXISTS
Aplazamiento; DROP TABLE IF
EXISTS Mensaje;

-- Jugador
CREAR TABLA Jugador (
playerID          INTEGRO,
nombre completo  VARCHAR(100) NO NULO,
has_match        BOOLEAN DEFAULT FALSE,
has_deferral     BOOLEAN DEFAULT FALSE,
CONSTRAINT pla_id_pk PRIMARY KEY (playerID)
);

-- Torneo
CREAR TABLA Torneo ( código
                  VARCHAR(10),
nombre          VARCHAR(100),
CONSTRAINT tou_cod_pk PRIMARY KEY (código)
);
```

```
-- Juego
CREAR TABLA Juego (
  gameId      INTEGER AUTO_INCREMENT,
  código      VARCHAR(10),
  playerId    INTEGER,
  matchdate   FECHA,
  resultado    VARCHAR(10) POR DEFECTO 'PENDIENTE',
  CONSTRAINT gam_gam_pk PRIMARY KEY (gameID),
  CONSTRAINT gam_cod_fk FOREIGN KEY (code) REFERENCIAS Torneo(code), CONSTRAINT
  gam_pla_fk FOREIGN KEY (playerID) REFERENCIAS Jugador(playerID),
  CONSTRAINT gam_res_ck CHECK (resultado E ('PENDIENTE','GANADO','PERDIDO','EMPAT
  ) N ES'))
);

-- Aplazamiento
CREAR TABLA Aplazamiento (
  deferralID  INTEGER AUTO_INCREMENT,
  código      VARCHAR(10),
  playerId    INTEGER,
  defdate     DATETIME,
  resultado    VARCHAR(10) POR DEFECTO 'SOLICITADO',
  CONSTRAINT def_def_pk PRIMARY KEY (deferralID),
  CONSTRAINT def_cod_fk FOREIGN KEY (code) REFERENCIAS Torneo(code), CONSTRAINT
  def_pla_fk FOREIGN KEY (playerID) REFERENCIAS Jugador(playerID),
  CONSTRAINT def_res_ck CHECK (resultado E ('SOLICITADO','CONCEDIDO','RECHAZADO'))
  ) N
);

CREAR TABLA Mensaje (
  messageID   ENTERO AUTO_INCREMENT,
  playerId    INTEGER,
  descripción VARCHAR(500),
  CONSTRAINT mes_mmc_pk PRIMARY KEY (messageID),
  CONSTRAINT mes_pla_fk FOREIGN KEY (playerID) REFERENCIAS Player(playerID)
  );

-- Datos
INSERT INTO Player (playerID, fullname) VALUES (1, 'Anatoly Karpov'); INSERT
INTO Player (playerID, fullname) VALUES (2, 'Garry Kasparov');
```

```
INSERT INTO Player (playerID, fullname) VALUES (3, 'Pepe Tableros');  
INSERT INTO Player (playerID, fullname) VALUES (4, 'John Lost');  
INSERT INTO Tournament (code, name) VALUES ('T01', 'Torneo local de Quart de  
INSERT INTO Tournament (code, name) VALUES ('T02', 'Torneo Internacional de A  
INSERT INTO Tournament (code, name) VALUES ('T03', 'Torneo Jaque con tomate')  
INSERT INTO Tournament (code, name) VALUES ('T04', 'Torneo Maestro Yoda');
```

D2 Clases mínimas y propiedades recomendadas

Propiedades sugeridas por la clase PlayerBean:

- private String stFullname;
 - torneo privado objTournament;
 - boolean privado bHasAMatch;
 - boolean privado bHasADeferral;
 - private LocalDateTime ldtNextMatchDate;
 - private LocalDateTime ldtNextDeferralDate;
-

Propiedades sugeridas por la clase GameBean:

- privado PlayerBean objPlayerBean;
-

Propiedades sugeridas de la clase DeferralBean:

- privado PlayerBean objPlayerBean;
-

Propiedades sugeridas por la clase MesssageBean:

- privado PlayerBean objPlayerBean;
-

Propiedades sugeridas para la clase de torneo:

- privado PlayerBean objPlayerBean;
-

Propiedades sugeridas por la clase DBBean:

- private Conexión cnDB = null;
- private PreparedStatement pstaSQL = null;

D.3. Salida sugerida

Leyendo a un jugador al azar...

=> Jugador: 'Pepe Tableros' ...

Leyendo un torneo al azar...

=> Torneo: 'Torneo Maestro Yoda' ...

¡HOLA, HABLA EL FRIJOL
MENSAJERO!

Partida establecida como PENDIENTE y registrada a nombre del jugador 'Pepe Tableros'
en el torneo 'Torneo Maestro Yoda' en la fecha 2024-03-08 y hora 18:00

Mensaje insertado.

¡HOLA, GAMEBEAN AL
HABLA!

Juego insertado.

Jugador actualizado. Tiene una cita de partido.

¡HOLA, HABLA EL FRIJOL
MENSAJERO!

Aplazamiento establecido como SOLICITADO y registrado a nombre del jugador "Pepe
Tableros".

en el torneo 'Torneo Maestro Yoda' en la fecha 2024-03-08 y hora 18:00

Mensaje insertado.

¡HOLA, DEFERRALBEAN AL
HABLA!

Aplazamiento insertado.

Jugador actualizado. Tiene un aplazamiento solicitado.

¡HOLA, HABLA EL FRIJOL
MENSAJERO!

Partida de ajedrez completada
establecida como DRAWS

```
*****

Mensaje insertado.
-----
¡HOLA, GAMEBEAN AL
HABLA!
Juego actualizado (DRAWS).
Jugador actualizado. Ya no tiene cita con el partido.
-----
----- ¡HOLA, HABLA EL
FRIJOL MENSAJERO!

*****

Solicitud de aplazamiento establecida
como RECHAZADA

*****

Mensaje insertado.
-----
----- ¡HOLA, DEFERRALBEAN
AL HABLA!
Aplazamiento actualizado (RECHAZADO).
Jugador actualizado. Ya no tiene solicitud de aplazamiento.
```



Con licencia [Creative Commons Attribution Share Alike License 4.0](https://creativecommons.org/licenses/by-sa/4.0/)