

# PRESA. UNIDAD 1. ACCESO A ARCHIVOS. TAREA EVALUABLE 1

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

Unidad 1. ACCESO A LOS EXPEDIENTES

**Tarea evaluable 1**

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. Tenga en cuenta que ChatGPT no es infalible ni todopoderoso.

Es una gran herramienta para acelerar tu trabajo una vez que dominas una materia, pero utilizarla como atajo a la hora de adquirir habilidades y conocimientos básicos socava gravemente tu aprendizaje. Si lo utiliza para obtener soluciones o consejos por su cuenta, consulte también detenidamente 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

Recomendamos seguir los siguientes estándares de codificación:

- Una instrucción por línea.
  - Agregue comentarios para que su código sea más claro y legible.
  - Utilizar la notación húngara para reconocer el tipo de variables a primera vista.
  - Recuerda que hay varias formas de implementar una solución, así que elige la que más te guste.
- Recomendamos encarecidamente utilizar soluciones basadas en buffer.

# A. Instrucciones y directrices

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

## 1. INFORMACIÓN GENERAL

Debe crear una aplicación Java por su cuenta que utilice los conceptos aprendidos durante la UNIDAD 1 (Semana 1 a Semana 5) para cumplir con una especificación proporcionada.

## 2. CRONOGRAMA Y EXPECTATIVAS

- Porcentajes dentro del PLAZO: 50% del total del PLAZO (AT2 supondría el otro 50%)
- Porcentajes dentro de la TAREA: 100% habilidades ADA (las habilidades en inglés deben ser APROBADAS).
- Fecha límite: **11:59 p. m. del domingo 5 de noviembre de 2023** (3 SEMANAS)

## 3. CALIFICACIÓN

Debe obtener 5 puntos sobre 10 en ADA y APROBADO en inglés para aprobar este TAREA EVALUABLE.

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

## 4. RECURSOS

Debes 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.

Siéntase libre de copiar y pegar código de CUALQUIER recurso siempre que comprenda cada parte, ya que se le pedirá que defienda su trabajo en una reunión individual.

## 5. PLAGIO

No debe permitir que otros estudiantes copien su trabajo y debe tener cuidado para evitar que esto suceda.

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

## 6. ENTREGA Y COMENTARIOS

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

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

## B. Detalles de la evaluación

**SÓLO SE PERMITE INGLÉS** para la realización de la tarea evaluable, tanto comentarios como textos explicativos/aclaradores.

1. CADA MÉTODO DEBE SER DESCRITO ADECUADAMENTE CON SUS PROPIAS PALABRAS. En el

Al principio de cada método debes agregar 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 proyecto en el IDE de Eclipse.

- PARÁGRAFO 1. Describa brevemente la solución brindada.
- PARÁGRAFO 2. Describa brevemente las dificultades encontradas.
- PARÁGRAFO 3. Describa brevemente varias posibles extensiones que recomendó.

## B.1. Características obligatorias

### Actividad (EVALUABLE)

Crear un programa en Java para gestionar JUGADORES de AJEDREZ en un Torneo de Ajedrez imprimiendo y utilizando un menú específico. Después de cada opción, el usuario debería ver el mismo menú hasta que se presione la opción cero.

ATENCIÓN: Utilice las excepciones adecuadas al acceder a archivos.

Opciones de menú:

- Presione 0 para "Salir"
- Presione 1 para "Obtener jugadores de ajedrez y puntuaciones (en CSV y XML)"
  - Para cada JUGADOR DE AJEDREZ necesitamos ID de jugador (Cadena), nombre completo (Cadena con espacios), país (Cadena con espacios), puntuación/calificación del juego 1 (puntuación1, Flotante), puntuación/calificación del juego 2 (puntuación2, Flotante). ) y puntuación/calificación del juego 3 (puntuación3, Flotante), agregado a una ArrayList de JUGADORES DE AJEDREZ. Un jugador que gana su juego, o que gana por defecto, recibe un punto (1); un jugador que pierde su juego, o que pierde por defecto, recibe cero puntos (0); y un jugador que termina su juego en empate recibe medio punto ( $\frac{1}{2}$ ). Por lo tanto, sólo hay 3 valores de puntuación posibles: 0, 0,5 y 1 punto.
  - Compruebe si el ID del jugador de ajedrez ya existe en la lista de matrices. En caso afirmativo, deberá mostrar un mensaje en la pantalla. Debes solicitar cada valor (en bucle) hasta que el usuario ingrese una identificación válida.
  - Una vez que se ingresa cero como ID, todos los JUGADORES de AJEDREZ se guardarán en un archivo CSV y XML, con el formato adecuado, sobrescribiendo todos los archivos completos si existen. Antes de guardar, debes verificar si el ArrayList de JUGADORES DE AJEDREZ está vacío para evitar ejecutar código innecesario.
  - Mientras lees ArrayList para crear el XML, debes agregar un nodo llamado playerscoretotal y guardar la suma de las puntuaciones de cada JUGADOR DE AJEDREZ.
  - ATENCIÓN: ¡score1, Score2 y Score3 deben ser FLOTANTES! Por cada JUGADOR DE AJEDREZ, debes solicitar cada valor (en bucle) hasta que el usuario ingrese un valor flotante válido.

- Llame al archivo CSV: chessplayers.csv
- Llame al archivo XML: chessplayers.xml
- Presione 2 para "Listar todos los jugadores de ajedrez almacenados (usando DOM)"
  - Simplemente lea el archivo XML e imprima la información de cada JUGADOR DE AJEDREZ. Puedes utilizar DOM.
- Presione 3 para "Generar HTML con todos los jugadores de ajedrez vía XSL"
  - Para editar y adaptar la plantilla XSL que puedes encontrar al final de este documento o cualquier otro XSL que puedas encontrar en Internet, debes leer el archivo XML y generar un HTML usando el XSL. La tabla de encabezado debe tener un color de fondo.
  - Llame al archivo HTML: chessplayers.html
  - Llame al archivo XSL: chessplayers.xsl

Ejemplo de menú:

\*\*\*\*\*

MENÚ

\*\*\*\*\*

=====

0. Salir

1. Obtenga jugadores de ajedrez y puntuaciones (en CSV y XML)

2. Listar todos los jugadores de

ajedrez 3. Generar HTML con todos los jugadores de ajedrez a través

de XSL 4. [opcional] Modificar la calificación del juego de un jugador de ajedrez 5.

[opcional] Generar HTML con candidatos a maestros de ajedrez a través de XSL

=====

Seleccione una opción:

Aquí tienes una SUGERENCIA de cómo podría quedar tu CSV:

```
playerID, Nombre, País, Puntuación1, Puntuación2, Puntuación3
ELO2780, ANATOLY KARPOV, Rusia, 1, 1, 1
```



ELO2851, GARRY KASPAROV, Azerbaiyán, 1, 1, 1  
 ELO2882, MAGNUS CARLSEN, Noruega, 1, 1, 1  
 ELO2530, ARTURO POMAR, España, 1, 0,5, 0  
 ELO2785, BOBBY FISCHER, USA, 1, 1, 1  
 ELO2725, JOSÉ RAÚL CAPABLANCA, Cuba, 1, 1, 1  
 ELO2100, JOHN LOST, United Kingdom, 0.5, 0, 0

Aquí tienes una SUGERENCIA de cómo podría verse tu XML:

```
<?xml versión="1.0" codificación="UTF-8" standalone="no"?>
<Jugadores>
  <Jugador>
    <playerid>ELO2780</skaterid>
    <playername>ANATOLY KARPOV</playername>
    <playercountry>Rusia</playercountry>
    <playerscore1>1</playerscore1>
    <playerscore2>1</playerscore2>
    <playerscore3>1</playerscore3> <
    jugadorespuntuacióntotal>3</jugadorespuntuacióntotal>
  </jugador>
  <Jugador>
    <playerid>ELO2851</skaterid>
    <playername>GARRY KASPAROV</playername>
    <playercountry>Azerbaiyán</playercountry>
    <playerscore1>1</playerscore1>
    <playerscore2>1</playerscore2>
    <playerscore3>1</playerscore3> <
    jugadorespuntuacióntotal>3</jugadorespuntuacióntotal>
  </jugador>
  <Jugador>
    <playerid>ELO2530</skaterid>
    <playername>ARTURO POMAR</playername>
    <playercountry>España</playercountry>
    <playerscore1>1</playerscore1>
```

```
<playerscore2>0.5</playerscore2>
```

```
<playerscore3>0</playerscore3>
```

```
<playerscoretotal>1.5</playerscoretotal>
```

```
</jugador>
```

```
</Jugadores>
```

## B.2. Características opcionales

### Actividad (EVALUABLE)

Opcionalmente, puedes implementar las siguientes entradas dentro del menú para alcanzar más de 8 puntos sobre 10 en esta TAREA EVALUABLE.

ATENCIÓN: Utilice las excepciones adecuadas al acceder a archivos.

Opciones de menú:

- Presione 4 para “[opcional] Modificar la calificación del juego de un jugador de ajedrez”
  - El Comité de Árbitros podrá revisar un partido por iniciativa propia o a petición de un jugador del torneo. Si el resultado del juego se ve afectado, la puntuación se modificará según la decisión tomada.
  - Después de solicitar el ID del jugador, el número de partida (1, 2 o 3) y la nueva puntuación, recorreremos el DOM y cambiamos la puntuación de la partida correspondiente para el jugador de ajedrez cuyo ID coincida con el ID introducido anteriormente. Además, actualizaremos el total puntaje.
  - Luego (sobre)escribimos el XML nuevamente.
- Presione 5 para “[opcional] Generar HTML con candidatos a maestros de ajedrez a través de XSL”
  - Un jugador de ajedrez será considerado candidato a maestro si ha alcanzado 3 puntos en los 3 partidos del torneo.
  - Leer el CSV y crear otro XML con los candidatos a maestro de ajedrez. Llama a Archivo XML: candidatos.xml
  - Cree otro XSL a partir del existente, cambiando el color de primer plano de la fuente a rojo al enumerar los jugadores de ajedrez.
  - Luego, aplique este nuevo XSL para obtener otro HTML:
    - Llame al archivo HTML: candidatos.html
    - Llame al archivo XSL: candidatos.xsl

## Patrón XSL sugerido

```
<?xml versión="1.0"?>
<xsl: hoja de estilo xmlns: xsl="http://www.w3.org/1999/XSL/Transform"
versión="1.0">
  <xsl: coincidencia de plantilla="/">
    <html>
      <cabeza>
        < tipo de estilo="texto/css">
          tabla.tfomt { borde:
            1px ; }

          td.colfmt
          { borde: 1px ;
            color de fondo: blanco; color:
            negro;
            alineación de texto:centro; }

          th
          { color de fondo: #2E9AFE; color:
            blanco; } </estilo>
        </cabeza>
        <cuerpo>
          <tabla clase="tfomt">
            <tr>
              <th style="width:100px"> ID del jugador:</th> <th
                style="width:350px"> Nombre completo :</th> <th
                style="width:200px">País:</th> <th
                style="width:100px">Puntuación del juego 1:</th>
```

```

<th style="width:100px">Puntuación del juego 2:</th> <th
style="width:100px">Puntuación del juego 3:</th> <th
style="width:100px">Puntuación total:< /th> </tr>

<xsl: para cada uno seleccione="Patinadores/Patinador">
<tr>

    <td clase="colfmt">
        <xsl: valor-de select="playerid" />
    </td>
    <td clase="colfmt">
        <xsl: valor-de select="nombre del jugador" />
    </td>
    <td clase="colfmt">
        <
            valor de select="playercountry" /> xsl:
    </td>
    <td clase="colfmt">
        <xsl: valor de select="playerscore1" />
    </td>
    <td clase="colfmt">
        <xsl: valor-de select="playerscore2" />
    </td>
    <td clase="colfmt">
        <
            valor de select="playerscore3" /> xsl:
    </td>
    <td clase="colfmt">
        <xsl: valor de select="playerscoretotal" />
    </td>
</tr>
</ xsl: para-cada> </

table>

</cuerpo>

</html>

</ xsl: plantilla>

</ hoja de estilo> xsl:

```

## C. Rúbrica de aprendizaje

### C.1. HAY habilidades

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

Estas notas serán invalidadas (nota 4) si no defiendes tu trabajo en una audiencia oral. entrevista.

ARTÍCULOS DE EVALUACIÓN	DETALLES DEL ARTÍCULO DE EVALUACIÓN	PUNTAJE)
Clases y métodos	Las clases y los métodos están estructurados. adecuadamente	0,75
Menú	El menú cumple con las especificaciones	0,75
Consigue jugadores de ajedrez y marcas (para CSV y XML)	La ArrayList está completa correctamente Se generan el CSV y el XML. adecuadamente	2.5
Lista de jugadores de ajedrez	Lee el XML correctamente (usando DOM) Imprime los datos de forma adecuada.	1.5
Genera HTML con todo el ajedrez reproductores a través de XSL	Lee la información del XML. Genera el HTML correctamente desde un XLS	2.5
[opcional] Modificar la calificación del juego de un jugador de ajedrez		1
[opcional] Genere HTML con candidatos a maestros de ajedrez a través de XSL		1

## C.2. conocimientos de inglés

Obligatorio ser COMPETENTE para aprobar esta parte.

EVALUACIÓN ELEMENTOS	DETALLES DEL ARTÍCULO DE EVALUACIÓN	PUNTAJE
Habilidades de escritura	Cada método se describe adecuadamente COMPETENTE/NO COMPETENTE	
	Se proporciona un texto adecuado (dentro del código o en un archivo de texto) para describir el EN utilizando TRES PÁRRAFOS	COMPETENTE/NO COMPETENTE
Habilidades orales	Utiliza un vocabulario apropiado para el propósito.	COMPETENTE/NO COMPETENTE
	Muestra fluidez y confianza.	COMPETENTE/NO COMPETENTE
Habilidades de comprensión		Cumplido ya que todos los materiales están en Inglés
Habilidades de lectura		Cumplido ya que todos los materiales están en Inglés



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