

Evaluación Parcial N°3

Sigla	Nombre Asignatura	Tiempo Asignado	% Ponderación
DSY1102	Desarrollo Orientado a Objetos	120 minutos	30%

1. Situación evaluativa

<input checked="" type="checkbox"/>	Ejecución	<input type="checkbox"/>	Entrega de encargo	<input type="checkbox"/>	Presentación
-------------------------------------	-----------	--------------------------	--------------------	--------------------------	--------------

2. Agente evaluativo

<input checked="" type="checkbox"/>	Heteroevaluación	<input type="checkbox"/>	Coevaluación	<input type="checkbox"/>	Autoevaluación
-------------------------------------	------------------	--------------------------	--------------	--------------------------	----------------

3. Tabla de Especificaciones

Resultado de Aprendizaje	Indicador de Logro (IL)	Ponderación Indicador Logro
RA3 Organiza proyectos de software, utilizando estructura y/o dependencia de paquetes, para dar orden lógico y cohesión a los componentes desarrollados.	IL 3.1 Utiliza los contenedores y componentes para visualizar la interfaz gráfica dando solución a requerimientos solicitados por el usuario en un contexto de negocios.	30%
	IL 3.2 Incorpora principios del patrón de arquitectura Modelo-Vista-Controlador (MVC) a una estructura de directorios coherente en el proyecto de software para una organización eficaz de la información.	20%
	IL3.3 Programa sentencias para la ejecución de consultas con base en los datos de un caso.	30%
	IL3.4 Agrupa componentes en módulos con funcionalidad o propósito común para la facilitación de la navegación y el mantenimiento del código.	10%
	IL3.5 Valida la estructura del proyecto basándose en los estándares de la industria y en las buenas prácticas de desarrollo de softwares.	10%
Total		100%

4. Instrucciones generales para el/la estudiante

Esta es una evaluación que corresponde a una ejecución práctica y tiene un **30%** de ponderación sobre la nota de presentación a la Evaluación final Transversal. El tiempo para desarrollar esta evaluación es de **120 minutos** y se realiza de manera **individual** en laboratorio PC avanzado.

En esta evaluación deberá usar librerías para crear una **interfaz gráfica** que permita al usuario interactuar con el sistema y para conectarse a una **base de datos** MySql utilizando el driver correspondiente. El desarrollo de todo el software deberá regirse por la arquitectura **MVC**.

Para dejar registro deberá realizar los siguientes puntos:

- Programar en el lenguaje JAVA utilizando el IDE NetBeans.
- Deberá dejar registro del trabajo realizado por medio de la plataforma de Blackboard (AVA), para ello debe **comprimir la carpeta de su proyecto renombrando el archivo con el formato “NombreApellido”** y subirla a la actividad correspondiente.

5. Evaluación

Una famosa empresa del entretenimiento creó un juego de cartas coleccionables llamado “ToteMon LCG”, el cuál desafía a cada jugador a ser el mejor chamán de la aldea. Esta aldea es famosa por crear unos poderosos Tótems de protección, de alimentación, de hidratación y sanación. Los cuales están asociados a los elementos **Tierra, Fuego, Agua y Viento**, respectivamente.

Para poder ser el mejor Chamán, cada jugador deberá elegir una combinación de Tótems en un mazo de 35 cartas con el cual podrá enfrentarse a los Chamanes rivales en una lucha intensa y desgastante.

Esta empresa, le pide realizar un Software informático para que los jugadores puedan crear sus mazos digitalmente según lo siguiente:

Interfaz Gráfica:

1. La interfaz gráfica debe contener los siguientes elementos:

The interface is divided into several functional areas:

- Registrar nueva carta**: Includes input fields for 'Nombre', a dropdown for 'Elemento' (currently showing 'Seleccionar ...'), and input fields for 'Ataque' and 'Defensa'. A 'Registrar carta' button is located below these fields.
- Registro de usuario**: Includes input fields for 'Nombre' and 'Correo'. An 'Agregar Usuario' button is located below these fields.
- Lista de cartas**: A large rectangular area for displaying a list of cards, with a header 'ID - Nombre - Elemento - Ataque - Defensa'.
- Nombre del mazo**: An input field for the deck name.
- ID Usuario** and **ID Carta**: Input fields for selecting a user and a card.
- Agregar cartas al mazo**: A button to add selected cards to the deck.
- Mostrar mazo**: A button to display the current deck.
- ID Usuario**: An input field for selecting a user to view their deck.
- Carta agregada exitosamente!**: A green text message indicating a successful action.
- ID - Nombre - Correo**: A list area for displaying user information.
- ID Mazo - Nombre mazo:**: A list area for displaying the contents of a selected deck, showing the ID, name, and attack/defense values of each card.

Registrar nueva carta

Nombre:

Elemento:

Seleccionar ...

Ataque:

Seleccionar ...

Tierra

Fuego

Aire

Agua

Nombre:

Correo:

Agregar Usuario

ID - Nombre - Correo

Lista de cartas

ID - Nombre - Elemento - Ataque - Defensa

Nombre del mazo:

ID Usuario:

ID Carta:

Agregar cartas al mazo

Carta agregada exitosamente!

Mostrar mazo

ID Usuario

ID Mazo - Nombre mazo:

1 - ID Carta - Nombre Carta

2 - ID Carta - Nombre Carta

3 - ID Carta - Nombre Carta

Requerimientos:

1. Se debe poder registrar una nueva carta con los campos: nombre, elemento (Fuego, Tierra, Agua o Aire), la cantidad de ataque (Integer) y la cantidad de defensa (Integer).
2. Cada carta registrada debe aparecer en una área de texto en una lista de todas las cartas registradas.
3. Se debe poder registrar a un nuevo usuario con los campos: nombre y correo.
4. Cada usuario registrado debe aparecer en un área de texto con su información.
5. Se debe poder asociar un usuario a una carta utilizando el ID de cada uno. **Cada usuario puede tener solo un mazo asociado.**
6. Se debe poder buscar un mazo asociado al ID de un usuario y se debe ver la información del mazo en una área de texto con toda la información del mazo.
7. Cuando se registra una carta al mazo, debe aparecer un mensaje en color verde “Carta agregada exitosamente”, de lo contrario un mensaje en color rojo “No se pudo agregar la carta”.
8. Todas las áreas de texto deben listar los datos como muestra la figura del ejemplo de GUI a implementar.

Validaciones, experiencia de usuario y restricciones:

1. Se debe verificar que el nombre del usuario tenga entre 2 y 15 caracteres como límites.
2. Cada carta puede aparecer hasta 3 veces en un mismo mazo. Por lo tanto no se puede agregar una cuarta carta con el mismo ID a un mazo. Para esto debe crear un método en el controlador para contar las cartas que hay en un mazo.
3. Cada vez que se registre una nueva carta, se debe dejar limpios los campos campo de texto para el nombre, el ataque y la defensa; y se tiene que dejar el comboBox del elemento en su posición inicial.
4. Cada vez que un nuevo usuario se debe limpiar el texto en el campo de texto para el nombre y para el correo .

Base de datos:

1. Se debe crear un esquema MySQL con el software MySQL Workbench en el servidor localhost:3306.
2. Se debe crear una tabla **cartas** para guardar la siguiente información:
 - a. idcarta → primary key, not null, número entero auto incremental
 - b. nombre → cadena de texto de 20 caracteres.
 - c. elemento → cadena de texto de 10 caracteres.
 - d. ataque → número entero
 - e. defensa → número entero
3. Se debe crear una tabla **usuarios** para guardar la siguiente información:
 - a. idusuario → primary key, not null, número entero auto incremental
 - b. nombre → cadena de texto de 20 caracteres.
 - c. correo → cadena de texto de 20 caracteres.
4. Se debe crear una tabla **mazos** para guardar la siguiente información:
 - a. idmazo → primary key, not null, número entero auto incremental
 - b. nombre → cadena de texto de 20 caracteres.
 - c. id_usuario → foreign key
 - d. id_carta → foreign key

Controller:

1. Se debe crear un controlador que utilice la Clase JdbcController para crear el registro de una nueva carta, un nuevo usuario y un nuevo mazo con las cartas correspondientes.
2. Se deben crear métodos que lean la información de la Base de Datos y que será usada para listar los datos en las áreas de texto.

Manejadores:

1. Se debe programar el método manejador del evento MouseClicked de cada botón.
 - a. Registrar carta:
 - i. extrae los datos de cada input y se envían al método del controlador que registra una nueva carta.
 - ii. Se registra la nueva carta y se muestran todas las cartas registradas en el área de texto correspondiente.
 - b. Registro de usuario:
 - i. Se extraen los datos de cada input y se envían al método del controlador que registra un nuevo usuario.

- ii. Se registra el usuario y se hace aparecer la lista completa de usuarios registrados en el área de texto correspondiente.
- c. Agregar cartas al mazo:
 - i. Cada usuario tiene solo un mazo, por lo tanto se extraen los datos de cada campo de texto y se envían al método controlador para agregar cartas al mazo.
 - ii. Se muestra un mensaje de éxito de color verde o de error de color rojo
- d. Mostrar mazo:
 - i. Se extrae el ID del usuario del campo de texto y se envía al método controlador para traer toda la información del mazo con ese usuario.
 - ii. Se muestra el resultado en el área de texto correspondiente.