

# Taller de Programación

## Guía 4

Estudiosos:

Juan David Rincon Muñoz

Julian Eduardo Lozano Rios

Universidad Manuela Beltran

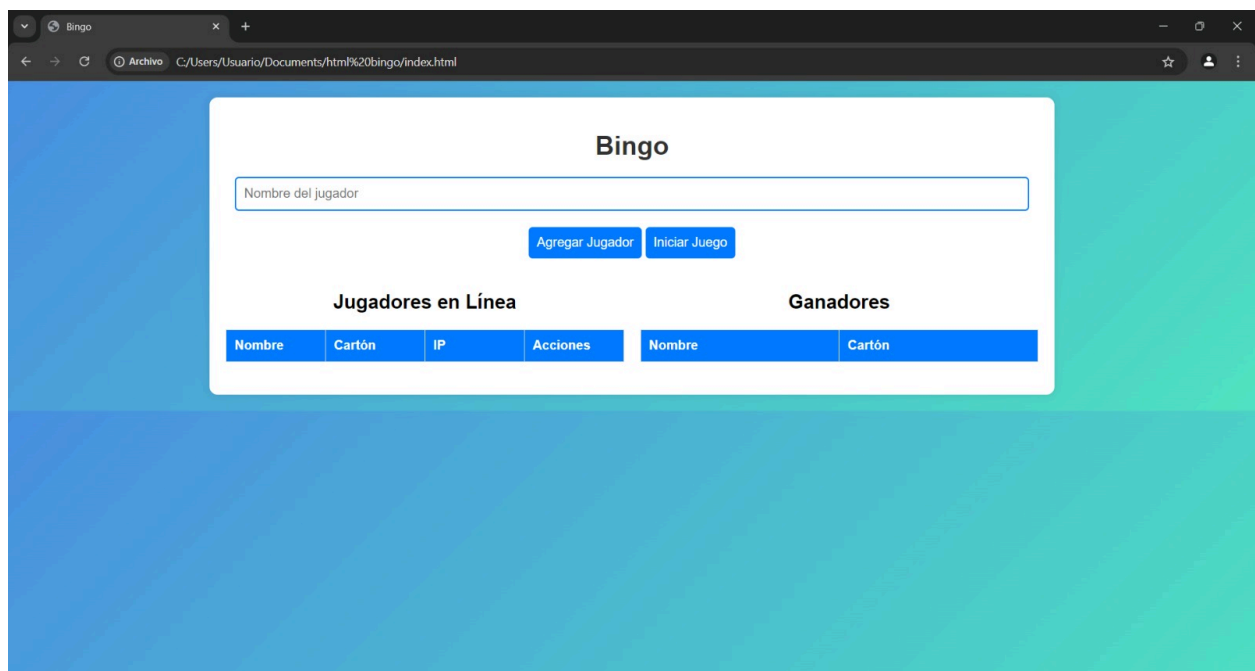
Ingeniería de software

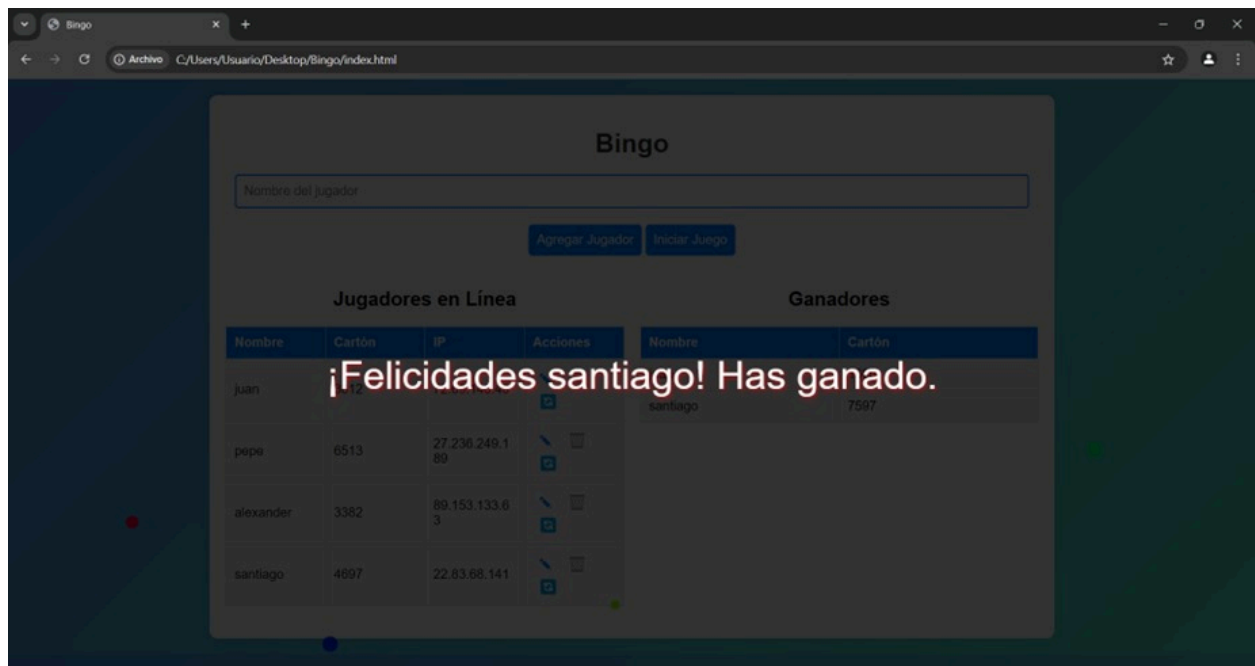
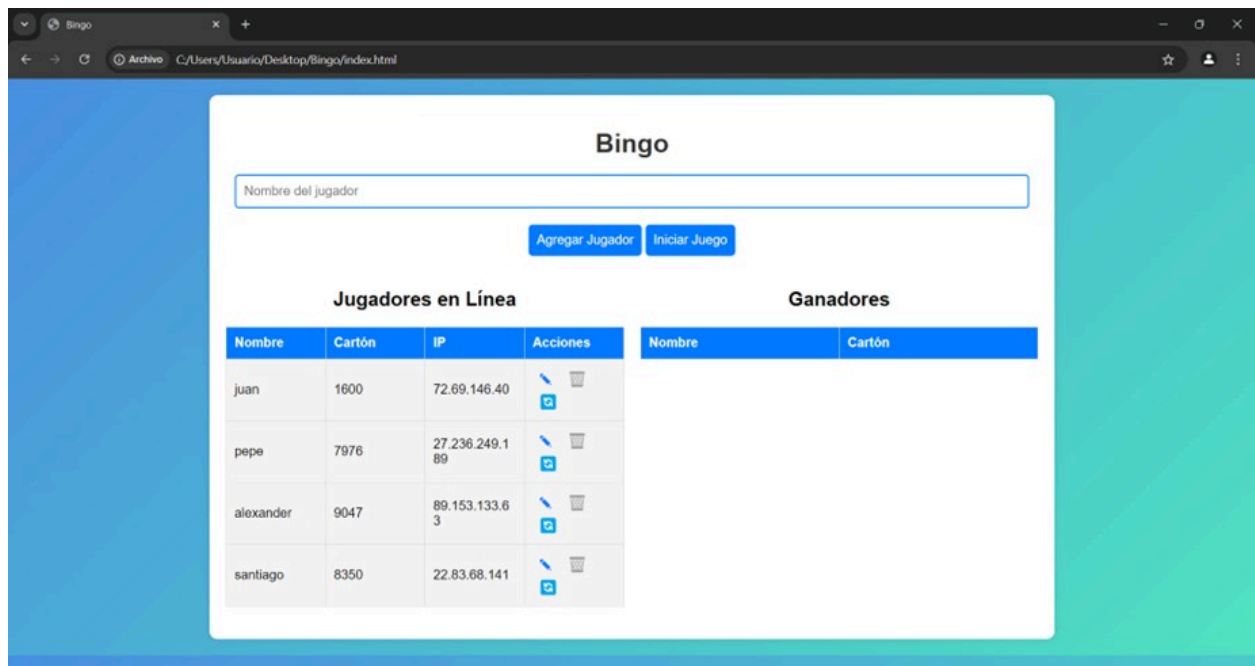
Olga Lucia Roa

14 de septiembre del 2024

## Sesión 2.

1. Diseñe e implemente una interfaz gráfica de usuario utilizando la librería Swing de Java o tecnología web (html, css y javascript o librerías de js) para realizar el Front-End del CRUD de la tabla. Recuerde validar campos de entrada.
2. Anexar los códigos fuente del ejercicio de implementar los patrones MVC y DAO para conexión a bases de datos y realizar el CRUD de una tabla Usuario.
3. Solucionar los puntos del cuestionario.
4. Diligencie todos los apartes de la guía de laboratorio y sustente en laboratorio el código realizado.





## **Actividad de Trabajo Autónomo**

### **1. Diferencia entre ODBC y JDBC**

ODBC (Open Database Connectivity) y JDBC (Java Database Connectivity) son interfaces que permiten la conexión entre aplicaciones y bases de datos, pero difieren en varios aspectos.

#### **ODBC:**

- Es una API independiente del lenguaje, lo que significa que puede ser utilizada con varios lenguajes de programación como C, C++, Python, etc.
- Requiere un controlador ODBC para cada base de datos con la que se desea trabajar.
- Generalmente se utiliza en plataformas Windows.

#### **JDBC:**

- Es una API específica para Java que permite a las aplicaciones Java interactuar con bases de datos.
- Depende de un driver JDBC específico para cada base de datos.
- Es una solución nativa para aplicaciones Java.

En conclusión; JDBC es específico para Java, mientras que ODBC es independiente del lenguaje de programación, aunque ambos permiten la interacción con bases de datos.

## **2. Diferencias y Similitudes entre base de datos relacional y bases de datos NoSQL**

### **Bases de Datos Relacionales (SQL):**

- Se basan en un modelo estructurado de tablas.
- Utilizan un lenguaje estándar: SQL.
- Garantizan propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
- Adecuadas para transacciones complejas y relaciones entre datos.

### **Bases de Datos NoSQL:**

- Utilizan modelos de datos flexibles como documentos, gráficos, clave-valor, o columnas.
- No necesariamente siguen el esquema rígido de las tablas.
- No garantizan propiedades ACID en todos los casos, pero son más escalables horizontalmente.
- Se utilizan en aplicaciones que requieren gran rendimiento, escalabilidad, y manejo de grandes volúmenes de datos no estructurados.

### **Similitudes:**

- Ambas almacenan y gestionan datos.
- Pueden ser distribuidas y escaladas.
- Soportan consultas eficientes, aunque con lenguajes y enfoques diferentes.

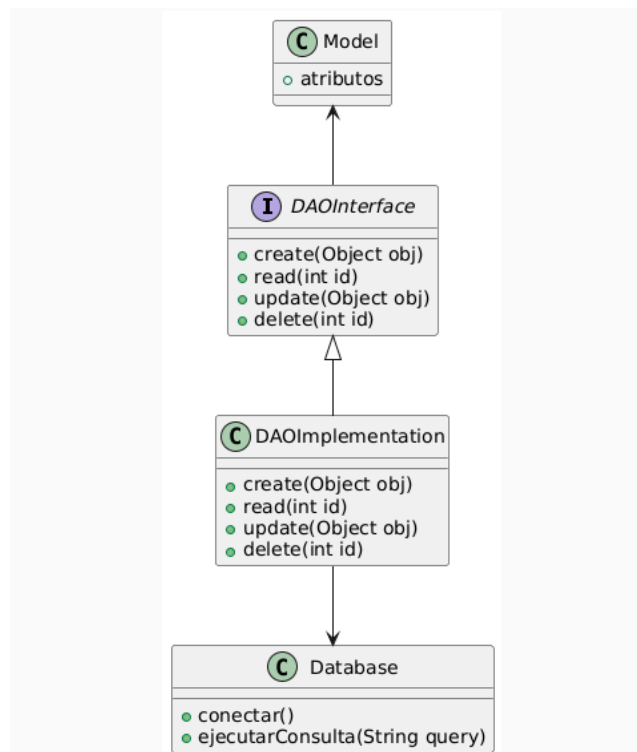
### 3. Definir Patrón Data Access Object (DAO), explique mediante un diagrama de UML

El patrón DAO es un patrón de diseño que abstrae la lógica de acceso a la base de datos del resto de la aplicación. Su propósito es centralizar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en una clase especializada para interactuar con la base de datos, facilitando el mantenimiento y el cambio de tecnologías de persistencia.

#### Diagrama UML de DAO:

##### Clases principales:

- DAOInterface: define los métodos de acceso a los datos (CRUD).
- DAOImplementation: implementa los métodos definidos en la interfaz.
- Model: representa el objeto de negocio que será persistido.
- Database: representa la fuente de datos.



#### **4. Definir patrones Modelo Vista Controlador (MVC) y Data Transfer Object (DTO)**

##### **Modelo Vista Controlador (MVC):**

- Modelo: Gestiona los datos y la lógica de la aplicación.
  - Vista: Representa la interfaz de usuario y muestra los datos del modelo.
  - Controlador: Gestiona la comunicación entre la Vista y el Modelo, manejando las interacciones del usuario y actualizando los datos en el Modelo.
- Beneficio: Separa las responsabilidades, facilitando el mantenimiento y la escalabilidad.

##### **Data Transfer Object (DTO):**

- Es un patrón utilizado para transportar datos entre diferentes capas de una aplicación, minimizando el número de llamadas entre el cliente y el servidor.
- Los DTOs son simples objetos que contienen datos y no implementan lógica de negocio.

#### **5. Defina que es JSON y para que se utiliza.**

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos, basado en texto y fácil de leer y escribir tanto para humanos como para máquinas. Se utiliza principalmente para transmitir datos entre un servidor y una aplicación web, pero su uso se ha extendido a muchas otras áreas debido a su simplicidad y compatibilidad.

##### **Características:**

- Estructurado en pares clave-valor.

-Es independiente del lenguaje, lo que lo hace ideal para la comunicación entre diferentes tecnologías.

-Muy utilizado en APIs y servicios web para el envío de respuestas y solicitudes.

**Usos principales:**

-Transmisión de datos entre cliente y servidor.

-Almacenamiento de configuraciones o datos de manera estructurada.

**Actividad de Comprobación del Trabajo Autónomo**

El estudiante realiza un video EN INGLÉS de mínimo 1 y máximo 3 minutos exponiendo el marco teórico y el código desarrollado en el ejemplo práctico propuesto en el procedimiento a utilizar.

Enlace: <https://youtu.be/FV6NARVGwRY?si=mQeTLGO7LRH7lmo4>

Enlace: <https://github.com/juandatiner/Guias-Taller-de-Programacion>