

Actividad 10-11

Para el inicio del código cree los paquetes logica_juego, modelo, Juego.

Para el paquete lógica_juego cree un archivo.java llamado “lógica”, para el paquete modelo uno llamado “Pais” y para el paquete juego un archivo llamado “juego”.

El archivo Pais es una clase que contiene los atributos del país en mi caso los atributos eran nombre, pista y código.

```
package modelo;

/**
 *
 * @author CI5
 */
import java.io.Serializable;

public class Pais implements Serializable {
    private String nombre;
    private String pista;
    private String codigo;

    public Pais(String nombre, String pista) {
        this.nombre = nombre;
        this.pista = pista;
        this.codigo = pista; // Asignar el código al campo pista
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getPista() {
        return pista;
    }

    public void setPista(String pista) {
        this.pista = pista;
    }

    public String getCodigo() {
        return codigo;
    }

    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }
}
```

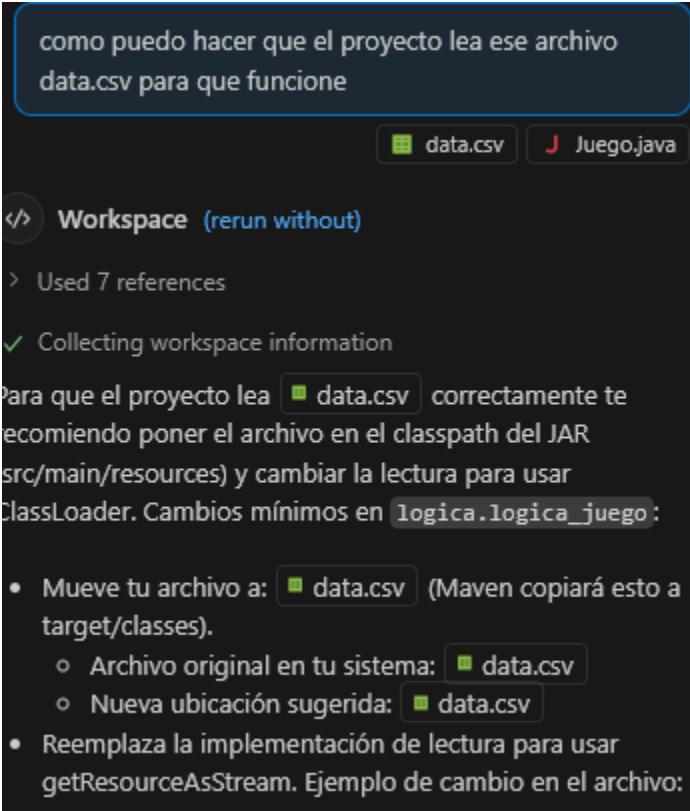
también cree un método para obtener una pista llamado `getPrimeraLetra()`, y otro que muestra el código del país llamado `mostrarCodigo()`.

```
// Método para obtener la primera letra del nombre
public char getPrimeraLetra() {
    if (nombre != null && !nombre.isEmpty()) {
        // Convierte el nombre a mayúsculas para la pista
        return nombre.toUpperCase().charAt(index:0);
    }
    return ' ';
}

public void mostrarCodigo() {
    System.out.println("Código del país: " + (codigo != null ? codigo : ""));
}
```

Para la clase lógica_juego cree toda la lógica de como iba a funcionar el juego importando varias librerías.

Lo primero que hice fue definir el arraylist y empezar a definir el método para cargar los países desde el csv. Para ello le pedí al chat copilot como podía colocar el archivo data.csv para que el programa lo leyera y me respondió:



como puedo hacer que el proyecto lea ese archivo data.csv para que funcione

data.csv Juego.java

Workspace (rerun without)

Used 7 references

Collecting workspace information

Para que el proyecto lea data.csv correctamente te recomiendo poner el archivo en el classpath del JAR (src/main/resources) y cambiar la lectura para usar ClassLoader. Cambios mínimos en logica.logica_juego:

- Mueve tu archivo a: data.csv (Maven copiará esto a target/classes).
 - Archivo original en tu sistema: data.csv
 - Nueva ubicación sugerida: data.csv
- Reemplaza la implementación de lectura para usar getResourceAsStream. Ejemplo de cambio en el archivo:

Con la sugerencia de código:

```

public class logica_juego {
    private ArrayList<Pais> paises;
    // ...existing code...

    // 🏆 Método para cargar datos del CSV desde classpath 🏆
    private void cargarPaisesDesdeCSV() {
        try (InputStream is = getClass().getClassLoader().getResourceAsStream("data.csv")) {
            if (is == null) {
                System.err.println("❌ No se encontró 'data.csv' en el classpath. Colóquelo en src/main/resources/");
                return;
            }
            try (BufferedReader br = new BufferedReader(new InputStreamReader(is, StandardCharsets.UTF_8))) {
                String linea;
                System.out.println("Cargando países desde classpath/data.csv...");
                while ((linea = br.readLine()) != null) {
                    String[] datos = linea.split(",");
                    if (datos.length >= 2) {
                        String nombre = datos[0].trim();
                        String pista = datos[1].trim();
                        if (!nombre.isEmpty()) {
                            paises.add(new Pais(nombre, pista));
                        }
                    }
                }
                System.out.println("✅ Países cargados: " + paises.size());
            }
        } catch (IOException e) {
            System.err.println("❌ Error al leer el archivo CSV: " + e.getMessage());
        }
    }
}

```

Al implementar el código y seguir las indicaciones no se encontraba la carpeta que recomendó así que le solicite que lo hiciera en una ruta que definiría con lo cual me sugirió:

```

private void cargarPaisesDesdeCSV() {

    File f = new File(CSV_FILE_PATH);

    if (f.exists() && f.isFile()) {

        try (BufferedReader br = new BufferedReader(new FileReader(f))) {

            System.out.println("Cargando países desde la ruta definida: " + CSV_FILE_PATH);

            String linea;

            // Saltar la línea de encabezado ("Name,Code")

            br.readLine();

            while ((linea = br.readLine()) != null) {

                // Lógica para manejar el formato CSV con campos entre comillas

```

```

String[] datos;

if (linea.startsWith("\\")) {
    int finCita = linea.indexOf("\\,");

    if (finCita != -1) {
        String nombre = linea.substring(1, finCita);
        String codigoConComa = linea.substring(finCita + 2);
        datos = new String[]{nombre, codigoConComa};
    } else {
        datos = linea.split(","); // Fallback si el formato es inusual
    }
} else {
    datos = linea.split(",");
}

// Asume el formato: Nombre, Código
if (datos.length >= 2) {
    String nombre = datos[0].trim();
    String pista = datos[1].trim();

    if (!nombre.isEmpty()) {
        paises.add(new Pais(nombre, pista));
    }
}

System.out.println("Países cargados: " + paises.size());
} catch (IOException ex) {
    System.err.println(" Error leyendo el archivo en la ruta definida: " +
ex.getMessage());
}

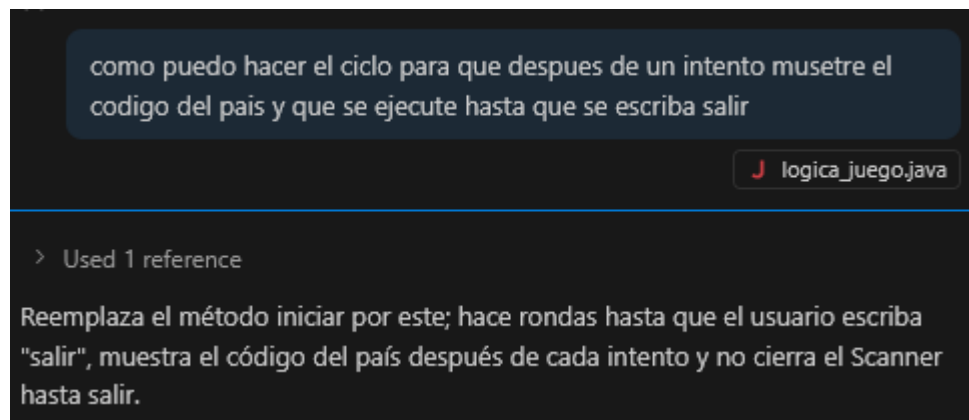
```

```

        System.err.println("Asegúrese de que el archivo no esté abierto y que la ruta sea correcta.");
    }
} else {
    System.err.println(" ERROR: No se encontró el archivo en la ruta ABSOLUTA definida:");
    System.err.println(CSV_FILE_PATH);
    System.err.println("Verifique que la ruta sea correcta y que el archivo se llame 'data.csv'.");
}
}
}

```

Al implementarlo implemente una lógica para que se inicie el juego, al intentarlo le solicite:



Con lo cual me sugirió un código, pero ese código no cumplía del todo lo que quería lograr con el juego, siendo este el código que me sugirió:

```

public void iniciar() {
    if (países.isEmpty()) {
        System.out.println("No hay países para jugar. Saliendo...");
        return;
    }

    Scanner scanner = new Scanner(System.in);

    Random random = new Random();

```

```
System.out.println("\n--- ¡Adivina el País! ---");

System.out.println("Escribe el nombre del país y presiona Enter. Escribe 'salir' para
terminar.");

while (true) {

    // Elegir un país aleatorio para la ronda

    int indiceAleatorio = random.nextInt(paises.size());

    Pais paisElegido = paises.get(indiceAleatorio);


    // Mostrar pista

    System.out.println("\nPISTA: El país que elegí empieza con la letra: " +
paisElegido.getPrimeraLetra());

    System.out.print("Tu respuesta (o 'salir'): ");

    String adivinanza = scanner.nextLine().trim();

    // Comando para salir

    if (adivinanza.equalsIgnoreCase("salir")) {

        System.out.println("Saliendo del juego. ¡Hasta luego!");

        break;

    }

    // Evaluar la respuesta

    if (adivinanza.equalsIgnoreCase(paisElegido.getNombre())) {

        System.out.println("\n¡CORRECTO! Adivinaste el país: " +
paisElegido.getNombre() + ".");

    } else {

        System.out.println("\n¡INCORRECTO. El país que elegí era: " +
paisElegido.getNombre() + ".");

    }

    // Mostrar el código del país (usa el método en modelo.Pais)
```

```

paisElegido.mostrarCodigo();

// Opcional: preguntar si quiere seguir (puede presionar Enter para continuar)
System.out.println("\nPresiona Enter para otra ronda o escribe 'salir' para
terminar.");

String linea = scanner.nextLine().trim();
if (linea.equalsIgnoreCase("salir")) {
    System.out.println("Saliendo del juego. ¡Hasta luego!");
    break;
}
}
scanner.close();
}

```

Lo que hace ese código era que después de no haber contestado en el primer intento el país de una vez se terminaba esa ronda mostraba el código y preguntaba si quería salir, por lo cual implemente el código pero modifique el ciclo para que funcionara en el ciclo de muestra la primera pista -> si es correcto muestra mensaje de felicitación, si es falso -> pregunta ¿Quiere la segunda pista s/n? -> si se elige s para a la siguiente pista, de lo contrario termina la ejecución cuando se escribe salir después de la primera pista o pistas del equipo.

```

Juego.java 1 J Pais.java 1 J logica_juego.java 1
go > src > main > java > logica > J logica_juego.java > logica_juego > iniciar()

public class logica_juego {
    public void iniciar() {
        if (países.isEmpty()) {
            System.out.println(x:"No hay países para jugar. Saliendo...");
            return;
        }

        Scanner scanner = new Scanner(System.in);
        Random random = new Random();

        System.out.println(x:"\n--- ¡Adivina el País! ---");
        System.out.println(x:"Escribe el nombre del país y presiona Enter. Escribe 'salir' para terminar.");

        while (true) {
            int indiceAleatorio = random.nextInt(países.size());
            País paisElegido = países.get(indiceAleatorio);

            String respuestaEvaluada; // Variable para almacenar la última respuesta válida.

            // PRIMERA PREGUNTA Y PISTA
            System.out.println("\nPISTA 1: El país que elegí empieza con la letra: " + paisElegido.getPrimeraLetra());
            System.out.print(s:"Tu respuesta (o 'salir'): ");
            String adivinanzaInicial = scanner.nextLine().trim();

            // Manejar comando de salir antes de la lógica principal
            if (adivinanzaInicial.equalsIgnoreCase(anotherString:"salir")) {
                System.out.println(x:"Saliendo del juego. ¡Hasta luego!");
                break;
            }

            // Asumimos que la respuesta a evaluar es la inicial, a menos que pida pista.
            respuestaEvaluada = adivinanzaInicial;

            // SEGUNDA PISTA (CICLO)
            System.out.print(s:"¿Quieres una segunda pista? (s/n): ");
            String respuestaPista = scanner.nextLine().trim();

            // Verifica si quiere la segunda pista
            if (respuestaPista.equalsIgnoreCase(anotherString:"s")) {
                System.out.print(s:"SEGUNDA PISTA: El código del país es: ");
                paisElegido.mostrarCodigo(); // Usa el método de tu clase País

                System.out.print(s:"Tu nueva adivinanza (o código): ");
                String nuevaAdivinanza = scanner.nextLine().trim();

                if (respuestaPista.equalsIgnoreCase(anotherString:"s")) {
                    System.out.print(s:"PISTA 2: El código del país es: ");
                    paisElegido.mostrarCodigo(); // Usa el método de tu clase País

                    System.out.print(s:"Tu nueva adivinanza (o código): ");
                    String nuevaAdivinanza = scanner.nextLine().trim();
                    respuestaEvaluada = nuevaAdivinanza;
                }
            }

            // VERIFICAR RESPUESTA FINAL
            // Permite adivinar por Nombre (getNombre) o por Código (getCodigo)
            if (respuestaEvaluada.equalsIgnoreCase(paisElegido.getNombre()) ||
                respuestaEvaluada.equalsIgnoreCase(paisElegido.getCodigo())) {
                System.out.println("\n ¡CORRECTO! Adivinaste el país: " + paisElegido.getNombre() + " (Código: " + paisElegido.getCodi
            } else {
                System.out.println("\n INCORRECTO. El país que elegí era: " + paisElegido.getNombre() + " (Código: " + paisElegido.get
            }

            // CICLO DE CONTINUACIÓN
            System.out.println(x:"\nPresiona Enter para otra ronda o escribe 'salir' para terminar.");
            String linea = scanner.nextLine().trim();

            if (linea.equalsIgnoreCase(anotherString:"salir")) {
                System.out.println(x:"Saliendo del juego. ¡Hasta luego!");
                break;
            }
        }
    }
}

```


En el archivo juego tengo el método main para inicializar lo que es la lógica_juego y el método iniciar(); importando el paquete lógica.lógica_juego;

```
package juego;
import logica.logica_juego;

/**
 *
 * @author CI5
 */
public class Juego {

    Run | Debug
    public static void main(String[] args) {
        logica_juego logica_juego = new logica_juego();
        logica_juego.iniciar();
    }
}
```

Prueba del código:

```
Cargando países desde la ruta definida: C:\Users\CI5\Documents\IPC1_Actividades_202405106\Actividad 10-11\Juego\src\main\resources\data.csv
Países cargados: 249

--- ¡Adivina el País! ---
Escribe el nombre del país y presiona Enter. Escribe 'salir' para terminar.

PISTA 1: El país que elegí empieza con la letra: A
Tu respuesta (o 'salir'): Argentina
¿Quieres una segunda pista? (s/n): s
PISTA 2: El código del país es: Código del país: AO
Tu nueva adivinanza (o código): Angola

¡CORRECTO! Adivinaste el país: Angola (Código: AO).

Presiona Enter para otra ronda o escribe 'salir' para terminar.
█
```

```
Presiona Enter para otra ronda o escribe 'salir' para terminar.

PISTA 1: El país que elegí empieza con la letra: S
Tu respuesta (o 'salir'): Samalia
¿Quieres una segunda pista? (s/n): n

INCORRECTO. El país que elegí era: Solomon Islands (Código: SB).

Presiona Enter para otra ronda o escribe 'salir' para terminar.
█
```

Presiona Enter para otra ronda o escribe 'salir' para terminar.

PISTA 1: El país que elegí empieza con la letra: M

Tu respuesta (o 'salir'): Marruecos

¿Quieres una segunda pista? (s/n): n

INCORRECTO. El país que elegí era: Marshall Islands (Código: MH).

Presiona Enter para otra ronda o escribe 'salir' para terminar.

salir

Saliendo del juego. ¡Hasta luego!

PS C:\Users\CIF5\Documents\IPC1_Actividades_202405106\Actividad 10-11> █