

Actividad 10-11

Capturas de los mensajes enviados a la IA para consultas o solicitudes de generación de código:

Usando el archivo `#file:data.csv` crea un juego con esos datos de ese archivo, crea objetos de tipo pais y guardalos en una arraylist para nutrir de informacion el juego

El juego necesita ambas clases para funcionar - `J Pais.java` define el modelo de datos y `Juego.java` usa esa clase para implementar la lógica del juego.

👍 🔄 Claude Sonnet 3.5 • 0.9x

.....

quiero todo en una misma clase

`J Pais.java:5`

Restore Checkpoint

Para jugar, se debe elegir un país aleatorio de la lista de países y darle una pista al usuario como: "El país que elegí empieza con" + primeraLetra del nombre del país que el sistema eligió.

`J JuegoPaises.java`

.....

y donde puede ver el usuario el codigo de cada país, porque en consolo no se muestra

`J JuegoPaises.java`

Enable Claude Sonnet 3.5 for all clients

.....

pongamolo mas facil ya que al momento de darle solo la inicial del pais es muy probable que no lo adivine, entonces podemos poner como varias letras del pais como morocos le mostramos M r c s

`J JuegoPaises.java`

.....

al final solo se debre mostrar si el usuario adivino o no, no des puntos

Capturas de métodos importantes:

Método Main:

```
Run | Debug
public static void main(String[] args) throws Exception {
    String ruta = "c:\\Users\\aleja\\Downloads\\data.csv";
    List<JuegoPaíses> países = cargarPaíses(ruta);
    System.out.println("Países cargados: " + países.size());

    Scanner sc = new Scanner(System.in);
    System.out.println(x:"\n¿Deseas ver la lista de países y sus códigos antes de empezar? (s/n): ");
    String respuesta = sc.nextLine().trim().toLowerCase();

    if (respuesta.equals(anObject:"s")) {
        mostrarListaPaíses(países);
        System.out.println(x:"\nPresiona ENTER para comenzar el juego...");
        sc.nextLine();
    }

    juegoQuiz(países);
}
```

Método mostrarListaPaíses:

```
static void mostrarListaPaíses(List<JuegoPaíses> países) {
    System.out.println(x:"\n=== LISTA DE PAÍSES Y SUS CÓDIGOS ===");
    for (JuegoPaíses pais : países) {
        System.out.printf(format:"%-40s %s%n", pais.getNombre(), pais.getCodigo());
    }
    System.out.println(x:"=====");
}
```

Método cargarPaíses:

```
static List<JuegoPaíses> cargarPaíses(String ruta) throws IOException {
    List<JuegoPaíses> lista = new ArrayList<>();
    try (BufferedReader r = Files.newBufferedReader(Paths.get(ruta))) {
        String linea = r.readLine(); // omitir cabecera
        while ((linea = r.readLine()) != null) {
            String[] parts = linea.split(regex:"(?=(?:[^\"]*"|"[^"]*"|'[^']*')*$)");
            if (parts.length >= 2) {
                String nombre = parts[0].trim();
                String codigo = parts[1].trim();
                if (nombre.startsWith(prefix:"\"") && nombre.endsWith(suffix:"\"")) {
                    nombre = nombre.substring(beginIndex:1, nombre.length() - 1);
                }
                lista.add(new JuegoPaíses(nombre, codigo));
            }
        }
    }
    return lista;
}
```

Método juegoQuiz:

```
static void juegoQuiz(List<JuegoPaises> paises) {  
    Scanner sc = new Scanner(System.in);  
    Random rnd = new Random();  
  
    System.out.println(x:"Quiz: adivina el país a partir de las letras mostradas. Escribe 'salir' para terminar");  
  
    while (true) {  
        JuegoPaises p = paises.get(rnd.nextInt(paises.size()));  
        String nombrePais = p.getNombre();  
        StringBuilder pista = new StringBuilder();  
  
        // Mostrar primera letra, algunas letras del medio y última letra  
        for (int i = 0; i < nombrePais.length(); i++) {  
            char c = nombrePais.charAt(i);  
            if (i == 0 || i == nombrePais.length()-1 || i % 2 == 0) {  
                pista.append(Character.toUpperCase(c));  
            } else {  
                pista.append(str: "_");  
            }  
            pista.append(str: " "); // Espacio entre cada letra  
        }  
  
        System.out.println("\nPISTA: Las letras del país son: " + pista.toString());  
        System.out.print(s:"¿Qué país es? (escribe el código): ");  
  
        String entrada = sc.nextLine().trim();  
        if (entrada.equalsIgnoreCase(anotherString:"salir")) {  
            System.out.println(x:"¡Gracias por jugar!");  
            break;  
        }  
  
        if (entrada.equalsIgnoreCase(p.getCodigo())) {  
            System.out.println("¡Correcto! El país era: " + p.getNombre() + "\n");  
        } else {  
            System.out.println("Incorrecto. El país era: " + p.getNombre() + " (" + p.getCodigo() + ")\n");  
        }  
    }  
    sc.close();  
}
```

Capturas de partidas jugadas:

Primero pregunta si deseamos ver la lista de países, si colocamos “s” nos la mostrará de lo contrario si colocamos “n”, no mostrará, pero de igual manera iniciará el juego.

```
Países cargados: 249
```

```
¿Deseas ver la lista de países y sus códigos antes de empezar? (s/n):  
s
```

```
=== LISTA DE PAÍSES Y SUS CÓDIGOS ===
```

Afghanistan	AF
Åland Islands	AX
Albania	AL
Algeria	DZ
American Samoa	AS

```
Quiz: adivina el país a partir de las letras mostradas. Escribe 'salir' para terminar.
```

```
PISTA: Las letras del país son: D _ N _ A _ K  
¿Qué país es? (escribe el código): DK  
¡Correcto! El país era: Denmark
```

```
PISTA: Las letras del país son: N _ R _ O _ K _ I _ L _ N D  
¿Qué país es? (escribe el código): Mexico  
Incorrecto. El país era: Norfolk Island (NF)
```

```
PISTA: Las letras del país son: F _ E _ C _ _ U _ A _ A  
¿Qué país es? (escribe el código): FU  
Incorrecto. El país era: French Guiana (GF)
```

```
PISTA: Las letras del país son: P _ R _ G _ A Y  
¿Qué país es? (escribe el código): PY  
¡Correcto! El país era: Paraguay
```

```
PISTA: Las letras del país son: I _ A Q  
¿Qué país es? (escribe el código): I
```

```
PISTA: Las letras del país son: S _ U _ H _ S _ D _ N  
¿Qué país es? (escribe el código): salir  
¡Gracias por jugar!
```

Apartado de decisiones tomadas:

- Cuando le pedí realizar almacenar la información del archivo CSV, me mando código me realizó dos clases, fue error mío porque no le especifique más cosas así que le mencione que quería todo junto, y me mando código en el cual almacenaba el archivo además de cargarlos omitía los encabezados lo cual es importante y lo realizo sin que le mencionara eso
- En ese momento el código que me proporciono prácticamente solo los almacenaba, aun no los mostraba, pero se podía ver que ya realizaba acciones como darle pistas al usuario e indicarle al usuario si había acertado o no, y también la opción de salir.
- Luego le solicite que el mismo sistema debería de elegir aleatoriamente el país a preguntar y las características que debía de decirle, en ese momento solo le dije que mostrara solo la inicial del nombre del país y el usuario debía elegir el país y colocar el código
- El código funcionado, pero aún no mostraba el listado de los países así que me corrigió el código específicamente el método main.
- En ese punto el código estaba casi listo pero el problema era que existen varios países con las mismas iniciales y así nunca se podría adivinar el país por lo que le pedí componer el código para que la pista que diera fuera mostrar una letra si y una no de todo el país.
- Al mandarme las correcciones ya funcionaba correctamente, ya solo le pedí que quitara los puntos por adivinar porque eso lo creo solito, y que si adivinaba el usuario el país o no le mandara un mensaje indicándole.

CODIGO GENERADO

```
1
2 import java.nio.file.*;
3 import java.io.*;
4 import java.util.*;
5
6 public class JuegoPaises {
7     private String nombre;
8     private String codigo;
9
10    public JuegoPaises(String nombre, String codigo) {
11        this.nombre = nombre;
12        this.codigo = codigo;
13    }
14
15    public String getNombre() {
16        return nombre;
17    }
18
19    public String getCodigo() {
20        return codigo;
21    }
22
23    @Override
24    public String toString() {
25        return nombre + " (" + codigo + ")";
26    }
27
28    Run | Debug
29    public static void main(String[] args) throws Exception {
30        String ruta = "c:\\Users\\aleja\\Downloads\\data.csv";
31        List<JuegoPaises> paises = cargarPaises(ruta);
32        System.out.println("Paises cargados: " + paises.size());
33
34        Scanner sc = new Scanner(System.in);
35        System.out.println(x:"\n¿Deseas ver la lista de países y sus códigos antes de empezar? (s/n): ");
36        String respuesta = sc.nextLine().trim().toLowerCase();
```

```
36
37     if (respuesta.equals(anObject:"s")) {
38         mostrarListaPaises(paises);
39         System.out.println(x:"\nPresiona ENTER para comenzar el juego...");
40         sc.nextLine();
41     }
42
43     juegoQuiz(paises);
44 }
45
46 static void mostrarListaPaises(List<JuegoPaises> paises) {
47     System.out.println(x:"\n=== LISTA DE PAÍSES Y SUS CÓDIGOS ===");
48     for (JuegoPaises pais : paises) {
49         System.out.printf(format:"%-40s %s\n", pais.getNombre(), pais.getCodigo());
50     }
51     System.out.println(x:"=====");
52 }
53
54 static List<JuegoPaises> cargarPaises(String ruta) throws IOException {
55     List<JuegoPaises> lista = new ArrayList<>();
56     try (BufferedReader r = Files.newBufferedReader(Paths.get(ruta))) {
57         String linea = r.readLine(); // omitir cabecera
58         while ((linea = r.readLine()) != null) {
59             String[] parts = linea.split(regex:"(?:[^\"]*"|"[^"]*"|'['']*')*$");
60             if (parts.length >= 2) {
61                 String nombre = parts[0].trim();
62                 String codigo = parts[1].trim();
63                 if (nombre.startsWith(prefix:"\"") && nombre.endsWith(suffix:"\"")) {
64                     nombre = nombre.substring(beginIndex:1, nombre.length() - 1);
65                 }
66                 lista.add(new JuegoPaises(nombre, codigo));
67             }
68         }
69     }
```

```

69     }
70     return lista;
71 }
72
73 static void juegoQuiz(List<JuegoPaises> paises) {
74     Scanner sc = new Scanner(System.in);
75     Random rnd = new Random();
76
77     System.out.println(x:"Quiz: adivina el país a partir de las letras mostradas. Escribe 'salir' para terminar.");
78
79     while (true) {
80         JuegoPaises p = paises.get(rnd.nextInt(paises.size()));
81         String nombrePais = p.getNombre();
82         StringBuilder pista = new StringBuilder();
83
84         // Mostrar primera letra, algunas letras del medio y última letra
85         for (int i = 0; i < nombrePais.length(); i++) {
86             char c = nombrePais.charAt(i);
87             if (i == 0 || i == nombrePais.length()-1 || i % 2 == 0) {
88                 pista.append(Character.toUpperCase(c));
89             } else {
90                 pista.append(str:"_");
91             }
92             pista.append(str:" "); // Espacio entre cada letra
93         }
94
95         System.out.println("\nPISTA: Las letras del país son: " + pista.toString());
96         System.out.print(s:"¿Qué país es? (escribe el código): ");
97
98         String entrada = sc.nextLine().trim();
99         if (entrada.equalsIgnoreCase(anotherString:"salir")) {
100             System.out.println(x:"¡Gracias por jugar!");
101             break;
102         }

```

```

103
104         if (entrada.equalsIgnoreCase(p.getCodigo())) {
105             System.out.println("¡Correcto! El país era: " + p.getNombre() + "\n");
106         } else {
107             System.out.println("Incorrecto. El país era: " + p.getNombre() + " (" + p.getCodigo() + ")\n");
108         }
109     }
110     sc.close();
111 }
112 }

```