

Todos (3/6) - Implementar carga de países y juego

quiero crear un juego en consola que lea los datos de un csv

Chat took too long to get ready.  
Please ensure you are signed in to GitHub and that the extension GitHub.copilot-chat is installed and enabled.

quiero crear un juego en consola que lea los datos de un csv en lenguaje java

> Implementing the workflow

Voy a crear un plan de trabajo (todo list) para organizar la creación del juego en Java que lea un CSV. Resultado esperado: una lista de tareas con exactamente una marcada "in-progress" para empezar. Luego actualizaré el plan antes de crear archivos.

Created 5 todos

> Updating task statuses

Starting (2/5) Crear archivos del proyecto

> Preparing CSV and README details

Ahora crearé los archivos del proyecto (código Java, CSV de ejemplo, README y run.bat) en la carpeta GameJavaCSV debajo de tu ruta actual. Debería generar los artefactos necesarios para compilar y ejecutar el juego en Windows. Luego compilaré y ejecutaré una prueba.

juego en windows. Luego compilaré y ejecutaré una prueba.

Created Main.java

Created CsvLoader.java

Created Question.java

Created Game.java

Created questions.csv

Auto approved for this session ([Edit](#))

Created README.md

Auto approved for this session ([Edit](#))

Created run.bat

Auto approved for this session ([Edit](#))

> Managing task progress

He creado los archivos del proyecto (código Java, CSV de ejemplo, README y run.bat). Ahora marco la tarea de creación como completada y paso a compilar y ejecutar una prueba.

Starting (3/5) Compilar y probar

```
cd /d
"c:\Users\josue\Documents\IPC
1\2025\Laboratorio\Actividades
\Actividad10-11\GameJavaCSV"
&& javac -d out src\*.java
```

Enable [shell integration](#) to improve command detection

> Evaluating terminal commands

```
cd /d
"c:\Users\josue\Documents\IPC
1\2025\Laboratorio\Actividades
\Actividad10-11\GameJavaCSV";
```

Add Context... Actividad10-11 +

Add context (#), extensions (@), commands

Agent GPT-5 mini

① Enable **shell integration** to improve command detection

```
Set-Location -Path
"c:\Users\josue\Documents\IPC
1\2025\Laboratorio\Actividades
\Actividad10-11\GameJavaCSV";
javac -d out src\*.java
```

① Enable **shell integration** to improve command detection

> Updating task list

✓ Updated todo list

### Progreso hasta ahora

- Plan y estructura: completado.
- Archivos creados: completado (código Java, CSV de ejemplo, README, run.bat).
- Compilación/prueba: no se pudo ejecutar aquí (entorno remoto/terminal no dispone de la ruta o JDK). Estado marcado como not-started (bloqueado).

### Archivos creados (ruta base):

c:\Users\josue\Documents\IPC  
1\2025\Laboratorio\Actividades\Actividad10-  
11\GameJavaCSV

- src/Main.java — entrada del programa.
- src/Game.java — lógica del juego (preguntas, lectura de opción, puntuación).
- src/CsvLoader.java — cargador simple de CSV (ignora líneas que comienzan con #).
- src/Question.java — POJO de pregunta.
- data/questions.csv — CSV de ejemplo con 5 preguntas.
- README.md — instrucciones de compilación/ejecución y formato del CSV.
- run.bat — script Windows para compilar y

ejecutar.

### Por qué no pude compilar aquí

- El comando de compilación falló porque el path creado no existe en el entorno donde se ejecuta el terminal o no hay JDK disponible desde este entorno. He intentado con variantes para cmd/PowerShell pero el sistema devolvió que la ruta no se encuentra.

### Qué puedes hacer localmente (pasos para Windows cmd.exe)

1. Abre una terminal (cmd.exe).
2. Sitúate en la carpeta del proyecto (la que contiene `src` y `data`), por ejemplo: `cd /d "c:\Users\josue\Documents\IPC1\2025\Laboratorio\Actividades\Actividad10-11\GameJavaCSV"`
3. Compila: `javac -d out src*.java`
4. Ejecuta: `java -cp out Main`

O simplemente doble clic en `run.bat` (o ejecutarlo desde cmd).

### Formato CSV aceptado (resumen rápido)

- Cada línea: pregunta,op1,op2,op3,op4,índice\_respuesta
- Índice de respuesta: 1..4 (se almacena como 0..3 internamente)
- Líneas que empiezan con # se ignoran

### Siguientes pasos que puedo hacer por ti

- Si quieres, puedo:
  - Añadir manejo de comillas y comas dentro de campos CSV (parseador más robusto).
  - Permitir pasar la ruta del CSV por

más robusto).

- Permitir pasar la ruta del CSV por argumento de línea de comandos.
- Añadir tests unitarios y validación adicional.
- Integrar un pequeño menú (selección de número de preguntas, modo cronometrado).
- Intentar compilar aquí si me confirmas que el path es accesible en este entorno o que quieres que use otra ruta.

¿Qué quieres que haga ahora?

- ¿Quieres que implemente alguna mejora (por ejemplo, soporte de comillas en CSV), o prefieres que te guíe para compilar/run localmente y revisar el comportamiento?



GPT-5 mini • 1x



quiero que lea el archivo csv que voy a colocar en la carpeta, se deben crear objetos de tipo Pais y guardarlos en una ArrayList para nutrir de información al juego.

Para jugar, el sistema 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.

Se debe mostrar un mensaje que le indique al usuario si adivinó o no.

Voy a añadir las clases para cargar países y el modo de juego de países, actualizar `Main` y el `README`; espero crear los archivos y luego compilar. Resultado: nuevos archivos `Pais.java`, `CsvPaisLoader.java`, `CountryGame.java`, cambios en `Main.java` y `README.md`. Luego intentaré compilar.

✓ Added 1 todo

✓ Created `Pais.java`

Auto approved for this session (Edit)

✓ Created `CsvPaisLoader.java`

Auto approved for this session (Edit)

✓ Created `CountryGame.java`

Auto approved for this session (Edit)

✓ Using "Apply Patch"

Auto approved for this session (Edit)

> Checking file paths

✓ Read `src`

✓ Searched for files matching `**/*Main.java`, no matches

> Deciding on file creation strategy

✓ Created `Main.java`

Auto approved for this session (Edit)

> Updating README file

✓ Created `README.md`

Auto approved for this session (Edit)

✓ Starting (4/6) Compilar y probar

juego  
> Updating README file

✓ Created `README.md`

Auto approved for this session (Edit)

✓ Starting (4/6) Compilar y probar

```
cmd.exe /c cd /d
"c:\Users\josue\Documents\IPC
1\2025\Laboratorio\Actividades
\Actividad10-11\GameJavaCSV"
&& javac -d out src\*.java
```

① Enable [shell integration](#) to improve command detection

> Executing PowerShell command

```
cmd.exe /c "cd /d
"c:\Users\josue\Documents\IPC
1\2025\Laboratorio\Actividades
\Actividad10-11\GameJavaCSV"
&& javac -d out src\*.java
```

① Enable [shell integration](#) to improve command detection

✓ Updated todo list

He añadido la funcionalidad solicitada y preparé todo para que lo pruebes localmente.

Resumen de cambios realizados

- Nuevas clases en `src/`:
  - `Pais.java` — POJO simple para un país (nombre, primera letra).
  - `CsvPaisLoader.java` — carga una lista de `Pais` desde `data/paises.csv` (ignora líneas vacías y comentarios `#`).
  - `CountryGame.java` — lógica del juego de países: el sistema elige un país al azar, muestra la pista "El país

juego de países: el sistema elige un país al azar, muestra la pista "El país que elegí empieza con: X" y el usuario escribe su respuesta; muestra si acertó o no.

- Cambios en `src/Main.java`:
  - Ahora comprueba si existe `data/paises.csv`. Si existe y contiene países, arranca `CountryGame`. Si no, usa el modo preguntas (archivo `data/questions.csv`) como antes.
- `README.md` actualizado con formato esperado para `paises.csv` y las instrucciones de compilación/ejecución.
- Los demás archivos (`Game.java`, `CsvLoader.java`, `Question.java`, `data/questions.csv`, `run.bat`) quedaron como antes.

#### Dónde colocar tu CSV de países

- Coloca el archivo en: `data/paises.csv`
- Formato admitido (simple):
  - Cada línea comienza con el nombre del país.
  - Las líneas que empiezan con `#` se ignoran.
  - Ejemplo de `data/paises.csv`:  
España México Argentina

#### Cómo compilar y ejecutar localmente en Windows (cmd.exe)

1. Abre cmd.exe y ve a la carpeta del proyecto (la que contiene `src` y `data`), por ejemplo:

```
cd /d "c:\Users\josue\Documents\
```

#### 2. Compila:

```
javac -d out src\*.java
```

#### 3. Ejecuta:

```
java -cp out Main
```

O ejecuta `run.bat` (doble clic o desde cmd).

#### Nota sobre la compilación desde aquí

- Intenté compilar en mi entorno, pero la ruta de tu disco no está accesible desde este entorno remoto (el compilador devolvió "The system cannot find the path specified"). Por ello no pude ejecutar la prueba aquí. Todo el código y archivos han sido creados en la estructura indicada para que los compiles y ejecute en tu máquina.

#### Siguientes pasos (elige uno)

- ¿Quieres que:
  1. Añada soporte para aceptar la ruta del CSV por argumento de línea de comandos? (útil para probar varios CSV sin mover archivos).
  2. Haga el comparador más tolerante (p. ej. ignorar tildes y signos diacríticos en la comparación de nombres de países)?
  3. Permita varias rondas o un contador de intentos por país?
  4. Genere automáticamente un `data/paises.csv` de ejemplo con