

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.nio.charset.StandardCharsets;
import java.util.Random;

public class ServidorHTTP {

    private static int intentosAdivina;
    private static int numeroSecreto;
    private static int marcadorUsuarioDados = 0;
    private static int marcadorServidorDados = 0;
    private static int rondaDados = 0;
    private static int marcadorUsuarioPPT = 0;
    private static int marcadorServidorPPT = 0;
    private static int rondaPPT = 0;

    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(8066);
        System.out.println("Servidor HTTP iniciado en el puerto 8066");
        System.out.println("Visita http://localhost:8066");
        while (true) {
            Socket cliente = serverSocket.accept();
            Thread hiloServidor = new HiloServidor(cliente);
            hiloServidor.start();
        }
    }

    private static class HiloServidor extends Thread {

        private final Socket cliente;
        public HiloServidor(Socket cliente) {
            this.cliente = cliente;
        }

        @Override
        public void run() {
            try (BufferedReader entrada = new BufferedReader(new InputStreamReader(cliente.getInputStream()));
                PrintWriter salida = new PrintWriter(cliente.getOutputStream(), true, StandardCharsets.UTF_8)) {
                // Lee la primera línea de la petición HTTP.
                String peticion = entrada.readLine();
                if (peticion == null || (!peticion.startsWith("GET") && !peticion.startsWith("POST"))) {
                    return; // Ignora la petición si no es GET o POST.
                }
                System.out.println("peticion: " + peticion);
                String ruta = peticion.split(" ")[1]; // Extrae la ruta solicitada.
                String linea;

                // Leer encabezados HTTP y determinar el tamaño del cuerpo.
                int contentLength = 0;
                while (!(linea = entrada.readLine()).isBlank()) {
                    System.out.println("línea: " + linea);
                    if (linea.startsWith("Content-Length: ")) {
                        contentLength = Integer.parseInt(linea.substring(16));
                    }
                }
                System.out.println("línea: vacía");

                // Leer el cuerpo si es un POST.
                StringBuilder cuerpo = new StringBuilder();
                if (peticion.startsWith("POST") && contentLength > 0) {
                    char[] buffer = new char[contentLength];
                    entrada.read(buffer, 0, contentLength);
                    cuerpo.append(buffer);
                }

                String respuesta; // Contendrá la respuesta generada por el servidor.

                if (ruta.equals("/")) {
                    respuesta = construirRespuesta(200, Paginas.html_index);
                } else if (ruta.startsWith("/adivina")) {
                    respuesta = manejarAdivina(cuerpo.toString());
                } else if (ruta.startsWith("/dados")) {
                    respuesta = manejarDados(cuerpo.toString());
                } else if (ruta.startsWith("/ppt")) {
                    respuesta = manejarPPT(cuerpo.toString());
                } else {
                    respuesta = construirRespuesta(404, Paginas.html_noEncontrado);
                }
                //System.out.println("Respuesta: " + respuesta);
                salida.println(respuesta);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

Tarea 03 – Servidor de Juegos HTTP

```
private String manejarAdivina(String cuerpo) {
    if (numeroSecreto == 0) {
        numeroSecreto = new Random().nextInt(100) + 1; // Genera un número aleatorio al iniciar el juego.
        intentosAdivina = 0; // Resetea los intentos.
    }

    int codigo = 200;
    String respuestaHTML;
    System.out.println("Cuerpo Adivina: " + cuerpo);
    try {
        if (!cuerpo.isEmpty()) {
            int numeroUsuario = Integer.parseInt(cuerpo.split("=")[1]);
            intentosAdivina++;

            if (numeroUsuario == numeroSecreto) {
                respuestaHTML = "<p>¡Felicidades! Has acertado el número " + numeroSecreto + " en " + intentosAdivina + " intentos.</p>";
                numeroSecreto = 0; // Reinicia el juego.
            } else if (intentosAdivina >= 10) {
                respuestaHTML = "<p>No has acertado en 10 intentos. El número era " + numeroSecreto + ". Pulsa <a href='/adivina'>aquí</a> para reiniciar el juego.</p>";
                numeroSecreto = 0; // Reinicia el juego.
            } else {
                respuestaHTML = "<p>Intento " + intentosAdivina + ": El número es " + (numeroUsuario < numeroSecreto ? "mayor" : "menor") + ".</p>";
            }
        } else {
            respuestaHTML = "<p>Introduce un número para empezar el juego.</p> ";
        }
    } catch (Exception e) {
        respuestaHTML = "<p>Error procesando tu número. Intenta de nuevo.</p>";
        codigo = 400;
        e.printStackTrace();
    }

    return construirRespuesta(codigo, Paginas.generarHtmlAdivina(respuestaHTML));
}

private String manejarDados(String cuerpo) {
    String resultado;
    int codigo = 200;
    System.out.println("cuerpoDados:" + cuerpo);

    try {
        if (!cuerpo.isEmpty()) {
            Random random = new Random();
            int dadoUsuario = Integer.parseInt(cuerpo.split("=")[1]);
            int dadoServidor = random.nextInt(6) + 1;

            if (dadoUsuario > dadoServidor) {
                marcadorUsuarioDados++;
                rondaDados++;
                resultado = "<p>Ronda " + rondaDados + " .Ganaste esta ronda. Tu dado: " + dadoUsuario + " - Dado del servidor: " + dadoServidor + "</p>";
            } else if (dadoUsuario < dadoServidor) {
                marcadorServidorDados++;
                rondaDados++;
                resultado = "<p>Ronda " + rondaDados + " .Perdiste esta ronda. Tu dado: " + dadoUsuario + " - Dado del servidor: " + dadoServidor + "</p>";
            } else {
                resultado = "<p>Ronda " + rondaDados + " .Empate en esta ronda. Ambos sacaron: " + dadoUsuario + "</p>";
            }

            if (marcadorUsuarioDados + marcadorServidorDados == 5) {
                if (marcadorUsuarioDados > marcadorServidorDados) {
                    resultado += "<p>¡Ganaste el juego! Marcador final: " + marcadorUsuarioDados + " - " + marcadorServidorDados + ". Vuelve a pulsar el botón para jugar de nuevo.</p>";
                } else {
                    resultado += "<p>Perdiste el juego. Marcador final: " + marcadorUsuarioDados + " - " + marcadorServidorDados + ". Vuelve a pulsar el botón para jugar de nuevo.</p>";
                }
                marcadorUsuarioDados = 0;
                marcadorServidorDados = 0;
                rondaDados = 0;
            }
        } else {
            resultado = "<p>Pulsa el botón para lanzar los dados. </p>";
        }
    } catch (Exception e) {
        resultado = "<p>Error procesando tu elección. Intenta de nuevo.</p>";
        codigo = 400;
        e.printStackTrace(); // Muestra errores en la consola.
    }

    return construirRespuesta(codigo, Paginas.generarHtmlDados(resultado));
}

private String manejarPPT(String cuerpo) {
    String[] opciones = {"Piedra", "Papel", "Tijeras"};
    String resultado;
    int codigo = 200;

    System.out.println("cuerpoPPT:" + cuerpo);
    try {
        if (!cuerpo.isEmpty()) {
            String eleccionUsuario = cuerpo.split("=")[1];
            String eleccionServidor = opciones[new Random().nextInt(3)];

            if (eleccionUsuario.equals(eleccionServidor)) {
                resultado = "<p>Ronda " + rondaPPT + " .Empate en esta ronda. Ambos eligieron: " + eleccionServidor + "</p>";
            } else if ((eleccionUsuario.equals("Piedra") && eleccionServidor.equals("Tijeras")) || (eleccionUsuario.equals("Papel") && eleccionServidor.equals("Piedra")) || (eleccionUsuario.equals("Tijeras") && eleccionServidor.equals("Papel"))) {
                marcadorUsuarioPPT++;
                rondaPPT++;
                resultado = "<p>Ronda " + rondaPPT + " .Ganaste esta ronda. Elegiste: " + eleccionUsuario + " - El servidor eligió: " + eleccionServidor + "</p>";
            } else {
                marcadorServidorPPT++;
                rondaPPT++;
                resultado = "<p>Ronda " + rondaPPT + " .Perdiste esta ronda. Elegiste: " + eleccionUsuario + " - El servidor eligió: " + eleccionServidor + "</p>";
            }

            if (marcadorUsuarioPPT + marcadorServidorPPT == 5) {
                if (marcadorUsuarioPPT > marcadorServidorPPT) {
                    resultado += "<p>¡Ganaste el juego! Marcador final: " + marcadorUsuarioPPT + " - " + marcadorServidorPPT + ". Vuelve a pulsar un botón para jugar de nuevo.</p>";
                } else {
                    resultado += "<p>Perdiste el juego. Marcador final: " + marcadorUsuarioPPT + " - " + marcadorServidorPPT + ". Vuelve a pulsar un botón para jugar de nuevo.</p>";
                }
                marcadorUsuarioPPT = 0;
                marcadorServidorPPT = 0;
                rondaPPT = 0;
            }
        } else {
            resultado = "<p>Pulsa un botón para jugar. </p>";
        }
    } catch (Exception e) {
        resultado = "<p>Error procesando tu elección. Intenta de nuevo.</p>";
        codigo = 400;
        e.printStackTrace();
    }

    return construirRespuesta(codigo, Paginas.generarHtmlPpt(resultado));
}

private String construirRespuesta(int codigo, String contenido) {
    return (codigo == 200 ? "HTTP/1.1 200 OK" : "HTTP/1.1 400 Bad Request") + "\n" //Línea inicial
        + "Content-Type: text/html; charset=UTF-8" + "\n" //Metadatos
        + "Content-Length: " + contenido.length() + "\n" //Content-Length
        + "\n" //Línea vacía
        + contenido; //Cuerpo
}
```