

UNIVERSIDAD EAFIT

ESPECIALIZACION EN DESARROLLO DE SOFTWARE ESPECIALIZACIÓN EN TELEINFORMÁTICA

ST725 - SISTEMAS DISTRIBUIDOS

EDWIN MONTOYA, emontoya@eafit.edu.co

Juego el Adivinador “Guess”

Implementar una aplicación distribuida que implemente un juego de adivinar un número entre 0 y 99. El juego debe ser concurrente, es decir, debe permitir jugar a varios jugadores en el mismo servidor. (si bien un jugador solo interactúa con el servidor y no con otros jugadores). Cada vez que se comienza un juego nuevo, el servidor de juegos generará un número aleatorio entre 0 y 99, el cual el cliente adivinará. El jugador puede realizar las siguientes operaciones sobre el objeto de juego:

“begin”: Iniciar de nuevo un juego

“test”: Envía un número al servidor para adivinar

“count”: Retorna del servidor, el número de veces que ha intentado adivinar.

“quit”: termina el juego

La interfaz para el componente del lado del “Servidor de Juegos” podría ser:

```
public interface IGuess {  
    public void begin();  
    public int test(int num);  
    public int count();  
    public void quit();  
}
```

“test” retorna:

0 si acierta

<0 si debe ser menor

>0 si debe ser mayor

“count” retorna:

n, el número de veces que ha intentado.

Debe tener en cuenta, que el componente del lado del servidor, debe identificar en todo momento al jugador que esta participando, como pueden haber varios simultaneamente, el “Servidor de Juegos” debe identificarlos claramente. Esto se puede lograr de muchas maneras, una posible solución es que cada vez que se inicie una sesión, le solicite un

“apodo” o “nickname” para asociarlo a la sesión. Una posible variante de la interfaz IGuess para manejar sesión podría ser:

```
public interface IGuess2 {  
    public void begin(String nickname);  
    public int test(String nickname, int num);  
    public int count(String nickname);  
    public void quit();  
}
```

Otra forma de mantener sesión, se implementa utilizando el concepto de “factory”, es decir, se le solicita al servidor que cree o destruya un objeto Guess, de acuerdo a peticiones del usuario, este es la manera que más se utiliza en objetos distribuidos, para esto el servidor debe implementar la siguiente interfaz:

```
public interface IGuessFactory {  
    public IGuess create(String nickname);  
    public void destroy(String nickname);  
}
```

Ejecución:

Una vez el jugador ha iniciado un juego nuevo, cada vez que el jugador envía al servidor un número para ser probado, el módulo cliente se comunica con el servidor de juegos enviándole el número, como resultado de esto, el jugador recibe la siguiente información:

“Ha acertado en X intentos”

“El número debe ser MAYOR”

“El número debe ser MENOR”

X es el número de veces que intento adivinar el número.

Este proceso se repite nuevamente, hasta que adivine el número, o el jugador desee reiniciar el juego, en cuyo caso, se debe generar un nuevo número aleatorio en el Servidor de Juegos y se debe poner a cero (0) en contador de intentos.