



UNIVERSIDADE DA CORUÑA

Facultad de Informática

3º Grado en Ingeniería Informática. Ingeniería de Software

Arquitectura Software

Curso 2016-2017

PRÁCTICA GRUPAL

ARQUITECTURA SOFTWARE

Chas Álvarez, Lucía (lucia.chas)

Cortón de Blas, Diego (diego.cortonde.blas)

Costa García, Sergio (sergio.costag)

Freire Paz, Adrián (adrian.freire.paz)

Toledano Franco, Saínza (sainza.toledano)

Vidal García, Sara (sara.vidal)

ÍNDICE

| | |
|---|---|
| 1. Descripción general no formal | 3 |
| 2. Requisitos generales | 4 |
| i. Requisitos funcionales | 4 |
| ii. Requisitos no funcionales | 5 |
| 3. Estilo arquitectónico | 6 |
| 4. Tácticas | 7 |
| 5. Diseño de alto nivel: Vista física | 7 |

1. DESCRIPCIÓN GENERAL NO FORMAL

La práctica consistirá en el diseño e implementación de un sistema lúdico en el que se podrá jugar al juego de cartas español "Siete y medio". Para ello, se realizará una extracción de requisitos funcionales y no funcionales, se elegirá la arquitectura software más adecuada para cumplirlos, se realizará el diseño y finalmente se implementará en lenguaje Erlang.

El sistema consta de varias salas de creación dinámica entre las que se repartirán los jugadores. Cuando el primer jugador entra al sistema, se inicia un temporizador y se espera la llegada de más jugadores. Cuando se alcance el aforo máximo permitido por sala o el temporizador termine, se enviará a los jugadores que hayan llegado al sistema a una sala para que puedan comenzar a jugar. Esto será realizado por el balanceador, que de nuevo se quedará a la espera de que lleguen nuevos jugadores.

Cada jugador recibirá 100 fichas al entrar en la sala, que podrá apostar convenientemente en las distintas partidas que dispute. Las fichas solo tendrán validez dentro de la sala en la que el jugador ha entrado, por lo que si sale y entra de nuevo en el sistema recuperará las 100 fichas iniciales.

Aunque no es el aspecto más relevante de la práctica, las normas a seguir para el juego del Siete y Medio son las que se detallan a continuación:

1. Por turnos, los jugadores indican la cantidad de fichas que desean apostar. Tendrán un tiempo limitado para realizar la apuesta; aquellos que no apuesten en ese tiempo serán expulsados de la sala.

2. Cada jugador recibe una carta boca abajo (carta cubierta), y la banca recibe una boca arriba (descubierta).

3. Comienza el juego. En cada turno, el jugador podrá elegir entre una serie de acciones: plantarse, abandonar, pedir carta cubierta y pedir carta descubierta. De nuevo, hay un tiempo limitado para elegir la acción a realizar, y si termina el tiempo antes de que el jugador haya realizado alguna acción, será expulsado de la sala.

- 3.1. Si el jugador se planta, se pasa el turno al siguiente jugador (o la banca, en caso de ser el último jugador).

- 3.2. Si el jugador abandona, es expulsado de la sala y la partida continúa sin él.

3.3. Si el jugador pide una carta descubierta, recibe una nueva carta y se pasa el turno al siguiente (o a la banca, en caso de ser el último jugador).

3.4. Si el jugador pide una carta cubierta, se descubre una de las cartas cubiertas que ya tuviese en su mano y se le da una nueva carta boca abajo, y se pasa el turno al siguiente (o a la banca, en caso de ser el último jugador).

4. Al final de cada ronda de juego, la banca puede realizar dos acciones: plantarse, o recibir una carta descubierta.

5. La partida termina cuando todos los jugadores se han plantado o la suma de sus cartas supera el siete y medio.

6. Se revela la puntuación de cada jugador y de la banca. Los resultados posibles son:

-La banca se acerca más al siete y medio que el jugador: Gana la banca, y el jugador pierde lo apostado.

-La banca y el jugador tienen la misma puntuación: Se produce un empate, y el jugador no gana ni pierde fichas.

-El jugador se acerca más al siete y medio que la banca: Gana el jugador, y el jugador gana el doble de fichas que las apostadas.

2. REQUISITOS GENERALES

2.1. REQUISITOS FUNCIONALES

RF-<01> Crear salas: Crear las salas que sean necesarias según la cantidad de jugadores que accedan al sistema.

RF-<02> Añadir jugadores a una sala: Añadir a los jugadores a las salas de forma balanceada según van entrando al sistema, considerando el máximo de jugadores permitidos por sala.

RF-<03> Gestionar partida: Controlar la partida y las distintas acciones vinculadas a ella:

- Apostar
- Recibir carta cubierta
- Recibir carta descubierta
- Abandonar
- Plantarse

2.2. REQUISITOS NO FUNCIONALES

RNF-<01> Robustez: El sistema debe asegurar que un fallo en un servicio no afecte al resto de servicios. Dentro del servicio que falla, se deberá controlar su correcta terminación y, en la medida de lo posible, su recuperación. Además, el sistema debe poder reaccionar a otras situaciones imprevistas de forma limpia y adecuada. En este caso, las salas (servicios) deben cerrarse y expulsar a los jugadores, de modo que estos puedan unirse a otras salas que estén en funcionamiento.

RNF-<02> Escalabilidad: El sistema debe poder aumentar su tamaño según lo requieran las circunstancias (inclusión de nuevos servicios, replicación de los recursos etc), y debe poder cambiarse su configuración sin perder la calidad presente en el sistema. Por ello se ha decidido elaborarlo con componentes fácilmente replicables.

RNF-<03> Rendimiento: Debido a la naturaleza lúdica del sistema, este debe rendir bien y proporcionar fluidez a las partidas, para que los jugadores no se sientan frustrados. Además, este rendimiento debe estar presente bajo cualquier cantidad de carga del servidor, reduciéndose en la menor medida posible. Por este motivo se ha decidido que el balanceador y la sala deberían ser el foco de las tareas de optimización del sistema.

RNF-<04> Usabilidad: El sistema debe ser sencillo de utilizar. Para ello es necesario que los controles sean intuitivos y fáciles de comprender, y que la curva de aprendizaje sea mínima.

RNF-<05> Eficiencia: El sistema cumple con sus funcionalidades sin realizar más pasos ni gastar más recursos de los necesarios, pudiendo hacer así que el sistema soporte más peticiones.

RNF-<06> Confiabilidad: Corresponde a la seguridad de que el sistema realizará correctamente sus funciones sin dar lugar a fallos.

RNF-<07> Disponibilidad: El sistema debe estar disponible y abierto a peticiones el mayor porcentaje de tiempo posible.

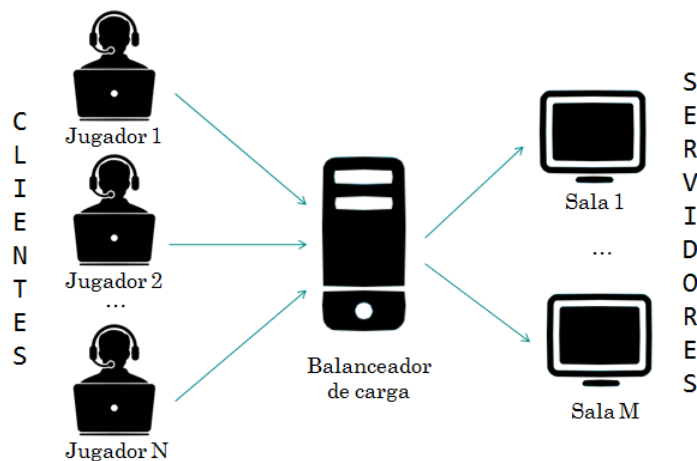
Los requisitos del **05** al **07** no se han tenido tanto en cuenta debido a que los principales requisitos no funcionales valorados o bien resuelven el mismo problema, o lo abarcan desde otra perspectiva que favorece más al sistema.

3. ESTILO ARQUITECTÓNICO

La arquitectura que se ha decidido utilizar es la de cliente-servidor. Esto se debe a que esta arquitectura proporciona una gran robustez, puesto que los fallos en un servicio no afectan al funcionamiento del resto de servicios ofrecidos por el sistema, ya que se basa en la organización de servicios independientes. Los jugadores serían los clientes, y las salas los servicios. Otra de las ventajas de esta arquitectura es la capacidad de escalabilidad que tiene, pudiendo añadir servidores o replicando los balanceadores.

Sin embargo, la arquitectura cliente-servidor presenta el inconveniente de aportar un bajo rendimiento al sistema debido a la inclusión de elementos intermedios para controlar el flujo hacia los servidores, como por ejemplo balanceadores o directorios.

Se ha decidido dar prioridad a la robustez por encima del rendimiento, ya que se considera de mayor importancia que el sistema funcione aunque alguna sala tenga un problema, que el rendimiento de las mismas. Además, para mitigar el problema del rendimiento, se optimizará en la medida de lo posible los elementos intermedios (en este caso el balanceador de carga).



4. TÁCTICAS

- **Rendimiento:** Para mejorar el rendimiento, optimización del balanceador de carga.
- **Disponibilidad:** Para mejorar la disponibilidad, se da un tiempo máximo para que los jugadores entren a una sala, apuesten o realicen una acción durante su turno.
- **Escalabilidad:** Replicación del balanceador, aunque esto conllevaría un equipo más potente para no perder rendimiento.

De cara al futuro sería interesante valorar la inclusión de tácticas de seguridad para permitir que los usuarios se registren en la aplicación y así poder llevar un mayor control de los mismos, de sus apuestas, partidas ganadas y fichas. Eso implicará utilizar tácticas como la autenticación de los usuarios y el aseguramiento de la integridad de los datos.

5. DISEÑO DE ALTO NIVEL: VISTA FÍSICA

