



UNIVERSIDADE DA CORUÑA
FACULTADE DE INFORMÁTICA
Departamento de Computación

**PROXECTO DE FIN DE CARREIRA DE ENXEÑERÍA
TÉCNICA EN INFORMÁTICA DE XESTIÓN**

**DESENVOLVEMENTO DUNHA APLICACIÓN
QUE PERMITA XOGAR, POR INTERNET, EN
TEMPO REAL AO BLACKJACK**

Autor: Alejandro Mosquera Basoa
Director: José María García-Tizón Iglesias
Data: A Coruña, 28 de Setembro de 2010

A mi madre

Agradecimientos

Quiero agradecer a toda la gente que ha confiado en el final de este proyecto, a mi tutor por comprender mis dudas y guiarme para realizarlo y a mis amigos y compañeros de carrera por no pensar en cuando se acabaría si no en como lo llevaba.

También agradecer a mi familia el apoyo que me han dado durante toda la carrera, sin ellos no hubiese sido posible ni si quiera empezarla, sobre todo mis padres, Rubén y Raquel.

Y a Paco, Antonio y Lola por haberme hecho como soy.

Índice

Agradecimientos.....	1
Índice	2
Índice de figuras.....	5
1. Introducción.....	7
2. Análisis de la situación actual	9
3. Objetivo.....	10
4. Palabras Clave	11
5. Análisis y requisitos del sistema	12
5.1 Casos de Uso.....	12
5.2 Descripción de los actores	16
5.3 Descripción de los Casos de Uso	17
5.4 Diagramas de Secuencia del Sistema	24
5.4.1 Autenticar.....	25
5.4.2 Registrarse.....	27
5.4.3 Apostar	29
5.4.4 Entrar en Mesa	30
5.4.5 Pedir Carta	32
6. Modelo de Clases.....	34
6.1 Diagrama de Clases del Dominio	34
6.2 Descripción de las Clases del Dominio	36
7. Diseño.....	39

7.1 Diagramas de Secuencia	40
7.1.1 Autenticarse.....	41
7.1.2 Registrarse.....	43
7.1.3 Apostar.....	45
7.1.4 Entrar en Mesa.....	47
7.1.5 Pedir Carta	49
7.2 Diagrama de Clases Software.....	51
7.3 Diagrama de Despliegue.....	53
7.4 Diagrama relacional de la Base de Datos.....	55
8. Implementación en las fases del proceso de desarrollo	57
8.1 Primer incremento.....	58
8.2 Segundo incremento.....	59
8.3 Tercer incremento.....	60
8.4 Cuarto incremento	61
8.5 Quinto incremento	63
8.6 Sexto incremento	64
8.7 Séptimo incremento.....	65
8.8 Octavo incremento	66
8.9 Noveno incremento	67
9. Pruebas	68
10. Manual	70
10.1 Login.....	71
10.2 Registro.....	74
10.3 Acceso a la aplicación	78
10.4 Sentado en Mesa.....	80
10.5 Barra de tareas	87

10.6 Contenido del CD	89
11. Requisitos y tutorial de instalación	90
11.1 Instalación de Java.....	91
11.2 Instalación del driver JDBC	92
11.3 Instalación de MySQL	93
12. Conclusiones	95
13. Líneas futuras.....	96
14. Bibliografía.....	98
15. Links de interes.....	99
16. Apéndice.....	100
16.1 Historia del BlackJack	101
16.1.1 Evolución de las reglas	102
16.2 Cliente.....	105
16.3 Java	106
16.4 MySQL	107
16.5 Netbeans	108
16.6 Servidor.....	109
16.7 UML.....	111
16.8 Palabras clave	113
16.9 Estado del Arte (Java).....	115
16.10 Metodología utilizada.....	117
16.11 MVC.....	120

Índice de figuras

1 Diagrama de Casos de Uso -Administrador-	13
2 Diagrama de Casos de Uso – Usuario Registrado -	14
3 Diagrama de Casos de Uso – Visitante -.....	15
4 Diagrama de Secuencia del Sistema –Autenticar-.....	26
5 Diagrama de Secuencia del Sistema –Registrarse-.....	28
6 Diagrama de Secuencia del Sistema –Apostar-	29
7 Diagrama de Secuencia del Sistema –Entrar en Mesa-	31
8 Diagrama de Secuencia del Sistema –Pedir Carta-	33
9 Diagrama de Clases del Dominio	35
10 Diagrama de Secuencia –Autenticarse-	42
11 Diagrama de Secuencia –Registrarse-	44
12 Diagrama de Secuencia –Apostar-.....	46
13 Diagrama de Secuencia –Entrar en Mesa-	48
14 Diagrama de Secuencia –Pedir Carta-.....	50
15 Diagrama de Clases Software	52
16 Diagrama de Despliegue.....	54
17 Modelo Entidad-Relación.....	56
18 Cliente primitivo	62
19 Servidor	70
20 Cliente –Login-.....	71
21 Cliente –Login Error-	72
22 Cliente –Login Error 2-.....	73

23 Cliente -Registro-	74
24 Cliente -Registro Error-	76
25 Cliente -Registro Error 2-	77
26 Cliente -Sala-	78
27 Cliente -Mesa-	80
28 Cliente -Mesa 2-	82
29 Cliente -Mesa 3-	83
30 Cliente -Mesa 4-	84
31 Cliente -Mesa 5-	85
32 Cliente -Barra de Tareas-	87
33 Cliente -Barra de Tareas 2-	88
34 Pagina Web	96
35 MySQL	107
36 Netbeans -6.8-	108
37 Iteraciones en el Proceso Unificado	119

1. Introducción

El objetivo de este proyecto es desarrollar una aplicación que permita jugar al BlackJack online, de manera que varios jugadores puedan seguir una partida como si se encontrasen físicamente juntos ante un mismo tapete.

A la hora de iniciar la partida, los jugadores tendrán la opción de seleccionar la mesa que prefieran entre aquellas configuradas por el administrador, pudiendo elegir entre mesas de entrenamiento (sin apostar puntos) y mesas de competición (mesas que tendrán efecto en el ranking de jugadores).

La posibilidad de seguir la tabla de ranking aumenta la competitividad y el atractivo del juego.

Se busca entonces :

- Crear una aplicación que permita una partida online al Blackjack.
Se podrá ejecutar desde cualquier Sistema Operativo con el único requisito de que el usuario tenga Java instalado en su ordenador.

- Permitir la creación y modificación de diversas mesas. El administrador dispondrá de las funcionalidades típicas de administración de mesas (usuarios, tipo de partida, puntos en juego, etc.).

Para que los usuarios puedan disfrutar de todas las opciones que permite el juego han de registrarse la primera vez que descarguen y ejecuten el programa, una vez hecho este paso, simplemente con autenticarse podrán hacer uso de todas estas opciones.

El servidor permitirá que varias mesas se encuentren con usuarios activos simultáneamente.

2. Análisis de la situación actual

A medida que evolucionan las tecnologías se hace más fácil la inclusión de ordenadores y conexiones en cualquier lugar del mundo, ampliando exponencialmente el uso de Internet.

En el caso de los juegos online se refleja perfectamente este aumento de usuarios de manera que lejos de repartirse con la masificación de aplicaciones online, el número de estos usuarios se ve aumentado.

Se podría decir que la realización de un juego online es una empresa con un éxito casi seguro y más en el caso de un juego de cartas puesto que el público al que se ofrece incluye prácticamente el rango de edades en su totalidad.

3. Objetivo

El objetivo del proyecto es la creación de una aplicación Cliente/Servidor que pueda dar el servicio de juego de BlackJack a los usuarios.

No podemos olvidar tampoco que el proyecto de fin de carrera lejos de ofrecer una solución competitiva exige un planteamiento capaz de hacer que el alumno aprenda algunas de las tecnologías de amplia difusión en el entorno empresarial, así como las metodologías que formalizan el desarrollo de un proyecto, además de conseguir la funcionalidad descrita en la fase de inicio.

4. Palabras Clave

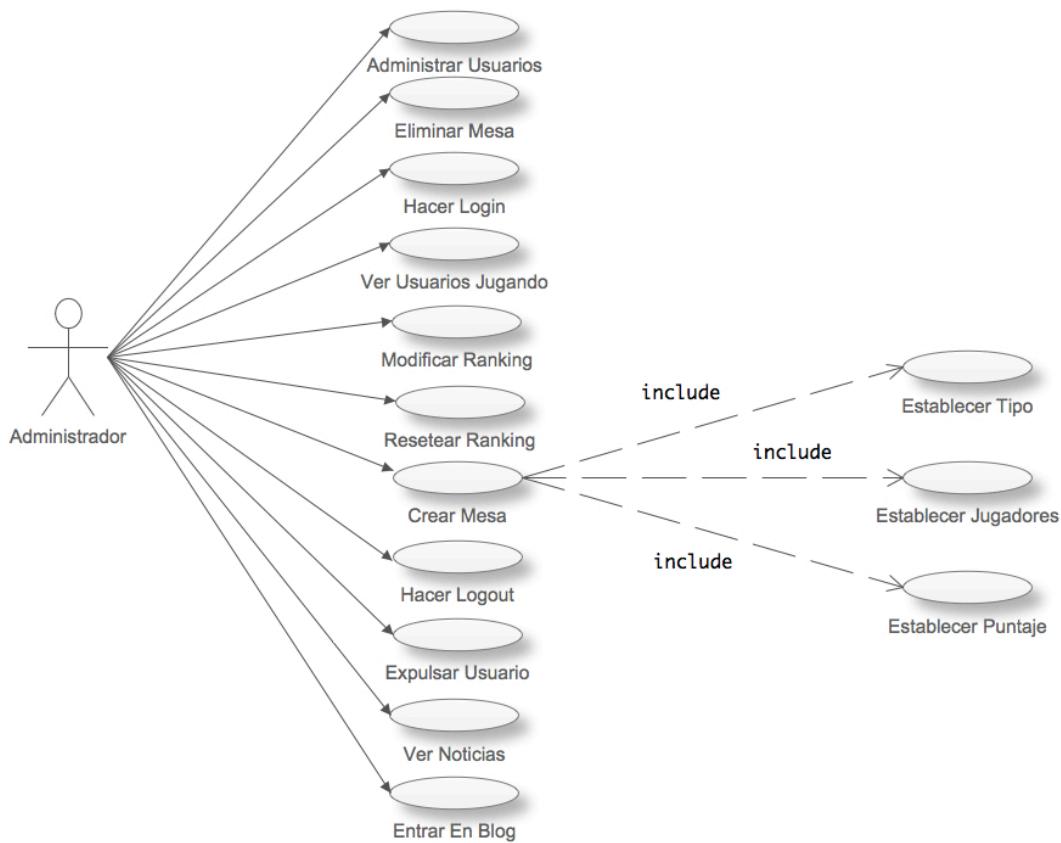
Cliente, Java, MySQL, Netbeans, Servidor, UML, MVC.

*.Existe una explicación de las mismas en el Apéndice.

5. Análisis y requisitos del sistema

5.1 Casos de Uso

Una de las características del Proceso Unificado es que está dirigido por los casos de uso. Una vez hecha la descripción inicial del sistema se procedió a recopilar los requisitos funcionales mediante un diagrama de casos de uso, así como a detectar los actores que interactuarían con el sistema.

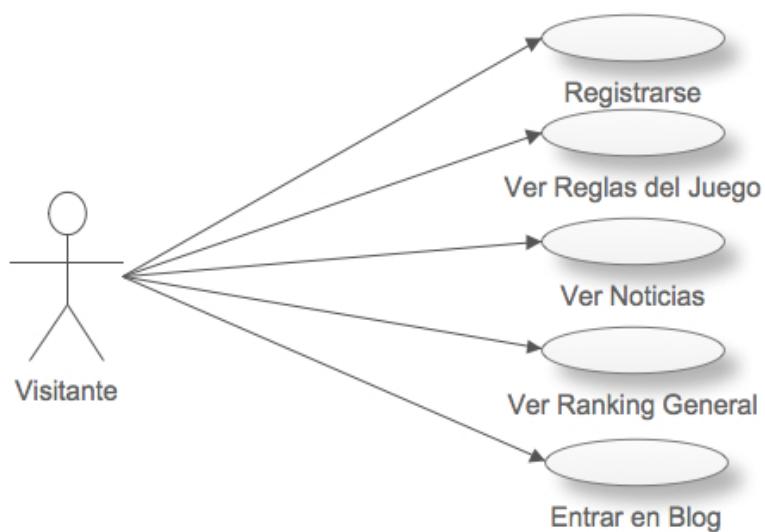


1 Diagrama de Casos de Uso -Administrador-

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



2 Diagrama de Casos de Uso - Usuario Registrado -



3 Diagrama de Casos de Uso – Visitante -

5.2 Descripción de los actores

Se le llama Actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos así como a entidades abstractas como el tiempo.

- Usuario Registrado : nombre que damos a las personas registradas y autenticadas en el sistema.
- Jugador : nombre que damos a los usuarios registrados que se encuentran dentro de alguna partida.
- Mirón : nombre que damos a los usuarios registrados que estan observando alguna mesa sin jugar en ella.
- Administrador : nombre que damos a la persona encargada del mantenimiento del Servidor.
- Visitante : nombre que damos a todas las personas que ejecutan la aplicación.

5.3 Descripción de los Casos de Uso

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.

<i>1. Autenticar</i>	
<i>Actor Principal</i>	Usuario Registrado
<i>Descripción</i>	El usuario introduce sus datos con el fin de conectarse al servidor.
<i>Secuencia</i>	<ol style="list-style-type: none"> 1. El Usuario Registrado introduce su nick y su password. 2. El Sistema Cliente envía los datos al Sistema Servidor. 3. El Sistema Servidor envía la confirmación al Sistema Cliente. 4. El Usuario accede a la Zona de Mesas.
<i>Pre Condición</i>	<ol style="list-style-type: none"> 1. El usuario debe estar registrado.
<i>Frecuencia</i>	Alta
<i>Escenario Alternativo</i>	<ol style="list-style-type: none"> 3. a. El usuario registrado introduce mal alguno de los datos. <ol style="list-style-type: none"> 1. El sistema servidor devuelve un error al Sistema cliente. 2. El sistema cliente devuelve error al usuario.

<i>2. Registrarse</i>	
<i>Actor Principal</i>	Visitante
<i>Descripción</i>	El visitante deja constancia de sus datos para ser recordado en las sucesivas visitas y para tener acceso a todas las funcionalidades.
<i>Secuencia</i>	<ol style="list-style-type: none"> 1. El Visitante introduce sus datos en el formulario de registro. 2. El Sistema Cliente comprueba los datos introducidos. 3. Se almacena el nuevo usuario.
<i>Pre Condición</i>	1. El nombre de usuario y el email no están duplicados en el sistema.
<i>Frecuencia</i>	Media

<i>Escenario Alternativo</i>	<p>2. a. El Visitante introduce un dato duplicado</p> <p>1. El sistema servidor devuelve un error al Sistema cliente.</p> <p>2. El sistema cliente devuelve el error al usuario</p> <p>2. b El Visitante introduce una contraseña sin una longitud mínima.</p> <p>1. El sistema servidor devuelve un error al Sistema cliente.</p> <p>2. El sistema cliente devuelve el error.</p>
<i>3. Apostar</i>	
<i>Actor Principal</i>	Jugador
<i>Descripción</i>	El jugador decide una cantidad que apostar para la partida.
<i>Secuencia</i>	<p>1. El Jugador selecciona la cantidad deseada para la apuesta.</p> <p>2. El Sistema Cliente envía la apuesta al Sistema Servidor.</p> <p>3. El Sistema Servidor envía la confirmación al Sistema Cliente.</p>

	4. El Sistema Cliente muestra la confirmación al Jugador.
<i>Pre Condición</i>	1. El Jugador está en su turno.
<i>Frecuencia</i>	Muy Alta.
<i>Escenario Alternativo</i>	3. a. El Jugador no dispone de saldo suficiente. 1. El sistema Servidor devuelve un error al Sistema Cliente. 2. El Cliente devuelve error al Jugador.

<i>4. Entrar en Mesa</i>	
<i>Actor Principal</i>	Jugador
<i>Descripción</i>	El jugador elige una mesa en la que sentarse a jugar y se sienta en ella.
<i>Secuencia</i>	1. El Jugador selecciona la mesa y el tipo de mesa en la que sentarse. 2. El Sistema Cliente envía la solicitud al Sistema Servidor. 3. El Sistema Servidor envía la

	<p>confirmación al Sistema Cliente.</p> <p>4. El Sistema Cliente muestra la confirmación al Jugador.</p>
<i>Pre Condición</i>	1. Hay alguna mesa disponible.
<i>Frecuencia</i>	Alta.
<i>Escenario Alternativo</i>	<p>3. a. En el momento de sentarse no quedan sitios libres.</p> <p>1. El sistema Servidor devuelve un error al Sistema Cliente.</p> <p>2. El Sistema Cliente devuelve el error al Jugador.</p>

<i>5. Pedir Carta</i>	
<i>Actor Principal</i>	Jugador
<i>Descripción</i>	El jugador solicita una carta con el fin de obtener una mejor jugada.
<i>Secuencia</i>	<p>1. El Jugador solicita una carta.</p> <p>2. El Sistema Cliente envía la solicitud al</p>

	<p>Sistema Servidor.</p> <p>3. El Sistema Servidor envía la carta al Sistema Cliente.</p> <p>4. El Sistema Cliente muestra la carta al Jugador.</p>
<i>Pre Condición</i>	<p>1. El jugador está en su turno y no ha superado los 21 puntos.</p>
<i>Frecuencia</i>	Muy Alta.
<i>Escenario Alternativo</i>	<p>4. a. Con la nueva carta el jugador supera los 21 puntos.</p> <p>1. El sistema Servidor devuelve un aviso al Sistema Cliente.</p> <p>2. El Sistema Cliente devuelve el aviso al Jugador.</p>

5.4 Diagramas de Secuencia del Sistema

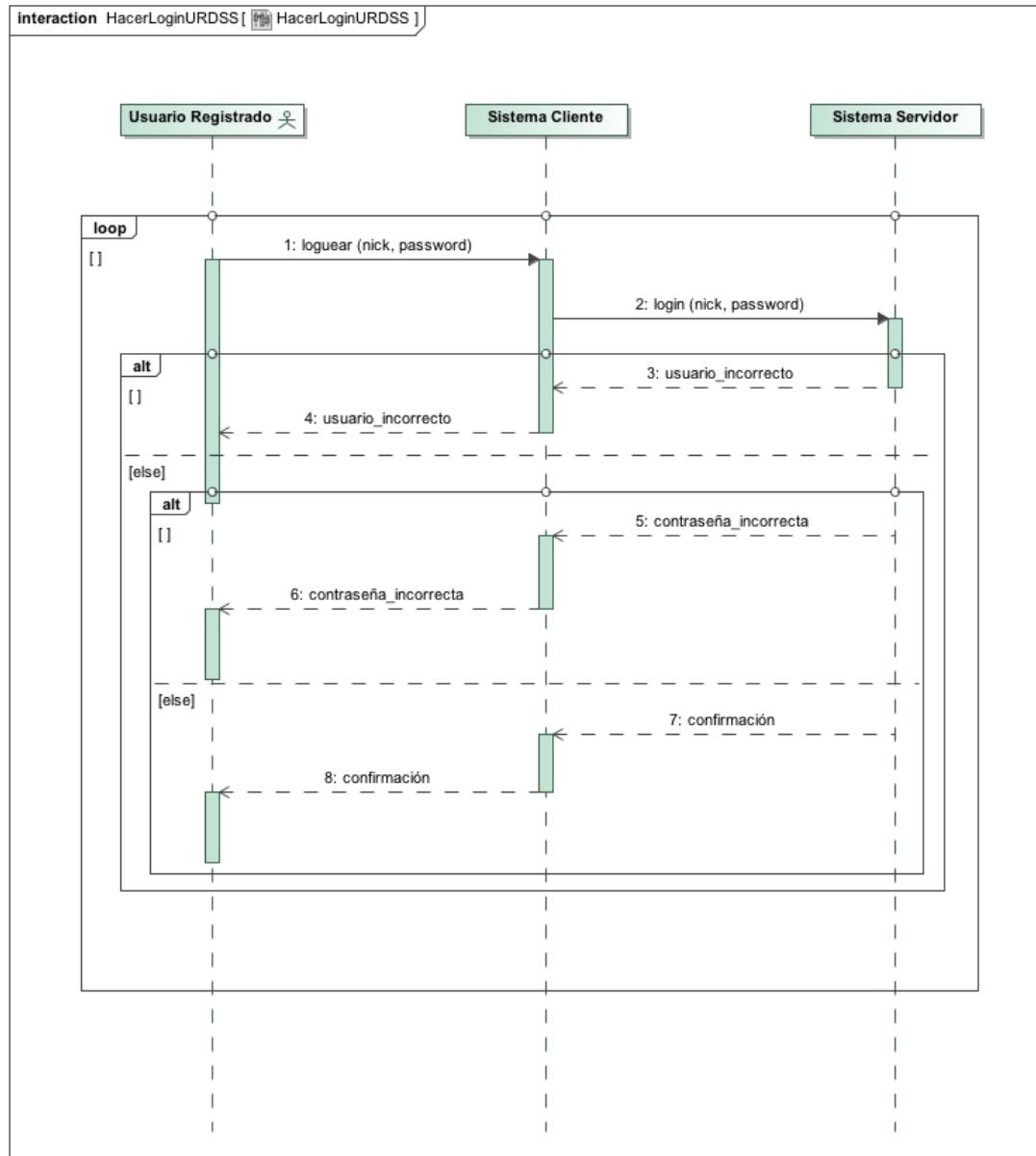
Un diagrama de secuencia del sistema es un gráfico que muestra, para un escenario particular de un caso de uso, los eventos que generan actores externos, su orden y los eventos entre sistemas. Todos los sistemas son tratados como cajas negras; el énfasis de los diagramas está en los eventos que cruzan la frontera del sistema, desde los actores al sistema.

A continuación se muestran los DSS asociados a los casos de uso explicados anteriormente.

5.4.1 Autenticar

Muestra el flujo de eventos que ocurren cuando un Usuario Registrado accede al sistema, mostrando también el flujo alternativo en caso de que introduzca mal algún dato.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

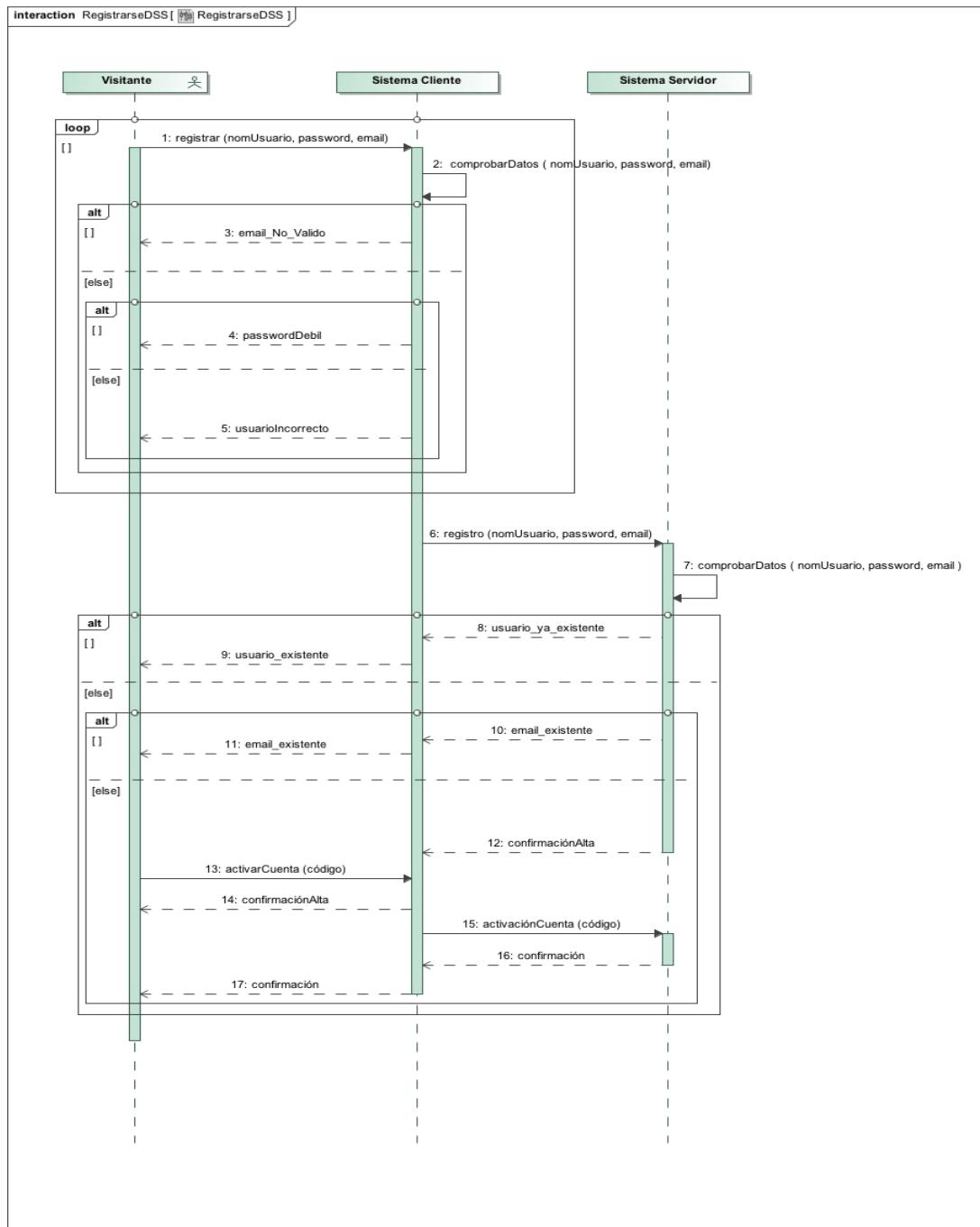


4 Diagrama de Secuencia del Sistema -Autenticar-

5.4.2 Registrarse

En este DSS se observa el flujo de eventos necesario para que un Visitante se Registre en el sistema.

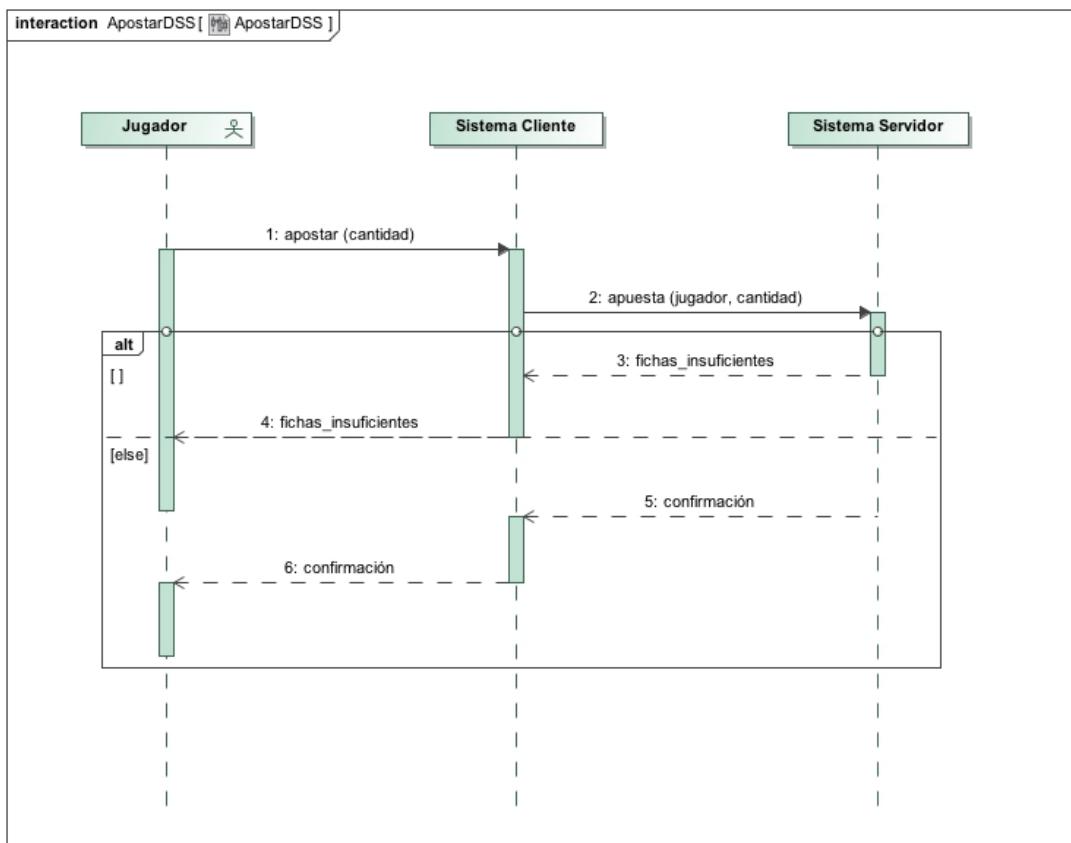
Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



5 Diagrama de Secuencia del Sistema -Registrarse-

5.4.3 Apostar

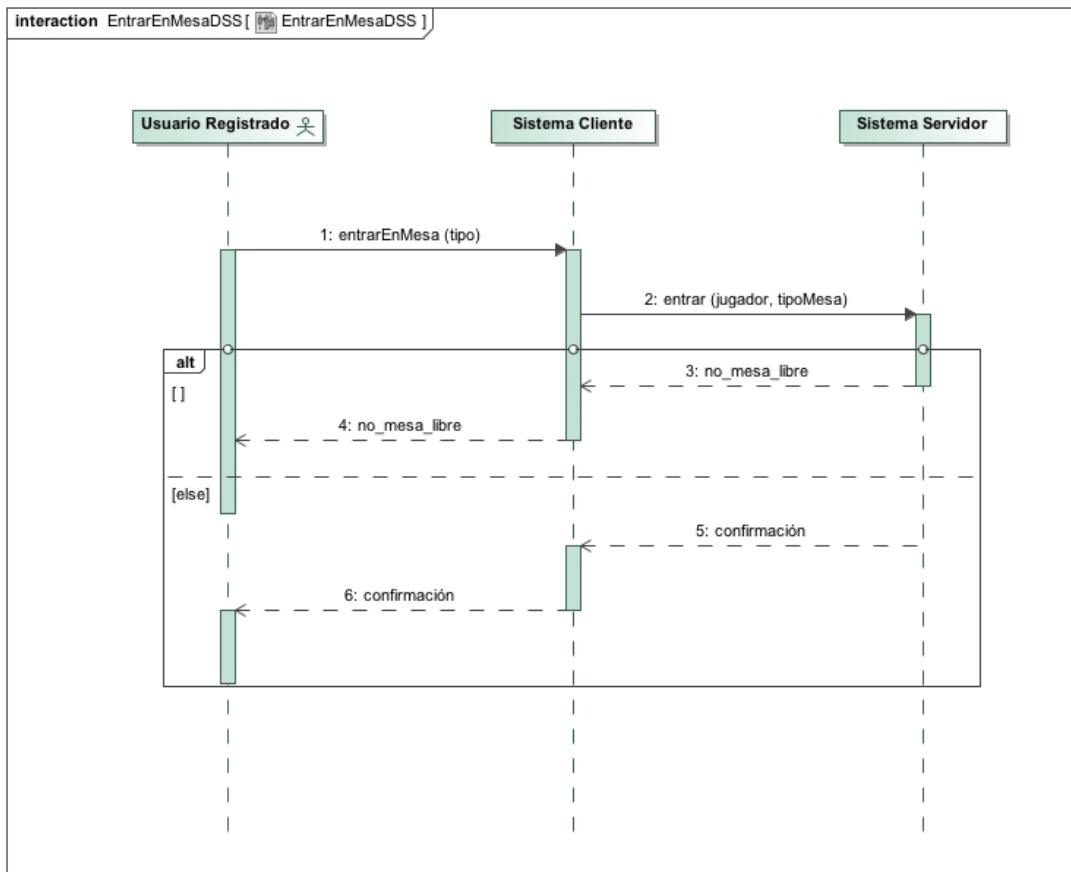
Muestra la sucesión de eventos necesaria para que un Jugador realice una apuesta a la hora de entrar en una mesa competitiva.



6 Diagrama de Secuencia del Sistema -Apostar-

5.4.4 Entrar en Mesa

Este DSS nos muestra el flujo de eventos necesario para que un Usuario Registrado entre en una Mesa de Juego. Mostrando también el flujo de eventos alternativo en caso de que la mesa se llene cuando el Jugador decide entrar en ella, este último punto sólo afectaría a las mesas competitivas, como este incremento es una línea futura se decidió que los usuarios pudieran entrar sin restricción en la mesa, lo cual se ha reflejado en la implementación y en los DS .

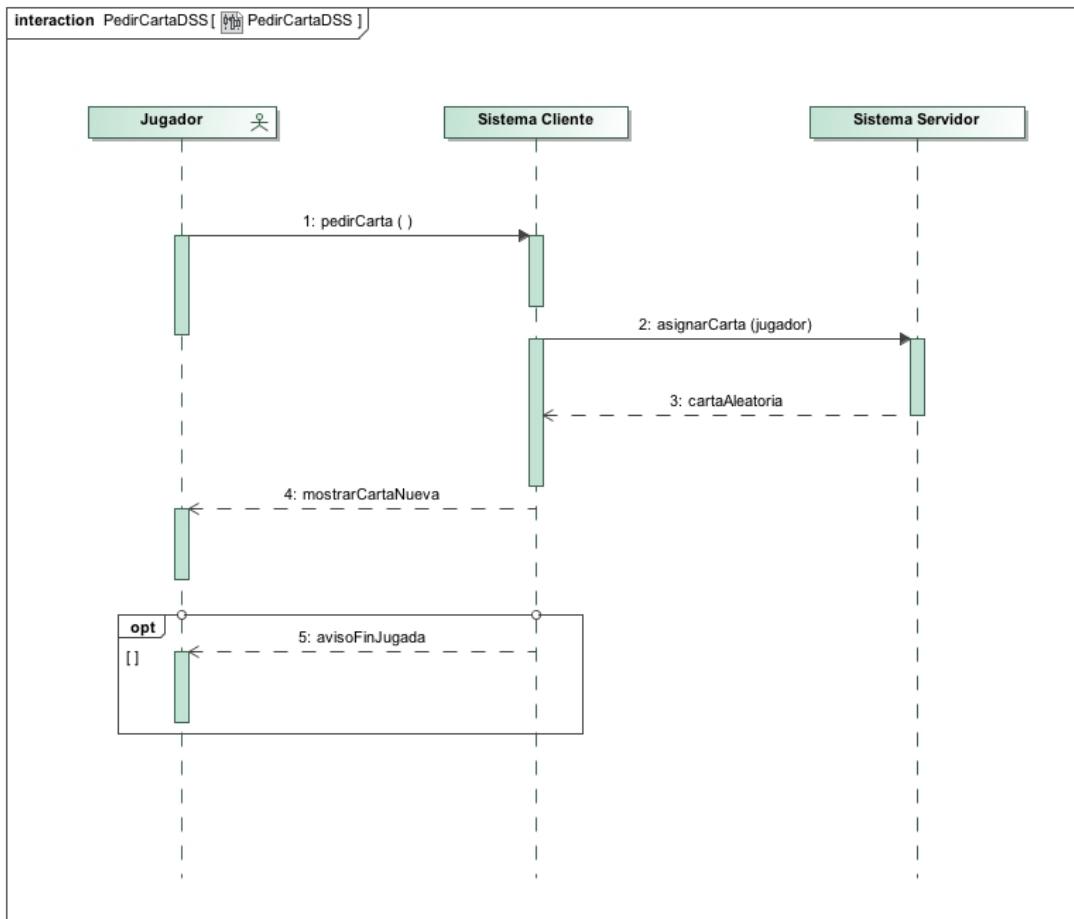


7 Diagrama de Secuencia del Sistema –Entrar en Mesa–

5.4.5 Pedir Carta

Este DSS nos muestra el flujo de eventos que sigue el Sistema cuando un Jugador decide pedir otra carta para mejorar su mano en la partida, así como el flujo alternativo en caso de que con esa carta supere los 21 puntos máximos.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



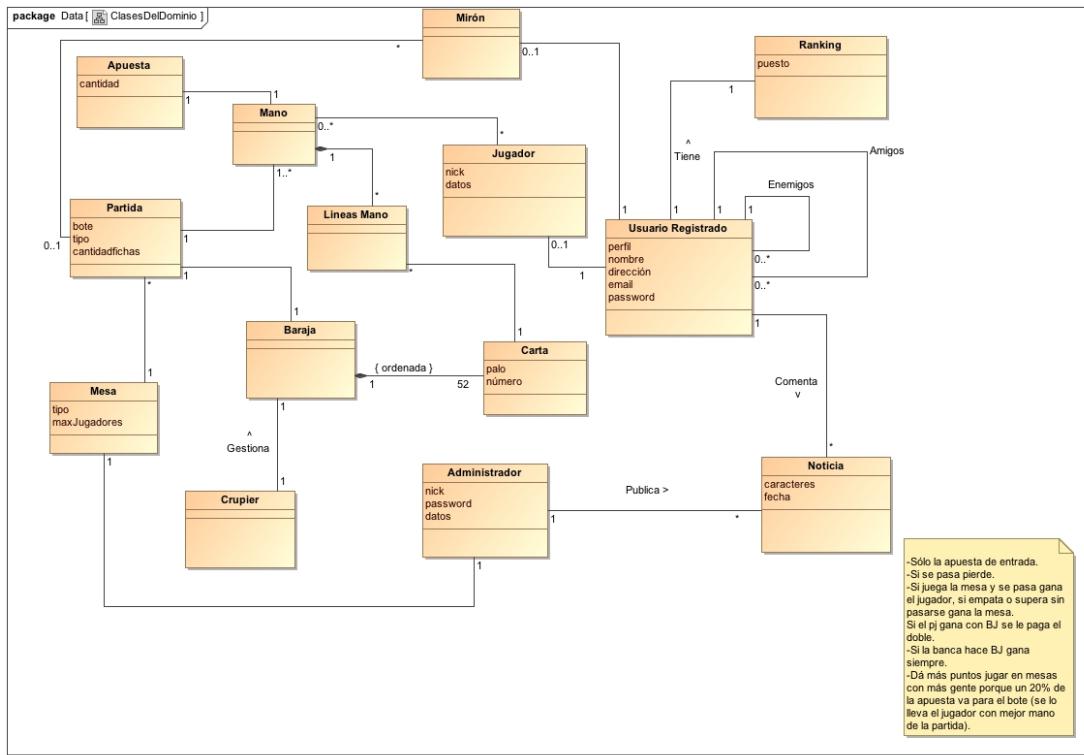
8 Diagrama de Secuencia del Sistema –Pedir Carta–

6. Modelo de Clases

6.1 Diagrama de Clases del Dominio

Es el diagrama que representa los conceptos del dominio que se está estudiando. Estos conceptos se relacionan de manera natural con las clases que los implementan, pero con frecuencia no hay una correlación directa. De hecho, los modelos conceptuales se deben dibujar sin importar (o casi) el software con que se implementarán, por lo cual se pueden considerar como independientes del lenguaje.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



9 Diagrama de Clases del Dominio

6.2 Descripción de las Clases del Dominio

Administrador: Encargado de la creación de mesas, eliminación y mantenimiento de los usuarios así como de la publicación de Noticias y demás aspectos de mantenimiento del Sistema.

Apuesta: Cantidad de puntos que pone en juego el cliente al recibir la primera carta, en función de lo que considere que vale.

Baraja: Conjunto de todas las cartas que se pueden repartir a los jugadores de una mesa.

Carta: Elemento con un número y un palo que es repartido a los jugadores al principio de la partida y durante su turno.

Crupier: Inteligencia artificial de la aplicación que se reparte cartas y compite contra los clientes.

Jugador: Cada uno de los usuarios que están en una mesa, pueden ser jugadores activos o mirones dependiendo de si están o no sentados en un asiento.

Líneas Mano: Cada una de las líneas que componen una mano con la carta correspondiente.

Mano: Conjunto de cartas que se van a utilizar para decidir el ganador de una partida, la mano es de un jugador en concreto y para una partida en concreto.

Mesa: Conjunto de asientos para la ejecución de partidas, cada mesa tiene sus jugadores que no pueden estar en otras mesas simultáneamente. En la mesa también pueden estar los mirones para observar la partida.

Mirón: Jugador no activo de una partida, puede observar el transcurso de la misma y sentarse en un asiento en caso de que haya alguno libre. Los mirones ven las manos tal y como ven los jugadores activos las manos de sus contrincantes.

Noticia: Novedad publicada en la web para informar a los usuarios de un suceso o cambio.

Partida: Cuando en una mesa de juego se encuentra un jugador activo y se reparte la primera carta, es cuando se inicia una partida, entonces con esta clase nos referimos a todo lo que conlleva desde su inicio hasta cuando el crupier se ha repartido y se ha calculado quien o quienes son los ganadores.

Ranking: Posición relativa de un usuario con relación al resto y en orden de puntos.

Usuario Registrado: Cliente que ha completado correctamente el formulario de registro de la aplicación.

7. Diseño

En este apartado se van a mostrar los diagramas realizados y su explicación correspondiente.

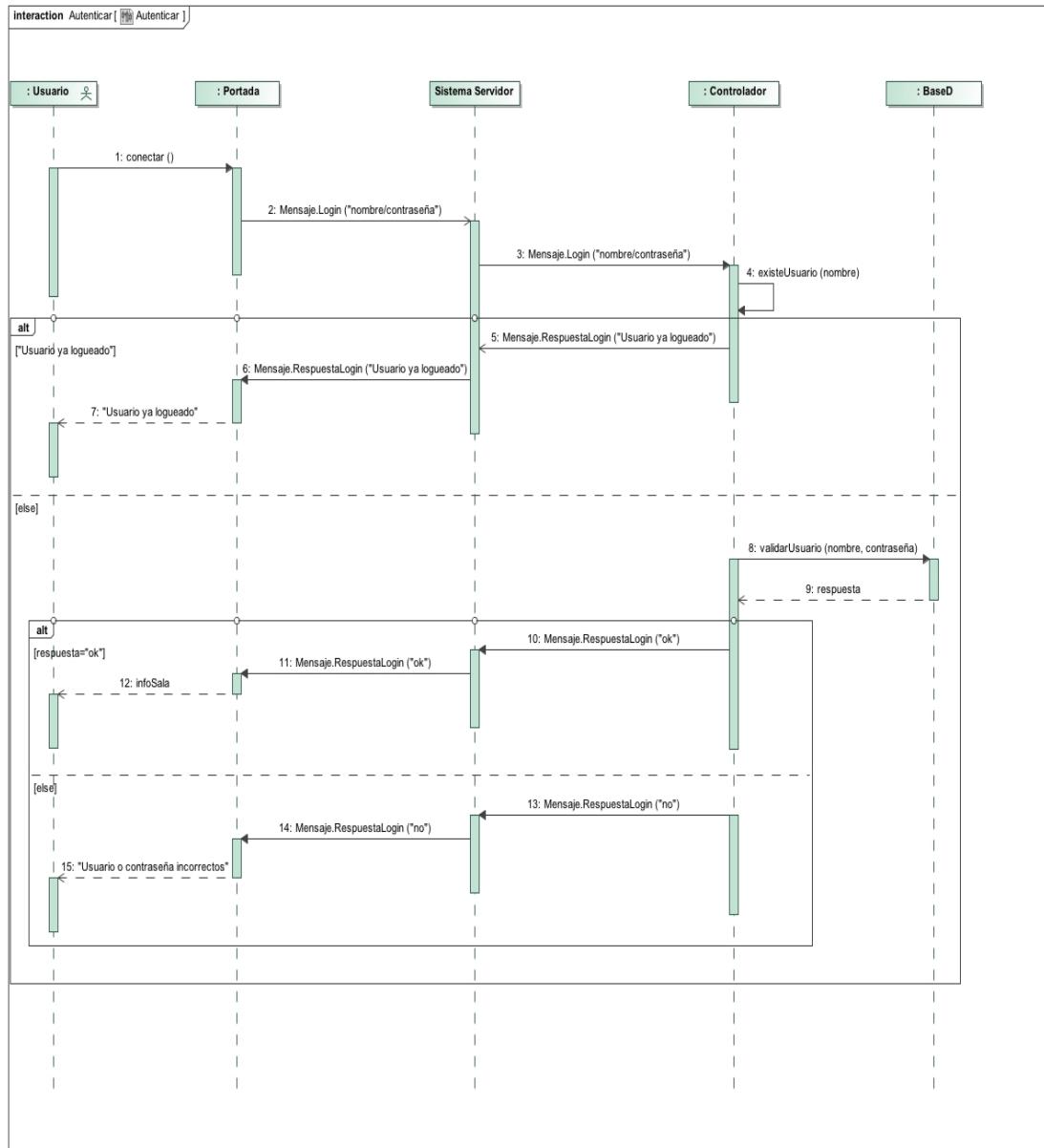
7.1 Diagramas de Secuencia

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones.

7.1.1 Autenticarse

En él podemos observar lo que sucede dentro del sistema cuando un usuario registrado realiza el caso de uso de autenticarse, es decir, introduce sus datos de usuario para conectarse al servidor, viendo los posibles errores que pueden suceder .

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

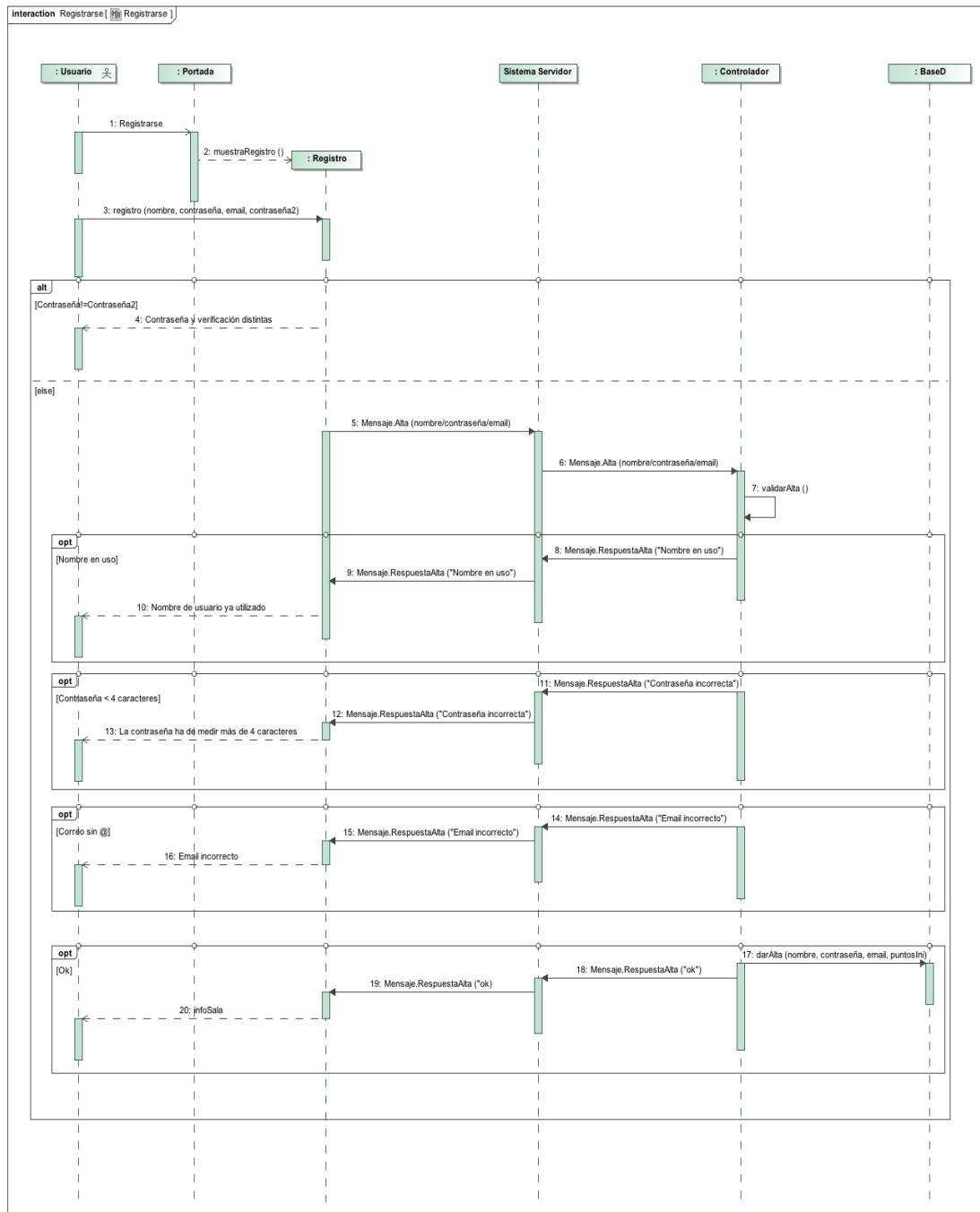


10 Diagrama de Secuencia -Autenticarse-

7.1.2 Registrarse

En este diagrama de secuencia observamos lo que sucede dentro del sistema cuando un usuario accede a la opción Registro en la portada de la interfaz cliente y como se procesa, como consecuencia obtendrá una vista de la sala ya logueado o bien un mensaje indicando que campo ha introducido incorrectamente en el registro.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

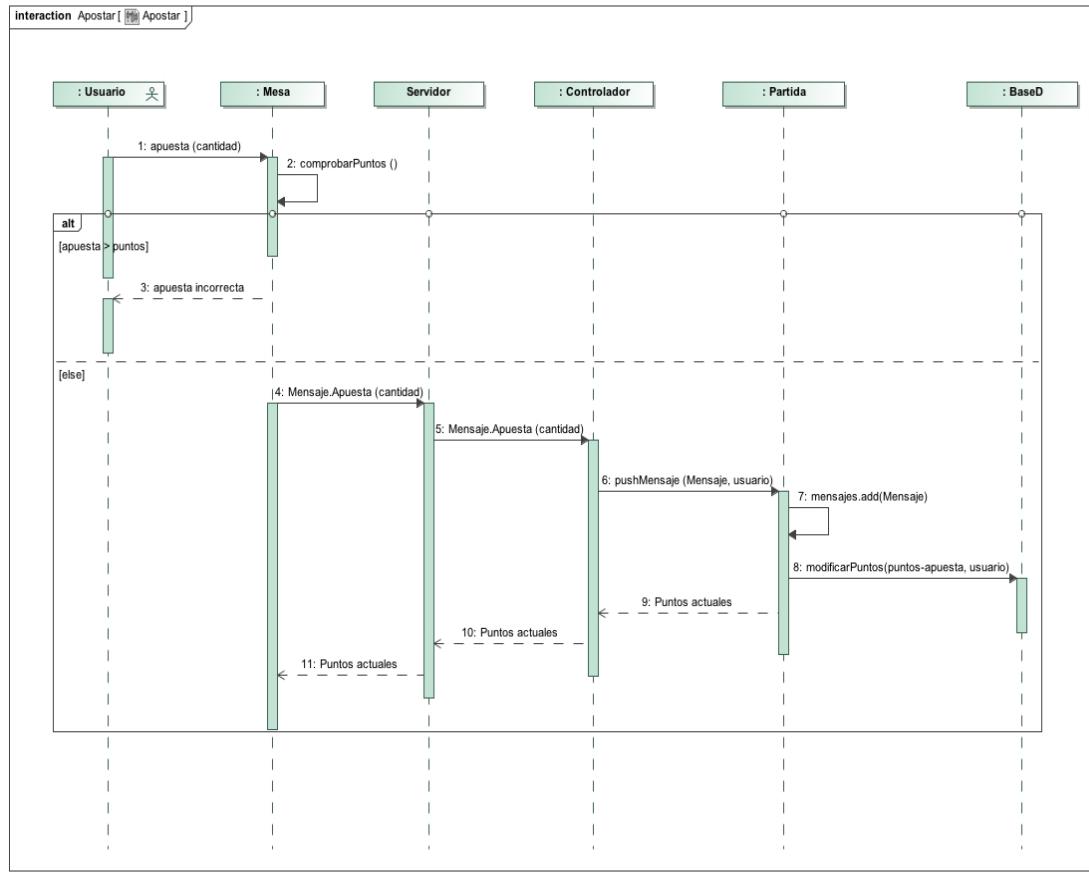


11 Diagrama de Secuencia -Registrarse-

7.1.3 Apostar

Muestra la sucesión de eventos necesaria para que un Jugador realice una apuesta a la hora de entrar en una mesa, también se recibe un error en caso de que un jugador quiera apostar más puntos de los que tiene.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

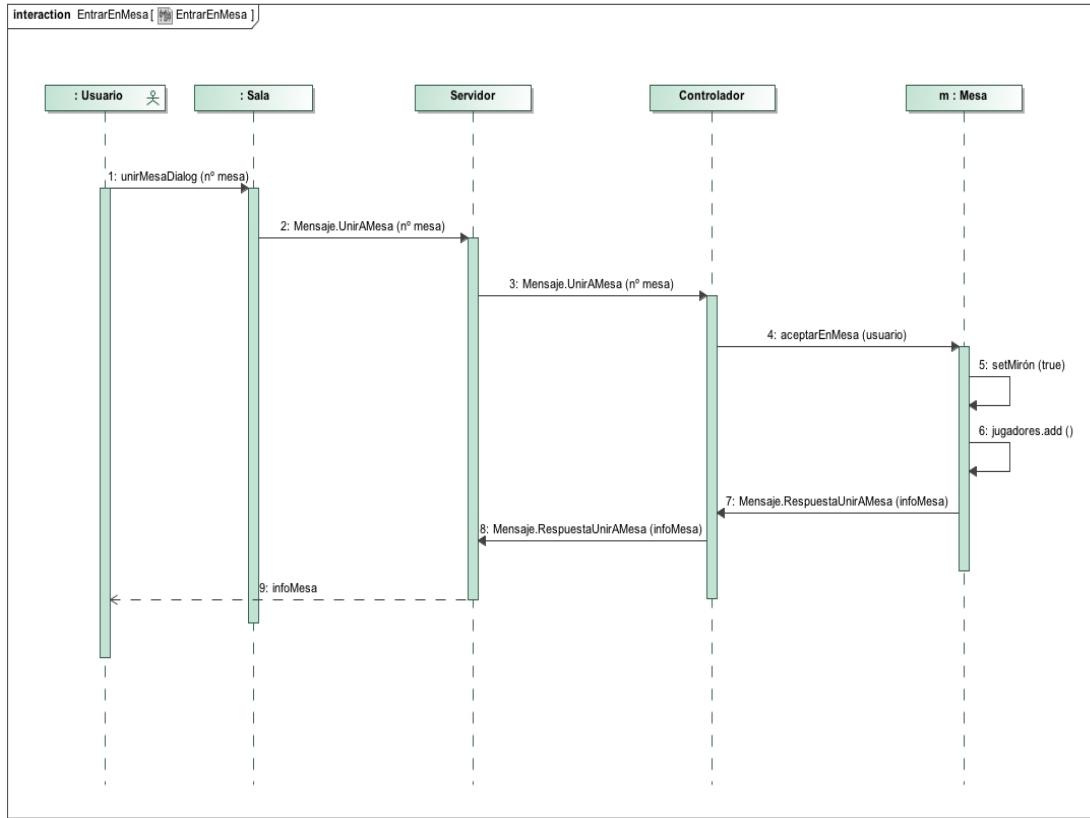


12 Diagrama de Secuencia -Apostar-

7.1.4 Entrar en Mesa

Este DS muestra la sucesión de eventos que ocurren tanto en el cliente como en el servidor a la hora de unir un jugador a la mesa, se decidió que todo jugador que no estuviera en otra mesa pudiera entrar en la que quisiera como Mirón y una vez allí poder sentarse en caso de haber sitio.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

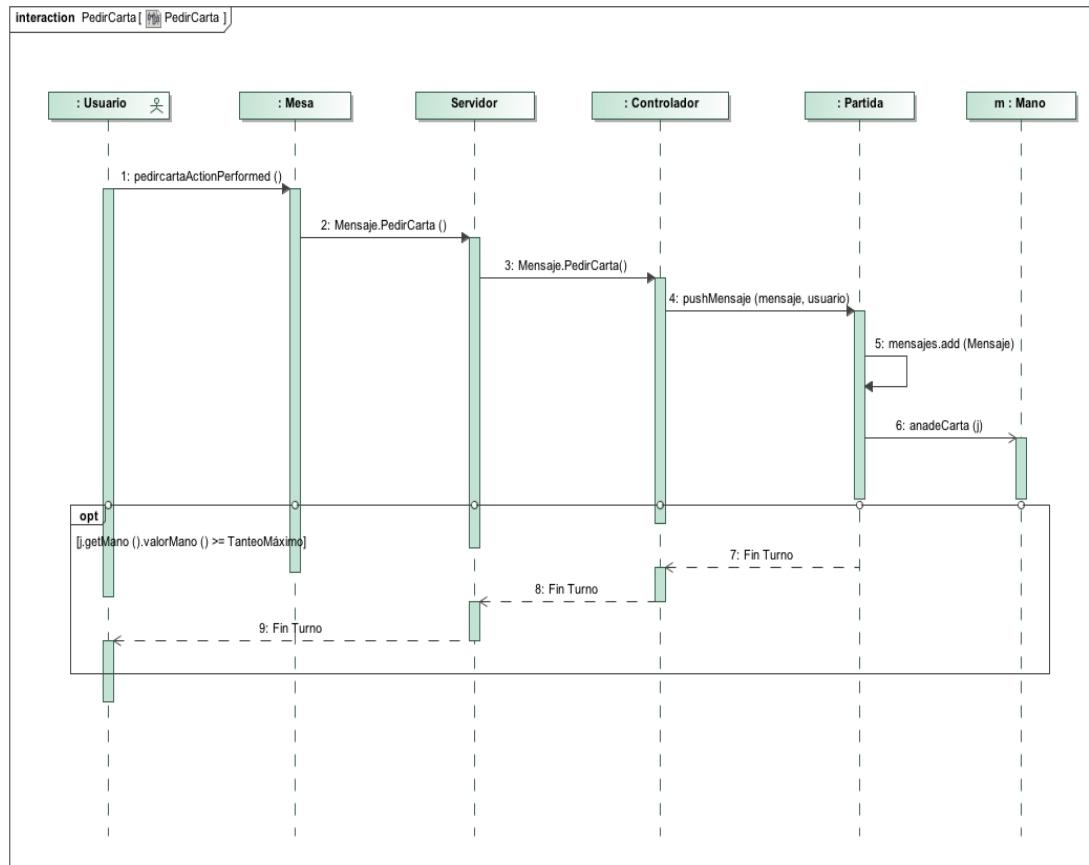


13 Diagrama de Secuencia -Entrar en Mesa-

7.1.5 Pedir Carta

En este DS se muestra el proceso de petición de carta cuando un jugador está en turno, también se refleja el flujo alternativo que ocurre cuando el jugador alcanza o pasa de 21 puntos, ya que en ambos casos se termina su turno.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

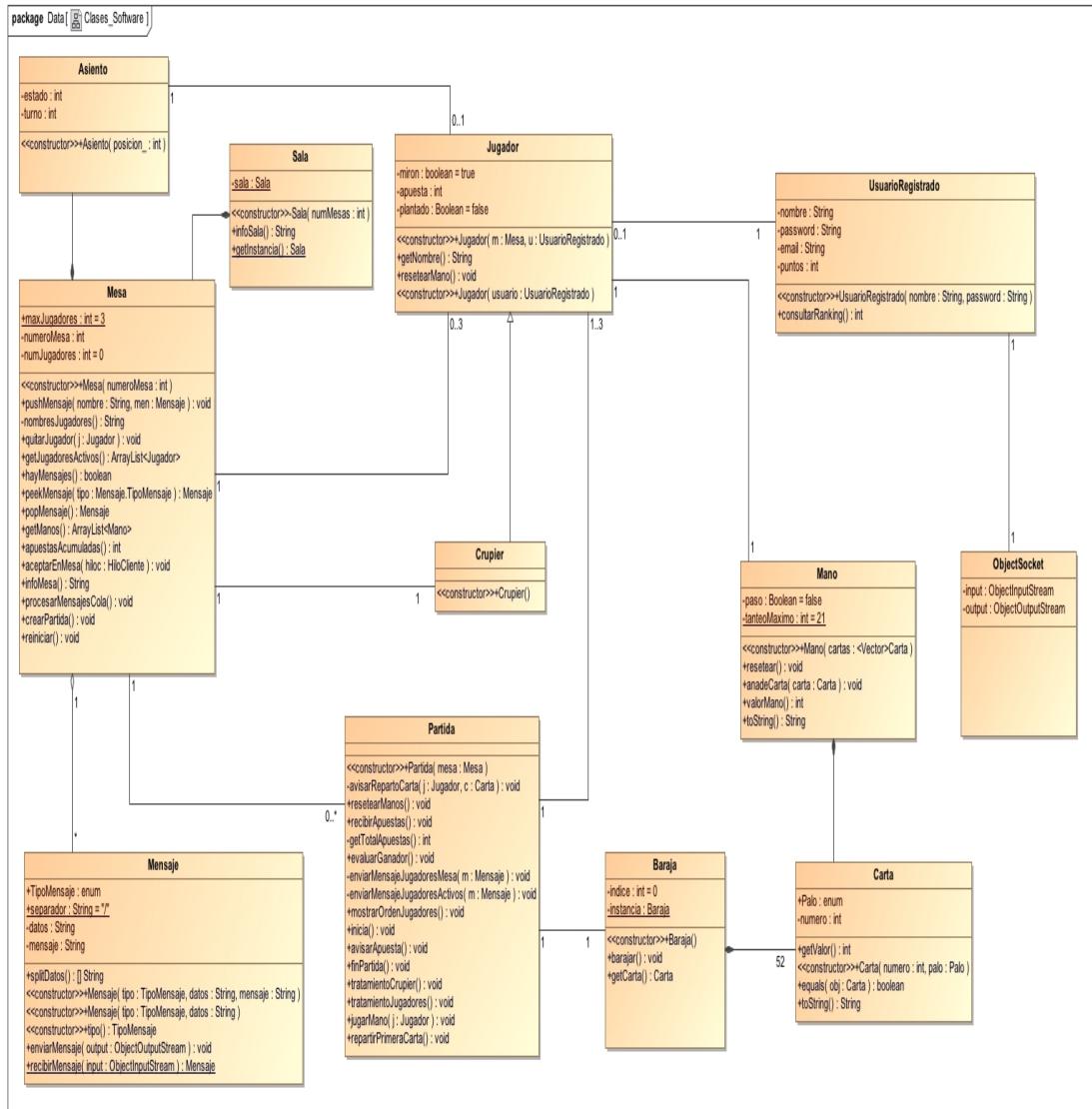


14 Diagrama de Secuencia -Pedir Carta-

7.2 Diagrama de Clases Software

Una vez expuesto el diseño dinámico del proyecto, continuamos con el diseño estático. El diagrama de clases software, que representa las diferentes clases que vamos a implementar y en las que se marcan todos sus atributos y sus métodos más importantes.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



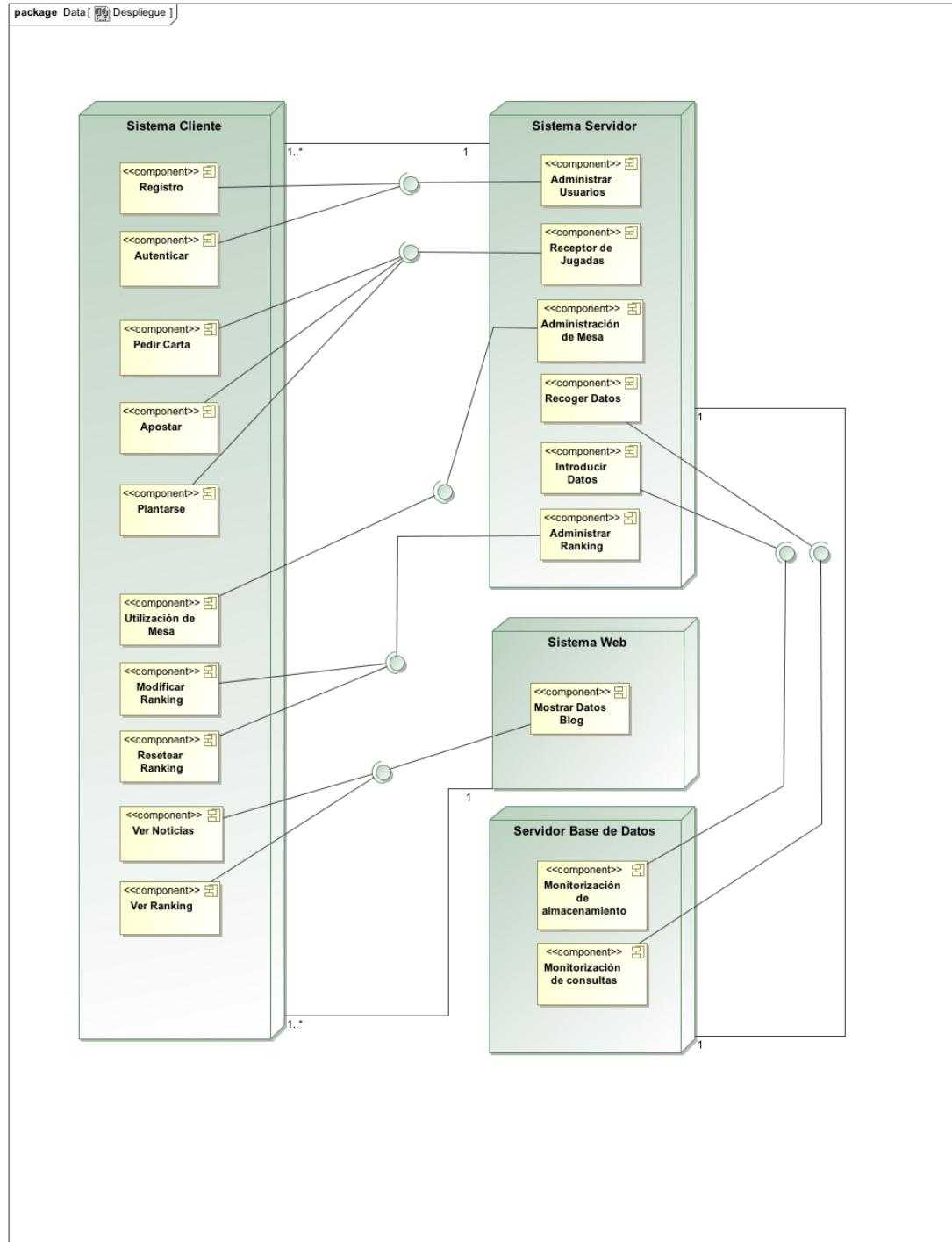
15 Diagrama de Clases Software

7.3 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Los elementos usados en diagramas del despliegue son nodos (demostrados como cubo), componentes (demostrados como caja rectangular, con dos rectángulos resaltando del lado izquierdo) y asociaciones.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



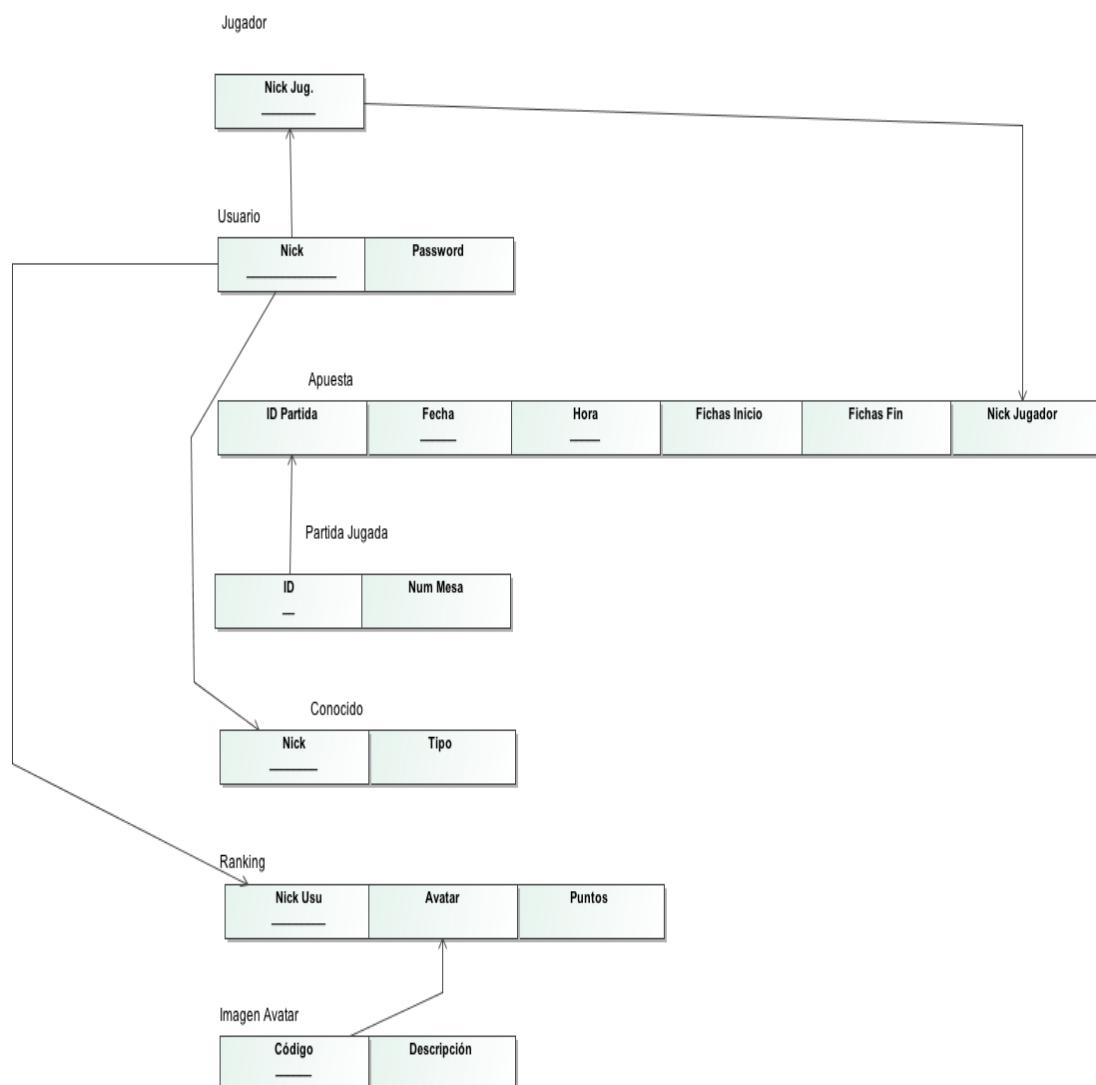
16 Diagrama de Despliegue

7.4 Diagrama relacional de la Base de Datos

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

El diagrama considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



17 Modelo Entidad-Relación

8. Implementación en las fases del proceso de desarrollo

En este apartado se explica el desarrollo e implementación de cada uno de los incrementos realizados, recordando que se ha utilizado el Proceso Unificado iterativo e incremental. Como tal es posible que alguno de los incrementos ya refleje el camino a la implementación de las líneas futuras.

8.1 Primer incremento

El primer incremento se basa en la captura de requerimientos del problema, en el análisis de los mismos y en una primera aproximación al diseño. En esta fase se completaron el resumen, los diagramas de casos de uso, clases conceptuales y DSS.

Así mismo se comenzó a realizar los Diagramas de Secuencia de bajo nivel para hacer una aproximación a como se implementaría la aplicación.

A pesar de sufrir variaciones a lo largo de los incrementos esta fase marcó el objetivo del proyecto y las líneas futuras del mismo.

8.2 Segundo incremento

En este incremento se continuó realizando el diseño de la aplicación completando los DS de bajo nivel y el diagrama de clases software, también se diseño el diagrama de colaboración.

Además de revisar los requerimientos y el análisis del primer incremento se realizó la parte básica central del proyecto, que incluía simplemente una mesa en la que se podía sentar un único jugador para realizar partidas contra el servidor.

El usuario apostaba cualquier cantidad positiva al recibir una cadena de texto que incluía el número y el palo de la carta, una vez que se pasaba o finalizaba la mano el crupier se repartía sus propias cartas. Al terminar se le daban más fichas al usuario para que pudiera seguir jugando.

Al entrar en la mesa el usuario recibía una cantidad de fichas de manera que si salía de la mesa o de la aplicación volvía a recibir ese mismo número de fichas.

Durante este incremento el servidor no tenía la posibilidad de acoger a más jugadores durante una partida.

8.3 Tercer incremento

Además de revisar la captura de requerimientos, el análisis y el diseño, en el cual se modificó el diagrama de clases software incorporando los Hilos que derivan de la clase Thread de Java, se incorporó la posibilidad de conectar a más de un jugador en una mesa, así como la incorporación de Threads al proyecto.

A su vez se comenzó la implementación de la BD la cual sólo constaba de la clase Usuario para incorporar la posibilidad de loguear en el siguiente incremento.

Con este incremento se comenzó la aproximación a la fase de pruebas, comprobando que se realizaban correctamente las inserciones en la BD y que funcionaba correctamente el Driver de JDBC.

8.4 Cuarto incremento

A partir de este incremento se convierte en tónica constante la revisión de los requerimientos, el análisis y el diseño, convirtiendo el proceso de implementación del proyecto en un verdadero proceso unificado, cabe destacar la comprobación que se ha hecho de que a medida que se avanza en los incrementos disminuye el análisis y aumentan la fase de implementación y la de prueba.

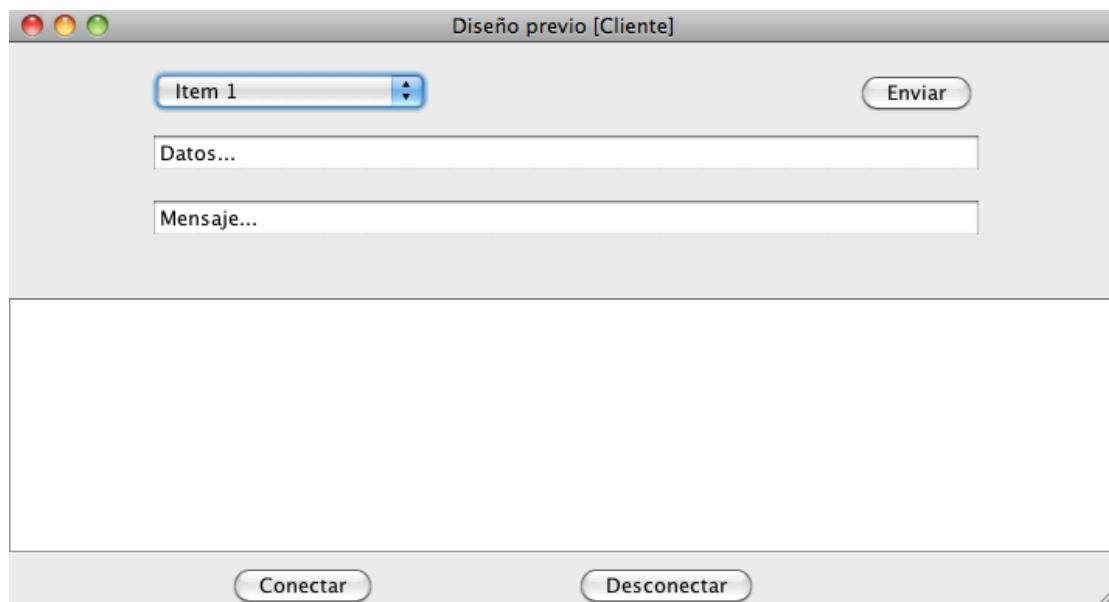
Este incremento supuso un avance notable en la implementación global del proyecto, al incorporar el inicio de la fase de diseño gráfico utilizando MVC .

Con este incremento se comenzó a utilizar la BD con JQBCD para el login de los usuarios, los cuales recibian un mensaje afirmativo en caso de loguear y uno negativo en caso de no conseguirlo, en caso negativo el usuario podía volver a intentar el logueo o registrarse.

En este incremento también se incorpozo el desarrollo del proyecto que incorpora los mensajes, encargado de poner en comunicación a cliente y servidor que saben de esta manera como responder a solicitudes.

En la interfaz se seleccionaba el tipo de mensaje que se queria enviar y se introducian manualmente los datos.

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



18 Cliente primitivo

8.5 Quinto incremento

En este incremento se introdujo la posibilidad de seleccionar entre varias mesas y de salir de mesa y entrar en otra durante la duración de una sesión.

Asimismo se introdujo la notificación de las cartas destapadas al resto de usuarios conectados en la mesa y los mensajes indicando quien era el ganador de la mesa.

También en este incremento se introdujo la limitación de puntos en las apuestas.

8.6 Sexto incremento

En este incremento se abandona la interfaz de prueba y se crea la interfaz cliente que combina la introducción de mensajes en el cliente para no tener que introducirlos manualmente.

También se crea la opción de registro que incluye una cantidad de números iniciales, desde este momento a la hora de hacer un registro se comprueba que el email contenga una '@', que el nombre de usuario y la contraseña tengan un tamaño adecuado y también que el login no esté ocupado por otro usuario previo.

A partir de este incremento se comienzan a guardar las partidas con sus datos de ganador, jugadores, fecha y número de mesa en la BD.

8.7 Séptimo incremento

En este incremento se incluye la posibilidad de entrar como mirón a una mesa, en este caso los mirones ven las bazas como los jugadores ven las de sus rivales, a partir de este momento todos los usuarios entran a una mesa como mirones y tienen la posibilidad de sentarse que devuelve un mensaje explicativo de si se ha podido sentar o en caso de estar llena la mesa, un mensaje explicando que no hay En asientos libres.

Con estas variaciones se modificó el apartado de análisis y diseño para adaptarlo a las nuevas características, como por ejemplo, la inclusión de un nuevo actor (El Mirón).

8.8 Octavo incremento

En el octavo incremento se mejoraron ciertos aspectos menores de la aplicación, como fueron los tiempos entre partida o la velocidad con la que el crupier se repartia sus cartas.

En este incremento se incorporó el diseño de la Sala y se adaptó a el una lista con el ranking de los 10 primeros jugadores ordenados por puntos y con sus puntos correspondientes.

En la información de la mesa a la que unirse se indica también cuales son los usuarios activos para la partida.

Esta información se refresca cada partida o cada vez que el usuario accede a la Sala. (Al loguearse o al salir de una mesa)

En este incremento a pesar de revisar las tres primeras etapas del proceso unificado ya no ha sido necesario modificar ni la captura de requerimientos, ni el análisis ni el diseño.

8.9 Noveno incremento

En el noveno incremento se finalizó el diseño de las mesas, se incluyeron los asientos y las cartas, se introdujo también una variación a la hora de establecer el turno, para que no se tuviera ventaja se decidió que a la hora de iniciar la partida se establecerían los turnos aleatoriamente.

Se han añadido consultas para saber cuantas partidas ha ganado un jugador, cuantas ha perdido, el número de puntos que ha ganado y los usuarios contra los que juega habitualmente, esto se ha realizado para futuras líneas como serán los amigos y enemigos. Así mismo este incremento se centró en la prueba de casos críticos para la aplicación para ver como respondía correctamente a todas las pruebas.

9. Pruebas

Las pruebas de software consisten en procesos que permiten validar y verificar un sistema, debe hacerse énfasis en las mismas durante las primeras fases del ciclo de vida porque cuanto antes se detecten los errores menos costosa es su corrección.

Debe aclararse que las metodologías de pruebas están orientadas a probar la existencia de errores y no la ausencia de ellos, lo que se persigue es detectar el mayor número de ellos posibles en la fase de diseño/desarrollo para subsanarlos cuanto antes.

En lo que respecta a este proyecto, se han realizado pruebas de unidad de cada una de las operaciones posibles dentro de la aplicación cliente, lo que se refleja en cada uno de los métodos del servidor, con la ayuda de la herramienta libre JUnit, programada y orientada a la realización de pruebas de unidad en lenguaje Java.

Para probar una clase con JUnit se crea otra con un método por cada uno de la clase a probar, estos métodos de test llevarán la anotación @Test y contendrán el código necesario para la cobertura de los casos de prueba requeridos. Un método de prueba será superado satisfactoriamente cuando no lance una excepción no esperada y cumpla las aserciones incluidas en su código (assertEquals(), assertFalse()...).

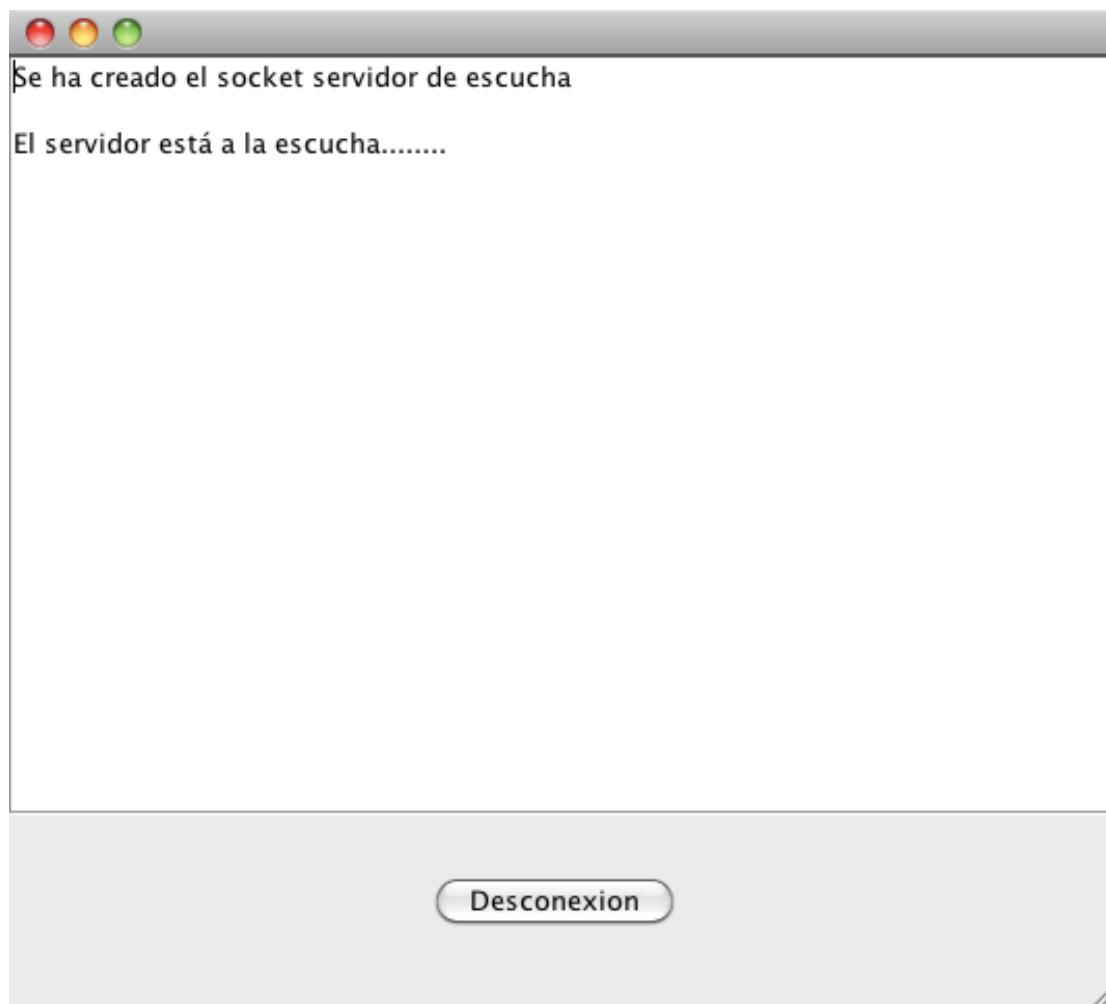
Como complemento a los test de unidad, se han realizado unas pruebas poniendo la aplicación accesible a un reducido grupo de personas que se han registrado e interactuado con el sistema desde los distintos sistemas operativos ya que la aplicación ha sido realizada desde Mac OS X Snow Leopard.

La finalidad de esta última prueba ha sido la validación del sistema como conjunto y el acercamiento a una situación de uso real.

Este último tipo de pruebas responden al nombre de pruebas de integración, se han realizado a través de internet donde los diferentes usuarios en sus respectivas máquinas han ejecutado la aplicación cliente y todas se han conectado a la misma aplicación servidor. Estos usuarios realizaron todas las funciones posibles que ofrece la aplicación, y se han comprobado las diferentes respuestas del servidor.

10. Manual

El manual de uso de la aplicación corresponde a la aplicación cliente, la aplicación servidor simplemente es una caja de texto que refleja mensajes para el administrador como podemos ver a continuación.



[19 Servidor](#)

10.1 Login

Al ejecutar la aplicación tanto desde su ejecutable como desde cualquier Software de programación, lo primero que veremos es la portada :



20 Cliente -Login-

La portada consta de varias partes, un campo para introducir el nombre del usuario, un campo para introducir la contraseña del usuario (de manera que no se muestra lo que está escribiendo el usuario), el botón

para conectarse que valida los datos e introduce al usuario a la Sala, el botón de Registro para que el usuario pueda registrarse en caso de no haberlo hecho antes y por último el menu donde encontramos la pestaña archivo que consta de 2 opciones que detallaremos más adelante.

En la siguiente imagen podemos ver el error típico que recibiríamos en caso de introducir mal algún dato a la hora de autenticarnos :



21 Cliente -Login Error-

Otro error tipo que también se recoje en el cliente sería intentar acceder estando logueado desde otro cliente, en este caso el error sería el siguiente :



22 Cliente -Login Error 2-

10.2 Registro

En caso de que el cliente no esté registrado en la aplicación optaría por pulsar el botón de Registro, en este caso se encontraría con la siguiente ventana :



[23 Cliente -Registro-](#)

Como podemos observar en esta captura de pantalla, la pantalla de Registro consta de 4 campos y 2 opciones, el primer campo “Nombre”, ser

refiere al nombre por el cual el usuario quiere ser identificado dentro de la aplicación, este nombre ha de tener una longitud adecuada y no estar ya registrado en la aplicación.

El segundo campo “E-Mail”, es el correspondiente al correo electrónico del usuario, como los demás ha de ser cumplimentado y uno de los requisitos es que nos encontremos con una @.

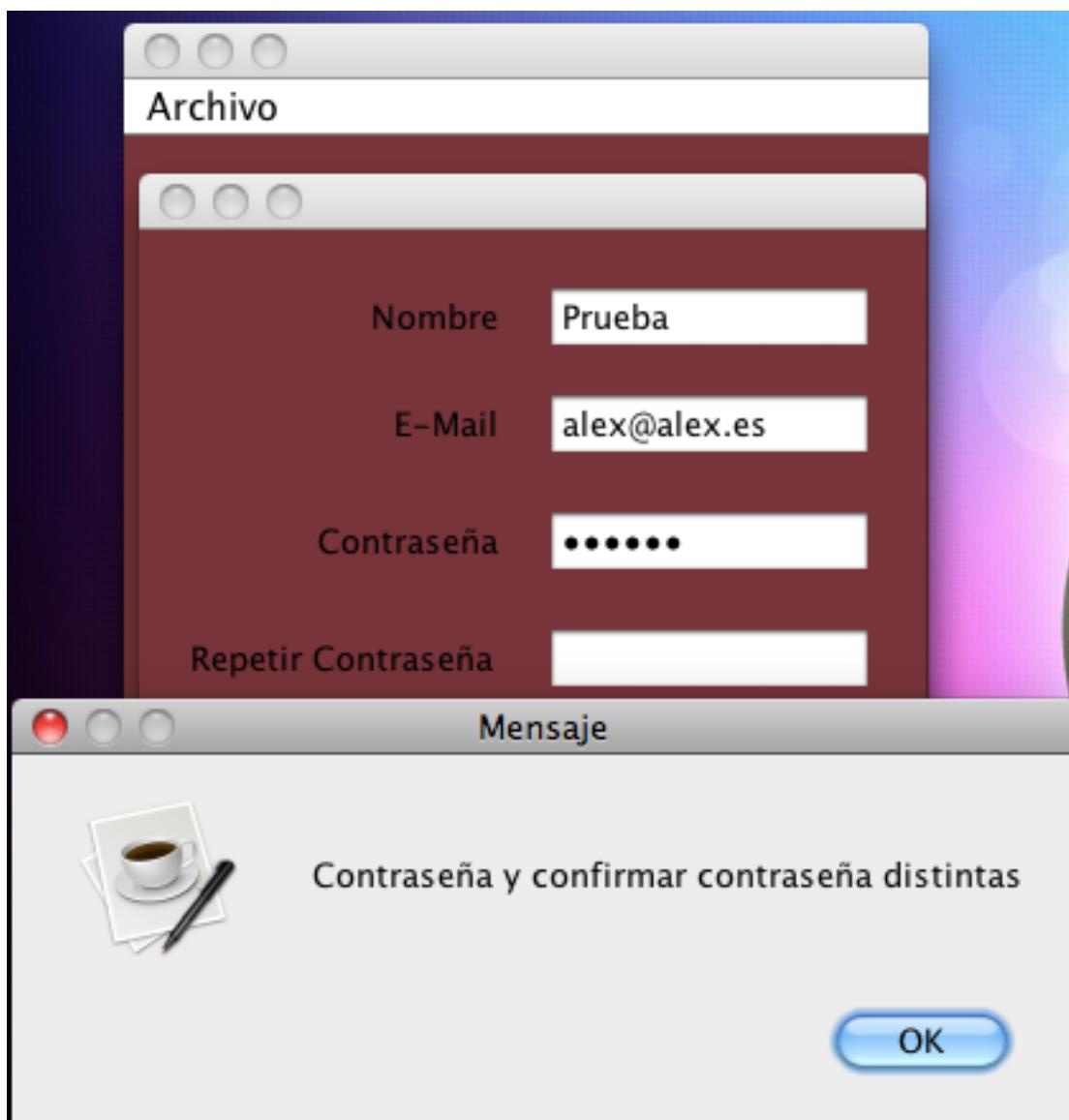
El tercer y cuarto campo corresponden a la contraseña y a la confirmación respectivamente, han de coincidir y tener un tamaño mínimo de 4 caracteres. El botón registro sirve para confirmar los datos.

Como ya hemos comentado el nombre de usuario ha de estar libre, este es un ejemplo del error que recibiríamos si intentamos registrar otro usuario con el mismo nombre :



24 Cliente -Registro Error-

Otro de los errores que podríamos recibir a la hora de registrarnos en la aplicación sería cuando no coinciden la contraseña y el campo de verificación de contraseña :



25 Cliente -Registro Error 2-

En caso de querer cancelar el Registro tenemos la opción en el botón “Cancelar” .

Una vez que un usuario completa correctamente los datos pasa directamente a la Sala.

10.3 Acceso a la aplicación

Una vez que un usuario se identifica correctamente en la aplicación lo primero que observará es la siguiente ventana:



26 Cliente -Sala-

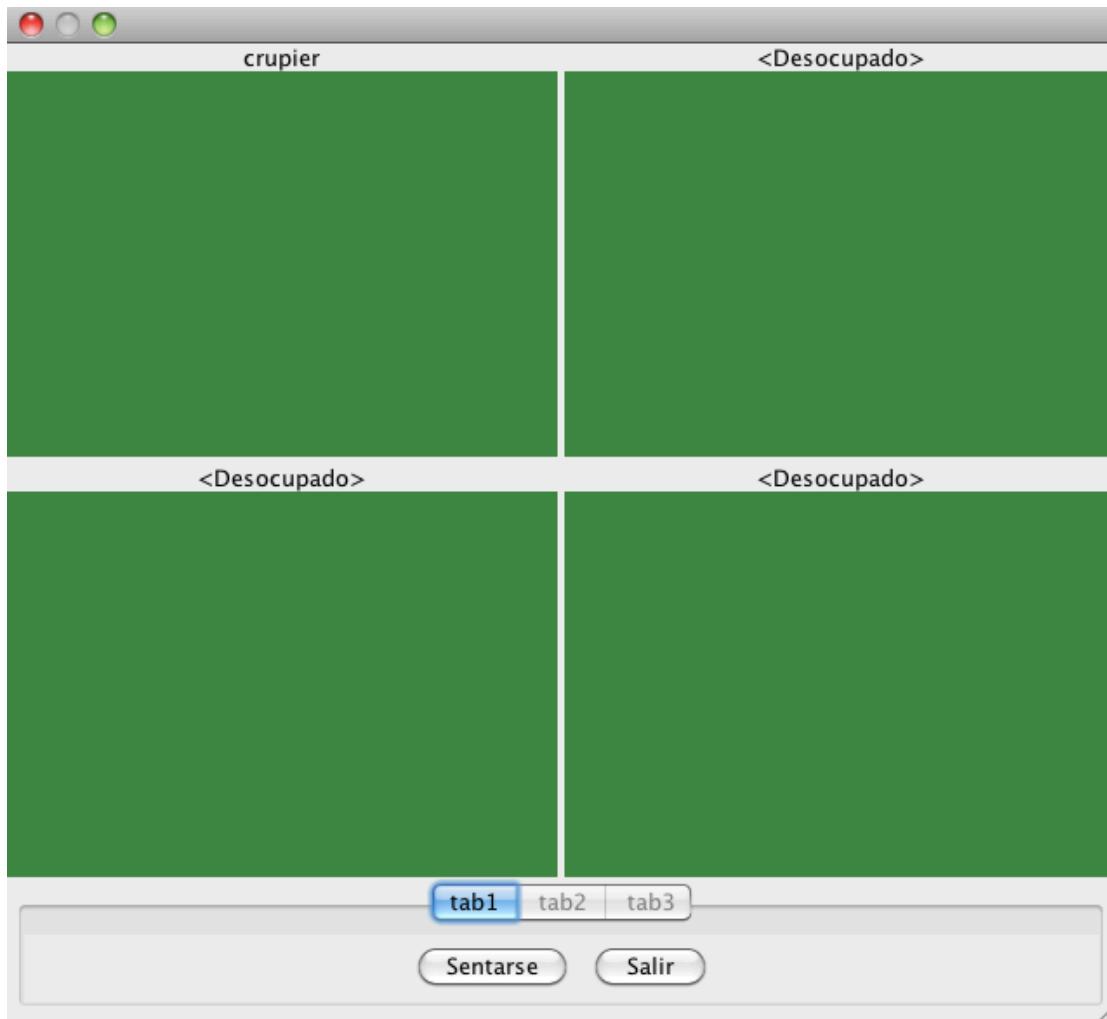
En esta ventana podemos observar varios campos y opciones, en la parte derecha se muestra el top 10 de los jugadores ordenados por puntos y con su nombre de usuarios, es decir, el ranking de los 10 mejores, este ranking se actualiza cada vez que el usuario accede a la sala, bien al loguear, bien al abandonar una mesa.

Otra de las opciones de la ventana sala es la de desconectar, el botón que se muestra debajo de la ventana da esta opción, una vez que el usuario se desconecta, cierra la aplicación y se desloguea automáticamente.

Por último y también la más importante, es la opción de unirse a mesa, el usuario puede ver en la parte izquierda de la sala la lista de las mesas, en esta lista se muestra el número de la mesa y los jugadores activos en ella (los mirones no aparecen en la lista).

10.4 Sentado en Mesa

Una vez que el usuario está sentado en la mesa lo primero que observará es la partida en curso, cuando un usuario se une a una mesa lo hace automáticamente como mirón :



[27 Cliente -Mesa-](#)

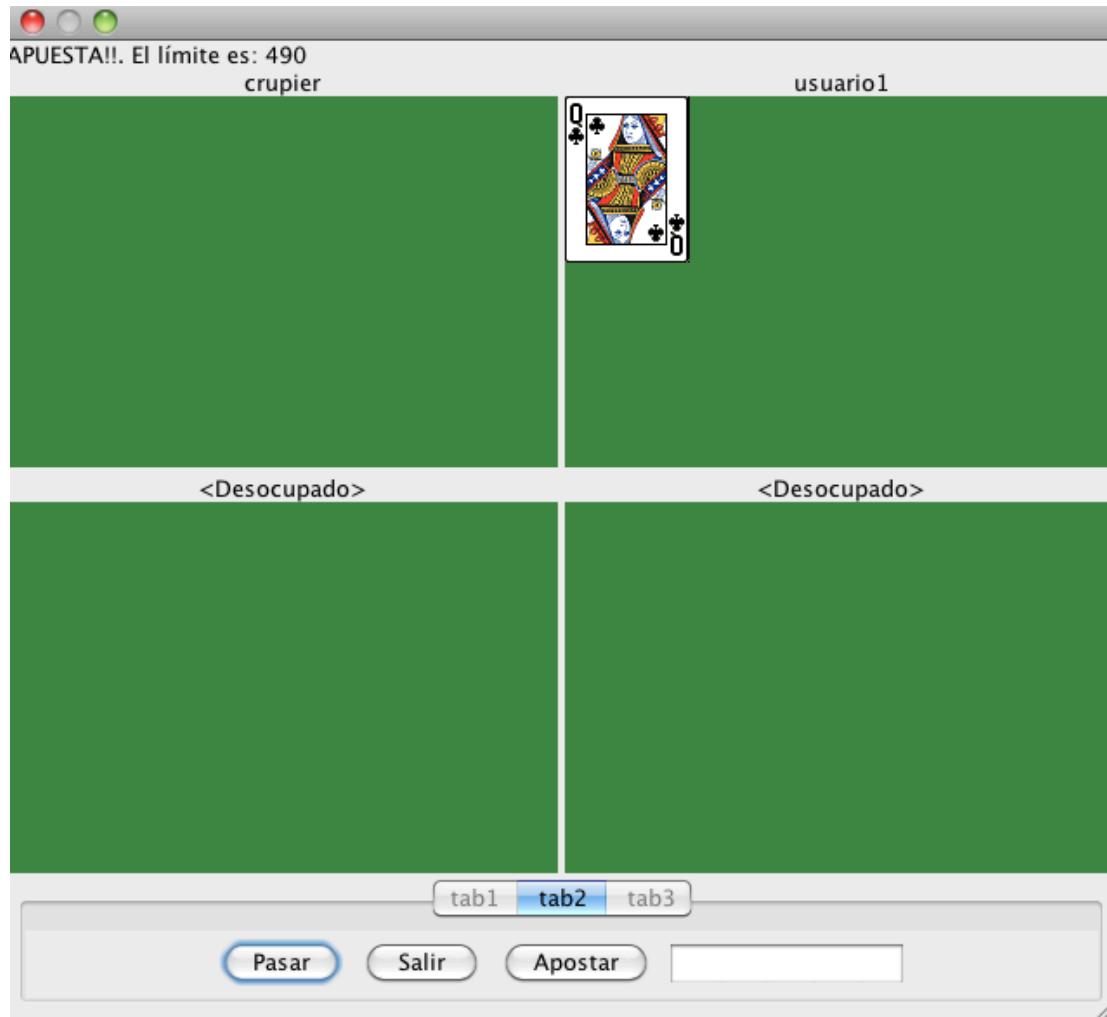
Al estar logueado como miron ve lo mismo que los usuarios que están jugando y no están en su turno, es decir, puede ver las cartas de los

contrincantes pero la primera se le muestra tapada.

A diferencia de los jugadores el Mirón solo tiene la opción de Salir o Sentarse, en caso de Salir volverá a la Sala y en caso de Sentarse recibirá un mensaje cuando la mesa este llena, en otro caso se sentará para el comienzo de la siguiente partida.

Cuando un usuario se sienta y no hay nadie en la mesa compite solamente contra el Crupier, la partida no inicia hasta que un usuario se sienta como jugador activo.

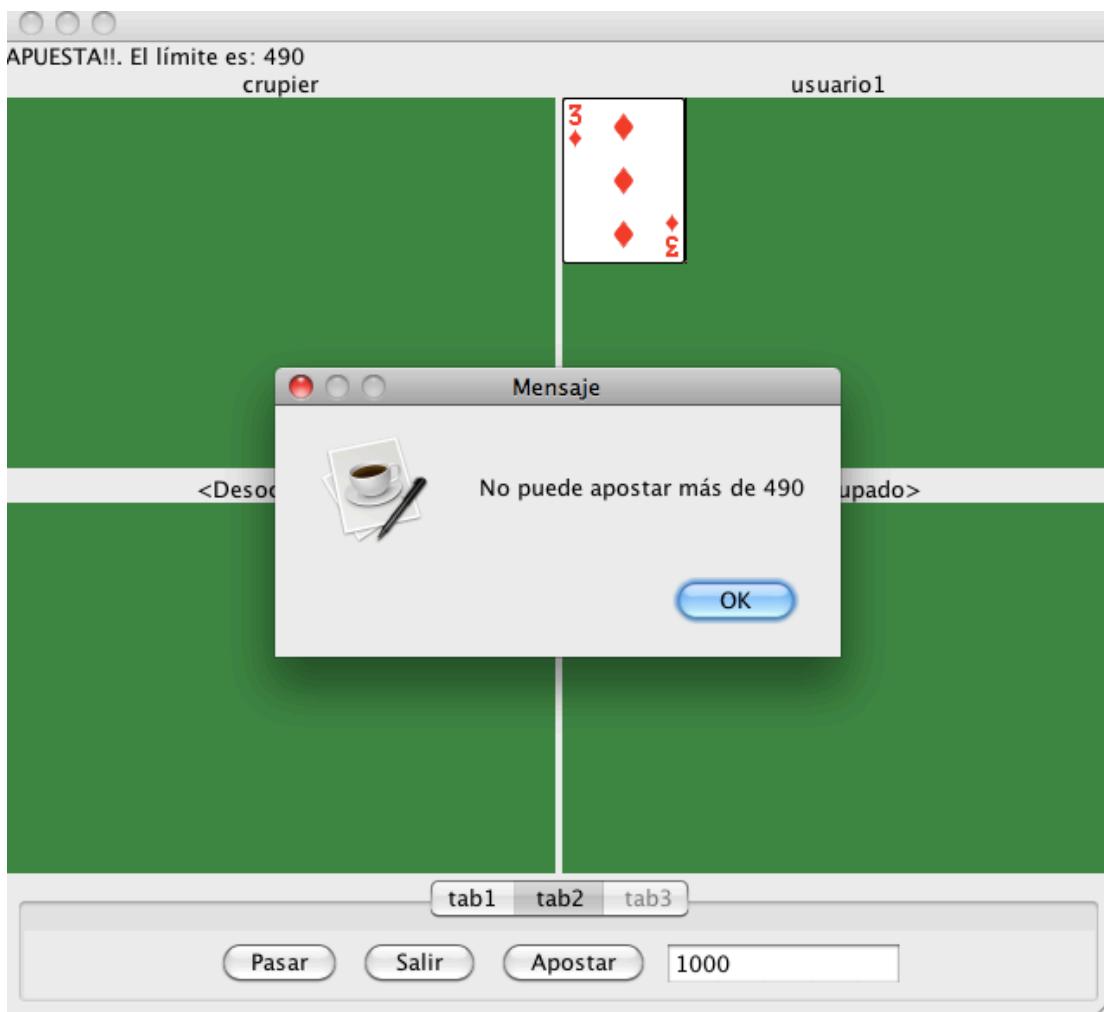
Una vez sentado y en turno (o directamente al sentarse si no había nadie en la partida), lo que verá es lo siguiente :



28 Cliente -Mesa 2-

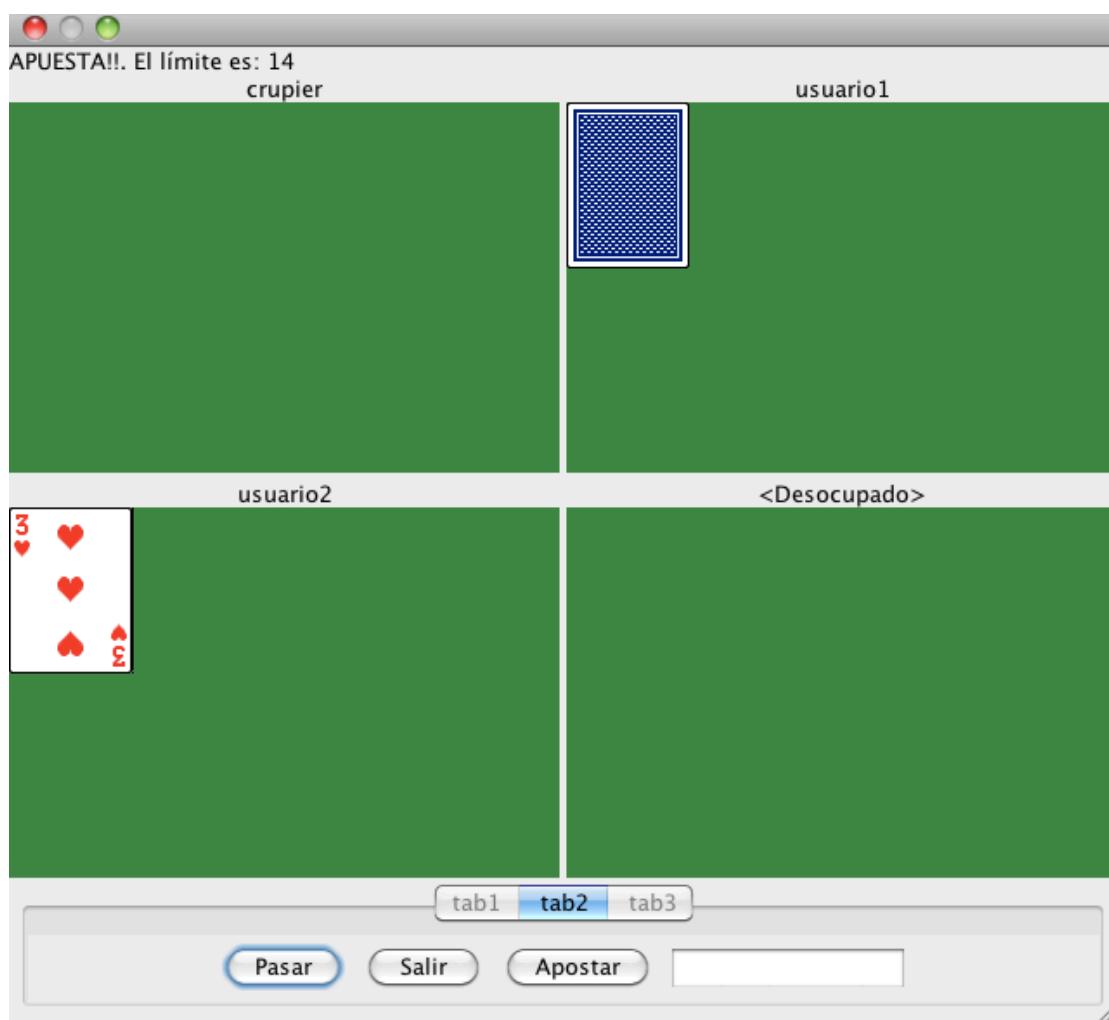
Como podemos observar ahora el jugador ve su carta y tiene la opción de Pasar (no apostar y no jugar en esta partida), Salir (Para volver a la Sala) y Apostar, como se ve arriba en el cuadro de mensajes del cliente, se indica al jugador que es su turno de apuesta y la cantidad máxima que puede apostar.

En caso de apostar más de lo que puede recibirá un mensaje conforme a que no es posible apostar esa cantidad, como vemos en el siguiente mensaje :



29 Cliente -Mesa 3-

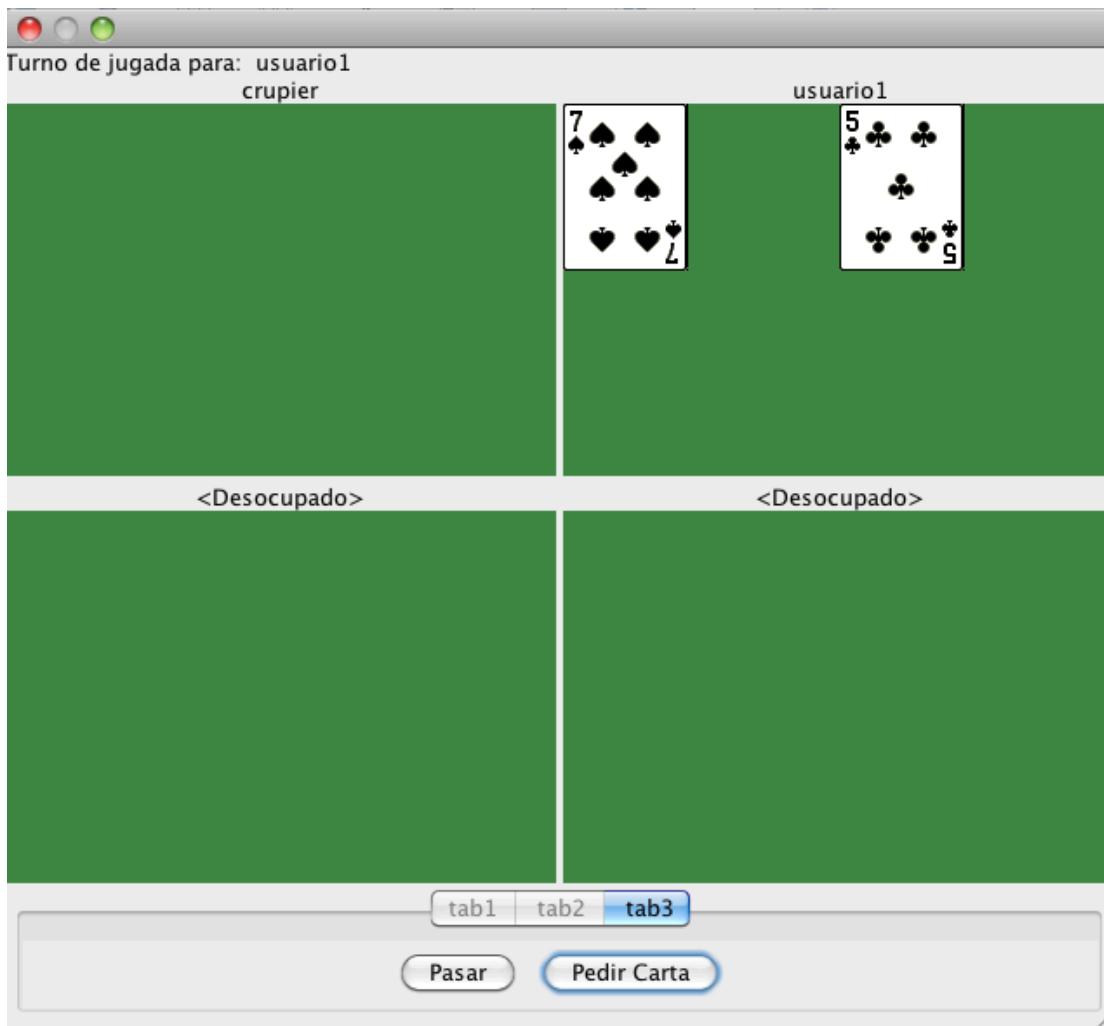
A continuación se muestra una captura de pantalla de cómo ven los Mirones la partida y los jugadores las manos de sus rivales :



30 Cliente -Mesa 4-

En este caso concreto el usuario2 es el jugador desde el que vemos la mesa, y el usuario1 su rival.

Cuando el jugador está en turno la pantalla que ve es la siguiente:



31 Cliente -Mesa 5-

En esta pantalla las opciones son 2, o pasar para finalizar el turno y dar por buena la mano que se tiene o bien pedir otra carta, en caso de obtener 21 puntos o más se termina la mano del jugador y se pasaría al siguiente.

Para preservar la igualdad en el juego aunque los asientos se distribuyen por orden de entrada, los turnos son aleatorios en cada partida.

Una vez terminada la partida se nos avisa en el cuadro de texto del resultado de la misma.

10.5 Barra de tareas

La barra de tareas que podemos observar en el menú Portada, nos dá dos opciones, o bien abandonar la aplicación (Botón Salir) o bien seleccionar la IP del servidor en caso de que no estemos ejecutando aplicación y servidor en un entorno local, para poder utilizar la aplicación desde otra ubicación, como el resto de funcionalidades está también ha sido probada desde otros equipos en la misma red y desde otros equipos en otras redes.



32 Cliente -Barra de Tareas-

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack



33 Cliente –Barra de Tareas 2-

10.6 Contenido del CD

Proyectos Netbeans con el código del PFC

Memoria en formatos .doc, .pdf e imágenes utilizadas.

Proyecto de MagicDraw con todos los diagramas UML realizados.

11. Requisitos y tutorial de instalación

Antes de pasar a la explicación de la instalación del diverso software destacar que pese a que la compatibilidad y la variedad de programas que tiene Mac OS X Snow Leopard, la cantidad de tutoriales o ayudas para el desarrollo de este proyecto en términos software ha sido muy inferior al material del que se habría dispuesto para Windows.

11.1 Instalación de Java

La instalación de Java para un usuario de Mac OS es tan sencilla como mantener al día las actualizaciones de software ya que se instala automáticamente con el Sistema Operativo.

En caso de necesitar Java para cualquier otra plataforma bastaría con visitar :

<http://www.java.com/es/download/manual.jsp>

Una vez se tenga instalado correctamente java con los requisitos predeterminados ya se podría utilizar la versión cliente del programa.

11.2 Instalación del driver JDBC

Para poder utilizar correctamente JDBC en la herramienta de desarrollo NetBeans es necesario adjuntar al proyecto el driver correspondiente, se puede encontrar en:

<http://developers.sun.com/product/jdbc/drivers>

Una vez en la página sólo habría que seleccionar la versión correcta y asignarla desde el proyecto.

11.3 Instalación de MySQL

La instalación de MySQL para Mac OS X Snow Leopard es tan sencilla como instalar el componente en preferencias del sistema y dejarlo ejecutado.

El programa puede encontrarse en :

<http://dev.mysql.com/downloads/>

1. Selecciona “Download” bajo “MySQL Community Server”
2. Luego selecciona Mac OS X “package format”
3. Como en este ejemplo se está utilizando OS 10.6 en un MacBook Air, la descarga apropiada es “Mac OS X 10.6 (x86)”.
4. Una vez descargado, haz clic en el paquete y lleva a cabo el proceso de instalación.
5. Una vez que terminó de instalarse puedes ver dónde fue instalado MySQL físicamente visitando el Finder.
6. Escribir /usr/local/ en la caja de texto, luego haz clic en “Ir”. Deberías ver dos carpetas MySQL - una es la principal (la que se llama “mysql-5.1....”), y la otra es una referencia/atajo a la carpeta principal (la que se llama “mysql”).
7. Durante este proceso se establece el nombre de la BD y la contraseña.

8. Para utilizarlo en modo consola simplemente abriríamos x11 y cd /usr/local/mysql
9. Para cambiar la contraseña /usr/local/mysql/bin/mysqladmin -u root password new_password_here.
10. Y para ejecutar por consola /usr/local/mysql/bin/mysql -u root -p.
11. Después por consola creas la BD con “Create Database Blackjack;”.
12. Y te conectas con “Connect Blackjack;”

12. Conclusiones

Como conclusión apuntar los objetivos realizados a la hora de implementar el proyecto:

1. Se ha conseguido implementar un juego funcional.
2. El juego es multiusuario.
3. Se lleva un registro de las partidas que juega cada jugador.
4. Se lleva un registro de los puntos de los jugadores.
5. Es multitarea en el sentido de albergar partidas en varias mesas simultáneamente y con varios jugadores.
6. Permite acceder a las partidas en modo Mirón.
7. Permite registro de usuarios y acceso personalizado y seguro con la contraseña.
8. Ranking de usuarios visible cuando un usuario loguea más de una vez que hace que el juego sea más competitivo y atractivo.

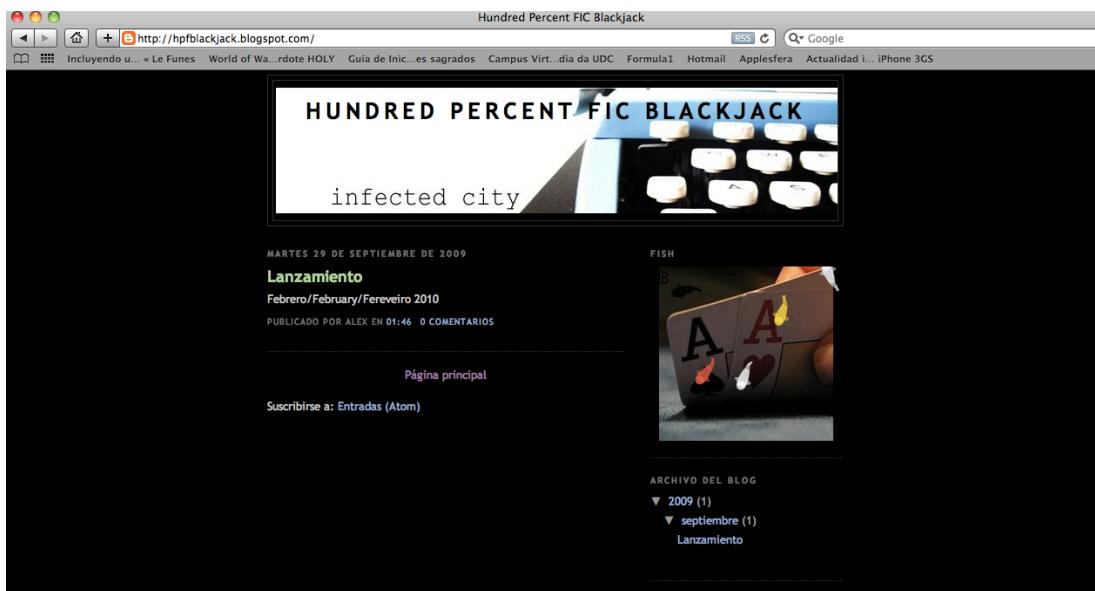
Como resumen dejar constancia de que se buscaba crear una aplicación que permitiese jugar online al BlackJack, esta tarea se ha cumplido con creces al añadir todas las funcionalidades descritas arriba y un control de errores.

Además el autor se ha familiarizado con la tecnología Java y con Mysql.

13. Líneas futuras

Gracias a los diagramas ya implementados se pueden añadir diversas líneas futuras:

- Desarrollo de una página web que albergue el cliente del juego, noticias y las reglas del mismo.



34 Pagina Web

- Implementación de un chat entre usuarios. Al haber implementado los Mensajes esta característica es sencilla de desarrollar puesto que sería enviar y recibir un tipo de Mensaje a mayores de los ya tratados.

- Incluir un algoritmo de encriptado de contraseñas.
- Desarrollar un sistema de conocidos, ya diseñado en el modelo relacional y en el diagrama de Clases.
- Crear partidas competitivas y no competitivas, tambien analizado.
- Una vez desarrollado este juego se podría permitir la posibilidad de agregar distintos tipos de juegos de baraja.

14. Bibliografía

- G. Reese, Database Programming with JDBC and Java, 2nd edition, O. Reilly, 2000.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
- G. Booch, I. Jacobson, J. Rumbaugh, Unified Modeling Language User Guide, 2nd edition, Addison-Wesley, 2005.
- K. Arnold, J. Gosling, D. Holmes, The Java Programming Language, 4th edition, Addison-Wesley, 2005.
- J. Crupi, D. Alur, D. Malks, Core J2EE Patterns, 2nd edition, Prentice Hall, 2003.

15. Links de interes

- <http://www.wikipedia.org/>
- <http://download.oracle.com/javase/6/docs/api/index.html>
- <http://www.netbeans.org>
- <http://www.eclipse.org>
- <https://admin.fic.udc.es/>
- <https://wiki.fic.udc.es/>

16. Apéndice

En el apéndice se han incluido las cuestiones que afectan al proyecto pero que no tienen cabida en otro apartado, como son las peculiaridades del juego o una introducción al software utilizado o la metodología utilizada.

16.1 Historia del Blackjack

El Blackjack es un popular juego de naipes cuyos orígenes se remontan al siglo XVI. Éste fue creado en Francia y, en un comienzo, se le dio el nombre de “veinte y uno”, denominación apropiada ya que el juego consiste en reunir, mediante la acumulación de cartas o el deshacerse de ellas, veintiún puntos antes de que lo logre el croupier, pero sin pasarse de esa cantidad. Este último inciso es aplicable tanto a los jugadores como al croupier.

Una vez que el juego comenzó a ganar popularidad, fue migrando a otras naciones europeas. En Inglaterra, el Veintiuno cambio su nombre. Se le asignó un nombre compuesto que surge de la unión de Jack (vocablo inglés para “Jota”), denominación de una de las cartas fundamentales para este divertimiento y black (“negro”, en inglés) en alusión al color de la espada, uno de los palos de la baraja francesa. La combinación de estos dos términos dio como resultado el nombre por excelencia de este juego, a saber, Blackjack.

16.1.1 Evolución de las reglas

Para el siglo XIX, luego de afianzarse en tierras anglosajonas, el blackjack llegó a América. Y la puerta de entrada fueron los Estados Unidos. Este juego de naipes se hizo popular, a pesar de no tener un derrotero fácil, debido principalmente a las prohibiciones que pesaban sobre los juegos de azar de cualquier índole en este país. Pero la situación comenzó a cambiar cuando se legalizó el juego en el estado de Nevada, para 1931, justo en el momento en que este juego de mesa se había convertido en uno de los favoritos de los estadounidenses.

En las décadas posteriores, otros estados hicieron lo propio, dando la oportunidad para que la fama del blackjack se extendiera aún más. A pesar de su arraigamiento en tierras anglosajonas, a este juego de naipes todavía se lo reconoce por su antiguo nombre, “veintiuno”, en muchos lugares. Se lo asocia con otros jugos con baraja como el “bacará” o el “siete y medio”, pero la popularidad del blackjack es inmensamente superior.

Las reglas del juego evolucionaron con él. Según se cuenta, durante los años 60', cuando la participación en este juego aumentaba a un ritmo vertiginoso, algunos jugadores habituales descubrieron maneras de “ayudar” al azar. Por ejemplo, notaron que, si seguían con detenimiento la partida y “contaban” las cartas propias, las de sus compañeros de mesa y

las del croupier podían establecer un método de ganarle a la banca prescindiendo de la buena suerte.

Los casinos se vieron forzados a modificar los reglamentos de manera progresiva (por ejemplo, aumentando la cantidad de cartas en cada mesa y la cantidad de veces que estas se barajan por juego) para evitar este tipo de prácticas. Pero a pesar de los cambios que ha experimentado a lo largo su extensa historia, la pasión que genera el blackjack se mantiene inalterable.

En Estados Unidos el juego ha sido estudiado minuciosamente. El pionero fue, allá en los años 60 Edward O. Thorp, un matemático empleado de IBM que simuló en ordenador millones de manos jugadas, llegando a la conclusión de que cada mano particular tiene una forma única de jugarse correctamente.

Al conjunto de estas formas únicas de jugar se le denomina estrategia básica, y su aplicación rigurosa permite recortar la ventaja del casino sobre el jugador. Sin esta estrategia básica, el juego en si posee una ventaja matemática de sobre 5 % pero en su correcto uso esta ventaja se reduce a 0.5%.

Thorp llegó también a la conclusión de que las cartas altas favorecen al jugador, ya que son la base para obtener una buena jugada al doblar, o para hacer un blackjack que se paga 3 a 2, mientras que las cartas

pequeñas favorecen al croupier, ya que le permiten hacer buenas las manos comprometidas (12, 13, 14, 15 o 16). Esto dio lugar al denominado conteo de cartas, técnica que consiste en no perder de vista las cartas jugadas, para establecer si entre las que quedan por jugar hay más cantidad de cartas altas o bajas, y apostar en consecuencia.

Ha habido contadores de cartas míticos, que obtuvieron grandes fortunas con esta técnica en los casinos. Ken Uston ha sido considerado por muchos expertos, el mejor contador de la historia.

Los contadores de cartas no están bien vistos en los casinos, y si el casino detecta, o simplemente sospecha que un jugador está contando, le invitarán a cambiar de juego, o sencillamente lo expulsarán del casino amparándose en el derecho de admisión.

De cualquier forma, los días de los contadores parecen llegar a su fin, ya que los casinos están instalando máquinas de barajar automáticas, lo cual hace imposible la aplicación de esta técnica. Existe una película, la cual expone muy bien este hecho: *21: blackjack*.

16.2 Cliente

El cliente es una aplicación informática que se utiliza para acceder a los servicios que ofrece un servidor, normalmente a través de una red de telecomunicaciones.

El término se usó inicialmente para los llamados terminales tontos, dispositivos que no eran capaces de ejecutar programas por sí mismos, pero podían conectarse a un ordenador central y dejar que éste realizase todas las operaciones requeridas, mostrando luego los resultados al usuario. Se utilizaban sobre todo porque su coste en esos momentos era mucho menor que el de un ordenador.

Actualmente se suelen utilizar para referirse a programas que requieren específicamente una conexión a otro programa, al que se denomina servidor y que suele estar en otra máquina. Ya no se utilizan por criterios de coste, sino para obtener datos externos (por ejemplo páginas Web, información bursátil o bases de datos), interactuar con otros usuarios a través de un gestor central, compartir información con otros usuarios (servidores de archivos y otras aplicaciones Groupware) o utilizar recursos de los que no se dispone en la máquina local (por ejemplo impresión).

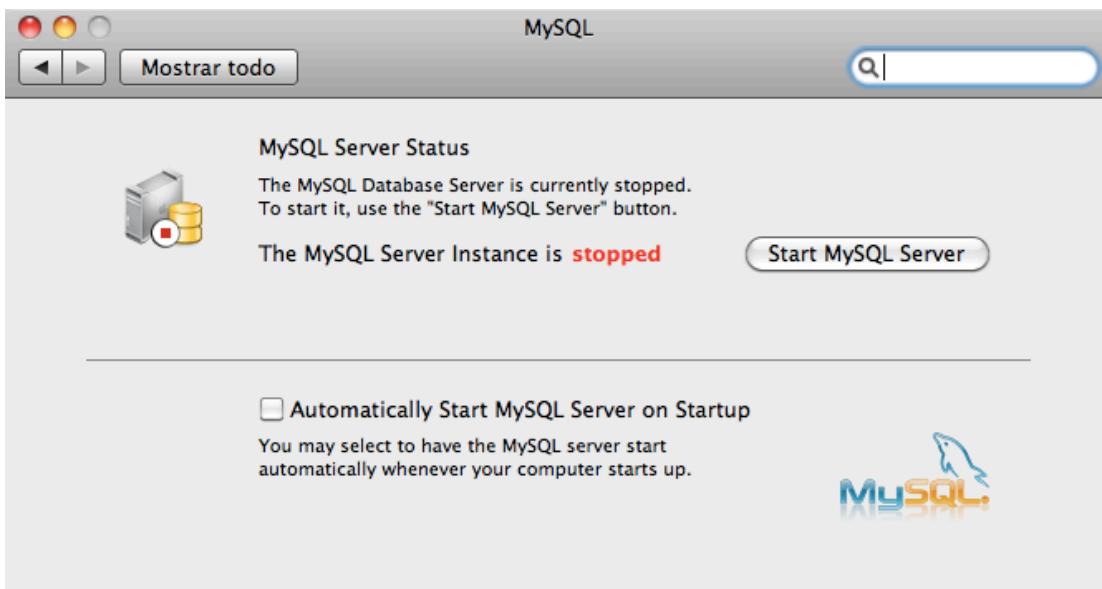
16.3 Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

16.4 MySQL



[35 MySQL](#)

El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) veloz, multi-hilo, multiusuario y robusto. El servidor esta proyectado tanto para sistemas críticos en producción soportando intensas cargas de trabajo como para empotrarse en sistemas de desarrollo masivo de software. El software MySQL tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL. MySQL es una marca registrada de MySQL AB.

16.5 Netbeans



36 Netbeans -6.8-

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

16.6 Servidor

En un entorno de aplicación Cliente-Servidor, el Servidor es una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Este es el significado original del término. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadores.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores Web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los *sistemas multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.

16.7 UML

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE).

UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

16.8 Palabras clave

Java EE: (Java Enterprise Edition): es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una *especificación*. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los suministradores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son *conformes a Java EE*; estandarizado por The Java Community Process / JCP.

JDBC (Java Database Connectivity): JDBC es una especificación de un conjunto de clases y métodos de operación que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. Lógicamente, al igual que ODBC, la aplicación de Java debe tener acceso a un driver JDBC adecuado. Este driver es el que implementa la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC y la base de datos real.

JUnit: *JUnit* es una librería para java que nos facilita las pruebas unitarias de nuestras clases (una prueba por clase) o funcionales (grupo de clases que trabajan juntas).

Combinada con un *build.xml* adecuado de *ant* o con *maven*, nuestras pruebas se pasan cada vez que se compila nuestro proyecto, por lo que las posibles meteduras de pata al rehacer el código quedan de inmediato de manifiesto.

Son importantes este tipo de pruebas y su ejecución automática en equipos de desarrollo en que todos tocan el código de todos o en los que se modifican continuamente clases ya hechas para añadir funcionalidad nueva.

MagicDraw: herramienta CASE destinada al diseño y modelado de software orientado a objetos con facilidades para trabajo en grupo. Permite la realización de diagramas UML 2.0 de manera sencilla.

MySQL: es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

NetBeans: IDE para el desarrollo de aplicaciones en cualquier lenguaje gracias a su carácter modular y extensible, aunque está especialmente enfocado a la programación en Java (Java, Java EE, Java ME, EJBs).

16.9 Estado del Arte (Java)

El lenguaje en sí mismo toma mucha de su sintaxis de C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros memoria.

Las aplicaciones Java están típicamente en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo

Desarrollo de aplicación que permita jugar por Internet en tiempo real al Blackjack

con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre.

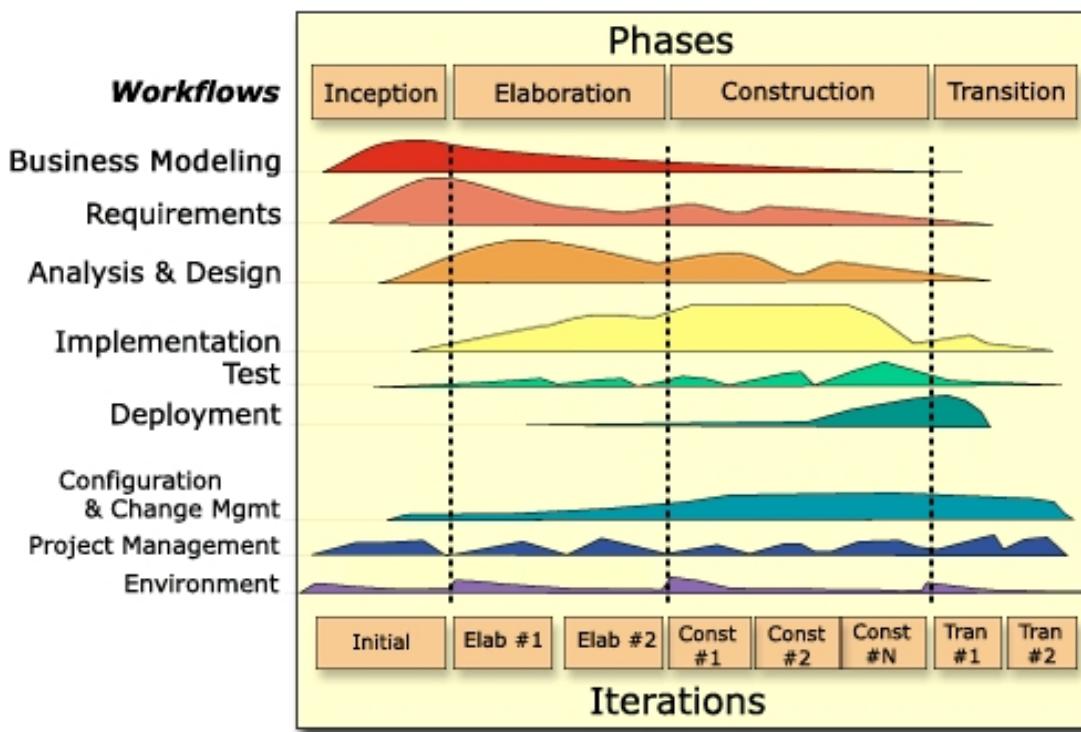
16.10 Metodología utilizada

La metodología escogida para la realización de este proyecto ha sido el Proceso Unificado (UP) o RUP.

Las características más importantes de esta metodología son :

- Enfocado a los riesgos: en el desarrollo de software, un error detectado en un estado tardío es mucho más costoso que uno detectado en las primeras fases *de* desarrollo. Por ello, el PU potencia que el desarrollador identifique los principales riesgos en las etapas tempranas.
- Manejado por casos de uso: Un caso de uso representa una pieza de funcionalidad en el sistema que le devuelve al usuario un resultado de valor. Los casos de uso sirven para capturar requerimientos funcionales. Todos los casos de uso de un sistema conforman el modelo de casos de uso. Se dice que el Proceso Unificado está manejado por casos de uso porque el desarrollador creará modelos de implementación y diseño que harán efectivos a los casos de uso planteados en la etapa de análisis. Cada modelo del proceso tendrá una correspondencia con el modelo de casos de uso.

- Centrado en la arquitectura: A pesar de que es cierto que los casos de uso manejan el proceso, éstos no lo hacen en soledad, sino que son desarrollados en conjunto con la arquitectura del sistema. Los casos de uso manejan la arquitectura del sistema y la arquitectura del sistema influencia en la selección de los casos de uso. En consecuencia, tanto la arquitectura del sistema como los casos de uso maduran a medida que el ciclo de vida continua.
- Iterativo e incremental: Para el PU la vida de un sistema se encuentra dividida en ciclos. Cada ciclo termina con un lanzamiento de diferentes modelos del producto. Cada ciclo consiste de cuatro fases: concepción, elaboración, construcción y transición. Cada fase se encuentra subdividida en iteraciones. Al final de cada fase se produce un “milestone” o punto de revisión. Una iteración es un mini proyecto que concluye con la entrega de algún documento o implementación interna. Dentro de cada iteración se realizan actividades de captura de requerimientos, análisis, diseño, implementación y prueba.



37 Iteraciones en el Proceso Unificado

16.11 MVC

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Elementos del patrón:

Modelo: datos y reglas de negocio

Vista: muestra la información del modelo al usuario

Controlador: gestiona las entradas del usuario

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de tarta, de barras, tabular, etc. Veamos cada componente:

- El modelo es el responsable de:

Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.

Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".

Lleva un registro de las vistas y controladores del sistema.

Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero batch que actualiza los datos, un temporizador que desencadena una inserción, etc.).

- El controlador es responsable de:

Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).

Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()".

- Las vistas son responsables de:

Recibir datos del modelo y mostrarlos al usuario.

Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).