

Ingeniería de Software I

Primer Cuatrimestre 2008

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico - Parte II

Diseño

Grupo 5

Integrante	LU	Correo electrónico
Echevarria, Pablo	133/00	pablohe@gmail.com
Farjat, Lucas	468/05	lacacks@gmail.com
Freijo, Diego	4/05	giga.freijo@gmail.com
Giusto, Maximiliano	486/05	maxi.giusto@gmail.com

Instancia	Tutor / Corrector	Nota
Entrega		
Reentrega		

Índice

1. Introducción	3
2. Escenarios y Diagramas de Secuencia	4
2.1. Funcionalidades generales entre los jugadores y el casino	4
2.2. Funcionalidades referentes a Tragamonedas	4
2.3. Funcionalidades referentes a Craps	5
2.4. Funcionalidades generales de los administradores	7
3. Pseudocódigo de operaciones más complicadas en lo algoritmo.	8
4. Diagrama de clases	8
4.1. Gráfico Diagrama de Clases	8
4.2. Explicación del Diagrama de Clases	9
5. Justificación, análisis y explicación del diseño	10
6. Modificación del protocolo	11
7. Trazabilidad	12
8. Conclusiones	13

1. Introducción

En esta sección describiremos los cambios con respecto al informe 1.

2. Escenarios y Diagramas de Secuencia

Planteamos 2 enfoques para la construcción de los escenarios y diagramas de secuencia. La primera fue basada en el echo de que contabamos con protocolos que inducian interacción de nuestras clases. El otro fue basado en los casos de uso.

2.1. Funcionalidades generales entre los jugadores y el casino

Escenario 1

- el jugador entra al casino.
- se loguea
- el casino autentica sus datos.
- da la respuesta

Escenario 2

- el jugador le pide salir al casino
- el casino verifica si puede hacerlo
- da la respuesta

Escenario 3

- elige jugar al tragamonedas, especifica el valor de ficha
- el casino valida el valor ingresado, resulta valido
- le da el número de mesa
- o le niega la entrada

2.2. Funcionalidades referentes a Tragamonedas

Escenario 4

- Un jugador con una mesa elegida para jugar
- inserta una cantidad válida de fichas en una máquina tragamonedas
- gira los rodillos.
- Ésta, luego de debitar el monto establecido con anterioridad correspondiente al pozo progresivo, consulta el tipo de jugada al casino
- le responde que es una jugada normal
- la máquina genera el resultado de la jugada en base a las probabilidades que le establecieron.
- Ésta termina arrojando un resultado ganador
- el jugador observa el resultado y tipo de jugada
- recibe el cobro.

Escenario 5

- Un jugador con una mesa elegida para jugar inserta una cantidad válida de fichas en una máquina tragamonedas

- gira los rodillos.
- debita el monto establecido con anterioridad correspondiente al pozo progresivo
- consulta el tipo de jugada al casino
- le responde que es una jugada normal
- la máquina genera el resultado de la jugada en base a las probabilidades que le establecieron
- termina arrojando un resultado perdedor
- el jugador observa el resultado y tipo de jugada.

Escenario 6 Un jugador con una mesa elegida para jugar inserta una cantidad válida de fichas en una máquina traga-monedas y gira los rodillos. Ésta, luego de debitar el monto establecido con anterioridad correspondiente al pozo progresivo, consulta el tipo de jugada al casino y le responde que es una jugada todos ponen. Luego la máquina genera el resultado de la jugada en base a las probabilidades que le establecieron. Ésta termina arrojando un resultado ganador para el jugador quién luego observa el resultado y tipo de jugada, recibe el cobro y la máquina debita en el pozo feliz el importe correspondiente.

Escenario 7 Un jugador con una mesa elegida para jugar inserta una cantidad válida de fichas en una máquina traga-monedas y gira los rodillos. Ésta, luego de debitar el monto establecido con anterioridad correspondiente al pozo progresivo, consulta el tipo de jugada al casino y le responde que es una jugada feliz. Luego la máquina genera el resultado de la jugada en base a las probabilidades que le establecieron. Ésta termina arrojando un resultado ganador para el jugador. Éste observa el resultado, el tipo de jugada y luego cobra la suma entre el pago por la apuesta ganada y el pozo feliz hasta ese momento. Luego el pozo feliz es reiniciado a su valor inicial.

Escenario 8 Un jugador con una mesa elegida para jugar inserta la máxima cantidad de fichas en una máquina tragamonedas luego de jugar en ella el mismo importe tantas veces como las necesarias para que pueda ganar el pozo progresivo y gira los rodillos. Ésta, luego de debitar el monto establecido con anterioridad correspondiente al pozo progresivo, consulta el tipo de jugada al casino y le responde que es una jugada de pozo progresivo. Luego la máquina genera el resultado de la jugada en base a las probabilidades que le establecieron. Ésta termina arrojando el resultado con pago máximo. El jugador observa el resultado y tipo de jugada. Luego cobra la suma entre el pago por la apuesta ganada y el monto total del pozo progresivo. Luego el pozo progresivo es reiniciado a su valor inicial.

2.3. Funcionalidades referentes a Craps

Escenario 9 Jugador de Craps entrando a una mesa

- El usuario decide entra a una mesa de craps
- verifica que no este jugando y que sea un jugador del casino
- el sistema crea la mesa en caso de que no especifique una
- lo ingresa
- lo notifica

Escenario 10 Invitado mirando una ronda

- El invitado esta en el casino
- decide ver el juego de craps de una mesa cualquiera de las abiertas
- Mira la ronda y luego sale

Escenario 11 Jugador que se hace tirador

- el jugador está en el juego de craps
- espera n rondas, hasta que le toca tirar, se convierte en tirador.

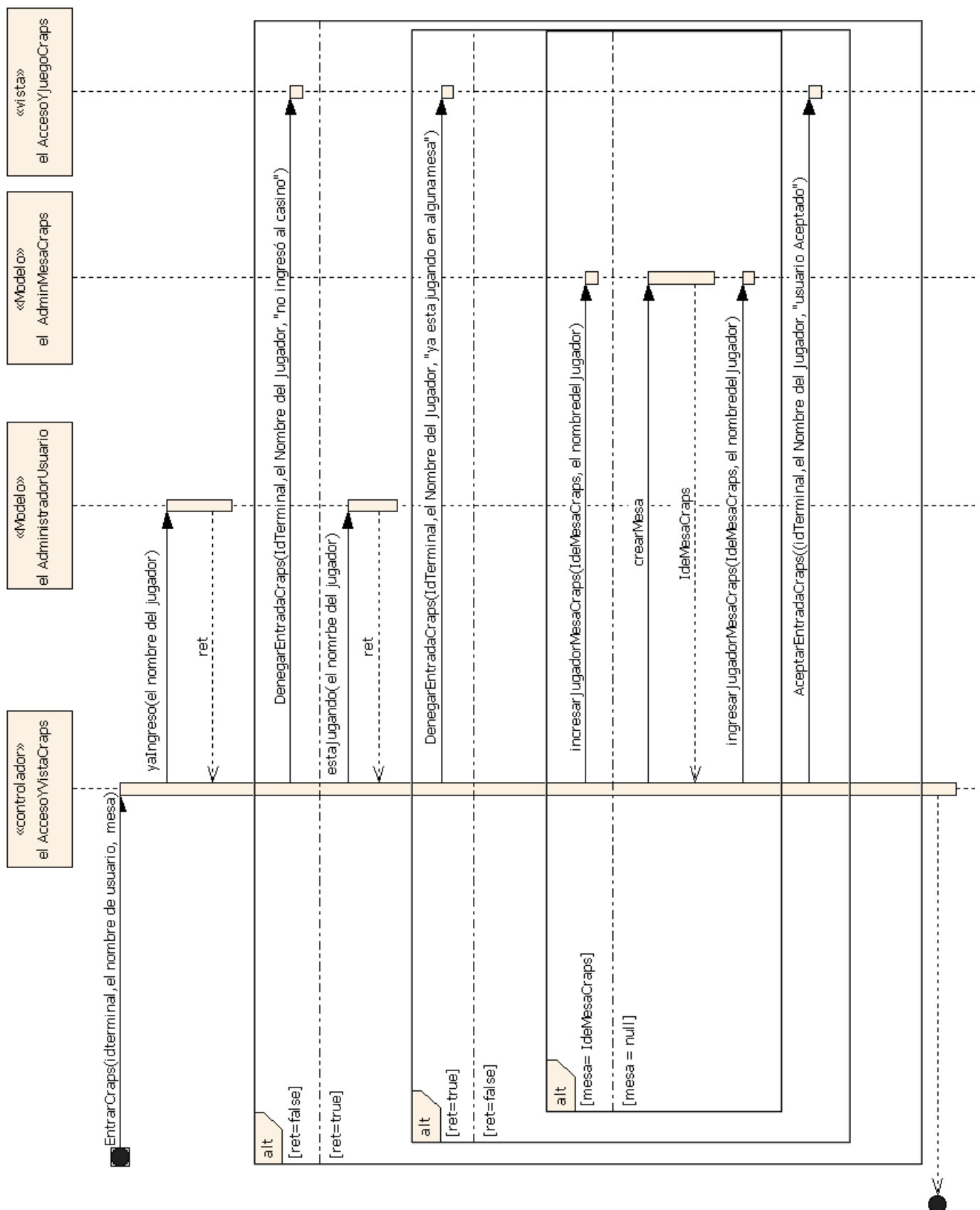


Figura 1: Jugador de Craps entrando a una mesa

- Tira los dados.
- No se hacen apuestas.
- Saca un seven out por lo que deja de ser tirador.

Escenario 12 Jugador haciendo apuesta de craps

- Un jugador esta en una mesa
- Decide hacer una apuesta de *campo* por una ficha de menor valor.
- Se hace el tiro, sale 7, pierde.
- En la siguiente jugada le toca tirar.
- Hace la misma apuesta.
- Sale 12.
- Gana su apuesta, se le paga 2 a 1.

Escenario 13

- está en el juego de craps en una mesa
- Decide hacer una apuesta *linea pase* por una ficha de menor valor.
- Se hace el tiro, sale 7, pierde.

Escenario 14

- Un jugador ingresa al juego de craps, elige una mesa
- aguarda n rondas hasta que decide hacer su apuesta
- Decide hacer una apuesta *venir* por una ficha de menor valor, por lo cual debe aguardar hasta que se establezca el punto.
- Se hace el tiro, sale 2, pierde.

Escenario 15

- Un jugador ingresa al juego de craps, elige una mesa
- aguarda n rondas hasta que decide hacer su apuesta, debe aguardar hasta que se establezca el punto.
- Decide hacer una apuesta *venir* por una ficha de menor valor.
- Se hace el tiro, sale 2, pierde.

2.4. Funcionalidades generales de los administradores

3. Pseudocódigo de operaciones más complicadas en lo algoritmo.

4. Diagrama de clases

4.1. Gráfico Diagrama de Clases

El enfoque en este diagrama es sobre el *Servidor*.

El grafico del modelo va en formato digital

4.2. Explicación del Diagrama de Clases

El módulo del casino se puede separar en dos partes:

- el módulo *Cliente*
- módulo *Servidor*

Agregando algo más a resolver: la comunicación entre los módulos. Dicha cuestión ya fue resuelta por el documento de Arquitectura Conceptual y Protocolo brindado por los clientes.

A continuación tenemos la explicación de las clases (o conjunto de ellas) más relevantes del diagrama.

- *Clases AdministradorJugador, AdministradorObservador y FachadaUsuario*. Éstas clases hacen el manejo del ingreso y egreso de jugadores y observadores. Además de las consultas sobre el saldo, la existencia (entre otras) de jugadores y observadores
- La *Clase Usuarios* contiene la lista de jugadores (normales y vip) y la lista de observadores. Sus operaciones están relacionadas con el manejo de las antedichas listas y obtención de datos sobre el jugador, mediante *ObtenerSaldoJugador*
- *Clase AdministradorMesaCraps* es la responsable del ingreso y egreso de jugadores en las mesas, decir qué mesas están abiertas, últimos resultados, entre otros. Básicamente se ocupa de administrar todo lo referente a las mesas y los tiros del juego craps.
- *Clase AdministradorMesaTragamonedas* al igual que la anterior administra lo referente a las mesas y los tiros, pero en este caso es sobre el juego de Tragamonedas.
- La *Clase Mesas* contiene la lista de mesas de craps y la lista de mesas tragamonedas. La *Clase MesaTragamonedas* y la *Clase MesaCraps* son las representaciones de las mesas del juego Tragamonedas y del juego Craps respectivamente.
- La *Clase JugadaCraps* contiene el resultado de los dados, a través de la *Clase Dado*. También contiene las apuestas efectuadas y por qué importe.
- La *Clase JugadaTragamonedas* contiene el resultado de los rodillos, a través de la *Clase ResultadoTragamonedas*. Pues esta contiene la *Clase RodilloTragamonedas* que tiene como atributo el valor del rodillo. También se relaciona con la *Clase PozoProgresivo*, de la cual se obtendrá el monto para realizar el pago y el consiguiente reseteo del mismo.
- La *Clase AdministradorPozos* muestra los saldos del pozo progresivo y del pozo feliz.
- La *Clase ReceptorPedidos* mediante su especialización levantará los distintos pedidos de la manera que sea necesaria.
- La *Clase DespachadorRespuestas* es análoga a la anterior y tiene el mismo comportamiento diferenciándose en que es para las respuestas. Hay una por puerto.
- *Clase DespachadorPedidos* interpretará el pedido y según de quién provenga se lo dará para procesar a alguna de las Clases *JuegoCraps*, *AccesoYVistaCraps*, *AccesoYVistaTragamonedas* o *AccesoYVistaCasino*.

5. Justificación, análisis y explicación del diseño

Se modeló basandonos en el patrón de diseño MVC. Con lo cual tenemos tres grandes paquetes:

- Modelo
- Controlador
- Vista

La recepción de los pedidos se hace con la Clase ReceptorPedidos, que está dentro del paquete Vista. Esta clase será especializada de la manera que sea necesaria. Esto brinda una mayor flexibilidad a la hora de tomar los pedidos y nos liga menos al tipo en el que llega el pedido.

Al levantarse los datos son guardados en una clase Pedido la cual tiene un diccionario en el cual las claves son del tipo texto y el resultado es un objeto. Esto es bueno, ya que permite flexibilidad ante los cambios de parámetros de entrada. Tanto de tipo como de cantidad.

La Clase DespachadorPedidos (paquete controlador) toma los parámetros de entrada y multiplexa según corresponda. Esta clase es singleton debido a que solo es necesaria una sola instancia para no tener problemas de concurrencia.

Las clases JuegoCraps, AccesoYVistaCraps, AccesoYVistaTragamonedas y AccesoYVistaCasino procesan los datos según corresponda. Los datos son obtenidos de los administradores que están en el paquete Modelo. Esto crea acoplamiento sobre dichas clases, pero no las crea sobre el resto del paquete modelo. Si bien el acoplamiento está, éste es entre paquetes que es menor que entre clases de distintos paquetes. Éstas clases son singleton.

Tenemos las Clases Singleton para la emisión de la respuesta: JuegoCraps, AccesoYVistaCraps, AccesoYVistaTragamonedas y AccesoYVistaCasino.

Estas clases se corresponden uno a uno con las del controlador, brindando las respuestas a los pedidos. En el caso de necesitar datos, que no fueron brindados por el controlador, también le hacen el pedido a las clases administrador del paquete Modelo.

Como son para emitir la respuesta están dentro del paquete Vista.

Para emitir la respuesta utilizarán un tipo llamado Respuesta. Esto permite desacoplar del tipo de respuesta y dejar a la clase especializada del Despachador de respuestas, la responsabilidad del formato de respuesta. Como es especialización en caso en que se modifique el modo en que se envía la respuesta se hereda una clase nueva. Favoreciendo el open-close.

6. Modificación del protocolo

Las extensiones del protocolo están en un documento adjunto.

7. Trazabilidad

8. Conclusiones