

Diseño y Desarrollo de Aplicaciones

Obligatorio



Mauro Carrero
66163



Fernando Gonzalez
141293

Grupo: N4A
Nombre del docente: Santiago Arreche
27/10/2014

Introducción

El siguiente trabajo consiste en la implementación de un sistema de juego del tipo Bingo. El sistema básicamente posibilitará ingresar, por parte de los usuarios administradores, la cantidad de filas y columnas que tendrán los cartones, cantidad máxima de jugadores y cartones por jugador y el valor del cartón. En tanto, los usuarios “Jugadores” podrán seleccionar con cuantos cartones jugar. Toda la información que visualizará, tanto el administrador, como los jugadores, se actualizará de forma automática.

Índice

Diagrama Conceptual de Dominio	4
Diagrama de Clases Completo	5
Autoevaluación.....	6
Funcionalidad	6
Interfaz de usuario.....	6
Diagramas de clases.....	6
Requerimientos de diseño y arquitectura	6
Datos de Prueba	7

Diagrama Conceptual de Dominio

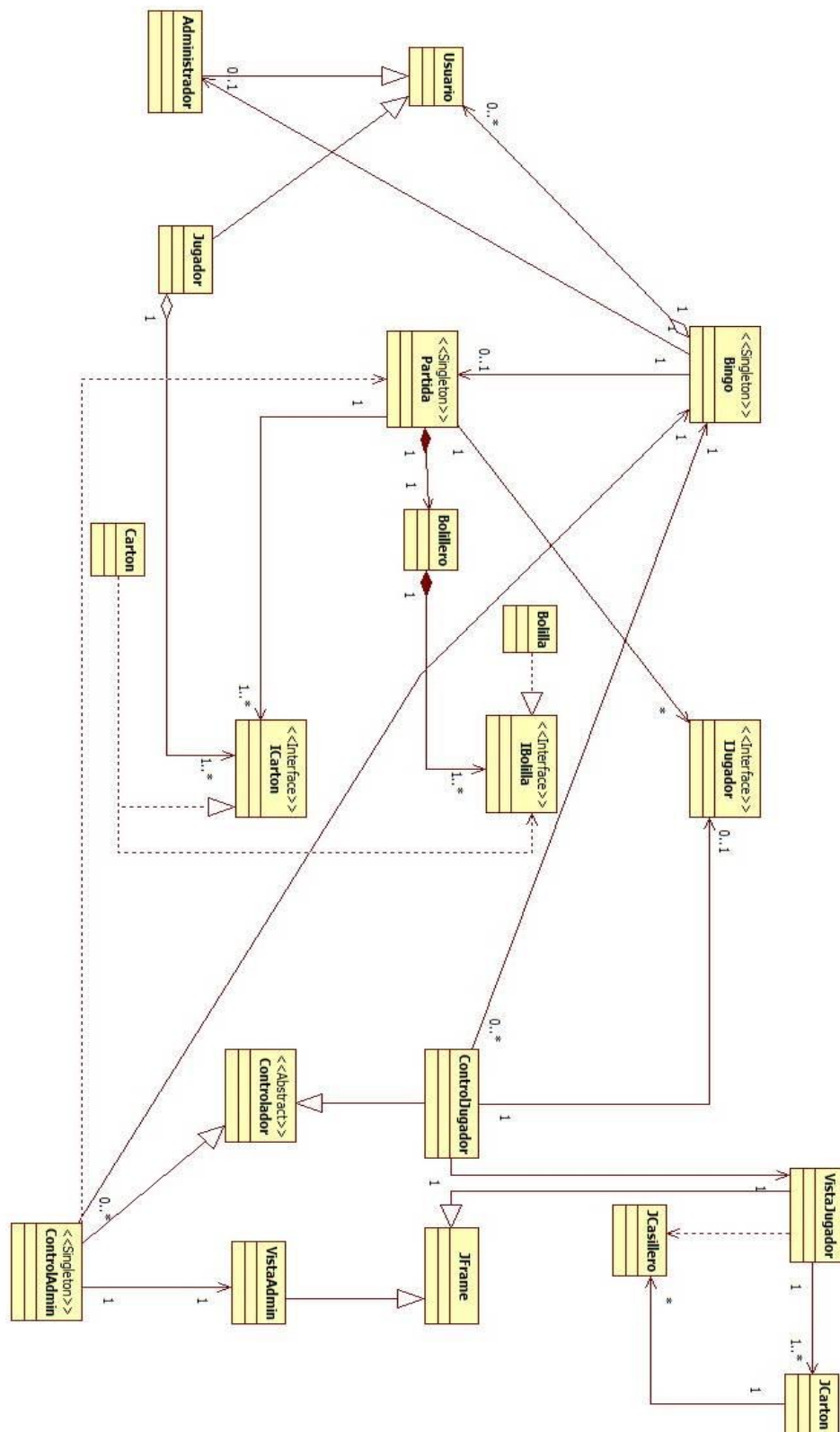
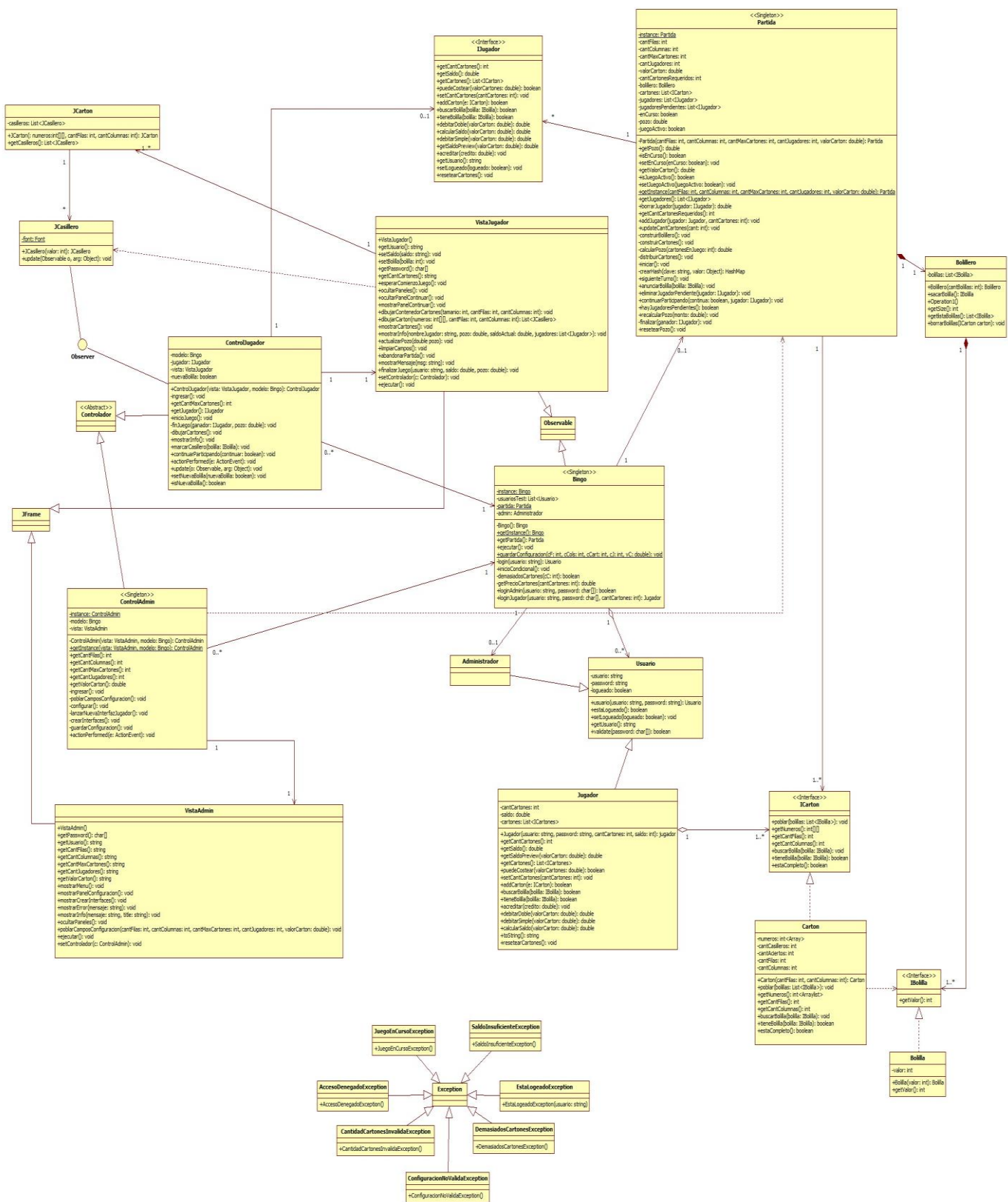


Diagrama de Clases Completo



Autoevaluación

Funcionalidad

- Se realizaron todas las funcionalidades requeridas por la letra de la primera entrega. Si bien en ésta primera instancia nos orientamos a ejecutar tanto la interfaz de los jugadores y administradores en un mismo equipo, se programó pensando en que en la segunda entrega es necesario que se ejecuten en distintos equipos, orientado a una estructura cliente-servidor. Utilizamos interfaces, utilizándolas como medio de transporte para la comunicación del cliente con el servidor, si bien este enfoque no está totalmente implementado, ya que en parte correspondería a una segunda entrega.
- En cuanto a mejoras, se nos ocurrieron algunas ideas que podrían o no implementarse en una segunda entrega, por ejemplo utilizar una clase “Pozo” con el objetivo de reducir las responsabilidades de la clase “Partida”. Se podría mejorar la forma en que borramos los jugadores cuando éstos abandonan la partida, básicamente minimizar la cantidad de recorridas que se ejecutan sobre los cartones del jugador que abandona buscando las bolillas a eliminar.

Interfaz de usuario

- Se contemplaron de forma idéntica los casos de uso de la descripción de la letra del obligatorio, inclusive los textos que se muestran a los usuarios. Intentamos dejar la interfaz de usuario lo más amigable posible, obviamente al no estar acostumbrados a trabajar con aplicaciones de escritorio, no se diseñó la mejor interfaz, pero funciona.

Diagramas de clases

- El diagrama de clases generó alguna duda en cuanto a la cardinalidad y relaciones entre algunas clases, sobre todo en lo que respecta a la interfaz de usuario. La concordancia del código con los diagramas es el adecuado.

Requerimientos de diseño y arquitectura

- **Maximizar la modularidad y claridad de código:** Los métodos están concretos y claros, alguno es poco más extenso que otro, todos se visualizan en una pantalla. Intentamos programar pensando en una estructura cliente-servidor, sobre todo en la separación de paquetes.
- **Minimizar la duplicación de código:** Prácticamente pudimos eliminar la duplicación de código delegando adecuadamente las distintas responsabilidades, entre las diferentes clases. **[Patrón Experto]**

- **Distribución de las aplicaciones:** Actualmente se encuentra todo en una misma aplicación pero a futuro se dividirán en una aplicación para el usuario jugador y otra para el usuario administrador la cual podría ser o no la aplicación servidor.
- **Información entre lógica e interfaz de usuario:** Las interfaces de usuario reciben la información desde la lógica y la maneja para presentarla como desee. En general las interfaces de usuario se comunican con su controlador y los controladores con la lógica, principalmente con la clase Bingo.
 - En éste caso se implementó el patrón **[Observer]** en dos **oportunidades:** entre la clase Bingo que es observada por todos los controladores del jugador, una vez que éstos son notificados por la clase Bingo, notifican a su vez a todos los JCasilleros (Observadores) que comparan la bolilla sorteada con su propio valor. Esto evita que se deba recorrer buscando todos los componentes de tipo casillero en forma iterativa.
- **Diseño de lógica:** El diseño de la lógica se hizo en base a que fuera flexible, reusable y escalable. La clase Bingo y la clase Partida se utilizan como fachada.

Concepto	Pts
Funcionalidad: Implementación de la funcionalidad solicitada en java	7
Diagramas de clase y concordancia del código con los diagramas	2
Requerimientos de diseño	10
Auto Evaluación	1
Total:	20

Datos de Prueba

USUARIO	CONTRASEÑA	ROL
fgonzalez	fgonzalez	Administrador
mcarrero	mcarrero	Administrador
j1	j1	Jugador
j2	j2	Jugador
j3	j3	Jugador
j4	j4	Jugador
j5	j5	Jugador