

# Trabajo Práctico 2

## AlgoPoly

[7507] Algoritmos y Programación III  
Curso 1  
Segundo cuatrimestre de 2017

Alumno	Padrón	Email
Simizu, Nicolás Martín:	94556	nicosimi993@gmail.com
Quino, Julian	94224	julian.quino2@gmail.com
Choque Scipione, Franco	98756	franco.choquescipione@gmail.com

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Supuestos</b>	<b>2</b>
<b>3. Modelo de dominio</b>	<b>2</b>
<b>4. Diagramas de clase</b>	<b>2</b>
<b>5. Detalles de implementación</b>	<b>3</b>
5.1. Comportamiento de los casilleros . . . . .	3
<b>6. Excepciones</b>	<b>4</b>
<b>7. Diagramas de secuencia</b>	<b>4</b>
<b>8. Diagramas de estado</b>	<b>4</b>
<b>9. Diagramas de paquetes</b>	<b>4</b>

## 1. Introducción

El presente trabajo tiene la finalidad de aplicar los conceptos aprendidos durante el curso, para la resolución de problemas, para ello, se tiene que desarrollar una aplicación que implemente el clásico juego del Monopoly.

## 2. Supuestos

Durante la implementación, se contemplaron situaciones que no son provistas en las indicaciones de la aplicación. Una de ellas, por ejemplo, es la situación cuando el jugador cae sobre el casillero de salida. Dado que no se explicita que debería suceder, se optó por dejar el casillero sin ninguna acción a realizar.

También se considera que los casilleros no van a cambiar su estado a lo largo del juego, refiriéndose a que una propiedad permanece como tal desde que inicia el juego hasta su fin.

## 3. Modelo de dominio

Para la implementación de la aplicación se utiliza una entidad que dirija su desarrollo y coordine a las demás y otra que funcione como interfaz de usuario. La primera se la denominó Tablero y la segunda, algoPoly.

Para desarrollar las entidades que componen al juego se utilizan las siguientes entidades:

**Tablero** : Esta entidad representa el tablero del juego, conteniendo los casilleros, referencias a los jugadores y estructuras(Casas y hoteles)

**Casillero** : Esta entidad representa a cada casillero del juego y depende de su posición en el tablero, cambiará su comportamiento.

**Jugador** : Esta entidad representa a un participante en la aplicación.

**Dado** : Esta entidad representa un dado del juego y tiene una única operación, obtener un número aleatorio del 1 al 6.

**Casa y Hotel** : Estas entidades representan las construcciones posibles para los casilleros que sean propiedades.

Con estas clases presentes, se busca proveer al usuario una manera sencilla de interactuar con la aplicación.

## 4. Diagramas de clase

En la figura 1 se presenta la relación entre las diferentes clases que componen la aplicación.

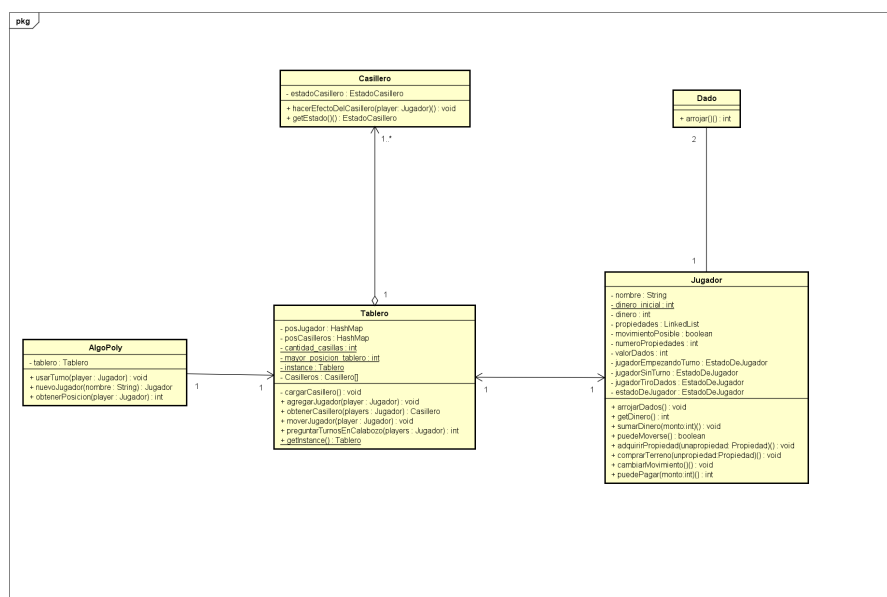


Figura 1: Diagrama de clases de algoPoly.

## 5. Detalles de implementación

### 5.1. Comportamiento de los casilleros

En la implementación de los diferentes comportamientos que puede tener las instancias de Casillero, se decidió por utilizar el patrón State, para simplificar como deberían enviarse los mensajes para ejecutar los efectos de cada casillero.

Como se muestra en la figura 2, se tienen categorías de comportamientos de casilleros: están divididos en Comprables y No comprables. Los comportamientos Comprables heredan de Estado-Casillero y contienen los métodos y atributos necesarios para que un casillero pueda comprarse. Así mismo, de Comprables heredan las subclases Propiedades y Servicios, ya que los casilleros de Propiedades permiten construir edificios (Casas u hoteles, según corresponda el comportamiento) mientras que los Servicios no.

En las especificaciones de la aplicación también se pide que las casillas comprables puedan intercambiarse, si bien esto es posible hacerlo agregando las funciones necesarias en la interfaz, las pruebas pedidas para la entrega N1 de la aplicación no exigen este comportamiento.

También hay dos categorías que no se incluyen, Cárcel y Policía: son dos casilleros que no pueden integrarse. El comportamiento de policía es simplemente mover el jugador a la casilla Cárcel. Por su parte, Cárcel lo único que hace es retringirle el movimiento al jugador hasta que pague la fianza o llegue a estar 3 turnos sin moverse, luego le devuelve la capacidad de moverse al jugador.

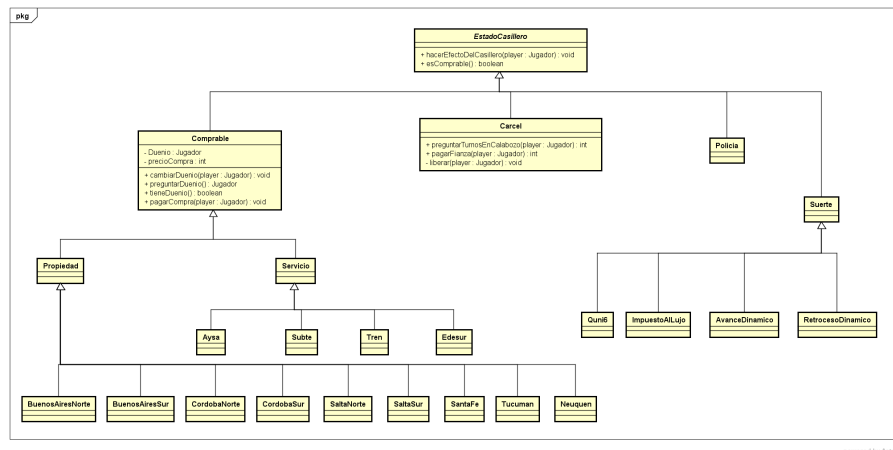


Figura 2: Diagrama de clases para el comportamiento del casillero.

## 6. Excepciones

**DineroInsuficiente** Esta excepción se creó para los casos en que un jugador intenta pagar y no dispone de suficiente dinero.

**TurnosEnCalabozoInvalido** Esta excepción se creó para los casos en que los turnos que el jugador permaneció en la carcel no son válidos para realizar la acción.

**JugadorYaTiroDados** Esta excepción se creó para los casos en que al jugador no se le permite volver a tirar los dados.

**JugadorNoTiroDados** Esta excepción se creó para los casos en que al jugador no se le permite terminar su turno sin antes tirar los dados.

**NoEsTurnoDelJugador** Esta excepción se creó para los casos en que la acción no sea legal en el estado del jugador.

## 7. Diagramas de secuencia

En la figura...

En la figura...

## 8. Diagramas de estado

## 9. Diagramas de paquetes

En la figura 3 se muestra el diagrama de paquetes de la aplicación. Se ajustó la distribución de las entidades requeridas por la aplicación según el patron MVC, que tiene por separado la vista, el modelo y el controlador.

La lógica del programa, las excepciones y como se implementan las clases para la aplicación se encuentran bajo el paquete Implementacion.programa. Este paquete representa la parte de modelo del MVC.

El paquete interfaz.usuario contiene la los comandos que se le permiten ejecutar al usuario para modificar el modelo.

El paquete vista contiene todos los archivos de imagen y sonido requeridos para dar la representación gráfica del modelo.

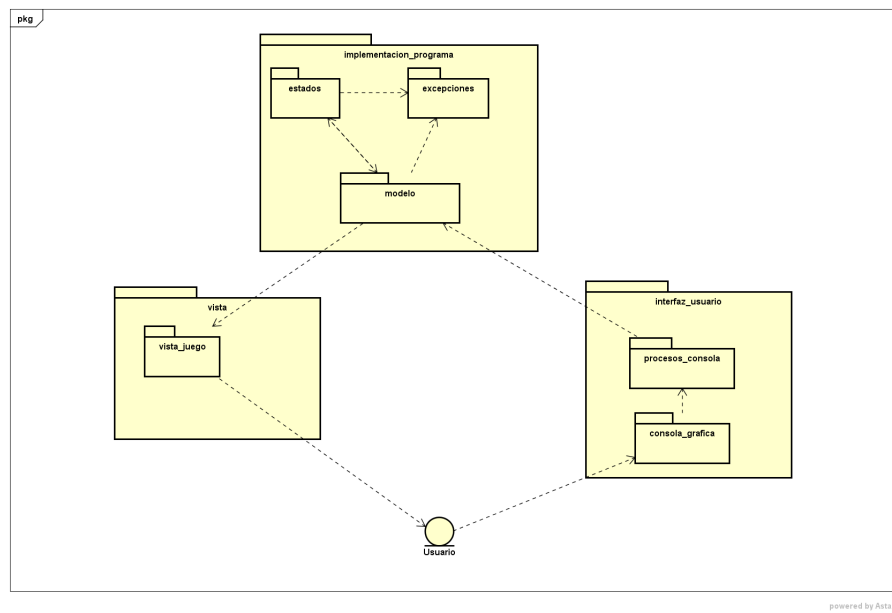


Figura 3: Diagrama de paquetes de algoPoly