

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III

Curso 1

Segundo cuatrimestre de 2017

Alumno	Padrón	Mail
Victoria Zambianchi	98570	viky.zambi@gmail.com
Federico del Mazo	100029	delmazofederico@gmail.com
Nicole Raveszani	101031	nraveszani@gmail.com
Florencia Rodríguez	100033	florrr1997@gmail.com

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Detalles de implementación	4
5. Diagramas de clase	6
6. Diagramas de Secuencia	6

1. Introducción

El presente informe reúne la documentación para la primera entrega del trabajo práctico número 2 en Java `.AlgoPoly`.

2. Supuestos

Avance y retroceso dinámico Estas dos clases devuelven uno que hace referencia a la cantidad de movimientos que debe realizar el jugador para llegar a su posición final. suponemos que las cuentas dadas para los distintos rangos de resultado de dado no puede ser negativos. Por ejemplo, que la cantidad de propiedades siempre será menor a 11, y de esta manera, en avance dinámico nunca se devolverá un número negativo, La misma lógica se aplica en retroceso dinámico.

Barrio El jugador es libre de decidir que barrios comprar y si quiere contruir casas u hoteles. El juego jamás le dejara contruir un hotel a un jugador que no tenga los barrios y edificaciones necesarias

3. Modelo de dominio

Paquete Modelo:

- Clase Jugador: representa a un jugador de AlgoPoly. Posee un nombre a elección del usuario y comienza el juego con determinado capital. Puede adquirir propiedades y edificar sobre las mismas en caso de tener el capital suficiente. Además, según en qué casillero se encuentre, el comportamiento del mismo puede afectar su capital o su capacidad de movimiento en los siguientes turnos.
- Clase Municipio: representa una entidad administrativa que se encarga de conocer y actualizar los propietarios y la cantidad de edificaciones (en caso de ser un barrio) de aquellas clases que implementen la interfaz Propiedad, cuando los jugadores ejecuten acciones sobre alguna propiedad. Al poseer una base de estos datos, puede decir quien es el propietario de cada propiedad, ceder la misma al banco de ser necesario, si un jugador es capaz de edificar en un barrio una casa o un hotel.
- Clase Tablero: representa el tablero de juego AlgoPoly. El tablero se llena al ser instanciado por todos los casilleros correspondientes al juego. Se encarga de seleccionar la posición siguiente de cada jugador basándose en su posición actual y en el número que salió en la tirada de dados.
- Clase Turno: representa un turno correspondiente a un jugador de AlgoPoly. En caso de que el jugador pueda moverse, le solicita al tablero el casillero siguiente al jugador y se lo asigna.
- Clase Dados: se instancia con dos números generados al azar del 1 al 6 y puede devolver la suma o si ambos números son idénticos.
- Clase Edificacion: representa una construccion que puede hacer una propiedad. Posee su costo de edificación y el alquiler que otorga.
- Interfaz Estado: la interfaz obliga a aquellos que la implementan (las clases Libre y Preso) a devolver si el movimiento es posible y a habilitar el movimiento.
- Clase Libre: representa que el jugador es libre de moverse.

- Clase Preso: representa que el jugador se encuentra preso en la Carcel. Para moverse es necesario que el jugador haya jugado 3 turnos o haber pagado la fianza. Para habilitar el movimiento se le solicita la fianza al jugador

Paquete Casilleros:

- Interfaz Casillero: toda clase que la implementa es la representación de una casilla. Esta interfaz las obliga a tener los métodos `accionAlCaer()`, que es la acción que se ejecuta cuando un jugador cae en determinado casillero, `nombre()` y `color()` que son necesarios en métodos de otras clases.
- Interfaz Propiedades: toda clase que la implementa es la representación de una propiedad, con esto nos referimos a aquellos casilleros que tienen la particularidad de poder ser adquiridos por un jugador y aumentar su capital bajo ciertas condiciones. Una propiedad puede tratarse tanto de un barrio como un servicio.
- Clase Barrio: representa a un barrio del juego AlgoPoly. Puede ser comprado y vendido por un jugador, el propietario puede edificar construcciones y al caer un jugador que no es el propietario debe pagar un alquiler a quien lo sea.
- Clase BarrioDoble: posee el mismo comportamiento que la clase Barrio, a diferencia de que se puede reemplazar a las casas por un hotel unicamente si el jugador es propietario ademas del barrio doble requerido.
- Clase Servicio: representa un servicio del juego AlgoPoly. Al igual que el barrio, puede ser comprado y vendido por un jugador. Si un jugador que no es el propietario cae en una instancia de este casillero, se le cobra un costo de servicio igual al costo original del servicio (que depende de si el jugador propietario posee o no la propiedad doble correspondiente) multiplicado por el numero de dados que saco.
- Clase Policia: representa a la policia del juego AlgoPoly. Cuando el jugador cae en Policia lo envia a la Carcel.
- Clase Carcel: representa una cárcel del juego AlgoPoly. Al caer en ella, el jugador es incapaz de moverse hasta que no se cumplan ciertas condiciones.
- Clase ImpuestoAlLujo: cuando un jugador cae en ella, lo único que hace es quitarle un
- Clase Quini6: representa una lotería. Al caer en ella, el jugador puede o no recibir dinero, depende de cuantas veces cayó previamente. La primera vez recibe 50000, la segunda 30000 y para el resto del juego ya no recibe dinero.
- Clase Salida: donde comienzan todos los jugadores. No tiene efecto alguno.
- Clase Dinamismo: esta clase abstracta es la representación de un casillero que posee la particularidad de ser dinámica. Con esto nos referimos a que cuando un jugador cae en estas casillas dinámicas, se establece un comportamiento específico, el cual hace que el jugador pueda avanzar o retroceder en el tablero.
- Clase AvanceDinamico: representa a un casillero que tiene la cualidad especial de hacer avanzar al jugador en el tablero dependiendo del número de dados que saco. Si el jugador en su partida caen en este casillero entonces debera avanzar cuanto se lo diga.
- Clase RetrocesoDinamico: representa a un casillero que tiene la cualidad especial de hacer retroceder al jugador en el tablero dependiendo del numero de dado. Si el jugador en su partida cae en este casillero entonces debera retroceder cuanto se lo diga.

Paquete Vista:

- Clase ContenedorEntrada : visualización de la presentación del juego dónde los tres jugadores pueden elegir sus nombres y el color de sus piezas, los nombre ingresados son luego utilizados para iniciar el juego.
- Clase ContenedorPrincipal : visualización del juego, este posee al tablero, la consola dónde se puede ver toda acción desarrollada, y por último a la derecha los botones que puede utilizar el jugador.
- Posición : controla la siguiente posición del jugador, tal que devuelve la nueva ubicación de este. También se utiliza al presentar el tablero, siendo esta una manera más automática de formarlo.
- VistaJugador : visualización del jugador, representado por un círculo del color elegido en el contenedor de entrada. Conoce su posición y se mueve debido a Posición.
- Paquete Controladores: El paquete controladores posee por una parte el conjunto de todos los botones manejados en el juego: -Tirar Dados -Comprar -Vender -Edificar casa -Edificar Hotel -Pagar Fianza (para poder salir de la cárcel) -Mute (para poder silenciar la música de fondo)

Siempre que se clickea un botón, es decir, se activa un handler de eventos, el visor y la vista del tablero se actualiza para poder estar al día con las últimas acciones del usuario. Por ejemplo, el pagar la fianza hace que el visor del usuario informe que ya no está más preso. Esto se logra en gran parte gracias a que el controlador de turnos es una clase Singleton.

Y por otro lado:

- ControladorDeTurno : esta es una clase Singleton, ya que hay solamente un juego en desarrollo y por ejemplo, los botones necesitan en todo momento saber quién lo está seleccionando, presuponiendo que lo toca quién esté jugando en el momento.
- EntradaUsuario : guarda toda información ingresada por el usuario en la entrada del juego.

4. Detalles de implementación

Paquete Modelo:

- Clase Jugador: es instanciado con un capital total de 100000, su casillero actual es null y su estado es una nueva instancia de libre. Recibe el nombre por parámetro. Todos los métodos relacionados con las propiedades, ya sean comprar, vender al banco, entre las más importantes, acceden al Municipio que se encarga de actualizar los respectivos datos. Para saber si el jugador es capaz de moverse, se debe verificar si su estado es una instancia de Libre o Preso. En el primer caso, el jugador es libre de moverse y puede lanzar los dados, en el caso contrario debe permanecer en esa casilla hasta que se ejecuten las acciones necesarias. Por ejemplo, si el jugador cayó en Carcel, es incapaz de moverse hasta que no pasen los días necesarios o se pague la fianza. Para lograr esto se uso el patron de diseño State ya que consideramos necesario que el comportamiento del jugador cambiara segun si puede o no moverse.
- Clase Municipio: el municipio fue creado como un Singleton debido a que necesitábamos un único lugar donde almacenar y actualizar la información correspondiente a las propiedades, el cual pudiera ser accedido por instancias de Jugador, tanto por Barrio y Servicio, por los controladores correspondientes a los botones y por Visor. Al usar un Singleton, pudimos garantizar un único punto de acceso a esa base de datos y encapsular la responsabilidad de

conocer, actualizar y otorgar permisos para los métodos relacionados a las propiedades y sus propietarios.

Con respecto al resto de la implementación, posee cinco diccionarios de los cuales 3 son actualizados al comienzo con la información de un archivo de texto y el resto se van actualizando a medida que se requiere su uso con los datos que reciben.

El diccionario propietarios: posee como clave el nombre de la propiedad y como valor la instancia del jugador al que le pertenece.

El diccionario hermano: posee como clave el nombre de la propiedad y su propiedad hermana.

El diccionario alquilerServicio: posee como clave el nombre del servicio y una dupla que contiene en primer lugar el servicio que se compra al tener la propiedad y en segundo lugar, el servicio que se cobra al tener la propiedad hermana.

El diccionario jugadorPropiedades: posee como clave el nombre del jugador y una lista de todas las instancias de propiedades que le pertenecen.

El diccionario cantEdificaciones: posee como clave el nombre de la propiedad y la cantidad de edificaciones que tiene hasta el momento.

- Clase Tablero: al igual que el Municipio, el tablero fue creado como un Singleton debido a que varias clases necesitan cambiar la posición del jugador y para encapsular ese comportamiento decidimos crear una única clase que pudiera ser accedida globalmente. Además, facilitó bastante la lectura del código ya que las clases no necesitan recibir al tablero como parámetro. Posee una lista en la que se almacenan todas las instancias creadas de los barrios.

Paquete Casilleros (todos implementan Casillero):

- Clase Barrio: posee atributos para el nombre del propietario, el nombre del barrio, su costo, la lista de casas que se hayan edificadas, el alquiler actual (que se actualiza con cada casa edificada) y el alquiler original (el alquiler con el que se creó el barrio), el valor del mercado (la cantidad de dinero que recibe el jugador al venderlo). Los métodos relacionados a comprar, vender y edificar se encargan de restarle o sumarle la cantidad correspondiente al capital del jugador y actualizar los datos en el municipio.
- Clase BarrioDoble: hereda de BarrioDoble y puede reemplazar a las casas por un hotel. Para poder edificar un hotel, debe comprobar que el jugador es usuario tanto de este barrio como del otro barrio que corresponde.
- Clase Servicio: posee atributos para costo, costo del mercado y nombre. Para calcular el cobro del servicio se verifica el propietario en la Municipalidad.
- Clase Carcel: la acción que se ejecuta cuando un jugador cae en este casillero es cambiarle su estado a una nueva instancia de Preso. Esto se puede ver más detallado en la parte donde se explica la implementación de Jugador.
- Clase Dinamismo: esta clase abstracta establece la acción que se va a ejecutar cuando un jugador caiga en el casillero y establece tres métodos abstractos para que sus hijas definan. Así logrando que cada casillero sepa cómo moverse dinámicamente por sí sola, logrando evitar código repetido.
- Clase AvanceDinamico: esta es una clase que hereda de Dinamismo. En la cual se definen los 3 métodos necesarios para poder calcular a qué casillero debe avanzar el jugador dependiendo del número que sacó en el dado. La acción que se ejecuta cuando el jugador cae en este casillero dinámico es que avance una cierta cantidad de casilleros más.
- Clase RetrocesoDinamico: esta es una clase que hereda de Dinamismo. En la cual se definen los 3 métodos necesarios para poder calcular a qué casillero debe retroceder el jugador dependiendo del número que sacó en el dado. La acción que se ejecuta cuando el jugador cae en este casillero dinámico es que retroceda una cierta cantidad de casilleros más.

5. Diagramas de clase

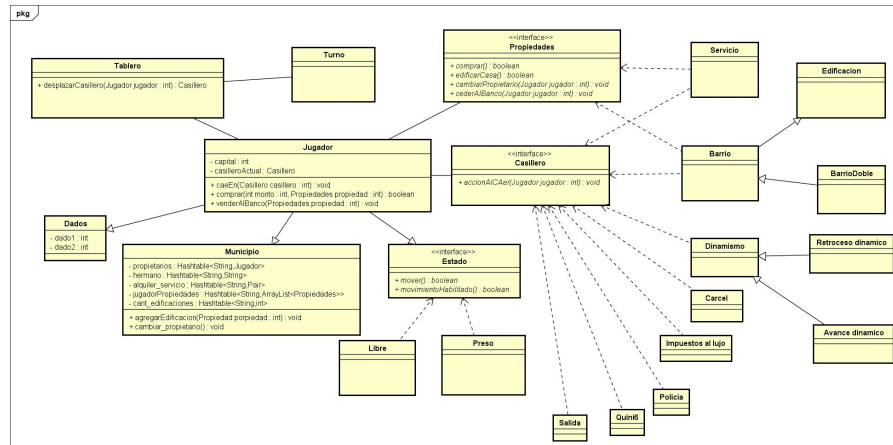
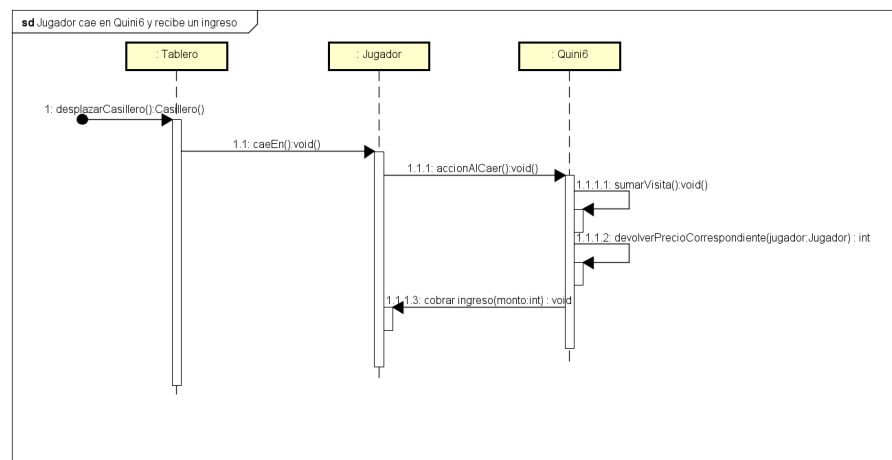


Figura 1: Diagrama de Clase.

6. Diagramas de Secuencia



powered by Astah

Figura 2: Un jugador cae en Quini6.

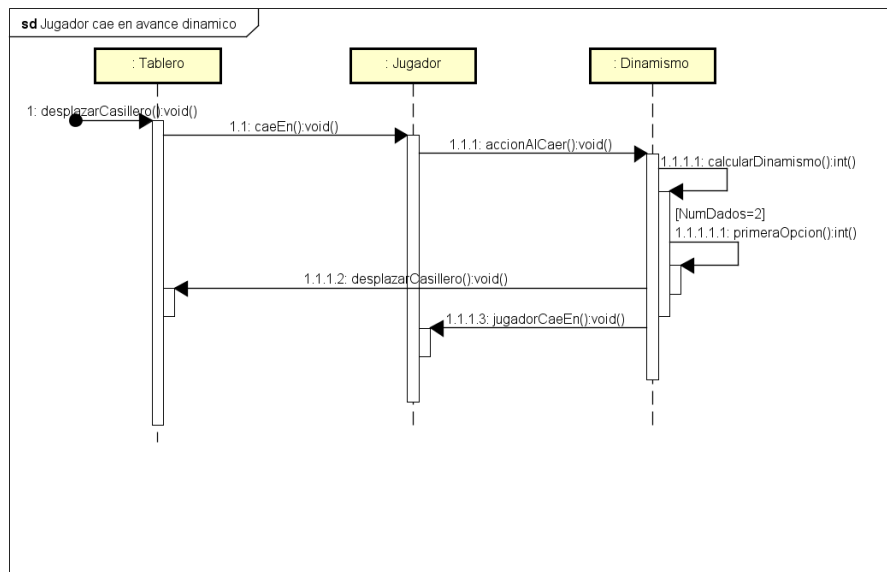


Figura 3: Un jugador cae en AvanceDinamico.

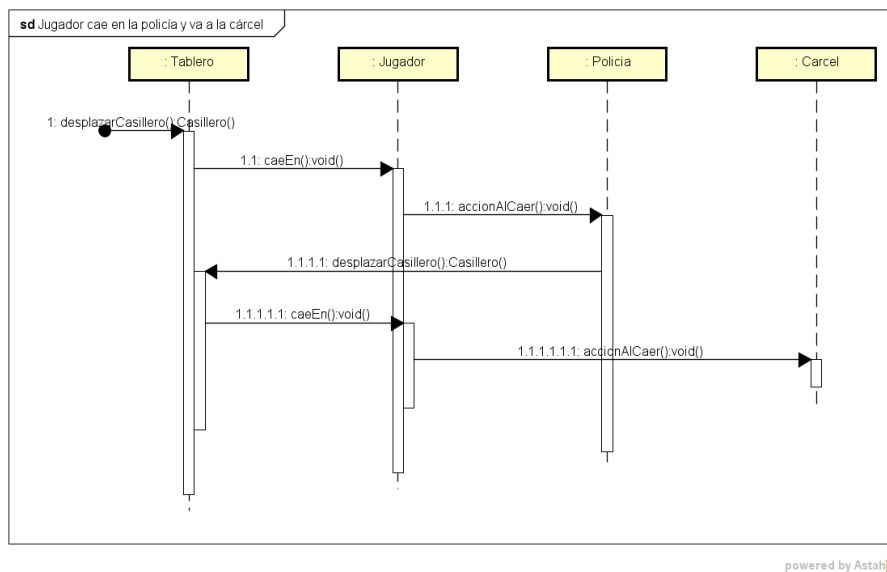


Figura 4: Un jugador cae en Policia.