

## Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III

Curso 1

Segundo cuatrimestre de 2017

Nombre	Padron	Mail
Gutiérrez, Matías	92172	matiasgutierrez@outlook.com
Federico Elias	96105	fedelias93@hotmail.com
Maria Soledad Escobar	97877	mariasolescobar21@gmail.com
Sebastian Luraschi	97177	sebastian.luraschi8@gmail.com

Fecha de entrega final: 30 de noviembre 2017

Tutor: Marcio Degiovannini

Comentarios:

## Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Objetivo del trabajo . . . . .	2
1.2. Consigna general . . . . .	2
1.3. Descripción de la aplicación a desarrollar . . . . .	2
<b>2. Supuestos</b>	<b>6</b>
<b>3. Modelo de dominio</b>	<b>6</b>
<b>4. Diagramas de clase</b>	<b>7</b>
<b>5. Detalles de implementación</b>	<b>9</b>
<b>6. Excepciones</b>	<b>10</b>
<b>7. Diagramas de secuencia</b>	<b>11</b>
<b>8. Diagrama de Paquetes</b>	<b>13</b>

# 1. Introducción

## 1.1. Objetivo del trabajo

Aplicar los conceptos enseñados en la materia a la resolución de un problema, trabajando en forma grupal y utilizando un lenguaje de tipado estático (Java)

## 1.2. Consigna general

Desarrollar la aplicación completa, incluyendo el modelo de clases e interfaz gráfica. La aplicación deberá ser acompañada por pruebas unitarias e integrales y documentación de diseño. En la siguiente sección se describe la aplicación a desarrollar.

## 1.3. Descripción de la aplicación a desarrollar

Se deberá desarrollar una aplicación que implemente un juego relacionado con el clásico juego de mesa MonoPoly. En el juego habrá 3 jugadores. Cada jugador comenzará con la misma cantidad de dinero equivalente a \$100000 (cien mil pesos) desde la posición de salida. El tablero tendrá la siguiente disposición:

IMPUESTO DE LUJO	SANTA FE	AYSA	SALTA NORTE	SALTA SUR	POLICÍA
CORDOBA NORTE	AlgoPoly				TREN
SUBTE					NEUQUÉN
AVANCE DINÁMICO					RETROCESO DINÁMICO
CORDOBA SUR					TUCUMÁN
CÁRCEL	Bs. As. - ZONA NORTE	EDESUR	Bs. As. - ZONA SUR	QUINI 6	<= SALIDA

### Casilleros:

- Salida: Los 3 jugadores comienzan el juego en el mismo lugar. Se elige aleatoriamente quién tira primero, segundo y tercero.
- Quini 6: El jugador que caiga en esta casilla recibe un premio de \$50000 (cincuenta mil pesos). Si ese jugador ya ganó una vez el Quini 6, la segunda vez que caiga en esa casilla recibe un premio de \$30000 (treinta mil pesos). Las sucesivas veces que el mismo jugador caiga en este casillero no cobrará ningún dinero.
- Buenos Aires Sur:
  - Precio terreno: \$20000
  - Alquiler: \$2000
  - Alquiler con 1 casa: \$3000
  - Alquiler con 2 casas: \$3500
  - Alquiler con Hotel: \$5000

- Construir casas cuestan \$5000 y hotel \$8000
4. EDESUR:
- Comprar la compañía: \$35000
  - Cobra 500 veces lo que dice los dados. O sea que si un jugador no dueño de la empresa cae en esa casilla tras haber sacado 12 en los dados =>deberá pagar:  $12 \times 500 = \$6000$   
Si también tiene la compañía AYSA cobrará 1000 veces lo sacado en los dados.
5. Buenos Aires Norte:
- Precio terreno: \$25000
  - Alquiler: \$2500
  - Alquiler con 1 casa: \$3500
  - Alquiler con 2 casas: \$4000
  - Alquiler con Hotel: \$6000
  - Construir casas cuestan \$5500 y hotel \$9000
6. Cárcel: un jugador al caer en este casillero debe esperar 3 turnos para salir. Es decir, recién cuando le toque por 4ta vez podrá moverse. Salvo que cuando ya haya pasado 1 turno de la cárcel (o sea en el turno 2 y 3 de espera) pague una fianza de \$45000.
7. Córdoba Sur:
- Precio terreno: \$18000
  - Alquiler: \$1000
  - Alquiler con 1 casa: \$1500
  - Alquiler con 2 casas: \$2500
  - Alquiler con Hotel: \$3000
  - Construir casas cuestan \$2000 y hotel \$3000
8. Avance dinámico: El jugador avanzará tantos casilleros como lo indica la siguiente lógica:
- a) Si sacó 2, 3, 4, 5 o 6 =>entonces avanza el número sacado menos 2 unidades
  - b) Si sacó 7, 8, 9 o 10 =>entonces avanza #cantidad de efectivo del jugador % numero sacado
  - c) Si sacó 11 o 12 =>avanza el numero sacado menos la sumatoria de propiedades del jugador.
- Tanto los terrenos como las casas como los hoteles suman como propiedad.
9. SUBTE:
- Comprar la compañía: \$40000
  - Cobra 600 veces lo que dice los dados. O sea que si un jugador no dueño de la empresa cae en esa casilla tras haber sacado 12 en los dados =>deberá pagar:  $12 \times 600 = \$7200$   
Si también tiene la compañía TRENES cobrará 1100 veces lo sacado en los dados.
10. Córdoba Norte:
- Precio terreno: \$20000
  - Alquiler: \$1300
  - Alquiler con 1 casa: \$1800

- Alquiler con 2 casas: \$2900
  - Alquiler con Hotel: \$3500
  - Construir casas cuestan \$2200 y hotel \$3500
11. Impuesto al Lujo: El jugador que caiga en esta casilla debe pagar el 10 % de todo su efectivo.
12. Santa Fe:
- Precio terreno: \$15000
  - Alquiler: \$1500
  - Alquiler con 1 casa: \$3500
  - Construir la única casa posible cuesta \$4000
13. AYSA:
- Comprar la compañía: \$30000
  - Cobra 300 veces lo que dice los dados. O sea que si un jugador no dueño de la empresa cae en esa casilla tras haber sacado 12 en los dados =>deberá pagar:  $12 \times 300 = \$3600$   
Si también tiene la compañía EDESUR cobrará 500 veces lo sacado en los dados.
14. Salta Norte:
- Precio terreno: \$23000
  - Alquiler: \$2000
  - Alquiler con 1 casa: \$3250
  - Alquiler con 2 casas: \$3850
  - Alquiler con Hotel: \$5500
  - Construir casas cuestan \$4500 y hotel \$7500
15. Salta Sur:
- Precio terreno: \$23000
  - Alquiler: \$2000
  - Alquiler con 1 casa: \$3250
  - Alquiler con 2 casas: \$3850
  - Alquiler con Hotel: \$5500
  - Construir casas cuestan \$4500 y hotel \$7500
16. Policía: Al caer en este casillero el jugador va a la cárcel.
17. TRENES:
- Comprar la compañía: \$38000
  - Cobra 450 veces lo que dice los dados. O sea que si un jugador no dueño de la empresa cae en esa casilla tras haber sacado 12 en los dados =>deberá pagar:  $12 \times 450 = \$5400$ .  
Si también tiene la compañía SUBTES cobrará 800 veces lo sacado en los dados.
18. Neuquén:
- Precio terreno: \$17000
  - Alquiler: \$1800
  - Alquiler con 1 casa: \$3800

- Construir la única casa posible cuesta \$4800
19. Retroceso dinámico: El jugador avanzará tantos casilleros como lo indica la siguiente lógica:
- a) Si sacó 2, 3, 4, 5 o 6 =>retrocede el número sacado menos la sumatoria de propiedades del jugador. Tanto los terrenos como las casas como los hoteles suman como propiedad.
  - b) Si sacó 7, 8, 9 o 10 =>retrocede #cantidad de efectivo del jugador % numero sacado
  - c) si sacó 11 o 12 =>retrocede el número sacado menos 2 unidades
20. Tucumán:
- Precio terreno: \$25000
  - Alquiler: \$2500
  - Alquiler con 1 edificio histórico: \$4500
  - Construir la casita de tucumpán (única construcción posible) cuesta \$7000

**Jugabilidad:**

- Es un juego por turnos. Hay 3 jugadores. En cada turno cada jugador debe lanzar un par de dados (números del 1 al 6 cada dado). El número obtenido será la suma de ambos resultados.
- Si un jugador al tirar los dados obtiene un doble numero (o sea 1 y 1 , o 3 y 3 , etc) entonces tira nuevamente. Si vuelve a sacar otro doble, no tira otra vez, sino que le toca el turno al jugador siguiente.
- El jugador debe mover en el sentido que indica la flecha del casillero de salida.
- Los efectos ocurren al caer en un casillero y no al pasar por ellos.
- Para poder edificar tanto en Córdoba como en Bs.As como en Salta es necesario que el jugador compre ambas zonas (norte y sur) para recién poder edificar.
- No se puede edificar en ningún terreno en el mismo turno en que se lo compra
- Antes de lanzar los dados, cada jugador tiene la posibilidad de edificar si se encuentra en condiciones de hacerlo: Ya sea tanto contar con el dinero que sale + tener el terreno.
- Para poder edificar un hotel, debe estar lleno de casas en su capacidad máxima. Es decir que si un jugador tiene 2 casas en Bs. As. sur y sólo 1 casa en Bs. As Norte NO puede construir hotel. Recién al contar con 2 casas en Bs. As sur y 2 casas en Bs. norte puede empezar a construir los hoteles (máximo 1 por cada terreno).
- Los hoteles reemplazan a las casas, es decir que cuando se edifica un hotel las 2 casas desaparecen.
- Los terrenos que no son dobles (es decir que no tiene un Norte y Sur) no pueden construir hoteles.
- Un jugador puede decidir vender tanto sus terrenos (con las casas u hoteles que tenga edificados) o las compañías antes de lanzar los dados. En ese caso el jugador perderá la titularidad de los mismos, los cuales pasan a estar disponibles para la compra para el resto de los jugadores. El jugador que vende cobra un 15% menos de la suma de todo lo que vende. Al venderse un terreno que tenía edificaciones las mismas desaparecen del mapa. Es decir que cuando un nuevo jugador compre ese terreno, estará vacío y deberá volver a construir las casas y luego los hoteles.

**Fin del juego:**

1. Cuando un jugador se quede sin dinero y sin propiedades quedará eliminado.
2. Si un jugador no tiene dinero, pero sí propiedades y debe afrontar un gasto, está obligado a vender (con la quita del 15 % ya explicada) para afrontar esos gastos. Si aún vendiendo todo, no puede afrontar el gasto, el jugador quedará eliminado también.
3. Ganará el último jugador restante.

## 2. Supuestos

A continuación se detallan los supuestos adoptados a lo largo del desarrollo del proyecto.

1. Antes de tirar los dados el jugador puede vender propiedades o bien construir en ellas cuantas veces lo desee.
2. Cuando el jugador cae en la cárcel, después del segundo turno que es cuando ya puede pagar fianza para salir, la paga automáticamente al tirar los dados, es decir, no se le pide confirmación de si quiere pagar la fianza.
3. El jugador no pierde el juego si no puede pagar la fianza para salir de la cárcel.
4. Cuando el jugador vende una propiedad con propiedad conocida, si ambas tienen edificaciones, al momento de vender una pierde las edificaciones de la propiedad que le queda. Se entiende que esto es también una regla del juego original.
5. El jugador no puede construir ni vender propiedades hasta que haya salido de la cárcel, aunque tenga las condiciones de pagar fianza. Es decir, el jugador paga fianza sólo al tirar los dados
6. El juego fija la cantidad de jugadores a tres

## 3. Modelo de dominio

Para modelar una solución al problema se aprovechó el patrón de comportamiento *Visitor*, donde el jugador "visita.<sup>a</sup> las celdas y estas determinan su comportamiento. Además, se usó el patrón *Command* para mostrarle al jugador las operaciones que puede realizar sobre las propiedades que posee, ya sea vender o construir casas o hoteles. Para modelar la situación de caer en la cárcel se usó el modelo de cambio de estados, donde el jugador puede tener estado normal y estado en la cárcel

A continuación se listan las Clases involucradas en la resolución del Trabajo Práctico, con una breve descripción de sus responsabilidades.

**AlgoPoly:** Clase principal. En la misma se crean las clases que componen el juego y se inicializa el juego.

**Jugador:** Clase que implementa la interfaz visitante. Puede jugar moviéndose entre las celdas del tablero.

**EstadoJugador:** Interfaz que encapsula el comportamiento de un jugador dependiendo del estado en el que se encuentra. De la misma heredan los estados *EstadoJugadorNormal* y *EstadoJugadorCarcel*.

**EstadoJugadorNormal:** Clase asociada al comportamiento de un jugador "normal", esto es un jugador que no está preso y puede moverse libremente por el tablero según las reglas del juego.

**EstadoJugadorCarcel:** Clase asociada al comportamiento de un jugador preso.

**Tablero:** Es la clase contenedora de las celdas que pueden ser visitadas por los jugadores. Recibe al jugador y hace que el mismo visite la celda que corresponda.

**turno:** Es la clase encargada de hacer que un jugador juegue un turno. Se encarga de tirar los dados y comunicarle al jugador que debe avanzar en el tablero visitando las celdas en la que cae.

**Visitable:** Interfaz asociada al patrón Visitor. Es de la cual extienden absolutamente todas las celdas del tablero, pues son las celdas las que pueden ser visitadas por cada jugador.

**Visitante:** Interfaz asociada al patrón Visitor. Es de la cual extiende la clase Jugador, pues es el que visita las celdas del Tablero.

**Propiedad:** Esta clase Abstracta es de la cual heredan las celdas que pueden ser adquiridas por un jugador, tanto Barrios como Servicios. En la misma se encuentran los métodos comunes entre Barrio y Servicio. A su vez la misma implementa la interfaz Visitable.

**Servicio:** Esta clase es de la cual heredan todas las clases de servicios como EDESUR, TREN, SUBTE y AYSA. La cual a su vez extiende de Propiedad e implementa la interfaz Visitable.

**Barrio:** Esta clase es de la cual heredan todas las clases de barrios como Buenos Aires, Córdoba, Salta, tanto Norte como Sur, Neuquén, Santa Fe y Tucumán. La misma a su vez extiende de Propiedad e implementa la interfaz Visitable.

**MovimientoDinamico:** Es la clase en la cual se encapsula el calculo a realizarse cuando un jugador cae en una celda como Avance o Retroceso dinámico. La misma es un atributo dentro de estas clases.

**Command:** Es la clase abstracta de la cual heredan cada una de las opciones posibles sobre las propiedades encapsuladas en un

**Dado:** Esta clase es la encargada de simular el tiro de dados. La misma es invocada desde la clase turno la cual es la única con la que se relaciona directamente.

El uso de estos patrones se comprenderá mejor al analizar los siguientes diagramas de clase y de secuencia.

## 4. Diagramas de clase

En el diagrama 1 se presenta un diagrama general de la solución, es importante destacar las interfaces *Visitable* y *Visitante*, importantes en la implementación del patrón *Visitor*. Para no dificultar la comprensión del diagrama no se incluyeron todas las clases que implementan la interfaz *Visitable* y se remarcó el paquete *especiales* indicando que todas las clases que pertenezcan al mismo, implementaran esta interfaz. También se dejó afuera lo relacionado a las celdas que el jugador puede adquirir, cuya estructura se muestra en la figura 2



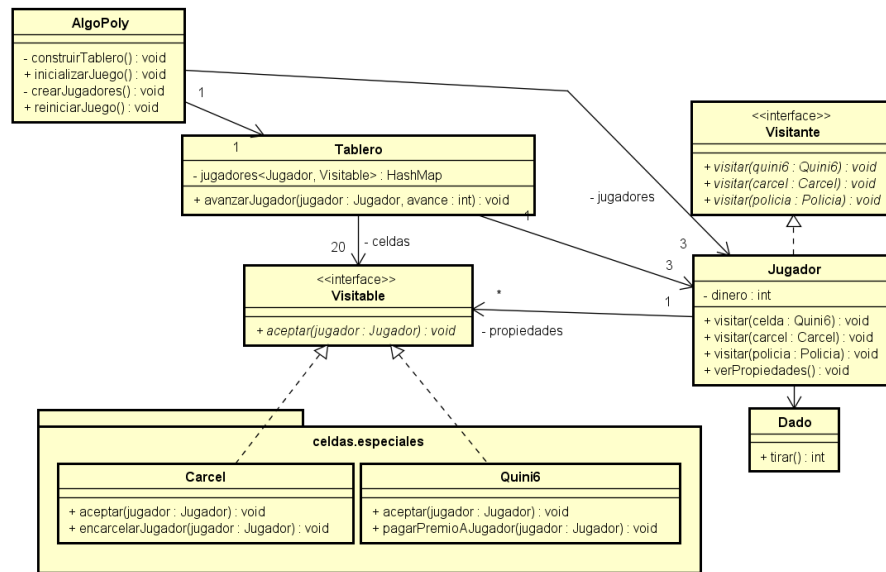


Figura 1: Diagrama general de clases.

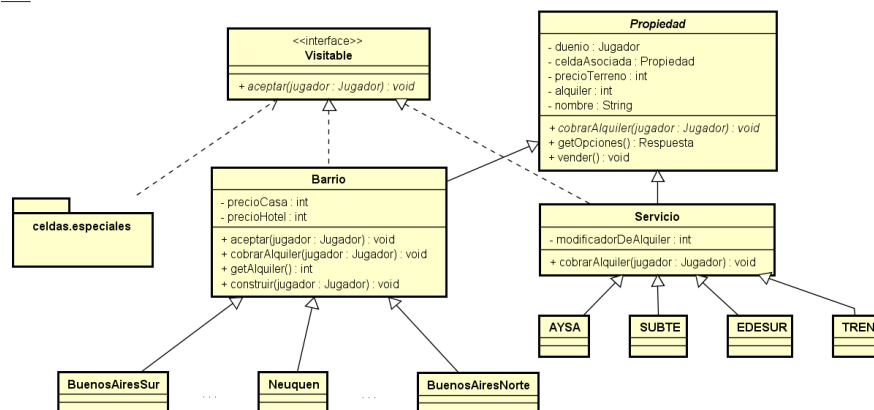


Figura 2: Diagrama de clases de celdas y propiedades.

El próximo diagrama demuestra la implementación tipo *Double dispatch* que se da entre las celdas y el jugador, específicamente con la celda Quini6.

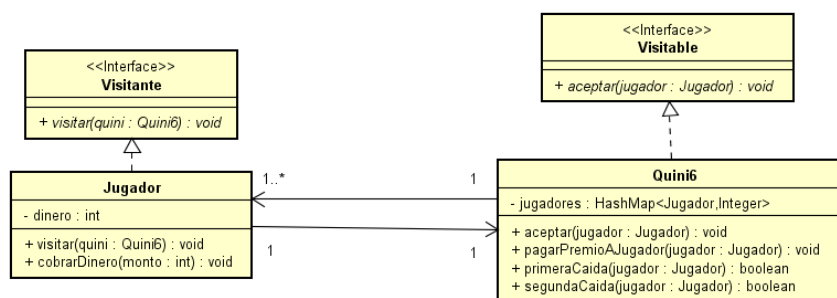


Figura 3: Diagrama que representa las clases y métodos utilizados para la funcionalidad del Quini6.

## 5. Detalles de implementación

- Para la implementación de la cárcel definimos una clase abstracta de estado para jugador, **EstadoJugador**, la cual la implementan **EstadoJugadorNormal** y **EstadoJugadorCarcel** mostrado en la figura 4. La primera realiza las acciones del jugador (vender, construir, moverse) normalmente de acuerdo a la implementación de cada una, en cambio la segunda implementa el comportamiento de un jugador que se encuentra en la cárcel, ya que este no debería poder vender, construir ni moverse mientras esté. La clase **EstadoJugadorCarcel** verifica los turnos restantes para que el jugador pueda realizar acciones ya sea pagando fianza o saliendo de la cárcel porque terminaron los turnos que corresponden. La responsabilidad está dividida entre **Carcel** y **EstadoJugadorCarcel**, **Carcel** sólo se encarga de meter, sacar o sacar con fianza al jugador, mientras que **EstadoJugadorCarcel** es la que realiza el comportamiento del mismo. En caso de que el jugador pueda realizar acción pagando fianza, el estado le manda un mensaje a la clase **Carcel** para que esta le cobre la fianza al jugador, ya que es la cárcel la que sabe cuanto debe pagar de fianza.

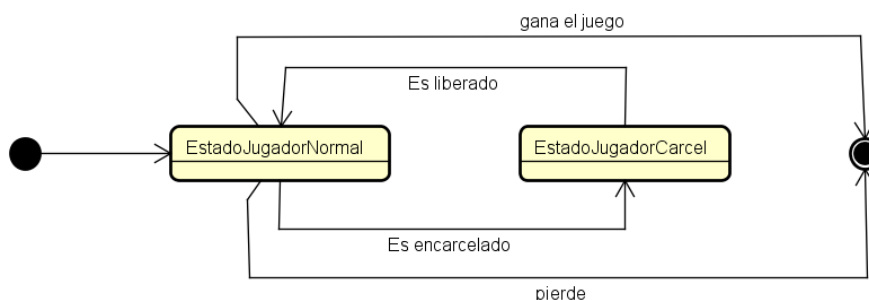


Figura 4: Diagrama de estados del jugador.

- El jugador pasando por cada celda se implementó con patrón de diseño "Visitor", donde la interfaz **Visitante** la implementa el jugador y la interfaz **Visitable** la implementan todas las celdas del tablero. Cada celda tiene el método para aceptar el **Jugador**, que invoca al método `visitar` del mismo, el cual se comporta distinto para cada celda.

- Para diferenciar de las celdas comprables entre los barrios y los servicios (tren, subte, etc.) se definió la clase Propiedad que encapsula las responsabilidades comunes de los barrios y las empresas, entre estas compra, alquiler, venta, precio de compra, verificar dueño, etc. definiendo como abstractos los métodos de cobrar alquiler y devolver las opciones disponibles de cada uno.
- Para la construcción de casas/hoteles en barrios, ya sean barrios simples o barrios compuestos, el barrio sabe la cantidad máxima de casas y hoteles que se pueden construir en él, también verifica que si ha alcanzado el límite de casas la siguiente construcción será un hotel, y si ya alcanzó el límite de hoteles no puede construir ninguno de los dos. En los barrios compuestos, cada barrio Norte conoce al barrio Sur correspondiente y viceversa, de esa forma el barrio sobre el que se quiere construir consulta a su asociado si tiene el mismo dueño antes de construir.
- Cuando el jugador cae en la celda Policía, esta debe mandarlo a la celda Cárcel, y ahí cambiar el comportamiento correspondiente. Para ello se definió un método de Jugador para indicarle a Tablero que debe posicionarlo en una celda específica, con lo cual cuando el policía al aceptar al jugador invoca a este método del mismo indicando que la celda a la que debe posicionarse es Cárcel.
- Cada Propiedad, ya sea Barrio o Servicio, cobra un alquiler al jugador que cae y si no puede pagarlo debe forzosamente vender propiedades que tenga, a elección. El usuario debe elegir esto por consola así que definimos un método dentro de Propiedad para crear un Alert con las propiedades que el jugador posea, luego elige la que va a vender y la Propiedad vuelve a intentar cobrarle el alquiler.

## 6. Excepciones

**JugadorNoTieneFondosParaPagarException:** Se utiliza para validar que el jugador tenga suficiente dinero para poder completar la acción que desea, ya sea pagar un alquiler, la fianza de la cárcel, comprar o construir.

**JugadorNoPuedeSalirDeLaCarcelException:** Se lanza cuando se intenta sacar al jugador de la cárcel y aún no ha pasado la cantidad mínima de turnos.

**JugadorNoCuentaConDineroSuficienteParaComprarException:** Se lanza cuando un jugador intenta comprar una propiedad pero no cuenta con el dinero suficiente.

**BarrioSimpleNoPuedeConstruirHotelException:** Se lanza cuando se intenta construir mas de una vez en un barrio simple, el cual admite una única construcción.

**JugadorFueTrasladadoALaCarcelException:** Se lanza esta excepción para informar al jugador que al caer en la celda policía el mismo sera trasladado a la cárcel.

**JugadorPerdioException:** Se lanza cuando un jugador pierde el juego. Lo cual sucede cuando afronta un gasto que no puede solventar.

**PropiedadConDuenioException:** Se lanza cuando un jugador cae en una propiedad con dueño, de esta manera se la atrapa y se hace el cobro del alquiler correspondiente.

**ConsultarCupraException:** Esta excepción es lanzada para poder indicarle al jugador que ha caído en una propiedad sin dueño y puede comprarla.

**JugadorNoTieneFondosParaConstruirEnTerrenoException:** Esta excepción se lanza al intentar construir en un terreno sin dinero para pagar el costo de la casa u hotel.

**FinDelJuegoException:** Se lanza cuando queda un único jugador en el juego. Se utiliza para indicar el ganador del juego.

## 7. Diagramas de secuencia

Primeramente se presentan, en las figuras 5 y 6, como se calculan avance y retroceso en las celdas dinámicas.

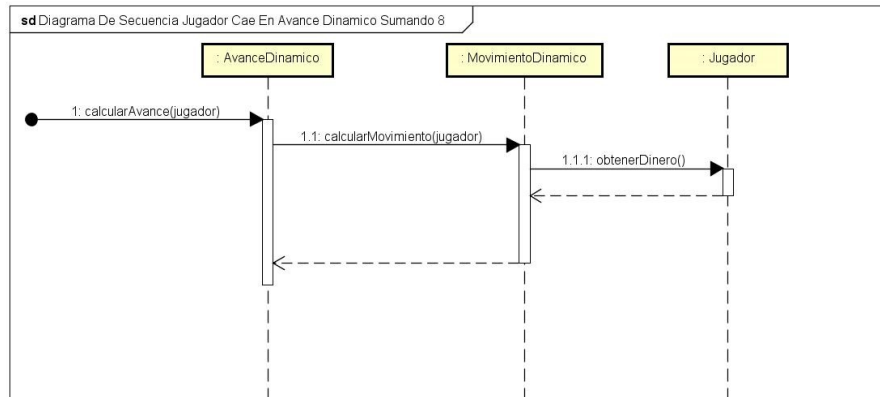


Figura 5: Diagrama de secuencia de Avance Dinamico.

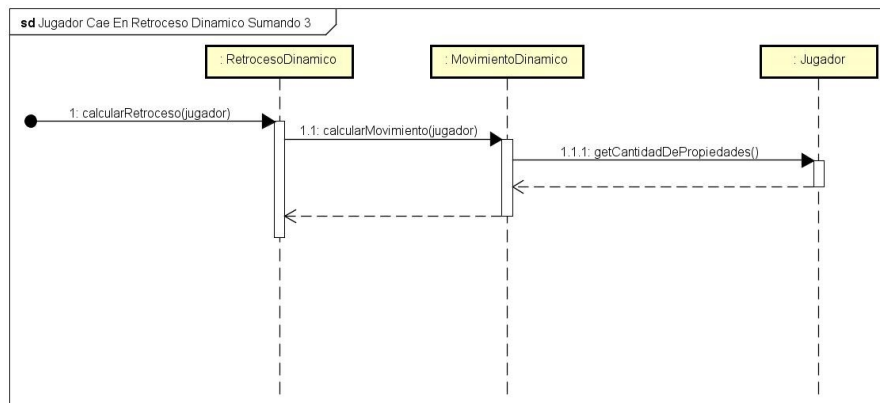


Figura 6: Diagrama de secuencia de Retroceso Dinámico.

Luego en la figura 7 se muestra un diagrama completo de la interacción entre el jugador y la celda Quini6.

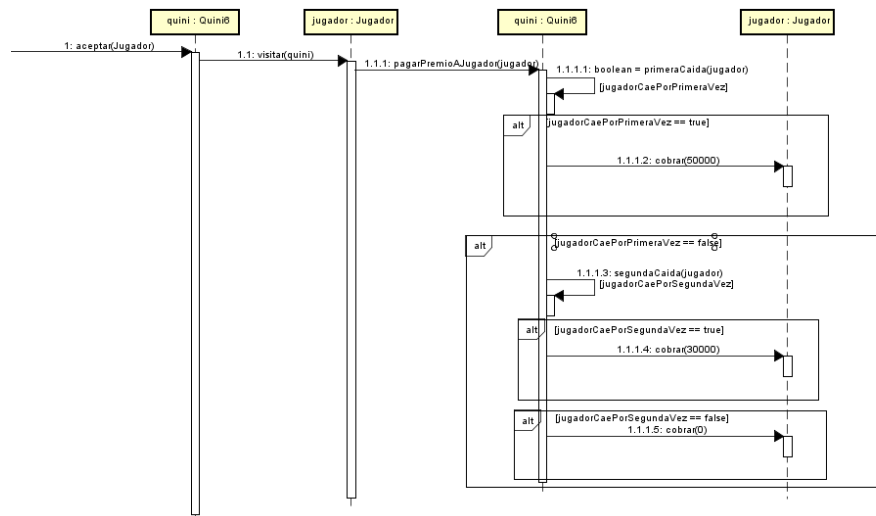
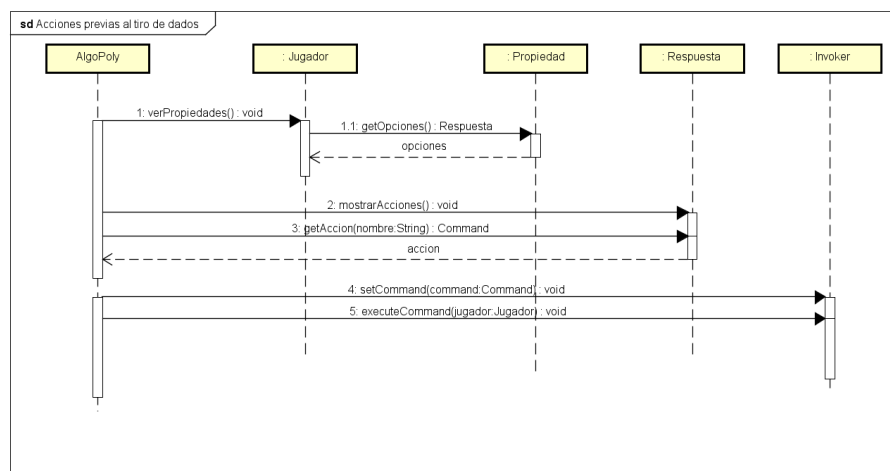


Figura 7: Diagrama de secuencia de la funcionalidad completa de Quini6, que representa el momento donde un jugador cae por primera vez, por segunda o más, cobrando lo correspondiente en cada caída.

A continuación en la figura 8 se muestra el proceso por el cual el jugador selecciona una de las posibles acciones sobre sus propiedades, es decir, vender, y construir cuando corresponda.



powered by Astah

Figura 8: Diagrama de secuencia de acciones posibles.

Por último, en la figurar 9 se muestra la estructura general del turno del jugador.

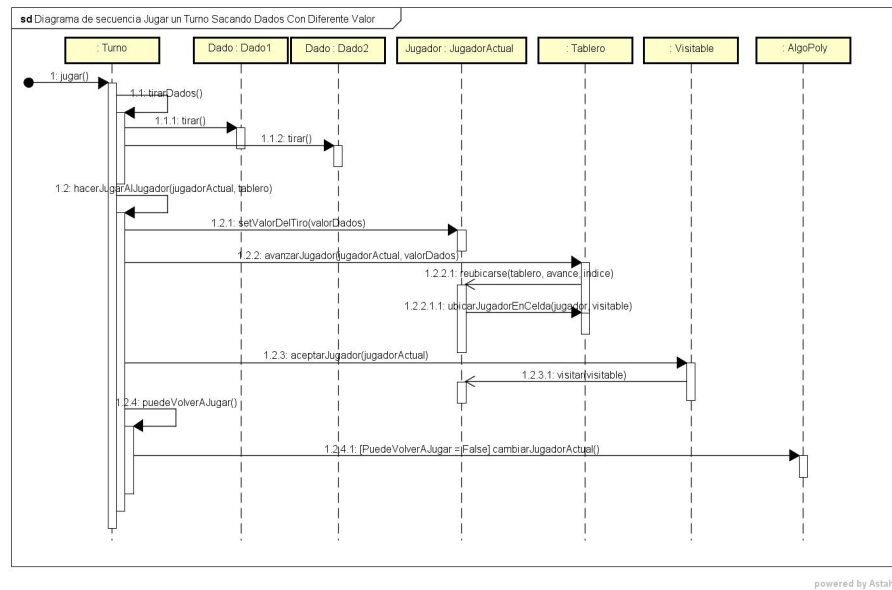


Figura 9: Diagrama de secuencia Jugar un turno.

## 8. Diagrama de Paquetes

Para complementar el análisis previo, se adjunta un diagrama de paquetes del proyecto realizado:

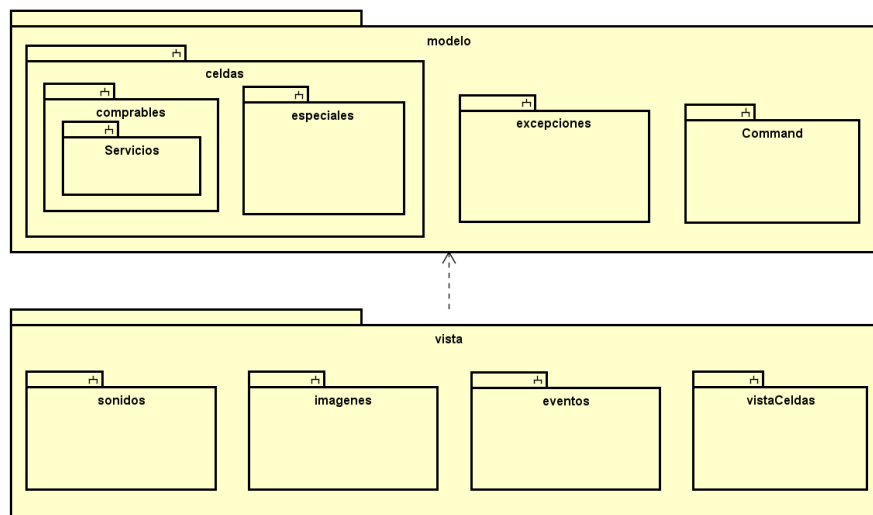


Figura 10: Diagrama de paquetes.