

# FIUBA - 7507

## Algoritmos y programación 3

### *Trabajo práctico 2: Batalla Navalgo*

1er cuatrimestre, 2013

(trabajo grupal)

Alumno:

Nombre	Padrón	Mail
Acevedo Facundo M.	94311	acv2facundo@gmail.com
Cerioti Denise	94369	denhjskf@gmail.com
Peña Maximiliano	93198	maxipenia@hotmail.com

Fecha de entrega: 12 de Julio de 2013

Tutor: Marcio Degiovannini

Nota Final:

## Tabla de contenidos

### [Introducción](#)

[-Objetivo del trabajo](#)

[Consigna general](#)

### [Descripción de la aplicación a desarrollar](#)

[-Contexto](#)

[Objetivo del juego](#)

[Puntos](#)

[Dinamica del juego](#)

[Disparos](#)

[Tablero](#)

[Naves](#)

### [Entregables](#)

[Forma de entrega](#)

### [Informe](#)

[-Supuestos](#)

### [Modelo de dominio](#)

[Diagramas de clases](#)

[Detalles de implementación](#)

[Excepciones](#)  
[Diagramas de secuencia](#)  
[Checklist de corrección](#)  
[Código](#)

## **Introducción**

### ***Objetivo del trabajo***

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

### ***Consigna general***

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

## **Descripción de la aplicación a desarrollar**

### ***Contexto***

La empresa Algo Ritmos SA dedicada al desarrollo de video juegos a decidido contratar a un grupo de programadores para implementar el juego Batalla Navalgo: un juego levemente inspirado en el clasico juego de Batalla Naval

### ***Objetivo del juego***

La Batalla Navalgo es un juego por turnos en el cual juega el jugador contra la computadora.

La computadora coloca sus naves en el tablero, y el jugador debe destruirlas. El jugador no tiene tablero, el juego consiste solamente en destruir las naves de la computadora.

### ***Puntos***

El jugador empieza el juego con una cantidad de puntos (10.000 puntos), y el objetivo es terminar el juego (destruyendo todas las naves) con la mayor cantidad de puntos.

Los puntos se descuentan de 2 formas:

- Con el simple paso del tiempo (10 puntos por turno)
- Con los disparos que hace el jugador (distinta cantidad de puntos de acuerdo al tipo de disparo)

### ***Dinamica del juego***

En cada momento el jugador puede ver el tablero de la computadora, junto con la ubicación y estado de las naves (al principio todas las naves estarán completas pero a lo largo del juego tendrán partes destruidas).

A diferencia de la batalla naval convencional, las naves en la batalla Navalgo tienen movimiento. Al colocar las naves en el tablero, la computadora asignará una dirección a cada una (8 direcciones posibles: lineales + diagonales).

Al pasar cada turno, las naves se mueven 1 posición en la dirección correspondiente.

Al "chocar" contra el borde del tablero, la dirección se invierte para continuar el movimiento en sentido opuesto.

Las naves no chocan entre sí, sino que se pueden superponer en el tablero. En ese caso, si el jugador efectúa un disparo sobre una posición, destruirá todas las partes de las naves que se encuentran superpuestas.

En cada turno del juego, el jugador podrá efectuar un disparo.

### ***Disparos***

Existen distintos tipos de disparo, con distinto costo y diferente comportamiento:

- Disparo convencional
  - Impacta en una posición, en el mismo momento que se hace el disparo
  - Costo: 200 puntos
- Mina submarina puntual con retardo
  - Se coloca la mina en una posición, pero el impacto se realiza 3 turnos después de haber sido colocada
  - Costo: 50 puntos

- Mina submarina doble con retardo
  - Idem anterior, pero al impactar destruye la posición actual + las posiciones adyacentes en un radio de 1 casilla
  - Costo: 100 puntos
- Mina submarina triple con retardo
  - Idem anterior, pero al impactar destruye la posición actual + las posiciones adyacentes en un radio de 2 casillas
  - Costo: 125 puntos
- Mina submarina por contacto
  - Se coloca la mina en una posición, pero el impacto se realiza cuando alguna nave pasa por sobre esa posición
  - Costo: 150 puntos

## ***Tablero***

El tablero es un cuadrado de 10 x 10 casilleros.

## ***Naves***

Las naves se diferencian por la cantidad de casilleros consecutivos que ocupan y su resistencia a los disparos.

La computadora puede colocar las naves en posición vertical u horizontal, y con alguna de las 8 de direcciones de movimiento posibles (lineales + diagonales).

La nave se considera destruida cuando se destruyen cada una de las partes (cada parte ocupa un casillero) de ellas.

La computadora colocará las siguientes naves en el tablero:

- 2 lanchas (naves de 2 casillas)
- 2 destructores (naves de 3 casillas, a las cuales solamente las dañan los disparos directos, las minas submarinas no les producen ningun daño)
- 1 buque (nave de 4 casillas, en la cual un impacto en cualquier parte de la nave la destruye por completo)
- 1 portaaviones (nave de 5 casillas)
- 1 rompehielos (nave de 3 casillas, pero en la cual cada parte requiere 2 impactos para ser destruida)

## **Entregables**

- Código fuente de la aplicación completa, incluyendo también: código de la pruebas, archivos de recursos
- Script para compilación y ejecución (ant o nant)
- Informe, acorde a lo especificado en este documento

## ***Forma de entrega***

A coordinar con el docente tutor.

## **Informe**

### ***Supuestos***

- El puntaje puede llegar a ser negativo.
- El juego no puede ser guardado antes de iniciarlo.
- Se puede guardar solo una sesion, la proxima vez que se guarde, sobrescribira a la anterior.

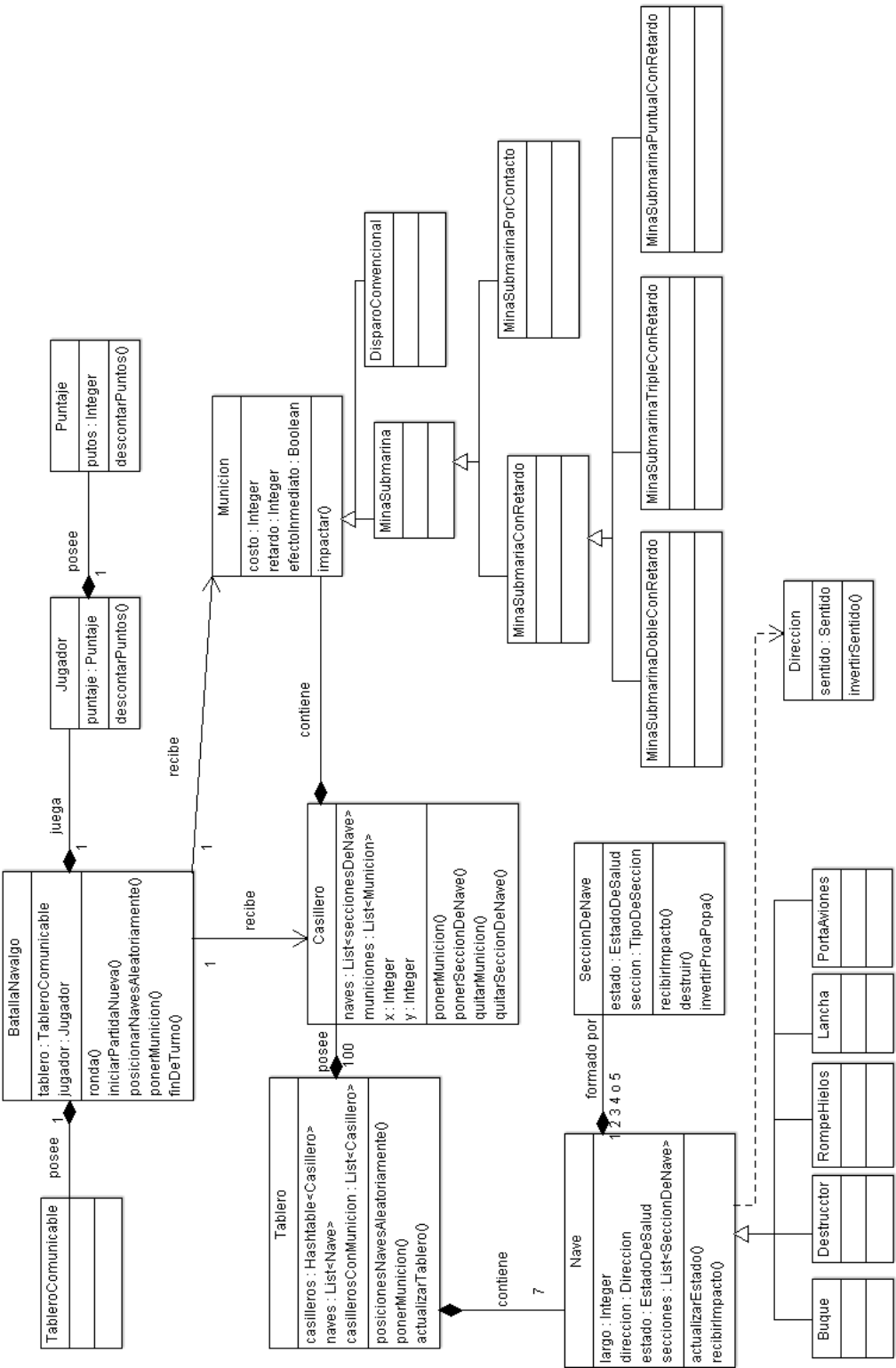
### **Modelo de dominio**

Para resolver el tp, las clases principales necesarias son: BatallaNavalgo, Tablero, Casillero, Nave, Jugador, Puntaje y Munición.

Pensamos a BatallaNavalgo como la clase principal del juego. Ésta contiene a TableroComunicable y a Jugador (que contiene Puntaje). BatallaNavalgo recibe Municion y el Casillero donde se desea realizar el impacto.

Las distintas municiones son clases que heredan de Munición, cada una con sus propiedades (Costo, tipo de impacto). Lo mismo con las naves, los diferentes tipos de naves heredan de Nave y cada una tiene su propio tamaño, tipo de impacto que requiere).

Diagramas de clases



## ***Detalles de implementación***

Las naves están conformadas por secciones, la sección media (que puede contar con ninguna, una o más), popa y proa. Tienen también un estado de nave: sana, dañada o destruída. Así como también cada sección tiene su propio estado.

Tablero está formado por una colección de Casilleros. Tablero sabe las municiones que hay en él y las naves sanas y/o dañadas. Tiene acceso a cada uno de sus Casilleros y a sus contenidos.

Cada casillero tiene una posición (x,y) y dos colecciones: una de municiones y otra de seccionDenave. Dentro de cada casillero puede haber más de una munición y más de una sección de nave.

Luego de cada turno, se actualiza el tablero: se revisan la colección de municiones de cada casillero y las que deben impactar en ese turno, lo hacen. Para generar el impacto, se revisa la colección de secciones de naves de ese casillero y se le hace el daño correspondiente.

## ***Excepciones***

Para resolver el trabajo práctico fue necesaria la excepción "idCasilleroInválido".

Se utilizó principalmente en la clase Tablero. Uno de los métodos de esta clase, verifica que la nave haya sido posicionada correctamente y que tenga casilleros libres al rededor para poder moverse, es decir, para que no quede bloqueada. Fue utilizada esta excepción para los casos en que los casilleros que rodean la nave son inválidos. También fue necesaria para el método obtenerCasillero y para obtenerAledanyos, que devuelve una lista con los casilleros cercanos al casillero pasado por parámetro.

## ***Diagramas de secuencia***

Diagrama de ronda() y ponerMunicion() :

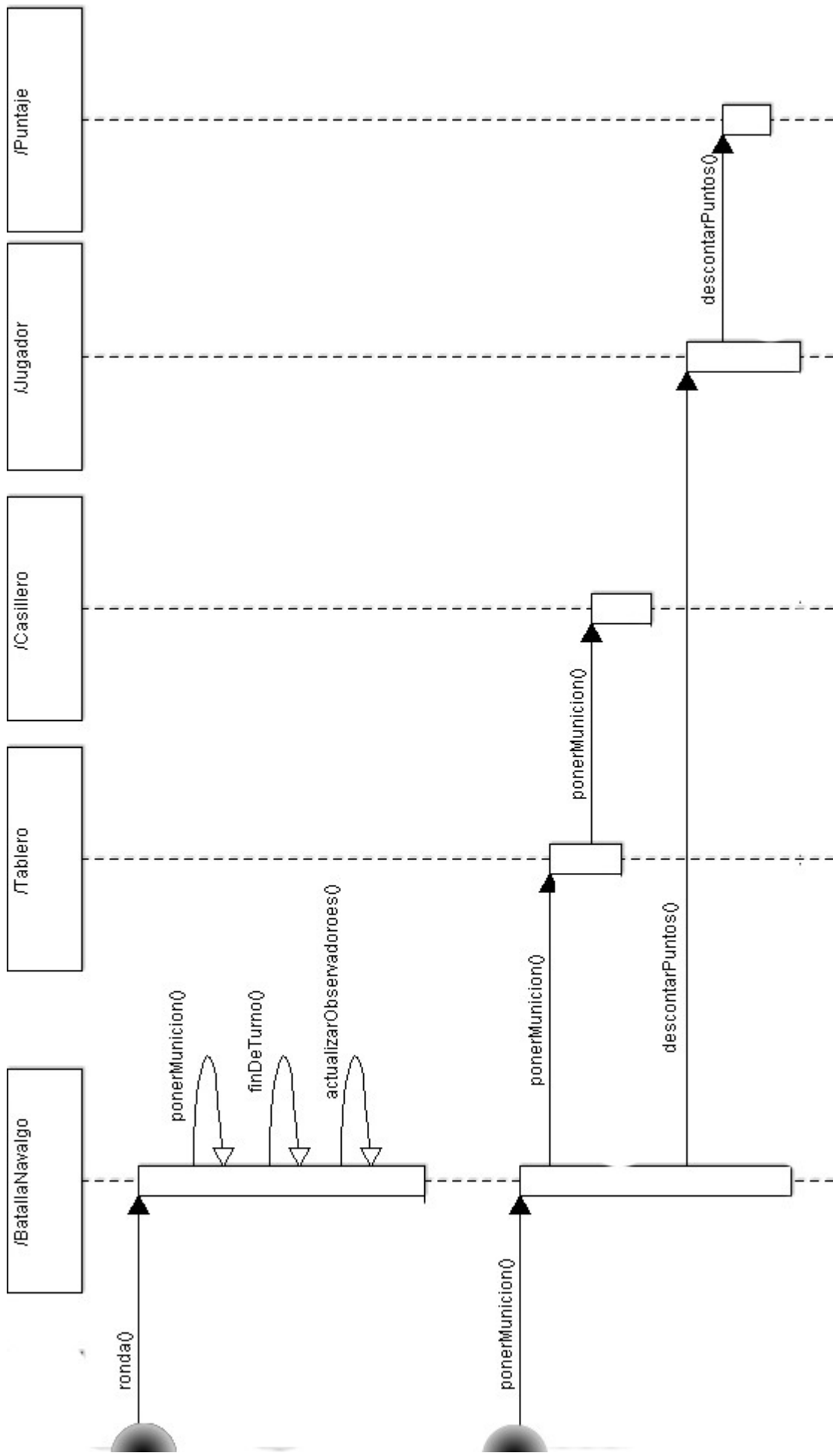
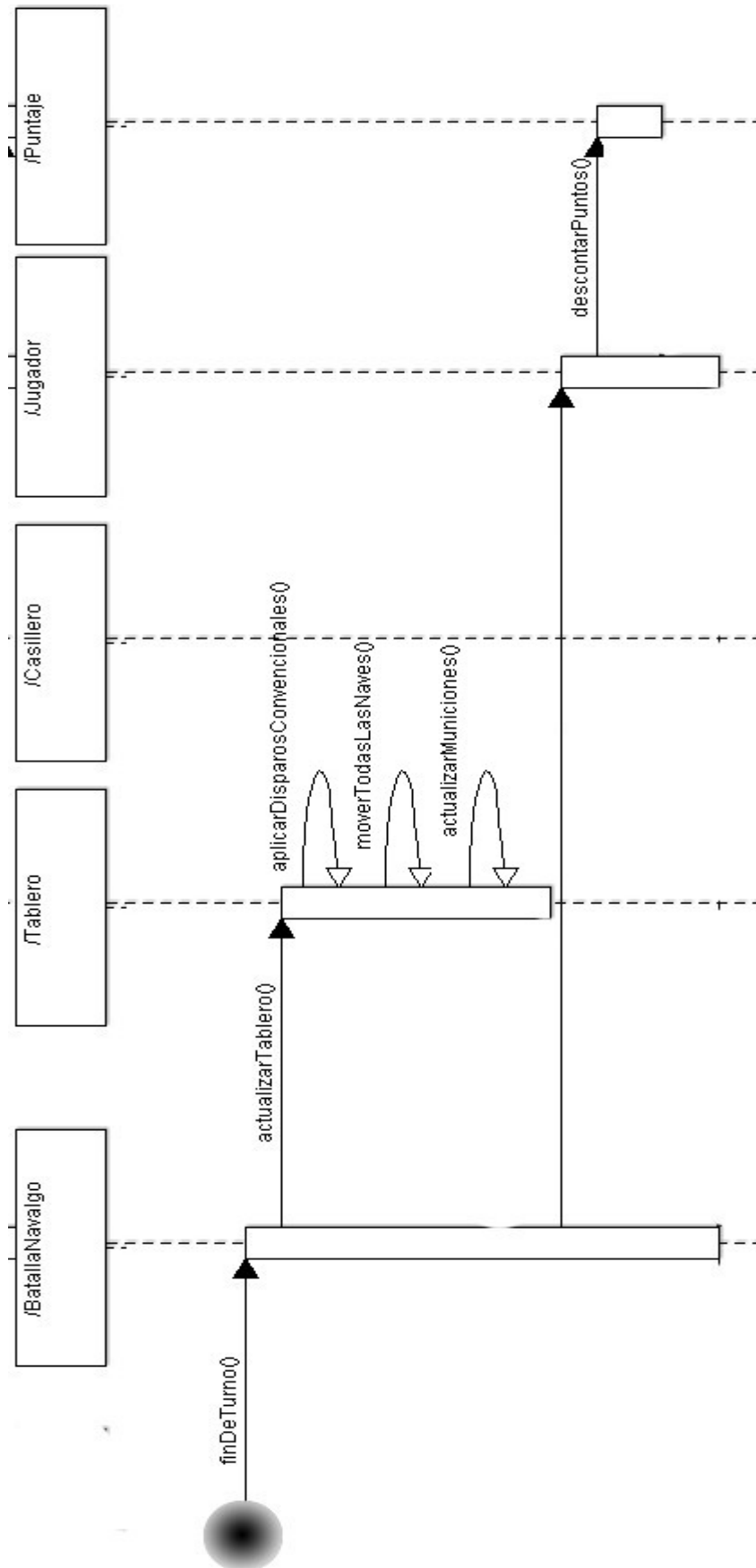




Diagrama de finDeTurno():



## **Código fuente**

(En documento aparte)

## **Checklist de corrección**

Esta sección es para uso exclusivo de la cátedra, por favor no modificar.

### **Carpeta**

#### **Generalidades**

- ¿Son correctos los supuestos y extensiones?
- ¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tipografía uniforme)

#### **Modelo**

- ¿Está completo? ¿Contempla la totalidad del problema?
- ¿Respeto encapsulamiento?
- ¿Hace un buen uso de excepciones?
- ¿Utiliza polimorfismo en las situaciones esperadas?

### **Diagramas**

#### **Diagrama de clases**

- ¿Está completo?
- ¿Está bien utilizada la notación?

#### **Diagramas de secuencia**

- ¿Está completo?
- ¿Es consistente con el diagrama de clases?
- ¿Está bien utilizada la notación?

#### **Diagrama de estados**

- ¿Está completo?
- ¿Está bien utilizada la notación?

### **Código**

#### **Generalidades**

- ¿Respeto estándares de codificación?
- ¿Está correctamente documentado?