



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
ESCUELA DE INGENIERÍA Y CIENCIAS  
CC3002 - 1 METODOLOGÍAS DE DISEÑO Y PROGRAMACIÓN

## TAREA 2

### Resumen

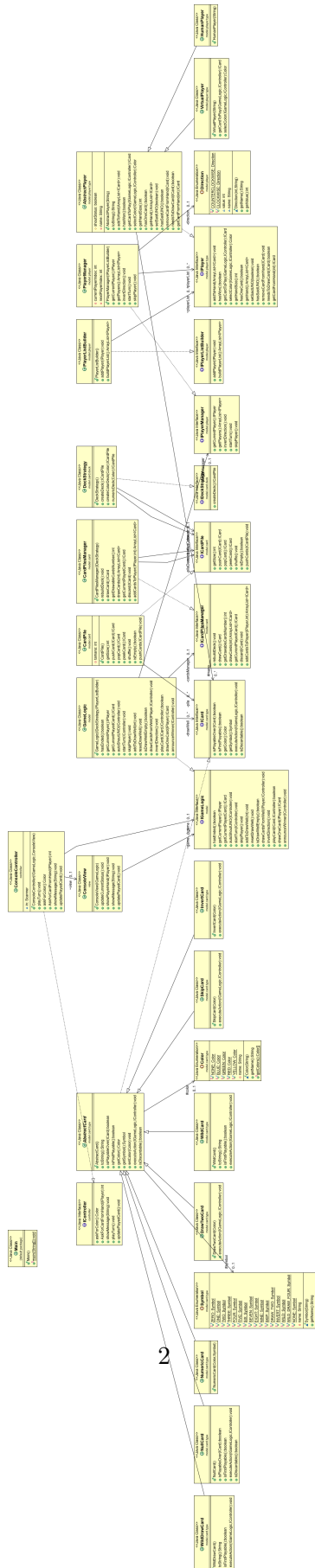
Daniela Campos  
danielacamposfischer@gmail.com  
<https://github.com/DaniCampos/cc3002-tarea2>

Profesor:  
Alexandre Bergel

Fecha:  
25 de Octubre del 2017



## 2



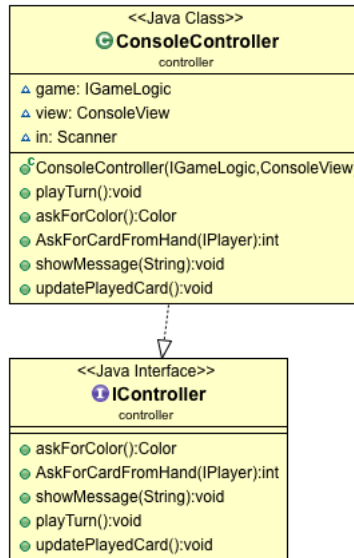


Figura 2: UML controller.package

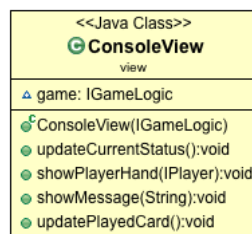


Figura 3: UML view.package



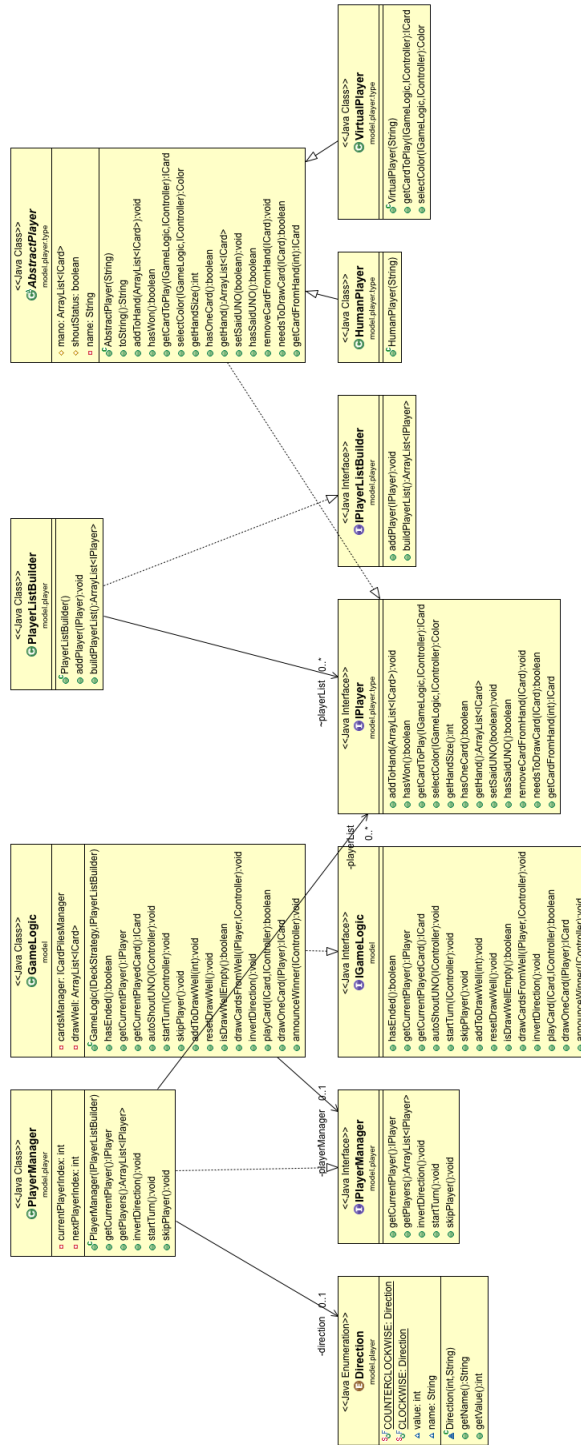


Figura 5: UML modelPlayer.package

## 2 Patrones de Diseño

Los patrones de diseño utilizados en la Tarea 2 fueron:

1. Adapter: Se adaptó la clase Stack de Java, utilizando la clase ICardPile.
2. Template: Se implementó la clase DeckStrategy, la cual podría ser utilizada para crear un mazo para cualquier juego de cartas, debido a que dentro de esta se implementaron los métodos createColorDeck y numericDeck, los cuales permiten crear un mazo de cartas de un color y uno con todas las cartas numéricas respectivamente. Se quería implementar una clase TestDeckStrategy, para poder testear la correcta creación de un mazo de cartas, pero por motivos de tiempo no fue posible.
3. NullObject: Se implementó la clase NullCard, con el objetivo de utilizarlo cuando fuera necesario sacar una carta, para así no tener que verificar que el arreglo que contuviera las cartas apropiadas para ser jugadas fuera nulo o estuviera vacío. A la hora de tratar de incorporar este comportamiento al juego, este fallaba, por lo que se priorizó la funcionalidad del juego por sobre la utilización de este patrón.
4. Factory: Se utilizó este patrón a la hora de crear la lista de jugadores y el mazo. Para el caso del mazo, se utilizó la interfaz IDeckStrategy para crear un mazo(Product) utilizando DeckStrategy(Factory), el cliente correspondía a CardPilesManager. Para crear la lista de jugadores(Producto), se utilizó la interfaz PlayerListBuilder(Factory) utilizando PlayerListBuilder, la cual era instanciada por PlayerManager(Cliente).
5. State: A la hora de realizar este resumen, se pensó en implementar un state patter para distinguir si es que el jugador había dicho uno o no.