

# Práctica Introducción a listas

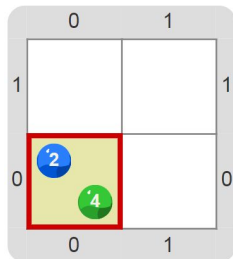
Introducción a la Programación 2do Semestre de 2018

## Manejo básico de listas

### Los siguientes ejercicios tienen interacción con el tablero de Gobstones

#### Ejercicio 1

Escribir la función `coloresCelda()` que describa una Lista de números de 4 elementos, donde el primer elemento representa la cantidad de bolitas azules en la celda actual, el segundo la cantidad de bolitas negras, el tercero la de bolitas rojas y el cuarto la de bolitas verdes.



Por ejemplo: si ejecutamos `coloresCelda()` en este tablero, la función describe la lista `[2,0,0,4]`

#### Ejercicio 2

Escribir la función `coloresVariasCeldas()` que describe una Lista de números de 4 elementos, donde el primer elemento representa la cantidad de bolitas azules en la celda actual, el segundo la cantidad de bolitas negras el celda un lugar al Este, el tercero la de bolitas rojas dos lugares al Este y el cuarto la de bolitas verdes 3 lugares al Este.

#### Ejercicio 3

Escribir la funcione `cantidadDeAzulesEn_(listaDeNúmeros)` que dada una lista de números de 4 elementos que representan la cantidad de bolitas de colores de la celda actual denote la cantidad de bolitas azules representadas por la lista. Luego realice `cantidadDeNegrasEn_`, `cantidadDeRojasEn_` y `cantidadDeVerdesEn_`.

#### Ejercicio 4

Escribir la función `hayMismaCantidad(listaDeNúmeros)` que dada una lista de números de 4 elementos que representan la cantidad de bolitas de colores de la celda actual, determine si hay la misma cantidad de bolitas de cada color.

### Ejercicio 5

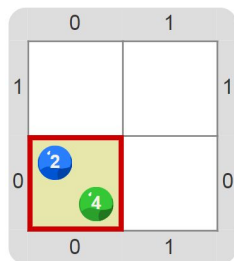
Escribir la función `hayDistintaCantidad(listaDeNúmeros)` que dada una lista de números de 4 elementos que representan la cantidad de bolitas de colores de la celda actual, determine si en ningún caso hay la misma cantidad de bolitas de cada color (es decir, son todas distintas).

### Ejercicio 6

Escribir la función `hayAlgunaCantidadIgual(listaDeNúmeros)` que dada una lista de números de 4 elementos que representan la cantidad de bolitas de colores de la celda actual, determine si en algún caso hay la misma cantidad de bolitas de dos colores (por ejemplo, hay la misma cantidad de rojas que de azules, o de azules que de verdes, etc.).

### Ejercicio 7

Escribir la función `direccionesAlBorde()` que retorne la lista de direcciones en las que el cabezal no se puede mover. En el siguiente tablero `direccionesAlBorde()` describe la lista [Sur, Oeste].



El cabezal en este caso no puede moverse ni al Sur, ni al Oeste.

### Ejercicio 8

¿Cómo puedo utilizar `direccionesAlBorde` para determinar si mi tablero tiene una única celda?

## Los siguientes ejercicios utilizarán diferentes tipos de Listas

### Ejercicio 9

Escribir la función `esSingular()` que, dada una Lista, describa verdadero si la lista tiene un único elemento.

### Ejercicio 10

Escribir la función `haySiguienteTramo(ruta)` que, dada una lista de Direcciones, y teniendo en cuenta que el siguiente tramo solo se contempla mirando la primer y segunda dirección de la ruta, describir verdadero si existe al menos un tramo completo.

### Ejercicio 11

Escribir la función `sigueUnaCurva(ruta)` que, dada una lista de Direcciones, y teniendo en cuenta que el siguiente tramo solo se contempla mirando la primer y segunda dirección de la ruta, describir verdadero si el tramo que sigue es una curva. La función debe ser total.

#### **Ejercicio 12**

Escribir la función `sigueCurvaADerecha(ruta)` que, dada una lista de Direcciones, y teniendo en cuenta que viene una curva, describir verdadero si el tramo que sigue es una curva hacia la derecha.

#### **Ejercicio 13**

Escribir la función `sigueCurvaAlzquierda(ruta)` que, dada una lista de Direcciones, y teniendo en cuenta que viene una curva, describir verdadero si el tramo que sigue es una curva hacia la izquierda.

#### **Ejercicio 14**

Escribir la función `sigueRecta(ruta)` que, dada una lista de Direcciones, describir verdadero si el tramo que sigue es una recta.

**Los siguientes ejercicios utilizarán el tipo Carta modelado en la clase teórica.**

#### **Ejercicio 15**

Escribir la función `primerCarta(cartasEnMano)` que, dada una lista de Cartas que representan las cartas que un jugador tiene en la mano dispuestas en el orden que las quiere jugar, describa la primer carta a jugar.

#### **Ejercicio 16**

Escribir la función `segundaCarta(cartasEnMano)` que, dada una lista de Cartas que representan las cartas que un jugador tiene en la mano dispuestas en el orden que las quiere jugar, describa la segunda carta a jugar.

#### **Ejercicio 17**

Escribir la función `tercerCarta(cartasEnMano)` que, dada una lista de Cartas que representan las cartas que un jugador tiene en la mano dispuestas en el orden que las quiere jugar, describa la tercer carta a jugar.

#### **Ejercicio 18**

Escribir la función `esFigura(carta)` que, dada una Carta, describa verdadero solo si es una figura. Se dice que una carta española es una figura, porque sus dibujos representan diferentes personajes. La carta con número 10 representa un paje medieval, número 11 un caballero, número 12 el rey.

#### **Ejercicio 19**

Escribir la función `luegoDeRobarCarta(cartasEnMano, mazo)` que, dada una lista de Cartas que representan las cartas que un jugador tiene en la mano y un mazo de Cartas del cual puede robar, describir sus cartas luego de robar.

#### **Ejercicio 20**

Escribir la función `luegoDeJugarCarta(cartasEnMano)` que, dada una lista de Cartas que representan las cartas que un jugador tiene en la mano y que solo puede jugar la primera, describir sus cartas luego de jugar su turno.

#### **Ejercicio 21**

Escribir la función `luegoDeJugarSegundaCarta(cartasEnMano)` que, dada una lista de Cartas que representan las cartas que un jugador tiene en la mano, describir sus cartas luego de jugar en su turno la segunda de todas sus cartas.

#### **Ejercicio 22**

Escribir la función `primerasTresCartas(mazo)` que, dado un mazo, describa una lista con sus primeras 3 cartas. "mazo" es una Lista de Cartas, con al menos 3.

#### **Ejercicio 23**

Escribir la función `tengoJugadaEnTira(tira, misCartas)` que, dada una tira de Cartas que se asume en orden creciente y `misCartas`, de las cuales puedo jugar solo la primera, describa si puede ser jugada en la tira o no. La condición que debe cumplir para ser jugada es que mi carta a jugar sea de diferente palo y menor a la primer carta de la tira.

#### **Ejercicio 24**

Escribir la función `puedeJugarCartaEnTira(tira, carta)` que, dada una tira de Cartas que se asume en orden creciente y una carta, describa la tira con la carta jugada en esa tira. Se asume que la carta cumple con las condiciones para ser jugada.

#### **Ejercicio 25**

Escribir la función `descubrirLaCarta(mazo)` que, dado un mazo, describa la siguiente carta a descubrir tomadas de a 3. Es decir la 3ra del mazo.

#### **Ejercicio 26**

Escribir la función `tieneEnvido(cartasEnMano)` que, dada una lista de 3 Cartas que representan las cartas que un jugador tiene en la mano, describa verdadero solo si tiene al menos 2 cartas del mismo Palo.

#### **Ejercicio 27**

Escribir la función `tieneFlor(cartasEnMano)` que, dada una lista de 3 Cartas que representan las cartas que un jugador tiene en la mano, describa verdadero solo si tiene las 3 cartas del mismo Palo.

### Ejercicio 28

Escribir la función `cuantoDeFlor(cartasEnMano)` que, dada una lista de 3 Cartas que representan las cartas que un jugador tiene en la mano, describa la suma de los números de las cartas que no son figura (es decir que la figura suma cero) en caso que tenga flor o cero en caso contrario.

### Ejercicio 29

Escribir la función `cuantoDeEnvio(cartasEnMano)` que, dada una lista de 3 Cartas que representan las cartas que un jugador tiene en la mano, describa cuánto se suma para el envío. Esto se calcula sumando las cartas del mismo palo (la figura cuenta como cero)

### Nuevo tipo: Boletos de Colectivo



Queremos modelar boletos de colectivo. Todo boleto tiene que contener ciertos datos que son obligatorios por las regulaciones vigentes. En particular, debe contener el número de línea al cual pertenece el boleto (Ej. 324, 159, 257, 85, etc.), la cantidad de tramos para la cual se sacó dicho boleto (siendo 1 el mínimo y 4 el máximo), la serie, que puede estar compuesta de números y letras, y el número del boleto, que corresponde a un número siempre de cinco dígitos.

### Ejercicio 30

Se pide que modele el tipo `BoletoDeColectivo`, con los datos necesarios. Escriba los tipos de los campos, y la invariante de representación.

### Ejercicio 31

Piense una estrategia para realizar la función `esCapicúa(boleto)` que, dado un boleto, determine si el número del mismo es capicúa. Un boleto es capicúa cuando el primer dígito es igual al último, y el segundo al anteúltimo.

### Ejercicio 32

¿Fue fácil la pensar la estrategia? ¿Se le ocurre otra representación de BoletoDeColectivo que haga más fácil resolver el problema?

### **Ejercicio 33**

Se pide realice la función costoDeBoleto(boleto). El costo del boleto es de \$15, si se abona un solo tramo, y aumenta en \$1 por cada tramo adicional.

### **Ejercicio 34**

Se pide realice la función tienenMismoNúmero(boletoUno, boletoDos), que dados dos boletos, denote Verdadero si ambos boletos tienen el mismo número de boleto.

### **Ejercicio 35**

Se pide realice la función sonParejaCapicúa(boletoUno, boletoDos), que dados dos boletos, denote Verdadero si el número del primer boleto es igual al número del segundo boleto dado vuelta. (Ej. los boletos con números 12345 y 54321 serían pareja)

### **Ejercicio 36**

Realice la función esCapicúaDePrimeraSerie(boleto) que dado un boleto indique si el mismo es capicúa de una primera serie. La primera serie debe ser la serie "1" o serie "A".