|  |  |
| --- | --- |
| **UNIVERSIDAD DE SANTIAGO DE CHILE**  **FACULTAD DE INGENIERÍA**  **DEPARTAMENTO DE INGENIERÍA EN INFORMÁTICA** | http://guiaweb.usach.cl/sites/default/files/inegrotrz_0.png |

**PARADIGMAS DE PROGRAMACIÓN**

**LAB2 PROLOG**

**SEBASTIAN IGNACIO VILLALOBOS PEREZ**

Profesor: Roberto González

Santiago de Chile

1 – 2022

Table of Contents

[1. Introducción 2](#_Toc103437275)

[2. Marco teórico 3](#_Toc103437276)

[2.1 Paradigma logico 3](#_Toc103437277)

[3. Descripción de la solución 4](#_Toc103437278)

[3.1 Caracteristicas de la solucion 4](#_Toc103437279)

[3.2 Alcances y limitaciones 4](#_Toc103437280)

[4. Herramientas y ambiente de trabajo 5](#_Toc103437281)

[5. Desarrollo 5](#_Toc103437282)

[5.1 TDAs 5](#_Toc103437283)

[5.2 cardsSetIsDobble 5](#_Toc103437284)

[5.3 cardsSetNthCard 5](#_Toc103437285)

[5.4 cardsSetFindTotalCards 5](#_Toc103437286)

[5.5 cardsSetMissingCards 5](#_Toc103437287)

[5.6 cardsSetToString 5](#_Toc103437288)

[5.7 dobbleGame 5](#_Toc103437289)

[5.8 dobbleGameRegister 5](#_Toc103437290)

[5.9 dobbleGameWhoseTurnIsIt 5](#_Toc103437291)

[5.10 dobbleGamePlay 5](#_Toc103437292)

[5.11 dobbleGameStatus 5](#_Toc103437293)

[5.12 dobbleGameScore 5](#_Toc103437294)

[5.13 dobbleGameToString 5](#_Toc103437295)

[6. Conclusiones 5](#_Toc103437296)

[7. Referencias 6](#_Toc103437297)

# Introducción

Dentro del ramo de paradigmas de la programación impartido en la Universidad de Santiago de Chile se ha propuesto representar el juegos de cartas **Dobble** desde distintos paradigmas. Para este primer laboratorio el desarrollo fue realizado con el paradigma logico.

Utilizando Prolog que fue uno de los primeros lenguajes de programación basados ​​en los conceptos de programación lógica.

A continuación se explicará cómo se han abordado los distintos requerimientos propuestos por el profesor.

# Marco teórico

El término paradigma de programación se refiere a un estilo de programación. No se refiere a un lenguaje específico, sino que se refiere a la forma en que programa.

Hay muchos lenguajes de programación que son bien conocidos, pero todos necesitan seguir alguna estrategia cuando se implementan. Y esa estrategia es un paradigma.

## Paradigma logico

El paradigma de la programación funcional ha estado en el centro de atención desde hace un tiempo debido a PROLOG, un lenguaje de programación funcional.

El paradigma de la programación funcional tiene sus raíces en las matemáticas y es independiente del lenguaje. El principio clave de este paradigma es la ejecución de una serie de funciones matemáticas. Todo el código está dentro de una función. Todas las variables están en el ámbito de la función.

En el paradigma de programación funcional, las funciones no modifican ningún valor fuera del alcance de esa función y las funciones mismas no se ven afectadas por ningún valor fuera de su alcance.

# Descripción de la solución

Para este laboratorio se implementará utilizando el paradigma logico el juegos **Dobble**, que cuenta con un conjunto finito de cartas (piezas o tarjetas). En cada tarjeta se listan n elementos (figuras, números, letras, etc.) diferentes. Además, para cualquier par de cartas, existe un y sólo un elemento en común. A partir del conjunto de cartas, existen diversas modalidades de juego en las que pueden participar 2 o más jugadores. ¿Será posible aplicar el lenguaje funcioanl utilizando PROLOG para representar el juego **Dobble**?

## Caracteristicas de la solucion

La solucion es implementada en el lenguaje de programacion PROLOG, utilizando el paradigma funcional, generando un set de cartas del juego Dobble valido y la modalidad de juego.

La solución debe poder:

1. Generar un mazo de cartas valido para Dobble
2. Consultar la cantidad de cartas y los símbolos.
3. Validar los mazos de cartas.
4. Validar los elementos que componen las cartas
5. Generar un juego validoo.
6. Registrar jugadores.
7. Consultar estado del juego.
8. Consultar el turno en el que se encuentra la partida.
9. Consultar los jugadores registrados y sus puntajes.
10. Finalizar el juego.

## Alcances y limitaciones

A continuacion se listan los alcances y limitaciones del proyecto:

* Respetar la definición de función en términos de conjunto de salida (dominio) y llegada (recorrido) sin efectos colaterales, además del nombre de las mismas (respetar mayúsculas y minúsculas).

# Herramientas y ambiente de trabajo

Para el desarrollo de este laboratorio se utilizan:

* Repositorio en github para lelvar un control de versiones. <https://github.com/sebastiavillalobos/lab2_2022>
* Se utiliza un notebook i3 con 8gb de ram con ubuntu 20.04.
* VisualStudioCode para editar archivos.

# Desarrollo

### 5.1 TDAs

### 5.2 cardsSetIsDobble

### 5.3 cardsSetNthCard

### 5.4 cardsSetFindTotalCards

### 5.5 cardsSetMissingCards

### 5.6 cardsSetToString

### 5.7 dobbleGame

### 5.8 dobbleGameRegister

### 5.9 dobbleGameWhoseTurnIsIt

### 5.10 dobbleGamePlay

### 5.11 dobbleGameStatus

### 5.12 dobbleGameScore

### 5.13 dobbleGameToString

# Conclusiones

Poder definir un buen TDA ayuda a tener un manejo completo de las funcionalidades del dato creado, y en este laboratorio no fue la excepción.

El uso de funciones de orden superior y distintos tipos de recursión fueron las principales herramientas utilizadas.

Como conclusión se logra dar respuesta a la pregunta antes planteada, es posible realizar la representación del juego **Dobble** desde el paradigma funcional y utilizando el lenguaje de programación PROLOG.

# Referencias