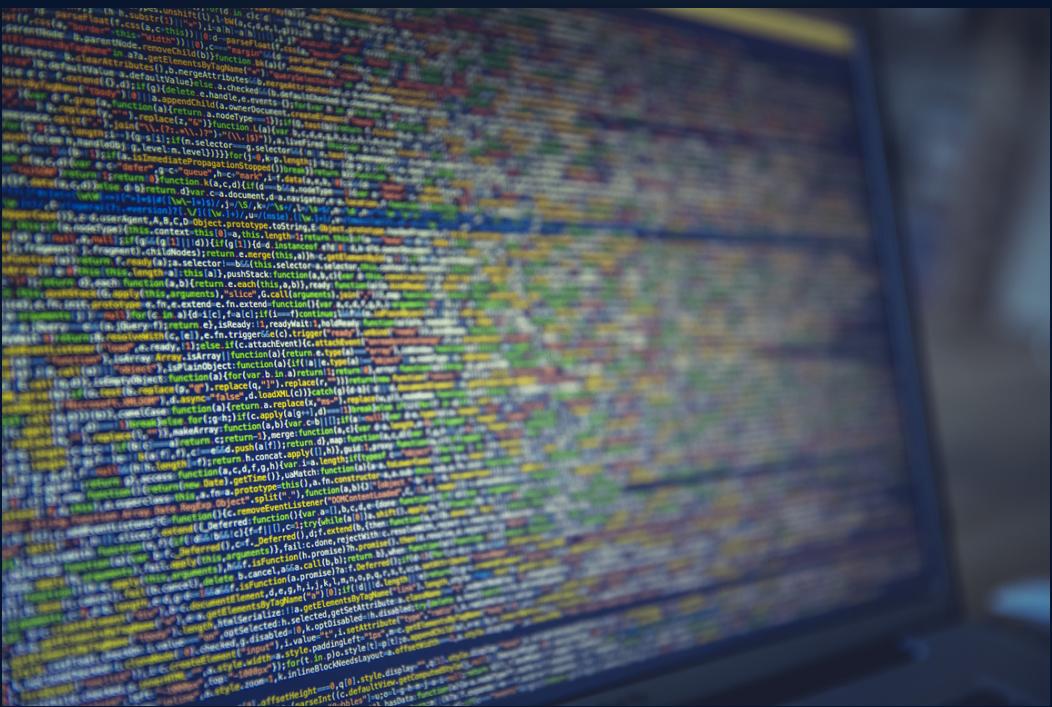


# TECHNICAL MANUAL



Kevin Estuardo Palacios  
Quiñonez

IPC2 SECCIÓN N

# INDEX

I INTRODUCTION

II OBJECTIVES

III ADDRESSED

IV TECHNICAL SPECIFICATIONS

    1. HARDWARE REQUIREMENTS

    2. SOFTWARE REQUIREMENTS

V PROGRAM LOGIC

VI RECOMMENDATIONS

VI CREDITOS

# I INTRODUCTION

This program consists of a simulation where two XML are introduced, one for players and one for gifts, both configure the program through their input and are responsible for plotting the queue of players, pile of prizes and plot the solution of the puzzle.

# II OBJECTIVES

The main objective of this program is to help the players that are mentioned in the XML to calculate their scores, and to show how the players' queue looks like graphically, to see their prize stacks, and to show them graphically through the Graphiz tool.

## III ADDRESSED

This manual is oriented to all the different programmers interested in the development of this application and to the assistant professor of Introduction to Computing and programming 2 of those who have the knowledge about these topics, showing a different way to solve the problem.

## IV TECHNICAL SPECIFICATIONS

### 1. HARDWARE REQUIREMENTS

- Desktop or Laptop Computer.
- Minimum 8GB of RAM.
- 20 GB of hard disk space available.
- Core i3 or higher processor.
- 64-bit processor.
- Screen with 1024\*768 pixels graphic resolution.

## 2. SOFTWARE REQUIREMENTS

- Windows 7 or higher installed
- Python 3.9.2
- Have Graphiz installed

## V PROGRAM LOGIC

### XML Structure Players

Ejemplo XML de entrada de configuracion

```
<?xml version="1.0" encoding="UTF-8"?>
<jugadores>
  <jugador>
    <datospersonales>
      <nombre>$nombre</nombre>
      <edad>$edad</edad>
    </datospersonales>
    <movimientos>36</movimientos>
    <tamaño>20</tamaño>
    <figura>arbol</figura>
    <puzzle>
      <celda f="1" c="0" />
      <celda f="1" c="4" />
      <celda f="1" c="3" />
      <celda f="2" c="3" />
      <celda f="3" c="4" />
```

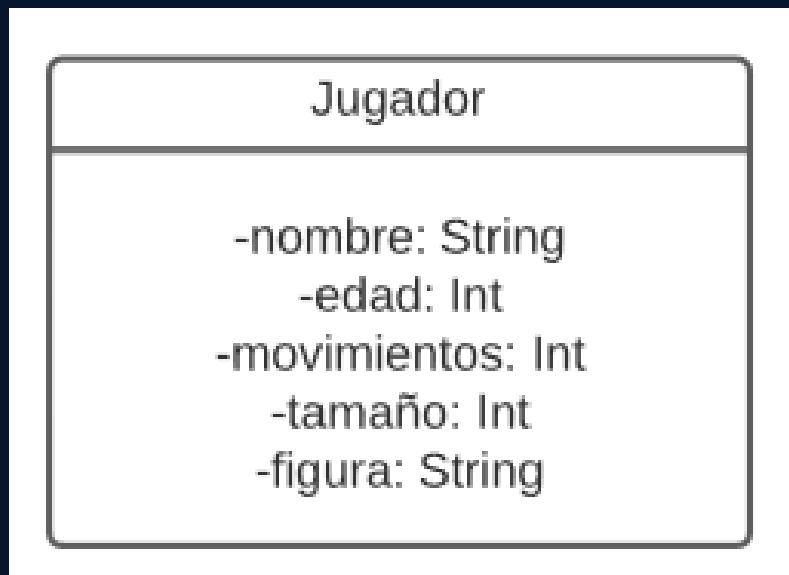
```
<celda f="4" c="1" />
<celda f="4" c="2" />
<celda f="4" c="3" />
<celda f="4" c="4" />
<celda f="0" c="2" />
</puzzle>
<solucion>
  <celda f="1" c="2" />
  <celda f="2" c="1" />
  <celda f="2" c="2" />
  <celda f="2" c="3" />
  <celda f="3" c="0" />
  <celda f="3" c="1" />
  <celda f="3" c="2" />
  <celda f="3" c="3" />
  <celda f="3" c="4" />
  <celda f="4" c="2" />
</solucion>
</jugador>
<jugador>....</jugador>
<jugadores>
```

# XML Structure Awards

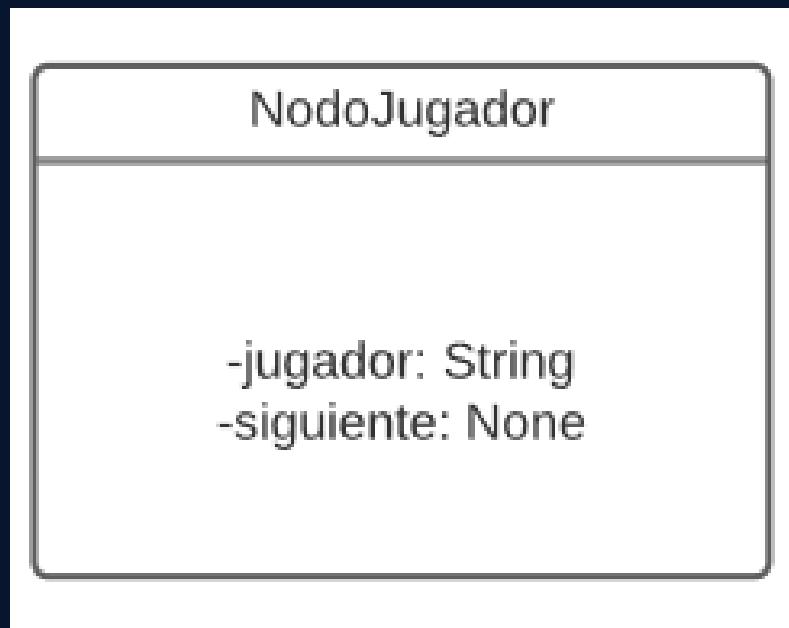
```
Ejemplo XML de entrada de premios
<?xml version="1.0" encoding="UTF-8"?>
<premios>
    <premio>
        <lugar>10</lugar>
        <regalo>Cupones de Descuento</regalo>
    </premio>
    <premio>
        <lugar>9</lugar>
        <regalo>Oso de Peluche</regalo>
    </premio>
    <premio>
        <lugar>8</lugar>
        <regalo>Pachon hermetico</regalo>
    </premio>
    <premio></premio>
</premios>
```

## Class Diagrams

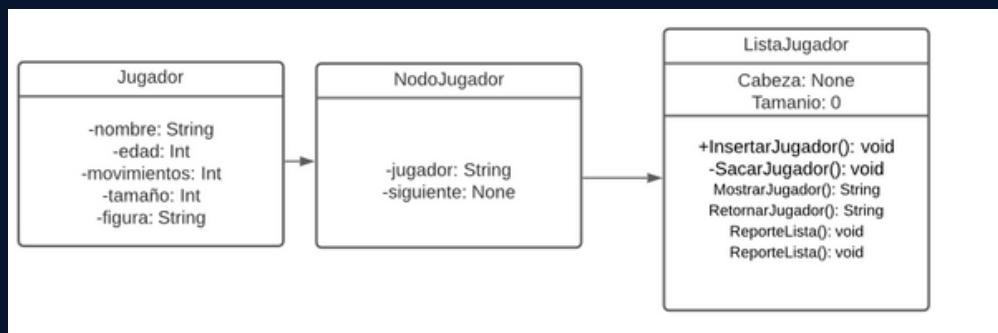
### Player Class



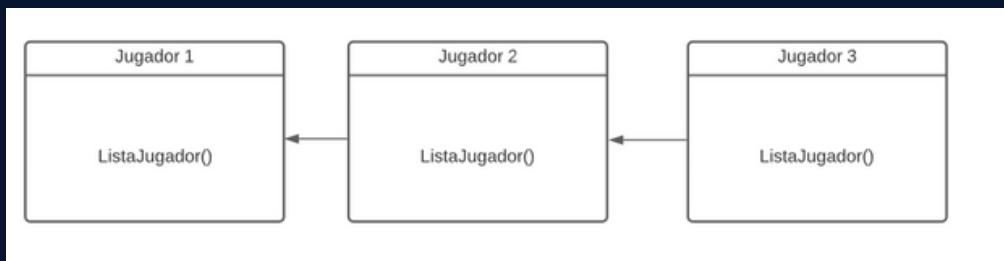
# NodePlayer



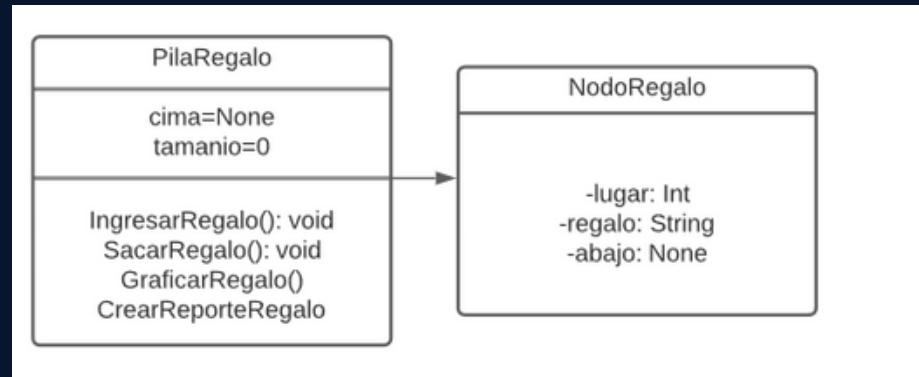
Connection between Player, PlayerNode and PlayerList



Graphical representation of a queue



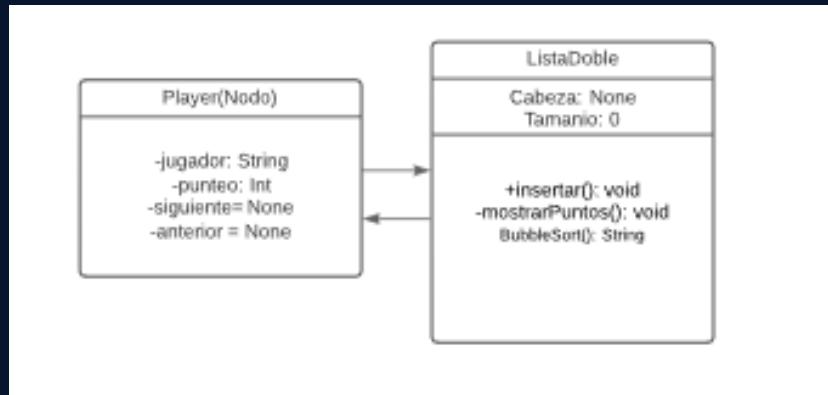
# Data Structure Gift



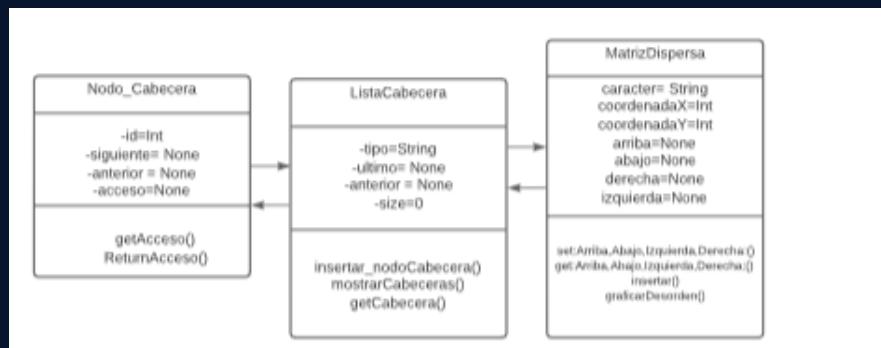
Graphical representation of a stack



## Graphical representation of ListaDouble



## Graphical Representation of Disperse Matrix



With the structures already proposed, the following algorithms were performed:

- Menu: Allows the user to choose one of the multiple options available in the program, the order of the functions must be respected for optimal operation.

b. Path: Allows the user to access an Askopenfiledialog to be able to load the file.

c. ProcessXml: Allows the user to process the xml file and verify the corresponding validations that will be carried out on the top 10 players.

d. ShowPlayer: by using a SimpleList a specific player that was loaded in the xml is searched for and then deleted.

e. RemovePlayer: by using a SimpleList, the player in the queue is removed.

f. CreateReportList: it is in charge of graphing the queue using the graphiz tool.

g. InsertGift: it is in charge of inserting the xml gifts by means of a node and the structure of a stack.

h. RemoveGift: as the function indicates this one is in charge of being able to remove a gift of the pile, if they are finished it will show a message.

i. plot gift: it is in charge of plotting the stack of gifts with the help of the graphiz tool.

### Honorable Mention

By using a double list, the score data and the player's name were taken in order to generate the top 10 players.

Bubblesort sorting: it was used to provide a sorting solution.

# VI RECOMMENDATIONS

- Please use the menu options in the order they are displayed.
- Have Graphiz installed.

# VII CREDITS

Prepared by the student Kevin Estuardo Palacios Quiñonez with ID number 201902278 for the course Introduction to Computing and Programming 2, in the country of Guatemala, delivered on December 20.