Algorithms and Data Structures

Unit Laboratory 2 - 2020-1

Mayor Andrés - A00359333, Jhonatan Arboleda A00358993

The Indomitable Spirit

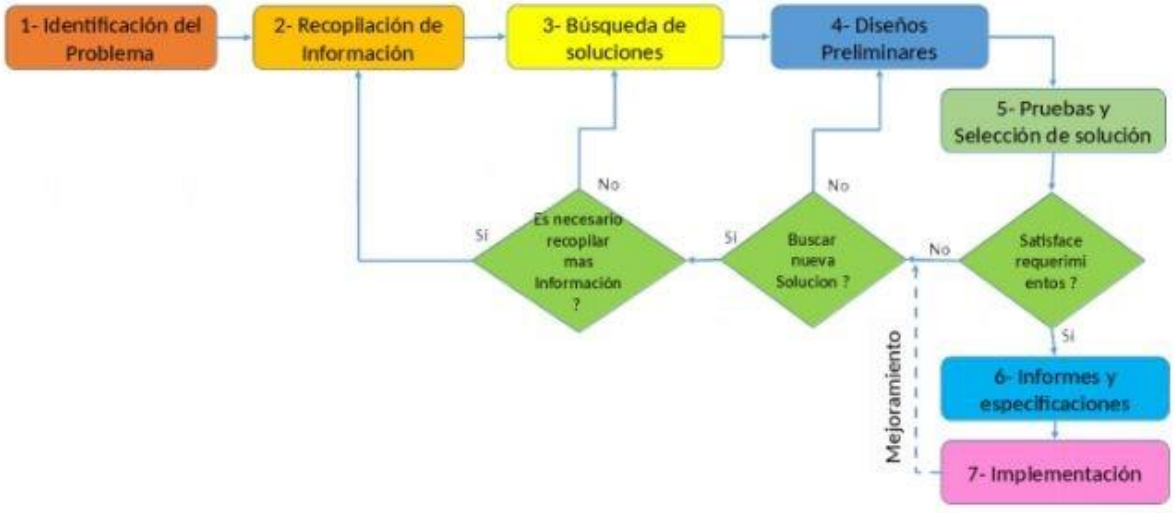
Context of the Problem.

The Indomitable Spirit is powered by a company in the city of Cali that handles a large casino game, and decided to enter the world of the houses of bets a hippodrome novel where horse racing will be played and also your customers can place bets , for this a optimal solution is sought for good deasarrollo of this game.

Solution development.

To solve the previous situation, the Engineering Method was Chosen to Develop the solution, following a systematic approach in Accordance With the problematic situation.

Based on the description of the Engineering Method of the book "Introduction to Engineering" by Paul Wright, it presents the following flow chart, Whose steps we will follow in the development of the solution.



Step 1. Problem Identification

Definition of the problem

Need to create a game where almanecemos entire flow of imformation entered by customers such as betting, riders compete, etc. In addition also it wants to implement efficient algorithms for proper operation of the game and thus allow the required options as revenge and consulting bet.

Specification of Functional Requirements. (In terms of input / output)

Name	**R. # 1. Horserider registration**
Summary	Registration Between 7 and 10 horseriders is required.
inputs	
• Name of the rider. • Name of the horse.	
Outputs	
Prime Numbers Have Been generated.	

Name	**R. # 2. Allow logging betting users.**
Summary	Allow registration betting users for 3 minutes.
inputs	
• Identification card. • Name. • Number of the Horse by which bet. • Amount wagered.	
Outputs	
The bet is registered.	

Name	**R. # 3.** Display the podium of the winners of the race.
Summary	You must display the podium of the winners of the race.
inputs	
• None.	
Outputs	
It has despliegado the podium of the winners of the race.	

Name	**R. # 4.** Consult bet.
Summary	Users can quickly check your bet, showing if your horse won or lost the race.
inputs	
• Identification card. • Registration bet	
Outputs	
information of the bet unfolds.	

Name	R. # 5. Allow rematch option.
Summary	a career can be repeated by the same riders, but in this case the first to reach the finish in the previous race will be the rider of the last track and the first track will occupy the rider who came last.
inputs	
• none	
Outputs	
The race has started.	

Name	R. # 6.Crear new career.
Summary	Users can create a new race where the entire flow is repeated.
inputs	
• none	
Outputs	
It has created a new career.	

Step 2. Information Collection

To have clarity on the concepts involved and how to play The Indomitable Spirit, it becomes a search for data structures that can be implemented for the efficient operation of the game, and to solve the problem effectively reach planted. For it is important perform this search solvents and reliable sources to find out which elements are part of the problem and those that are not.

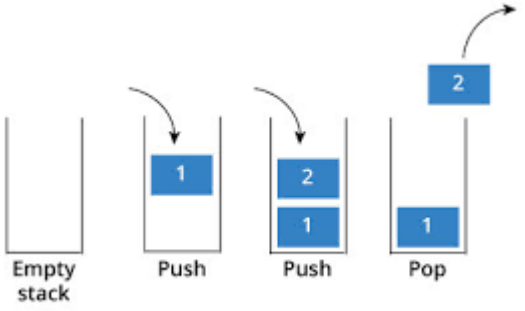
Definitions

Racing tracks occur in closed loop, also known racetracks, usually ovals or, more rarely, triangles. There are also open tracks.The floor is sand or grass. In addition to the track's racetracks have stands with a panoramic view of racing for the fans and betting windows for sale.

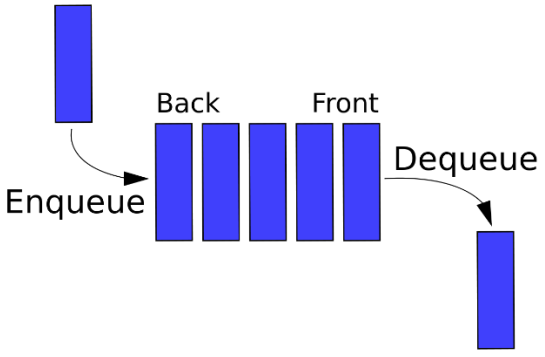
Source: https://es.wikipedia.org/wiki/Turf_(h%C3%ADpica)

Data Structures to be used in the project:

- **Stack**: It's a collection of elements, the LIFO That Follows order. LIFO stands for Last In First Out, Which Means Which is inserted element will be removed Most Recently first.

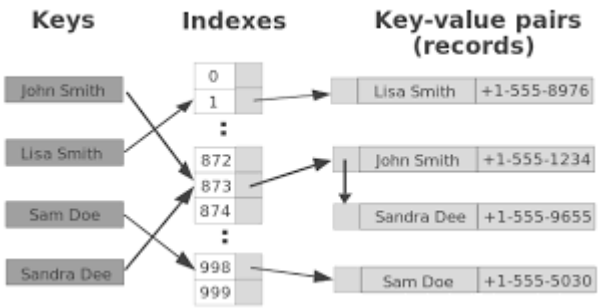


- **Queue**: It's a data structure That Follows the FIFO principle. Means FIFO First In First Out ie the element added first in the queue will be the first one to be removed. Elements are always added to the back and removed from the front.



Source: https://www.hackerearth.com/practice/notes/stacks-and-queues/

- **Hashtable**: It's a data structure implements an associative array That abstract data type, a structure That can map keys to values. A hash table hash function to use to compute an index into an array of buckets or slots, from Which the Desired value can be found.



Source: https://en.wikipedia.org/wiki/Hash_table

Step 3. Search for solutions.

In this situation, the key problem is the ordering of data to achieve positioning and efficient search. Here are some alternatives are presented about the data structures that can be taken.

Alternatives:

- Using basic data structures as are the ArrayList or arrange for ordering data.
- Use more complex structures such as queues, stacks and hash tables for ordering data.

Step 4. Preliminary designs.

Then the options will be analyzed to see which the best is to take.

- Option 1: These data structures manage to sort the data in an effective but not efficient enough in terms of time and convenience.
- Option 2: These structures a little more complex if they can sort data efficiently enough and with great comfort for the user. In addition, part of Generics.

Step 5. Evaluation and Selection.

In this step we analyze the options on some criteria.

Criteria: Complies with Generics solution.

0: It should be different code each time, 1: Methods can be reused.

	Criterion
Option 1	0
Option 2	one

In the use structures like hash tables we can access in a more efficient information of the horses and the positions they occupied. It is also easier and more efficient to use the batteries in the case of reverse queues methods are with them as they are the push and pop. As for the queues, these fulfill exactly the function you need to run a horse race.

We use these structures to achieve overall efficiency in the aggregation, sorting and searching data. This will achieve run the horse race, revenge and betting, meeting all requirements posed by the problem.

Step 6. Preparation of reports and specifications.

The reports are not in this document because of the size of the class diagram does not allow to show properly. In the folder “docs” there are the reports of class diagram, the functional requirements, the TAD and the testcase design.

Step 7. Implementation of design.

In the following repository is the implementation of the solution proposed above: https://github.com/AndresMayor/TheIndomitableSpirit