Engineering method

1. **Identification of the problem**

Identification of needs and symptoms:

- The 4 sections of the videogame store must be identified.
- Section 1 corresponds to the entrance of the store where access is allowed.
- Section 2 corresponds to the search section where the customer makes quick inquiries about the videogames and the list of games chosen is ordered according to the shelf of the store where they are.
- Section 3 corresponds to the section of physical copies of videogames, where the client collects the games present in his/her list.
- Section 4 corresponds to the payment section, where after queuing, the corresponding amount is paid to one of the cashiers.
- The queries made by the client must be very fast searches
- The two required sorting algorithms must have a time complexity less than or equal to O (n^2).

Definition of the problem:

The millionaire wants to have a representation of how his videogame store would work.

2. **Compilation of information**

In another document the functional requirements and non-functional requirements will be made.

Some terms that should be clear are:
Time complexity: The amount of time taken by an algorithm to run, as a function of the length of the input.
Sorting algorithms: An algorithm that puts elements of a list into an order.

Another aspect to consider is the context of the problems we may face:
Sorting problems: Sorting a large number of items can take a substantial amount of computing resources. Like searching, the efficiency of a sorting algorithm is related to the number of items being processed. For small collections, a complex sorting method may be more trouble than it is worth. The overhead may be too high. On the other hand, for larger collections, we want to take advantage of as many improvements as possible. (Taken from 6.6. Sorting — Problem Solving with Algorithms and Data Structures (runestone.academy))
Searching problems: Searching is the algorithmic process of finding a particular item in a collection of items. A search typically answers either True or False as to whether the item is present. On occasion it may be modified to return where the item is found. (Taken from 6.2. Searching — Problem Solving with Algorithms and Data Structures (runestone.academy))

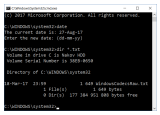
3. **Search of creative solutions**

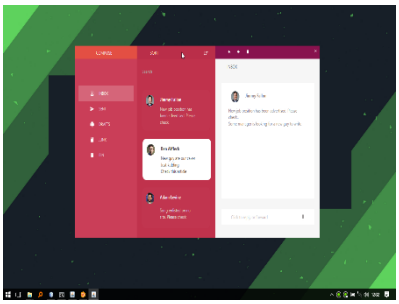
The methods found are as follows:

a. **Virtual reality simulation of the purchase process:** Through a simulation program of a physical space in 3D and the use of virtual

reality glasses, the millionaire can visualize very precisely how the video game store would work.

b. **System console program:** A java program that, given some inputs in plain text, returns some outputs also in plain text. It would work through the computer console





c. **Program with graphic interface and use of generic linear data structures:** A java program that through a simple but friendly interface with the user, allows to simulate the operation of the store given some inputs indicating the cases to be tested. This program would return outputs in a more orderly and clear way. It allows to have an order based on known data structures (queues, stacks,etc)

4. **Transition from formulation of ideas to preliminary designs**

The idea "Virtual reality simulation of the purchase process" is discarded. This is because the members of the group for this project do not yet have the knowledge to do something so complex. Besides that, it is quite expensive, and the representation does not have to be so exact to understand the operation of the store.

Careful review of the other proposals leads us to the following:

System console program with arrays:

- Uses plain text and console inputs
- Give outputs in plain text by console
- Does not have an optimal way of doing operations.

Program with graphic interface and use of generic linear data structures:

- Get the inputs through an interface
- Returns the outputs in an ordered and exemplified form
- Since it uses specific data structures for the problem, it allows to reduce the time complexity

5. **Evaluation and selection of the best solution**

Criteria:

Criterion A. Precision of the solution. The alternative gives a:

[2] Exact solution

[1] Approximate solution

Criterion B. Average efficiency. In general, the program will use operations with complexity:

[5] Constant

[4] Greater than constant

[3] Logarithmic

[2] Lineal

[1] Impossible to assume an average

Criterion C. Completeness. How many solutions does it deliver?

[3] All

[2] More than one if there are but not all

[1] Just one or none

Criterion D. Ease of algorithmic implementation

[2] Compatible with the basic arithmetic operations of modern computer equipment

[1] Not fully compatible with the basic arithmetic operations of modern computer equipment

Criterion E. User friendliness.

[4] Friendly with absolutely any user

[3] Friendly with users with at least some technological notion

[2] Difficult to understand if the user does not have some experience with applications

[1] Little or no friendliness

Evaluation

	Criterion A	Criterion B	Criterion C	Criterion D	Criterion E	Total
Alternative 1: System console program with arrays	2	1	3	2	2	10
Alternative 2: Program with graphic interface and use of generic linear data structures:	2	2	3	2	3	12

Selection

According to the previous evaluation, Alternative 2 should be selected since it obtained the highest score according to defined criteria.