**THE ENGINEERING METHOD**

**Context:** A foreigner decided to invest in Cali. He want to build a videogame store with an innovative way of service.

**Desarrollo de la solución**

In order to solve the situation, we were faced, we decided to use the engineering method to focus on a systematic view that match with the problem.

Following the steps of the engineering method from the book “introduction to engineering” from Paul Wright, we defined the next diagram which describe the steps we are going to use in the process.

Diagrama

Descripción generada automáticamente

1. **PROBLEM IDENTIFICATION.**

Usually, in a common game store, the customer tend to take a long period of time searching each game of interest, or even the new ones in order to try them. Also, some times by the unuseful interaction with the staff could bring more delays int the time people spend inside the store.

*Symptoms and necessities*

* Customers need, before entering the store, to know the catalogue that’s available
* The catalogue must have reviews of the games availables in the store when the customer need them.
* The customer must be able to add games in a shopping list.
* The shopping listo f each customer, after it’s finished, has to generate a id code that the customer Will use to enter the store.
* Each game has to have it’s own id code.
* The store Will have a group of stages that let the customers to make fast revisions.
* When the customer use the list code, the program Will load the game’s names to the first stage. Where Will be found the rack it is placed or where to search for it.
* The game list must be organized based in the location of each rack in oder to créate the best route.
* If the game it’s sold out, it’s code won’t appear in the games list.
* The customer must be able to choose between 2 different type of sorting algorithm to organize the route.
* The customer must have a automatized basket that Will follow the steps to track the games.
* Each client must have a waiting time depending on the time it walked through.
* The order by the customers get to the cashier it’s given by the time the customer walk through the store plus the time it took to get the games.
* All customers must do a unic queue even if there’s more than 1 cashier.
* If there are cashiers working along, the customers Will be checked out in the as much as the cashiers available.
* The exit order can variate depending on how long the cashier takes to check out the games which it’s equal to the number of games to buy.
* In the check out, the games Will be organized in a way that the last videogame put in the basket will be the first checked out.
* After completing the check out, the program must show the total costo f the games buyed.

*Problem definition*

The problem requires to develop a software that let the customers caleños know more about how the store Works. The possible solution should be able to simulate de process of buying videogames since the start of the application until the checkout wanting to solve the problem.

1. **Data recopilation**

In order to solve the problem we have to aknowledge the process of the store itself and show it to the customers. In the following part we explain the concepts we searched to be able to solve the problema using the simulated software.

**Generics:**

[[1]](#footnote-1)Generics mean parameterized types. The idea is to allow type to be a parameter to methods, classes, and interfaces. Using Generics, it is possible to create classes that work with different data types. An entity such as class, interface, or method that operates on a parameterized type is called a generic entity.

**Stacks:**

[[2]](#footnote-2)A stack is an array or list structure of function calls and parameters used in modern computer programming and CPU architecture. Similar to a stack of plates at a buffet restaurant or cafeteria, elements in a stack are added or removed from the top of the stack, in a “last in first, first out” or LIFO order.

**Queue:**

[[3]](#footnote-3)A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first. The difference between [stacks](https://www.geeksforgeeks.org/stack-data-structure/)and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

**TAD:**

[[4]](#footnote-4)An Abstract Data Type it’s a group of operations defined b yan specification that doesn’t depend on any representation.

**Tablas Hash:**

[[5]](#footnote-5)Hash Table is a data structure which stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value. Access of data becomes very fast if we know the index of the desired data.

**Github:**

[[6]](#footnote-6)GitHub is one of the world’s largest community of developers. It’s an intricate platform that fosters collaboration and communication between developers. GitHub has several useful features that enable development teams to work together on the same project and easily create new versions of software without disrupting the current versions

1. **Creative ideas research**

Para desarrollar las alternativas expuestas a continuación se realizó la técnica de lluvia de ideas: Generación espontánea de ideas diseñadas para resolver un problema. Sin embargo, cabe destacar que el dueño de la tienda solicito un software que simule el funcionamiento del establecimiento. Por ende, todas las alternativas esta inclinadas al desarrollo de una solución simulada, eficaz y concreta.

*Alternativa 1. Simulación en 3D*

Esta alternativa se trata de un programa hecho en Java que simula, con una visualización 3D (en realidad aumentada), la entrada y salida de los clientes a la tienda de videojuegos. Usando estructuras de datos tales como Colas (simulando la fila de clientes), Stacks (simulando la canasta y el carrito de los usuarios) y Hash Tables (simulando las estanterías y listas de los clientes) se puede llegar a una simulación aproximada del orden de salida de los clientes a la tienda basados en la cantidad de juegos, orden de llegada y tiempos estimados en la tienda de cada uno de los clientes. Con una interfaz que permita que el usuario pueda ver de manera aproximada y en primera persona, los movimientos que realizaría por cada sección gráficamente ejemplificando la experiencia real de ir a la tienda con el nuevo e innovador sistema de atención.

Interfaz de usuario gráfica

Descripción generada automáticamente*Alternativa 2. Simulación por GUI*

Se trata de un programa que implementaría un software de simulación en GUI (Graphical User Interface) es decir, una simulación con interfaz gráfica como la mayoría de las aplicaciones o software actuales. Software hecho en java, que simula la entrada y salida de los clientes a la tienda de videojuegos. Usando estructuras de datos como: Colas (simulando la fila de clientes a la hora de pagar en la caja), Stacks (simulando la canasta y el carrito de los juegos por usuario) y Hash Tables (simulando las estanterías y listas de los clientes)

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja*Alternativa 3. Simulación por CLI*

Opción la cual implementaría un software de simulación en CLI (command line interface) es decir, una simulación usando línea de comandos en la consola o terminal del dispositivo que se implemente. Software hecho en java, que simula la entrada y salida de los clientes a la tienda de videojuegos. Usando estructuras de datos como: Colas (simulando la fila de clientes a la hora de pagar en la caja), Stacks (simulando la canasta y el carrito de los juegos por usuario) y Hash Tables (simulando las estanterías y listas de los clientes)

1. **TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS PRELIMINARES (mockup)**

A continuación, se descartarán ideas con el fin de acotar las opciones y así elegir con mayor facilidad la solución más viable.

Alternativa 1. Programa de simulación con vista 3D de la tienda, ya que por la dificultad que supone la simulación en 3D, su implementación puede ser imprecisa y no proveer una óptima solución al problema.

La revisión de las otras alternativas nos lleva a lo siguiente:

(diseños)

1. **EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN**

**PESAR CRITERIOS PARA EVALUAR**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_FINAL DEL SOFTWARE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **PREPARACIÓN DE INFORMES Y ESPECIFICACIONES**

Los documentos de especificación, diseño y requerimientos que posee el programa a implementar se encuentran en la carpeta docs del repositorio de Github donde está almacenado el proyecto.

1. **IMPLEMENTACIÓN DEL DISEÑO**

La implementación se encuentra en el repositorio de github, está realizado en código Java con Javafx.

**Imagen tomada de:**

[**https://www.alamy.es/mujer-de-mediana-edad-con-gafas-de-realidad-virtual-jugando-videojuegos-en-3d-viendo-simulacion-futurista-emocionado-experiencia-image417392506.html**](https://www.alamy.es/mujer-de-mediana-edad-con-gafas-de-realidad-virtual-jugando-videojuegos-en-3d-viendo-simulacion-futurista-emocionado-experiencia-image417392506.html)

1. https://www.geeksforgeeks.org/generics-in-java/ [↑](#footnote-ref-1)
2. https://www.thoughtco.com/definition-of-stack-in-programming-958162 [↑](#footnote-ref-2)
3. https://www.geeksforgeeks.org/queue-data-structure/ [↑](#footnote-ref-3)
4. http://webdiis.unizar.es/~elvira/eda/material0304/TADespec/TAD.pdf [↑](#footnote-ref-4)
5. https://www.tutorialspoint.com/data\_structures\_algorithms/hash\_data\_structure.htm [↑](#footnote-ref-5)
6. https://blog.devmountain.com/what-is-github-and-how-do-you-use-it/ [↑](#footnote-ref-6)