**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.
* El listado de videojuegos debe organizarse dependiendo de la ubicación de cada una de las estanterías, de tal forma que el comprador siga la mejor ruta.
* Si el juego se encuentra agotado, su código no aparecerá en la lista ordenada según la ubicación de su respectiva estantería.
* El cliente debe poder elegir entre 2 algoritmos de ordenamiento para cumplir la tarea de organizar su lista dependiendo de la ubicación de cada estantería.
* Al cliente se le debe asignar un cesto automatizado que le sigue para ir ubicando los juegos que va encontrando, uno encima del otro, siguiendo el orden suministrado en la etapa previa.
* Cada cliente debe tener un tiempo dependiendo de su entrada a la tienda, al cual se le suma el tiempo que tardó el cliente buscando los videojuegos.
* El orden en el que los clientes entran a la caja está dado por el tiempo en el que entró el cliente a la tienda sumado el tiempo que le tomó recoger los juegos.
* Todos los clientes deben hacer una única cola, aunque haya varios cajeros o puntos de atención.
* En caso de haber puntos de atención disponibles, se atenderán en simultáneo tantos clientes como puntos de atención disponibles haya.
* El orden de salida puede variar, pues depende de cuánto se demore en la caja cada cliente, lo cual está dado por la cantidad de elementos que van a comprar.
* Los videojuegos, al momento de ser facturados el orden será de la forma en que el último juego añadido al cesto deberá ser el primero facturado y empacado.
* Después de completar la facturación, debe indicarse el costo total de los juegos listados por cada cliente.

*Definición del problema*

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. **RECOPILATION OF DATA**

En relación con la problemática la cual es dar a conocer el funcionamiento de la tienda a la población caleña, cabe mencionar, las siguientes interrogantes como los algoritmos o implementaciones que harán realidad el funcionamiento del software simulado:

**Generics:** La programación con generics nos permite crear código reusable en múltiples entidades. Con este tipo de código genérico se trabajará automáticamente con el tipo de datos pasados a su parámetro de tipo. También, permiten al compilador informar de muchos errores de compilación que hasta el momento solo se descubrirán en tiempo de ejecución, al mismo tiempo, permiten eliminar los cast simplificando, reduciendo la repetición y aumentando la legibilidad del código.

**Stacks:** La clase Stack es una clase de las llamadas de tipo LIFO (Last In - First Out  en español Último en Entrar - Primero en Salir). Es una pila, la cual tiene una secuencia de elementos del mismo tipo en la que el acceso a la misma se realiza por un único lugar denominado cima: Las operaciones que caracterizan la pila son las de introducir un nuevo elemento sobre la cima (push) y la de extraer el elemento situado en la cima (pop).

**Queue:** Una Cola o Queue es una estructura de datos que sigue la Filosofía FIFO (First In – First Out en español Primero en Entrar - Primero en Salir). Los elementos pueden ser insertados en cualquier momento, pero solamente el elemento que ha estado en la cola más tiempo puede ser removido en cualquier momento.

**TAD:** Un Tipo abstracto de datos (TAD) es un conjunto de datos u objetos al cual se le asocian operaciones. El TAD provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en cómo están implementadas dichas operaciones.

**Tablas Hash:** Una Hashtable Java es una estructura de datos que utiliza una función hash para identificar datos mediante una llave o clave. En otras palabras, sería bastante conveniente poder mantener la capacidad de acceder fácilmente a cualquier descripción a partir de su clave, pero al mismo tiempo utilizar una cantidad razonable de memoria. Estas son dos de las propiedades que posee una **Tabla Hash**.

Por otro lado, es importante mencionar y explicar un poco en donde se va a encontrar contenida toda la información de la aplicación.

**Github:** Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, la plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que, como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo. Git es uno de estos sistemas de control, que permite comparar el código de un archivo para ver las diferencias entre las versiones, restaurar versiones antiguas si algo sale mal, y fusionar los cambios de distintas versiones. También permite trabajar con distintas ramas de un proyecto, como la de desarrollo para meter nuevas funciones al programa o la de producción para depurar los bugs.

Así pues, Github es un portal para gestionar las aplicaciones que utilizan el sistema Git. Además de permitirte mirar el código y descargarte las diferentes versiones de una aplicación, la plataforma también hace las veces de red social conectando desarrolladores con usuarios para que estos puedan colaborar mejorando la aplicación.

1. **BÚSQUEDA DE SOLUCIONES CREATIVAS**

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)