



CRUD ARRAY

Práctica 1

DESCRIPCIÓN BREVE

Documentación referente a la aplicación desarrollada en la asignatura “Programación”

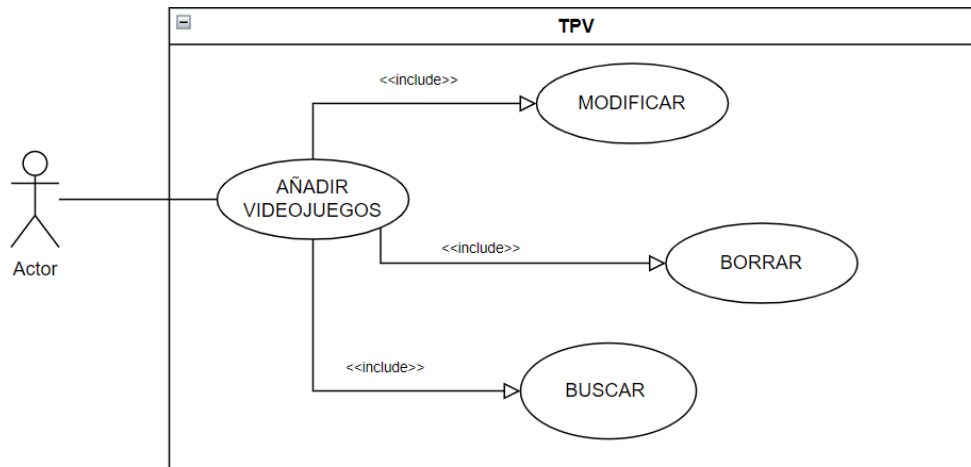
ANGELA GARCIA HUELAMO

Entornos de desarrollo

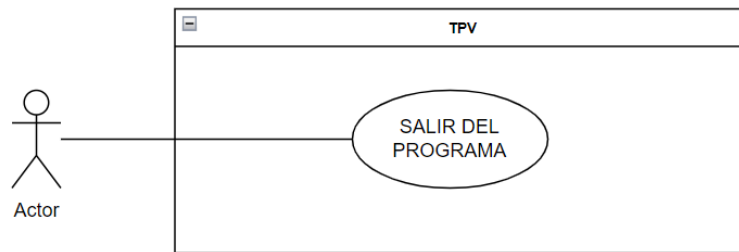
Contenido

ANÁLISIS DE CASOS DE USO	2
ANÁLISIS RFTP	3
GIT-HUB	7

ANÁLISIS DE CASOS DE USO



DESCRIPCIÓN		
EL USUARIO AÑADE VIDEOJUEGOS POR NOMBRE - PRECIO - CÓDIGO. EL USUARIO BORRA - MODIFICA - BUSCA POR DATOS.		
PRECONDICIONES	POSTCONDICIONES	
AÑADIR → NINGUNA BORRAR, MODIFICAR, BUSCAR → AÑADIR	DATOS DE LOS VIDEOJUEGOS AÑADIDOS EL USUARIO PUEDE BORRAR, MODIFICAR Y BUSCAR VIDEOJUEGOS A TRAVÉS DE LOS DATOS CREADOS	
INTERFACES	CLASES	BASES DE DATOS
NINGUNA	NINGUNA	NINGUNA



DESCRIPCIÓN		
EL USUARIO SALE DEL PROGRAMA		
PRECONDICIONES	POSTCONDICIONES	
1) AÑADIR DATOS y/o REALIZAR ACCIONES EL MENÚ 2) INICIAR EL PROGRAMA Y SALIR DIRECTAMENTE	NINGUNA	
INTERFACES	CLASES	BASES DE DATOS
NINGUNA	NINGUNA	NINGUNA

ANÁLISIS RFTP

R1 – Añadir videojuego al TPV (Terminal punto de venta).

R1F1 – Insertar nombre de videojuegos a través de consola (5 nombres)

R1F1T1 – Construir un array de tipo String (longitud de 5)

R1F1T1P1 – Añadir 5 nombres y ver si da algún error de longitud.

R1F2 – Insertar precios de videojuegos a través de consola (5 precios)

R1F2T1 – Construir un array de tipo double (longitud de 5)

R1F2T1P1 – Añadir 5 precios y ver si da algún error de longitud.

R1F3 – Insertar código de videojuegos a través de consola (5 códigos)

R1F3T1 – Construir un array de tipo int (longitud de 5)

R1F3T1P1 – Añadir 5 códigos y ver si da algún error de la longitud.

R2 – Buscar videojuego por nombre, código o precio.

R2F1 – Buscar videojuegos por su nombre.

R2F1T1 – Crear un método llamado searchNombreVideojuegos que recorra un ciclo for y, mediante una variable String extra, busquemos si el nombre que nos da el cliente por consola coincide alguna posición del array. Para ello, nos ayudaremos de una variable boolean, también.

R2F1T1P1 – Buscar un nombre que hayamos añadido anteriormente y ver si lo encuentra dentro del array.

R2F1T1P2 – Buscar un nombre que no se haya añadido y ver si no lo encuentra.

R2F2 – Buscar videojuegos por su precio.

R2F2T1 - Crear un método llamado searchPrecioVideojuegos que recorra un ciclo for y, mediante una variable String extra, busquemos si el nombre que nos da el cliente por consola coincide alguna posición del array. Para ello, nos ayudaremos de una variable boolean, también.

R2F2T1P1 – Buscar un precio que hayamos añadido anteriormente y ver si lo encuentra dentro del array.

R2F2T1P2 – Buscar un precio que no se haya añadido y ver si no lo encuentra.

R2F3 – Buscar videojuegos por su código.

R2F3T1 - Crear un método llamado searchCódigoVideojuegos que recorra un ciclo for y, mediante una variable String extra, busquemos si el nombre que nos da el cliente por consola coincide alguna posición del array. Para ello, nos ayudaremos de una variable boolean, también.

R2F2T1P1 – Buscar un código que hayamos añadido anteriormente y ver si lo encuentra dentro del array.

R2F3T1P2 – Buscar un código que hayamos añadido anteriormente y ver si no lo encuentra.

R3 – Borrar videojuego por nombre, código o precio.

R3F1 – Borrar videojuegos por su nombre.

R3F1T1 – Crear un método llamado deleteNombreVideojuegos que pida al usuario por consola, el nombre a borrar. Recorrer mediante un ciclo for para saber si el nombre dado coincide con alguno guardado en el array de nombres para llevar a cabo la acción de borrar. De ser así, dejará elegir al usuario agregar un nuevo nombre o no.

R3F1T1P1 – Introducir un nombre que hayamos añadido anteriormente e introducir un nombre nuevo.

R3F1T1P2 – Introducir un nombre que hayamos añadido anteriormente y no introducir ningún nombre nuevo, es decir, dejar vacía la posición.

R3F2 – Borrar videojuegos por su precio.

R3F2T1 – Crear un método llamado deletePreciosVideojuego que pida al usuario por consola, el precio a borrar. Recorrer mediante un ciclo for para saber si el precio dado coincide con alguno guardado en el array de precios para llevar a cabo la acción de borrar. De ser así, dejará elegir al usuario agregar un nuevo precio o no.

R3F1T1P1 – Introducir un precio que hayamos añadido anteriormente e introducir un precio nuevo.

R3F1T1P2 – Introducir un precio que hayamos añadido anteriormente y no introducir ningún precio nuevo, es decir, dejar vacía la posición.

R3F3 – Borrar videojuegos por su código.

R3F2T1 – Crear un método llamado deleteCodigosVideojuegos que pida al usuario por consola, el código a borrar. Recorrer mediante un ciclo for para saber si el código dado coincide con alguno guardado en el array de códigos para llevar a cabo la acción de borrar. De ser así, dejará elegir al usuario agregar un nuevo código o no.

R3F1T1P1 – Introducir un código que hayamos añadido anteriormente e introducir un código nuevo.

R3F1T1P2 – Introducir un código que hayamos añadido anteriormente y no introducir ningún código nuevo, es decir, dejar vacía la posición.

R4 – Modificar videojuego por nombre, código o precio.**R4F1 – Modificar videojuego por su nombre.**

R4F1T1 – Crear un método llamado updateNombreVideojuego que pida al usuario por consola, el nombre a modificar. Recorrer mediante un ciclo for para saber si el nombre dado coincide con alguno guardado en el array de nombres para llevar a cabo la acción. De ser así, dejará modificar el nombre.

R4F1T1P1 – Introducir un nombre que hayamos añadido anteriormente y sustituirlo por otro nombre nuevo.

R4F2 – Modificar videojuego por su precio.

R4F1T1 – Crear un método llamado updatePrecioVideojuego que pida al usuario por consola, el precio a modificar. Recorrer mediante un ciclo for para saber si el precio dado coincide con alguno guardado en el array de precios para llevar a cabo la acción. De ser así, dejará modificar el precio.

R4F1T1P1 – Introducir un precio que hayamos añadido anteriormente y sustituirlo por otro precio nuevo.

R4F3 – Modificar videojuego por su código.

R4F1T1 – Crear un método llamado updateCodigoVideojuego que pida al usuario por consola, el código a modificar. Recorrer mediante un ciclo for para saber si el código dado coincide con alguno guardado en el array de códigos para llevar a cabo la acción. De ser así, dejará modificar el código.

R4F1T1P1 – Introducir un código que hayamos añadido anteriormente y sustituirlo por otro código nuevo.

R5 – Salir del programa.

R5F1 – Añadir una opción para que el programa finalice.

R5F1T1 – Crear un método llamado outMenu que, cuando el usuario escriba el número correcto para llamar a este método, finalice el programa.

R5F1T1P1 – Introducir el número correcto para salir del programa.

GIT-HUB

<https://github.com/garciahuelamo/CRUDArray.git>