

PRÁCTICA #2

PROGRAMACIÓN ORIENTADA A OBJETOS

PROGRAMACIÓN EN PYTHON

Integrantes:
Jean Carlo Montoya Castro
Simon Henao Angarita
Michael Steven Ramirez Reyes
Juan José García Vásquez

Profesor:
Jaime Alberto Guzman Luna

Universidad Nacional de Colombia Sede Medellín
Facultad de Minas
Medellín, Colombia.
Mayo 2022

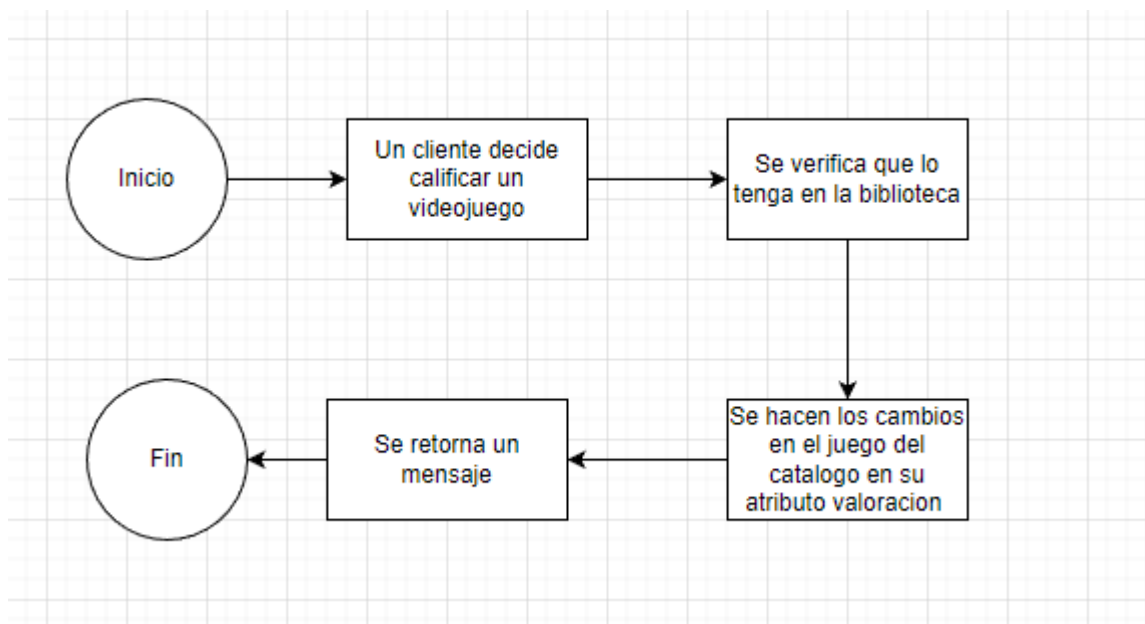
Descripción 5 funcionalidades:

1. Calificar Videojuego: describe el proceso de la valoración hecha por un cliente a los videojuegos del catálogo.

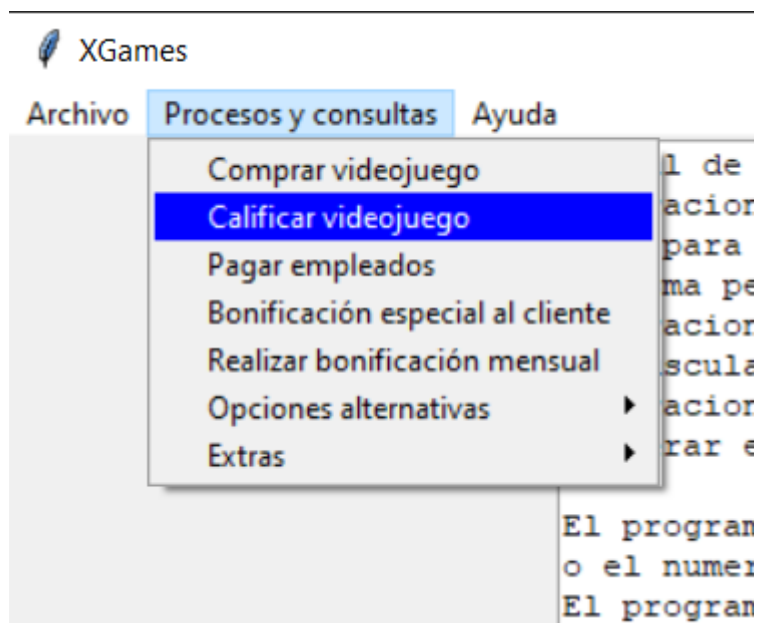
Clases que intervienen:

- Cliente
- Catálogo
- Videojuego

Secuencia:



Menú de opciones:



Llenado manual de calificación:

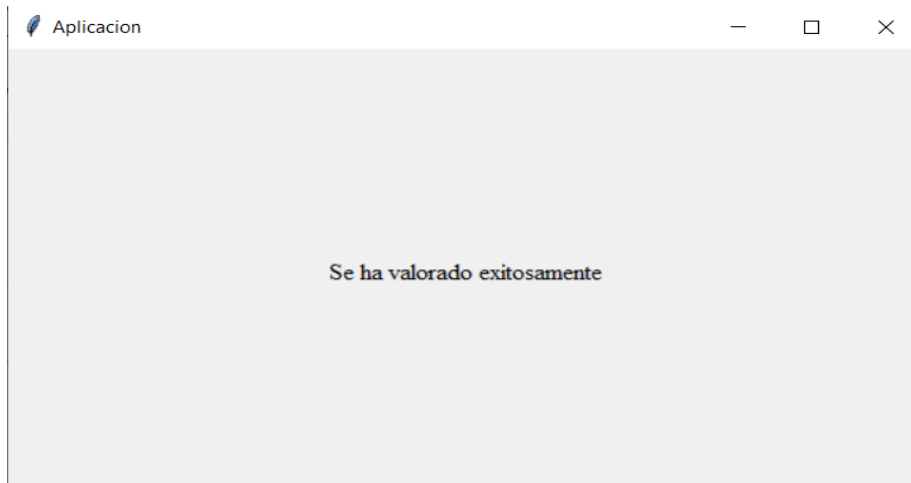
Calificar

Calificar manualmente un videojuego

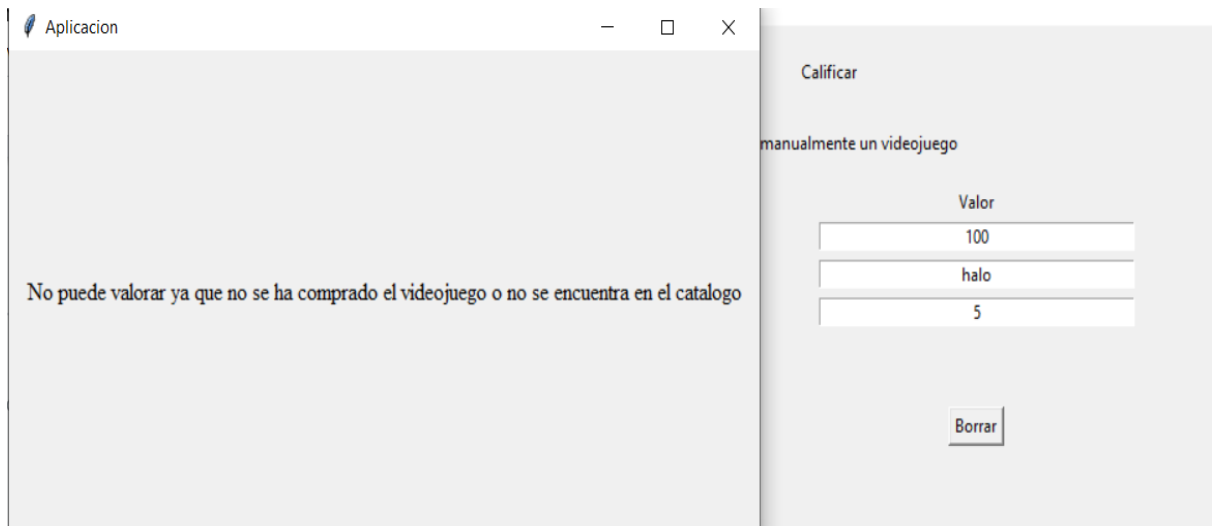
| Criterios | Valor |
|--------------|-------------|
| Cedula | 100 |
| Videojuego | silent hill |
| Calificacion | 5 |

Resultados en pantalla:

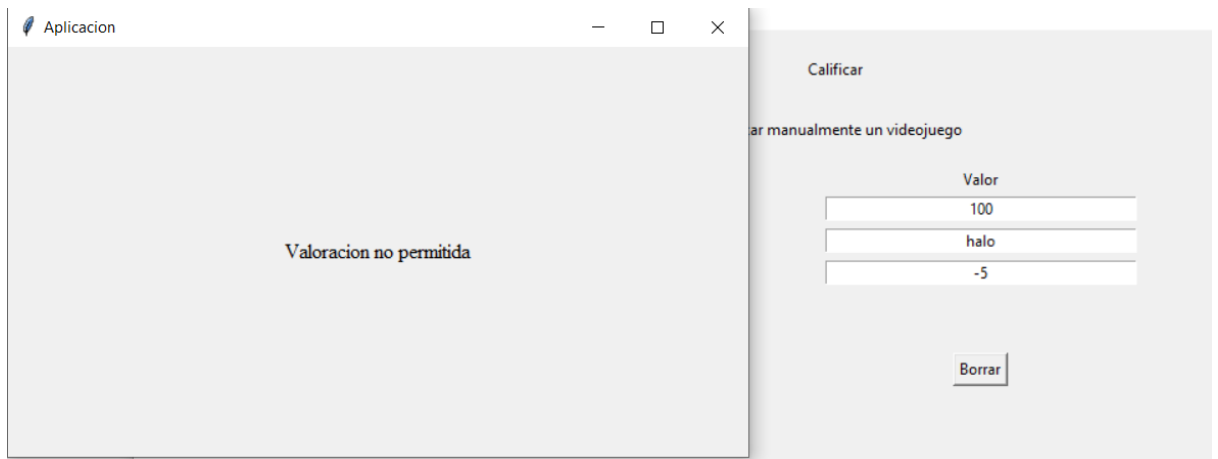
Caso 1. Valorado correctamente:



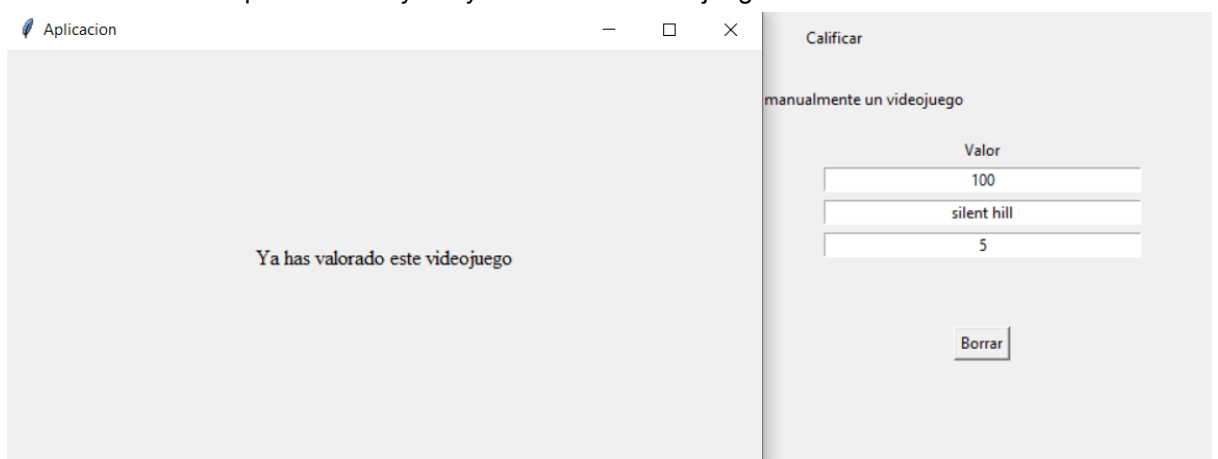
Caso 2.El cliente no posee el videojuego en su biblioteca o este no se encuentra en el catálogo:



Caso 3.El cliente ingresa un número de valoración no permitido:



Caso 4.En caso de que el cliente ya haya valorado el videojuego con anterioridad:

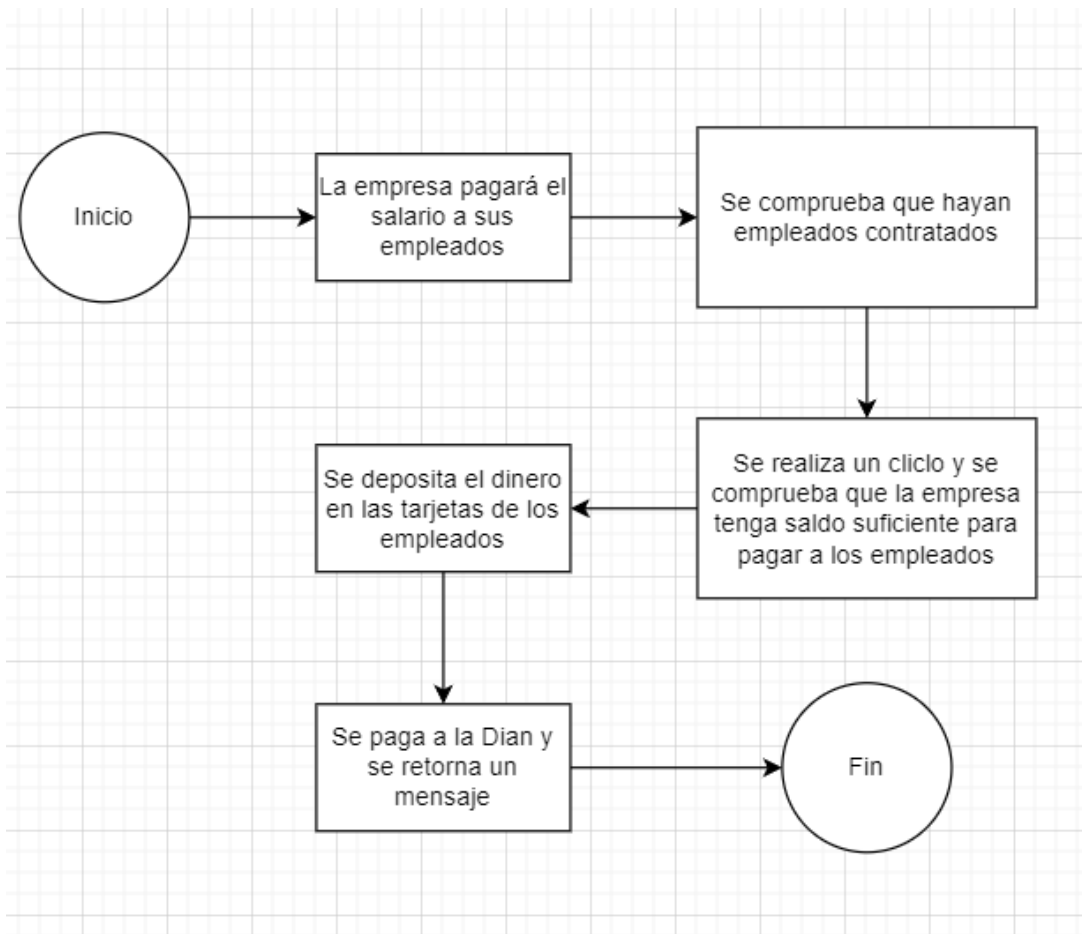


2. Pagar empleados: describe el proceso de pago de la Empresa a sus empleados


Clases que intervienen:

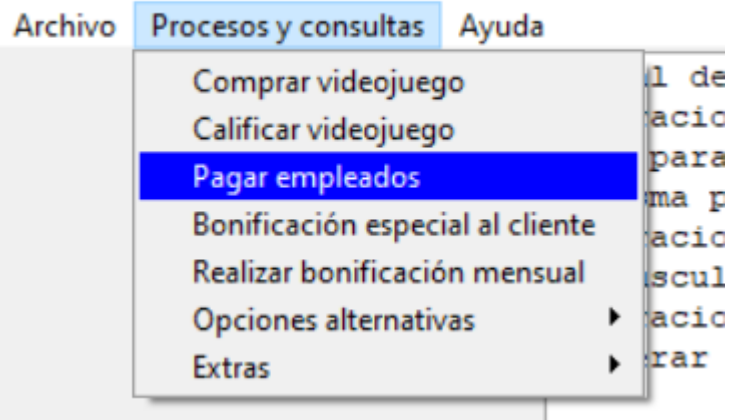
- Empresa
- Tarjeta
- Empleado
- Dian

Secuencia:




Menú de opciones:

 XGames




Resultados en pantalla:

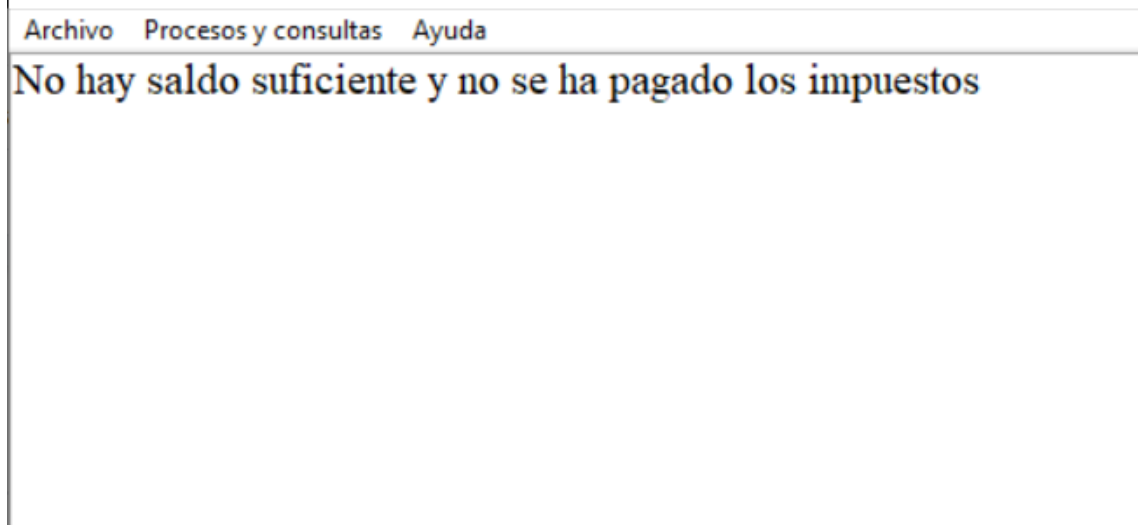
Caso 1. La empresa realiza el pago exitosamente.

 XGames

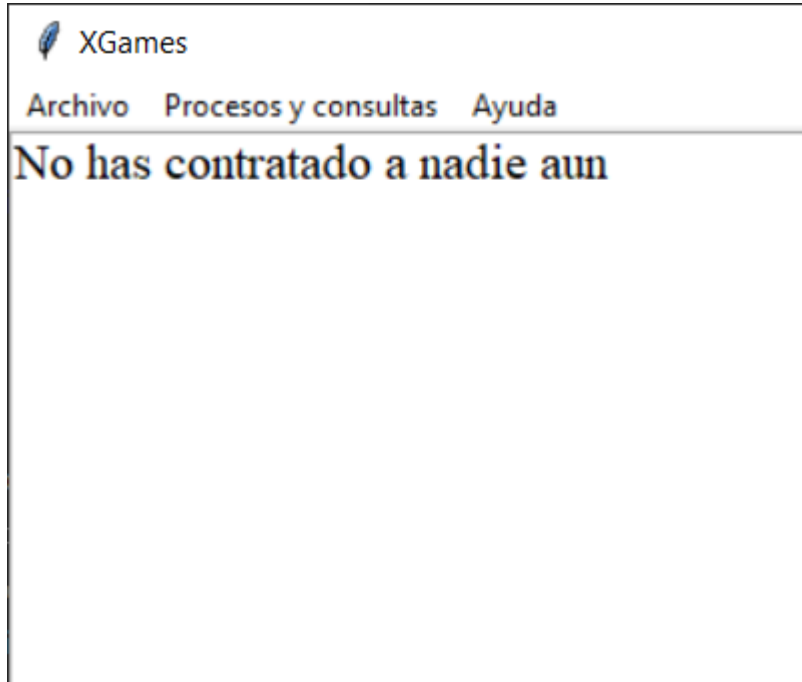


Caso 2. La empresa no tiene dinero para pagar a sus empleadas ni los impuestos.

 XGames



Caso 3. La empresa no tiene ningún empleado.

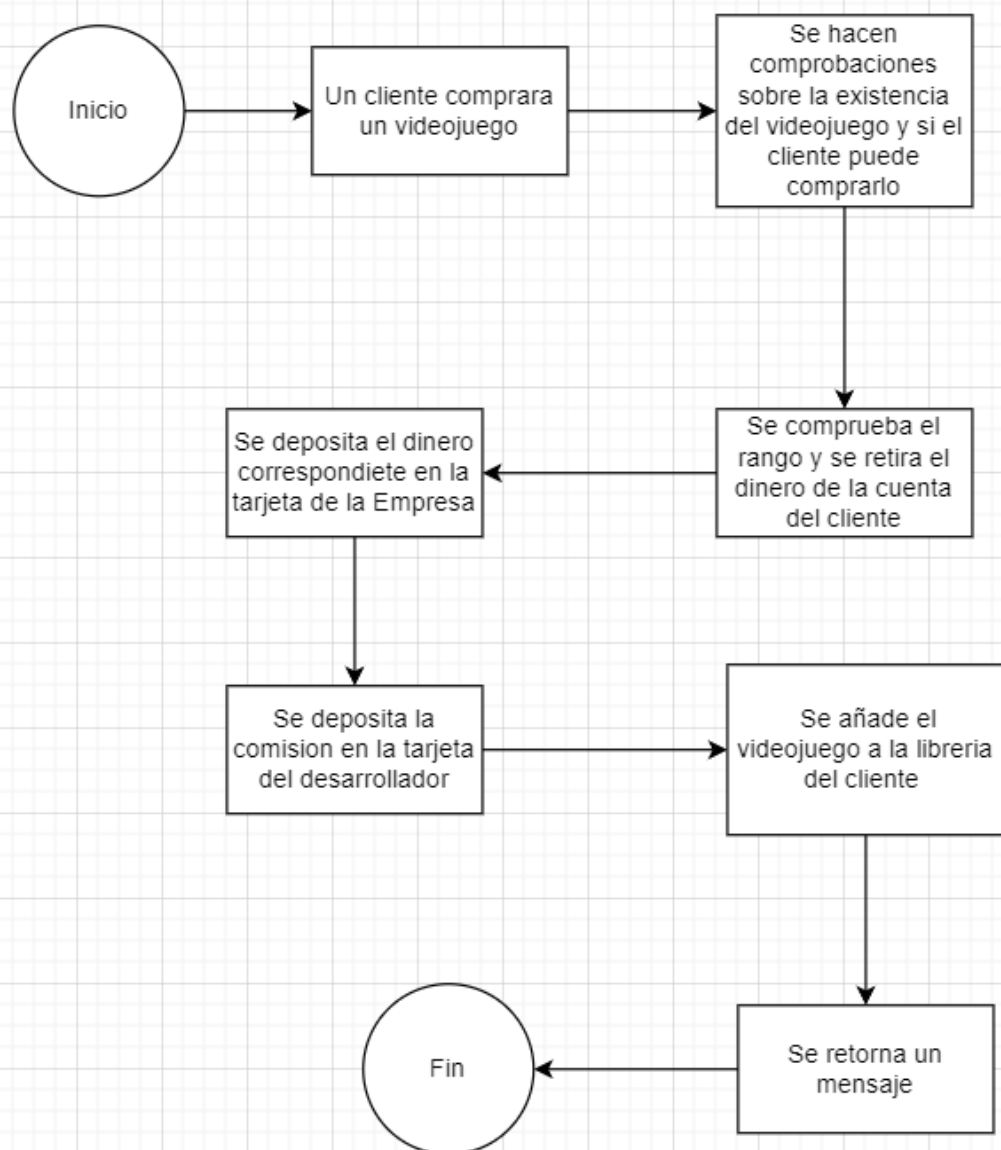


3. Comprar videojuego: describe el proceso de un cliente a la hora de comprar un videojuego

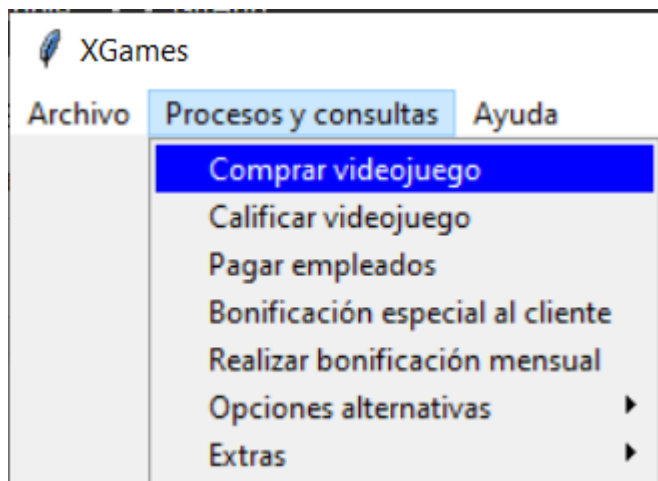
Clases que intervienen:

- Cliente
- Empresa
- Desarrollador
- Tarjeta
- Catálogo
- Rango

Secuencia:



Menú opciones:



Comprar un videojuego

Comprarle manualmente un juego a un cliente

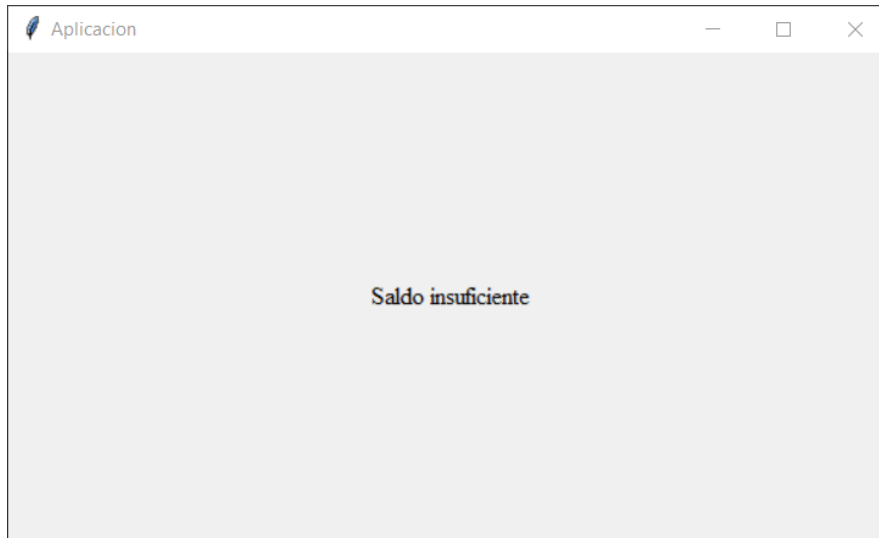
| Criterios | Valor |
|------------|----------------------|
| Cedula | <input type="text"/> |
| Videojuego | <input type="text"/> |

Resultados en pantalla:

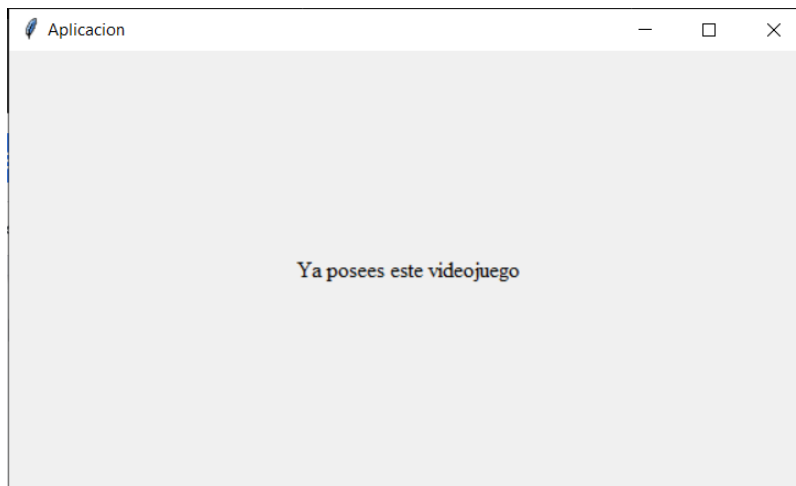
Compra exitosa:



Saldo insuficiente:



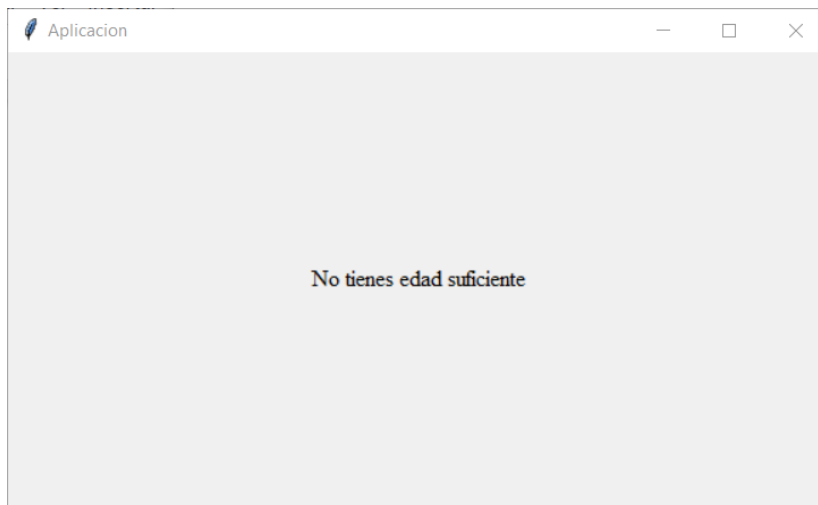
Ya posees este videojuego:



Videojuego no existe:



No tienes edad suficiente:

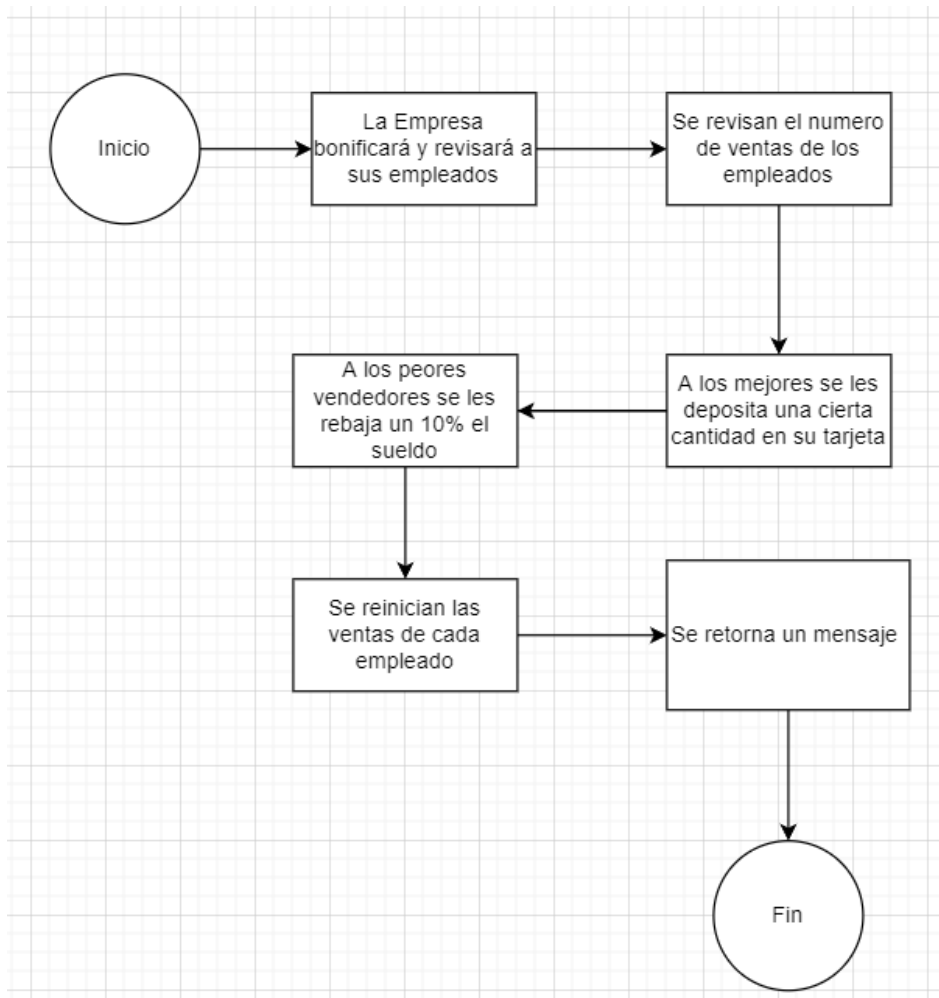


4. Bonificación empleado: se bonificará a los empleados cuyas ventas sean considerablemente altas y se rebajara el saldo de los que tengan las ventas por debajo de lo solicitado

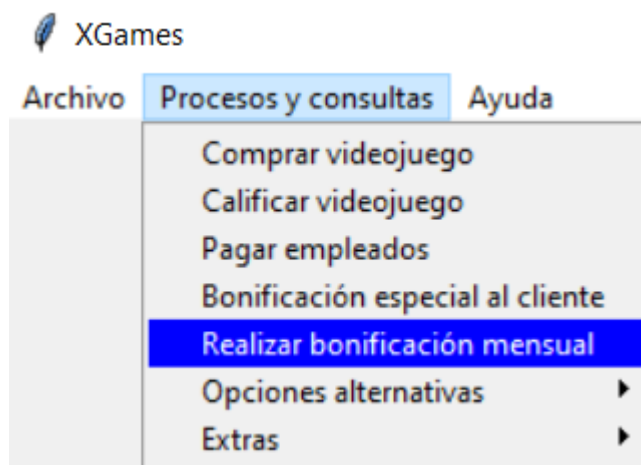
Clases que intervienen:

- Empresa
- Empleado
- Tarjeta

Secuencia:

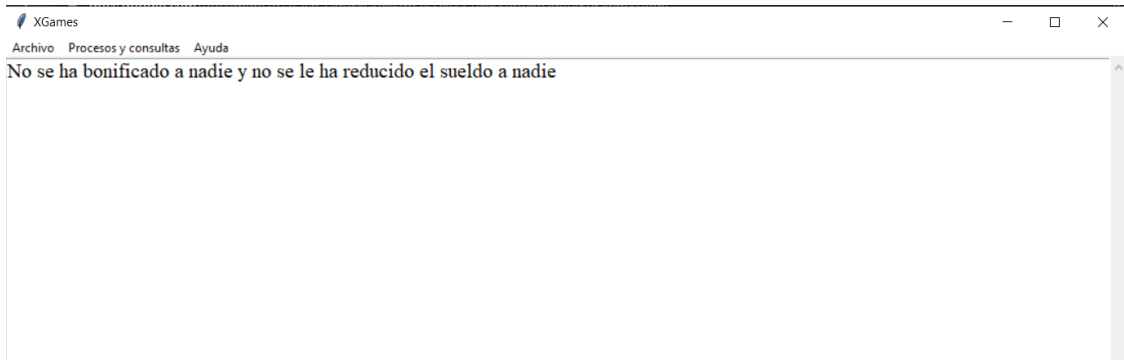


Menú de opciones:

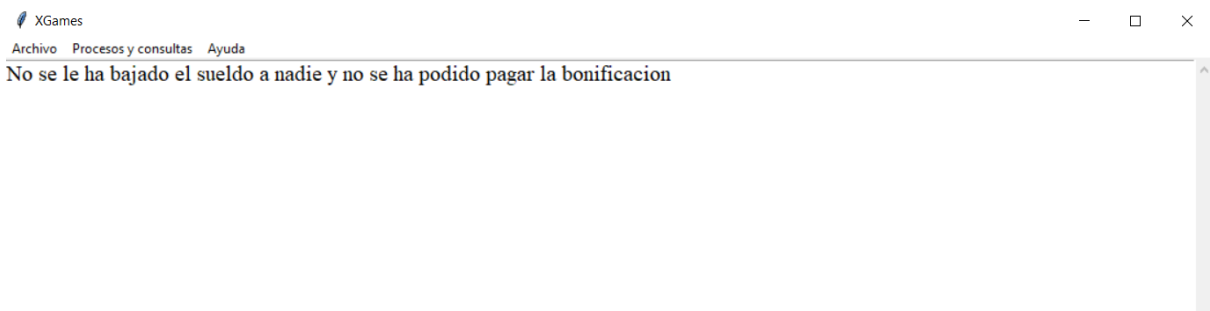


Resultados en pantalla:

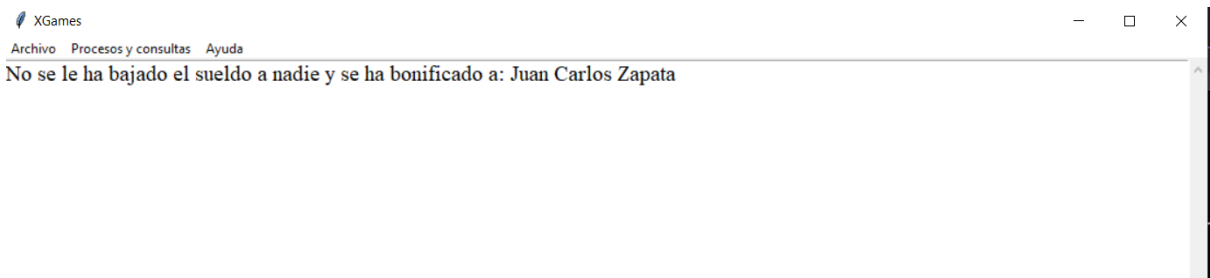
No se ha bonificado a nadie y no se le ha reducido el sueldo a nadie:



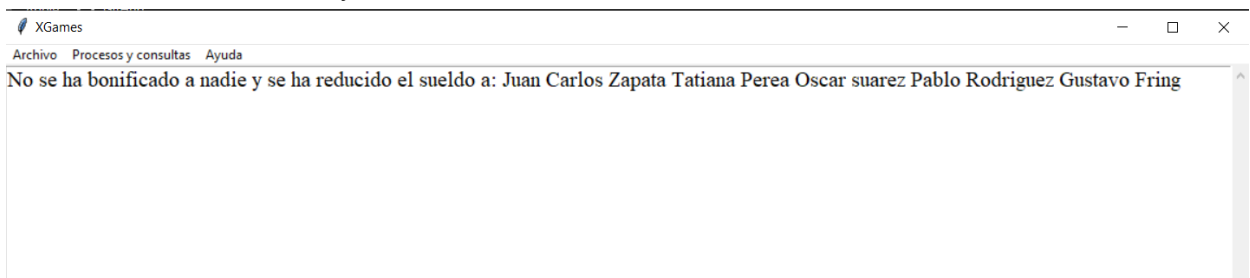
No se le ha bajado el sueldo a nadie y no se ha podido pagar la bonificación:



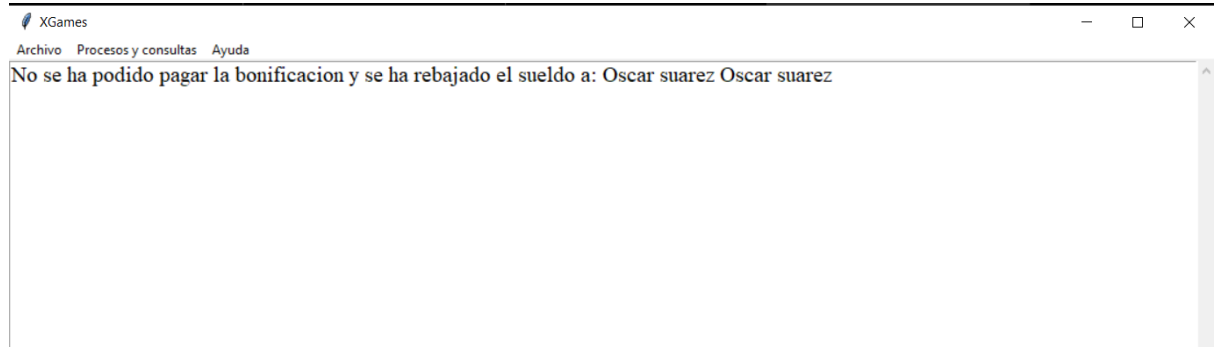
No se le ha bajado el sueldo a nadie y se ha bonificado a:



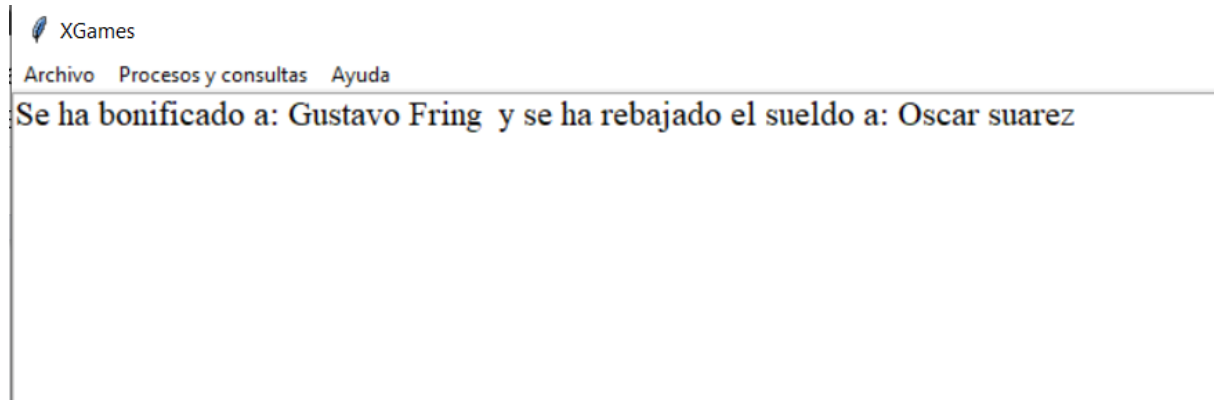
No se ha bonificado a nadie y se ha reducido el sueldo a:



No se ha podido pagar la bonificación y se ha rebajado el sueldo a:



Se ha bonificado a: Gustavo Fring y se ha rebajado el sueldo a: Oscar suarez:

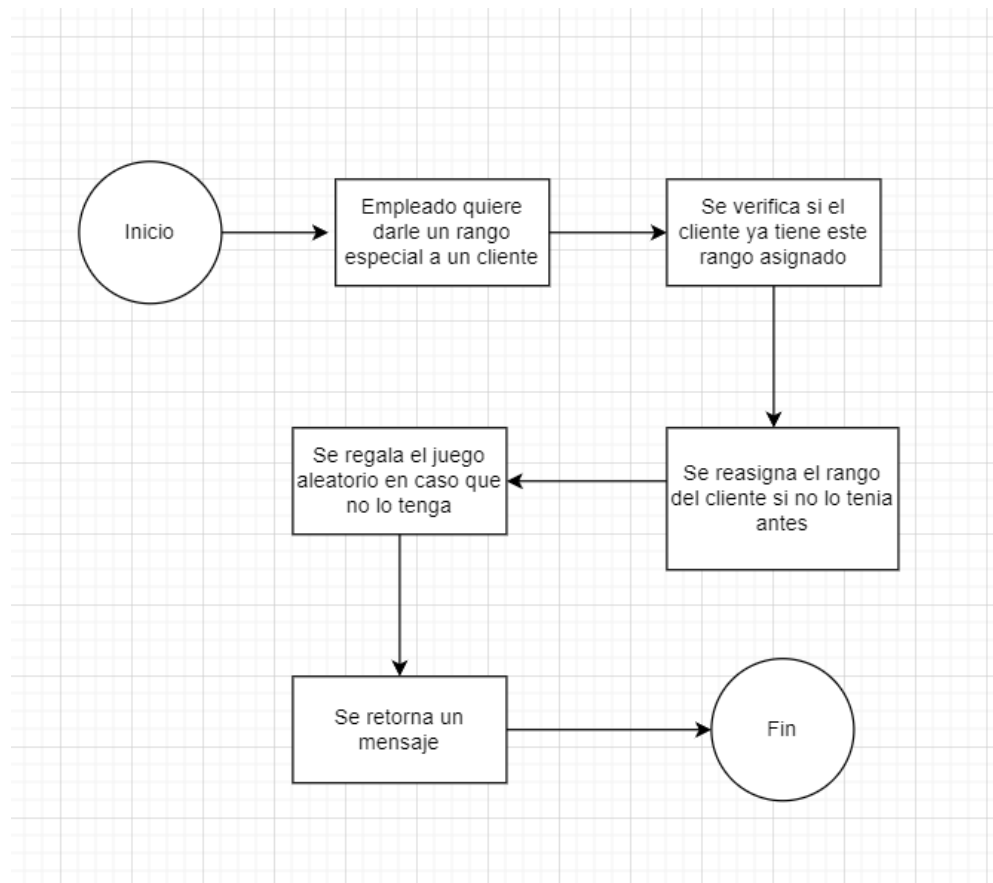


5. Rango especial: un empleado asigna el rango de un cliente de la lista de clientes, se le asigna el rango Especial y se le regala un juego del catálogo al azar en caso que tenga suerte

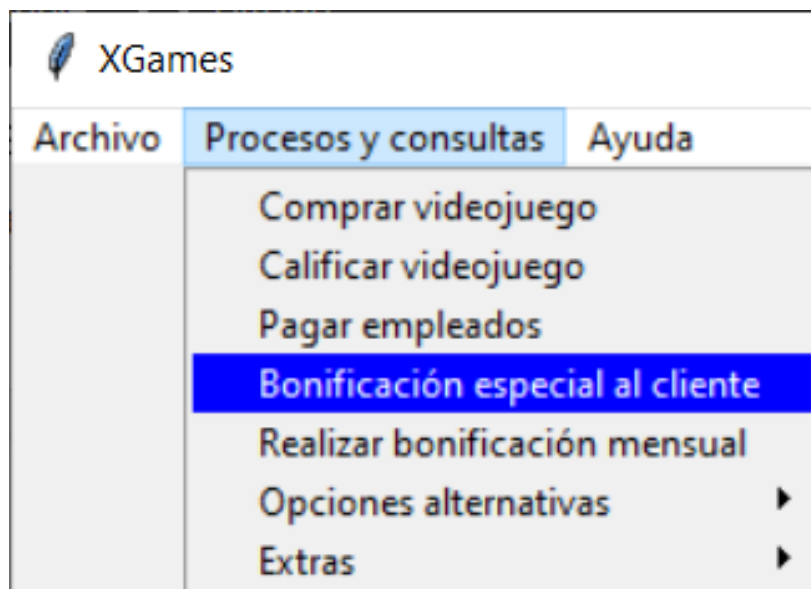
Clases que intervienen:

- Cliente
- Empleado
- Catálogo
- Videojuego
- Rango

Secuencia:



Menú de opciones:



Bonificar a un cliente

Cambiarle el RANGO a un cliente

| Criterios | Valor |
|-----------|----------------------|
| Cedula | <input type="text"/> |

Resultados en pantalla:

El cliente ya tiene el rango:

Aplicacion

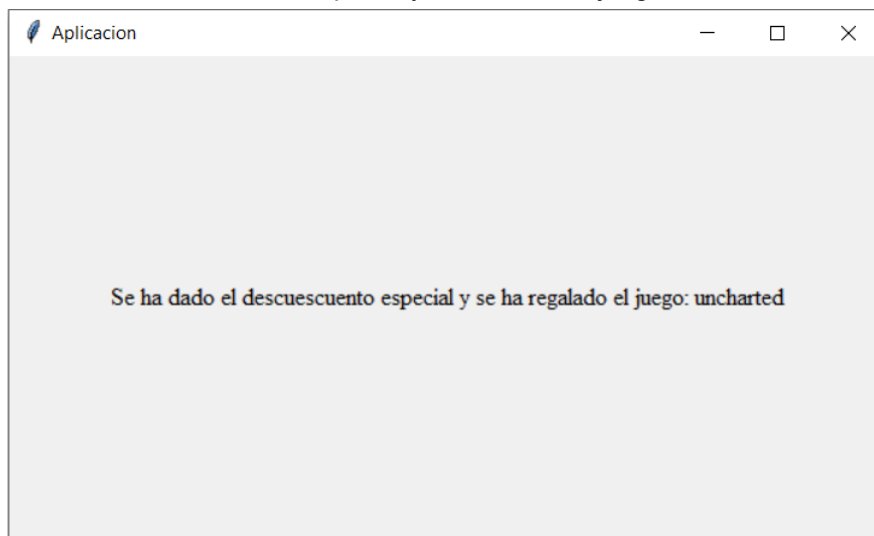
El cliente ya tiene el rango

Se ha reasignado el pero no se ha tenido suerte con el juego:

Aplicacion

Se ha reasignado el descuento especial pero no se ha tenido suerte con el juego

Se ha dado el descuento especial y se ha dado el juego:



Objetos:

Persona: Permite guardar información personal y la reutilización de código.

Empleado: Son los trabajadores de la empresa la clase Empleado hereda de la clase Persona. Esta se encarga de generar descuentos, vender videojuegos, modificar precios.

Empresa: Clase encargada de todo lo relacionado con la gestión de los empleados y los recursos de la empresa, esta clase se relaciona con Persona, Desarrollador, Tarjeta, Cliente, Empleado, Videojuego y Dian.

Tarjeta: Es la clase encargada de manejar el dinero puede ingresarlo o sacarlo, está relacionada con la clase Persona, Desarrollador, Empresa, Cliente y Empleado. Aquí se encuentra el saldo de cada uno de los actores con los que está relacionada.

Desarrollador: Clase donde se pide el ingreso de un nuevo videojuego al catálogo de la empresa, cobrando una comisión por la venta de los videojuegos. Esta clase está relacionada con Empresa, Videojuego, Tarjeta y Dian.

Dian: Clase abstracta que implementa de Gobierno y que obliga a la empresa y desarrolladores a pagar impuestos.

Gobierno: Interface que calcula la cantidad de impuestos que deben pagar la empresa y los desarrolladores.

Rango: Es un enum que caracterizará al cliente con el número de compras, este le dará descuentos dependiendo del rango en que se encuentre.

Catálogo: Contiene una lista de videojuegos estática, sirve también para buscar videojuegos y verificar su existencia.

Videojuego: Guarda toda la información del producto de venta que son los videojuegos. Adicionalmente esta verifica si la edad del cliente es apta para el videojuego.

Manual de usuario:

Manual de usuario:

Aclaración 1: este programa fue hecho en un primer momento para multi-usuarios, pero para fines de evaluación se ha modificado para que todo sea manejado por una misma persona.

Aclaración 2: todo lo que se ingrese que sea tipo "String" se deberá ingresar en minúscula.

Aclaración 3: Cerrar el programa desde la opción en menu, de lo contrario podría generar errores.

Aclaración 4. Si se quieren guardar los datos de ingreso debes darle en el primer botón del menú "Archivo" y luego oprimas salir, esto te llevará a la ventana de inicio donde nuevamente si quieres guardar debes usar la opción de salir.

El programa ofrece una serie de opciones encontradas en el menú, en procesos y consultas se encontrarán las 5 funcionalidades y en las opciones desplegadas las funciones extra.

El usuario deberá clicar cada una de estas para acceder a cada ventana de las funcionalidades, ya encontrándose en estas ventanas el usuario al hacer las acciones y al llenar los campos requeridos oprimirá el botón aceptar; seguido a esto se le desplegará una ventana de diálogo, ya sea de un proceso exitoso o de un mal manejo con una alerta.

Si el usuario quiere hacer de nuevo la acción y llenar el campo nuevamente se dispondrá a usar el botón borrar y así podrá llenar el formulario nuevamente.

Cabe aclarar que el programa terminará de ejecutar al seleccionar la opción 9, de lo contrario seguirán funcionando las otras opciones.

Finalmente, luego de la ejecución de cada opción aparecerá un mensaje en pantalla.

Al abrir el programa, se encuentran unas capturas, fotos y descripciones de los creadores, luego en la ventana principal del admin aparece:

Archivo:

- 1) **Salir:** devuelve a la ventana anterior
- 2) **Aplicación:** da un breve resumen del objetivo de la empresa

Procesos y consultas:

- 1) **Comprar videojuego:** ingresar la cédula del cliente y nombre del juego
 - 2) **Calificar videojuego:** ingresa la cédula del cliente, el nombre del juego y la calificación(0-10)
 - 3) **Pagar empleados(automática):** paga a los empleados
 - 4) **Bonificación especial a cliente:** ingresa el nombre del cliente
 - 5) **Realizar bonificación mensual(automática):** bonifica a los empleados
 - 6) **Opciones alternativas:** se obtiene:
 - 6.1) **Mostrar detalles empleados(automática):** muestra los detalles de los empleados.
 - 6.2) **Mostrar detalles empresa(automática):** muestra detalles de la empresa útiles.
 - 6.3) **Mostrar catálogo(automática):** muestra el catálogo y aspectos de cada juego.
 - 6.4) **Mostrar cliente específico:** ingresa la cédula del cliente al que quiere ver.
 - 7) **Extras:** se obtiene:
 - 7.1) **Añadir videojuego:** añade manualmente un videojuego
- Nota:** Tener en cuenta que en el apartado de calificaciones las edades para los videojuegos estándar, solo se permitirán: +18, +16, +12, +7, +3.
- 7.2) **Contratar empleado:** se añade un empleado manualmente.
 - 7.3) **Ingresa dinero tarjeta:** se selecciona al cliente y la cantidad de dinero

Ayuda:

- 1) **Acerca de:** da el nombre de los autores

Información importante:

Cédula de los clientes actuales:

- 100
- 200
- 300
- 400

- 500

Juegos catálogo actual:

- fifa
- dark souls
- gta v
- silent hill
- uncharted

Juegos en la base de datos: son los del catálogo y :

- minecraft
- forza horizon
- call of duty
- los sims
- crash bandicoot

Cédulas empleados actuales:

- 12101509
- 66078005
- 10101000
- 10351022
- 10257622

Explicación excepciones:

Vamos a explicar en este caso las 6 excepciones que aparecen en el nivel más bajo de la jerarquía de herencia(en nuestro caso hay una adicional)

Las clases que heredan de FieldException son EmptyException, NumberException y NegativeException, la primera en este caso se encarga de informar si una entrada en algún formulario no se llenó completamente, la segunda se encarga de informar si se ingresó un dígito en un lugar donde debía ubicarse un String y la última se encarga de informar si alguno de los dígitos ingresados es negativo(lo cual para nuestro programa no tiene sentido y puede arruinar fácilmente el programa)

Por último, las clases que heredan de UserException son NotFoundException, ClienteException, CashException y RangoException. La primera se encarga de informar si el videojuego buscado no está disponible o no existe, la segunda informa si algún cliente buscado no existe en el programa, la tercera informa si a la hora de comprar un videojuego, el cliente no cuenta con dinero suficiente para hacer la transacción y la última nos informa cuando se está tratando de dar una bonificación especial a un cliente que ya la tiene asignada

Diagrama de clases UML:

