**INSTITUCO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY**

**PROYECTO FINAL**

**MATERIA:**

Fundamentos de Programación

**PROFESOR:**

Francisco Alonso Alvizo Coronado

**EQUIPO:**

Carlos René Arteaga Hernández

Esteban Azuara Aguirre

Alejandro Sanchez Ponce

Jorge Ricardo Cabrales Reyes

**FECHA DE ENTREGA:**

25 de Noviembre del 2016

Código en NetBeans:

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package pkgfinal;

// Librerias utilizadas para la elaboracion del programa:

import java.util.\*;

import java.io.File;

import java.io.FileReader;

import java.io.PrintWriter;

import java.io.BufferedReader;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Calendar;

/\*\*

\*

\* @author virtual

\*/

public class Final {

// Declaracion de variables globales:

public static int total = 100,total2=100;

// Limite de produuctos que se pueden registrar en el inventario

public static int indice=0,iE=0,Cantidad=0,clave2=0,indice2=0,ticket=0;

// Variables utilizadas para crear el registro del Inventario:

public static int [] rC = new int [total]; // Registro de la clave del rpoducto

public static String [] rD = new String [total]; // Registro de la descripccion del producto

public static int [] rT = new int [total]; // Registro del Total de productos en existencia

public static double [] rP = new double [total]; // Registro del precio unitario del producto

// Variables utilizadas para generar las notas de venta:

public static int [] nTicket = new int [total];

public static int [] nC = new int [total]; // Nota de la clave

public static String [] nD = new String [total]; // Nota de la descripcion

public static int [] nT = new int [total]; // Nota de la cantidad a comprar

public static double [] nP = new double [total]; // Nota del precio unitario

public static double [] nPST = new double [total]; //Nota del Importe

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

//

leerInventario("c:/temp/Inventario.txt");

int opc=0; // La opcion del usuario al inicio del programa es cero

while (opc!=4){ // Condicion para cuando el usuario eliga la opcion numero cuatro

Menu(); // Impresion del menu

Scanner entradaOpcion = new Scanner(System.in); // Escanner por teclado de la opcion elegida por el usuario

opc = entradaOpcion.nextInt(); // Registro de la opcion ingresada por el usuario

switch(opc){

case 1: // Metodos a utilizar para la opcion numero uno:

agregarInventario(); // Metodo para agregar productos nuevos al inventario

break;

case 2: // Metodos a utilizar para la opcion numero dos:

Venta(); // Metodo para el registro de ventas

break;

case 3: // Metodos a utilizar para la opcion numero tres:

grabarInventario("c:/temp/inventario.txt"); // Metodo para grabar

// los registros del Inventario

grabarNota("c:/temp/cierre"); // Metodo para grabar la nota

// de las ventas realizadas

break;

}

}

}

public static void Menu(){ // Metodo para la impresion del menu:

System.out.println(" \*\*\* Bienvenido a la miscelanea EL POLLO BRUJO \*\*\* ");

System.out.println(" \*\*\* Por favor introduzca una opcion del menu \*\* ");

System.out.println("1.- Registrar Inventario"); // Opcion numero uno

System.out.println("2.- Registrar Venta"); // Opcion numero dos

System.out.println("3.- Cierre del Dia"); // Opcion numero tres

System.out.println("4.- Salir"); // Opcion numero cuatro

}

public static void leerInventario (String nombreArchivo){

// Paso 1.- Instanciamos un objeto de la clase File y una variable cadena

File archivo = new File(nombreArchivo + "Inventario.txt");

String cadena = "";

try {

FileReader lectura = new FileReader(nombreArchivo);

BufferedReader bufferL = new BufferedReader(lectura);

//Paso 2.- Recorremos el archivo.

while (cadena!=null){ //Mientras la cadena no sea nula

cadena = bufferL.readLine(); // Leemos líena por línea el archivo.

String[] cadena2= cadena.split(",");

rC[indice]=Integer.parseInt(cadena2[0]); // Lectura de las lineas que

//

rD[indice]=cadena2[1];

rT[indice]=Integer.parseInt(cadena2[2]);

rP[indice]=Double.parseDouble(cadena2[3]);

indice = indice + 1;

}

//Paso 3.- Cerramos las instancias de BufferedReader y FileReader.

bufferL.close();

lectura.close();

} catch (Exception e) {

System.out.println(" \*\*\* ESTE ARCHIVO NO EXISTE \*\*\* ");

}

return;

}

public static boolean busqueda(int clave){ // Metodo para buscar un producto en base a la clave asignada:

iE = 0; // Variable para realizar la busqueda de la clave del producto

boolean encontrado = false; // Variable con la cual confirmamos

// que la clave del producto ha sido encontrada

for(int i=0; i<indice; i++){ // Condicion usada para encontrar la clave del producto

if(clave==rC[i]){ // Si la clave del producto se encuantra la variable

// "enconctrado" se vuelve true

encontrado = true;

iE = i; // La variable "iE" toma el valor del arreglo

}

}

return encontrado; // Se devuelve la clave del producto

}

public static boolean busqueda2(String clave){ // Metodo para buscar

iE = 0; // Variable utilizada para buscar la clave del producto con "\*"

boolean encontrado = false; // Se declara la variable "encontrado" como falsa

String[] cadena;

if (clave.indexOf(42) != -1){ // Condicion para saber si la variable "clave"

// posee o no un asterisco

cadena = clave.split("\\\*"); // Si la variable "clave" contiene un asterisco

Cantidad = Integer.parseInt(cadena[0]); // separamos la cadena en cantidad

clave2 = Integer.parseInt(cadena[1]); // separamos la parte restante en la variable "clave"

}else{

Cantidad = 1; // Si la variable "clave" no contiene asterisco

// esta variable va a tomar el valor de uno (Un solo producto)

clave2 = Integer.parseInt(clave); // Funcion para transformar la variable "clave" de String a un entero

}

for(int i=0; i<indice; i++){

if(clave2 == rC[i]){ // Condicion para saber si la variable "clave" existe en el arreglo

encontrado = true; // Si la variable "clave" existe

iE = i; // Esta toma el valor del arreglo

}

}

return encontrado; // Se regresa la variable con el valor del arreglo

}

public static void agregarInventario(){ // Metodo utilizado para ingresar

// nuevos productos al inventario

int clave; // Declaracion de la variable "clave" como tipo entero

clave=Integer.parseInt(solicitarDato("Ingresa la Clave del producto:"));

// Se le solicita al usuario que ingrese la clave del producto para agragarlo al inventario

if (busqueda(clave) == true){ // Si la variable "clave" existe

System.out.println("ESTE CLAVE YA EXISTE"); // Se le notifica al usuario que la clave

// que este ingreso ya se encuentra registrada en el arreglo

}else{

// Se agrega:

rC[indice] = clave; // Registro de la clave del producto

rD[indice] = solicitarDato("Ingresa la Descripcion del producto:"); // Registro de la descrippcion

// del producto

rT[indice] = Integer.parseInt(solicitarDato("Ingresa la Existencia de productos:")); // Registro de la cantidad

// de productos existentes en el inventario

rP[indice]=Double.parseDouble(solicitarDato("Ingresa el precio unitario del producto:")); // Registro del precio unitario

// del producto

System.out.println("REGISTRO ALAMCENADO CON EXITO"); // Se le notifica al usuario que el registro de todos

// los campos se ha ejecutado de manera exitosa

indice++; // Se aumenta "indice" para que el usuario continue

// registrando mas productos en el inventario

mostrarInventario(); // Metodo que le muestra al usuario el inventario

}

}

public static String solicitarDato(String Mensage){ // Metodo para solicitar

String Dato; // Se crea la variable "Dato" de tipo String

System.out.println(Mensage); // Despliega el mensaje al usuario

Scanner entradaDato = new Scanner(System.in); // Escanea la entrada del dato

// ingresado por el usuario

Dato = entradaDato.nextLine(); // Se registra el dato ingresado por el usuario

return Dato; // Se devuelve la variable "Dato"

}

public static void mostrarInventario(){ // Metodo utilizado para mostrar el Inventario al usuario

// Mensaje que indica el total de articulos que se encuentran en el inventario:

System.out.println("=================================");

System.out.println("TOTAL DE ARTICULOS EN INVENTARIO");

System.out.println("=================================");

for (int i=0;i<=indice-1;i++){

// Funcion que muestra cada producto que se encuentra en el inventario

// cons sus respectivos campos ( Clave, descripcion, cantidad y precio unitario )

System.out.println(rC[i]+" "+rD[i]+" "+rT[i]+" "+rP[i]);

}

// Mensaje que indica el final del inventario que se le esta mostrando al usuario

System.out.println("=================================");

System.out.println(" FINAL DEL INVENTARIO");

System.out.println("=================================");

}

public static void Venta(){ // Metodo utilizado para realizar el proceso de venta de productos

String clave=""; // Funcion para aceptar cualquier tipo de cadena

ticket++; // Funcion para ir aumentando los tickets de la venta de productos, hasta que

// el usuario decida salir de la ejecucion de la opcion de ventas

while (clave.compareTo("Q")!=0){ // Condicion que le permite al usuario seguir ejecutnado

// la opcion de ventas, mientras este no presione la tecla "Q"

clave=solicitarDato("Producto:"); // Metodo que le solicita al usuario

// la clave del producto para buscarla en el arreglo

if (clave.compareTo("Q")!=0){ // Si el ausuario ingresa la tecla "Q"

// termina de ejecutar la opcion de ventas

System.out.println("Vendido ---> Ticket Clave Descrpcion Cantidad Precio Importe");

if (busqueda2(clave) == true){// Si la clave ingresada por el usuario

// se encuentra en el arreglo se realiza la venta

//System.out.println("Antes --->"+rC[iE]+" "+rD[iE]+" "+rT[iE]+" "+rP[iE]);

nTicket[indice2] = ticket;

nC[indice2] = rC[iE]; // Nota de la clave

nD[indice2] = rD[iE]; // Nota de la descripcion

nT[indice2] = Cantidad; // Nota de la cantidad a comprar

nP[indice2] = rP[iE]; // Notda del precio

nPST[indice2] = Cantidad\*rP[iE]; //Suobtotal

rT[iE]=rT[iE]-Cantidad;//modificar inventario

//System.out.println("Despues --->"+rC[iE]+" "+rD[iE]+" "+rT[iE]+" "+rP[iE]);

System.out.println("Vendido --->"+nTicket[indice2]+" "+nC[indice2]+" "+nD[indice2]+" "+Cantidad+" $"+nP[indice2]+" $"+nPST[indice2]);

indice2++;

}else{

System.out.println("ESTE CLAVE NO EXISTE"); // Mensaje que notifica al usuario

}

}

}

double vTotal = 0; // Se crea la variable "vTotal" como tipo double con un valor de cero

for (int i=0; i<=indice2-1;i++){

if (nTicket[i] == ticket){

vTotal=vTotal+nPST[i]; // Funcion para sumar todos los importes de los tickets

}

}

System.out.println("TOTAL ---> $"+vTotal); // Mensaje que e notifica al usuario

// el dinero total de la venta realizada

}

public static void grabarInventario(String nombreArchivo){ //Metodo para crear el Archivo donde se registran las Datos

// Paso 1.- Instanciamos un objeto de la clase File

// Al instanciar escribimos como parámetro

// El nombre del archivo para manipularlo

File archivo = new File(nombreArchivo);

// Paso 2.- Si no existe el archivo

if(!archivo.exists()){

try{

// try nos sirve para manejar excepciones. En caso de que algo

// pueda salir mal.

// Creamos un archivo nuevo.

archivo.createNewFile();

} catch (IOException ex){

ex.printStackTrace();

}

}else{

// archivo.delete();

try{

archivo.delete();

archivo.createNewFile();

}catch (IOException ex){

ex.printStackTrace();

}

// System.out.println("Este archivo ya existe. Por favor introduzca un nombre diferente para poder guardar el archivo:"); //Impresion del mensaje de Error

// nombreArchivo = pedirNombre(nombreArchivo);

// crearArchivo(nombreArchivo, arr);

}

// Paso 3.- Escritura en el archivo

try{

// Instanciamos un objeto de la clase PrintWriter

// como parámetros enviamos el la instancia de File y el formato de

// archivo de texto

PrintWriter escribir = new PrintWriter (nombreArchivo, "utf-8");

// Escribimos el contenido del archivo.;

for(int i=0; i<indice; i++){

escribir.print(rC[i]+","+rD[i]+","+rT[i]+","+rP[i]);

escribir.println();

//Si es el primero en la linea lista agrega una separacion en el texto

}

//Cerramos el archivo.

escribir.close();

}

catch (Exception e){

e.printStackTrace();

}

}

public static void grabarNota(String nombreArchivo){ //Metodo para crear el Archivo donde se registran las Datos

//Paso 1.- Instanciamos un objeto de la clase File

//Al instanciar escribimos como parámetro

//El nombre del archivo para manipularlo

Calendar fecha = Calendar.getInstance(); // Funcion que permite tomar los datos

// de la fecha actual del equipo en el cual

// se este ejecutando el programa

int año = fecha.get(Calendar.YEAR); // Funcion que recupera el año

int mes = fecha.get(Calendar.MONTH) + 1; // Funcion que recupera el mes

int dia = fecha.get(Calendar.DAY\_OF\_MONTH); // Funcion que recupera el dia del mes

int hora = fecha.get(Calendar.HOUR\_OF\_DAY); // Funcion que recupera la hora del dia

int minuto = fecha.get(Calendar.MINUTE); // Funcion que recupera el minuto de la hora

int segundo = fecha.get(Calendar.SECOND); // Funcion que recupera el segundo del minuto

String fechas =Integer.toString(año)+"\_"+Integer.toString(mes)+"\_"+Integer.toString(dia);

File archivo = new File(nombreArchivo+fechas+".txt");

//Paso 2.- Si no existe el archivo

if(!archivo.exists()){

try{

// try nos sirve para manejar excepciones. En caso de que algo

// pueda salir mal.

// Creamos un archivo nuevo.

archivo.createNewFile();

} catch (IOException ex){

ex.printStackTrace();

}

}else{

// archivo.delete();

try{

archivo.delete();

archivo.createNewFile();

}catch (IOException ex){

ex.printStackTrace();

}

// System.out.println("Este archivo ya existe. Por favor introduzca un nombre diferente para poder guardar el archivo:"); //Impresion del mensaje de Error

// nombreArchivo = pedirNombre(nombreArchivo);

// crearArchivo(nombreArchivo, arr);

}

// Paso 3.- Escritura en el archivo

try{

// Instanciamos un objeto de la clase PrintWriter

// como parámetros enviamos el la instancia de File y el formato de

// archivo de texto

PrintWriter escribir = new PrintWriter (nombreArchivo+fechas+".txt", "utf-8");

// Escribimos el contenido del archivo.

double vTotal=0;

for(int i=0; i<indice2; i++){

System.out.println("Vendido --->"+nTicket[i]+" "+nC[i]+" "+nD[i]+" "+nT[i]+" $"+nP[i]+" $"+nPST[i]);

escribir.print("Vendido --->"+nTicket[i]+" "+nC[i]+" "+nD[i]+" "+nT[i]+" $"+nP[i]+" $"+nPST[i]);

escribir.println();

vTotal=vTotal+nPST[i]; // Calula el importe total de la venta realizada

// Si es el primero en la linea lista agrega una separacion en el texto

}

escribir.print("TOTAL ---> $" + vTotal);

escribir.println();

System.out.println("TOTAL ---> $" + vTotal);

// Cerramos el archivo.

escribir.close();

}

catch (Exception e){

e.printStackTrace();

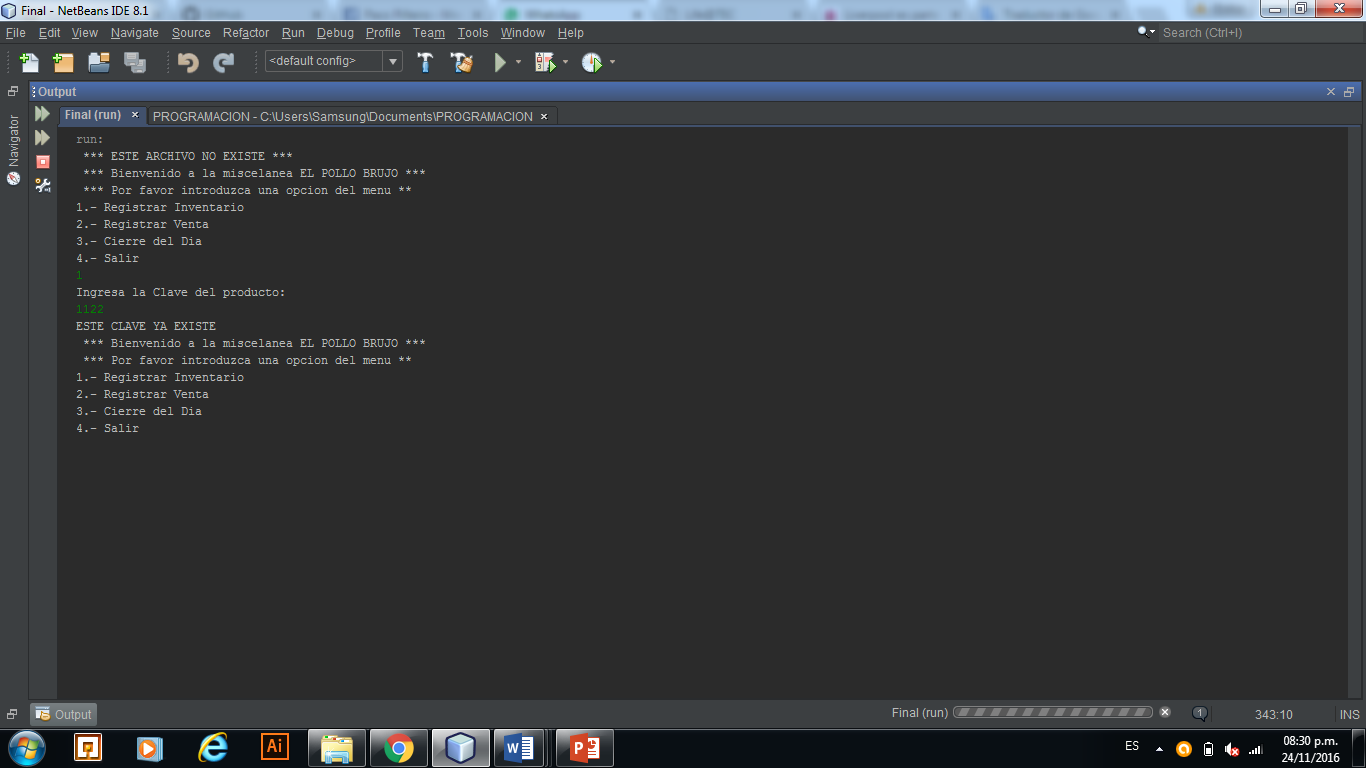
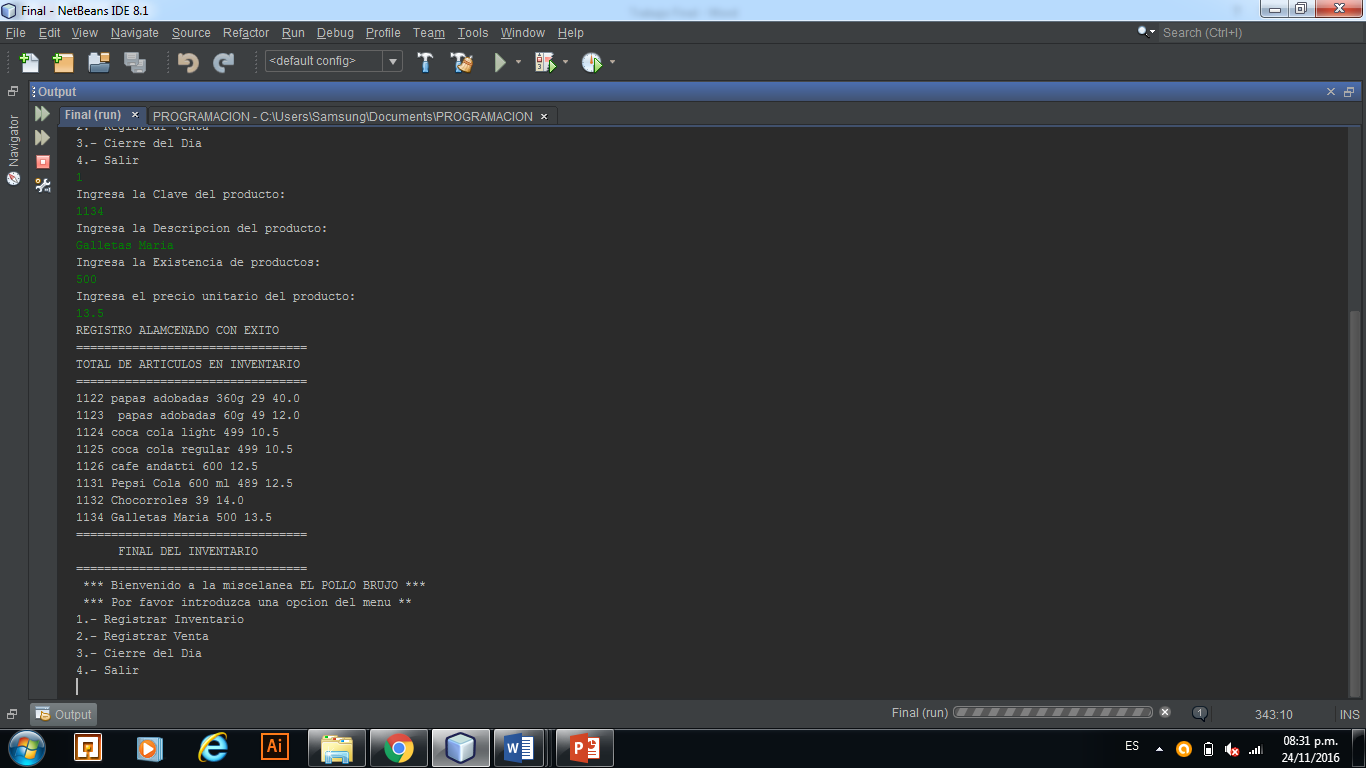
}

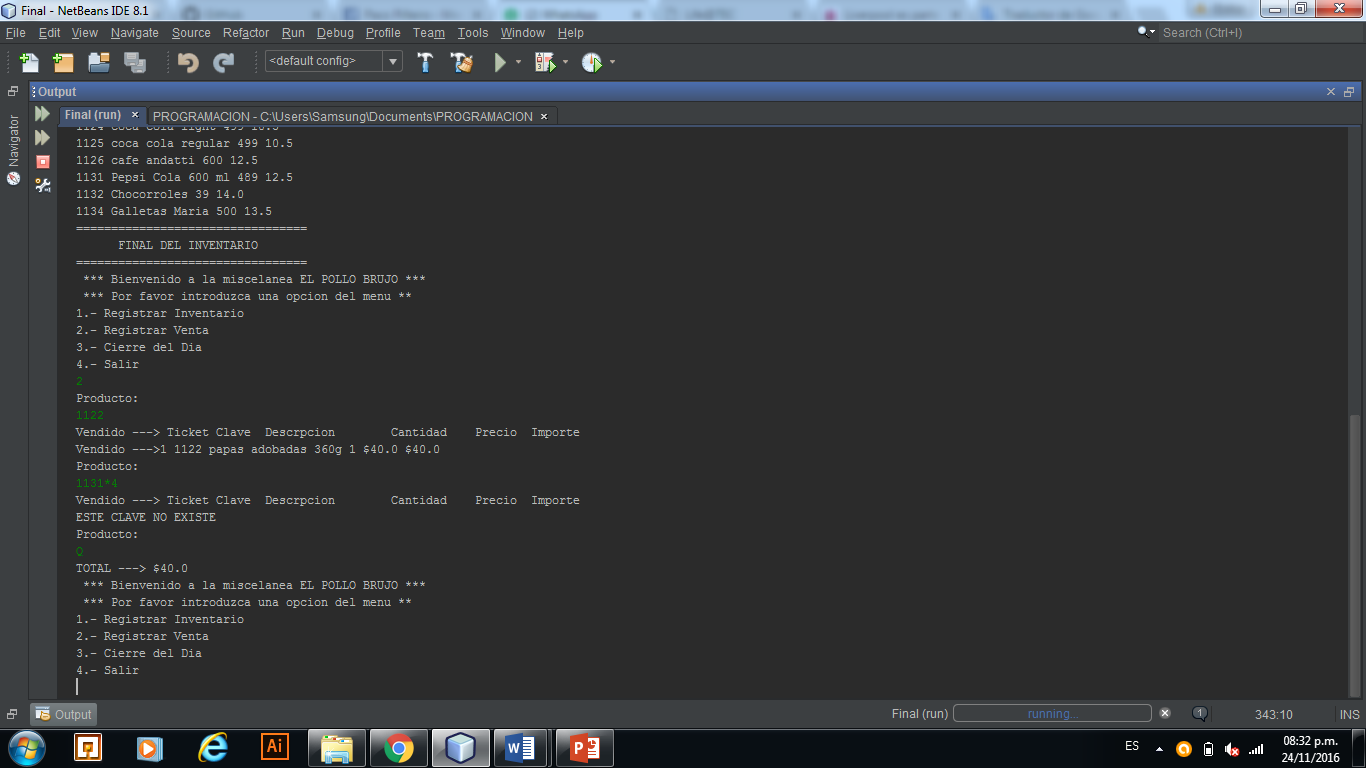
}

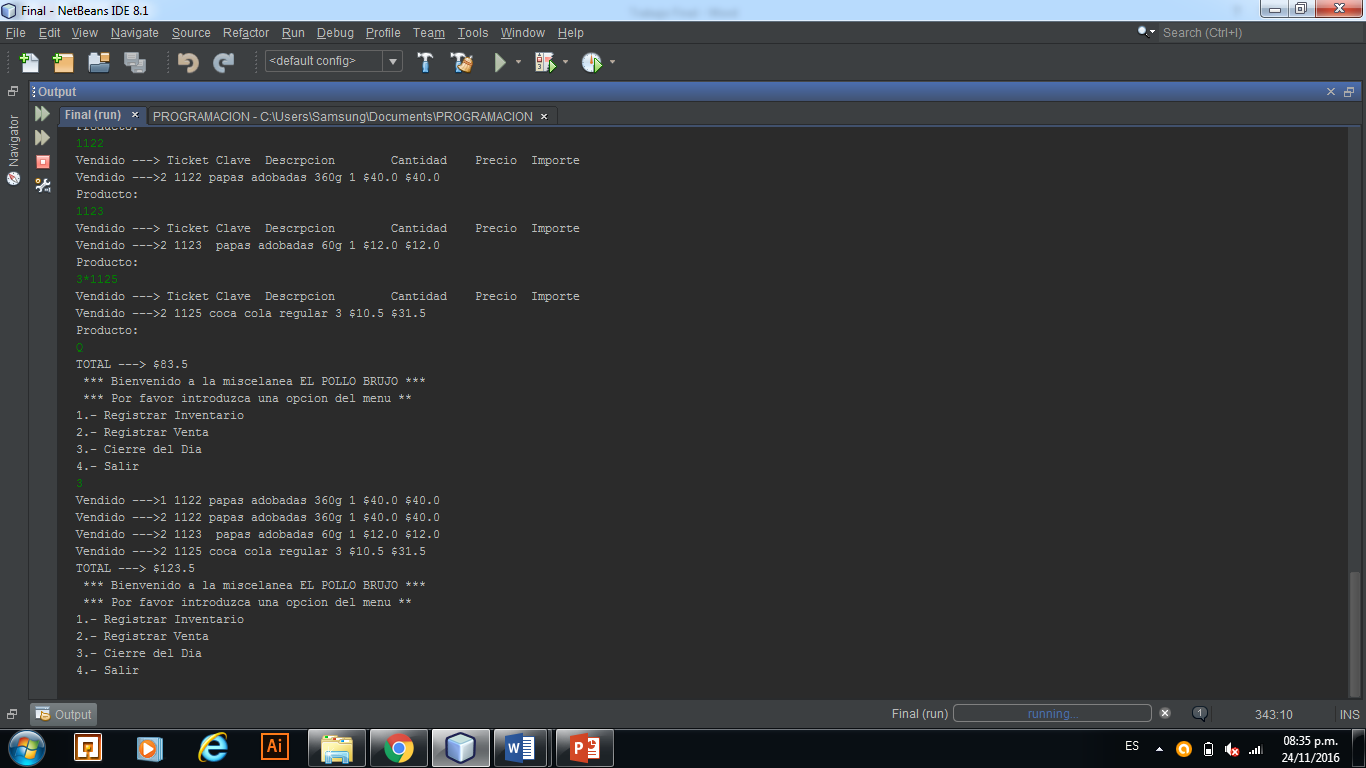
}

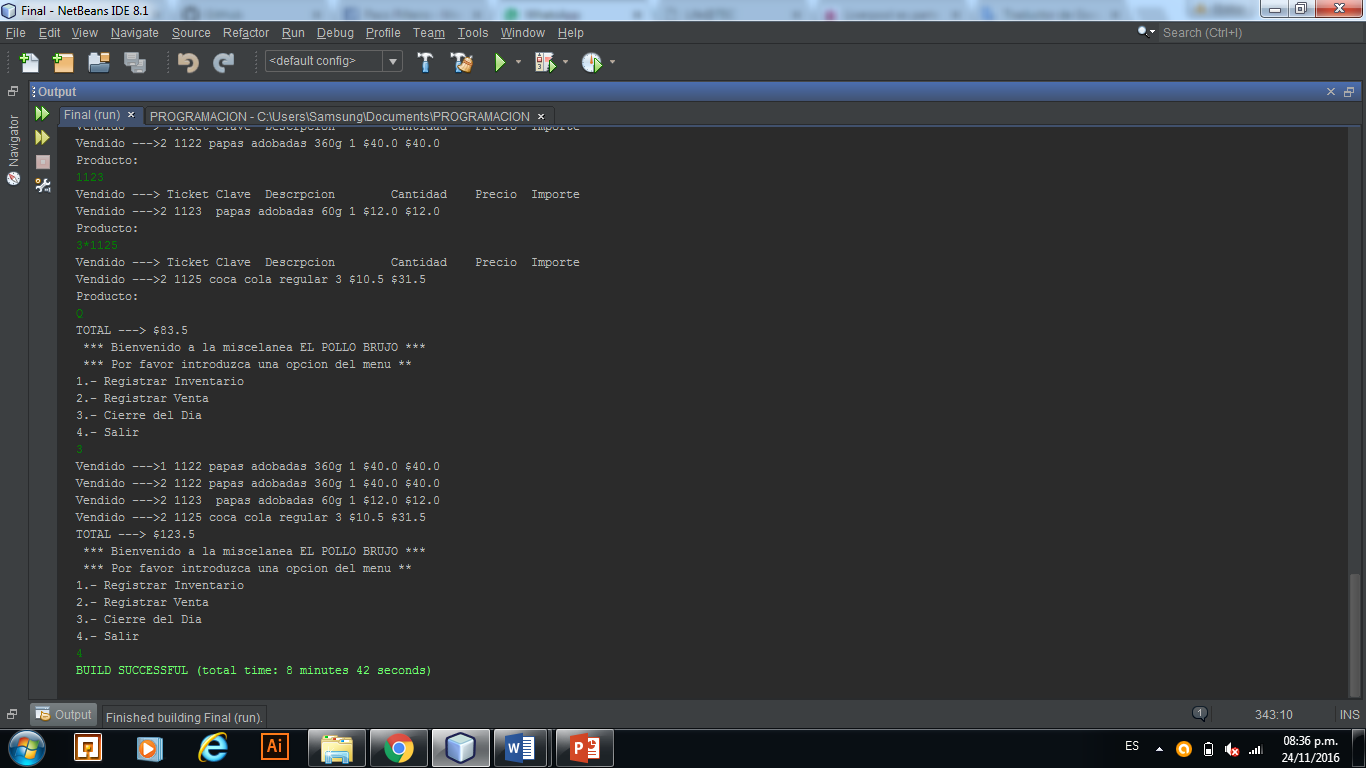
// FIN DEL PROGRAMA

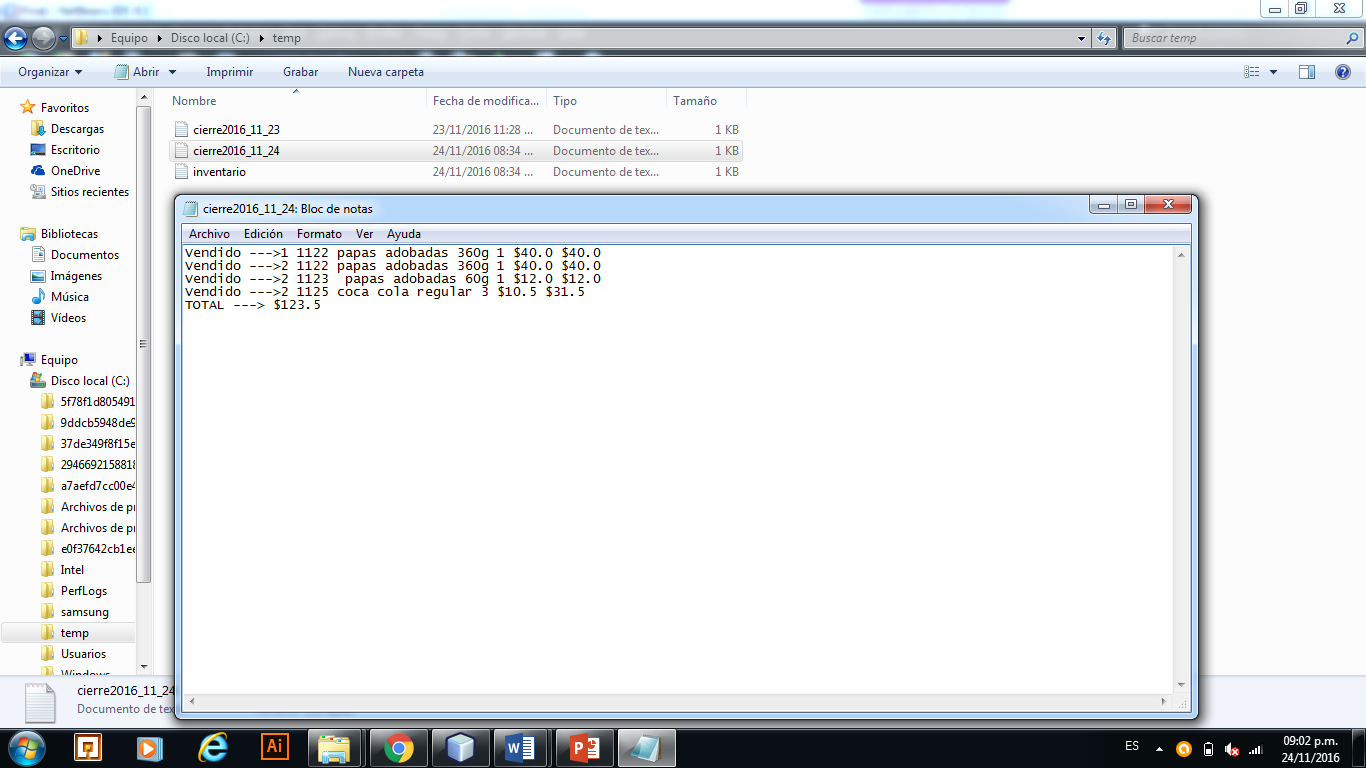
Evidencia del programa en funcionamiento (Registro de inventario, creación de archivos, creación de notas, y registro de ventas):

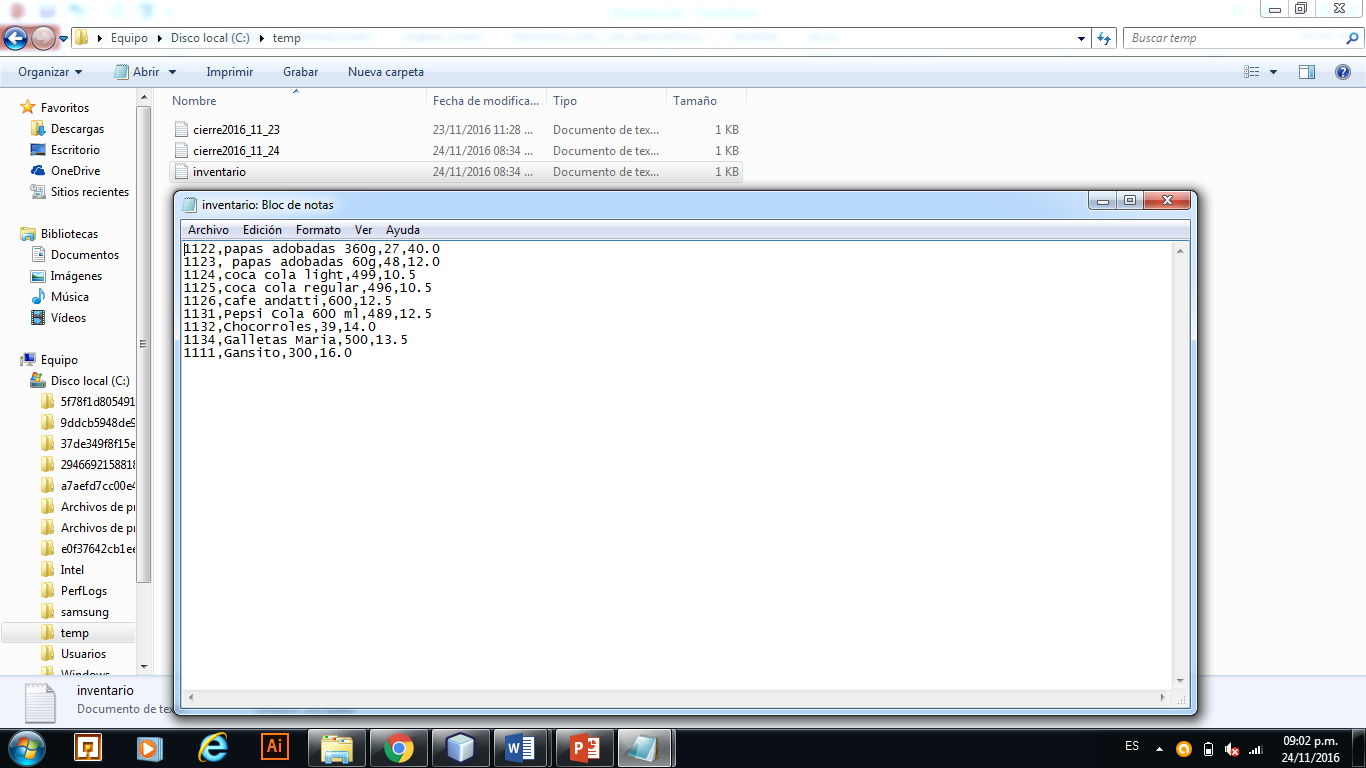












|  |  |  |  |
| --- | --- | --- | --- |
| Carlos René Arteaga Hernández | Participación activa.  Las aportaciones de mi compañero fueron fundamentales para el logro del objetivo. | Nivel de compromiso.  Mi compañero cumplió a tiempo con las actividades asignadas. | Comunicación.  Mi compañero retroalimentó al resto del equipo sobre sus avances, dudas y soluciones. |
| Esteban Azuara Aguirre | 10 | 10 | 10 |
| Alejandro Sánchez Ponce | 10 | 10 | 10 |
| Jorge Ricardo Cabrales Reyes | 10 | 10 | 10 |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Esteban Azuara Aguirre | Participación activa.  Las aportaciones de mi compañero fueron fundamentales para el logro del objetivo. | Nivel de compromiso.  Mi compañero cumplió a tiempo con las actividades asignadas. | Comunicación.  Mi compañero retroalimentó al resto del equipo sobre sus avances, dudas y soluciones. |
| Carlos René Arteaga Hernández | 10 | 10 | 10 |
| Alejandro Sánchez Ponce | 10 | 10 | 10 |
| Jorge Ricardo Cabrales Reyes | 10 | 10 | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| Alejandro Sánchez Ponce | Participación activa.  Las aportaciones de mi compañero fueron fundamentales para el logro del objetivo. | Nivel de compromiso.  Mi compañero cumplió a tiempo con las actividades asignadas. | Comunicación.  Mi compañero retroalimentó al resto del equipo sobre sus avances, dudas y soluciones. |
| Esteban Azuara Aguirre | 10 | 10 | 10 |
| Carlos René Arteaga Hernández | 10 | 10 | 10 |
| Jorge Ricardo Cabrales Reyes | 10 | 10 | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| Jorge Ricardo Cabrales Reyes | Participación activa.  Las aportaciones de mi compañero fueron fundamentales para el logro del objetivo. | Nivel de compromiso.  Mi compañero cumplió a tiempo con las actividades asignadas. | Comunicación.  Mi compañero retroalimentó al resto del equipo sobre sus avances, dudas y soluciones. |
| Esteban Azuara Aguirre | 10 | 10 | 10 |
| Alejandro Sánchez Ponce | 10 | 10 | 10 |
| Carlos René Arteaga Hernández | 10 | 10 | 10 |

**Proyecto Final por equipo.**

La tiendita de la esquina requiere operar con un sistema que le ayude a realizar sus ventas, se necesita llevar un control de las existencias de los productos que vende, y llevar un control de notas de venta para que al cerrar el día puedan contabilizar los ingresos. El sistema deberá contar con el siguiente menú.

Bienvenido a mi tiendita.

Por favor introduzca una opción del menú:

1. Inventario.
2. Venta.
3. Cierre del día.
4. Salir.

**Detalle.**

1. La opción Inventario deberá permitir la captura de los productos, los cuales tendrán los siguientes campos: Cantidad existente, Clave, Descripción, Precio, cada campo deberá definir el tipo de dato correcto. No se debe permitir el registro de productos repetidos.
2. La opción Venta deberá cargar la información de inventario a un arreglo (memoria), para realizar la venta pedirá la clave del producto, buscará esta clave en el arreglo; si la encuentra hará tres cosas, ingresar el producto con los campos cantidad, clave, descripción, unitario e importe(este último es resultado de multiplicar cantidad \* unitario) en un arreglo de nota, mostrarla en pantalla, y descontar la cantidad de artículos del inventario, este proceso se repetirá hasta que el usuario presione la tecla Q. Para introducir la cantidad de artículos el sistema deberá interpretar si se introduce solamente la clave considerará como un solo artículo, pero si se introduce un número seguido de un asterisco y después la clave como n artículos. Ejemplo: 3\*121343 – Esto deberá interpretarse como 3 piezas del artículo con la clave. Al concluir la captura, mostrará en pantalla el total y guardará en un archivo la nota, el nombre de este archivo será la fecha de la nota; si no encuentra la clave, deberá mostrar un mensaje diciendo que el producto no existe.
3. La opción Cierre generará un reporte de las notas creadas, integrando todos los artículos vendidos y acumulando los totales de la nota, deberá mostrarlo en pantalla y crear un archivo con el nombre cierre + fecha.
4. Se deberá presentar el proyecto por GitHub el 24 de noviembre antes de la media noche. También deberán presentar un entregable impreso que incluya el código del programa y las salidas de pantalla con los ejemplos, la impresión de los archivos Inventario, Notas y Cierre. Presentarán el programa en operación después del examen el día 25 de noviembre.

**Rúbrica.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rasgos. | Max. | **Excepcional** | Aceptable | Bueno | Amateur |
| Despliega menú con opciones indicadas, y lo vuelve a desplegar después de ejecutar la opción 1, 2 o 3 sale del sistema al presionar 4 u otra tecla. | 5 puntos. | **5** | 4 | 3 | 1 |
| Al presionar la opción Inventario. | 30 puntos. | **30** | 25 | 10 | 5 |
| 1. Permite la captura de los productos utilizando los campos indicados. | 10 |  |  |  |  |
| 1. Los campos están definidos con los tipos de datos correctos. | 10 |  |  |  |  |
| 1. La captura no permite registros duplicados. | 10 |  |  |  |  |
| Al presionar la opción Venta: | 40 puntos. | **40** | 35 | 20 | 5 |
| 1. Se carga la información del inventario en un arreglo. | 8 |  |  |  |  |
| 1. La captura de la clave respeta la nomenclatura definida | 8 |  |  |  |  |
| 1. Al capturar la clave se almacena el producto en un arreglo con los campos indicados. | 8 |  |  |  |  |
| 1. Si no se encuentra el producto, el programa lo indica y continúa. | 8 |  |  |  |  |
| 1. Al concluir la captura con la tecla Q se muestra el total correcto de la compra y se guarda en un archivo. | 8 |  |  |  |  |
| Al presionar la Opción Cierre:   1. Muestra en pantalla un reporte con las notas creadas, integrando todos los artículos vendidos y acumula los totales de cada nota mostrando un Gran Total del día. | 15 puntos. | **15** | 12 | 8 | 5 |
| El código está documentado, los comentarios explican el código y demuestran la lógica utilizada por el alumno. | 5 puntos. | **5** | 4 | 3 | 1 |
| El código está en la organización Github del equipo antes del “deadline”. Lo presenta impreso, engargolado y con imágenes de las salidas de pantalla. | 5 puntos. | **5** | 4 | 3 | 1 |
| **Total por equipo. (TE)** | **100 puntos.** |  |  |  |  |
| **Evaluación Individual** | **=TE\*CE/10** |  |  |  |  |

**Coevaluación (CE)**.

Cada alumno presentará una tabla evaluando a sus compañeros bajo los rasgos señalados. Evalúa en una escala del 0 al 10, donde **0 es totalmente en desacuerdo** y **10 es totalmente de acuerdo**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mi nombre** | **Participación activa.**  Las aportaciones de mi compañero fueron fundamentales para el logro del objetivo. | **Nivel de compromiso.**  Mi compañero cumplió a tiempo con las actividades asignadas. | **Comunicación.**  Mi compañero retroalimentó al resto del equipo sobre sus avances, dudas y soluciones. |
| Integrante 1 |  |  |  |
| Integrante 2 |  |  |  |
| Integrante 3 |  |  |  |