José Emiliano Pérez Garduño

2CV1 Programación Orientada a Objetos

Práctica 4

1.-Modificar el ejemplo 4 para que acepte más de un registro:

namespace RegistroAlumno

{

class Datos

{

struct Alumno

{

public string carnet;

public string Nombre;

public void Mostrar(Alumno X)

{

Console.WriteLine("\tNombre: " + Nombre);

Console.WriteLine("\tCarnet: " + carnet);

}

public void Registrar(string c, string n)

{

carnet = c;

Nombre = n;

}

}

static void Main(string[] args)

{

int op, O;

Console.Title = "Utilizando estructuras y procedimientos";

Console.WriteLine("Escribir cuantos alumnos seran introducidos:");

int tam = int.Parse(Console.ReadLine());

Alumno [] alumnos = new Alumno[tam];

do

{

Console.Clear();

Console.WriteLine("MENU");

Console.WriteLine("1.-Ingresar.");

Console.WriteLine("2.-Mostrar.");

Console.WriteLine("3.-Ingresar opción.");

op = int.Parse(Console.ReadLine());

switch (op)

{

case 1:

for (int i = 0; i < tam; i++)

{

Console.Clear();

Console.WriteLine("\tNombre: ");

string n = Console.ReadLine();

Console.WriteLine("\tCArnet: ");

string c = Console.ReadLine();

alumnos[i].Registrar(c, n);

}

break;

case 2:

Console.Clear();

for (int i = 0; i < tam; i++)

{

Console.WriteLine("Nombre:" + alumnos[i].Nombre);

Console.WriteLine("Carnet:" + alumnos[i].carnet);

}

break;

}

Console.WriteLine("\tSi desea continuar presione 1, sino presione 2.");

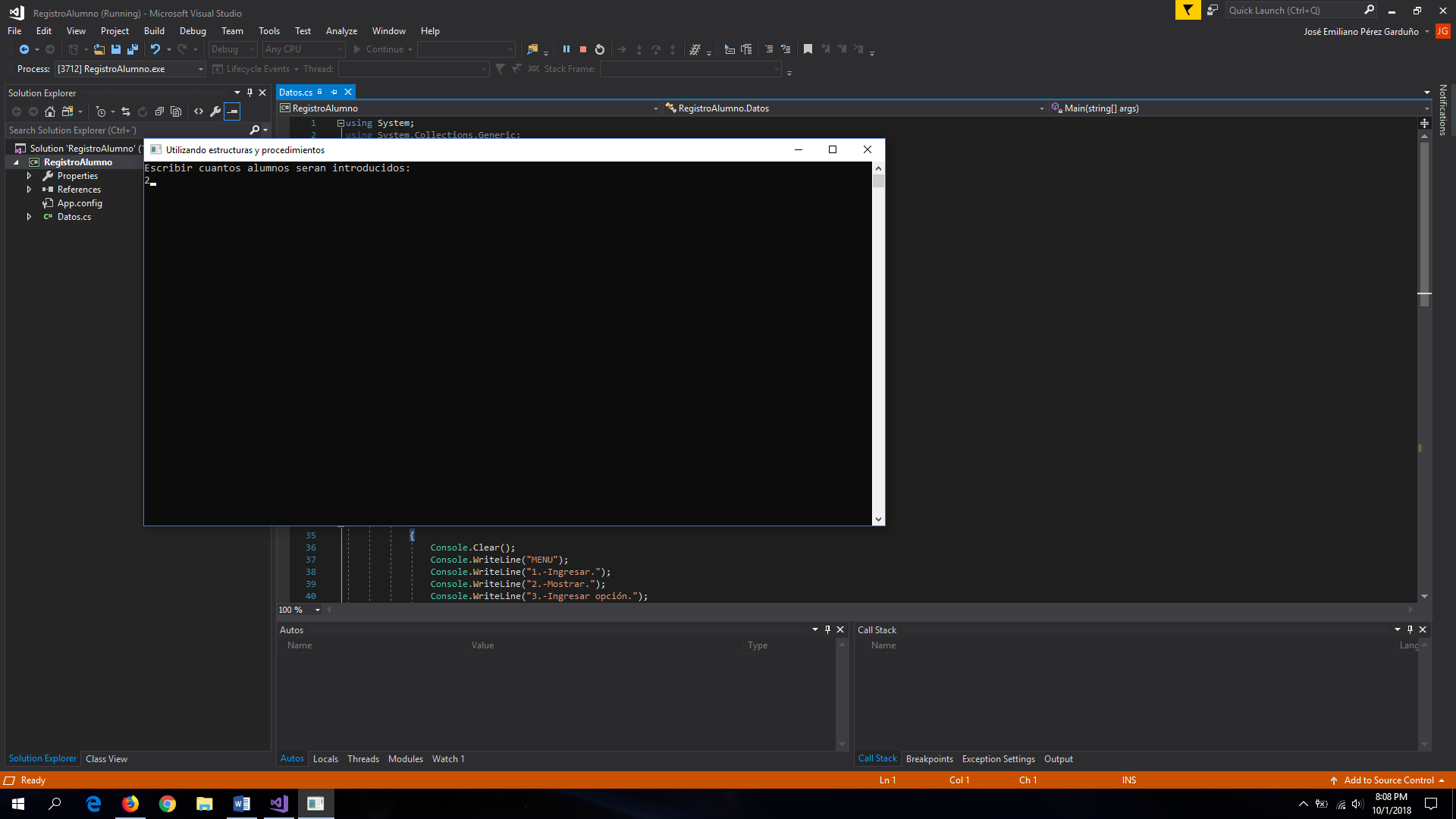
O = int.Parse(Console.ReadLine());

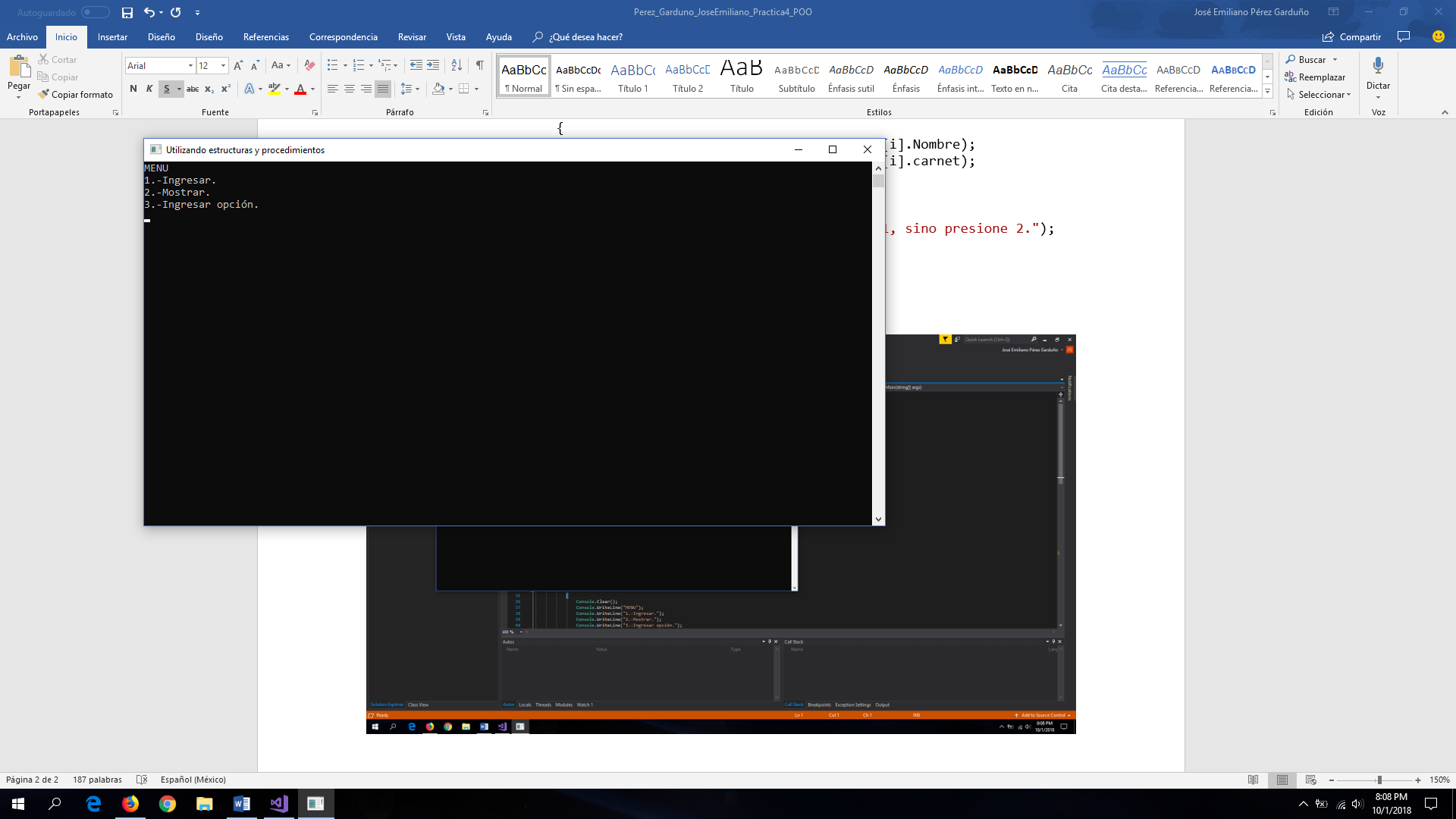
} while (O == 1);

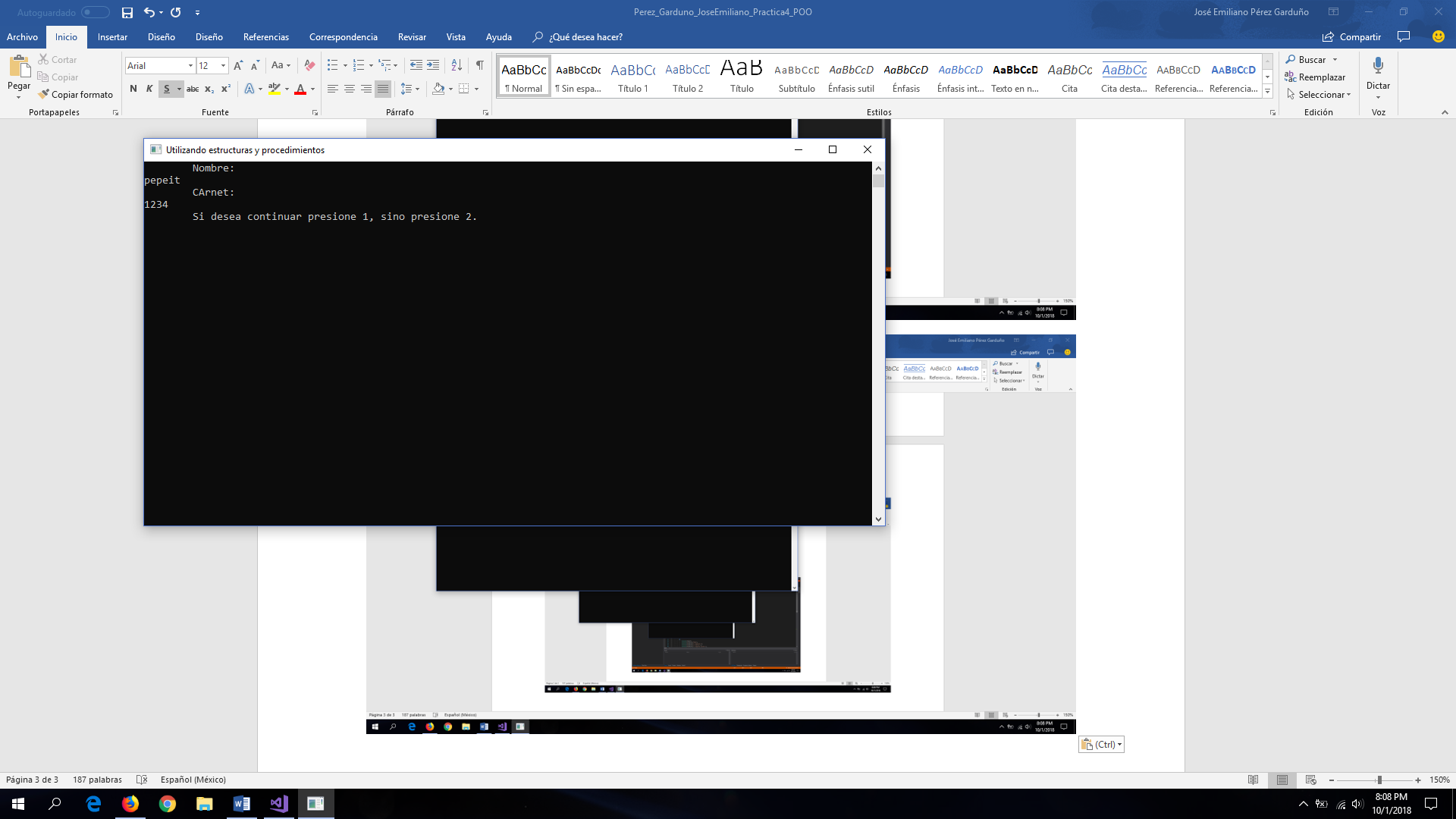
}

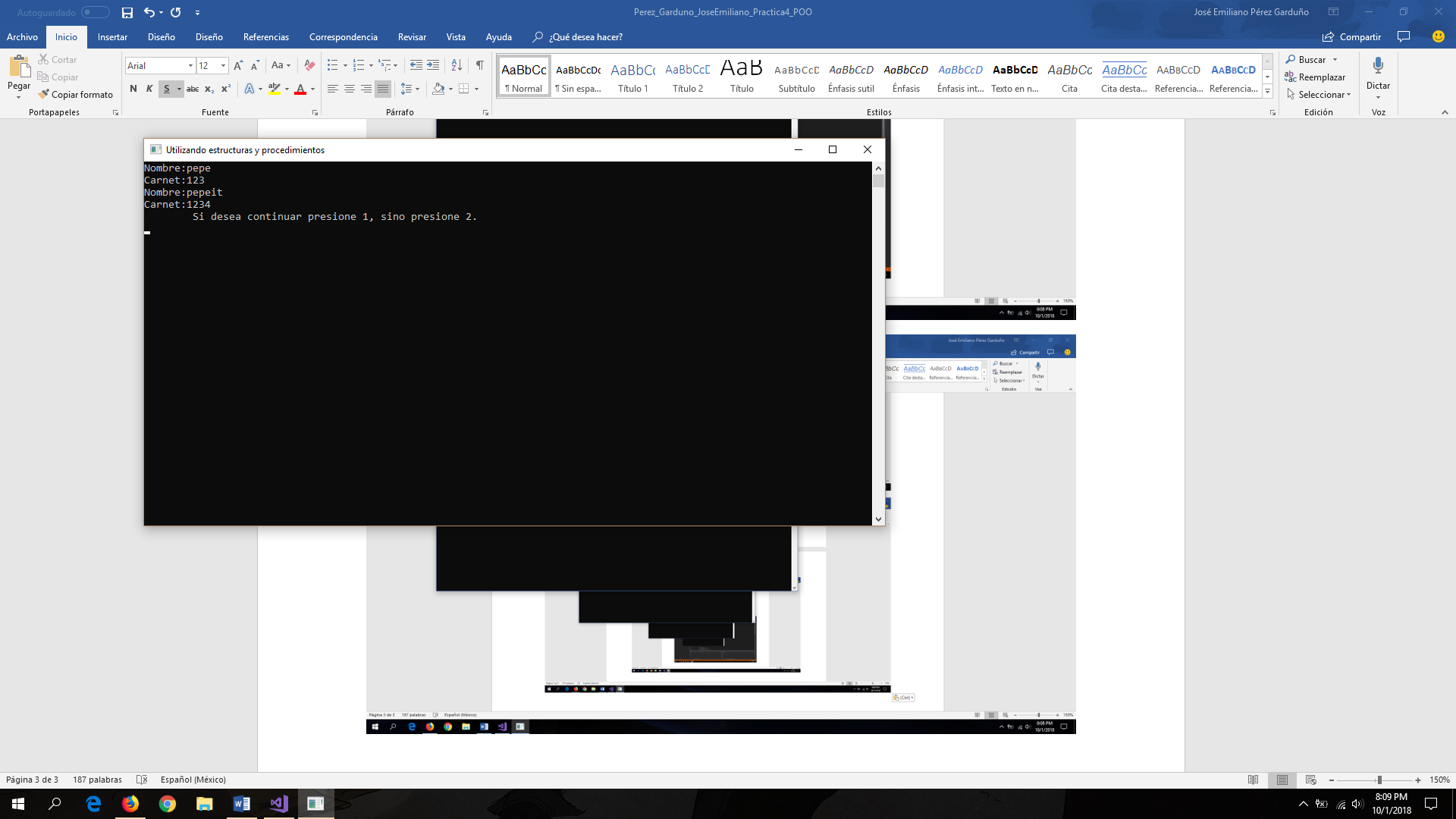
}

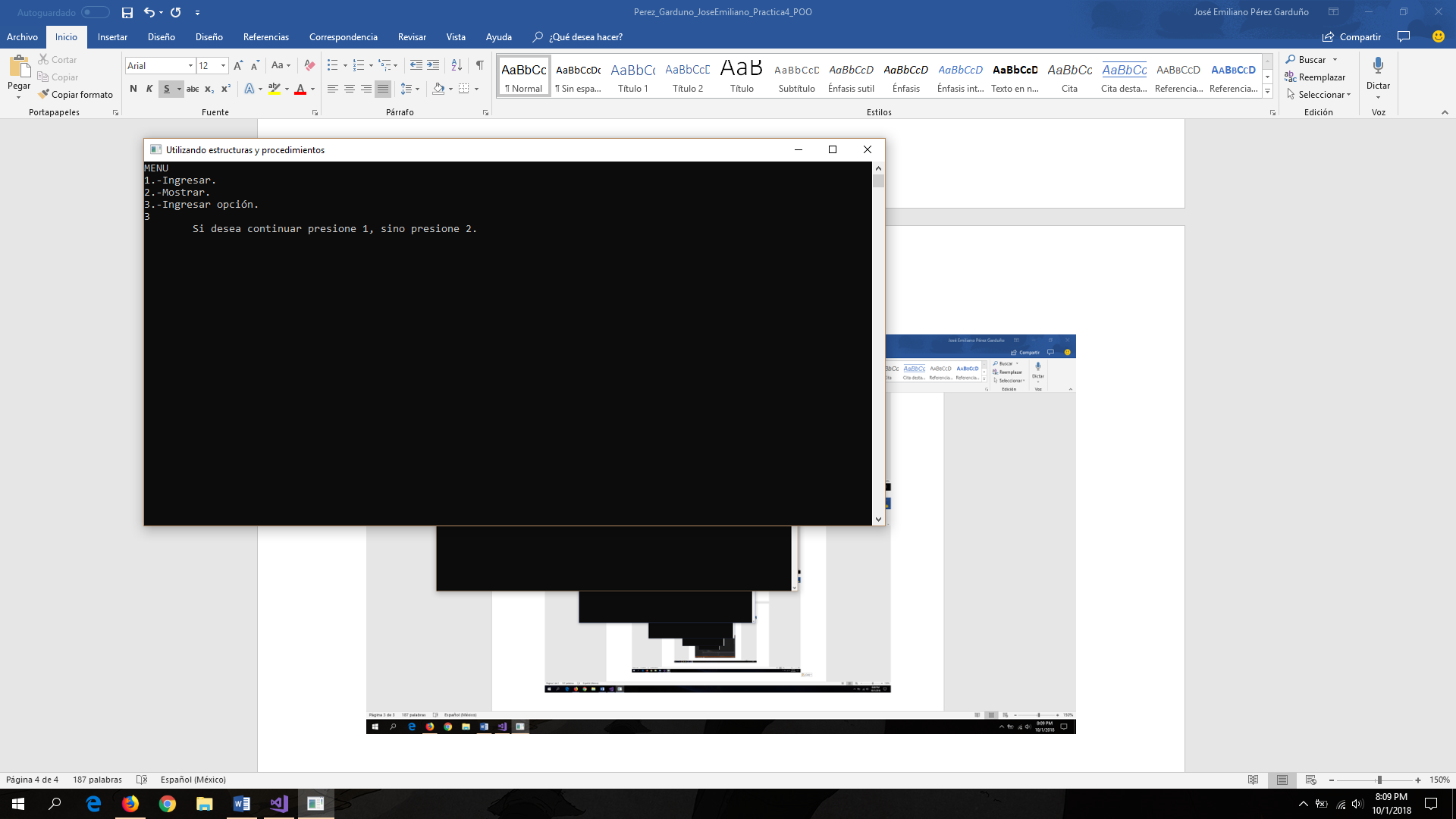
}











2.- Desarrollar un programa que simule una agenda:

namespace Agenda

{

class Program

{

struct Agenda

{

public string Nombre;

public string DIU;

public string Direccion;

public string Telefono;

public string email;

public string oficio;

public void Registrar(string n, string d, string dir, string tel, string mail, string of)

{

Nombre = n;

DIU = d;

Direccion = dir;

Telefono = tel;

email = mail;

oficio = of;

}

public void Mostrar()

{

Console.WriteLine("Nombre: " + Nombre);

Console.WriteLine("DIU: " + DIU);

Console.WriteLine("Dirección: " + Direccion);

Console.WriteLine("Telefono: " + Telefono);

Console.WriteLine("Email: " + email);

Console.WriteLine("Oficio: " + oficio);

}

}

static void Main(string[] args)

{

int Op = 1, Opcion, TAM;

Console.Title = "Agenda Telefónica";

Console.WriteLine("Ingresar el número de datos que serán registrados:");

TAM = int.Parse(Console.ReadLine());

Agenda[] registros = new Agenda[TAM];

do

{

Console.Clear();

Console.WriteLine("1.-Registrar");

Console.WriteLine("2.-Mostrar");

Console.WriteLine("3.-Buscar");

Console.WriteLine("4.-Salir");

Console.WriteLine("Ingresar Opción:");

Opcion = int.Parse(Console.ReadLine());

switch (Opcion)

{

case 1:

for (int i = 0; i < TAM; i++)

{

Console.Clear();

Console.WriteLine("Nombre:");

string n = Console.ReadLine();

Console.WriteLine("DIU (debe ser unico):");

string d = Console.ReadLine();

Console.WriteLine("Dirección:");

string dir = Console.ReadLine();

Console.WriteLine("Telefono:");

string tel = Console.ReadLine();

Console.WriteLine("Email:");

string mail = Console.ReadLine();

Console.WriteLine("Oficio o Trabajo:");

string of = Console.ReadLine();

registros[i].Registrar(n, d, dir, tel, mail, of);

}

break;

case 2:

Console.Clear();

for (int i = 0; i < TAM; i++)

{

Console.WriteLine("\nEl contacto {0} es", i + 1);

registros[i].Mostrar();

}

break;

case 3:

string busqueda;

int j = 0;

Console.Clear();

Console.WriteLine("Escribrir el DIU para buscar");

busqueda = Console.ReadLine();

while (j < TAM)

{

if (busqueda == registros[j].DIU)

{

Console.WriteLine("Los datos del contacto con DIU {0} son:", busqueda);

registros[j].Mostrar();

}

j++;

}

break;

case 4:

Environment.Exit(-1);

break;

default:

Console.WriteLine("Opción no valida");

Console.ReadKey();

Environment.Exit(-1);

break;

}

Console.WriteLine("Para continuar escriba 1.");

Op = int.Parse(Console.ReadLine());

} while (Op == 1);

}

}

}

