

Laboratorio de Programación Orientada a Objetos

Práctica 4: Clases y objetos



Equipo No: 4.

Integrantes:

Arriaga Mejía José Carlos

Fragoso Islas Ana Cecilia

Medina Perabeles Rodrigo

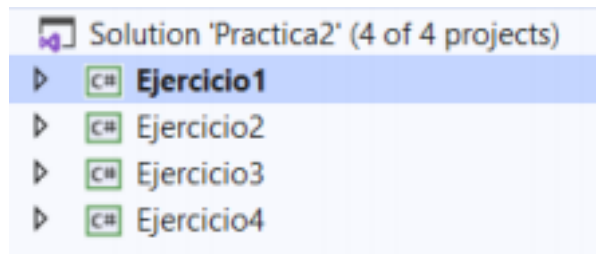
Pérez Duarte Brenda Elizabeth

Fecha de realización:

22/10/2020

Práctica 4. Clases y Objetos

Realizar los siguientes programas en Visual Studio, deberán de generar un nuevo proyecto llamado Practica4EqYY (dónde YY es el número de su equipo) y dentro de la solución se creará un proyecto con el nombre del ejercicio (ver imagen ejemplo).



- 1) Realizar las clases para los siguientes prismas: cubo, pirámide, cilindro y esfera. En cada uno deberá de pedir si se quiere calcular el área, el perímetro y el volumen (determinar los atributos y métodos necesarios).
Elaborar un menú para seleccionar el prisma y la opción para seleccionar el área, perímetro o el volumen.

```

1  /*
2  Arriaga Mejia Jose Carlos
3  Fragoso Islas Ana Cecilia.
4  Medina Perabeles Rodrigo.
5  Perez Duarte Brenda Elizabeth.
6
7  Descripción:
8  Realizar las clases para los siguientes prismas: cubo, pirámide, cilindro y esfera.
9  En cada uno deberá de pedir si se quiere calcular el área, el perímetro y el volumen
10 (determinar los atributos y métodos necesarios).
11 Elaborar un menú para seleccionar el prisma y la opción para seleccionar el área, perímetro o el volumen.
12 */
13 using System;
14
15 namespace Ejercicio1
16 {
17     0 referencias
18     class Principal
19     {
20         0 referencias
21         static void Main(string[] args)
22         {
23             try
24             {
25                 int op, opi, opi2;
26                 Cubo c = new Cubo();
27                 Esfera e = new Esfera();
28                 Piramide p = new Piramide();
29                 Cilindro ci = new Cilindro();
30                 do
31                 {
32                     Console.Clear();
33                     Console.WriteLine("Este programa calculara el area, volumen y perimetro (Si es que se pueda) " +
34                         "de distintas figuras geometricas");
35                     Console.Write("Menu\n" +
36                         "1.-Cubo\n" +
37                         "2.-Esfera\n" +
38                         "3.-Cilindro\n" +
39                         "4.-Piramide\n" +
40                         "5.-Salir\n" +
41                         "Elija una figura: ");
42                     op = Convert.ToInt32(Console.ReadLine());

```

```

41 switch (op)
42 {
43     case 1:
44         Console.Clear();
45         Console.WriteLine("Eligio al Cubo, por favor ingrese el tamaño de su lado: ");
46         c.SetLado(Convert.ToSingle(Console.ReadLine()));
47         do
48         {
49             Console.Clear();
50             Console.WriteLine("Cubo\n" +
51                 "Menu interno\n" +
52                 "1.-Perimetro\n" +
53                 "2.-Area\n" +
54                 "3.-Volumen\n" +
55                 "4.-Salir\n" +
56                 "Elija una opcion: ");
57             opi = Convert.ToInt32(Console.ReadLine());
58             switch (opi)
59             {
60                 case 1:
61                     c.SetPerimetro();
62                     Console.WriteLine("El perimetro del Cubo es de: " + c.GetPerimetro() + " unidades");
63                     break;
64                 case 2:
65                     c.SetArea();
66                     Console.WriteLine("El area del Cubo es: " + c.GetArea() + " unidades cuadradas");
67                     break;
68                 case 3:
69                     c.SetVolumen();
70                     Console.WriteLine("El volumen del Cubo es: " + c.Getvolumen() + " unidades cubicas");
71                     break;
72                 case 4:
73                     Console.WriteLine("Saliendo del menu del Cubo.");
74                     break;
75                 default:
76                     Console.WriteLine("Opcion Invalida.");
77                     break;
78             }
79             Console.WriteLine("Presione cualquier tecla para continuar.");
80             Console.ReadLine();
81         } while (opi != 4);

```

```

82 static void Main(string[] arg
83 {
84     Console.Clear();
85     Console.WriteLine("Eligio a la Esfera, por favor ingrese el tamaño de su Radio: ");
86     e.SetRadio(Convert.ToSingle(Console.ReadLine()));
87     do
88     {
89         Console.Clear();
90         Console.WriteLine("Esfera\n" +
91             "Menu interno\n" +
92             "1.-Area\n" +
93             "2.-Volumen\n" +
94             "3.-Salir\n" +
95             "Elija una opcion: ");
96         opi = Convert.ToInt32(Console.ReadLine());
97         switch (opi)
98         {
99             case 1:
100                 e.SetArea();
101                 Console.WriteLine("El area de la Esfera es: " + e.GetArea() + " unidades cuadradas");
102                 break;
103             case 2:
104                 e.SetVolumen();
105                 Console.WriteLine("El volumen de la Esfera es: " + e.Getvolumen() + " unidades cubicas");
106                 break;
107             case 3:
108                 Console.WriteLine("Saliendo del menu de la Esfera.");
109                 break;
110             default:
111                 Console.WriteLine("Opcion Invalida.");
112                 break;
113             }
114             Console.WriteLine("Presione cualquier tecla para continuar.");
115             Console.ReadLine();
116         } while (opi != 3);
117     } while (true);
118     case 3:
119         Console.Clear();
120         Console.WriteLine("Eligio al Cilindro, por favor ingrese el tamaño del Radio de la base: ");
121         ci.SetRadio(Convert.ToSingle(Console.ReadLine()));
122         Console.WriteLine("Ingrese la altura del Cilindro: ");
123         ci.SetAltura(Convert.ToSingle(Console.ReadLine()));

```

```

124 do
125 {
126     Console.Clear();
127     Console.WriteLine("Cilindro\n" +
128         "Menu interno\n" +
129         "1.-Area\n" +
130         "2.-Volumen\n" +
131         "3.-Salir\n" +
132         "Elija una opcion: ");
133     opi = Convert.ToInt32(Console.ReadLine());
134     switch (opi)
135     {
136         case 1:
137             ci.SetArea();
138             Console.WriteLine("El area de la Cilindro es: " + ci.GetArea() + " unidades cuadradas");
139             break;
140         case 2:
141             ci.SetVolumen();
142             Console.WriteLine("El volumen de la Cilindro es: " + ci.Getvolumen() + " unidades cubicas");
143             break;
144         case 3:
145             Console.WriteLine("Saliendo del menu de la Cilindro.");
146             break;
147         default:
148             Console.WriteLine("Opcion Invalida.");
149             break;
150     }
151     Console.WriteLine("Presione cualquier tecla para continuar.");
152     Console.ReadLine();
153 } while (opi != 3);
154 break;
155 case 4:
156 do
157 {
158     Console.Clear();
159     Console.WriteLine("Piramide\n" +
160         "Menu interno\n" +
161         "1.-Triangulo\n" +
162         "2.-Cuadrilatero\n" +
163         "3.-Poligono de 5 lados o mas lados\n" +
164         "4.-Salir\n" +
165         "Elija la figura que tiene como base de la Piramide: ");
166     opi = Convert.ToInt32(Console.ReadLine());
167     switch (opi)
168     {
169         case 1:
170             Console.Clear();
171             Console.WriteLine("Piramide Triangular (Tetraedro)\n" +
172                 "Ingrese el tamaño del lado: ");
173             p.SetLado(Convert.ToSingle(Console.ReadLine()));
174             do
175             {
176                 Console.Clear();
177                 Console.WriteLine("Tetraedro");
178                 Console.WriteLine("Menu interno 2\n" +
179                     "1.-Area\n" +
180                     "2.-Volumen\n" +
181                     "3.-Perimetro\n" +
182                     "4.-Salir\n" +
183                     "Elija una opcion: ");
184                 opi2 = Convert.ToInt32(Console.ReadLine());
185                 switch (opi2)
186                 {
187                     case 1:
188                         p.SetAT();
189                         Console.WriteLine("El area de la piramide es: " + p.GetArea() + " unidades cuadradas");
190                         break;
191                     case 2:
192                         p.SetVT();
193                         Console.WriteLine("El volumen de la piramide es: " + p.GetVolumen() + " unidades cubicas");
194                         break;
195                     case 3:
196                         p.SetPT();
197                         Console.WriteLine("El perimetro de la piramide es: " + p.GetP() + " unidades");
198                         break;
199                     case 4:
200                         Console.WriteLine("Saliendo del menu del Tetraedro.");
201                         break;
202                     default:
203                         Console.WriteLine("Opcion Invalida.");
204                         break;
205                 }
206             }
207             Console.WriteLine("Presione cualquier tecla para continuar.");
208             Console.ReadLine();

```

```

208         } while (opi2 != 4);
209         break;
210     case 2:
211         Console.Clear();
212         Console.WriteLine("Piramide con un cuadrilatero de base\n" +
213             "Ingrese el tamaño del lado menor de la base: ");
214         p.SetLado(Convert.ToSingle(Console.ReadLine()));
215         Console.WriteLine("Ingrese el valor del lado mayor de la base: ");
216         p.SetLado2(Convert.ToSingle(Console.ReadLine()));
217         Console.WriteLine("Ingrese la altura de la piramide: ");
218         p.SetAltura(Convert.ToSingle(Console.ReadLine()));
219         do
220         {
221             Console.Clear();
222             Console.WriteLine("Piramide con un cuadrilatero de base");
223             Console.WriteLine("Menu interno 2\n" +
224                 "1.-Area\n" +
225                 "2.-Volumen\n" +
226                 "3.-Perimetro\n" +
227                 "4.-Salir\n" +
228                 "Elija una opcion: ");
229             opi2 = Convert.ToInt32(Console.ReadLine());
230             switch (opi2)
231             {
232                 case 1:
233                     p.SetAC();
234                     Console.WriteLine("El area de la piramide es: " + p.GetArea() + " unidades cuadradas");
235                     break;
236                 case 2:
237                     p.SetVC();
238                     Console.WriteLine("El volumen de la piramide es: " + p.GetVolumen() + " unidades cubicas");
239                     break;
240                 case 3:
241                     p.SetPC();
242                     Console.WriteLine("El perimetro de la piramide es: " + p.GetP() + " unidades");
243                     break;
244                 case 4:
245                     Console.WriteLine("Saliendo del menu de la Piramide con un cuadrilatero de base.");
246                     break;
247                 default:
248                     Console.WriteLine("Opcion Invalida.");
249                     break;
250             }
251             Console.WriteLine("Presione cualquier tecla para continuar.");
252             Console.ReadLine();
253         } while (opi2 != 4);
254         break;
255     case 3:
256         Console.Clear();
257         Console.WriteLine("Piramide con un poligono de 5 o mas lados de base\n" +
258             "Ingrese el tamaño de una lado de la base: ");
259         p.SetLado(Convert.ToSingle(Console.ReadLine()));
260         Console.WriteLine("Ingrese el numeros de lados que tiene la base: ");
261         p.SetNumLados(Convert.ToInt32(Console.ReadLine()));
262         Console.WriteLine("Ingrese la altura de la piramide: ");
263         p.SetAltura(Convert.ToSingle(Console.ReadLine()));
264         do
265         {
266             Console.Clear();
267             Console.WriteLine("Piramide con un poligono de " + p.GetNumLados() + " lados de base");
268             Console.WriteLine("Menu interno 2\n" +
269                 "1.-Area\n" +
270                 "2.-Volumen\n" +
271                 "3.-Perimetro\n" +
272                 "4.-Salir\n" +
273                 "Elija una opcion: ");
274             opi2 = Convert.ToInt32(Console.ReadLine());
275             switch (opi2)
276             {
277                 case 1:
278                     p.SetAP();
279                     Console.WriteLine("El area de la piramide es: " + p.GetArea() + " unidades cuadradas");
280                     break;
281                 case 2:
282                     p.SetVP();
283                     Console.WriteLine("El volumen de la piramide es: " + p.GetVolumen() + " unidades cubicas");
284                     break;
285                 case 3:
286                     p.SetPP();
287                     Console.WriteLine("El perimetro de la piramide es: " + p.GetP() + " unidades");
288                     break;
289                 case 4:
290                     Console.WriteLine("Saliendo del menu de la Piramide con un poligono de 5 " +
291                         "o mas lados de base.");

```

```

292         break;
293     default:
294         Console.WriteLine("Opcion Invalida.");
295         break;
296     }
297
298     Console.Write("Presione cualquier tecla para continuar.");
299     Console.ReadLine();
300
301     } while (opi2 != 4);
302     break;
303
304     case 4:
305         Console.WriteLine("Saliendo del menu de las Piramides.");
306         Console.Write("Presione cualquier tecla para continuar.");
307         Console.ReadLine();
308         break;
309
310     default:
311         Console.WriteLine("Opcion Invalida.");
312         Console.Write("Presione cualquier tecla para continuar.");
313         Console.ReadLine();
314         break;
315     }
316     } while (opi!=4);
317     break;
318
319     case 5:
320         Console.WriteLine("Saliendo del programa.");
321         Console.Write("Presione cualquier tecla para continuar.");
322         Console.ReadLine();
323         break;
324
325     default:
326         Console.WriteLine("Opcion Invalida.");
327         Console.Write("Presione cualquier tecla para continuar.");
328         Console.ReadLine();
329         break;
330     }
331     } while (op != 5);
332 }
333 catch (FormatException)
334 {
335     Console.WriteLine("Error de formato de ingreso, por favor reinicie el programa.");
336     Console.Read();
337 }

```

```

1  using System;
2
3  namespace Ejercicio1
4  {
5      3 referencias
6      class Cilindro
7      {
8
9          1 referencia
10         public Cilindro()
11         {
12             //Constructor por defecto
13         }
14
15         1 referencia
16         public void SetRadio(float radio)
17         {
18             if (radio < 0)
19             {
20                 radio = -1 * radio;
21                 this.radio = radio;
22                 Console.WriteLine("No puede haber radios negativas, por lo tanto la convertimos en positivo.");
23                 Console.Write("Presione cualquier tecla para continuar.");
24                 Console.ReadLine();
25             }
26             else
27             {
28                 this.radio = radio;
29             }
30         }
31
32         1 referencia
33         public void SetAltura(float altura)
34         {
35             if (altura < 0)
36             {
37                 altura = -1 * altura;
38                 this.altura = altura;
39                 Console.WriteLine("No puede haber alturas negativas, por lo tanto la convertimos en positivo.");
40                 Console.Write("Presione cualquier tecla para continuar.");
41                 Console.ReadLine();
42             }
43             else
44             {
45                 this.altura = altura;
46             }
47         }
48
49         1 referencia
50         public void SetArea()
51         {
52             this.area = 2 * Convert.ToSingle((System.Math.PI)) * radio * (altura + radio);
53         }
54
55         1 referencia
56         public float GetArea()
57         {
58             return this.area;
59         }
60
61         1 referencia
62         public void SetVolumen()
63         {
64             this.volumen = Convert.ToSingle((System.Math.PI)) * radio * radio * altura;
65         }
66
67         1 referencia
68         public float Getvolumen()
69         {
70             return this.volumen;
71         }
72     }
73 }

```

```

1  using System;
2
3  namespace Ejercicio1
4  {
5      3 referencias
6      class Piramide
7      {
8          private float altura, area, volumen, lado, lado2, ATP, ATC, perimetro;
9          private int nl;
10         1 referencia
11         public Piramide()
12         {
13             //constructor por defecto
14         }
15
16         2 referencias
17         public void SetAltura(float altura)
18         {
19
20             if (altura < 0)
21             {
22                 altura = -1 * altura;
23                 this.altura = altura;
24                 Console.WriteLine("No puede haber alturas negativas, por lo tanto la convertimos en positivo.");
25                 Console.Write("Presione cualquier tecla para continuar.");
26                 Console.ReadLine();
27             }
28             else
29             {
30                 this.altura = altura;
31             }
32         }
33
34         3 referencias
35         public void SetLado(float lado)
36         {
37             if (lado < 0)
38             {
39                 lado = -1 * lado;
40                 this.lado = lado;
41                 Console.WriteLine("No puede haber lados negativas, por lo tanto la convertimos en positivo.");
42                 Console.Write("Presione cualquier tecla para continuar.");
43                 Console.ReadLine();
44             }
45             else
46             {
47                 this.lado = lado;
48             }
49         }
50
51         1 referencia
52         public void SetLado2(float lado2)
53         {
54             if (lado2 < 0)
55             {
56                 lado2 = -1 * lado2;
57                 this.lado2 = lado2;
58                 Console.WriteLine("No puede haber lados negativas, por lo tanto la convertimos en positivo.");
59                 Console.Write("Presione cualquier tecla para continuar.");
60                 Console.ReadLine();
61             }
62             else
63             {
64                 this.lado2 = lado2;
65             }
66         }
67
68         1 referencia
69         public void SetNumLados(int nl)
70         {
71             if (nl < 5)
72             {
73                 this.nl = 5;
74                 Console.WriteLine("El numero minimo de lados es 5, así que pusimos como el numero de lados.");
75                 Console.Write("Presione cualquier tecla para continuar.");
76                 Console.ReadLine();
77             }
78             else
79             {
80                 this.nl = nl;
81             }
82         }
83
84         1 referencia
85         public void SetAT()
86         {
87             this.area = lado * lado * Convert.ToSingle((System.Math.Sqrt(3)));
88             //area=lado^2*raiz(3)
89         }
90     }
91 }

```



```

79     public void SetAC()
80     {
81         this.area = (((2 * lado + 2 * lado2) * Convert.ToSingle(System.Math.Sqrt(altura*altura+(lado*lado)/4))) / 2)*(lado * lado2);
82         //area=(perimetro de la base)(altura del triangulo, no de la piramide)/2+(area de la base)
83     }
84     1 referencia
85     public void SetAP()
86     {
87         this.area = ((n1 * lado * Convert.ToSingle(System.Math.Sqrt(altura * altura + System.Math.Sqrt(lado * lado - (lado * lado) / 2))) / 2) + ((lado * n1 * Convert.ToSingle(System.Math.Sqrt(lado * lado - (lado * lado) / 2))) / 2);
88         //area=(perimetro de la base)(altura del triangulo, no de la piramide)/2+(area de la base)
89     }
90     1 referencia
91     public void SetVT()
92     {
93         this.volumen = (lado * lado * lado * Convert.ToSingle(System.Math.Sqrt(2))) / 12;
94         //volumen=(lado^3*raiz(2))/12
95     }
96     1 referencia
97     public void SetVC()
98     {
99         this.volumen = (lado * lado2 * altura) / 3;
100         //volumen=(base*altura)/3
101     }
102     1 referencia
103     public void SetVP()
104     {
105         this.volumen = (((lado * n1 * Convert.ToSingle(System.Math.Sqrt(lado * lado - (lado * lado)/2))) / 2) * altura) / 3;
106         //volumen=(base*altura)/3
107     }

```

```

104     public float GetArea()
105     {
106         return area;
107     }
108     3 referencias
109     public float GetVolumen()
110     {
111         return volumen;
112     }
113     1 referencia
114     public int GetNumLados()
115     {
116         return n1;
117     }
118     1 referencia
119     public void SetPT()
120     {
121         this.perimetro = 6 * lado;
122     }
123     1 referencia
124     public void SetPC()
125     {
126         atc();
127         this.perimetro = 4 * (Convert.ToSingle(System.Math.Sqrt(ATC * ATC + (lado * lado) / 4))) + (4 * lado);
128     }
129     1 referencia
130     public void atc()
131     {
132         this.ATC = Convert.ToSingle(System.Math.Sqrt(altura * altura + (lado * lado) / 4));
133     }
134     1 referencia
135     public void SetPP()
136     {
137         atp();
138         this.perimetro = n1 * (Convert.ToSingle(System.Math.Sqrt(ATP*ATP+(lado*lado)/4))) + (n1 * lado);
139     }
140     1 referencia
141     public void atp()
142     {
143         this.ATP = Convert.ToSingle(System.Math.Sqrt(altura * altura + System.Math.Sqrt(lado * lado - (lado * lado) / 2)));

```

```

138     public float GetP()
139     {
140         return perimetro;
141     }
142 }
143

```

```

1  using System;
2
3  namespace Ejercicios
4  {
5      [reference]
6      class Esfera
7      {
8          private float radio, area, volumen;
9          [reference]
10         public Esfera()
11         {
12             //Constructor por defecto
13         }
14         [reference]
15         public void SetRadio(float radio)
16         {
17             if (radio < 0)
18             {
19                 radio = -1 * radio;
20                 this.radio = radio;
21                 Console.WriteLine("No puede haber radios negativos, por lo tanto la convertimos en positivo.");
22                 Console.WriteLine("Presione cualquier tecla para continuar.");
23                 Console.ReadLine();
24             }
25             else
26             {
27                 this.radio = radio;
28             }
29         }
30         [reference]
31         public void SetArea()
32         {
33             this.area = 4 * Convert.ToSingle((System.Math.PI)) * radio * radio;
34         }
35         [reference]
36         public float GetArea()
37         {
38             return this.area;
39         }
40         [reference]
41         public void SetVolumen()
42         {
43             this.volumen = (4 * Convert.ToSingle((System.Math.PI)) * radio * radio * radio) / 3;
44         }
45         [reference]
46         public float GetVolumen()
47         {
48             return this.volumen;
49         }
50     }
51 }

```

```

1  using System;
2
3  namespace Ejercicios
4  {
5      [reference]
6      class Cubo
7      {
8          private float perimetro, area, volumen, lado;
9          [reference]
10         public Cubo()
11         {
12         }
13         [reference]
14         public void SetLado(float lado)
15         {
16             if (lado < 0)
17             {
18                 lado = -1 * lado;
19                 this.lado = lado;
20                 Console.WriteLine("No puede haber lados negativos, por lo tanto la convertimos en positivo.");
21                 Console.WriteLine("Presione cualquier tecla para continuar.");
22                 Console.ReadLine();
23             }
24             else
25             {
26                 this.lado = lado;
27             }
28         }
29         [reference]
30         public void SetPerimetro()
31         {
32             this.perimetro = this.lado * 12;
33         }
34         [reference]
35         public float GetPerimetro()
36         {
37             return this.perimetro;
38         }
39         [reference]
40         public void SetArea()
41         {
42             this.area = this.lado * this.lado * 6;
43         }
44         [reference]
45         public float GetArea()
46         {
47             return this.area;
48         }
49         [reference]
50         public void SetVolumen()
51         {
52             this.volumen = this.lado * this.lado * this.lado;
53         }
54         [reference]
55         public float GetVolumen()
56         {
57             return this.volumen;
58         }
59     }
60 }

```

```

Este programa calculara el area, volumen y perimetro (Si es que se puede) de distintas figuras geometricas
Menu
1.-Cubo
2.-Esfera
3.-Cilindro
4.-Piramide
5.-Salir
Elija una figura: _

```

```

Elegio al Cubo, por favor ingrese el tamaño de su lado: _

```

```

Cubo
Menu interno
1.-Perimetro
2.-Area
3.-Volumen
4.-Salir
Elija una opcion: 1
El perimetro del Cubo es de: 60 unidades
Presione cualquier tecla para continuar.

```

```

Elegio a la Esfera, por favor ingrese el tamaño de su Radio:
_

```

```

Esfera
Menu interno
1.-Area
2.-Volumen
3.-Salir
Elija una opcion: 1
El area de la Esfera es: 314.1593 unidades cuadradas
Presione cualquier tecla para continuar.

```

```

Piramide
Menu interno
1.-Triangulo
2.-Cuadrilatero
3.-Poligono de 5 lados o mas lados
4.-Salir
Elija la figura que tiene como base de la Piramide: _

```

```

Piramide Triangular (Tetraedro)
Ingrese el tamaño del lado: _

```

```

Tetraedro
Menu interno 2
1.-Area
2.-Volumen
3.-Perimetro
4.-Salir
Elija una opcion: 2
El volumen de la piramide es: 3.18198 unidades cubicas
Presione cualquier tecla para continuar._

```

REVISIÓN: _____

- 2) Realizar un cajero automático, el usuario ingresará su número de cuenta y contraseña para iniciar sesión, una vez dentro podrá realizar los siguiente en su cuenta:
- Consultar Saldo
 - Retirar efectivo
 - Transferir a otro usuario (imprimir saldo del cliente)
 - Pagar servicio

El cuentahabiente tiene como datos: nombre, apellido, correo, saldo inicial (\$1750), empresa.

El cuentahabiente a quien se le hará la transferencia tiene como saldo inicial \$200. Al finalizar una transferencia se podrá cerrar sesión y entrar a la sesión del cuentahabiente que se le transfirió para ver su saldo nuevo.

```

1  /*
2  Arriaga Mejía José Carlos.
3  Fragoso Islas Ana Cecilia.
4  Medina Perabeles Rodrigo.
5  Pérez Duarte Brenda Elizabeth.
6
7  * 2) Realizar un cajero automático, el usuario ingresará su número de cuenta y contraseña para iniciar sesión, una vez dentro podrá
8  * realizar los siguiente en su cuenta:
9  a. Consultar Saldo
10 b. Retirar efectivo
11 c. Transferir a otro usuario (imprimir saldo del cliente)
12 d. Pagar servicio
13 El cuentahabiente tiene como datos: nombre, apellido, correo, saldo inicial ($1750), empresa.
14 El cuentahabiente a quien se le hará la transferencia tiene como saldo inicial $200. Al finalizar una transferencia se podrá
15 cerrar sesión y entrar a la sesión del cuentahabiente que se le transfirió para ver su saldo nuevo.
16
17 Equipo 04
18
19 22/10/2020
20 */
21
22 using System;
23 using System.Collections.Generic;
24 using System.Linq;
25 using System.Security.Cryptography.X509Certificates;
26 using System.Text;
27 using System.Threading;
28
29 namespace Ejemplo1
30 {
31
32     0 referencias
33     class Program
34     {
35         0 referencias
36         static void Main(string[] args)
37         {
38             Cajero c = new Cajero();
39             /*Usuario 1:
40             * Id = 12345
41             * contraseña = 12345
42             * Nombre: Tony
43             * Apellido: Stark
44             * correo: ironman@avengers.com
45             * Saldo inicial: $1750
46             * Empresa: Avengers
47             * */

```

```

46     string n1 = "Tony", a1 = "Stark", correo1 = "Ironman@avengers.com", em1 = "Avengers";
47     double s1 = 1750;
48
49     /*Usuario 2:
50     * id = 98765
51     * contraseña = 98765
52     * Nombre: Steve
53     * Apellido: Rogers
54     * correo: capitanaamerica@avengers.com
55     * Saldo inicial: $0
56     * Empresa: Avengers
57     */
58     string n2 = "Steve", a2 = "Rogers", correo2 = "capitanaamerica@avengers.com", em2 = "Avengers";
59     double s2 = 200;
60
61     double s1t = s1, s2t = s2, aux;
62
63
64
65     string id, contra;
66     int opc;
67     string bucle1;
68     do
69     {
70         bucle1 = "0";
71         Console.Clear();
72         Console.WriteLine("Hola, Bienvenido a su cajero automático");
73         Console.WriteLine("Ingrese su ID de usuario:");
74         id = Convert.ToString(Console.ReadLine());
75         Console.WriteLine("Ahora ingrese su contraseña: ");
76         contra = Convert.ToString(Console.ReadLine());
77         Console.WriteLine("Verificando");
78         Console.WriteLine("...");
79         Thread.Sleep(1000);
80         Console.WriteLine("...");
81         Thread.Sleep(1000);
82         Console.WriteLine("...");
83         Thread.Sleep(1000);
84
85         if (id == "12345" && contra == "12345")
86         {
87             do
88             {
89                 bucle1 = "1";
90                 Console.Clear();

```

```

90         Console.Clear();
91         c.menu(a1);
92         opc = Convert.ToInt32(Console.ReadLine());
93         switch (opc)
94         {
95             case 1:
96                 c.consultasaldo(s1, a1);
97                 break;
98             case 2:
99                 s1 = c.retiro(s1, a1);
100
101                 break;
102             case 3:
103
104                 s2 = c.transferencia(s1, s2, a1);
105                 if (c.op == 0)
106                 {
107                     aux = s2 - s2t;
108                     s1 = s1 - aux;
109                     s2t = s2;
110                 }
111                 break;
112             case 4:
113
114                 s1 = c.pago(s1, a1);
115
116                 break;
117             case 5:
118                 c.info(n1, a1, correo1, s1, em1);
119                 break;
120
121         }
122     } while (opc != 6);
123
124 }
125
126 else
127 if (id == "98765" && contra == "98765")
128 {
129     do
130     {
131         bucle1 = "1";
132         Console.Clear();
133         c.menu(a2);
134         opc = Convert.ToInt32(Console.ReadLine());

```

```

135         switch (opc)
136         {
137             case 1:
138                 c.consultasaldo(s2, a2);
139                 break;
140             case 2:
141                 s1 = c.retiro(s2, a2);
142                 break;
143             case 3:
144                 s2 = c.transferencia(s2, s1, a2);
145                 if (c.op == 0)
146                 {
147                     aux = s1 - s1t;
148                     s2 = s2 - aux;
149                     s1t = s1;
150                 }
151                 break;
152             case 4:
153                 s2 = c.pago(s2, a2);
154                 break;
155             case 5:
156                 c.info(n2, a2, correo2, s2, em2);
157                 break;
158         }
159     } while (opc != 6);
160 }
161 }
162 else
163 {
164     Console.WriteLine("\nSu ID y/o contraseña son incorrectos.\nDesea volver a intentar?:\nPresione 1 para si");
165     bucle1 = Convert.ToString(Console.ReadLine());
166     Console.WriteLine("...");
167     Thread.Sleep(1000);
168     Console.WriteLine("...");
169     Thread.Sleep(1000);
170     Console.WriteLine("...");
171     Thread.Sleep(1000);
172 }
173 }
174 if (bucle1 != "1")
175     bucle1 = "0";
176 }

```

```

174         if (bucle1 != "1")
175             bucle1 = "0";
176     }
177     while (bucle1 != "0");
178     Console.ReadLine();
179 }
180 }
181 }
182 }

```

```

Hola, Bienvenido a su cajero automático          usuario1 : id=12345, contra=12345

Ingrese su ID de usuario:                        usuario2 : id=98765, contra=98765
12345
Ahora ingrese su contraseña:
12345
Verificando
.....

```

```

Hola Sr(a).Stark que operacion deseas realizar
1.-Consulta de saldo
2.- Retiro
3.- Transferencia
4.- pago de servicio
5.- consultar info
6.- cerrar sesion

```

```

Bienvenido Sr(a). Stark
Su saldo actual es de: $ 1750
Presione cualquier tecla para continuar y luego oprima ENTER.

```

```

Bienvenido Sr(a). Stark
Su saldo actual es de: $ 1750
Cuanto desea retirar:
1000
...
Su retiro se realizó con éxito.
Ahora su saldo es de: $ 750
....._

```

```

Bienvenido Sr(a). Stark
Su saldo actual es de: $ 750
Cuanto desea transferir:
100
...
Su transferencia se realizó con éxito.
Ahora su saldo es de: $ 650
....._

```

```

Bienvenido(a) Sr(a). Stark

Tenemos pagos disponibles:
1.-Agua
2.-Gas
3.-Luz
Ingrese el numero del que desea pagar:
_

```

```

Bienvenido(a) Sr(a). Stark

Tenemos pagos disponibles:
1.-Agua
2.-Gas
3.-Luz
Ingrese el numero del que desea pagar:
1
Ahora ingrese el monto a pagar:
129.5
...
Su pago se realizó con éxito:
Ahora su saldo es de $520.5
...

```

```

Nombre: Tony
Apellido: Stark
Correo: ironman@avengers.com
Saldo: $ 650
Empresa: Avengers
Presione cualquier tecla para continuar y luego ENTER.

```

```

Bienvenido Sr(a). Rogers
Su saldo actual es de: $ 400
Presione cualquier tecla para continuar y luego oprima ENTER.

```

REVISIÓN: _____

- 3) Realizar la AgendaPoo Free, crear la Clase Contacto donde se almacenará la información, deberá de contar con las siguientes opciones.
- Guardar contacto (Nombre, apellido, teléfono, correo y dirección)
 - Buscar Contacto
 - Ver contactos

Código

The image displays two screenshots of a Visual Studio IDE showing C# code for a contact agenda application.

Top Screenshot: Shows the initial setup of the `Agenda` class. The code includes comments in Spanish, using statements for `System.Collections.Generic`, `System.Linq`, and `System.Text`, and a namespace `Practica4`. The `Main` method initializes four `Contactos` objects and sets up a loop.

```

1  /* Arriaga Mejía José Carlos
2  Fragozo Islas Ana Cecilia
3  Medina Perabeles Rodrigo
4  Pérez Duarte Brenda Elizabeth
5
6  Práctica 1 - Ejemplo 7
7  Descripción: Se realizará una agenda de contactos con capacidad de 4 contactos, con las opciones de agregar contacto,
8  buscar contacto y ver contacto.
9  Equipo : 04
10 Fecha 22/10/2020*/
11
12 using System;
13 using System.Collections.Generic;
14 using System.Linq;
15 using System.Text;
16
17 namespace Practica4
18 {
19     class Agenda
20     {
21     }
22
23     static void Main(string[] args)
24     {
25         string info;
26         int rep, opc;
27         int NC = 0;
28         int tel;
29         Contactos Contacto1 = new Contactos();
30         Contactos Contacto2 = new Contactos();
31         Contactos Contacto3 = new Contactos();
32         Contactos Contacto4 = new Contactos();
33     }
34 }

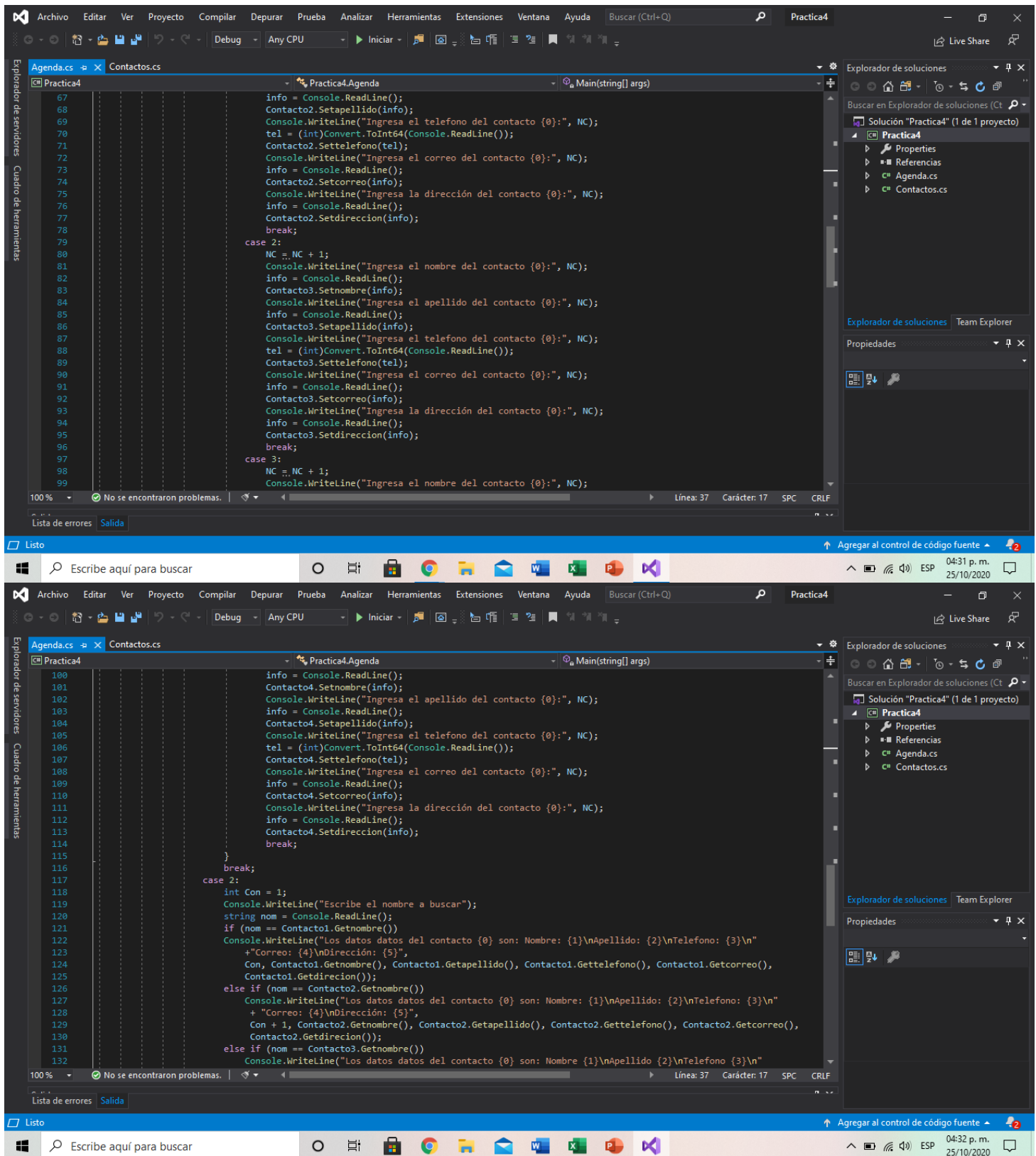
```

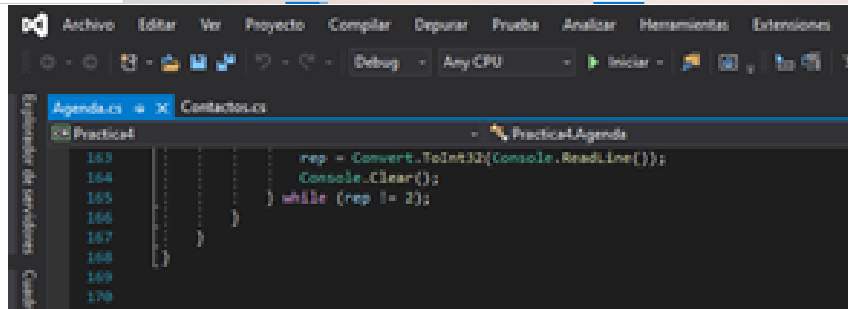
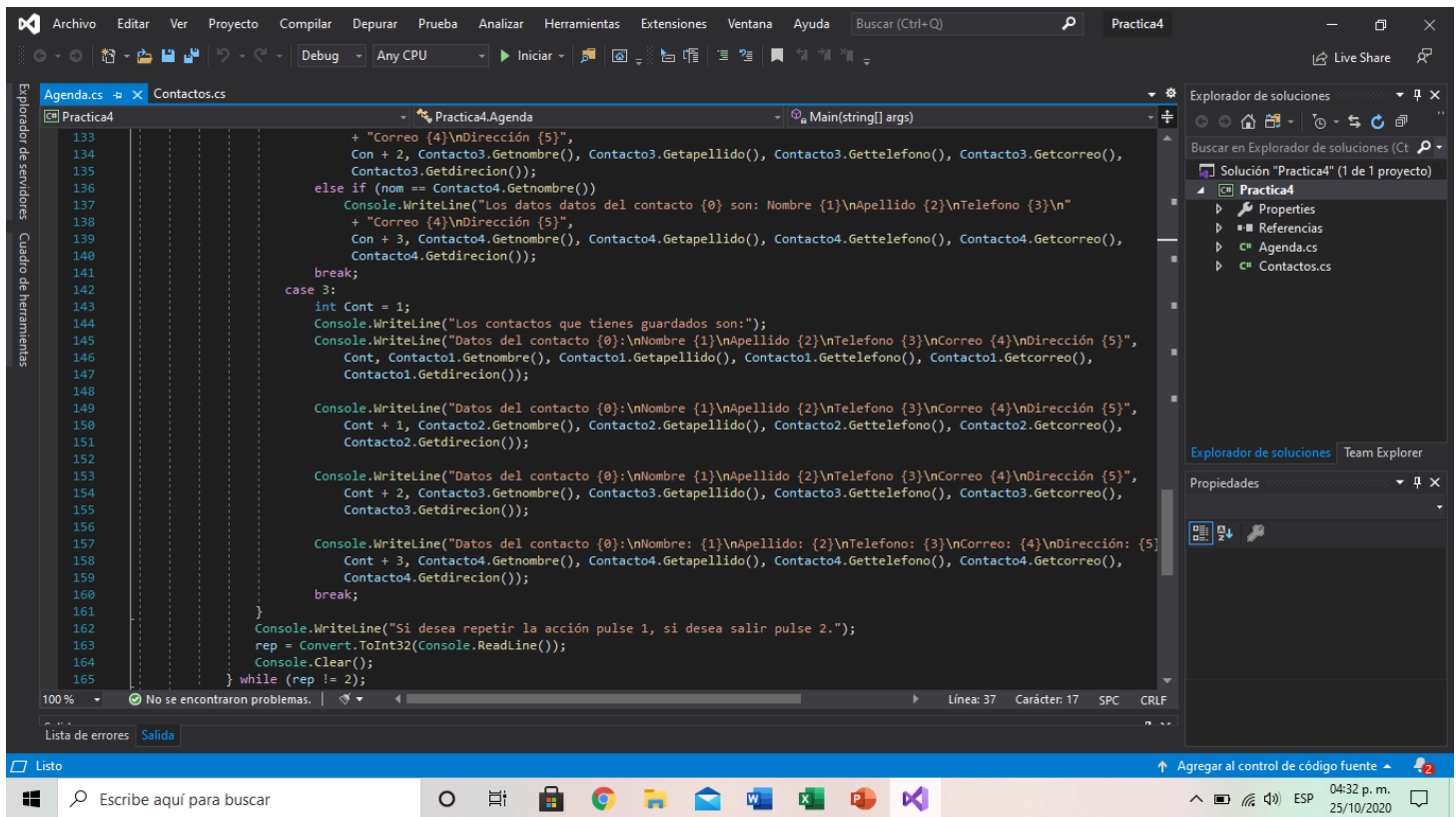
Bottom Screenshot: Shows the implementation of the `Main` method. It includes a welcome message, a loop to read user input, and a switch statement to handle different actions: adding a contact (case 0), adding another contact (case 1), or viewing contacts (case 2).

```

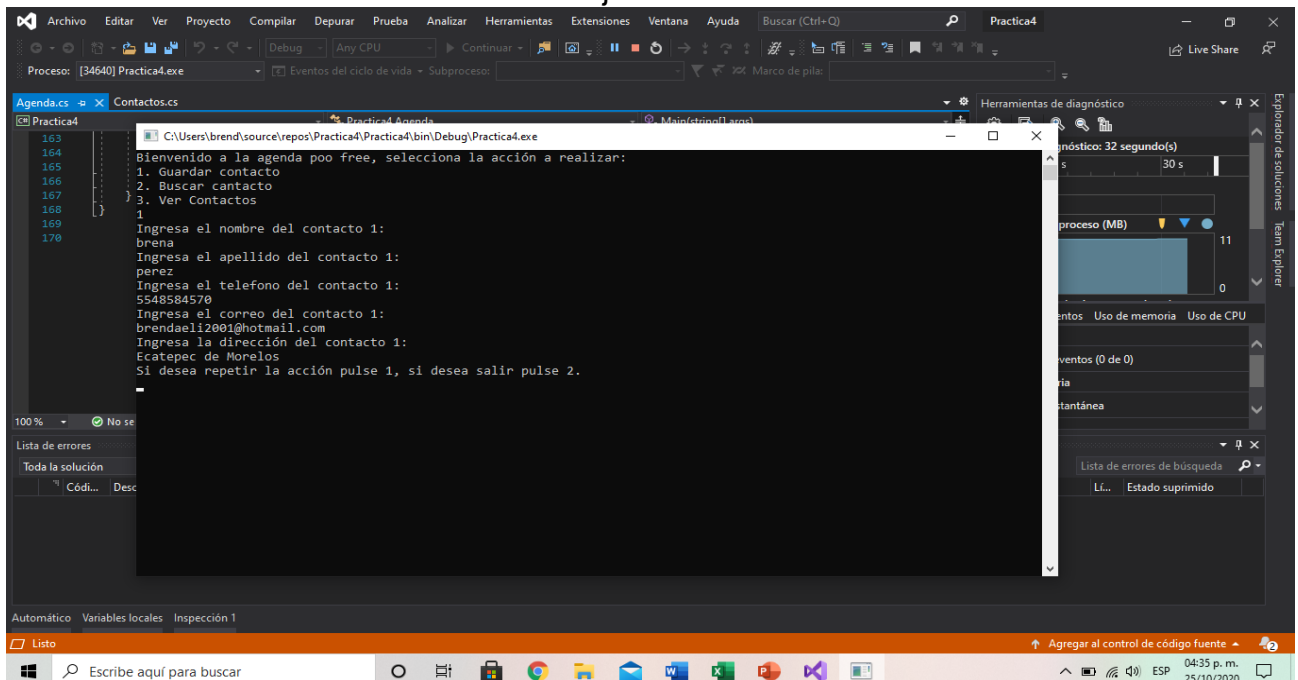
34 Console.WriteLine("Bienvenido a la agenda poo free, selecciona la acción a realizar: \n1. Guardar contacto\n2. Buscar cant
35 \n3. Ver Contactos");
36
37 opc = Convert.ToInt32(Console.ReadLine());
38 switch (opc)
39 {
40     case 1:
41         switch (NC)
42         {
43             case 0:
44                 NC = NC + 1;
45                 Console.WriteLine("Ingresa el nombre del contacto {0}:", NC);
46                 info = Console.ReadLine();
47                 Contacto1.Setnombre(info);
48                 Console.WriteLine("Ingresa el apellido del contacto {0}:", NC);
49                 info = Console.ReadLine();
50                 Contacto1.Setapellido(info);
51                 Console.WriteLine("Ingresa el telefono del contacto {0}:", NC);
52                 tel = (int)Convert.ToInt64(Console.ReadLine());
53                 Contacto1.Settelefono(tel);
54                 Console.WriteLine("Ingresa el correo del contacto {0}:", NC);
55                 info = Console.ReadLine();
56                 Contacto1.Setcorreo(info);
57                 Console.WriteLine("Ingresa la dirección del contacto {0}:", NC);
58                 info = Console.ReadLine();
59                 Contacto1.Setdireccion(info);
60                 break;
61             case 1:
62                 NC = NC + 1;
63                 Console.WriteLine("Ingresa el nombre del contacto {0}:", NC);
64                 info = Console.ReadLine();
65                 Contacto2.Setnombre(info);
66                 Console.WriteLine("Ingresa el apellido del contacto {0}:", NC);

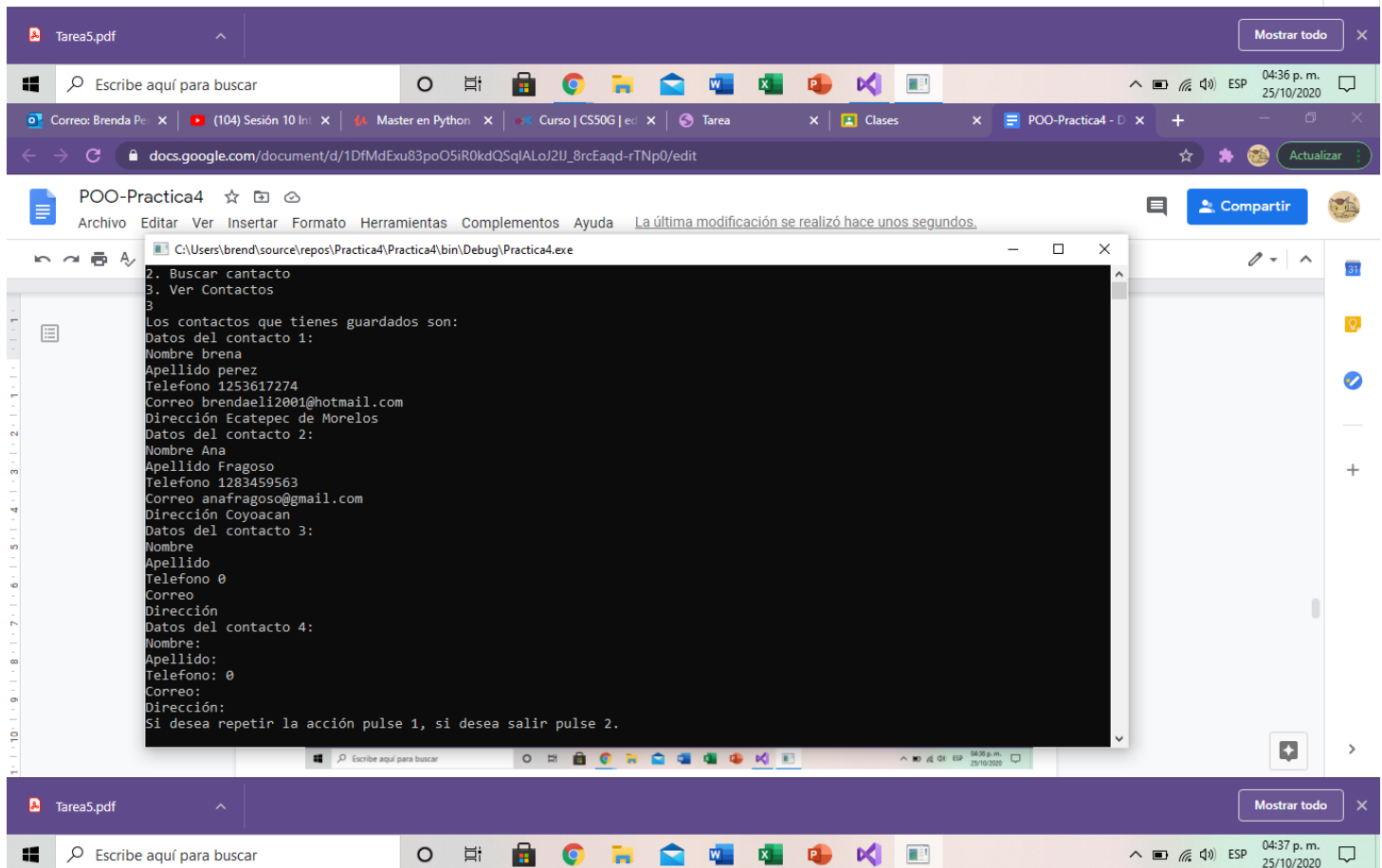
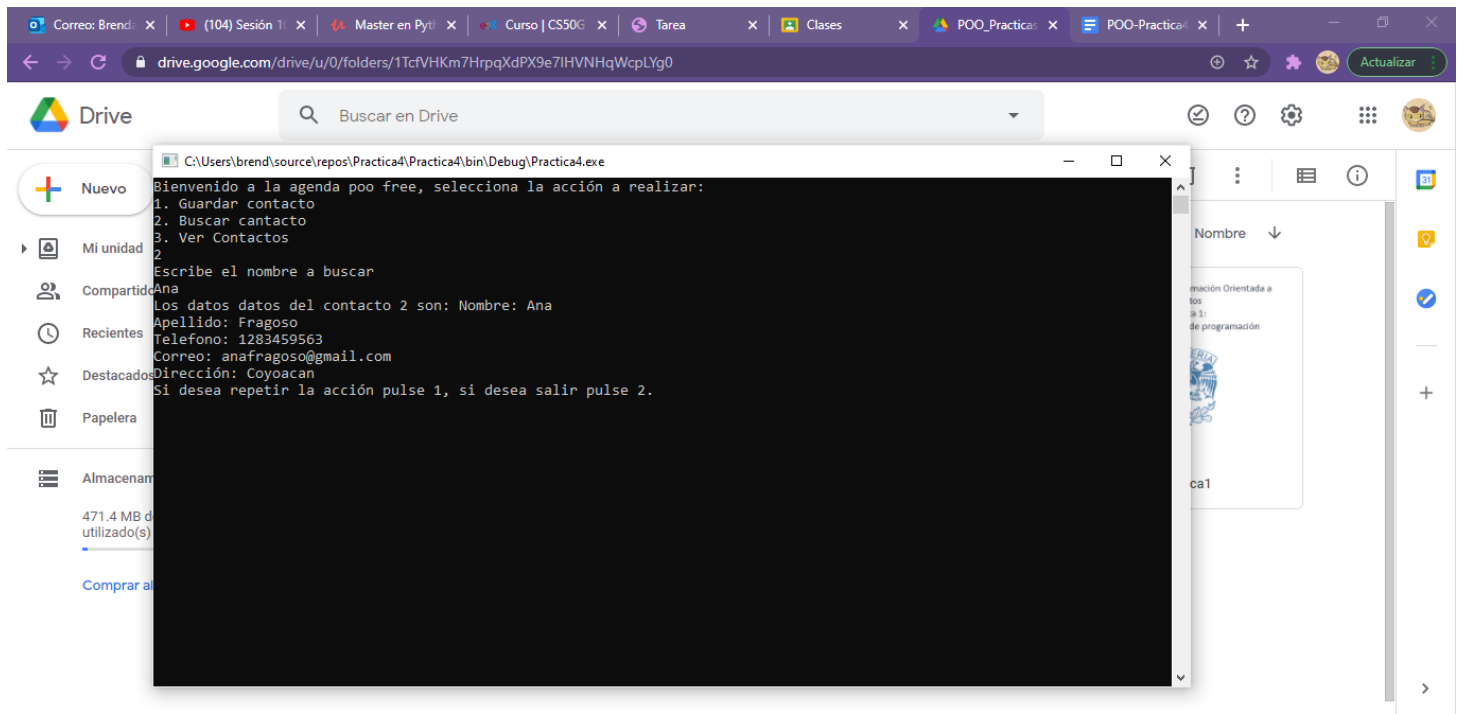
```



Ejecución





REVISIÓN: _____

- 4) Realizar un programa que simule las Ordenes que ingresan a la cocina de un restaurante, se deberá de poder
- Ingresar nueva orden (platillo, cantidad, bebida, mesa)
 - Revisar ordenes
 - Entregar orden

```

1  using System;
2
3  namespace ejercicio084
4  {
5      3 referencias
6      public class restaurante
7      {
8          string platillo;
9          int cantidad;
10         string bebida;
11         int mesa;
12
13         //métodos
14         1 referencia
15         public restaurante()
16         {
17         }
18         8 referencias
19         public void nuevaOrden()
20         {
21             Console.WriteLine("¿Qué deseas ordenar?");
22         }
23         0 referencias
24         public void revisarOrden()
25         {
26         }
27         1 referencia
28         public void entregarOrden()
29         {
30             Console.WriteLine("Espera un momento, por favor. La orden está siendo entregada...");
31             System.Threading.Thread.Sleep(4000);
32             Console.WriteLine("¡La orden fue entregada con éxito!");
33         }
34         1 referencia
35         public void guardaPlatillo(string plato)
36         {
37             platillo = plato;
38         }
39         1 referencia
40         public void guardaCantidad(int n)
41         {
42             cantidad = n;
43         }
44         1 referencia
45         public void guardaBebida(string soda)
46         {
47             bebida = soda;
48         }
49         1 referencia
50         public void guardaMesa(int posicion)
51         {
52             mesa = posicion;
53         }
54         //Métodos para guardar y mostrar los atributos
55         1 referencia
56         public string obtenPlatillo()
57         {
58             Console.WriteLine(platillo);
59             return platillo;
60         }
61         1 referencia
62         public int obtenCantidad()
63         {
64             Console.WriteLine("Pediste "+cantidad+" "+platillo+".");
65             return cantidad;
66         }
67         1 referencia
68         public string obtenBebida()
69         {
70             Console.WriteLine("Tu bebida es: "+bebida+".");
71             return bebida;
72         }
73         1 referencia
74         public int obtenMesa()
75         {
76             Console.WriteLine("Tu mesa es la número "+mesa+".");
77             return mesa;
78         }
79     }
80 }

```

```

¡Hola!, bienvenido a The Ledbury. :)
¿Qué deseas hacer?

1.Ingresar nueva orden.
2.Revisar orden.
3.Entregar orden.
4.Salir
1
¿Qué se te antoja hoy?
Huevos con tocino
¿Qué cantidad de Huevos con tocino?
1
¡Perfecto!
¿Qué apeteces para tomar?
awa
¡Buena elección!
Por último, ¿cuál es tu número de mesa?
5
Tu orden ha sido enviada. ¡Gracias!
-

```

```

¡Hola!, bienvenido a The Ledbury. :)
¿Qué deseas hacer?

1.Ingresar nueva orden.
2.Revisar orden.
3.Entregar orden.
4.Salir
2
Tu orden es:
Huevos con tocino
Pediste 1 Huevos con tocino.
Tu bebida es: awa.
Tu mesa es la número 5.

```

```

¡Hola!, bienvenido a The Ledbury. :)
¿Qué deseas hacer?

1.Ingresar nueva orden.
2.Revisar orden.
3.Entregar orden.
4.Salir
3
Espera un momento, por favor. La orden está siendo entregada...
¡La orden fue entregada con éxito!

```

REVISIÓN: _____

Conclusiones.***Arriaga Mejía José Carlos.***

En esta práctica utilizamos varios conceptos importantes para la programación orientada a objetos, utilizamos distintas clases y las instanciamos, también aprendimos que son los constructores, para que funcionan y cómo usarlos. Además, aprendimos a encapsular atributos y sus formas de guardar y acceder a la información encapsulada.

Pérez Duarte Brenda Elizabeth.

Gracias a esta práctica pudimos aplicar los conocimientos adquiridos en la semana sobre la creación y utilización de clases para hacer un poco más dinámicos o sencillos para realizar códigos más identificables en sus partes, así como aplicamos las variables Set (para guardar algún dato) y Get (para poder mostrar algún elemento, algo así como llamarlo de la variable en la que se encuentra).

Medina Perabeles Rodrigo.

Esta práctica se me complicó un poco debido al nuevo uso de clases para mí. Una vez que comprendí como usar las clases me di cuenta que nos ayudan a ahorrarnos demasiado código y demasiadas horas de trabajo, y hasta hacen más sencillo el proceso. Puedo concluir que la programación va más allá de una función. Es un conjunto de cosas que hay que tener claras las bases para llegar a un resultado.

Fragoso Islas Ana Cecilia.

En esta práctica, con ayuda de los conocimientos previos que estuvimos recordando en los temas pasados, pudimos implementar ahora sí la parte Orientada a Objetos, la cual fue muy interesante y sencilla de realizar debido a lo que habíamos trabajado en teoría. Para concluir puedo mencionar que es fundamental tener muy presentes los conceptos de la teoría para poder aplicarlos en la práctica, ya que como antes habíamos trabajado con programación estructura, puede resultar algo difícil acostumbrarnos a la nueva forma de programación.