

```
1 namespace ResumenFunciones
2 {
3     public class ExerciseCabanesBook
4     {
5         // EJERCICIO 0.1
6         // FUNCION que comprueba si un texto es un Palindromo.
7         public static bool IsPalindrome(string texto)
8         {
9             int j = texto.Length - 1;
10
11             for (int i = 0; i < texto.Length; i++)
12             {
13                 if (texto[i] != texto[j])
14                     return false;
15                 if (texto == "")
16                     return false;
17                 j--;
18             }
19             return true;
20         }
21
22         // EJERCICIO 0.2
23         // FUNCION que indica si un elemento esta dentro de un array.
24         public static bool ContainsElement(int[,] array, int row, int column)
25         {
26             if (row > array.GetLength(0) || column > array.GetLength(1))
27                 return false;
28             return true;
29         }
30
31         // EJERCICIO 1.11.2
32         // FUNCION que imprime la tabla de multiplicar de un numero dado.
33         public static void MultiplyChart(int number)
34         {
35             for (int i = 0; i < 11; i++)
36             {
37                 Console.WriteLine($"{number} x {i} = {number * i}");
38             }
39         }
40
41         // EJERCICIO 1.11.4
42         // FUNCION que devuelve la conversion de grados Celsius.
43         public static (double, double) CalculateConversion(double celsius)
44         {
45             double Kelvin;
46             double Fahrenheit;
47
48             Fahrenheit = (celsius * 1.8) / 10.0 + 32.0;
49             Kelvin = celsius + 273;
```

```
50
51     // Sintaxis para devolver una 'tupla'
52     return (Kelvin, Fahrenheit);
53 }
54
55 // EJERCICIO 2.1.8.1-2
56 // OPERADOR ternario.
57 public static double GetAbsolute(double number)
58 {
59     // Sintaxis de una condicional ternaria
60     // tipo variable = condicion a cumplir ? true : false;
61
62     double result = number > 0 ? number : -number;
63     return number > 0 ? number : -number;
64 }
65
66 // EJERCICIO 2.1.8.1-3
67 // OPERADOR ternario.
68 public static int GetMajor(int number1, int number2)
69 {
70     return (number1 > number2) ? number1 : number2;
71 }
72
73 // EJERCICIO 2.1.9.1
74 // OPERADOR switch
75 public static string TextChar(char letter)
76 {
77     switch (letter)
78     {
79         case 'a':
80         case 'e':
81         case 'i':
82         case 'o':
83         case 'u':
84             return "Se trata de una vocal";
85         default:
86             return "No es una vocal";
87     }
88 }
89
90 // FUNCION que devuelve conversiones de distintos tipos.
91 public static int CharToNumber(char character)
92 {
93     return (int)character;
94 }
95
96 public static char NumberToChar(int number)
97 {
98     return (char)number;
99 }
100
101 public static int StringToInt(string text)
102 {
```

```
103         return Int32.Parse(text);
104     }
105
106     public static string IntToString(int number)
107     {
108         return number.ToString();
109     }
110
111     // FUNCION que devuelve un booleano si el caracter es mayor.
112     public static bool IsMajorChar(char character, char character2)
113     {
114         return character > character2;
115     }
116
117     // FUNCION de un programa para averiguar un numero.
118     public static void GuessNumber()
119     {
120         Random r1 = new Random();
121
122         int random = r1.Next(0, 10);
123         int count = 0;
124
125         int answer = -1;
126         while (random != answer || count < 5)
127         {
128             Console.WriteLine("Averigua el número");
129
130             answer = Int32.Parse(Console.ReadLine());
131             if (random == answer)
132                 Console.WriteLine("Acertaste!");
133             if (random < answer)
134                 Console.WriteLine("El numero es menor");
135             if (random > answer)
136                 Console.WriteLine("El numero es mayor");
137             count++;
138         }
139         if (random != answer)
140             Console.WriteLine("Has perdido");
141     }
142
143     // FUNCION programa de un conversor de unidades.
144     public static void UnitConversor()
145     {
146         bool exit = false;
147
148         while (!exit)
149         {
150             Console.WriteLine("¿Que conversor quiere utilizar?:");
151             Console.WriteLine("[0]Euros a Dolares\n" +
152                 "[1]Celsius a Fahrenheit\n" +
153                 "[2]Centimetros a Metros\n" +
154                 "[3]Centimetros a Pulgadas\n" +
155                 "[4]Caballos de vapor a Kilovatios\n" +
```

```
156         "[5]Salir");
157
158         string choice = Console.ReadLine();
159
160         if (choice == "5")
161         {
162             exit = true;
163         }
164
165         else
166         {
167             Console.WriteLine("Introduzca la cantidad:");
168             string number = Console.ReadLine();
169             double value = Int32.Parse(number);
170
171             switch (choice)
172             {
173                 case "0":
174                     Console.WriteLine(value * 1.06);
175                     break;
176                 case "1":
177                     Console.WriteLine(value * (9 / 5) + 32);
178                     break;
179                 case "2":
180                     Console.WriteLine(value * 0.01);
181                     break;
182                 case "3":
183                     Console.WriteLine(value * 2.54);
184                     break;
185                 case "4":
186                     Console.WriteLine(value * 0.73);
187                     break;
188                 default:
189                     break;
190             }
191         }
192     }
193 }
194
195 // INSTRUCCION 'CONTINUE'
196 // TODO: [EXAMEN] control de bucles con 'continue'.
197 public static void ContinueFunction()
198 {
199     for (int i = 2; i < 20; i++)
200     {
201         if (i == 13)
202             continue;
203         Console.WriteLine(i);
204     }
205 }
206
207 // INSTRUCCION 'BREAK'
208 // TODO: [EXAMEN] control de bucles con 'break'.
```

```
209     public static void BreakFunction()
210     {
211         bool exit = false;
212
213         while (!exit)
214         {
215             Console.WriteLine("Escriba 'exit' para salir");
216             string answer = Console.ReadLine();
217             if (answer == "exit")
218                 exit = !exit;
219         }
220     }
221
222     // FUNCION QUE CALCULA EL MCM.
223     // Es el numero mas pequeño de los multiplos comunes.
224     // TODO: [EXAMEN] Funcion MCM.
225     public static int CalculateMCM(int number1, int number2)
226     {
227         int result = 1;
228
229         for (int i = 2; i < number1 || i < number2; i++)
230         {
231             if (number1 % i == 0 && number2 % i == 0)
232                 result = i;
233         }
234         return result;
235     }
236
237     // FUNCION QUE CALCULA EL MCM DE UNA LISTA.
238     // TODO: [EXAMEN] Funcion MCM de una lista de enteros.
239     public static int CalculateMCM(List<int> list)
240     {
241         int result = 1;
242         int minor = Colecciones.GetMinor(list);
243
244         for (int i = 2; i < minor; i++)
245         {
246             for (int j = 0; j < list.Count; j++)
247             {
248                 if (list[j] % i == 0)
249                     result = i;
250                 else
251                     result = 1;
252             }
253         }
254         return result;
255     }
256
257     // FUNCION QUE CALCULA EL MCD.
258     // MCD es el mayor numero que divide todos los elementos.
259     // TODO: [EXAMEN] Funcion MCD.
260     public static int CalculateMCD(int number1, int number2)
261     {
```

```
262         int result = number1 * number2;
263         int minor = number1 < number2 ? number1 : number2;
264
265         for (int i = result; i > minor; i--)
266         {
267             if (i % number1 == 0 && i % number2 == 0)
268                 result = i;
269         }
270         return result;
271     }
272
273     // FUNCION que acepta 5 numeros y devuelve su media.
274     public static int CalculateAverage(int n1, int n2, int n3, int n4, int n5)
275     {
276         return (n1 + n2 + n3 + n4 + n5) / 5;
277     }
278
279     // FUNCION que calcula el número de vocales que tiene un texto.
280     public static int CalculateVowels(string text)
281     {
282         int result = 0;
283         for (int i = 0; i < text.Length; i++)
284         {
285             result += CountVowels(text[i]);
286         }
287         return result;
288     }
289
290     // FUNCION que cuenta vocales.
291     public static int CountVowels(char letter)
292     {
293         switch (letter)
294         {
295             case 'a':
296             case 'e':
297             case 'i':
298             case 'o':
299             case 'u':
300                 return 1;
301             default:
302                 return 0;
303         }
304     }
305
306     // FUNCION que realiza la cuenta atras de caracteres.
307     public static void CountLetters(char letter)
308     {
309         int count = 0;
310         for (char c = letter; c > 'a'; c--)
311         {
312             Console.WriteLine(c);
313             count++;
```

```
314     }
315     Console.WriteLine(count);
316 }
317
318 // FUNCION que cuenta letras.
319 public static void CountLetters(string text)
320 {
321     int count1 = 0;
322     int count2 = 0;
323     int count3 = 0;
324     int resto = 0;
325     int conteo = 0;
326
327     foreach (char letter in text)
328     {
329         conteo = letter == 'a' ? count1++ : count1 += 0;
330         conteo = letter == 'b' ? count2++ : count2 += 0;
331         conteo = letter == 'c' ? count3++ : count3 += 0;
332         resto++;
333     }
334     Console.WriteLine("Jugadorx 1: " + count1);
335     Console.WriteLine("Jugadorx 2: " + count2);
336     Console.WriteLine("Jugadorx 3: " + count3);
337     Console.WriteLine("Banca: " + resto);
338 }
339
340
341 // FUNCION que indica si un numero está dentro de un intervalo.
342 public static bool IsInsideLimits(int n1, int n2, int n3)
343 {
344     return (n1 < n2 && n2 < n3);
345 }
346
347 // FUNCION que calcula nóminas.
348 public static void CalculateNominas(string name, double euroHour, double oldness)
349 {
350     Console.WriteLine("Introduzca el número de horas trabajadas");
351     double workHours = Int32.Parse(Console.ReadLine());
352     Console.WriteLine($"Le trabajadore {name} ha trabajado este mes {workHours}, " +
353         $"a razón de {euroHour} € por hora." +
354         $"Le corresponde un abono de {workHours * euroHour} € debido a su " +
355         $"antigüedad de {oldness} años.");
356 }
357 }
358 }
```