INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

Índice

1. Objetos....1
1.1. Código de Clase1.java....1
1.2. Ejecución de Clase1.java....2
1.3. Conclusiones....8
2. Programas con más de un Objeto....9
2.1. Código de Clase2.java....9
2.2. Objetos Creados al Ejecutar Clase2.java....9
2.3. Conclusiones....9

1. Objetos.

Los objetos pueden imaginarse como pequeños objetos que nacen, viven, se reproducen, y mueren dentro de la memoria de la computadora. Tienen la siguiente estructura interna:

- Atributos: variables que les permiten almacenar información y comunicarse con otros objetos.
- Métodos: secuencias de instrucciones que le permiten al objeto realizar acciones y manipular sus atributos.

Los objetos son creados por otros objetos llamados **clases**, que a su vez también pueden tener sus propios atributos y métodos.. He aquí un ejemplo de un programa que usa objetos.

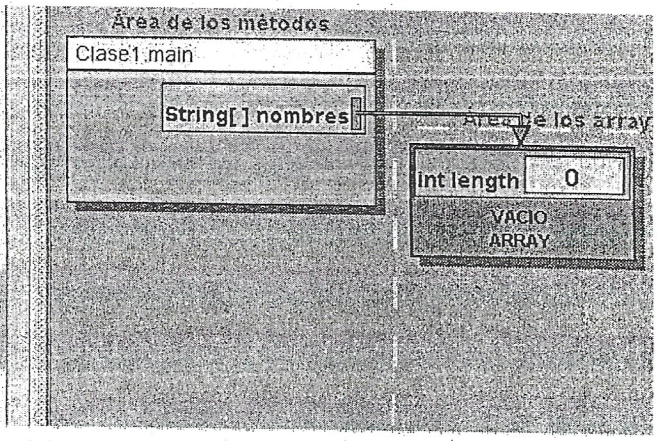
1.1. Código de Clase1.java

```
1  public class Clase1 {
2      // Este es un atributo y se pone uno de estos en cada objeto
3      private String nombre;
4      // Este es un atributo y se pone uno de estos en cada objeto
5      private String telefono;
6      /*
7       * Constructor, Tiene el mismo nombre que la clase y se ejecuta cada
8       * vez que se crea un programa.
9       */
10     public Clase1(String telefono, String nombre) {
11         this.nombre = nombre + " Pérez";
12         this.telefono = telefono;
13     }
14     /*
15      * Este es un método que sirve para mostrar los datos del objeto. Todos
16      * los objetos llevan uno de estos.
17      */
18     public String toString() {
19         // Primero se forma una cadena pegando los textos
20         // entre comillas y las variables. El resultado es
21         // devuelto
22         return "Clase1(" + nombre + ", " + telefono + ")";
23     }
24     // La palabra static indica que es un método de la clase.
25     // Solo hay una copia de este método y le pertenece a
26     // la clase. Con él inicia la ejecución del programa.
27     public static void main(String[] nombres) {
28         // Se crea un objeto de la clase "Clase1" con sus
29         // atributos nulos
30         // refAna es una referencia que apunta al objeto
31         Clase1 refAna = new Clase1("01-800-Manita", "Ana");
32         String datos = refAna.toString();
33         // Muestra la información del objeto apuntado por
34         // refAna
35         System.out.println(datos);
36     }
37 }
```

1.2. Ejecución de Clase1.java

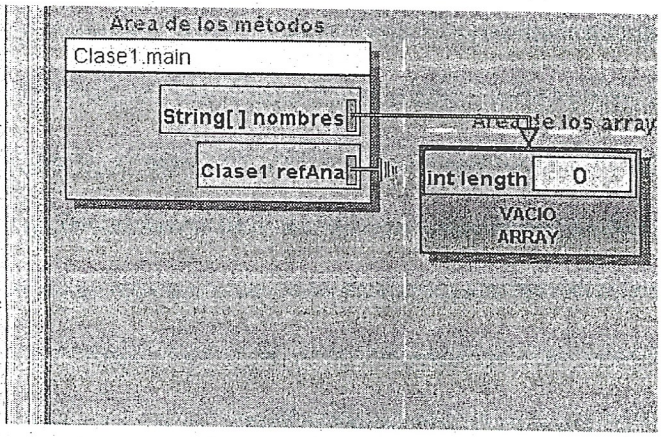
1. Se ejecuta el método "Clase1.main". Se crea un registro de activación que recibe un arreglo de "String". Si no se proporcionan parámetros de ejecución, el arreglo está vacio

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



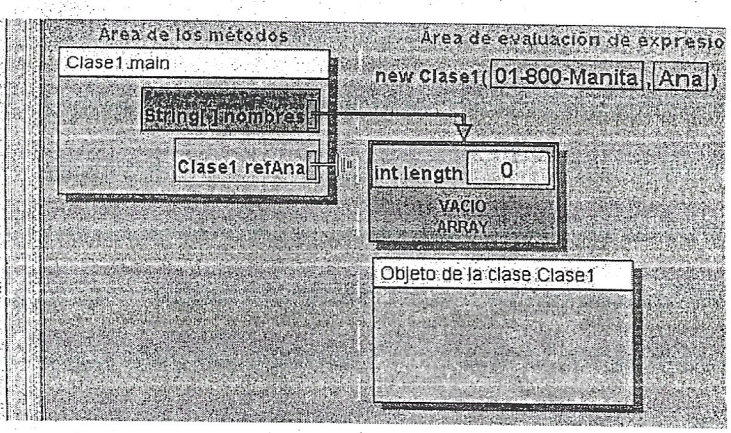
2. Se crea una referencia llamada "refAna" dentro del registro de activación de "Clase1.main".

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



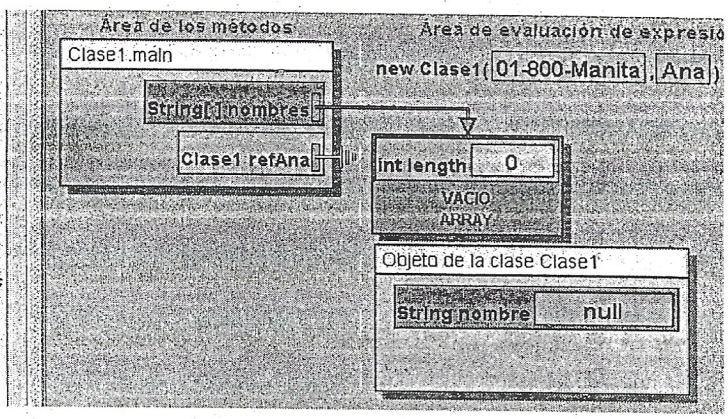
3. Se crea en el Heap un objeto de la clase "Clase1".

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



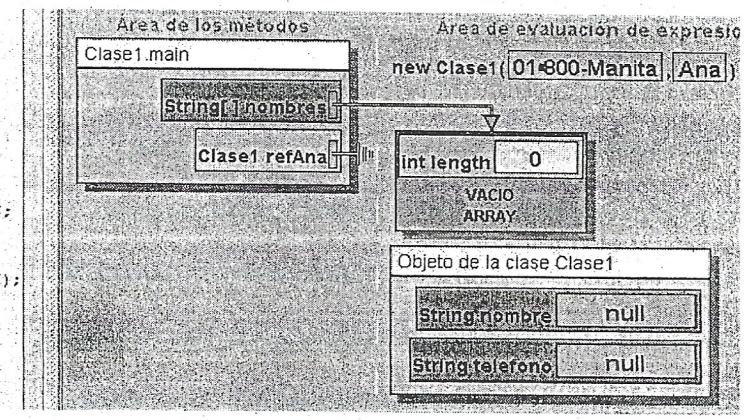
4. Se le añade el atributo nombre con el valor null.

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```

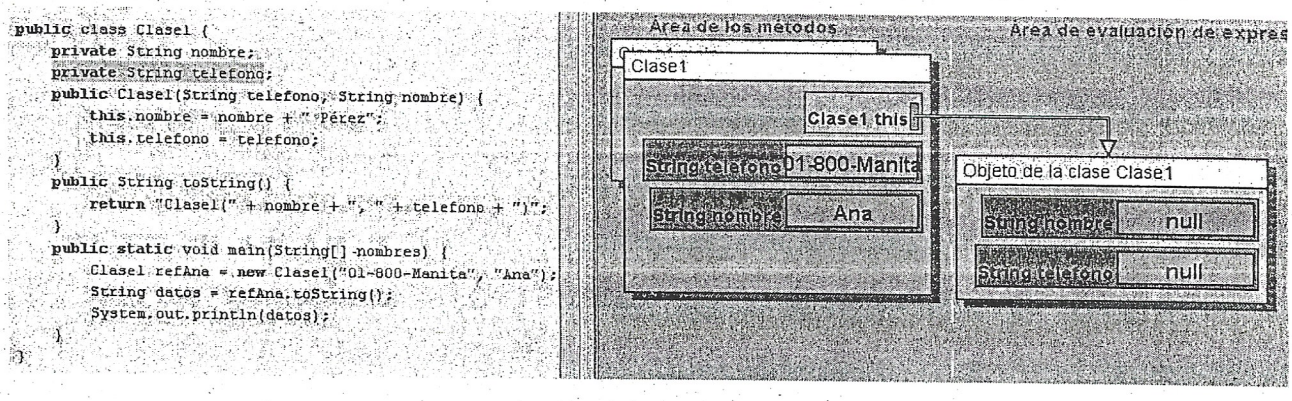


5. Se añade el atributo teléfono con valor null , así como el método "toString( )".

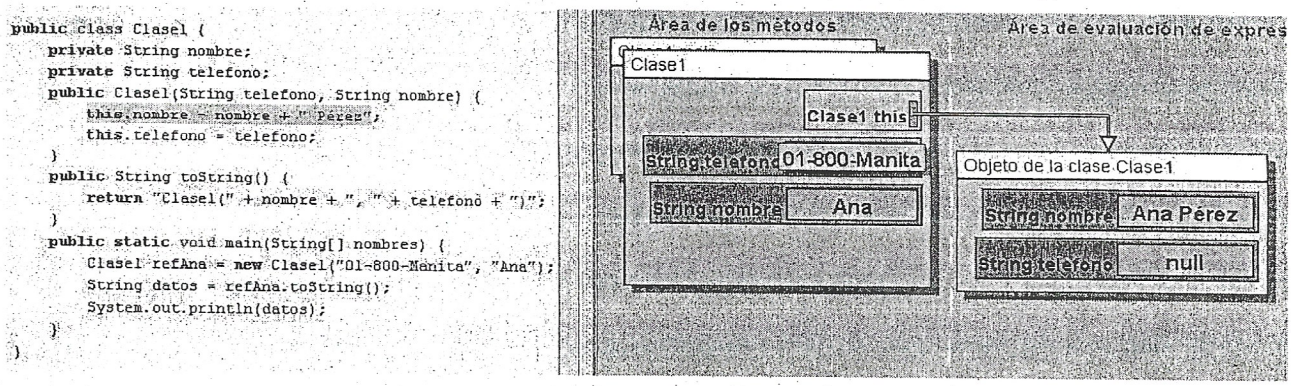
```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



6. Se ejecuta el constructor. Se crea un registro de activación que tiene la referencia this, que apunta al objeto creado, y los parámetros asignados en orden. El primer valor, o sea "01-800-Manita" de asigna al primer parámetro: "telefono". El segundo valor, "Ana" se asigna al segundo valor: "nombre".

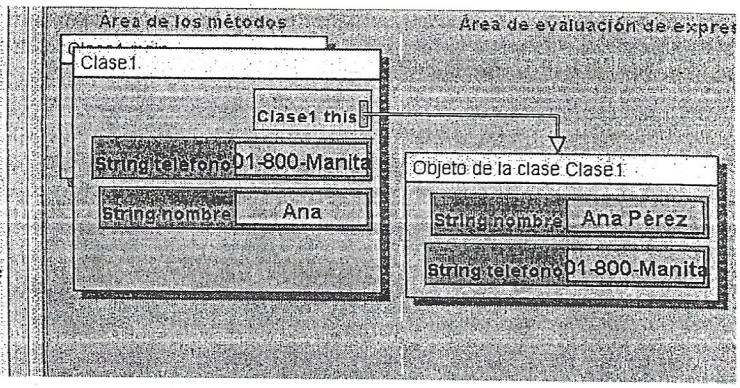


7. Se realiza this.nombre = nombre + " Pérez".
   - La variable "nombre" se busca primero en el registro de activación, donde es encontrada.
   - Si queremos hacer referencia al atributo "nombre" del objeto usamos "this.nombre".
   - En este paso primero se concatena el parámetro "nombre", cuyo valor es "Ana", con el texto " Pérez".
   - El resultado es "Ana Pérez" y se asigna al atributo "nombre" en el objeto.



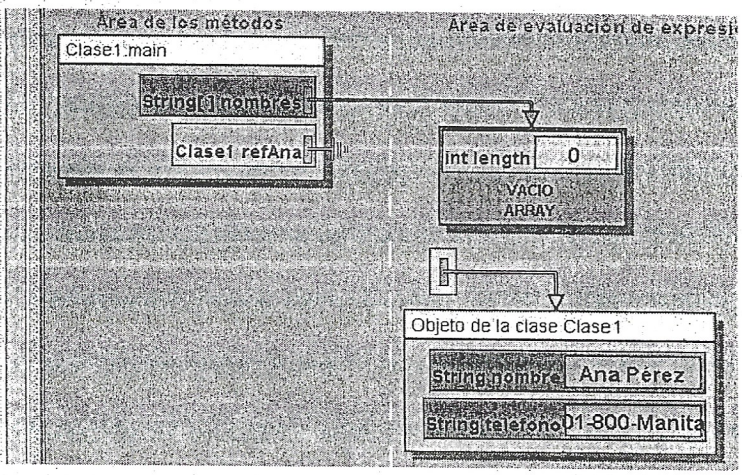
8. Se asigna el valor del parámetro "telefono", que es "01-800-Manita" al atributo "telefono" del objeto.

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



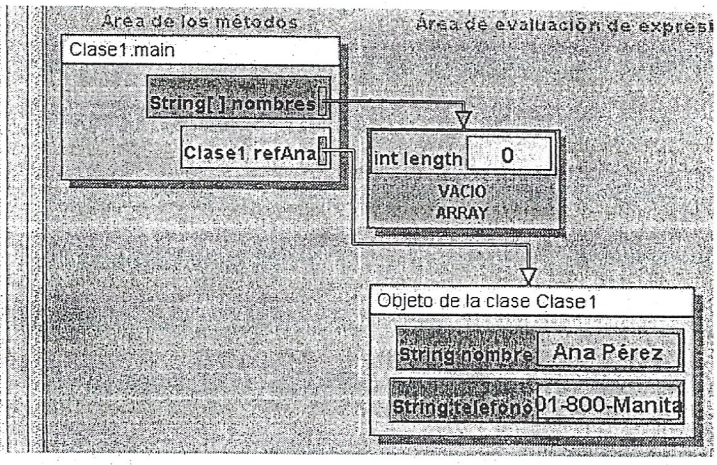
9. Se termina el constructor y se elimina el registro de activación.

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



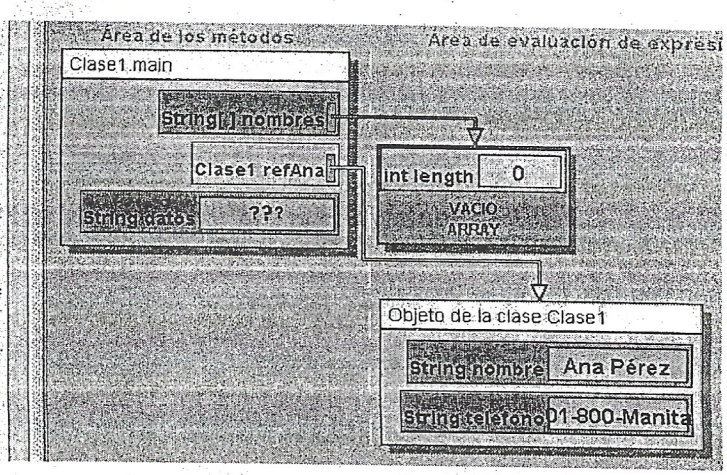
10. El operador "=" hace que "refAna" apunte al objeto recién creado.

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



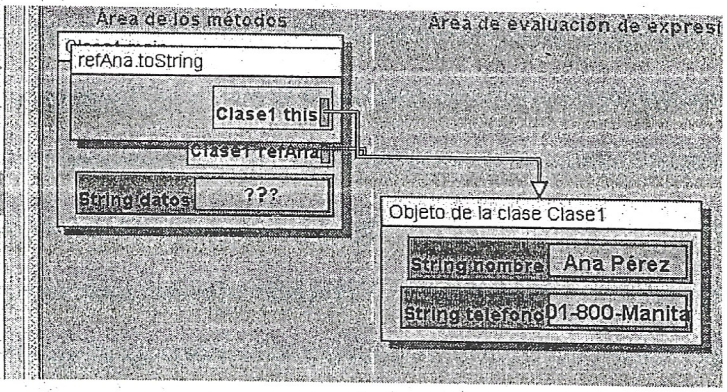
11. Se crea la variable String datos.

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre +", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



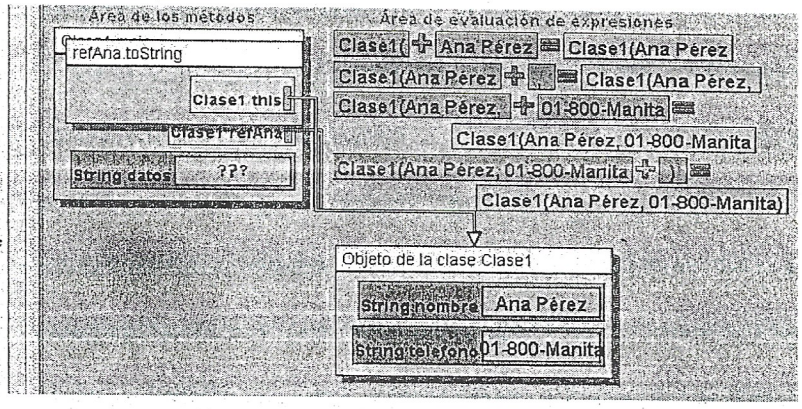
12. Se invoca el método "toString( )". Se crea un registro de acitvación. Se crea la referencia "this", que apunta al objeto a utilizar.

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



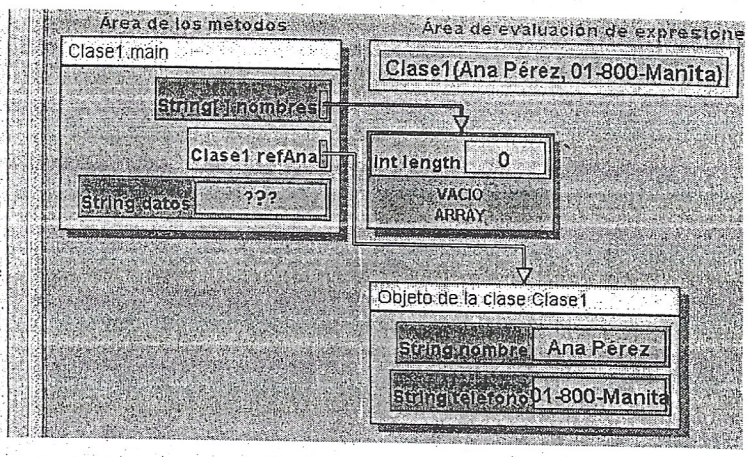
13. Se realiza la instrucción "Clase1(" + nombre + ", " + telefono + ")". Para ubicar el valor de "nombre", se le busca primero en el registro de activación. Pero como no se halla, se busca en el objeto apuntado por "this".

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



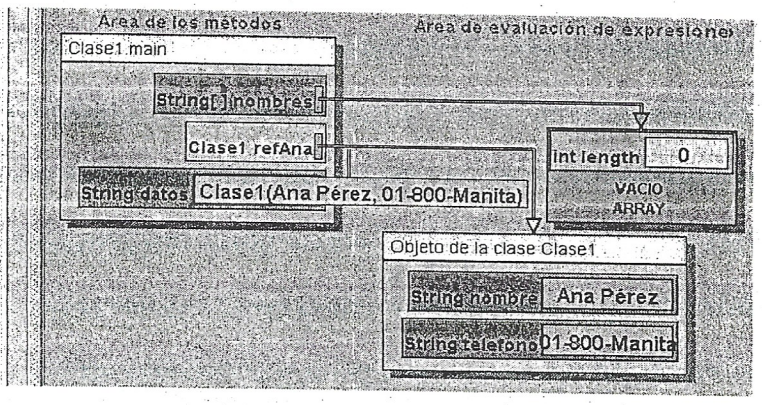
14. Se devuelve el valor calculado y se termina la ejecución del método, por lo cual se elimina el registro de activación.

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



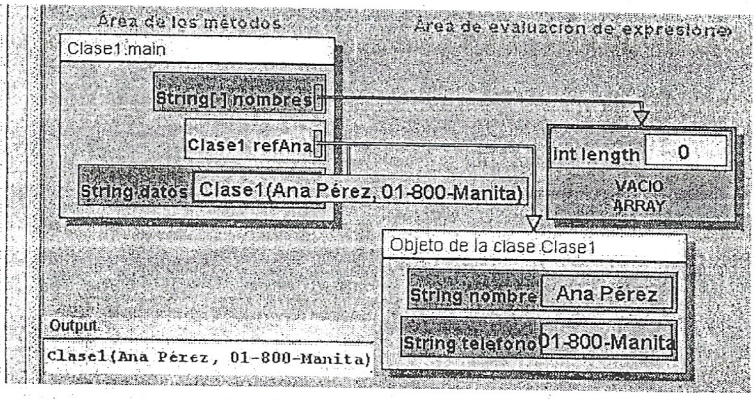
15. El valor devuelto se asigna a la variable "datos".

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



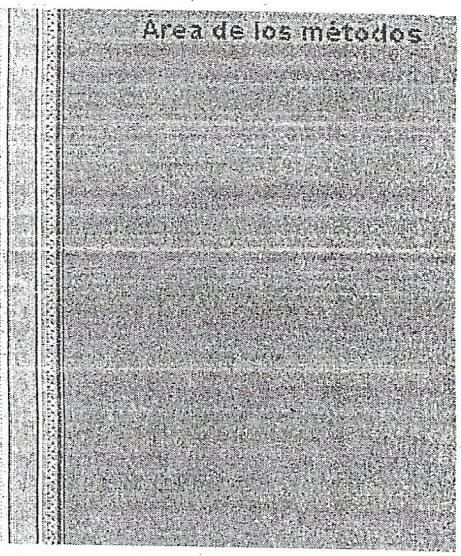
16. Se muestra el valor de "datos".

```
public class Clase1 {
    private String nombre;
    private String telefono;
    public Clase1(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clase1(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clase1 refAna = new Clase1("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



17. Finalmente se elimina el registro de activación y termina la ejecución del programa.

```
public class Clasel {
    private String nombre;
    private String telefono;
    public Clasel(String telefono, String nombre) {
        this.nombre = nombre + " Pérez";
        this.telefono = telefono;
    }
    public String toString() {
        return "Clasel(" + nombre + ", " + telefono + ")";
    }
    public static void main(String[] nombres) {
        Clasel refAna = new Clasel("01-800-Manita", "Ana");
        String datos = refAna.toString();
        System.out.println(datos);
    }
}
```



1.3. Conclusiones.

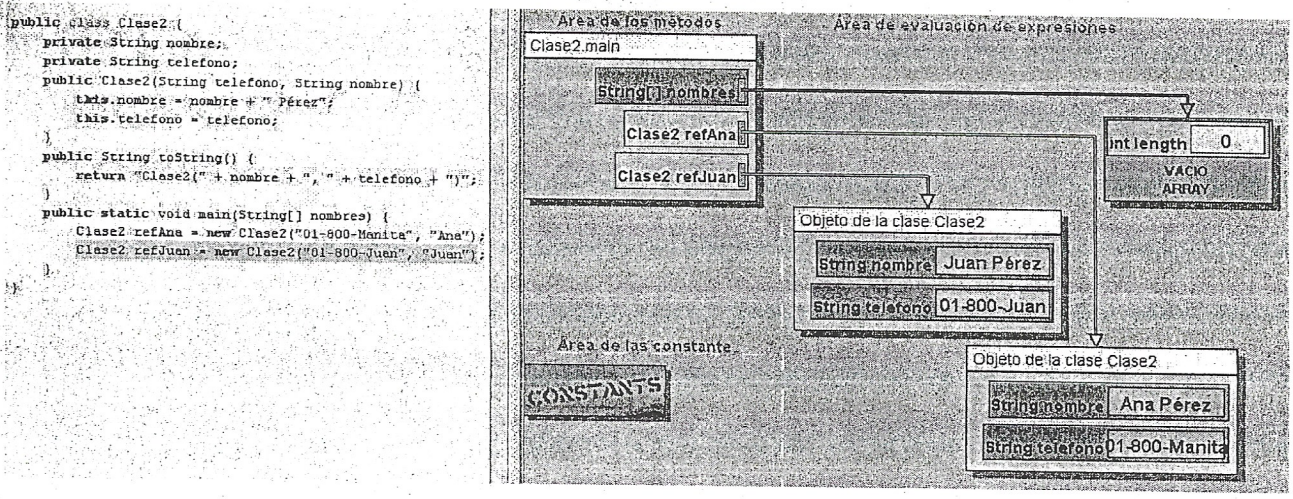
- Cada vez que se invoca un método se crea un registro de activación. En él se incluyen los parámetros recibidos.
- Si el método invocado no es "static", se incluye la referencia "this" en el registro de activación y le apunta al objeto que se está usando.
- Si el método invocado es "static", no se incluye la referencia "this", pues no le pertenece a los objetos.
- Cuando se crea un objeto, primero se le añaden los atributos con los siguientes valores:
  - 0 si es numérico
  - false si es boolean
  - null si es una referencia a un objeto. (Esto incluye el tipo String.)
- Posteriormente a la creación del objeto se invoca el constructor.

2. Programas con más de un Objeto.

2.1. Código de Clase2.java

```
1  public class Clase2 {
2      private String nombre;
3      private String telefono;
4      public Clase2(String telefono, String nombre) {
5          this.nombre = nombre + " Pérez";
6          this.telefono = telefono;
7      }
8      public String toString() {
9          return "Clase2(" + nombre + ", " + telefono + ")";
10     }
11     public static void main(String[] nombres) {
12         Clase2 refAna = new Clase2("01-800-Manita", "Ana");
13         Clase2 refJuan = new Clase2("01-800-Juan", "Juan");
14     }
15 }
```

2.2. Objetos Creados al Ejecutar Clase2.java



2.3. Conclusiones.

- Cada objeto tiene su propio juego de atributos, con sus propios valores.