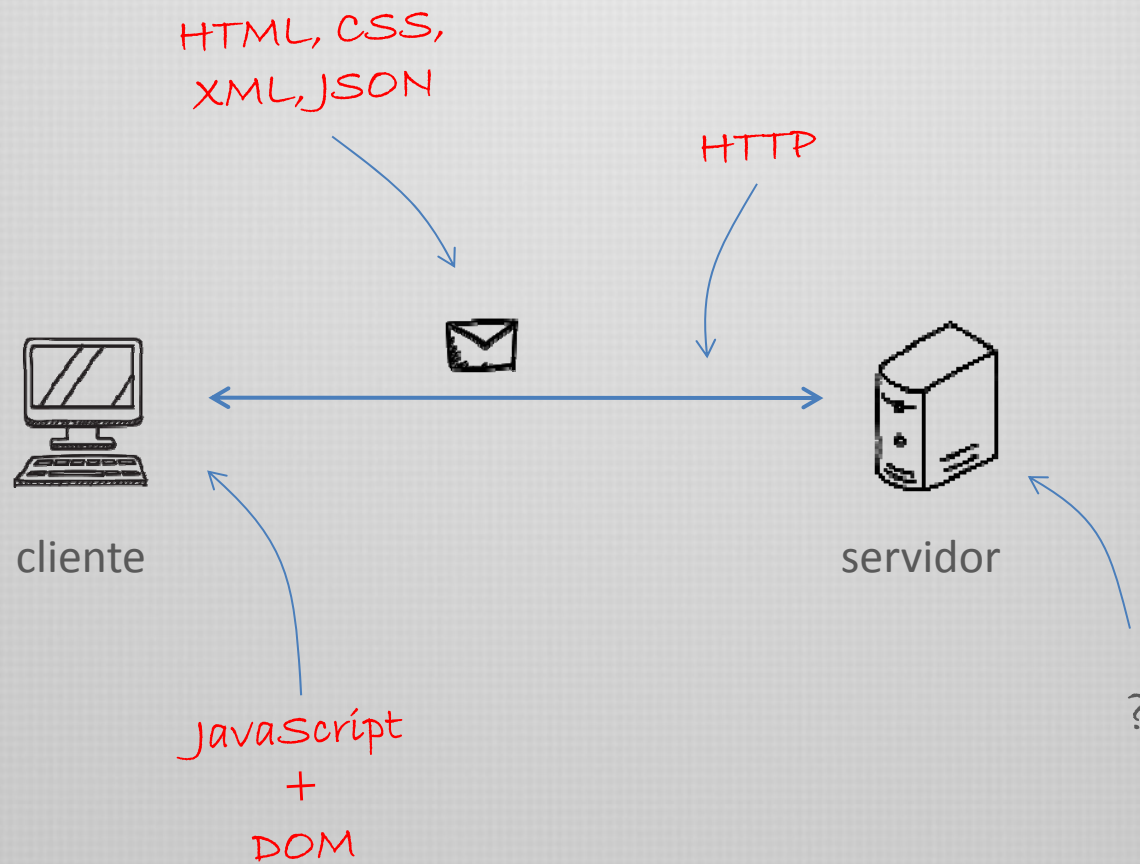


# Ingeniería de Aplicaciones Web

*Diego C. Martínez*

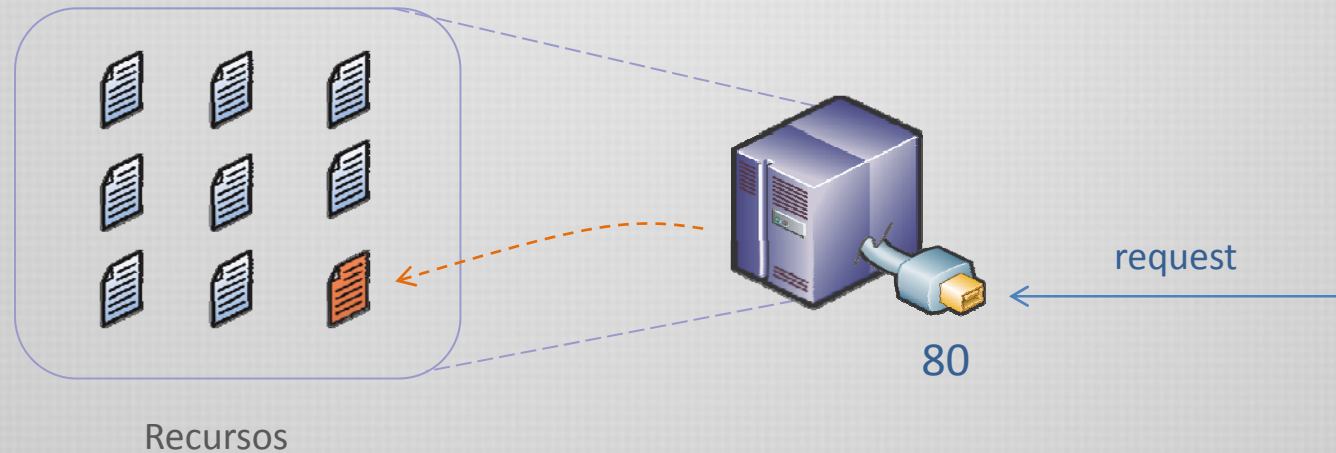
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur

# Tecnologías web



# Servidores Web

En su expresión más simple, el servidor es básicamente un **servidor de recursos**:

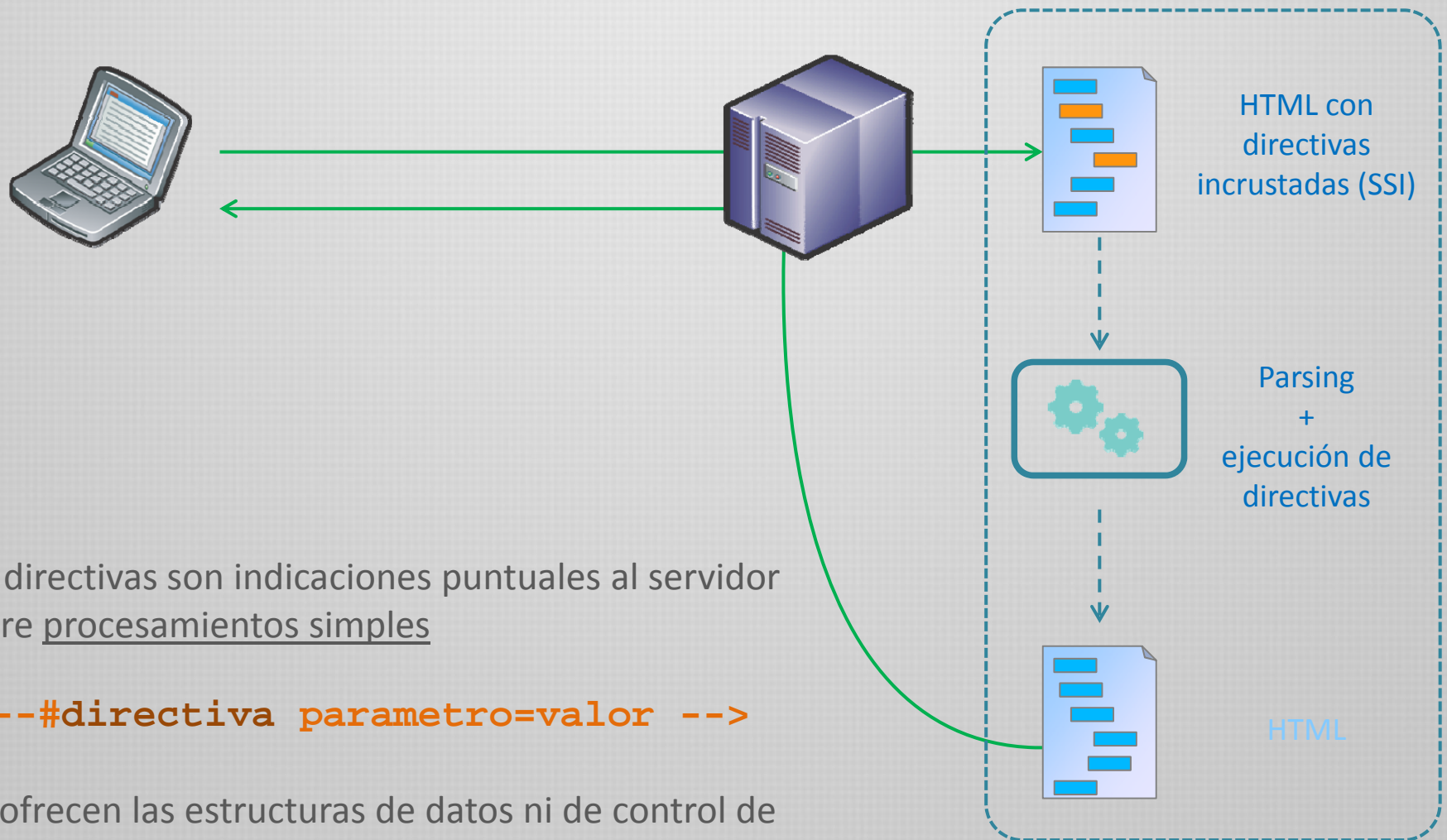


Pero siguiendo este esquema, el origen de la respuesta siempre es **estático**.

*La web en sus orígenes : "page replacement"*

Sin embargo, en muchos casos es deseable consultar información que es fruto de un procesamiento previo: consultas a bases de datos, respuestas a formularios, consultas a otros sistemas etc. .

## Procesando una respuesta – Server Side Includes



Las directivas son indicaciones puntuales al servidor sobre procesamientos simples

```
<!--#directiva parametro=valor -->
```

No ofrecen las estructuras de datos ni de control de un lenguaje de alto nivel.

Archivo **.shtml**



# Server Side Includes

Algunas directivas comunes:

```
<!--#config timefmt="%A %B %d, %Y" -->
```

```
Hoy es <!--#echo var="DATE_LOCAL" -->
```

---

```
Ultima modificacion: <!--#flastmod file="index.html" -->
```

---

```
<!--#include virtual="/footer.html" -->
```

---

```
<!--#printenv -->
```

---

```
<!--#set var="cost" value="\$100" -->
```

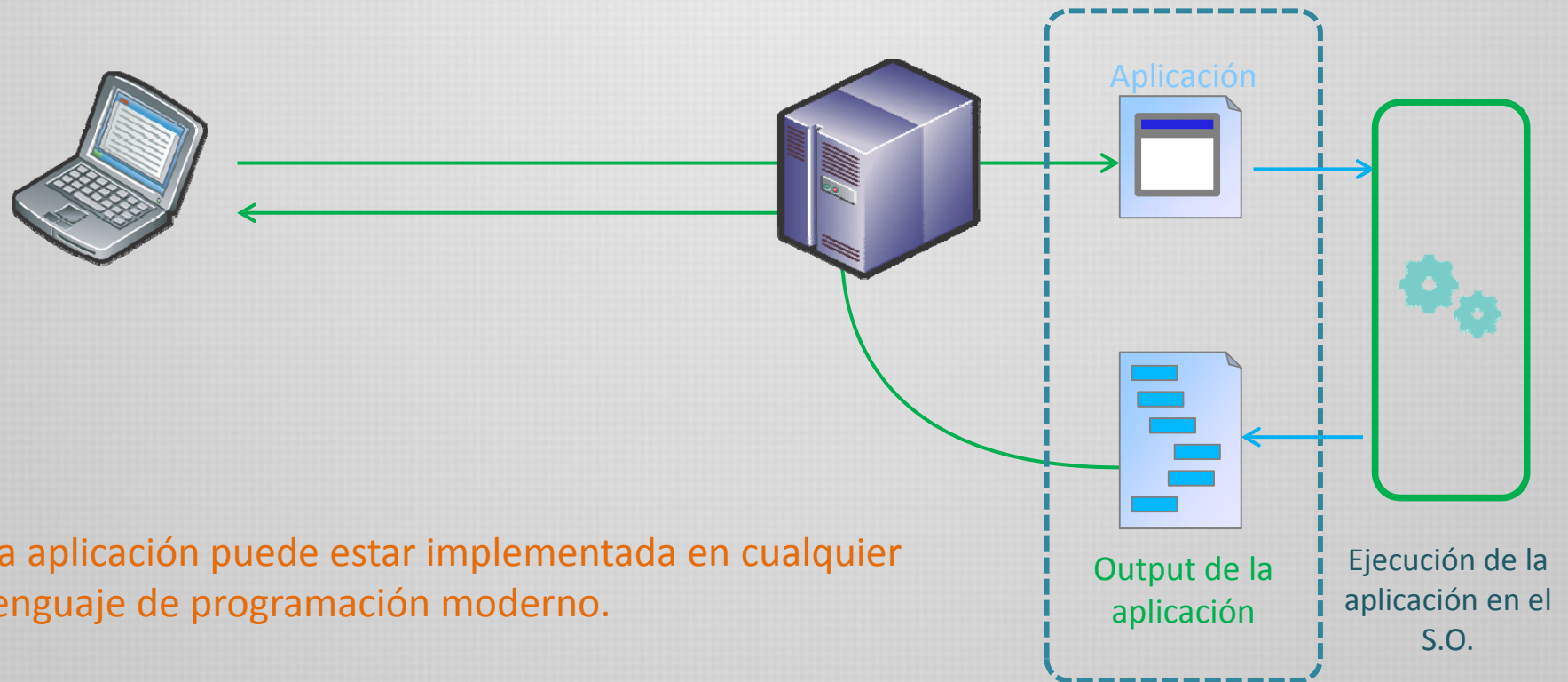
```
<!--#if expr="test_condition" -->
```

```
<!--#elif expr="test_condition" -->
```

```
<!--#else -->
```

```
<!--#endif -->
```

## Procesando una respuesta – Aplicaciones CGI



La aplicación puede estar implementada en cualquier lenguaje de programación moderno.

La integración con el servidor es simple pero rústica, por medio de la entrada-salida estándar :(

La aplicación no necesariamente está basada en tecnologías para la web o Internet.

# CGI Scripts

**CGI (Common Gateway Interface)** es un estándar que establece una interfaz entre aplicaciones externas y servidores de información, como HTTP o FTP.

*Básicamente es un mecanismo que permite a todo el mundo ejecutar un programa determinado, residente en un servidor de información. Por este hecho se les suele denominar aplicación CGI, o scripts CGI.*



Rob McCool

*Esto requiere algunas consideraciones de seguridad. Por ejemplo, usualmente las aplicaciones a ejecutarse se mantienen en directorios fuera del alcance normal.*

Como es un estándar de interfaz, cualquier programa puede ejecutarse como aplicación CGI: puede ser implementado en C, C++, Fortran, PERL, Unix Shell, etc.

No utiliza la línea de comandos para recibir información, sino variables de entorno...

**QUERY\_STRING** es todo lo que sigue luego del primer ? del URL.

**PATH\_INFO** es la ruta para información adicional que puede requerir la aplicación.



# CGI Scripts - Perl

Uno de los lenguajes más populares para la implementación de aplicaciones CGI es Perl, creado en 1987 por Larry Wall.

El nombre Perl es una derivación de *pearl* (perla en inglés).

a.k.a PERL -- Practical Extraction and Report Language



Es un lenguaje de programación dinámico, y el que mayoritariamente contribuyó al título de *scripts* que suelen tener las aplicaciones CGI.

<http://www.perl.org/>

La sintaxis es similar a la del lenguaje C, pero con claras diferencias por la necesidad de ser dinámico.



## CGI – Perl - ejemplos

### HTML

```
<form method="post" action="http://localhost/cgi-bin/mostrar.pl" >  
  Texto: <input type="text" name="mitexto"><br>  
  <input type="submit" value="Enviar">  
  <input type="reset" value="Limpiar">  
</form>
```

Script CGI  
(PERL)

```
#!/c:\Program Files\Perl\bin\perl.exe  
$inputlength=$ENV{'CONTENT_LENGTH'};  
read(STDIN,$formdata,$inputlength);  
($name,$value)=split /=/, $formdata;  
print "Content-type: text/html\n";  
print "\n";  
print "  
<HTML>  
<HEAD>  
<TITLE>Resultados:</TITLE>  
</HEAD>  
<BODY>  
<H1>Esto es lo que se ha ingresado:</H1>  
<BR>  
$name = $value  
</BODY>  
</HTML>";
```

## Procesando una respuesta – Scripting

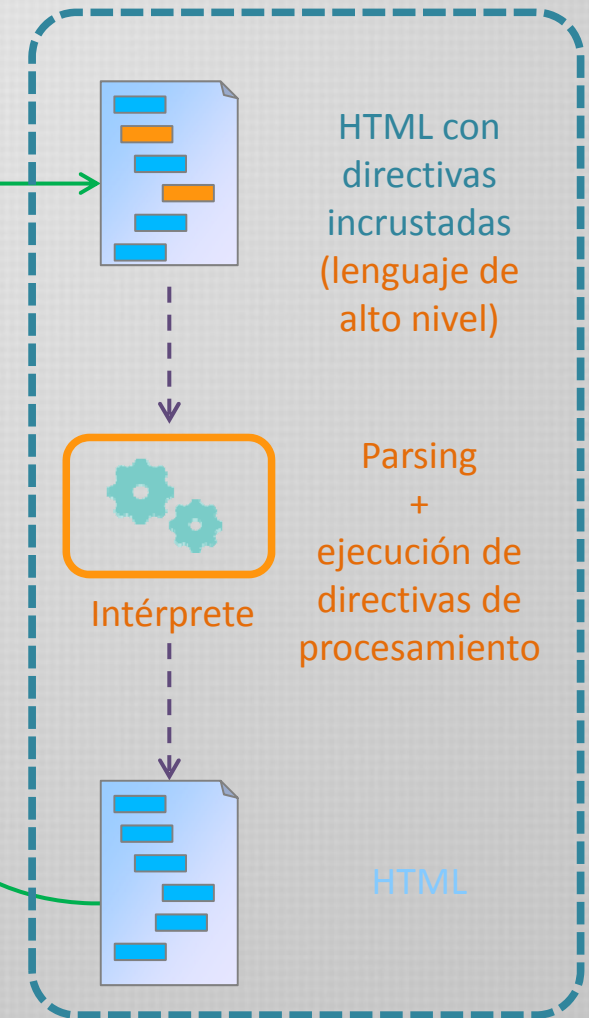


El servidor es asociado con un intérprete de un lenguaje de alto nivel.

El lenguaje de script puede utilizarse directamente en los documentos HTML

Las directivas de procesamiento son ahora segmentos de código.

Se dispone de estructuras de control y datos de un lenguaje de alto nivel.  
Mayor poder de procesamiento.



# PHP



El lenguaje PHP surge en 1995 bajo el nombre PHP/FI, creado por Rasmus Lerdorf.

Originalmente Rasmus creó el lenguaje como reemplazo de unos scripts Perl que había hecho para su página personal.

PHP significaba “*Personal Home Page*” y FI “*Form Interpreter*”.



Rasmus Lerdorf

Luego de liberar PHP, junto con Zeev Suraski y Andi Gutmans reescriben el intérprete y nace PHP3 en 1997, siendo testado publicamente en 1998.

En 1999 reescriben el núcleo de PHP para obtener Zend Engine.

En 2004 sale PHP5, con varias modificaciones, entre ellas un mejor modelo de objetos, deficiente en la versión anterior.

Hoy es usado por muchos sitios gracias al desarrollo de varios productos, la mayoría GNU: [phpBB](#), [WordPress](#), [MediaWiki](#), [Joomla](#), [Mambo](#), etc.

# PHP

La sintaxis y las características orientadas a objetos del lenguaje PHP(5) son fáciles de aprender, pues guardan similitud con lenguajes vistos anteriormente.

```
if (cond)
{ ... }
else
{ ... }

while (cond) {
...
}

do{
...
} while (cond)

for ($i=1; $i<=5; $i++)
{
...
}
```

PHP es *inscrutable* en HTML, similar a JavaScript  
Todo código restante HTML se envía tal cual al cliente.

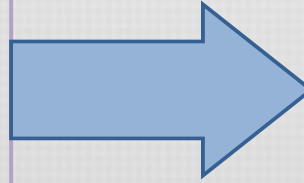
Tag inicial	Tag final
<?php	?>
<?	?>
<script language="php">	?>
<%	%>



# PHP

```
<html>
<head>
  <title>Ejemplo</title>
</head>
<body>
  <?php
    echo "Hola Mundo!";
  ?>
</body>
</html>
```

**hola.php**



```
<html>
<head>
  <title>Ejemplo</title>
</head>
<body>
  Hola Mundo!
</body>
</html>
```

Resultado recibido por el  
cliente

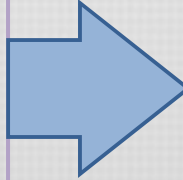
# PHP

El script .php no necesariamente debe incluir código HTML...

```
<?php
echo "<HTML>";
echo "<TITLE>Ejemplo</TITLE>";
echo "<BODY>";
echo "Hola Mundo!";
if date('w')=5
    { echo "Hoy es sabado"; }
else
    { echo "Hoy no es sabado"; }
echo "</BODY></HTML>";

?>
```

todophp.php




```
<HTML>
<TITLE>Ejemplo</TITLE>
<BODY>
Hola Mundo! Hoy no es sabado
</BODY>
</HTML>
```

Resultado  
recibido por el  
cliente

# PHP

Pueden implementarse funciones...

*Las variables  
comienzan con \$*



```
<?php
function cuadrado($num)
{
    return $num*$num;
}

$cartel1 = "El cuadrado de ";
$cartel2 = " es ";
$numero=4;

echo $cartel1;
echo $numero;
echo $cartel2;
echo cuadrado($numero);

echo "<BR>";

$numero=10;
echo $cartel1.$numero.$cartel2.cuadrado($numero);

?>
```

# PHP - Modelo de objetos

visibilidad

constructor

```
<?php

class MiClase {
    public $v;

    public function __construct(){
        $this->v="Hola"
    }

    public function mostrar() {
        echo $this->v;
    }

    public function getV(){
        return $this->v;
    }

}

?>
```



# Java

Java es un lenguaje de programación orientado a objetos, creado en 1990 por *Sun Microsystems*.



Su escenario de trabajo es mayoritariamente el de las aplicaciones distribuídas y la programación en red.

Sus premisas iniciales son:

*Uso de la metodología OO*

*Ejecutar un mismo programa en diferentes SO*

*Soporte para redes de computadoras*

*Diseñado para ejecutar código remotamente con seguridad*

*Fácil de usar*



Programas Java:

*Applets* : se ejecutan en browsers, incrustados en páginas web.

*Applications* : se ejecutan directamente sobre la Java Virtual Machine.

*Servlets*: se ejecutan en un servidor que posee un Contenedor Web.

*Midlets*: se ejecutan en la JVM para dispositivos móviles.

*Cada cual posee una estructura particular, con ciertos métodos específicos.*

# Java 2 Platform

