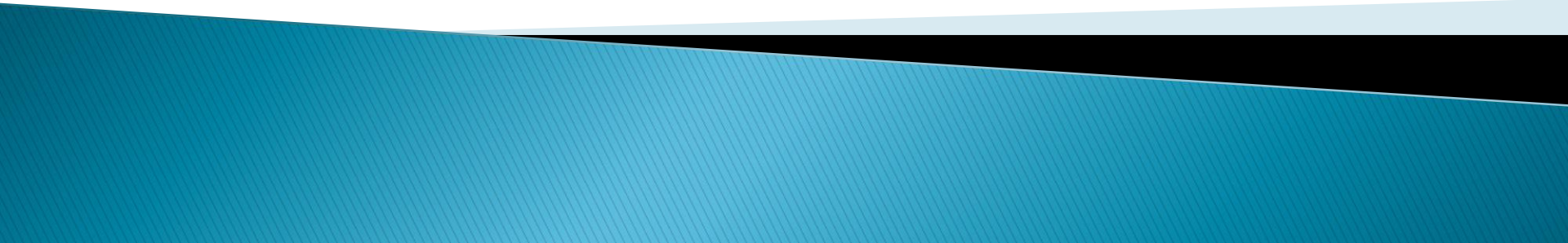


Mantenimiento de datos en la aplicación



Mantenimiento de datos

➤ Compartir datos entre los componentes de una aplicación:

- Atributos de petición

- Atributos de sesión

- Atributos de aplicación

- Cookies

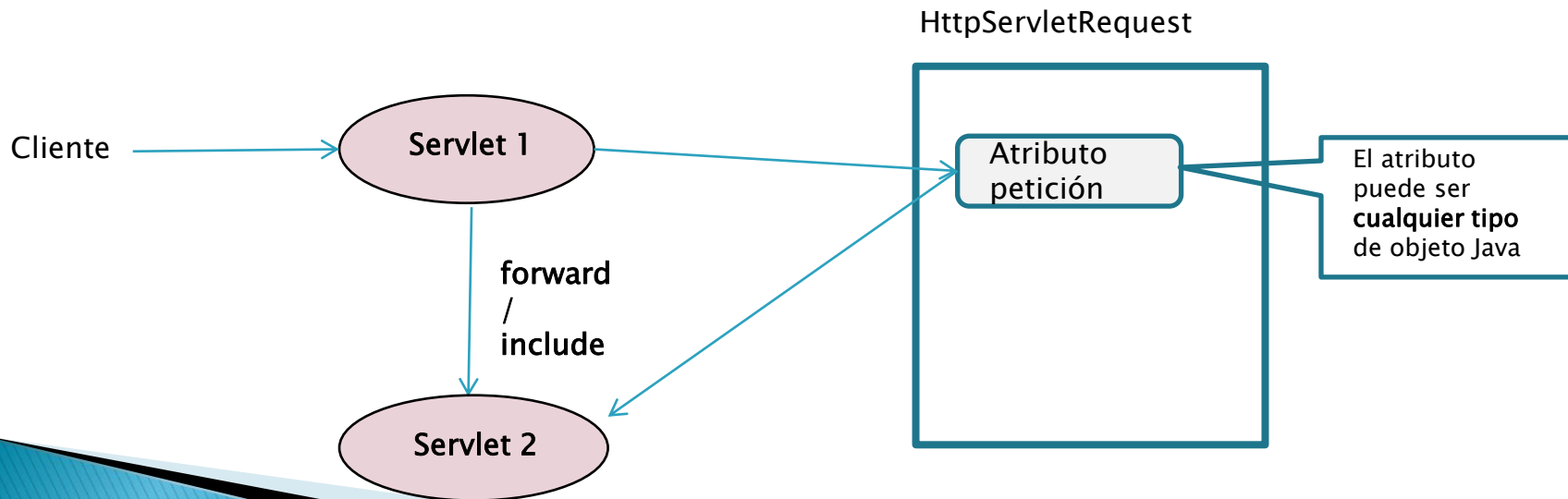
} Memoria del servidor de aplicaciones

} Disco duro del cliente

Atributos de petición

Definición

- Permite almacenar datos que son compartidos entre todos los servlets que se ejecutan en la misma petición
- Se almacenan en el objeto HttpServletRequest



Acceso a atributos

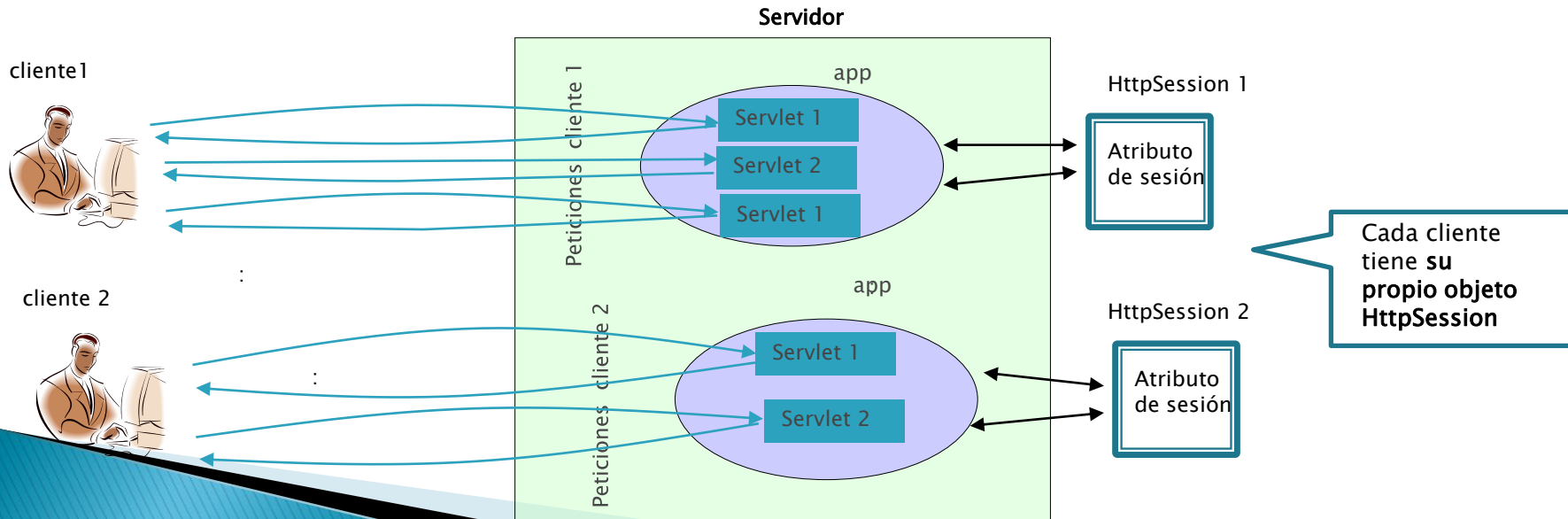
➤ Para establecer y recuperar atributos de petición, se emplean los siguientes métodos de `HttpServletRequest`:

- `void setAttribute(String nombre, Object value)`. Almacena un atributo con el nombre especificado en el primer parámetro y cuyo valor se indica en el segundo
- `Object getAttribute(String nombre)`. Devuelve el valor del atributo cuyo nombre se indica. Si no existe, devuelve null

Atributos de sesión

Características

- Permite almacenar datos que son compartidos entre todos los componentes de la aplicación durante la sesión de usuario
- Se almacenan en el objeto HttpSession



Acceso a atributos

- Se debe obtener el objeto `HttpSession` del usuario mediante el método *`getSession()`* de `HttpServletRequest`:

```
//Si la sesión de usuario ya existe, se  
//recupera, sino se crea una nueva  
HttpSession s=request.getSession();
```

- Para establecer y recuperar atributos de petición, utilizaremos los métodos *`setAttribute()`* y *`getAttribute()`* de `HttpSession`.

Control de sesiones

➤ Tiempo máximo de inactividad de una sesión:

```
<session-config>  
    <session-timeout>10</session-timeout>  
</session-config>
```

web.xml

➤ Eliminar atributos de sesión:

- método `removeAttribute(String nombre)` de `HttpSession`

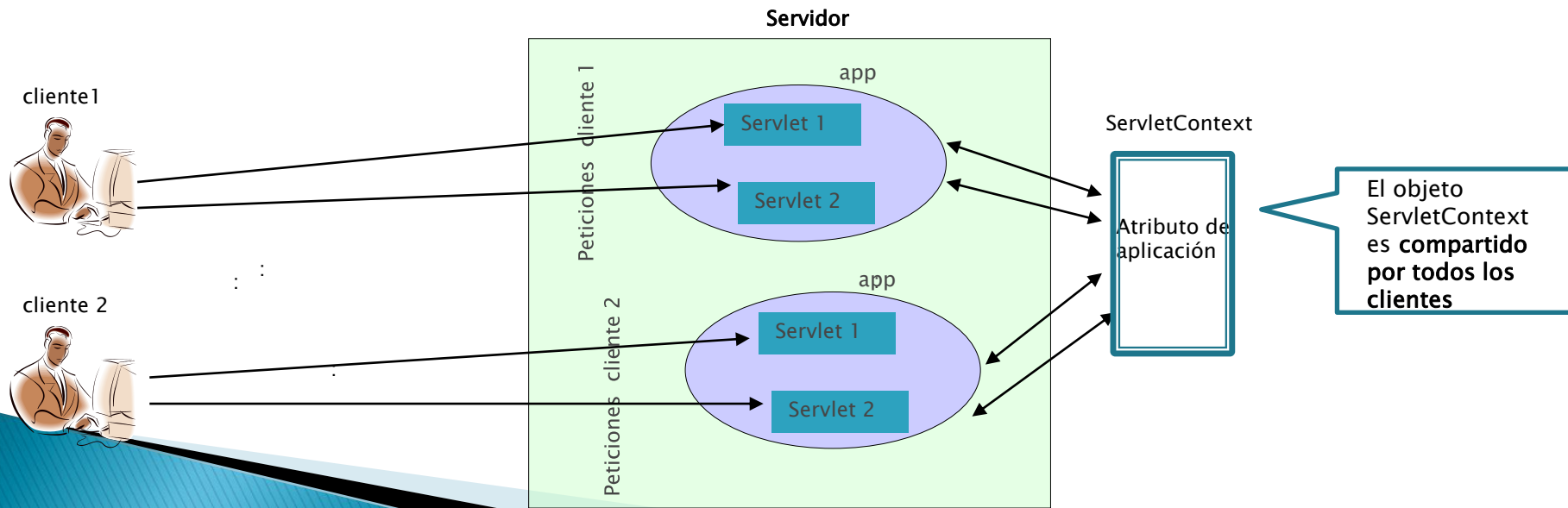
➤ Invalidar una sesión:

- método `invalidate()` de `HttpSession`

Atributos de aplicación

Características

- Permite almacenar datos que son compartidos entre todos los usuarios de la aplicación
- Se almacenan en el objeto ServletContext



Acceso a atributos

- Se debe obtener el objeto `ServletContext` mediante el método *`getServletContext()`* de `HttpServletRequest`:

```
//Obtiene el objeto ServletContext  
//que es el mismo para todos los usuarios  
ServletContext context=request.getServletContext();
```


- Para establecer y recuperar atributos de petición, utilizaremos los métodos *`setAttribute()`* y *`getAttribute()`* de `ServletContext`.

Control de atributos de aplicación

- No hay un timeout. El objeto Servlet Context se destruye al detener la aplicación
- Eliminar atributos de aplicación:
 - método `removeAttribute(String nombre)` de `HttpSession`

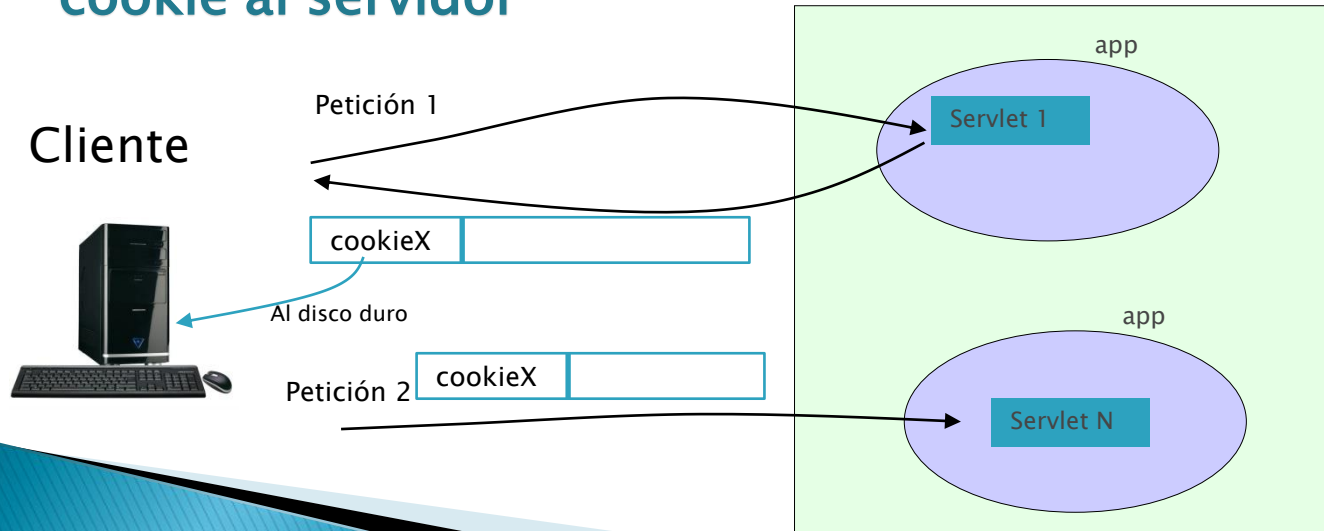
Cookies

Características

- Datos creados por la aplicación y que se almacenan en el disco duro del cliente
 - Solo admite cadenas de caracteres
 - El tiempo de vida es limitado
 - Se manejan a través de `javax.servlet.http.Cookie`
- 

Funcionamiento

- La cookie es creada por el servidor y enviada al cliente en la cabecera de la respuesta.
- En posteriores conexiones, el cliente envía de nuevo la cookie al servidor



Creación de una cookie

1. Creación del objeto Cookie:

```
Cookie ck=new Cookie("codigo","E27");
```

2. Establecimiento de periodo de vida:

```
ck.setMaxAge(18000);
```

3. Añadir cookie a la respuesta:

```
response.addCookie(ck);
```



Recuperación de una cookie

1. Obtención de array de cookies:

```
Cookie[] cookies=request.getCookies();
```

2. Recorrido y búsqueda:

```
for(Cookie ck:cookies){  
    if(ck.getName().equals("codigo")){  
        String valor=ck.getValue();  
    }  
}
```