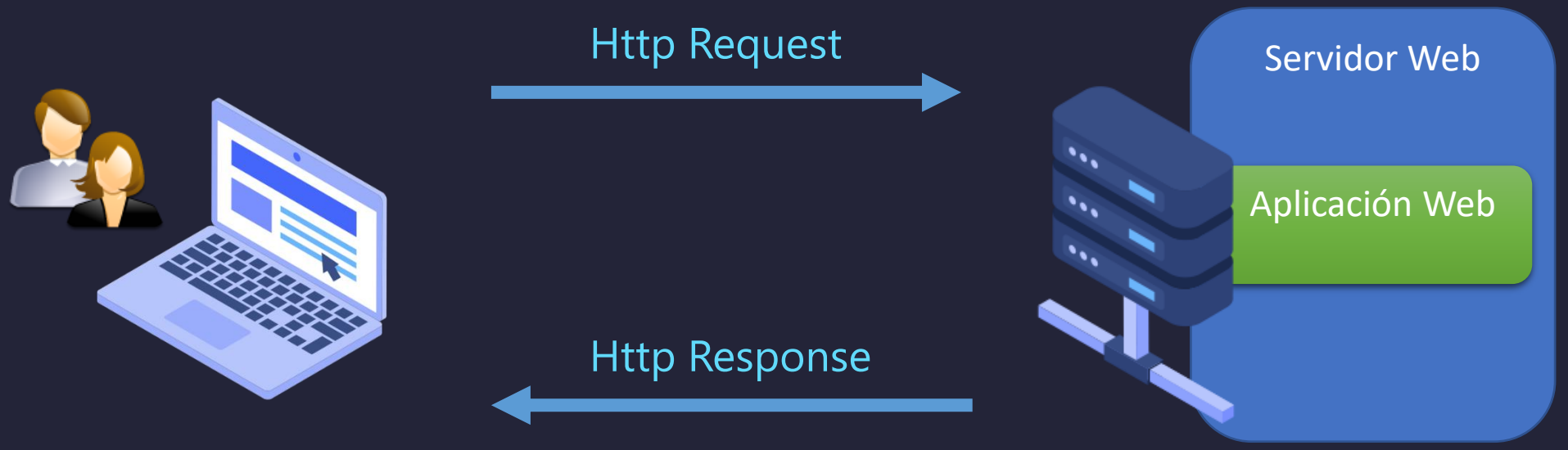
Qué es HTTP

El Protocolo de transferencia de hipertexto permite la comunicación entre un cliente, normalmente un navegador web y un servidor web para compartir datos en diferentes formatos, como html, json, imágenes, videos, pdf, Excel etc) en la World Wide Web.



Conceptos

Métodos de petición HTTP

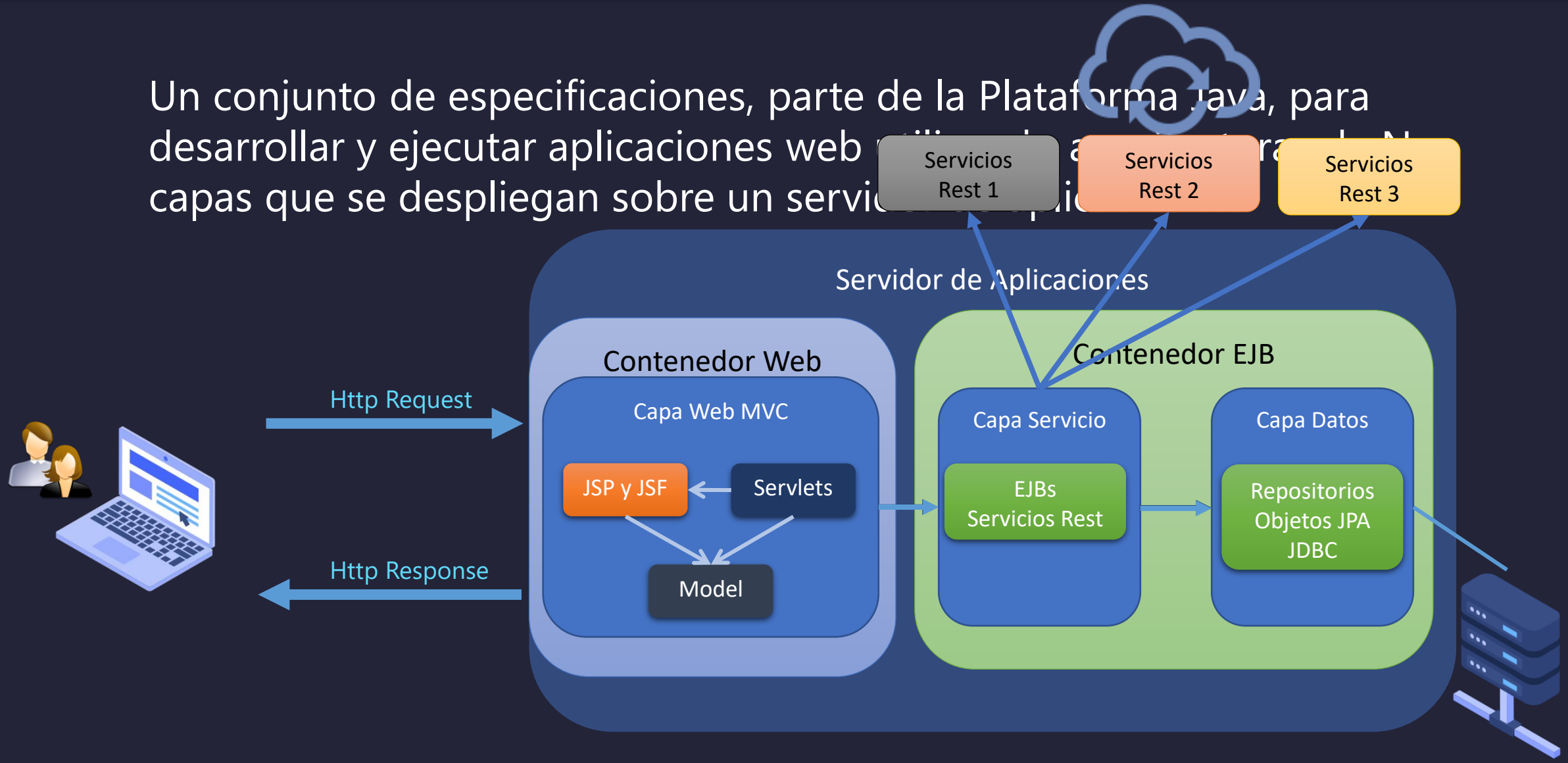
Cabeceras HTTP de petición (request)

Cabeceras HTTP de respuesta (response)

Códigos de estado de respuesta HTTP

¿Qué es Java EE?

Un conjunto de especificaciones, parte de la Plataforma Java, para desarrollar y ejecutar aplicaciones web [illegible] capas que se despliegan sobre un servi[illegible]



JAKARTA EE

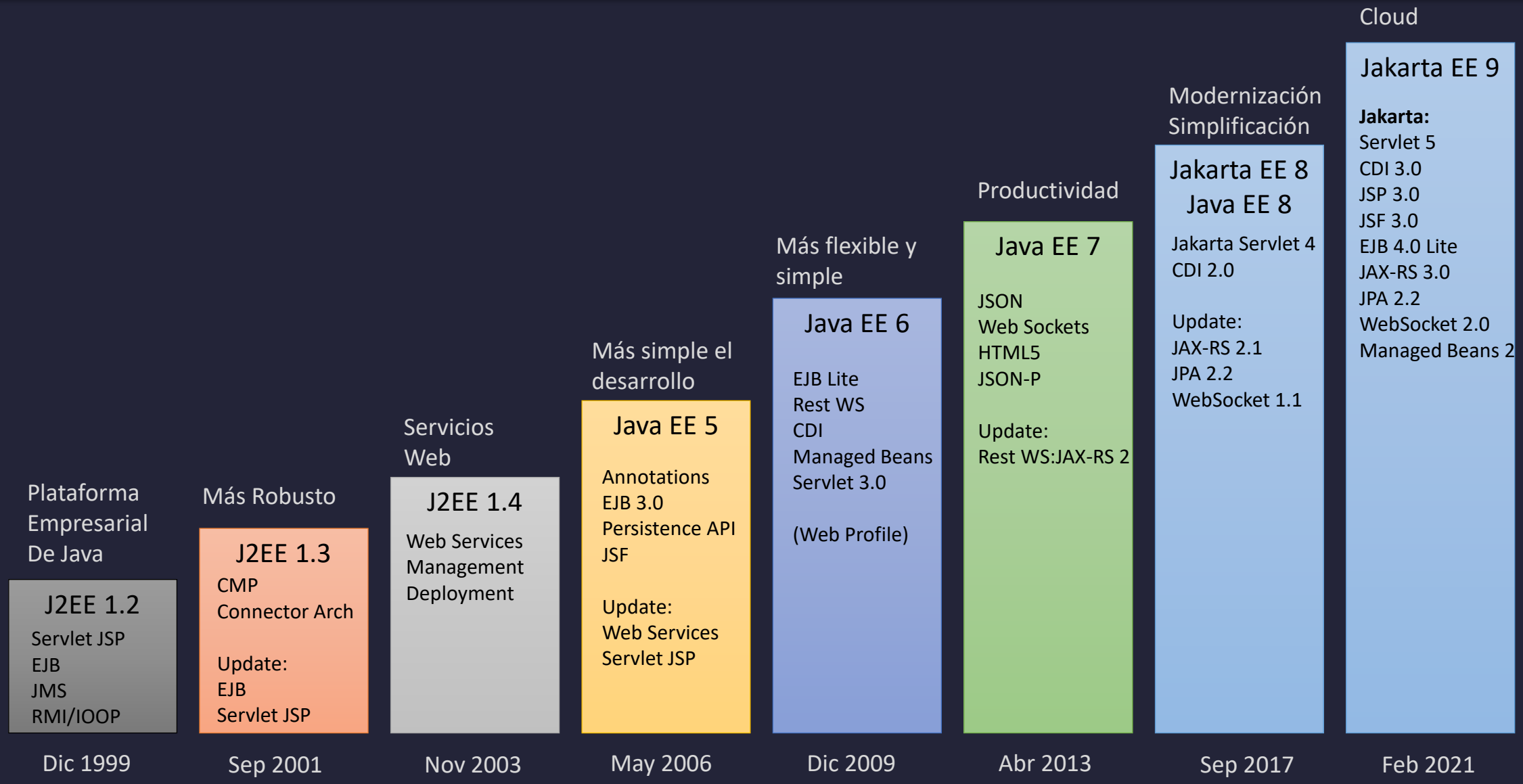
Características

- Componentes web
- EJB 4.0
- Interoperabilidad
- Comunicación remota
- Control de la concurrencia
- Seguridad
- Transaccionalidad
- Eventos utilizando JMS (Java Messaging Service)
- Persistencia JPA 3.0
- Servicios de nombres y de directorio JNDI
- Inyección de Dependencia CDI 3.0
- JAX-WS SOAP
- JAX-RS Rest

Especificaciones Jakarta EE9

- Jakarta Servlet 5.0
- Jakarta Server Pages 3.0
- Jakarta Server Faces 3.0
- Jakarta Contexts and Dependency Injection 3.0
- Jakarta Managed Beans 2.0
- Jakarta Persistence 3.0
- Jakarta Transactions 2.0
- Jakarta Enterprise Java Beans 4.0 Lite
- Jakarta RESTFul Web Services 3.0
- Jakarta MVC
- Jakarta Messaging
- Jakarta Authentication y Authorization 2.0
- Jakarta Mail
- …

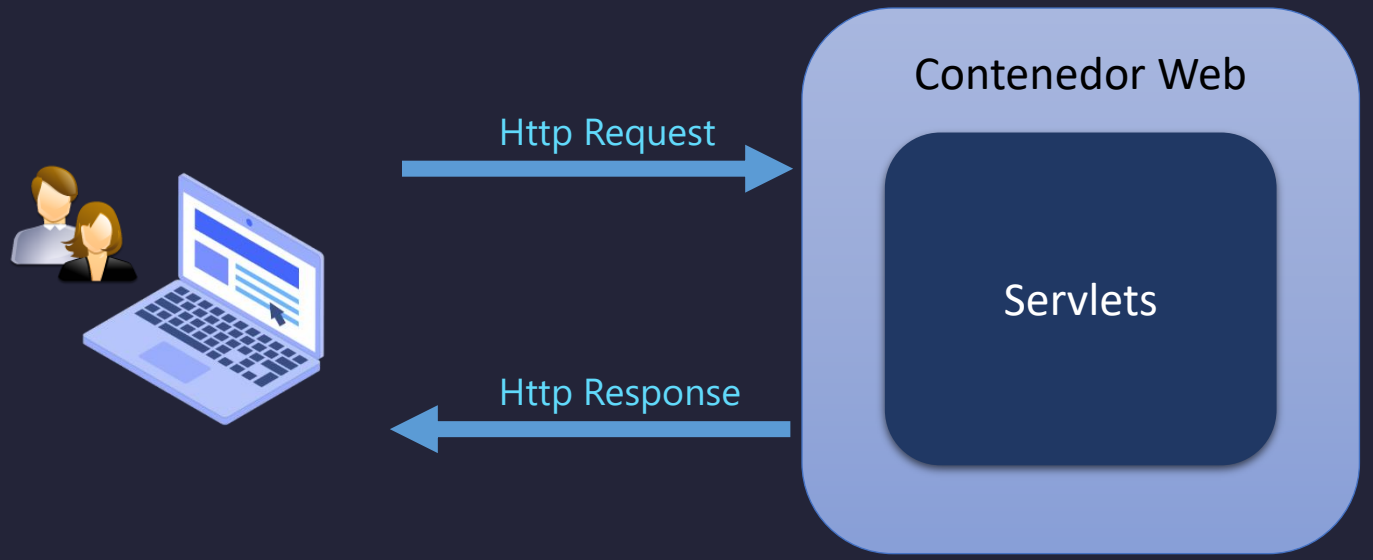
Evolución


Plataforma Empresarial De Java
J2EE 1.2
Servlet JSP
EJB
JMS
RMI/IOOP
Dic 1999
Más Robusto
J2EE 1.3
CMP
Connector Arch
Update:
EJB
Servlet JSP
Sep 2001
Servicios Web
J2EE 1.4
Web Services
Management
Deployment
Nov 2003
Más simple el desarrollo
Java EE 5
Annotations
EJB 3.0
Persistence API
JSF
Update:
Web Services
Servlet JSP
May 2006
Más flexible y simple
Java EE 6
EJB Lite
Rest WS
CDI
Managed Beans
Servlet 3.0
(Web Profile)
Dic 2009
Productividad
Java EE 7
JSON
Web Sockets
HTML5
JSON-P
Update:
Rest WS:JAX-RS 2
Abr 2013
Modernización Simplificación
Jakarta EE 8
Java EE 8
Jakarta Servlet 4
CDI 2.0
Update:
JAX-RS 2.1
JPA 2.2
WebSocket 1.1
Sep 2017
Cloud
Jakarta EE 9
Jakarta:
Servlet 5
CDI 3.0
JSP 3.0
JSF 3.0
EJB 4.0 Lite
JAX-RS 3.0
JPA 2.2
WebSocket 2.0
Managed Beans 2
Feb 2021

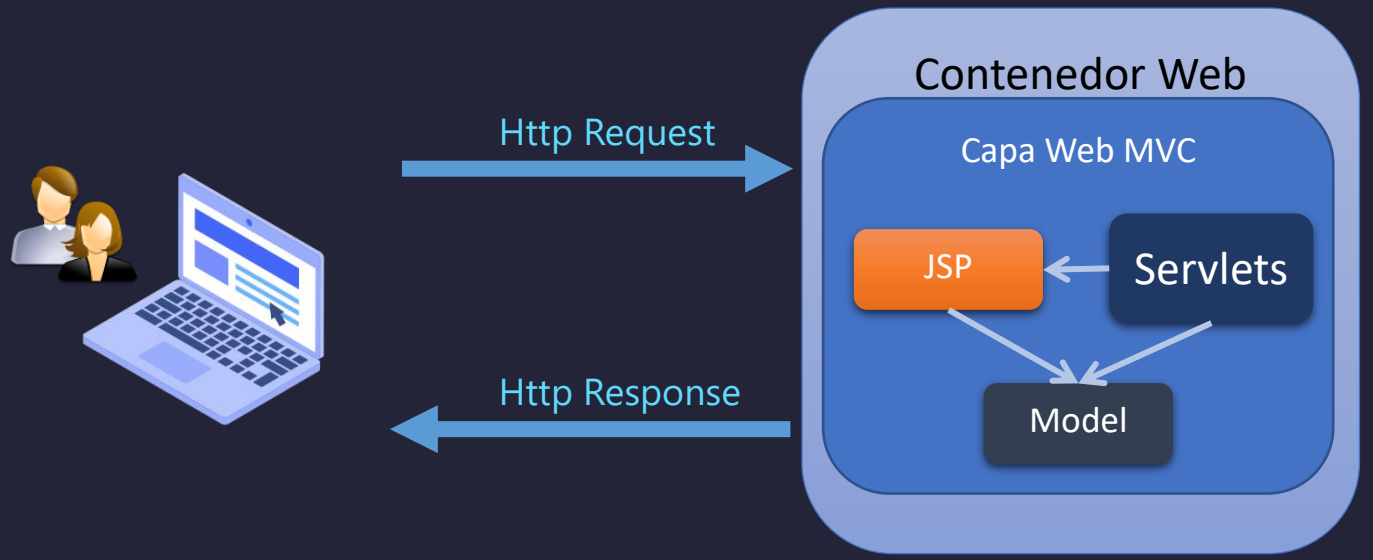
JAKARTA EE

¿Qué es un Servlet?

Es una clase y objeto Java utilizado para implementar una pagina web dinámica con características HTTP de petición y respuesta.



Patrón MVC


Contenedor Web
Capa Web MVC
JSP
Servlets
Model
Http Request
Http Response

JAKARTA EE

Request y Response

¿Qué es un request (petición web)?

- Información que es enviada desde un cliente hacia el servidor
- Datos ingresados e enviados por el usuario
- Método Http (Get o Post)
- Cabeceras HTTP (headers)

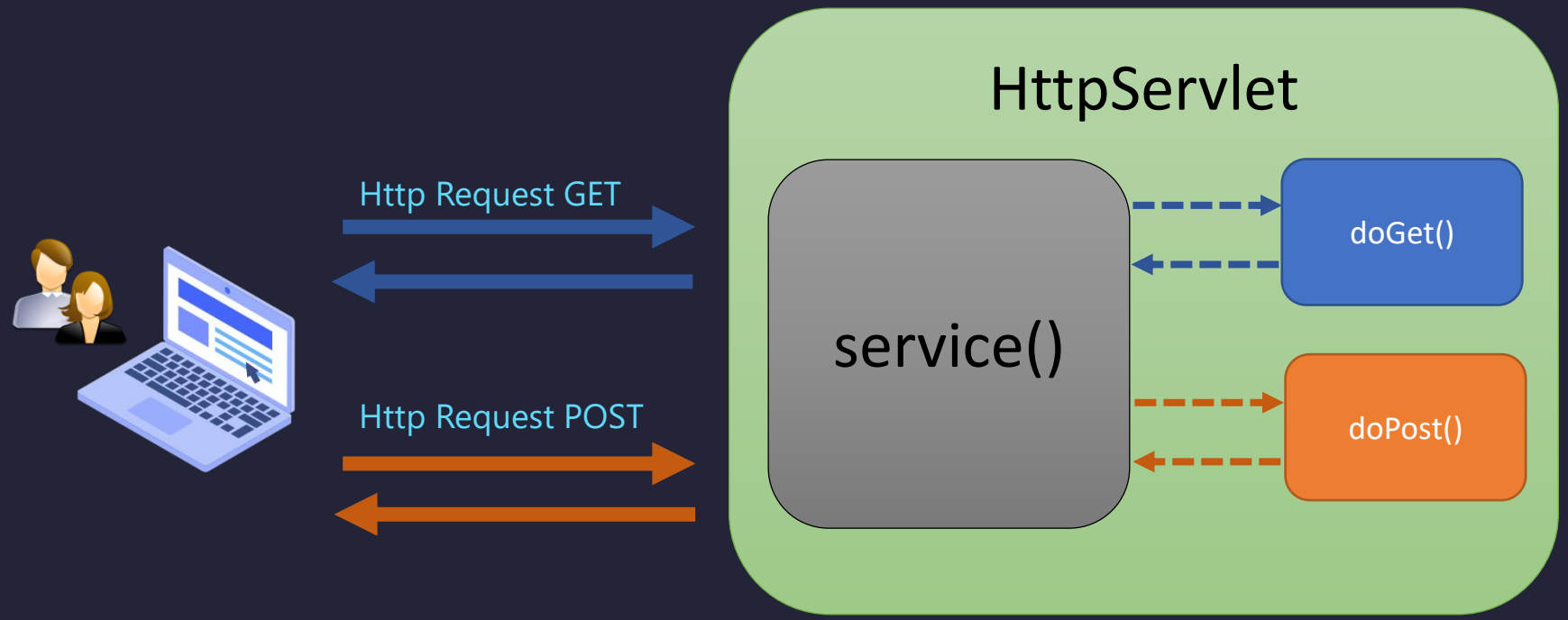
¿Que es un response (respuesta web)?

- Información que es enviada al cliente desde el servidor
- Texto(html, plain, json, xml) o datos binarios (imágenes, pdf, videos)
- HTTP headers, cookies, etc

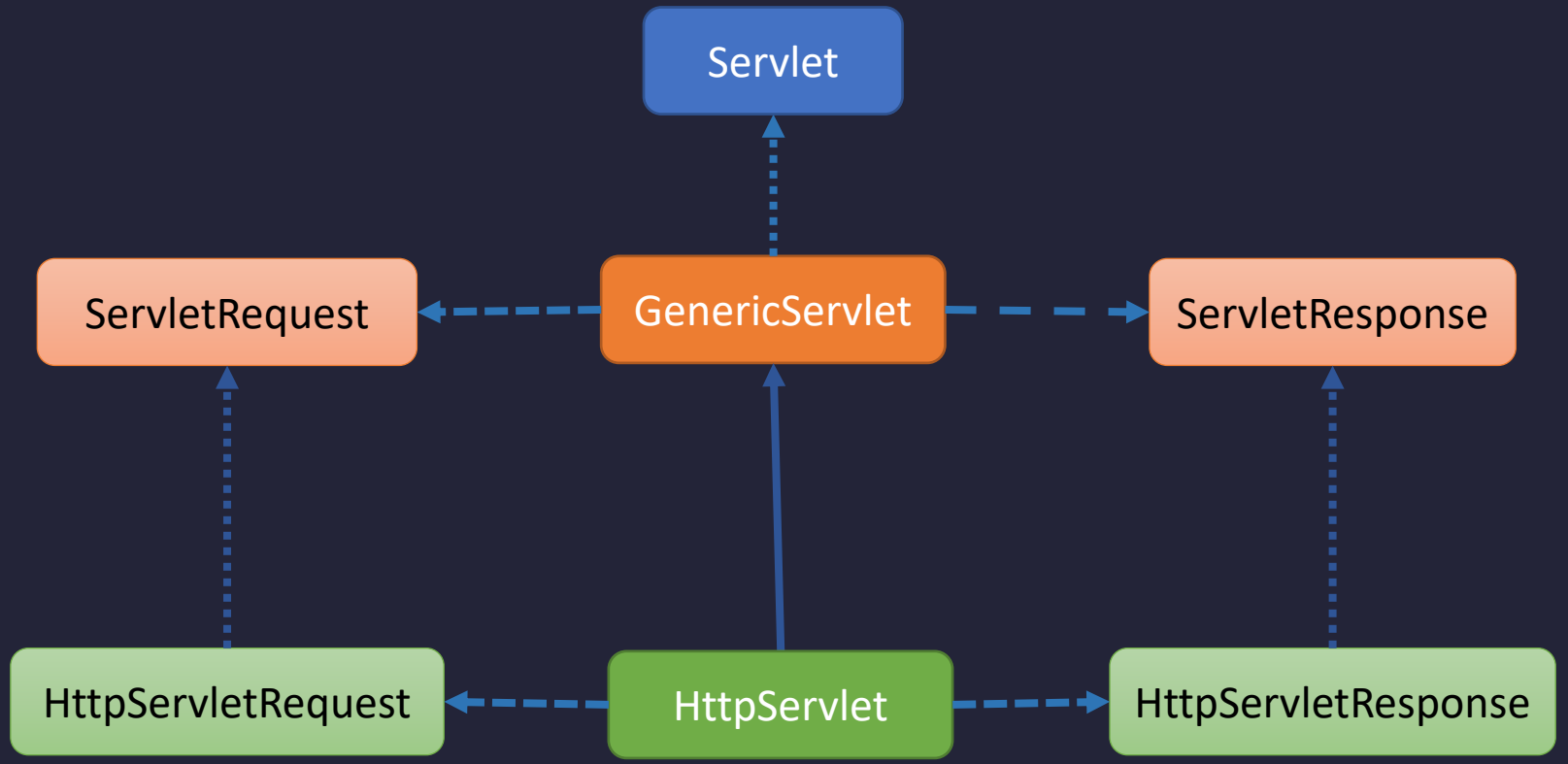
Métodos HTTP soportados

Los métodos HTTP soportados por el API Servlet son 7:

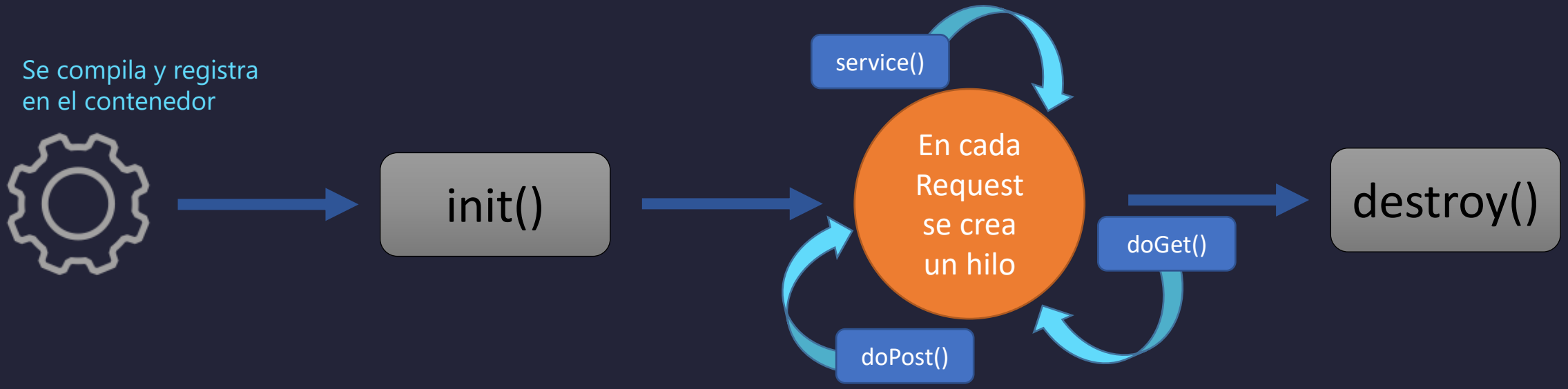
- doDelete
- doGet
- doHead
- doOptions
- doPost
- doPut
- doTrace.



Clases e Interfaces del Servlet



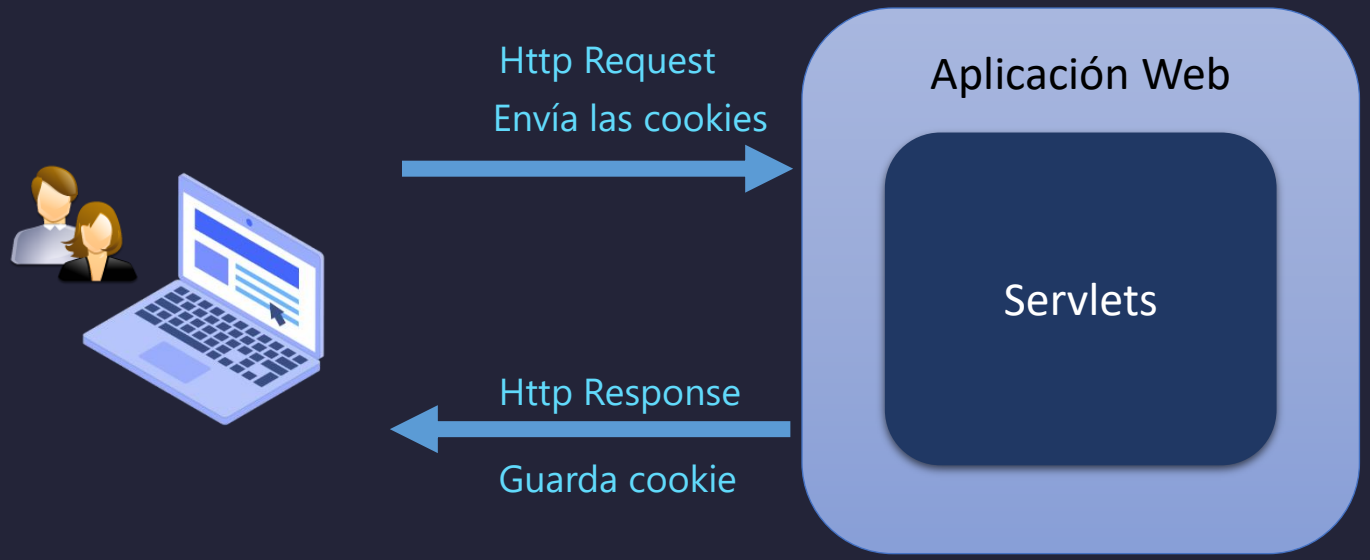
Ciclo de Vida de un Servlet



JAKARTA EE

¿Qué es una Cookie?

Una cookie son datos del usuario almacenados en el navegador web (lado del cliente) y los servidores la utilizan cuando se comunican con el cliente.



JAKARTA EE

Manejo de estados

Una de las características de las peticiones http es que no manejan estado de los datos del request, por eso es la importancia del manejo de sesiones o cookies http

Las cookies nos permiten un forma para mantener información del usuario entre peticiones y poder recordarlas

Existen dos formas de mantener información del usuario, una es usando cookies y otra el objeto HttpSession del api servlet.

JAKARTA EE

Trabajar con Cookies en Servlet

Crear una cookie

- Para enviarla al cliente, necesitamos crearla con un nombre y un valor asociado, luego agregarla a la respuesta:

```
Cookie username = new Cookie("username", "andres");
response.addCookie(username);
```

Leer una cookie del request

- Las cookies son enviadas desde el cliente al servidor

```
Cookie[] cookies = request.getCookies();
String username = Arrays.stream(cookies)
        .filter(c -> "username".equals(c.getName()))
        .map(Cookie::getValue)
        .findFirst()
        .orElse(null);
```

Trabajar con Cookies en Servlet

Eliminar una cookie

- Necesitamos crearla nuevamente con mismo nombre y asignando cero en el método setMaxAge:

```
Cookie username = new Cookie("username", "");
username.setMaxAge(0);
response.addCookie(username);
```

JAKARTA EE

Métodos importantes

- setDomain(String dominio)
- getDomain()

```
Cookie username = new Cookie("username", "andres");
username.setDomain("ejemplo.com");
```

- setMaxAge(int seg)
- getMaxAge()

```
username.setMaxAge(60*60);
```

- setPath(String path)
- getPath();

```
cookie.setPath("/webapp");
```

JAKARTA EE

Métodos importantes

- setValue(String valor)
- getValue()

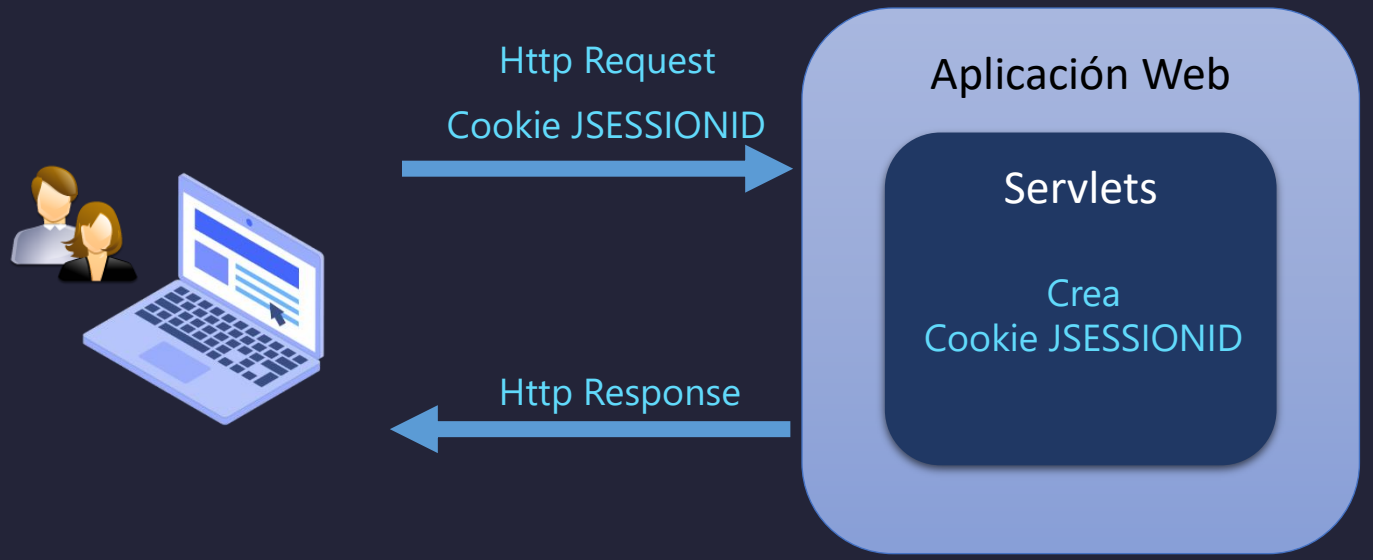
```
Cookie username = new Cookie("username", "andres");
username.setValue("admin");
out.println(username.getValue()); // imprime admin
```

- getName()

```
Cookie username = new Cookie("username", "andres");
out.println(username.getName()); // imprime username
```

HttpSession

HttpSession es otra opción para almacenar datos del usuario que sean persistentes en diferentes request y se almacena en el lado del servidor.



Manejo de estados

Los datos no se comparten entre diferentes objetos de sesión (el cliente solo puede acceder a los datos desde su sesión)

Las sesiones http nos permiten un forma para mantener información del usuario entre peticiones y poder recordarlas

También contiene pares clave-valor, pero en comparación con una cookie, una sesión puede contener un objeto como valor.

Trabajar con Session en Servlet

Crear una sesión http

- Se crea de forma automática por cada cliente o navegador web y la accedemos mediante el objeto request:

```
HttpSession session = request.getSession();
```

Obtener un objeto o valor de la sesión http

```
String username = session.getAttribute("username");
```

Guardar un objeto en la sesión http

```
session.setAttribute("username", usuario);
```

JAKARTA EE

Trabajar con Session en Servlet

Eliminar un valor de la sesión del cliente

```
session.removeAttribute("username");
```

Eliminar o invalidar la sesión actual del cliente

```
session.invalidate();
```

JAKARTA EE

Otros métodos importantes

- isNew()
- getCreationTime()
- getLastAccessedTime()
- getMaxInactiveInterval()
- setMaxInactiveInterval(int interval)