

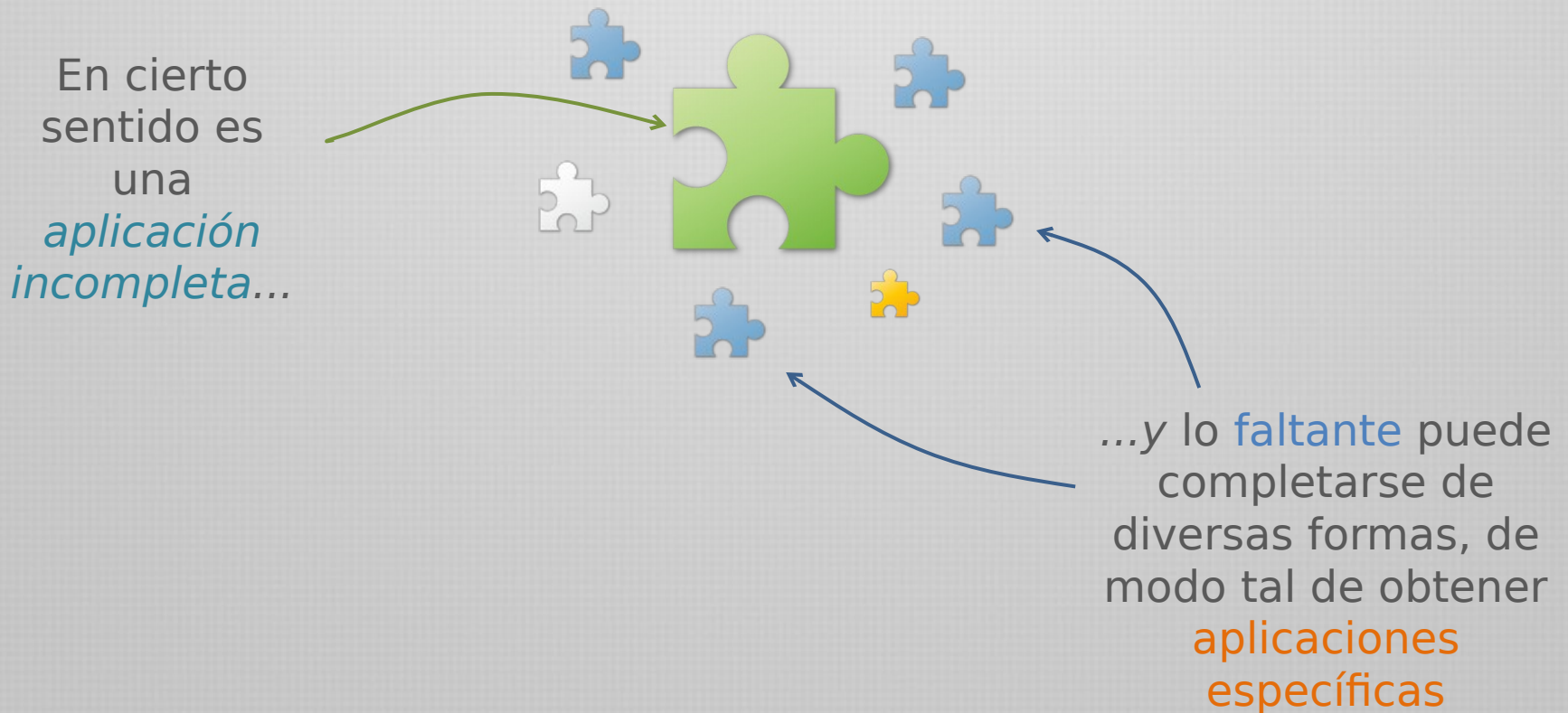
Tecnología de Programación

Martín L. Larrea

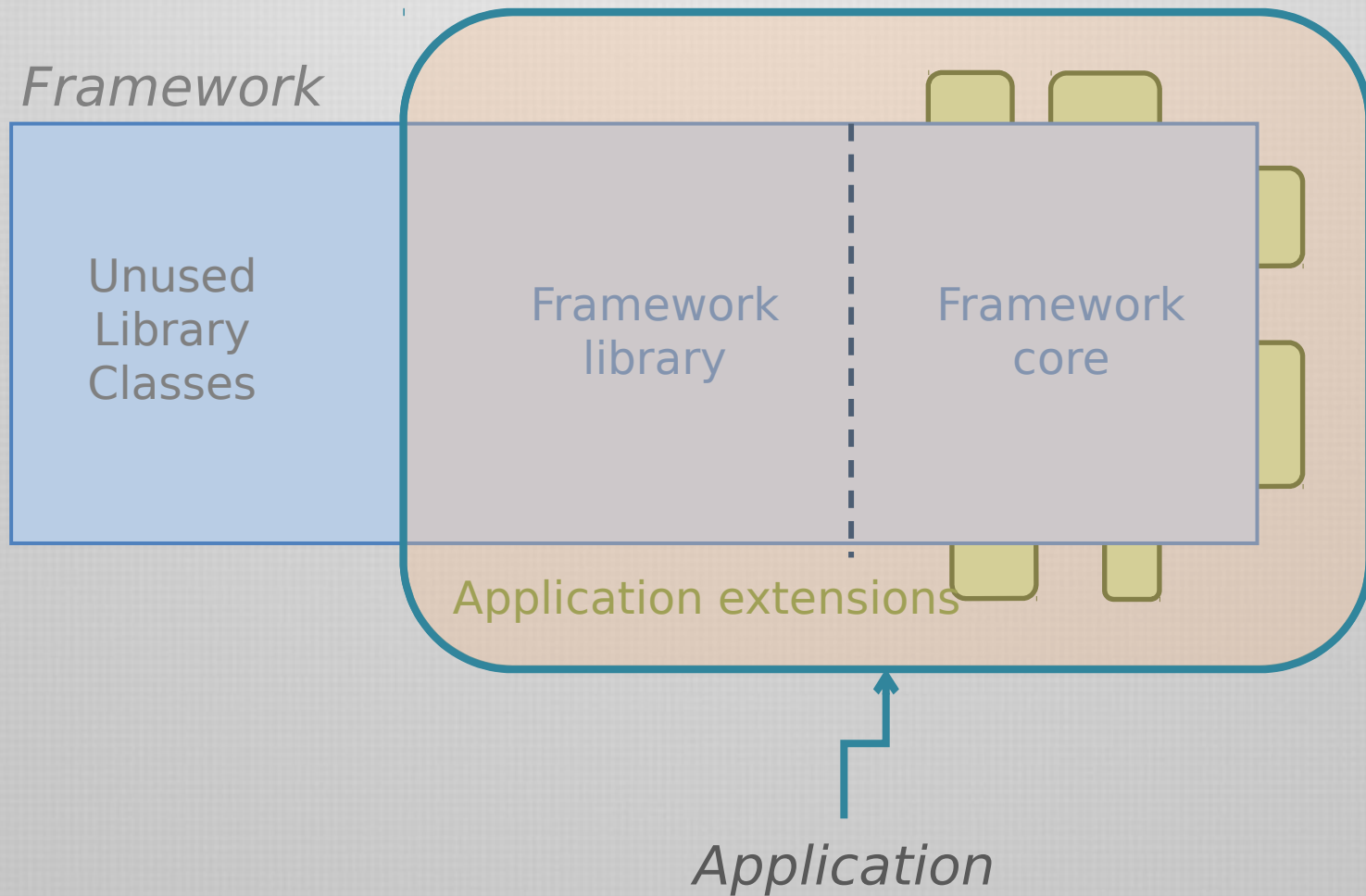
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Frameworks

Un framework es un **diseño e implementación parcial** para una aplicación en un **dominio específico**.



Partes de un framework



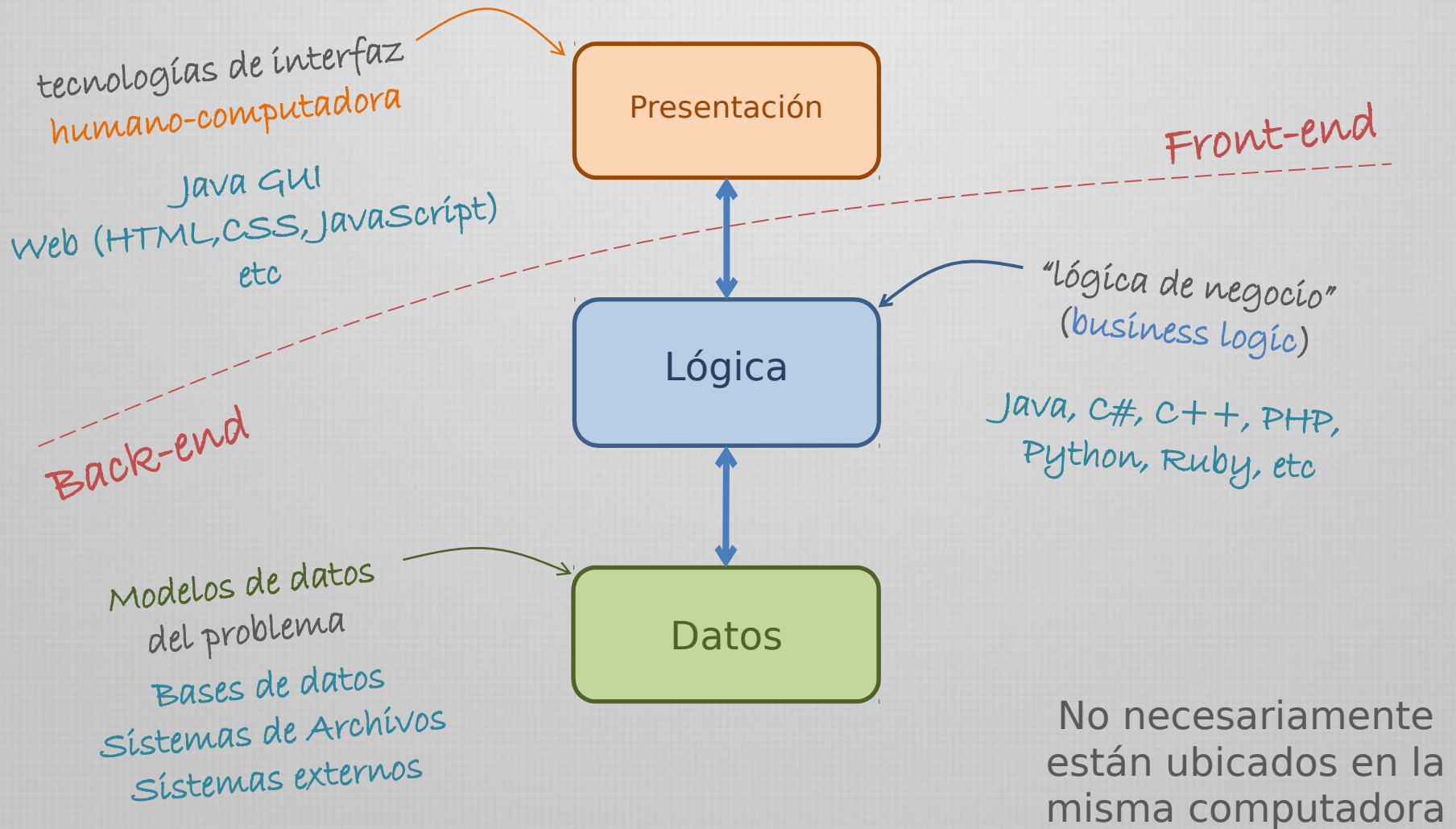
Estado actual

Actualmente el desarrollo de software es
“*framework-centric*”

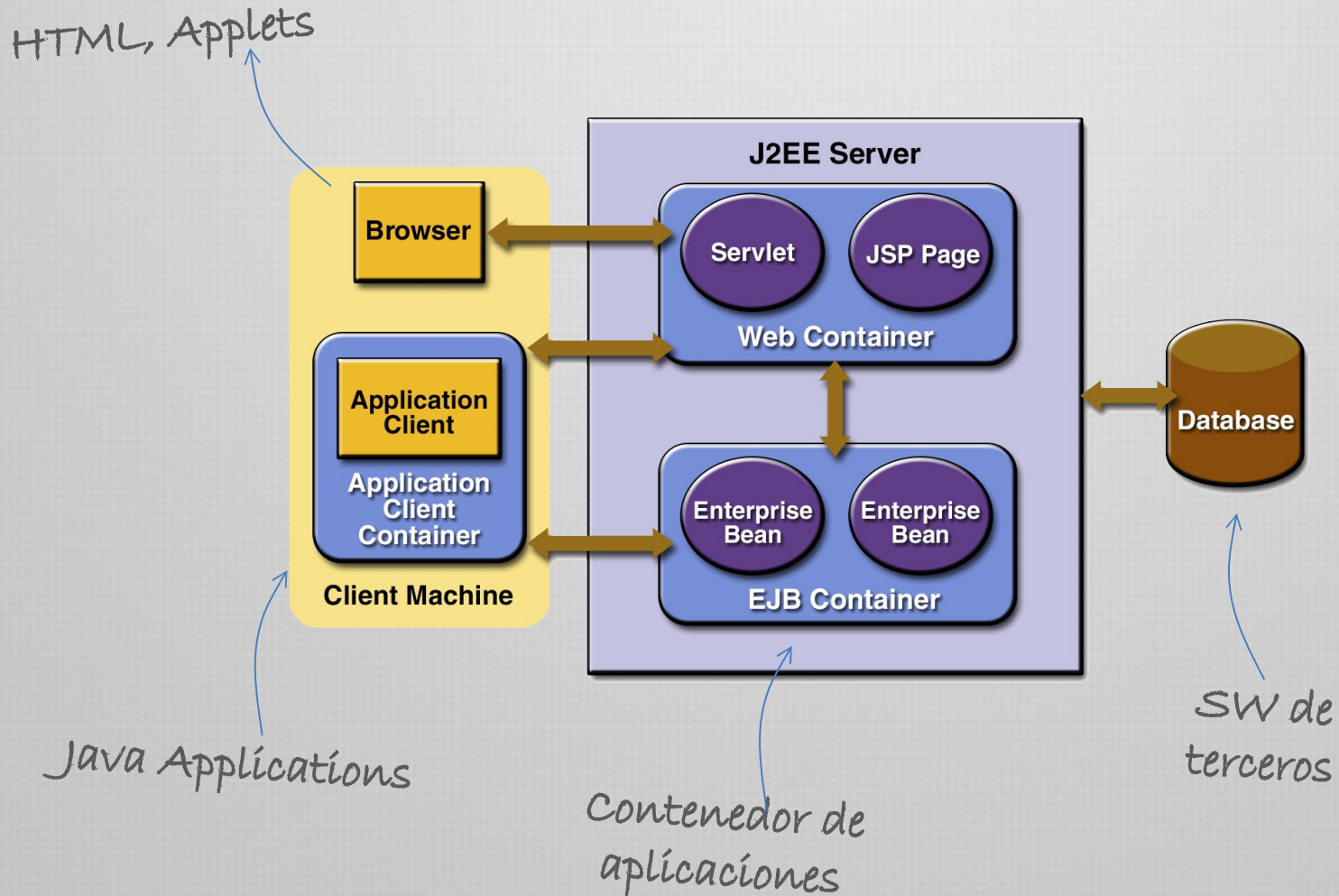


Implementan arquitecturas de software
adecuadas para los sistemas actuales
principalmente **multi-capa** y **basados en la web**

Multi-tier applications

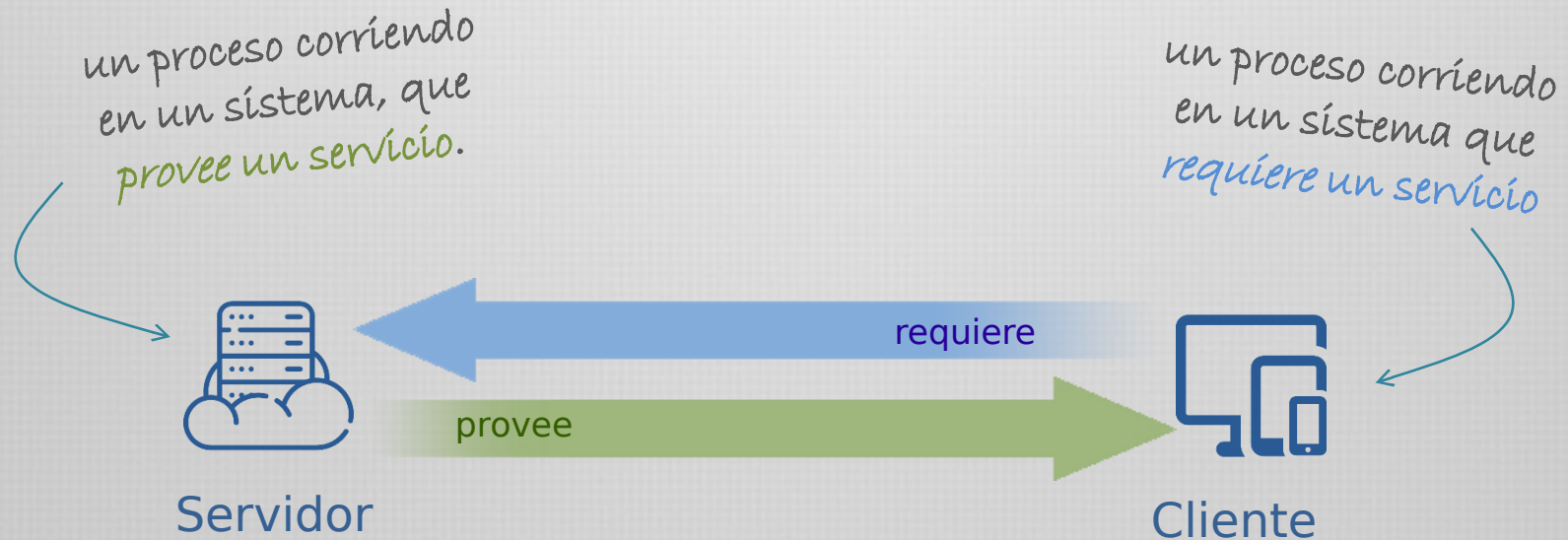


Ejemplo: Java Enterprise Edition



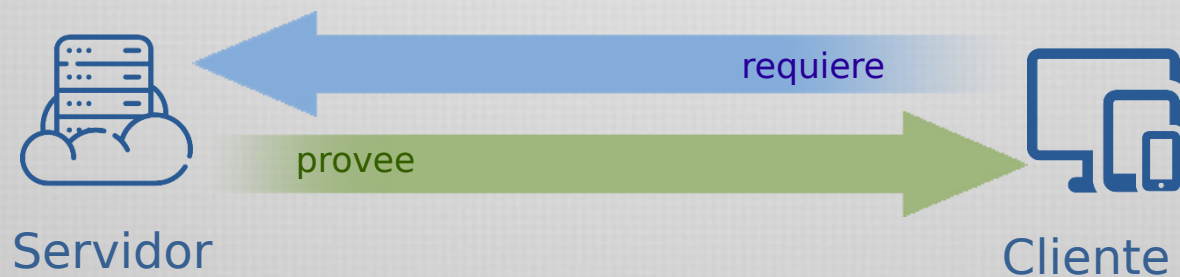
Escenario cliente-servidor

El escenario de trabajo web es el del modelo *cliente-servidor*



*Las computadoras son ilustrativas!
Un sistema puede alojar más de un servidor o cliente*

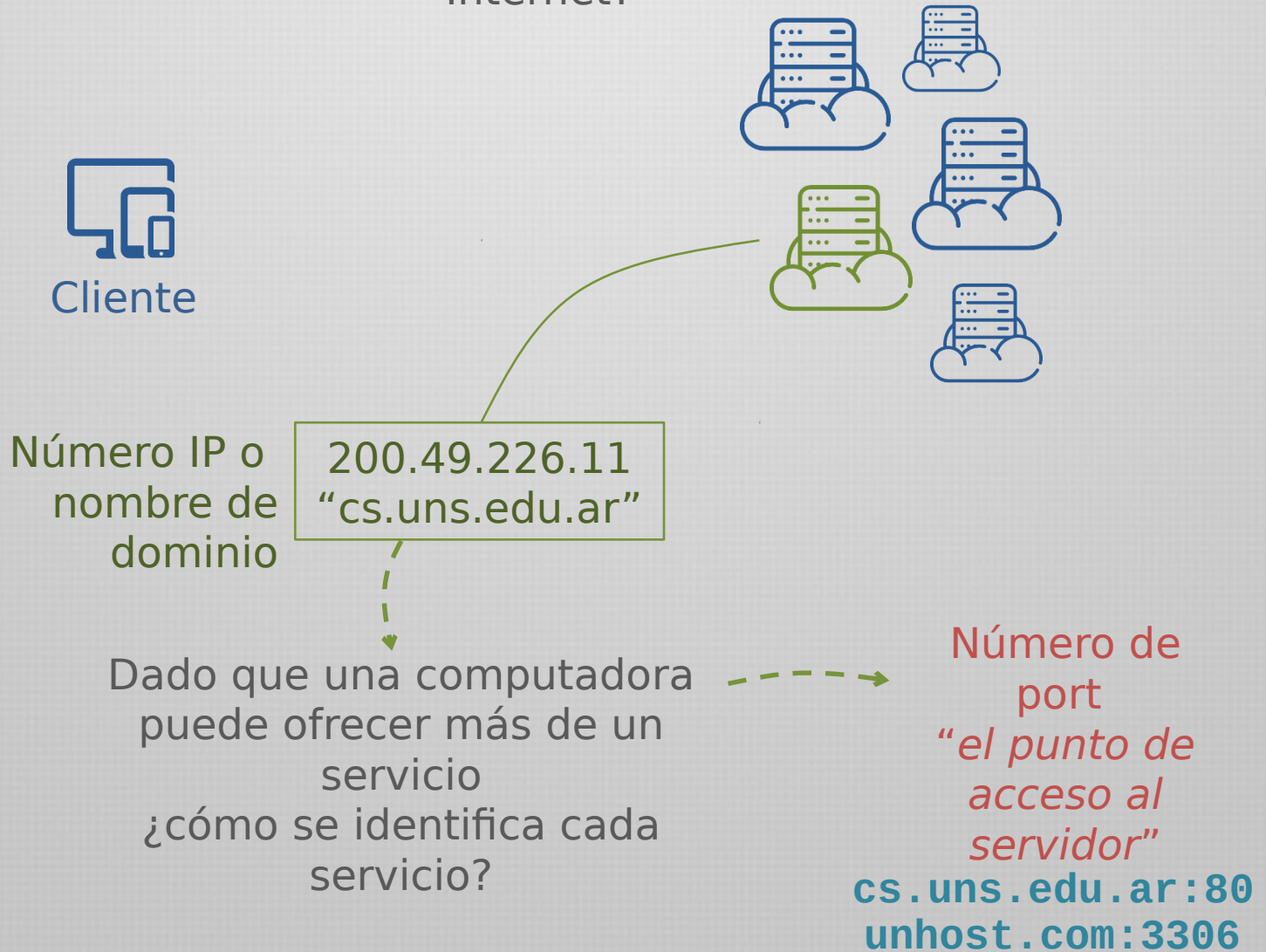
Escenario cliente-servidor



El programa cliente debe *saber ubicar* al servidor y *comprender* su servicio.

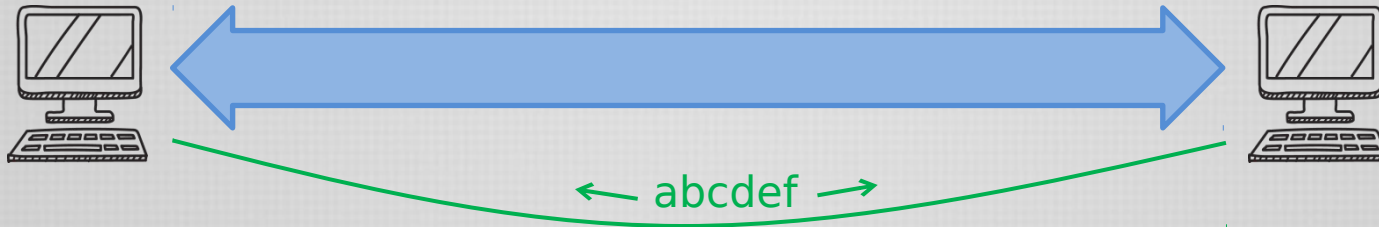
Escenario cliente-servidor

¿cómo el cliente identifica al servidor entre todas las computadoras de Internet?



Internet

Internet define una forma de **conexión** de redes heterogéneas.



Los **usos** de esta conexión son variados y cada uno determina a su vez, diferentes **protocolos de comunicación**.

Entre ellos:

envío y recepción de mensajes
(SMTP)

envío y recepción de archivos
(FTP)

... además la posibilidad de explorar documentos

Web

1989



Tim Berners-Lee



World Wide Web

los
documentos
están escritos
en hipertexto
(HTML)



El protocolo de
comunicación es
HTTP



Navegadores



En 1994 Berners-Lee funda el *World Wide Web Consortium* (W3C) en el MIT, con apoyo de DARPA.

La idea central era asegurar la **compatibilidad** por medio de la definición de estándares, denominados **W3C Recommendations**.

HTTP

HTTP es el protocolo de red para la Web.

"Hypertext Transfer Protocol"

protocolo de aplicación para la entrega de **documentos** variados

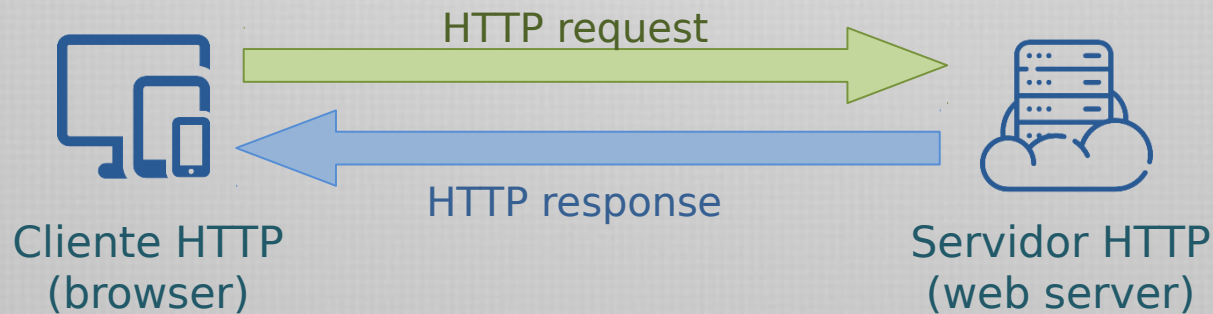
protocolo de comunicación define

"recursos"

ubicables por medio del **URL**.

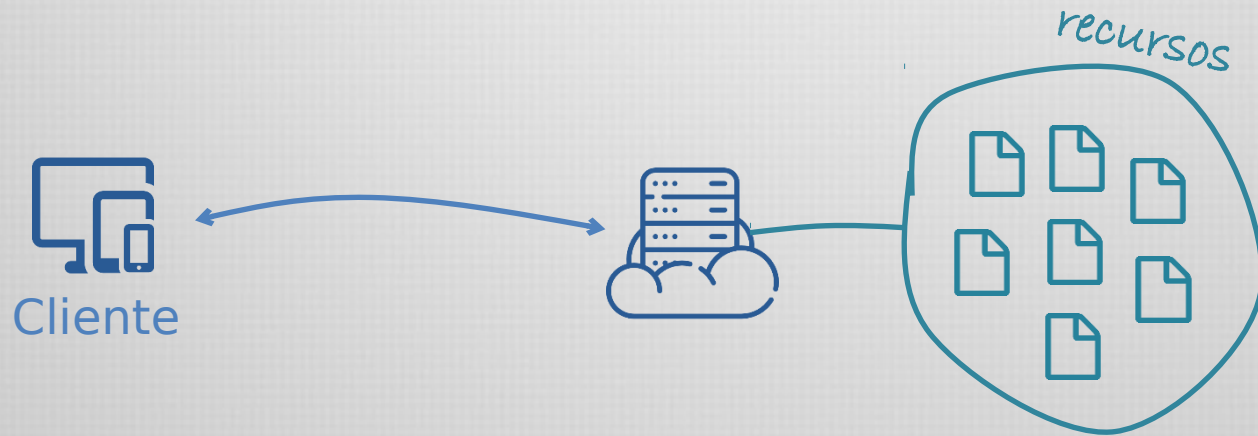


tipo y estructura de los mensajes
las reglas del diálogo



Servidores web

El servidor web actúa como una **interfaz** entre un conjunto de **recursos** y los **clientes**.

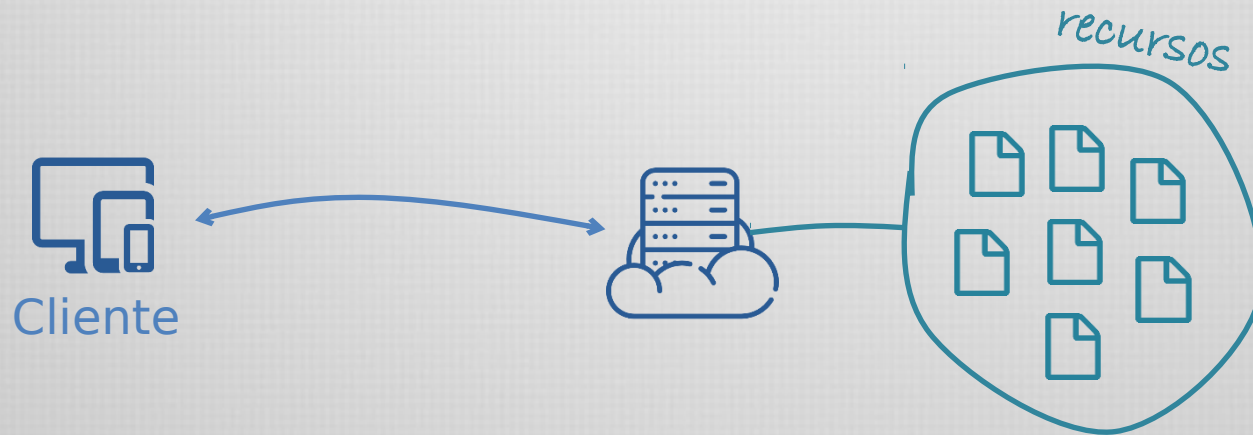


ubicados en una carpeta específica, denominada **document root** o **docroot**.

htdocs
public_html
www
webroot
webaps

Servidores web

El servidor web actúa como una **interfaz** entre un conjunto de **recursos** y los **clientes**.



Para solicitar un recurso del servidor,
es necesario mencionar
el **host** y el **camino hacia el recurso**

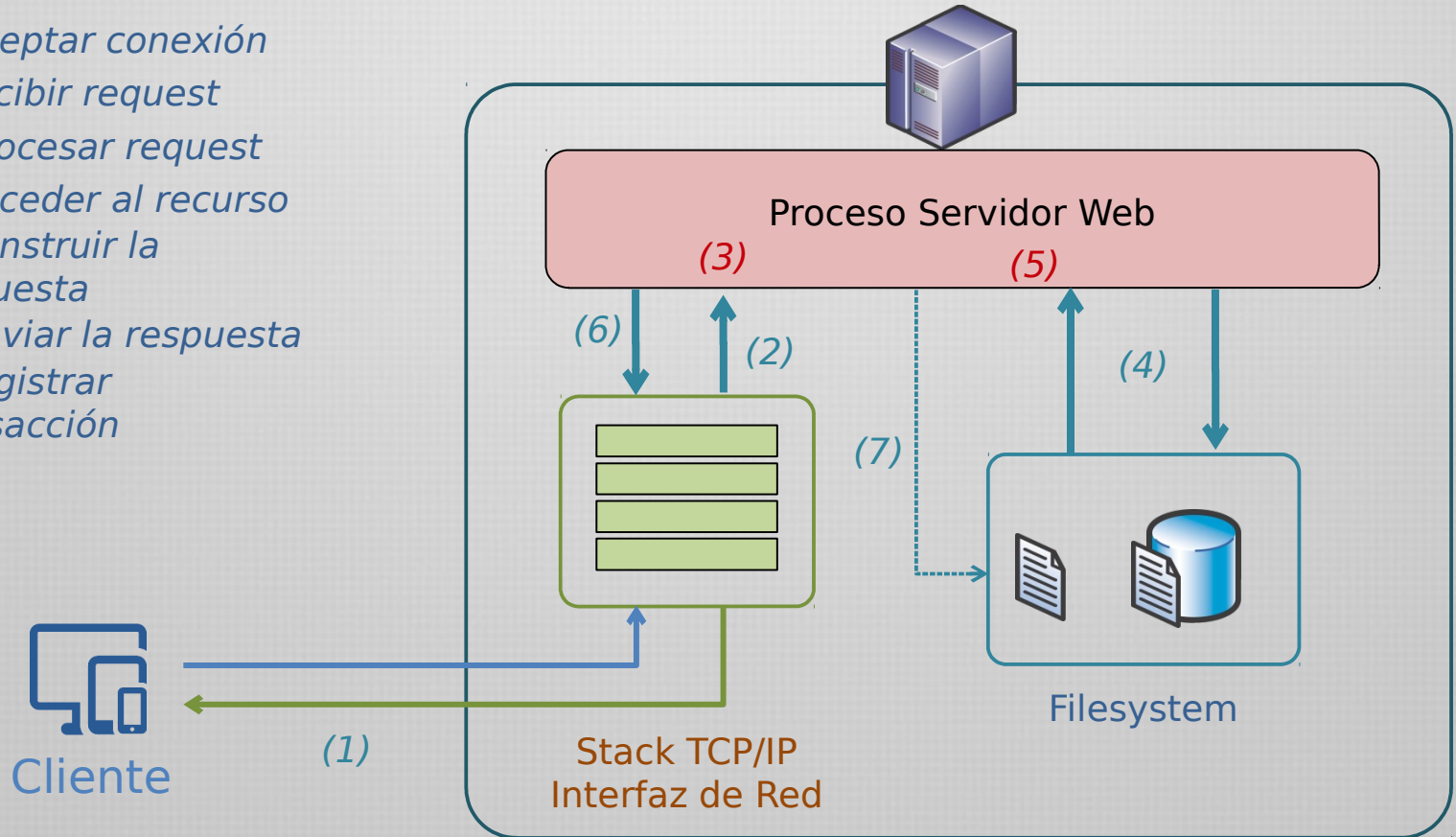


No necesariamente el camino físico:
`http://unhost.com/~john/index.html`
puede ser
`/home/users/john/web/index.html`
en el servidor `unhost.com`

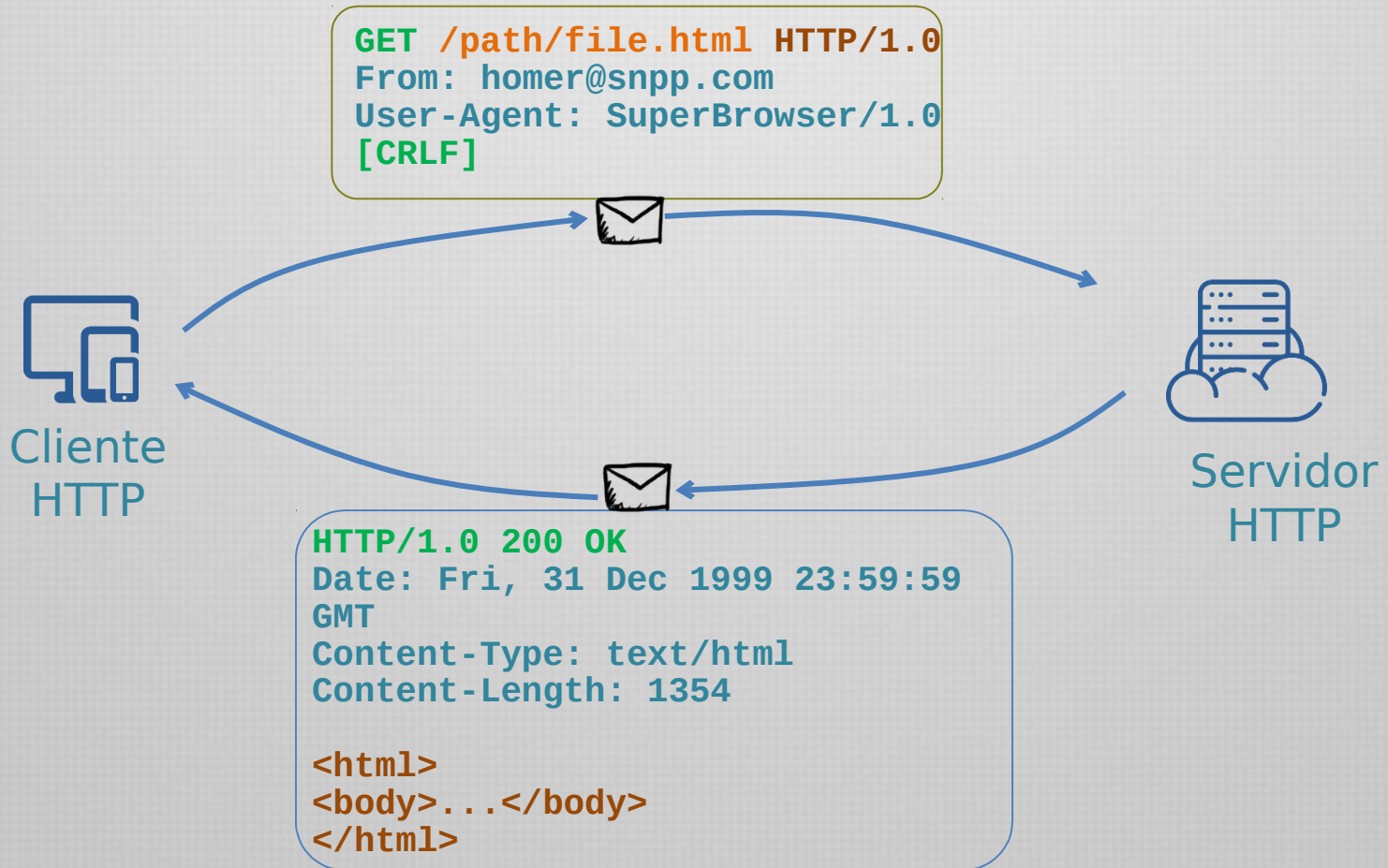
Servidores web

El esquema de trabajo de un servidor web es el siguiente:

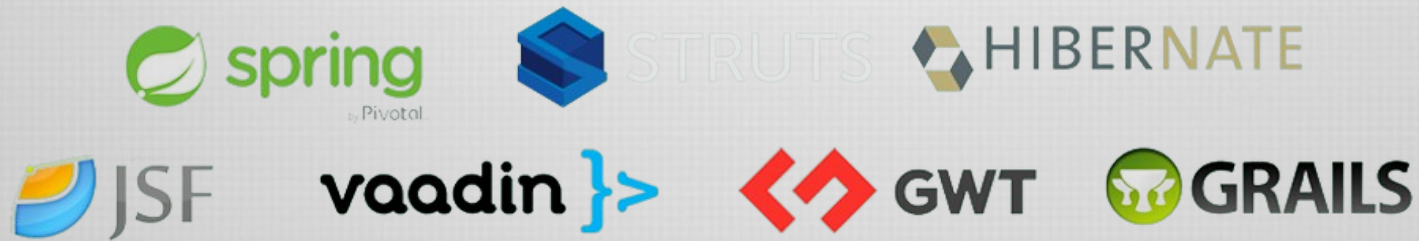
1. aceptar conexión
2. recibir request
3. procesar request
4. acceder al recurso
5. construir la respuesta
6. enviar la respuesta
7. registrar transacción



Requests - responses



Java Frameworks



Existen muchos frameworks en Java
Predominantemente para el escenario web,
tecnología indispensable en la actualidad

GWT



Google Web Toolkit es un framework para el desarrollo de aplicaciones web, enfocado en el **front-end**

Traduce Java a JavaScript
(no es bytecode sobre la JVM)

Se utiliza en muchos productos web de Google
AdWords, AdSense, Flights, Hotel Finder, Offers, Wallet, Blogger

GWT

Pasos generales para la creación de una app GWT

Crear un proyecto GWT

por ejemplo, usando Google Plugin para Eclipse

Diseñar la Aplicación

identificar requerimientos, restricciones, estrategias

Diseñar la Interfaz de usuario

identificar componentes necesarios y su distribución (layout)

Codificar

administrar eventos de usuario, implementar la lógica necesaria

Debug

Development Mode

Compilar

GWT Compiler – producto final

GWT - Esquema habitual

```
imports...
```

```
public class MyAplicacion implements EntryPoint {
```

*Clase de GWT
(framework hook)*



```
//Atributos
```

```
private TextBox mitexto = new TextBox();
```

```
private Button unboton = new Button("Click");
```

```
public void onModuleLoad() {
```

```
    // Armar interfaz
```

```
    mainPanel.add(mitexto);
```

```
    ...
```

```
    // Ubicar esta app en el HTML
```

```
    RootPanel.get("lugarMiApp").add(mainPanel);
```

```
    // Oyente
```

```
    unboton.addClickHandler(new ClickHandler() {
```

```
        public void onClick(ClickEvent event) {
```

```
            atenderClick();
```

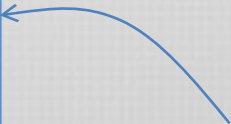
```
        }
```

```
    });
```

```
    ...
```

```
}
```

*Al cargar la
página web*

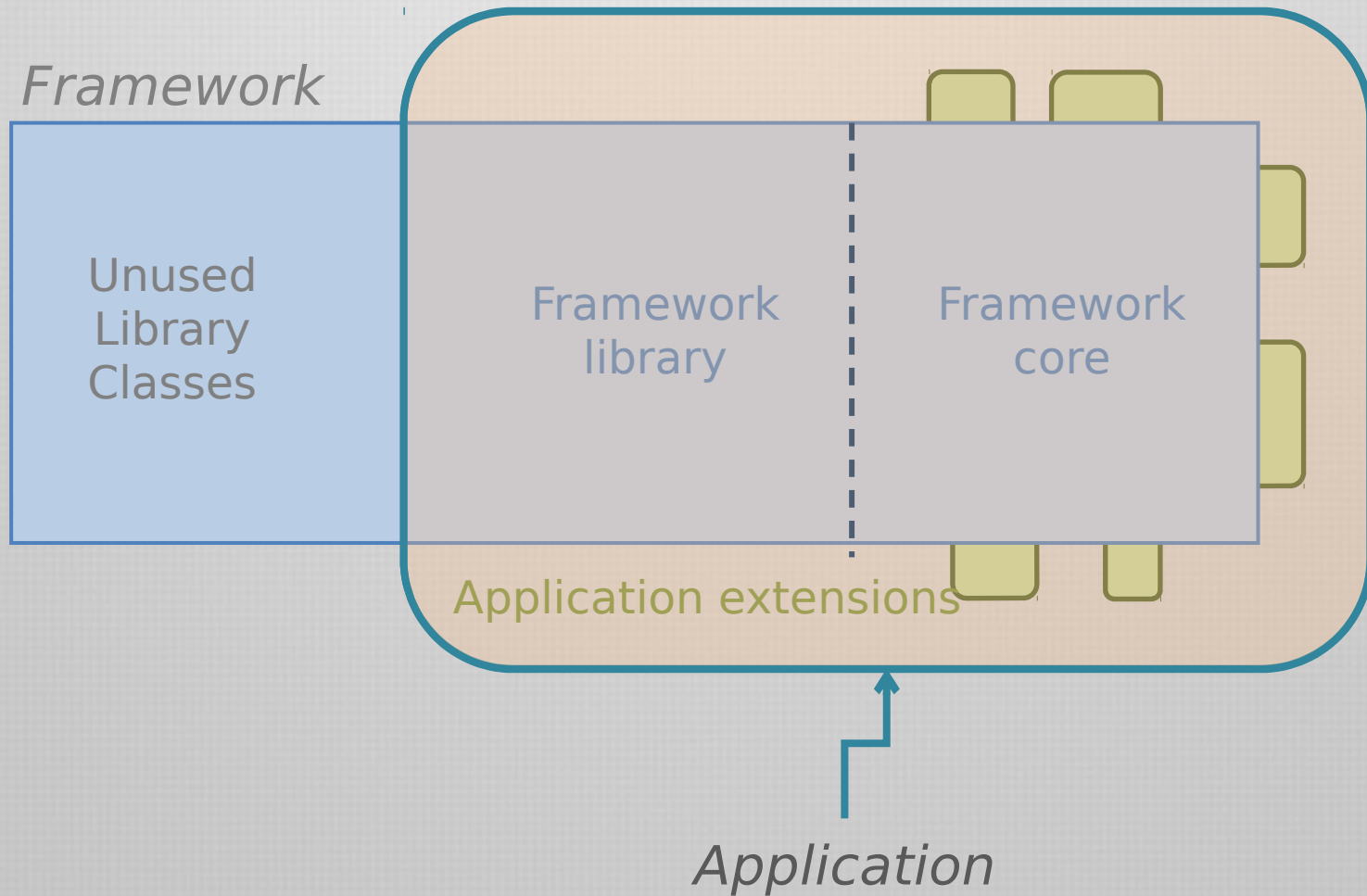


```
private void atenderClick() {...}
```

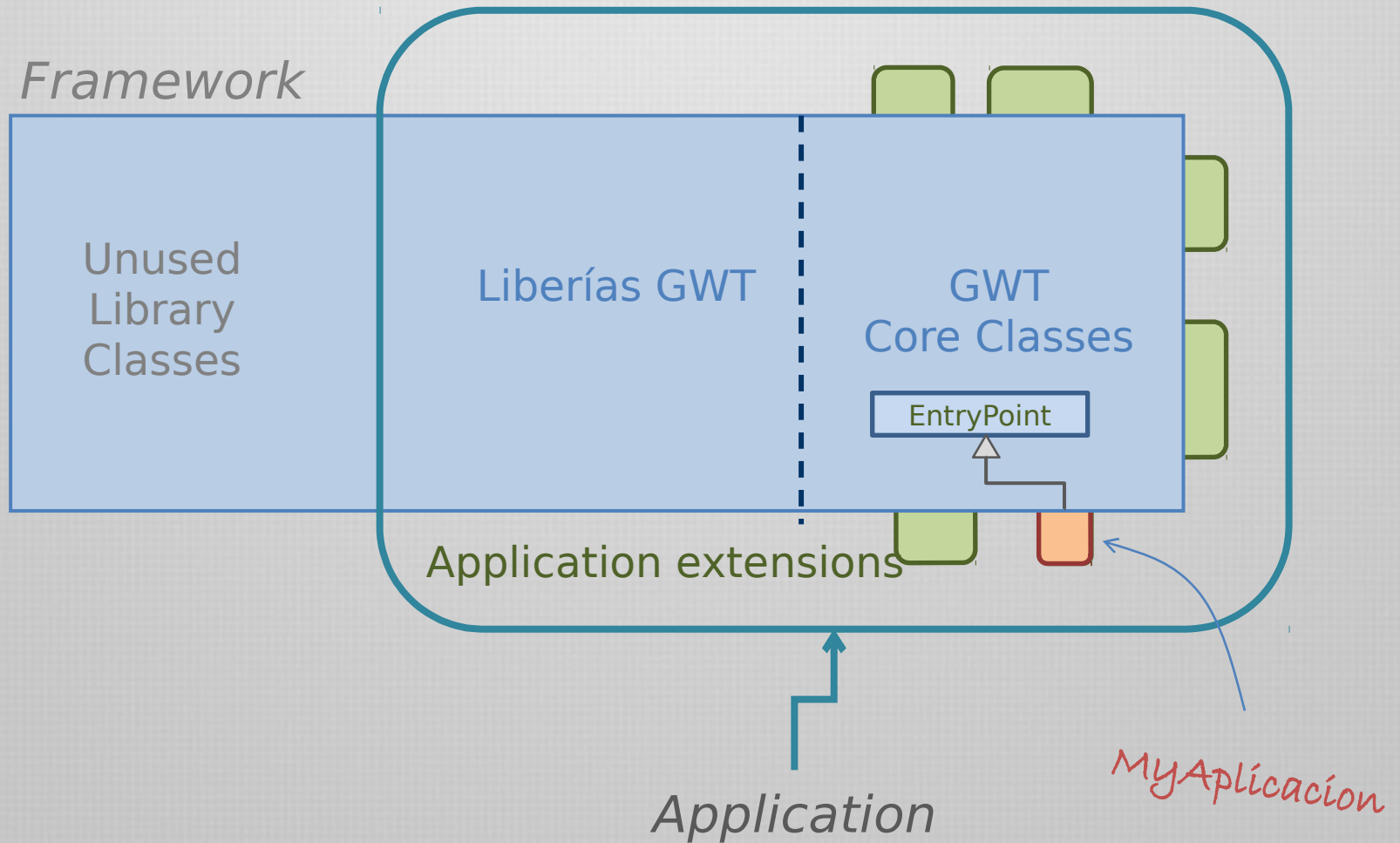
```
...
```

```
}
```

Partes de un framework



Partes de un framework



GWT



<http://www.gwtproject.org/doc/latest/tutorial/index.html>