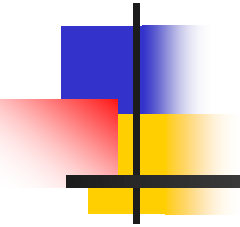


Tema 8: Introducción al Desarrollo de Aplicaciones Web





Índice

- Ejemplo de interfaz web para ws-movies
- Aplicaciones web del lado servidor
 - Definición
 - Modelo de interacción
 - Arquitecturas típicas
 - Modelo de desarrollo
- Aplicaciones web SPA
 - Motivación
 - Definición
 - Modelo de interacción
 - Arquitectura típica
 - Modelo de desarrollo

Ejemplo de interfaz web para ws-movies

Movies — Hello John!

[Logout](#)
[Add](#)
[Search](#)
[...](#)

Title

Copyright © UDC

Movies — Hello John!

[Logout](#)
[Add](#)
[Search](#)
[...](#)

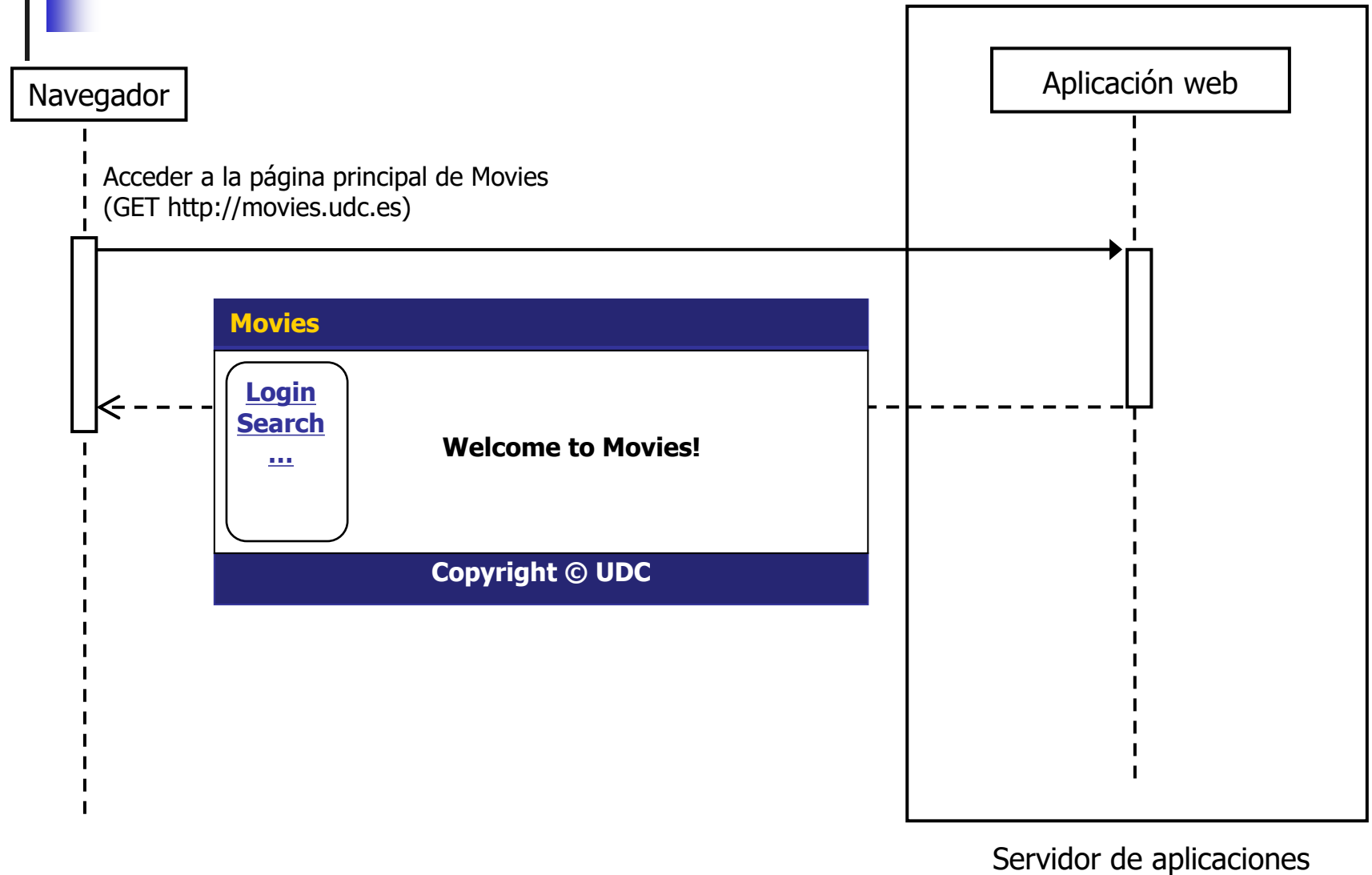
Title

The Matrix	136 min	...
The Matrix Reloaded	138 min	...
The Matrix Revolutions	129 min	...

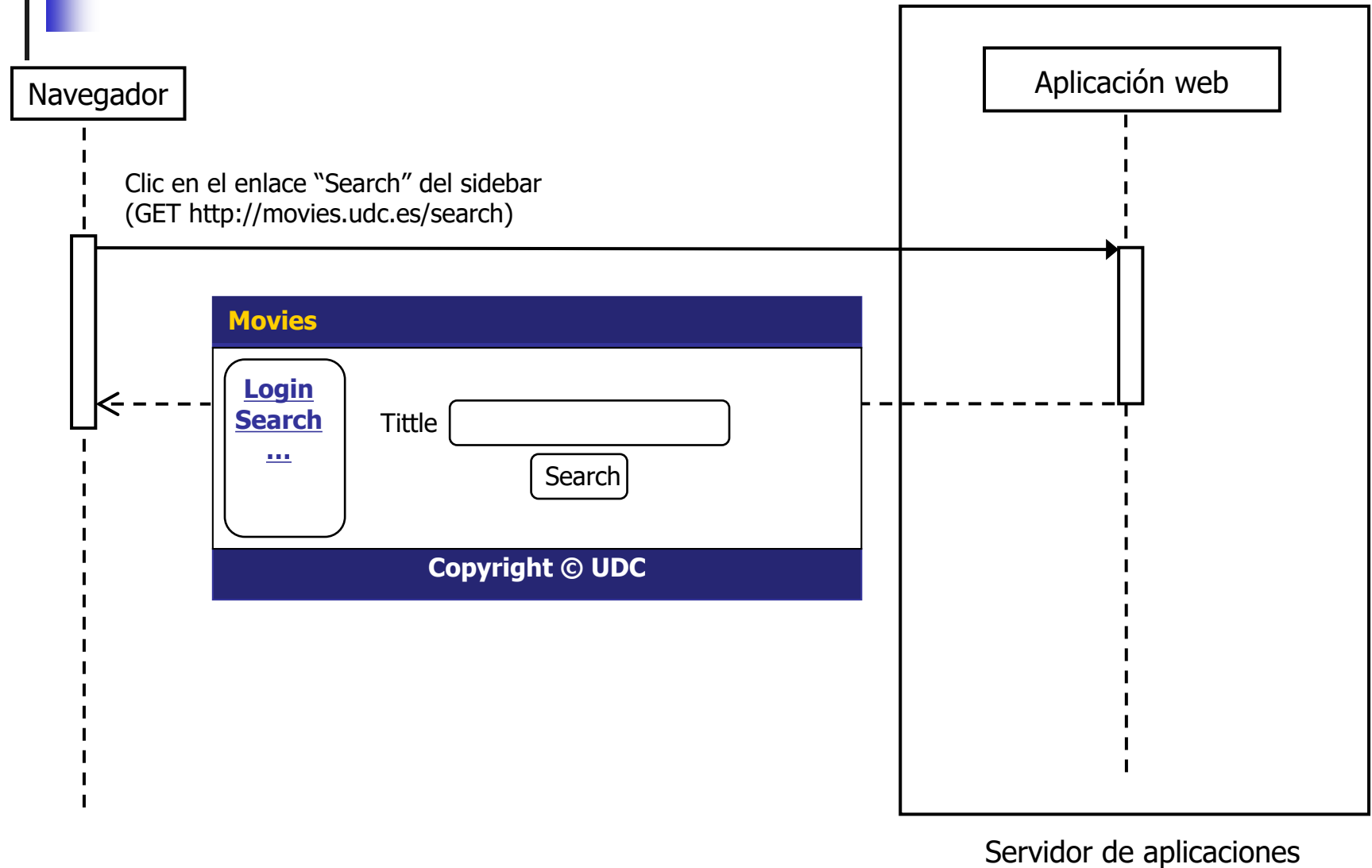
Copyright © UDC

- Una **aplicación web del lado servidor** es aquella que genera dinámicamente el HTML que visualiza el navegador en el lado servidor
- Se instala en un servidor de aplicaciones
 - El servidor de aplicaciones recibe peticiones HTTP procedentes del navegador y se las pasa a la aplicación a la que van dirigidas
 - Usando una API de programación, la aplicación genera el HTML de la página de respuesta y el servidor de aplicaciones incluye ese HTML en el cuerpo de la respuesta HTTP
 - Ejemplo: en Java una aplicación web del lado servidor se puede empaquetar en un fichero WAR, instalable en un servidor de aplicaciones Java EE estándar (e.g. Tomcat)

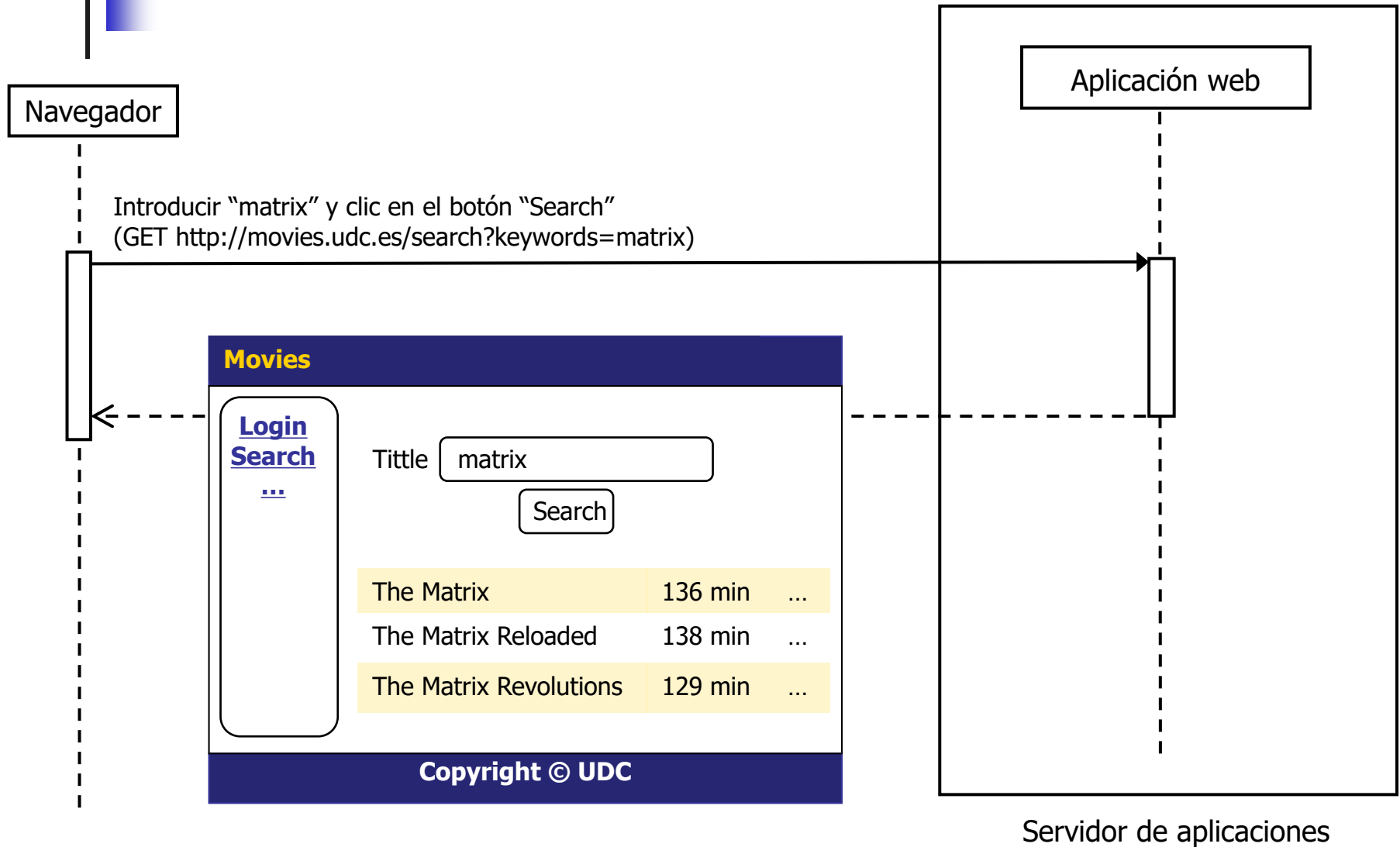
[Aplicaciones web del lado servidor] Modelo de interacción



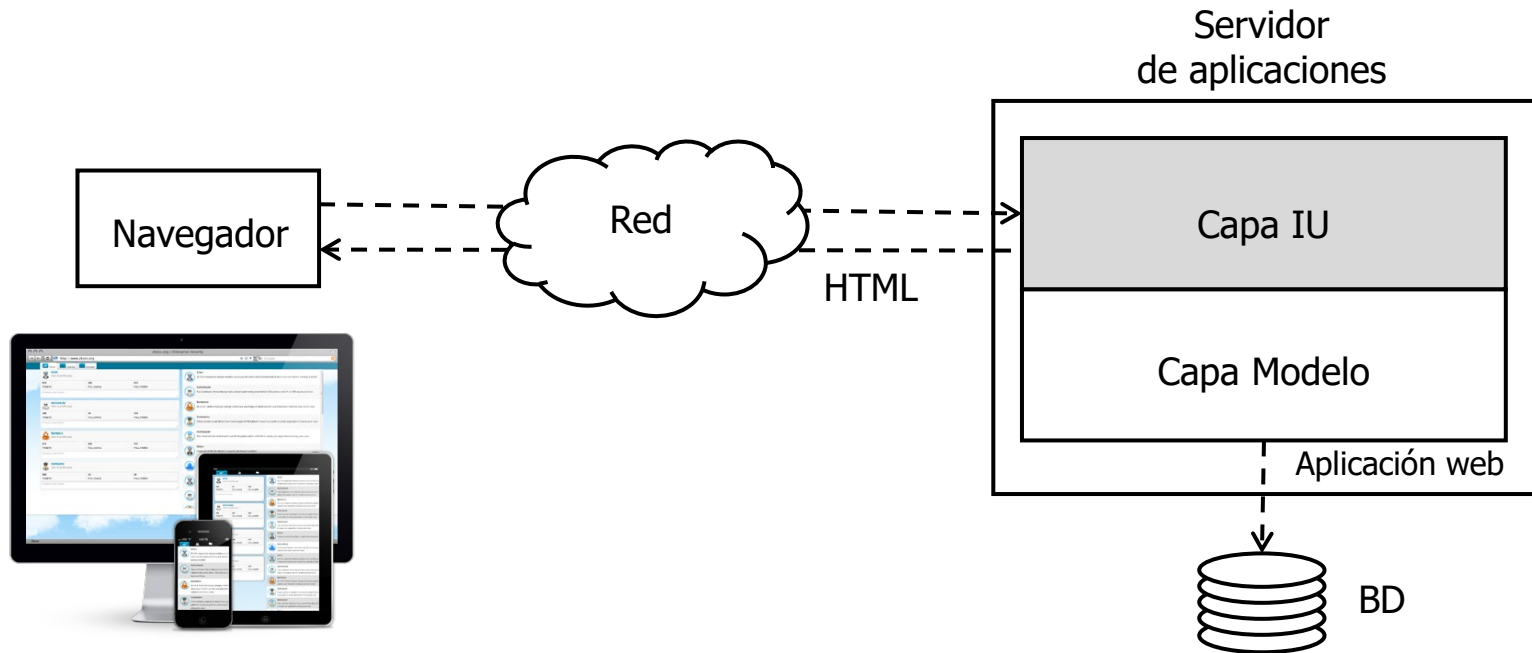
[Aplicaciones web del lado servidor] Modelo de interacción



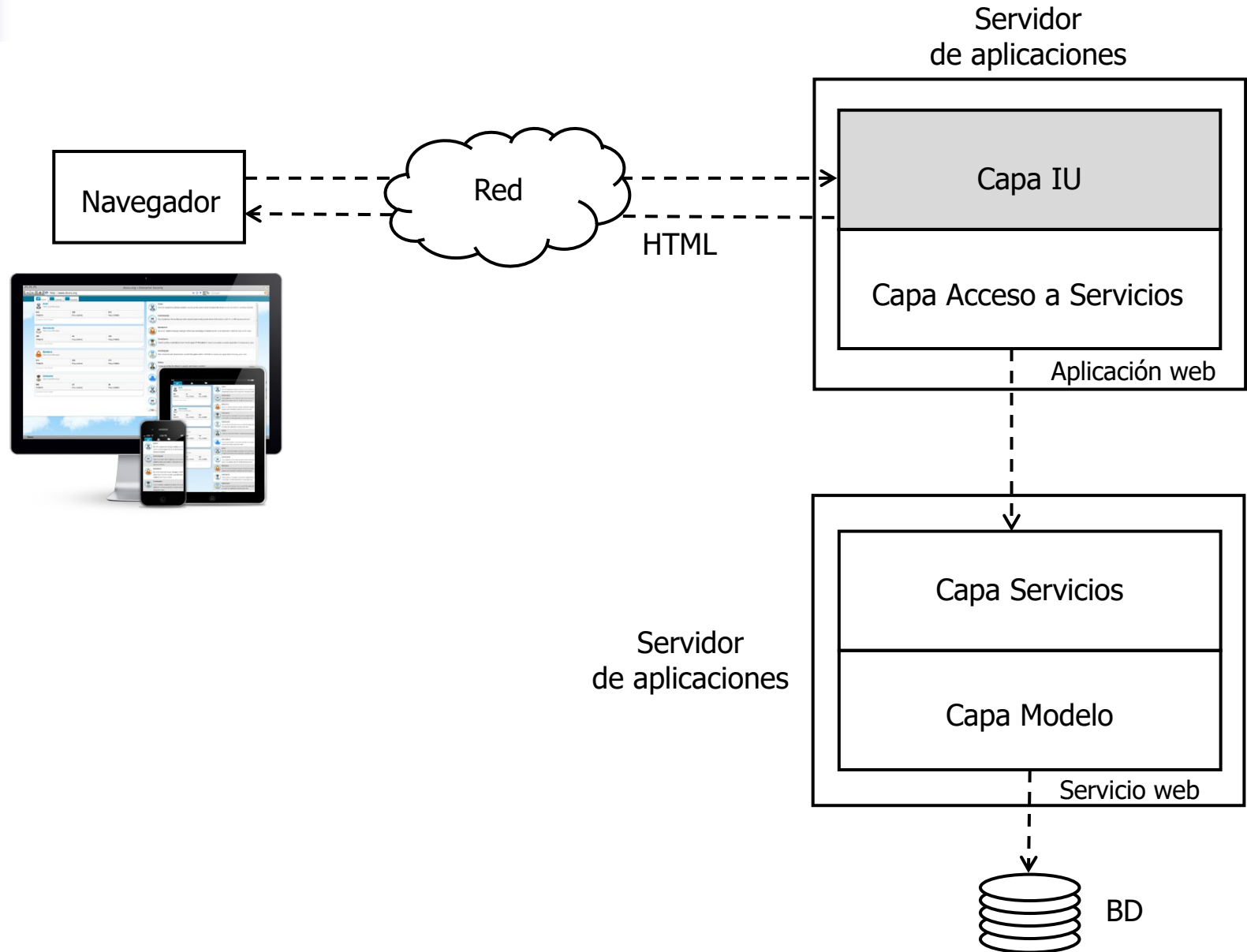
[Aplicaciones web del lado servidor] Modelo de interacción



[Aplicaciones web del lado servidor] Arquitectura con capa Modelo local



[Aplicaciones web del lado servidor] Arquitectura con capa Modelo remota



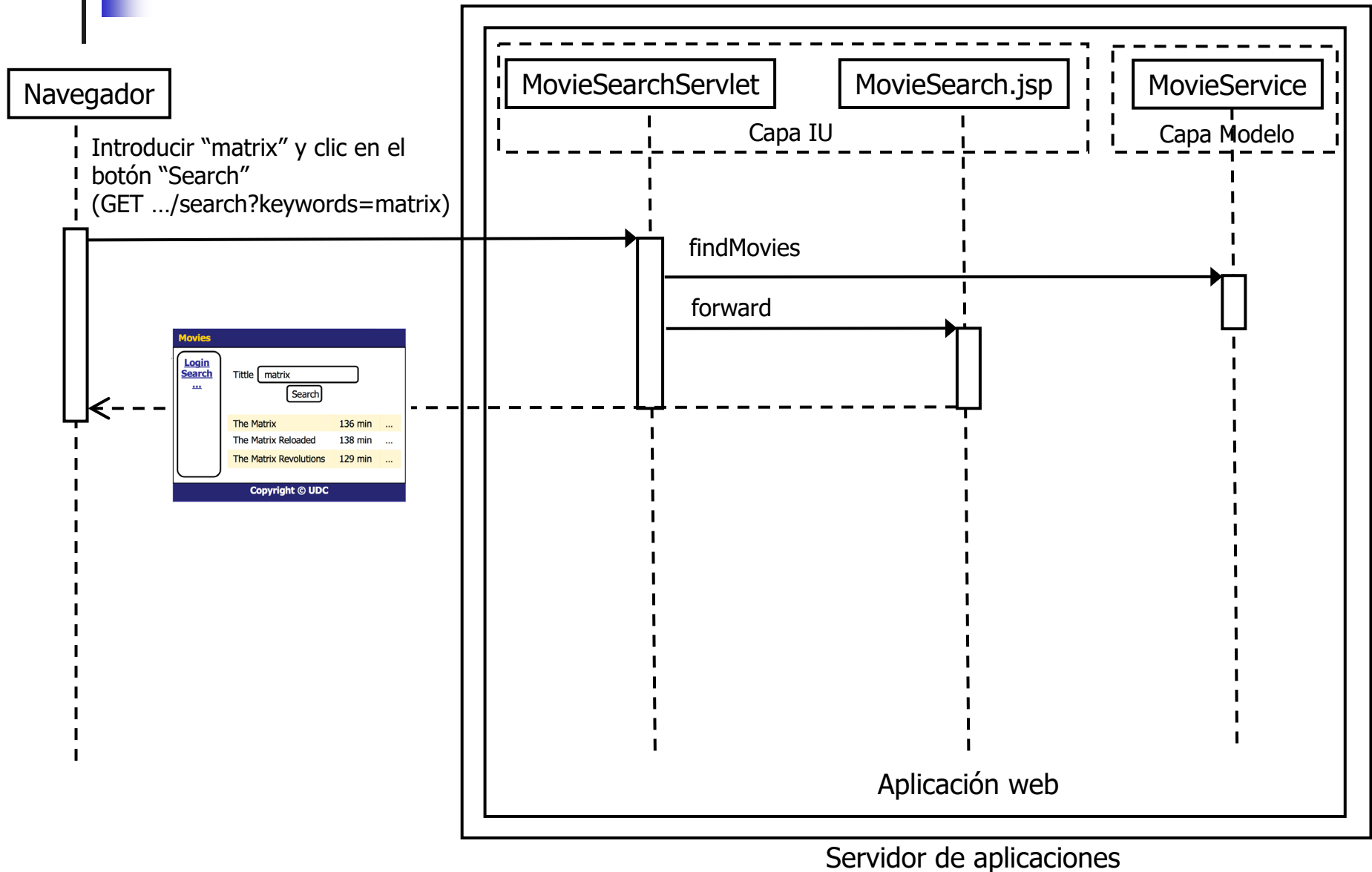
- Arquitectura con capa Modelo local
 - Más sencilla de desarrollar y gestionar
- Arquitectura con capa Modelo remota
 - Seguramente necesaria cuando la capa Modelo es usada también por otras aplicaciones
 - Seguramente necesaria cuando la capa Modelo está desarrollada en un lenguaje distinto a la capa IU

- La capa IU de la aplicación web tiene que procesar cada petición HTTP procedente del navegador y devolver una respuesta HTTP con el HTML de la página resultado
- En consecuencia, para implementar la capa IU, por cada petición HTTP que se puede recibir, típicamente hay que
 - [1] Implementar un trozo de código que permita recibir la petición (e.g. `/search?keywords=matrix`)
 - [2] Acceder a los parámetros de la petición
 - [3] Invocar un caso de uso de la capa Modelo (`MovieService.findMovies`)
 - [4] Generar el HTML de la página resultado y devolverlo en la respuesta HTTP

- Para facilitar la generación del HTML de la página resultado (paso [4]), los frameworks de desarrollo de aplicaciones del lado servidor proporcionan el concepto de **plantilla**
- Una plantilla es un fichero que se ejecuta en el lado servidor y permite generar el HTML de una página resultado de manera ágil
- Normalmente se implementa una plantilla para cada página resultado
- El aspecto de una plantilla recuerda al de una página HTML estática, pero a diferencia de ésta dispone de elementos para
 - Realizar bifurcaciones
 - Realizar iteraciones
 - Imprimir valores
 - Etc.

- Existen múltiples frameworks para el desarrollo ágil de aplicaciones web del lado servidor, disponibles para los lenguajes más populares
- Ejemplos de frameworks para Java: Spring MVC, Struts, JavaServer Faces, Tapestry, Wicket, etc.
- A modo de ejemplo, las siguientes transparencias esbozan una implementación “artesanal” del caso de uso de búsqueda de películas en la capa IU usando servlets para procesar las peticiones HTTP y JSP (JavaServer Pages) como sistema de plantillas
 - Profesionalmente se usaría uno de los frameworks mencionados anteriormente

[Aplicaciones web del lado servidor] Modelo de desarrollo





[Aplicaciones web del lado servidor] Ejemplo: MovieSearchServlet

```
public class MovieSearchServlet extends HttpServlet {

    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException {

        String keywords = request.getParameter("keywords");
        MovieService movieService =
            MovieServiceFactory.getService();
        List<Movie> movies = movieService.
            findMovies(keywords);

        request.setAttribute("movies", movies);
        request.getRequestDispatcher("MovieSearch.jsp").forward(
            request, response);

    }

}
```



[Aplicaciones web del lado servidor] Ejemplo: MovieSearch.jsp

```
<< ... Generate header and sidebar ... >>
```

```
<div id="content">
```

```
<form method="GET" action="search">
```

```
  <div class="field">
```

```
    <label for="keywords">Title</label>
```

```
    <span class="entry">
```

```
      <input type="text" id="keywords" name="keywords"
```

```
        value="<c:out value="param.keywords"/>"
```

```
        size="20" maxlength="255"/>
```

```
    </span>
```

```
  </div>
```

```
  <div class="button">
```

```
    <input type="submit" value="Search"/>
```

```
  </div>
```

```
</form>
```




[Aplicaciones web del lado servidor] Ejemplo: MovieSearch.jsp

```
<c:if test="${!empty movies}">
```

```
<table class="movies">
```

```
<c:forEach var="movie" items="${movies}">
```

```
<tr>
```

```
<td><c:out value="${movie.title}"/></td>
```

```
<td><c:out value="${movie.runtime}"/></td>
```

```
<< ... >>
```

```
</tr>
```

```
</c:forEach>
```

```
</table>
```

```
</c:if>
```

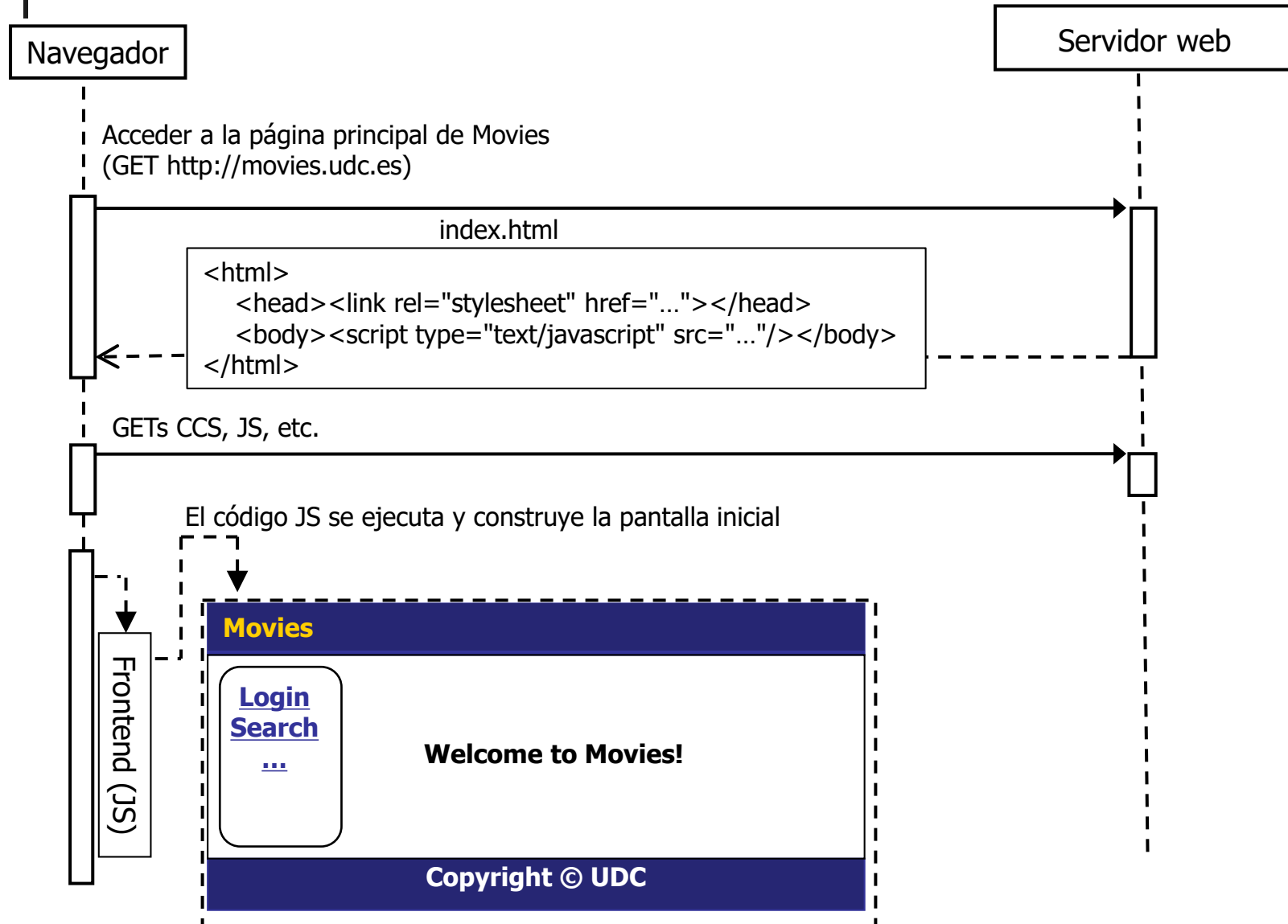
```
</div>
```

```
<< ... Generate footer ... >>
```

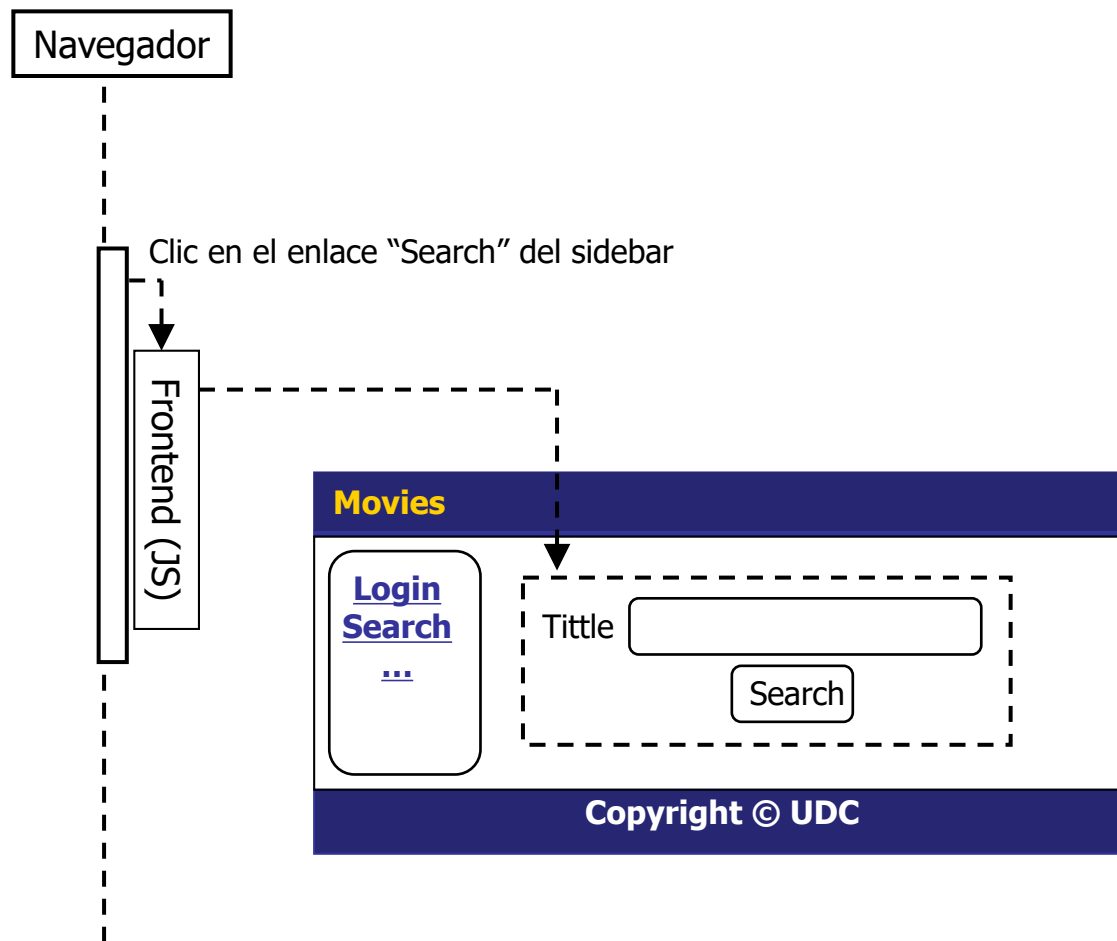
- En una aplicación web del lado servidor, la respuesta a cada petición HTTP incluye todo el HTML que tiene que visualizar el navegador
 - La respuesta a la búsqueda de películas devuelve la cabecera, el sidebar, el formulario, la tabla con las películas y el pie de página
 - Y en realidad, lo único que ha variado con respecto a lo que estaba visualizando el usuario es el área que contiene la información sobre las películas
- Sería más eficiente e interactivo devolver sólo los datos correspondientes a la parte que ha variado
 - Ejemplo: que la aplicación web sólo devuelva los datos de las películas que concuerdan con la búsqueda y que cuando lleguen al navegador, de alguna manera, se inserten en una tabla en la página que está visualizando el usuario

- Una **aplicación web SPA (Single Page Application)** es aquella cuya capa IU está implementada en JavaScript y corre dentro del navegador
- La capa IU (**frontend**) accede a la capa Modelo a través de un servicio (**backend**), normalmente REST/JSON
- Backend
 - Se instala en un servidor de aplicaciones
- Frontend
 - Se puede instalar en un servidor web que sirva contenido estático
 - En el caso más sencillo consta de: index.html + código JavaScript + CSS/imágenes/etc.
 - index.html
 - Página HTML (muy sencilla) devuelta por GET /url-aplicación
 - Contiene el HTML mínimo de la IU (e.g. <head> con <link href=...> a un fichero CSS y <body> con <script src=...> a un fichero con el código JavaScript)
 - El código JavaScript genera dinámicamente los elementos de la interfaz a medida que el usuario interactúa con ella, modificando directamente el árbol DOM de la (única) página que visualiza el navegador

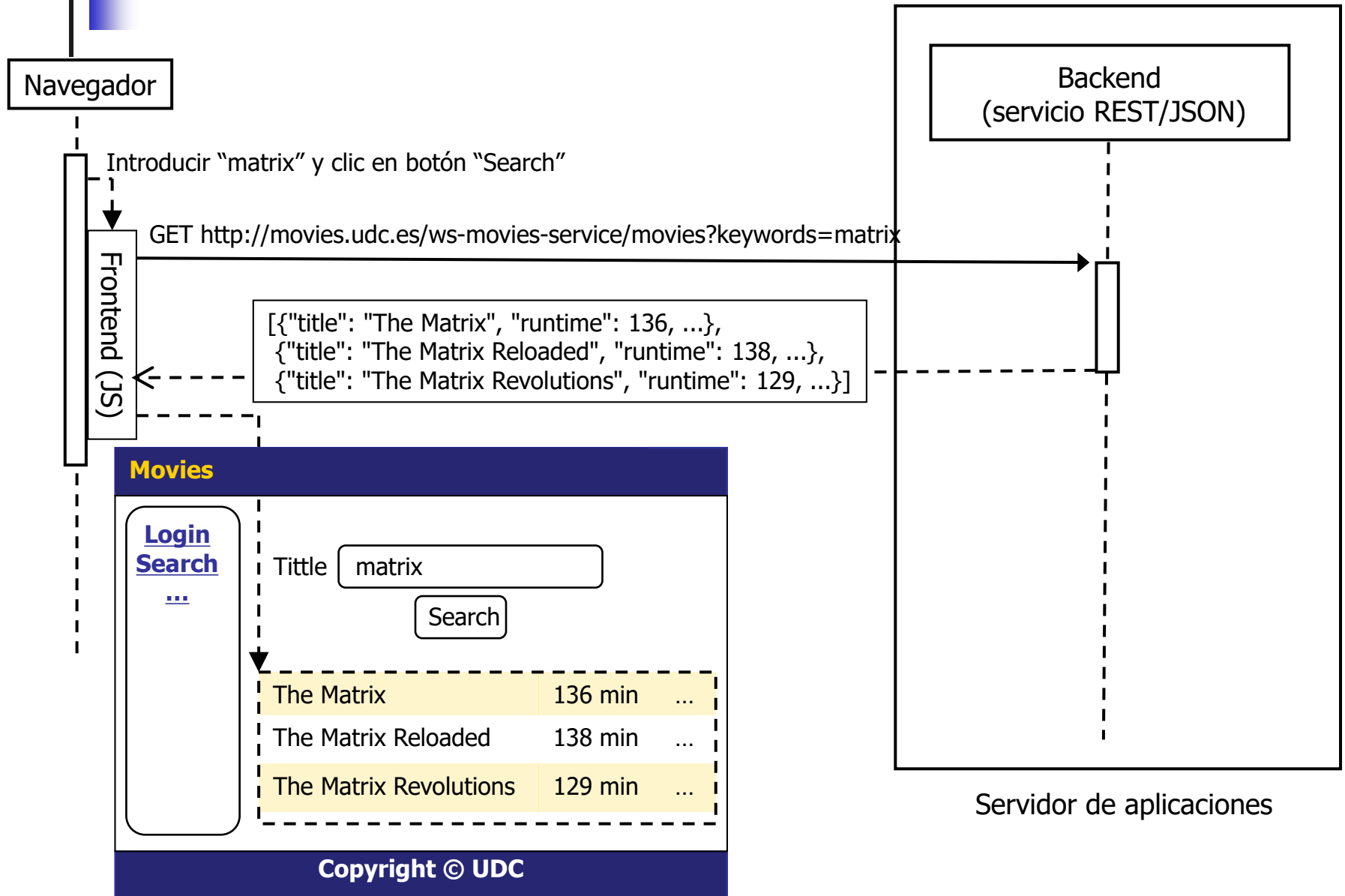
[Aplicaciones web SPA] Modelo de interacción



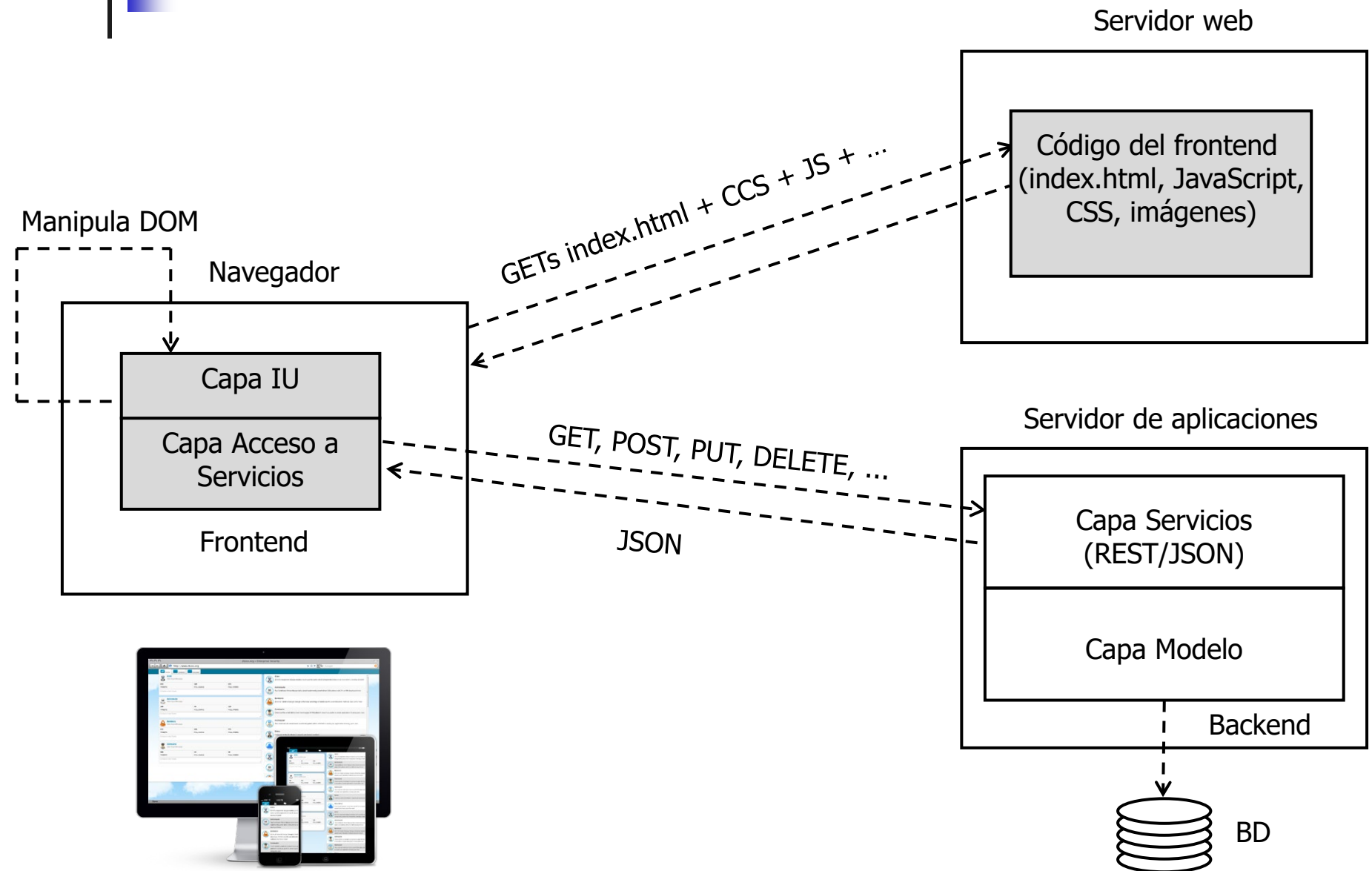
[Aplicaciones web SPA] Modelo de interacción



[Aplicaciones web SPA] Modelo de interacción



[Aplicaciones web SPA] Arquitectura típica



■ Modelo de ejecución asíncrono

- Las operaciones de E/S en JavaScript se ejecutan de forma asíncrona
- Después de invocar una operación de E/S, el hilo de ejecución del navegador continúa sin esperar a que se termine la ejecución de la operación
- Una vez que termine la ejecución de la operación, se notifica al llamador invocando una función con el resultado de la operación
- Ejemplo

```
fetch('http://.../ ws-movies-service/movies?keywords=matrix')  
  .then(function(response) {<<process response>>})  
  .catch(function(error) {<<process error>>})
```

Se invoca cuando llegue la respuesta

Se invoca si hay un fallo de red

- A este estilo de interacción entre el navegador (frontend) y lado servidor (backend), característico de las aplicaciones web SPA, se le acuñó originalmente con el término AJAX (**A**ynchronous **J**avaScript + **X**ML)
 - Sin embargo, lo normal hoy es usar JSON como lenguaje de intercambio por su facilidad de conversión a/desde un objeto JavaScript
- Algunas aplicaciones web del lado servidor incluyen algunas interacciones AJAX en sitios donde una interacción clásica ofrecería una experiencia de usuario demasiado pobre

- La capa IU de la aplicación web tiene que
 - Reaccionar a los eventos producidos por las interacciones del usuario
 - Realizar peticiones HTTP al backend (a través de la capa de Acceso a Servicios)
 - Modificar el árbol DOM de la página del navegador para incluir los resultados
- Existen múltiples frameworks para desarrollar ágilmente la capa IU de una aplicación SPA
 - Los más populares: Angular (Google), React (Facebook) y Vue
- Además, todos los lenguajes populares disponen de frameworks para el desarrollo ágil de servicios REST
 - Ejemplo en Java: Spring Framework