

## DESARROLLO DE APLICACIONES WEB AVANZADO

### LABORATORIO N° 9 INTRODUCCIÓN NEXT JS



Alumnos		Nota
Grupo		
Fecha de Entrega		
Docente	Omar Castañeda Rodríguez	

## OBJETIVOS:

Comprender el funcionamiento básico de Next.js.

Diferenciar entre CSR (Client Side Rendering) y SSR (Server Side Rendering).

## SEGURIDAD:



### Advertencia:

**En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.**

## FUNDAMENTO TEÓRICO:

- Revisar el texto guía que está en el campus Virtual.

## NORMAS EMPLEADAS:

- No aplica

## RECURSOS:

- En este laboratorio cada alumno trabajará con un equipo con Windows 11.

## METODOLOGÍA PARA EL DESARROLLO DE LA TAREA:

- El desarrollo del laboratorio es individual

## PROCEDIMIENTO:

### Requisitos Previos

- Tener instalado Node.js (<https://nodejs.org>)
- Tener instalado npm o yarn
- Editor de código como Visual Studio Code

### Ejercicio 1: Hola Mundo en Next.js

1. Abre Visual Studio Code.
2. Crea una nueva carpeta: nextjs-lab
3. Abre la terminal en esa carpeta:

```
npx create-next-app@latest
```

4. Acepta los valores de la siguiente forma:

```
✓ What is your project named? ... hello-world
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like your code inside a `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to use Turbopack for `next dev`? ... No / Yes
✓ Would you like to customize the import alias (`@/*` by default)? ... No / Yes
```

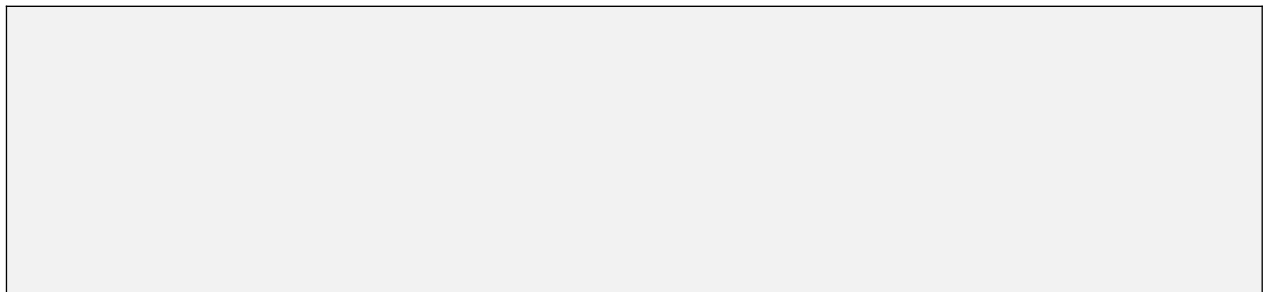
5. Ingresa al proyecto creado e inicia el servidor:

```
cd hello-world
npm run dev
```

6. Luego nos ubicamos en nuestro navegador e ingresamos a **localhost:3000**  
7. Modifica el archivo `src/app/page.tsx` y coloca el siguiente código:

```
1
2 <ol className="list-inside list-decimal text-sm/6 text-center sm:text-left font-[family-name:var(--font-geist-mono)]"&>
3   <li className="mb-2 tracking-[-.01em]">
4     Get started by editing(" ")
5     <code className="bg-black/[.05] dark:bg-white/[.06] px-1 py-0.5 rounded font-[family-name:var(--font-geist-mono)] font-semibold">
6       src/app/page.tsx
7     </code>
8   </li>
9
10  <li className="tracking-[-.01em]">
11    Save and see your changes instantly.
12  </li>
13  <li className="tracking-[-.01em]">
14    Hello world!.
15  </li>
16 </ol>
```

Captura del resultado



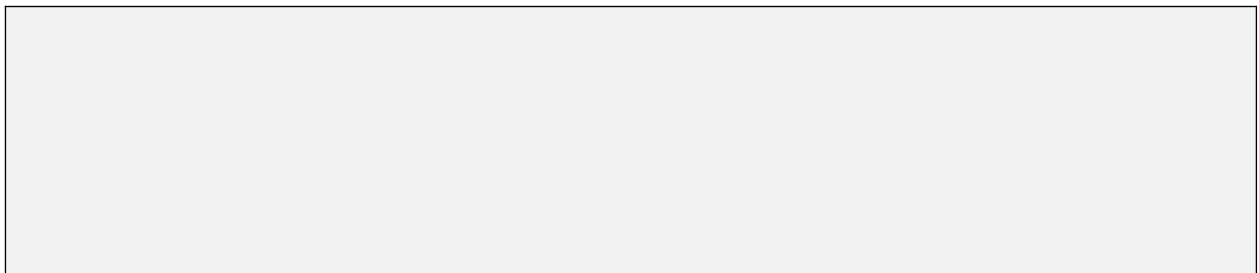
## Ejercicio 2: CSR (Client-Side Rendering)

Crear un componente que muestre la hora actual y se actualice al hacer clic.

1. Crea un nuevo proyecto con next
2. Acepta los valores por defecto
3. Modifica el archivo `src/app/page.tsx` y coloca el siguiente código:

```
1  'use client';
2
3  import { useState } from 'react';
4
5  export default function CSRPage() {
6    const [hora, setHora] = useState(new Date().toLocaleTimeString());
7
8    const actualizarHora = () => {
9      setHora(new Date().toLocaleTimeString());
10   };
11
12   return (
13     <main className="p-8">
14       <h1 className="text-2xl font-bold mb-4">Renderizado en el Cliente (CSR)</h1>
15       <p>Hora actual: {hora}</p>
16       <button
17         onClick={actualizarHora}
18         className="mt-4 px-4 py-2 bg-blue-500 text-white rounded"
19       >
20         Actualizar hora
21       </button>
22     </main>
23   );
24 }
```

Captura



### Ejercicio 3: SSR (Server-Side Rendering)

Mostrar la hora actual renderizada desde el servidor.

1. Crea un nuevo proyecto con next
2. Acepta los valores por defecto
3. Modifica el archivo src/app/**page.tsx** y coloca el siguiente código:

```
1 'use client';
2
3 import { useState } from 'react';
4
5 export default function CSBPage() {
```

```
17     onClick={actualizarHora}
18     className="mt-4 px-4 py-2 bg-blue-500 text-white rounded"
19   >
20     Actualizar hora
21   </button>
22 </main>
23 );
24 }
```

Captura

## TAREA

### Lista de Productos con Filtro por Categoría (CSR)

Objetivo:

- Mostrar una lista de productos.
- Filtrar productos por categoría.
- Usar useState y useEffect.

Instrucciones:

- Crea una página en /app/tarea/page.tsx
- Muestra una lista de productos usando CSR.
- Agrega un botón que permita filtrar los productos por categoría.
- Usa al menos un useState y un useEffect.

***Colocar capturas del código y capturas de ejecución***

***Grabar explicando el código y ejecución de este en máximo 4 minutos***

**OBSERVACIONES:** *(Las observaciones son las notas aclaratorias, objeciones y problemas que se pudo presentar en el desarrollo del laboratorio)*

- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_

**CONCLUSIONES:** *(Las conclusiones son una opinión sobre tu trabajo, explicar cómo resolviste las dudas o problemas presentados en el laboratorio. Además de aportar una opinión crítica de lo realizado)*

- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_
- \_\_\_\_\_  
\_\_\_\_\_