

**COMPONENTE FRONTEND DEL PROYECTO FORMATIVO Y PROYECTOS DE  
CLASE (LISTAS DE CHEQUEO)**

**GA7-220501096-AA4-EV03**

**DANIEL ALEJANDRO TIJARO SANTOS  
JENNIFER ADRIANA TIJARO SANTOS**

**PROGRAMA ANALISIS Y DESARROLLO DE SOFTWARE-2664877  
SERVICIO NACIONAL DE APRENDIZAJE-SENA**

## Contenido

<b>Introducción.....</b>	<b>3</b>
<b>Objetivo.....</b>	<b>3</b>
<b>Desarrollo .....</b>	<b>4</b>
<b>Artefactos del Ciclo del Software .....</b>	<b>4</b>
<b>Estructura del Proyecto.....</b>	<b>4</b>
<b>Implementación de Componentes .....</b>	<b>5</b>
<b>Componente Header .....</b>	<b>5</b>
<b>Uso de Herramientas de Versionamiento.....</b>	<b>6</b>
<b>Lista de chequeo.....</b>	<b>6</b>
<b>Conclusiones .....</b>	<b>8</b>

# Introducción

El presente documento describe la codificación del módulo frontend del proyecto formativo DISCAPSICO utilizando React JS, como parte del componente "Desarrollo de frontend con React JS". Este proyecto se enfoca en la gestión de datos relacionados con discapacidades psicosociales en Bogotá. La implementación se basa en los artefactos desarrollados previamente, incluyendo diagramas de clases, diagramas de casos de uso, historias de usuario, diseños y prototipos. El objetivo es desarrollar una interfaz de usuario intuitiva y eficiente que cumpla con los requisitos funcionales y de calidad definidos.

## Objetivo

El objetivo de esta evidencia de producto es realizar la codificación del módulo frontend del proyecto DISCAPSICO aplicando los conceptos y prácticas aprendidas en el desarrollo de frontend con React JS. Se busca crear una interfaz de usuario que permita la gestión de información de manera eficaz, cumpliendo con los estándares de codificación y utilizando herramientas de versionamiento.

## Desarrollo

### Artefactos del Ciclo del Software

Antes de iniciar la codificación, es esencial tener en cuenta los siguientes artefactos previamente desarrollados:

- **Diagrama de Clases:** Define la estructura de los datos y las relaciones entre las entidades.
- **Diagramas de Casos de Uso:** Describe las interacciones del usuario con el sistema.
- **Historias de Usuario:** Detalla las funcionalidades desde la perspectiva del usuario.
- **Diseños y Prototipos:** Proveen una guía visual de la interfaz de usuario.
- **Informe Técnico:** Documenta el plan de trabajo para la construcción del software.

### Estructura del Proyecto

El proyecto se ha creado utilizando **Create React App** para establecer un entorno de desarrollo React eficiente. La estructura del proyecto incluye los siguientes directorios y archivos clave:

```
DISCAPSICO/  
├─ public/  
│   └─ index.html  
│   └─ ...  
├─ src/  
│   └─ components/  
│       │   └─ Header.js  
│       │   └─ Footer.js  
│       │   └─ LoginForm.js  
│       │   └─ RegistrationForm.js  
│       │   └─ Dashboard.js  
│       │   └─ ReportSection.js  
│       │   └─ ...  
│   └─ styles/  
│       │   └─ App.css  
│       │   └─ Header.css  
│       │   └─ ...  
│   └─ App.js  
│   └─ index.js  
│   └─ ...  
├─ .gitignore  
├─ package.json  
└─ README.md
```



## Implementación de Componentes

A continuación se presenta un resumen de la implementación de algunos componentes clave del proyecto:

### Componente Header

```
// src/components/Header.js
import React from 'react';
import './Header.css';

const Header = () => {
  return (
    <header className="header">
      <h1>DISCAPSICO</h1>
      <nav>
        <ul>
          <li><a href="/">Home</a></li>
          <li><a href="/create">Crear Registro</a></li>
          <li><a href="/consult">Consultar</a></li>
          <li><a href="/logout">Cerrar Sesión</a></li>
        </ul>
      </nav>
    </header>
  );
}

export default Header;
```

### Componente LoginForm

```
1 // src/components/LoginForm.js
2 import React, { useState } from 'react';
3 import './LoginForm.css';
4
5 const LoginForm = () => {
6   const [email, setEmail] = useState('');
7   const [password, setPassword] = useState('');
8
9   const handleSubmit = (e) => {
10     e.preventDefault();
11     // lógica de autenticación
12     console.log('Email:', email);
13     console.log('Password:', password);
14   }
15
16   return (
17     <div className="login-form">
18       <h2>Inicia Sesión</h2>
19       <form onSubmit={handleSubmit}>
20         <label>
21           Email address:
22           <input type="email" value={email} onChange={(e) => setEmail(e.target.value)} required />
23         </label>
24         <label>
25           Password:
26           <input type="password" value={password} onChange={(e) => setPassword(e.target.value)} required />
27         </label>
28         <button type="submit">Sign In</button>
29       </form>
30       <p>¿Primera vez en DISCAPSICO? <a href="/register">Regístrate ahora.</a></p>
31     </div>
32   );
33 }
```

## Uso de Herramientas de Versionamiento

El proyecto se ha versionado utilizando Git y se ha alojado en GitHub para facilitar la colaboración y el control de versiones. A continuación, se presenta un ejemplo de los comandos básicos utilizados para el versionamiento:

```
# Inicializar un nuevo repositorio Git
git init

# Agregar todos los archivos al repositorio
git add .

# Realizar el primer commit
git commit -m "Initial commit"

# Conectar el repositorio local a un repositorio remoto en GitHub
git remote add origin https://github.com/usuario/discapsico.git

# Subir los cambios al repositorio remoto
git push -u origin master
```

## Lista de chequeo

La lista de chequeo presentada a continuación está diseñada para asegurar que todos los aspectos fundamentales de la codificación del módulo frontend del proyecto DISCAPSICO se han abordado correctamente. Este checklist guiará la revisión del código y la documentación, garantizando la calidad y la conformidad con los estándares de desarrollo.

La lista de chequeo asegura que todos los aspectos clave del desarrollo frontend con React JS se han abordado correctamente. Seguir esta lista ayuda a mantener altos estándares de calidad y facilita la revisión y el mantenimiento del código. Es fundamental revisar cada punto y marcarlo una vez completado para garantizar que el proyecto cumpla con los requisitos y las mejores prácticas del desarrollo de software.

Componente frontend del proyecto formativo y proyectos de clase (listas de chequeo)			
		SI	NO
E	<b>Estructura del Proyecto</b>		
1	¿El proyecto se ha creado utilizando Create React App?		
2	¿El proyecto tiene una estructura clara y organizada?		
3	¿Existe un directorio components para los componentes React?		
4	¿Existe un directorio styles para los archivos CSS?		
5	¿El archivo App.js y index.js están correctamente configurados?		
I	<b>Implementación de Componentes</b>		
6	¿Se han implementado los componentes clave (Header, LoginForm, Dashboard, etc.)?		
7	¿Cada componente tiene su propio archivo y está bien organizado?		
8	¿Los componentes siguen la convención de nombres en PascalCase?		
9	¿Cada componente tiene comentarios explicativos sobre su funcionalidad?		
U	<b>Uso de Props y State</b>		
10	¿Se utilizan props para pasar datos de padres a hijos?		
11	¿Se utiliza el state para gestionar el estado interno del componente?		
12	¿El manejo de state y props está claramente documentado?		
G	<b>Gestión de Eventos</b>		
13	¿Se manejan correctamente los eventos en los componentes?		
14	¿Se han implementado eventos como onClick, onChange, etc.?		
15	¿Los métodos de manejo de eventos están claramente definidos y documentados?		
C	<b>Ciclo de Vida del Componente</b>		
16	¿Se utilizan correctamente los métodos del ciclo de vida del componente?		
17	¿Métodos como componentDidMount, están implementados cuando es necesario?		
ES	<b>Estilos y CSS</b>		
18	¿Los estilos están separados en archivos CSS?		
19	¿Cada componente tiene su propio archivo CSS?		
20	¿Se utilizan nombres de clases claros y consistentes?		
21	¿Los estilos CSS siguen los estándares de la organización?		
US	<b>Uso de Herramientas de Versionamiento</b>		
22	¿El proyecto se ha inicializado con Git?		
23	¿Se ha utilizado un repositorio remoto (por ejemplo, GitHub)?		
24	¿El repositorio tiene commits claros y descriptivos?		
25	¿El historial de commits muestra un desarrollo incremental y bien documentado?		
CU	<b>Cumplimiento de Estándares de Codificación</b>		
26	¿El código sigue los estándares de codificación establecidos?		
27	¿El código está bien indentado y estructurado?		
28	¿Se utilizan comentarios para explicar bloques de código complejos?		
29	¿Se han eliminado los códigos comentados y las líneas innecesarias?		
DC	<b>Documentación</b>		
30	¿Se ha incluido documentación adecuada en el proyecto?		
31	¿Existe un README que explique cómo ejecutar el proyecto?		
32	¿La documentación describe los componentes y su funcionalidad?		
33	¿Hay instrucciones sobre cómo contribuir al proyecto?		

## Conclusiones

El desarrollo del módulo frontend del proyecto DISCAPSICO utilizando React JS ha permitido la implementación de una interfaz de usuario moderna, eficiente y fácil de usar. La codificación basada en los artefactos del ciclo de vida del software ha asegurado que el diseño cumpla con los requisitos funcionales y de calidad. El uso de herramientas de versionamiento ha facilitado el control de versiones y la colaboración. Este trabajo ha proporcionado una experiencia práctica valiosa en el desarrollo de aplicaciones frontend, preparando a los participantes para futuros proyectos en el ámbito profesional.