



Agencia de Habilidades
para el Futuro

<Talento
Tech />

React JS

Clase 15 | Despliegue de la Aplicación

¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase.

Índice

Estilización con Bootstrap o styled-components

- **Introducción a Bootstrap o styled-components para estilizar componentes.**
- **Creación de un diseño básico y responsive.**
- **Aplicación de estilos en componentes (botones, formularios, productos).**

Despliegue de la Aplicación

- Despliegue de la aplicación en plataformas como Vercel o Netlify.
- Pruebas de funcionamiento en el entorno de producción.
- Revisión de los pasos de despliegue y configuración.

Revisión Final y Presentación del Proyecto

- Despliegue de la aplicación final.
- Revisión de la integración de todas las funcionalidades desarrolladas (CRUD, autenticación, carrito, rutas protegidas).

Objetivos de la Clase

1 Publicar en la Nube

Desplegar la aplicación en Vercel o Netlify.

2 Verificar Funcionamiento

Asegurar que la app funcione correctamente en producción.

3 Revisar Configuración

Analizar los pasos para un despliegue exitoso.

Despliegue de la Aplicación en Vercel o Netlify

Preparación para el Despliegue

Una vez que la aplicación está lista y optimizada, el siguiente paso es publicarla en un servidor para que cualquier usuario pueda acceder a ella.

Antes de desplegar, es importante asegurarse de que el código está limpio y sin errores.

Revisar Código

Asegurar que todo funcione en el entorno local.

Instalar Dependencias

Ejecutar npm install para actualizar módulos.

Configurar Variables

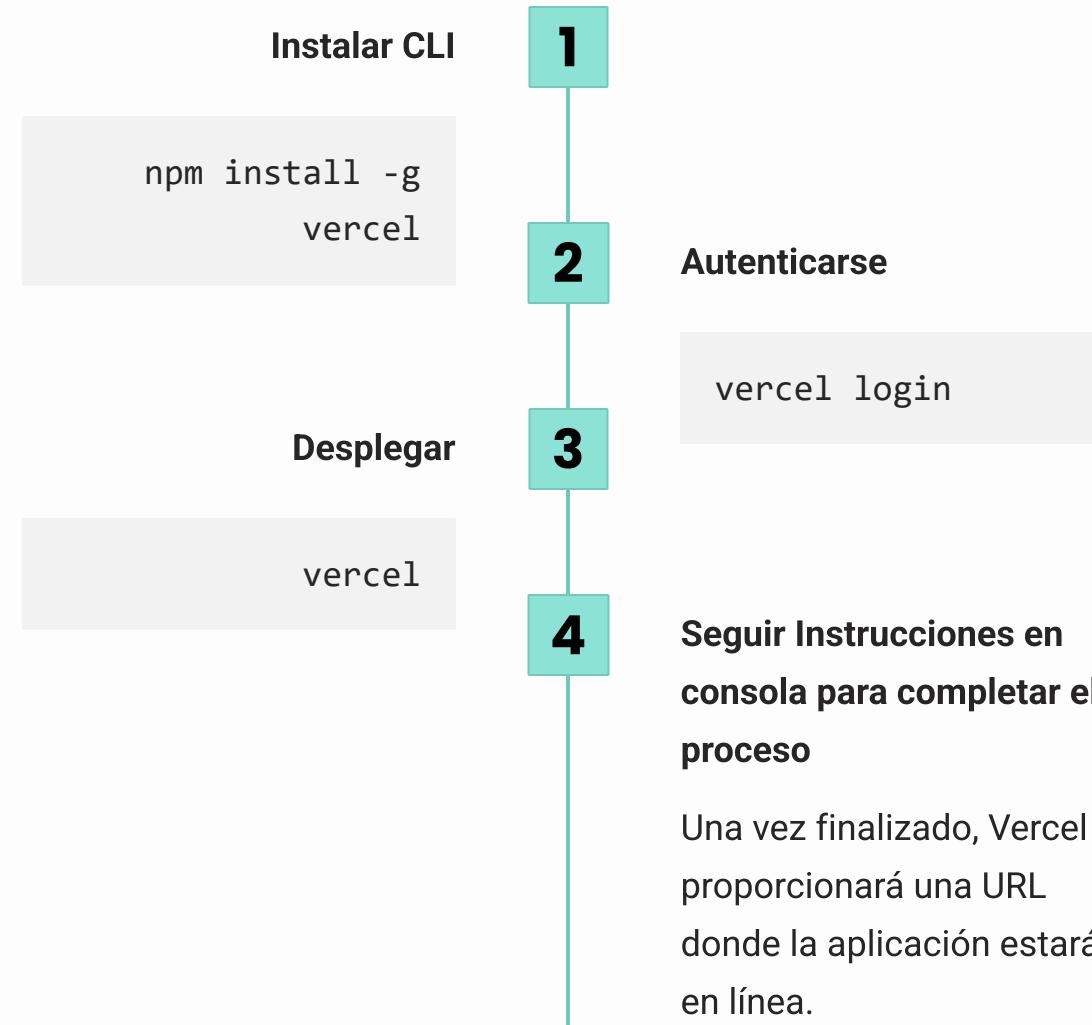
Establecer correctamente las variables de entorno.

Commit Final

Realizar un último commit en GitHub.



Despliegue en Vercel



Despliegue en Netlify.



Pruebas en Producción

Pruebas en Producción

1 Cargas de Página

Verificar que todas las rutas funcionen correctamente.

2 Responsividad

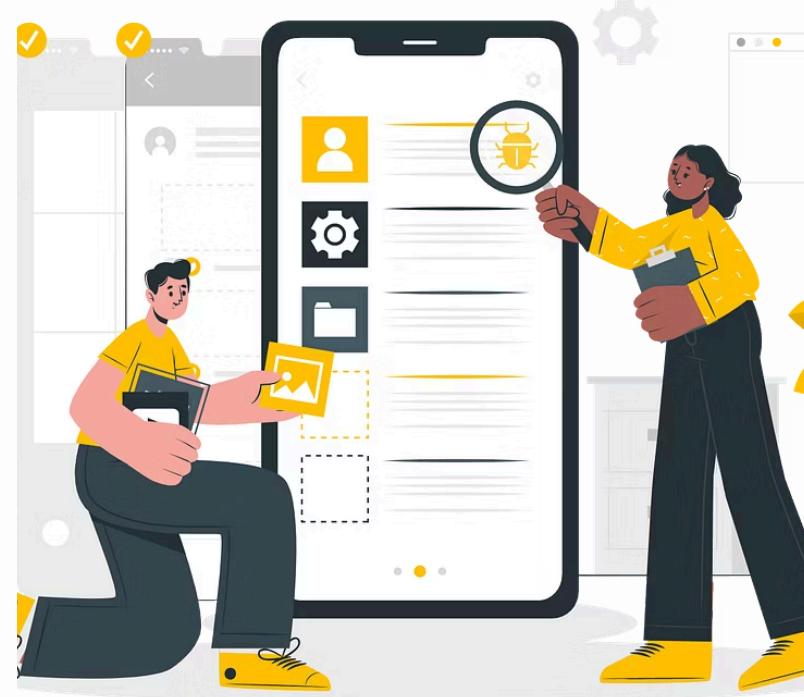
Probar en diferentes dispositivos y navegadores.

3 Interacciones

Validar formularios, botones y funcionalidades.

4 Rendimiento

Analizar tiempos de carga con DevTools.



Herramientas de Análisis



[Google Lighthouse](#)

Evalúa rendimiento y accesibilidad.



[PageSpeed Insights](#)

Analiza velocidad de carga.



[DevTools Responsive](#)

Simula diferentes dispositivos.

Revisión final y configuración

Errores Comunes y Soluciones

App No Carga

- ✓ Revisar que el build (npm run build) se haya generado correctamente.
- ✓ Asegurar que la carpeta de salida (dist o build) es la correcta en la configuración del hosting.

Problemas de Rutas

- ✓ En Vercel y Netlify, asegurarse de usar react-router-dom en modo BrowserRouter con una configuración adecuada.
- ✓ Para Netlify, agregar un archivo _redirects en public/ con el siguiente contenido:

```
/* /index.html 200
```

Variables de Entorno

- ✓ En Vercel o Netlify, configurar manualmente las variables en la sección de "Environment Variables"

Reflexión Final

Aprendimos a desplegar nuestra aplicación de React en plataformas como Vercel y Netlify, asegurándonos de que funcione correctamente en un entorno de producción.

Realizamos pruebas para verificar la responsividad, la accesibilidad y el rendimiento, asegurando una experiencia fluida para el usuario final.

El despliegue es un paso fundamental en el desarrollo, ya que permite que el proyecto pase de una fase local a estar accesible en Internet, listo para su uso real.



Recursos Adicionales



Vercel Docs

Documentación oficial de Vercel.



Netlify Docs

Guías y tutoriales de Netlify.



React Router

Configuración de rutas en React.



Google Lighthouse

Preguntas para Reflexionar

1 Vercel vs Netlify

¿Qué ventajas tiene cada plataforma?

2 Problemas de Rutas

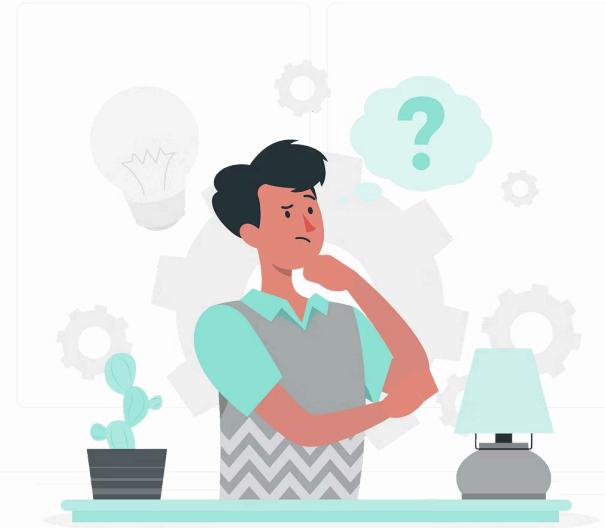
¿Cómo solucionarías issues con React Router DOM?

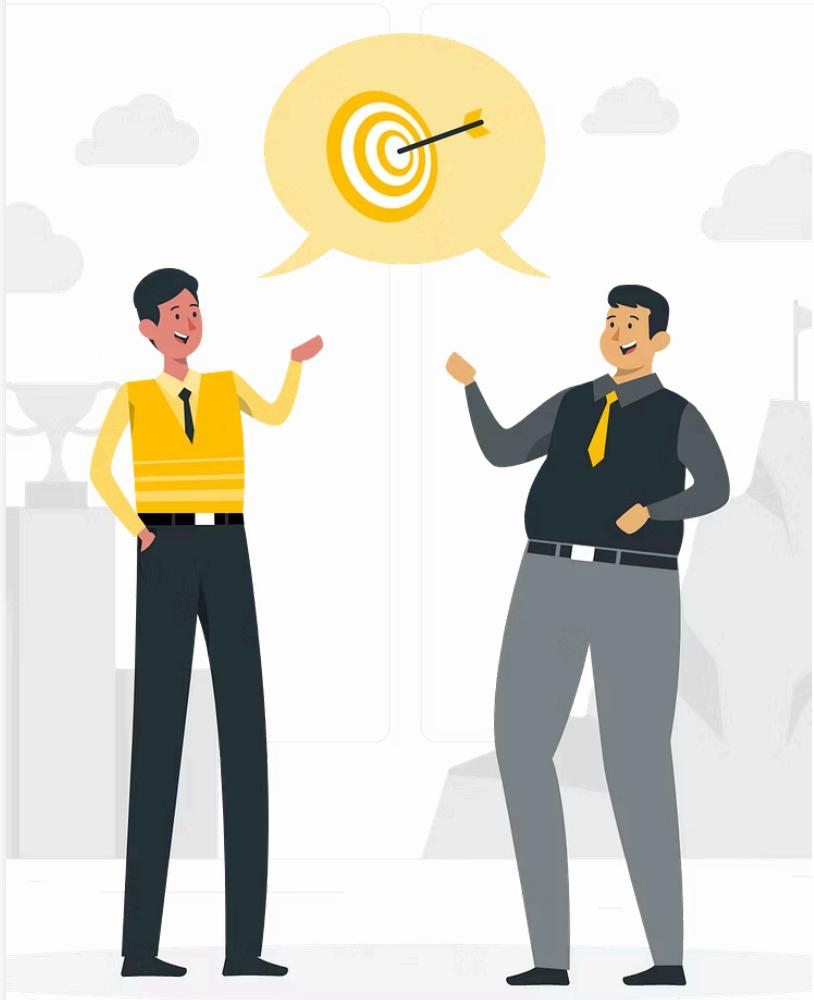
3 Importancia de Pruebas

¿Por qué son cruciales antes de publicar?

4 Optimización de Velocidad

¿Qué medidas tomarías para mejorar la carga?





Próximos Pasos

- 1
- 2
- 3
- 4

Despliegue Final

Publicar la aplicación completa en producción.

Revisión Integral

Verificar todas las funcionalidades desarrolladas.

Monitoreo Continuo

Implementar herramientas de seguimiento y análisis.

Mejora Continua

Planificar actualizaciones y optimizaciones futuras.



Consignas

Entrega Final del Proyecto



Entrega Final del Proyecto - Consignas

¡Hemos llegado a la entrega final del proyecto! Es el momento de presentar la versión completa y optimizada del eCommerce que hemos desarrollado. Como en cualquier entorno profesional, esta entrega simula la presentación del producto final a nuestro cliente de Talento Lab. Se espera que el sitio cumpla con todos los requerimientos establecidos y brinde una experiencia de usuario fluida, accesible y visualmente atractiva.



Entrega Final

A continuación, se detallan los puntos que deben estar implementados para la entrega final:

📌 Requerimiento #1: Gestión del Carrito y Autenticación de Usuarios

Objetivo: Implementar un carrito de compras funcional y restringir el acceso a secciones privadas mediante autenticación de usuarios.

Carrito de Compras con Context API

- Implementar un **CarritoContext** que gestione los productos agregados.
- Permitir **agregar, eliminar y vaciar el carrito**.
- Mantener el estado global con Context API.

Autenticación de Usuarios

- Crear un **AuthContext** para manejar el estado de autenticación.
- Implementar un **login simulado con localStorage**.
- Restringir el acceso al carrito y otras secciones a usuarios autenticados con **rutas protegidas**.



Entrega Final

📌 Requerimiento #2: CRUD de Productos con MockAPI

Objetivo: Permitir la administración completa del catálogo de productos mediante operaciones de creación, lectura, actualización y eliminación.

Formulario para Agregar Productos

- Implementar un **formulario controlado con useState**.
- Validar que los campos sean correctos:
 - **Nombre obligatorio.**
 - **Precio mayor a 0.**
 - **Descripción mínima de 10 caracteres.**
- Enviar los datos a **MockAPI** mediante una **solicitud POST**.

Edición y Eliminación de Productos

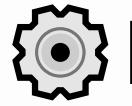
- Permitir la edición de productos utilizando **MockAPI** y **Context API**.
- Mostrar **mensajes de error y confirmaciones** al usuario.

Implementar un **modal de confirmación** antes de eliminar un producto.

Manejo de Errores

- Mostrar **mensajes de error** en pantalla si hay problemas con la API.

Manejar estados de **carga y error** al obtener los productos.



Entrega Final

📌 Requerimiento #3: Optimización de Diseño y Responsividad

Objetivo: Mejorar la apariencia y la accesibilidad del sitio utilizando herramientas modernas de diseño y estilización.

Diseño Responsivo con Bootstrap y Styled-components

- Implementar el **sistema de grillas de Bootstrap** para adaptar el contenido a distintos dispositivos.
- Usar **styled-components** para personalizar los estilos y hacer el código más modular.

Interactividad Mejorada con React Icons y React Toastify

- Agregar **iconos** en botones y elementos interactivos con **React Icons**.

Implementar **React Toastify** para mostrar **notificaciones de éxito y error**.

SEO y Accesibilidad con React Helmet

- Modificar el **<title>** y **<meta>** con React Helmet para mejorar el SEO.

Asegurar que los elementos interactivos tengan etiquetas **ARIA** para accesibilidad.



Entrega Final

A continuación, se detallan los puntos que deben estar implementados para la entrega final:

📌 Requerimiento #4: Funcionalidades de Búsqueda y Paginación

Objetivo: Mejorar la usabilidad y navegación del catálogo de productos.

Barra de Búsqueda

- Implementar una barra de búsqueda que permita a los usuarios filtrar los productos por nombre o categoría.
- Asegurar que la búsqueda sea rápida y eficiente, mostrando los resultados conforme el usuario escribe.v

Paginador de Productos

- Implementar un paginador que divida los productos en varias páginas.
- Asegurar que los usuarios puedan navegar entre las páginas sin problemas, mejorando la experiencia de usuario.



Entrega Final

A continuación, se detallan los puntos que deben estar implementados para la entrega final:

📌 Requerimiento #5: Preparación para el Despliegue

Objetivo: Ajustar los últimos detalles para que la aplicación esté lista para ser publicada en un servidor.

Pruebas de Compatibilidad

- Verificar el funcionamiento en **móviles, tablets y escritorios.**

Revisar **tiempos de carga** y **experiencia de usuario.**

Optimización del Código

- Revisar el código y eliminar elementos innecesarios.

Asegurar que el **estado global** esté **bien gestionado.**

Documentación Básica

Incluir instrucciones en el **[README.md](#)** sobre instalación y uso de la aplicación.



✖️ **IMPORTANTE:**

Esta entrega representa el cierre del proyecto y debe reflejar un producto funcional, bien estructurado y con una experiencia de usuario optimizada. ¡Éxitos en la presentación final! 🚀

