



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL RESISTENCIA
Ingeniería en Sistemas de Información

Desarrollo de Software
Trabajo Práctico Integrador

Elaborado por grupo: 1 Año: 3ro

Integrantes:

Intile, Agustin

Correo: balbuenaagustin11@gmail.com

Machado Benitez, Matias Emmanuel

Correo: machadobenitezmatias88@gmail.com

Martin Rodich, Celeste

Correo: cmartinrodich@gmail.com

Suarez, Juan Pablo

Correo: juanpbsrz@gmail.com

Velázquez, María Victoria

Correo: vicvelazquez002019@gmail.com

Zappa, Estefano

Correo: zappaestefano18@gmail.com

Zibelman, Irina

Correo: irinazibelman@gmail.com

Profesores:

Villaverde, Jorge Eduardo

Fernandez, Jose Alejandro

FECHA DE PRESENTACIÓN

05/12/24

ÍNDICE

INTRODUCCIÓN.....	4
DESARROLLO.....	5
Canva.....	5
Trello.....	5
Cómo se utiliza Trello en el desarrollo de aplicaciones:.....	5
1. Planificación del Proyecto:.....	5
2. Gestión de Tareas:.....	5
3. Gestión de la Prioridad:.....	5
4. Revisión y Retroalimentación:.....	6
5. Integración con Otras Herramientas:.....	6
6. Seguimiento del Progreso:.....	6
Beneficios que encontramos al usar Trello en desarrollo del sitio web:.....	6
Node.js.....	6
Características principales:.....	6
React.....	8
¿Por qué usamos React?.....	8
¿Para qué sirve?.....	8
Uso de React en el Proyecto.....	8
Cómo se está utilizando React en este caso:.....	8
Componentes Importados:.....	8
Definición del Componente Principal:.....	8
Contenido de la Página:.....	9
Ejemplo de cómo React aporta valor en el proyecto:.....	9
Sin React (HTML estático):.....	9
Con React:.....	9
Next.js.....	10
Características principales:.....	10
¿Para qué se utiliza?.....	10
Strapi.....	11
Características Principales:.....	11
¿Para que se usa?.....	11
Render.....	11
Características Principales:.....	11
¿Para qué se usa?.....	12
Vercel.....	12
NeonDB.....	12
Características Principales:.....	12
¿Para qué se usa?.....	13
Cloudinary.....	13
Características Principales:.....	13
¿Para que se utiliza?.....	13
Expandiendo los Problemas del Desarrollo del Software.....	14
1. Dificultades con Git.....	14

2. Errores de código en Next.js.....	14
3. Desafíos con el diseño y CSS.....	14
4. Inconvenientes con la API y Strapi.....	14
5. Errores de carga en las páginas.....	14
6. Dudas sobre el despliegue del proyecto.....	14
CONCLUSIÓN.....	15
BIBLIOGRAFÍA.....	16

INTRODUCCIÓN

En el presente documento se expondrá las tecnologías utilizadas para la resolución del Trabajo Práctico Integrador de Desarrollo de Software. El escenario planteado se basa en la Bienal Internacional de Escultura del Chaco. La organización de dicho evento, requiere de la planificación, análisis, desarrollo e implementación de un sistema que ayude a gestionar los eventos, escultores y opiniones del público.

DESARROLLO

A lo largo del trabajo se necesito de distintas herramientas, cada una con una finalidad específica. Las elecciones se basaron en el conocimiento con el que contaba el equipo, si por alguna otra experiencia se estaba vinculando a dicha herramienta, y el aporte que podía realizar la misma al desarrollo del sistema de gestión.

Al inicio del proyecto se habían arrojado varias posibilidades, se investigaron y las seleccionadas fueron: **Canva**, **Trello**, **Node.js**, **Javascript React**, **Next.js**, **Strapi**, **GitHub**. Durante el proyecto se fueron agregando otras tecnologías, en función de las necesidades que surgían. A continuación se detallan las siguientes características y usos que fuimos dando a las tecnologías ya mencionadas.

Canva

Canva fue una herramienta usada inicialmente para crear un modelo de la página, tomando en cuenta las secciones que deseábamos tener, la navegación de la misma, los elementos a crear y el estilo global a seguir. A partir de este diseño se fueron discutiendo cambios y agregando elementos hasta llegar al proyecto final.

Trello

Trello es una herramienta de gestión de proyectos visual basada en el método Kanban. Es ampliamente utilizada en entornos de desarrollo de aplicaciones para organizar tareas, colaborar con equipos y mantener un seguimiento eficiente del progreso del proyecto. A continuación se detallan las formas más comunes en las que Trello se utiliza en el desarrollo de aplicaciones.

Cómo se utiliza Trello en el desarrollo de aplicaciones:

1. Planificación del Proyecto:

- Al inicio de un proyecto, se crea un tablero de Trello donde se definen las listas para las tareas principales y secundarias.
- Las tareas se desglosan en tarjetas detalladas que describen las funcionalidades o características que se deben desarrollar.

2. Gestión de Tareas:

- Los desarrolladores, diseñadores y otros miembros del equipo asignan tareas a través de las tarjetas y las mueven entre las listas para indicar su estado (p. ej., de "To Do" a "In Progress").
- Las tareas se pueden subdividir en subtareas utilizando listas de verificación dentro de las tarjetas, lo que facilita el seguimiento de los detalles.

3. Gestión de la Prioridad:

- Las tarjetas se pueden etiquetar con colores para indicar la prioridad (alta, media, baja).

- Esto ayuda a los equipos a enfocar sus esfuerzos en las tareas más críticas y a asegurarse de que las funciones esenciales se completan a tiempo.

4. Revisión y Retroalimentación:

- Los miembros del equipo pueden agregar comentarios a las tarjetas y etiquetar a otros miembros para realizar revisiones o proporcionar retroalimentación sobre el trabajo.
- Al finalizar una tarea, se mueve a la lista **Review** y, una vez aprobada, se mueve a **Done**.

5. Integración con Otras Herramientas:

- Trello se integra fácilmente con otras herramientas populares en el desarrollo de aplicaciones, como **GitHub**, **Slack** y **Jira**. Esto permite tener actualizaciones automáticas de commits, pull requests y tareas relacionadas.

6. Seguimiento del Progreso:

- A medida que las tarjetas se mueven a través de las listas, el equipo puede ver visualmente el progreso del proyecto y asegurarse de que no haya cuellos de botella en el flujo de trabajo.
- Las fechas de vencimiento ayudan a cumplir con los plazos establecidos, mientras que las etiquetas y checklists mantienen el trabajo organizado y enfocado.

Beneficios que encontramos al usar Trello en desarrollo del sitio web:

- **Visibilidad:** Proporciona una visión clara y organizada del progreso del proyecto.
- **Colaboración:** Facilita la comunicación y coordinación entre los miembros del equipo.
- **Flexibilidad:** Se adapta a metodologías ágiles como Scrum o Kanban.
- **Simplicidad:** Fácil de usar, no requiere una curva de aprendizaje compleja.

Node.js

Node.js es un entorno de ejecución de JavaScript construido sobre el motor V8 de Google, diseñado principalmente para el desarrollo backend.

Características principales:

Modelo asíncrono: Se basa en un modelo de programación no bloqueante y orientado a eventos, lo que lo hace altamente eficiente para manejar múltiples solicitudes al mismo tiempo.

- **Monohilo:** Aunque utiliza un solo hilo para ejecutar JavaScript, su arquitectura permite realizar tareas en paralelo gracias al uso de un bucle de eventos y el soporte de librerías subyacentes.
- **Rendimiento:** Es ideal para construir aplicaciones de red rápidas, escalables y aplicaciones web en tiempo real, como chats o sistemas de colaboración en línea.

Con Node.js, en este proyecto utilizamos JavaScript tanto en el frontend como en el backend, unificando el lenguaje en todo el flujo de desarrollo de la aplicación.

Para un mejor manejo de tecnologías y aprovechamiento de tiempo utilizamos dos frameworks muy conocidos que se complementan: **React** y **Next.js**.

React

React es una **biblioteca de JavaScript** desarrollada por Facebook que se utiliza para construir interfaces de usuario dinámicas y componentes reutilizables en aplicaciones web.

¿Por qué usamos React?

Componentes reutilizables: React permite dividir una interfaz en piezas más pequeñas (componentes) que pueden ser reutilizadas en diferentes partes de la aplicación.

1. **Actualizaciones rápidas:** Gracias a su **DOM virtual**, React actualiza únicamente las partes de la página que cambian, mejorando el rendimiento.
2. **Reactivo y dinámico:** Hace que las aplicaciones respondan a cambios en tiempo real, como actualizaciones de datos o interacciones del usuario.
3. **Facilidad para trabajar con datos:** Maneja estados y propiedades (props) para mostrar contenido dinámico.

¿Para qué sirve?

- Crear **aplicaciones web modernas** como sitios interactivos, dashboards y tiendas en línea.
 - Construir **interfaces complejas** en proyectos grandes manteniendo el código organizado y escalable.
-

Uso de React en el Proyecto

El archivo “page.jsx” está escrito en **React**, específicamente usando la sintaxis de **JSX** (JavaScript XML), que es propia de este framework. React se utiliza aquí para estructurar y renderizar componentes de una página web de manera modular y dinámica.

Cómo se está utilizando React en este caso:

Componentes Importados:

- Se están importando componentes como **ImageBar** y **InfoExtra** desde otras rutas del proyecto. Esto sugiere una estructura modular, donde diferentes partes de la página están separadas en archivos individuales.
- También se importan estilos CSS desde un archivo llamado page.module.css, lo que indica que se está utilizando **CSS modular** para el diseño.

Definición del Componente Principal:

- **Home** es un componente funcional que representa la página principal de esta aplicación.

- Contiene elementos HTML estructurados, como encabezados (<h2>, <h1>), párrafos (<p>), secciones (<section>) y un contenedor principal (<main>).
- Se utiliza una clase CSS dinámica (className={styles.algo}) para aplicar estilos a los elementos, lo cual es común en aplicaciones React con CSS modular.

Contenido de la Página:

- La página incluye información relacionada con la **Bienal de Esculturas de Chaco**, un evento cultural. Esto se ve en las citas y descripciones dentro del componente.

Ejemplo de cómo React aporta valor en el proyecto:

Sin React (HTML estático):

En una página HTML tradicional, todo el contenido estaría fijo. Si se quisiera actualizar un componente como el encabezado o añadir una nueva sección, habría que modificar el HTML directamente en varios lugares.

Con React:

React permite crear **componentes reutilizables** como **ImageBar** y **InfoExtra**, que pueden integrarse en cualquier parte de la aplicación sin duplicar código. Por ejemplo:

- **ImageBar** podría ser un “carrusel de imágenes” reutilizable en otras páginas.
 - **InfoExtra** podría ser un bloque de información adicional que se muestra condicionalmente según la interacción del usuario.
-

Next.js

Next.js es un **framework de desarrollo web** basado en **React**, creado por **Vercel**, que permite construir aplicaciones web modernas con un enfoque en rendimiento, escalabilidad y experiencia del desarrollador.

Características principales:

1. Renderizado híbrido y optimizado:

- **Renderizado en el servidor (SSR):** Genera el HTML dinámicamente en el servidor antes de enviarlo al cliente.
- **Generación estática (SSG):** Pre-renderiza páginas durante la compilación para mejorar la velocidad y el SEO.
- **Incremental Static Regeneration (ISR):** Permite generar páginas estáticas de forma incremental tras su despliegue.

2. Rendimiento mejorado:

- Optimización automática de imágenes.
- Soporte para división de código (code-splitting) y carga diferida (lazy-loading).
- Gestión inteligente de rutas.

3. Soporte completo para API:

- Puedes crear rutas API directamente dentro de la estructura de carpetas.

4. Experiencia del desarrollador:

- Desarrollo rápido con recarga en tiempo real (hot-reloading).
- TypeScript integrado y soporte para CSS Modules y estilos globales.

5. SEO amigable:

- Genera contenido fácilmente indexable por motores de búsqueda gracias al pre-renderizado.

6. Middleware y routing avanzado:

- Define rutas con un enfoque basado en archivos y configura middleware para lógica específica.

¿Para qué se utiliza?

Next.js es ideal para:

- **Aplicaciones web dinámicas:** E-commerce, blogs, foros.
 - **Aplicaciones web de una sola página (SPA) o múltiples páginas (MPA).**
 - **Sitios web estáticos:** Portfolios, documentación, landing pages.
 - **Aplicaciones complejas:** Dashboards, aplicaciones empresariales.
-

Su enfoque en el desarrollo de aplicaciones rápidas, escalables y optimizadas convierte a Next.js en una herramienta poderosa para este proyecto. Consideramos que un gran volumen de personas ingresará al sitio web diseñado, y garantizar una interacción rápida y fluida será clave para el éxito de la Bienal

Strapi

Strapi es un potente CMS de código abierto basado en Node.js que permite gestionar contenidos y personalizar aplicaciones de manera sencilla. Su documentación se organiza en tres secciones principales: Guía de Usuario, Documentación para Desarrolladores y Strapi Cloud. Ofrece funcionalidades como creación y administración de tipos de contenido, manejo de usuarios y permisos, biblioteca multimedia y más, todo gestionable desde un panel de administración intuitivo.

Características Principales:

1. **Creación y Administración de Tipos de Contenido:** Permite el diseño de esquema de datos de manera visual y administrar su contenido fácilmente desde un panel de control.
2. **Manejo de Usuarios y Roles:** Incluye un sistema robusto de permisos y autenticación para asegurar el acceso controlado.
3. **Documentación Completa:** Cuenta con Guía de Usuario, Documentación dirigida a desarrolladores y Sobre Strapi Cloud.

¿Para que se usa?

En nuestro proyecto lo utilizamos para el lado del backend.

Render

Render es una plataforma de hosting en la nube, un PaaS (Platform as a Service), usada para desplegar y escalar aplicaciones web, estáticas y dinámicas, servicios backend, bases de datos, etc.

A comparación de otras plataformas como Heroku, AWS y otras, está pensada por el lado de la simplicidad y automatización.

Características Principales:

1. **Despliegue:** Cuenta con una integración directa a los servicios de versionado, como Github o Gitlab, lo que permite configurar despliegues automáticos ante cambios en los repositorios.
2. **Soporte a múltiples servicios:** Permite desplegar aplicaciones web, APIs, procesos Cron, procesos en segundo plano, base de datos, etc.
3. **Capacidades Adicionales:** Al realizar el despliegue, Render presta a las aplicaciones con un SSL gratuito y utiliza un CDN (Red de Entrega de Contenido) para asegurar la carga rápida y protección de la aplicación.

- 4. Variedad de opciones de precio:** Ofrece gran variedad de opciones de servicio, para usuarios avanzados a casuales; contando con opciones gratuitas para prácticas o demos de la aplicación.

¿Para qué se usa?

Decidimos usar Render como host del CMS Strapi, debido a su simplicidad en el proceso de despliegue, la existencia de documentación y tutoriales para guiarnos, y contar con una opción gratuita de uso.

Vercel

Vercel es una plataforma de despliegue enfocada en el Frontend, similar a Render, usando una conexión al repositorio deseado, Vercel carga y despliega la aplicación web en la nube y nos otorga de un dominio para acceder, actualizando la aplicación ante cualquier cambio en el código.

Vercel otorga a las aplicaciones de un CDN, SSL y despliegue automático de manera predeterminada, sin necesidad de configuraciones adicionales.

Es usado por muchos tipos de servicios web, para landing pages, PWA, sistemas de documentación, prototipos, proyectos Open Source, etc.

Decidimos usar Vercel como despliegue del frontend por estar trabajando con Next.js, ya que Vercel cuenta con optimizaciones para trabajar con estos frameworks, lo que mejora la velocidad y estabilidad del sitio.

NeonDB

NeonDB es una solución de bases de datos diseñadas específicamente para la nube. Está construido sobre PostgreSQL potenciando la flexibilidad, rendimiento y escalabilidad de un proyecto. Cuenta con una arquitectura serverless, es decir que elimina la necesidad de administrar servidores físicos.

Características Principales:

- 1. PostgreSQL:** Al basarse en PostgreSQL combina la robustez y familiaridad del mismo.
- 2. Serverless:** Al eliminar la necesidad de manejar los servidores físicos, la gestión de la infraestructura y la escalabilidad, permite que el almacenamiento y cómputo varía acorde a la demanda, pensando en las aplicaciones nativas de la nube.
- 3. Recuperación Automática:** Asegura la disponibilidad e integridad de los datos.

¿Para qué se usa?

Strapi por defecto trabaja con SQLite, un gestor de base de datos ligero y rápido, que, sin embargo, no es posible de usar en la nube. Se descubrió mediante recomendaciones en la web a NeonDB, que nos permitía resguardar la información de Strapi en PostgreSQL de una manera sencilla y segura.

Al realizar el despliegue en Render mediante su opción gratuita descubrimos que no contaba con almacenamiento persistente, esto significaba una pérdida total de la base de datos almacenada una vez el servidor quedará inactivo por un tiempo. NeonDB solucionaba esa desventaja, integrándose a Render de manera directa mediante las variables de ambiente.

Cloudinary

Cloudinary es una plataforma en la nube que se especializa en la gestión y entrega de contenido multimedia, como imágenes y videos. Esta herramienta utiliza un enfoque basado en APIs para facilitar la integración en aplicaciones web y móviles. Su arquitectura está diseñada para ofrecer escalabilidad, confiabilidad y un alto rendimiento en la manipulación de medios.

Características Principales:

- 1. Transformaciones:** Se pueden modificar atributos de los archivos multimedia en tiempo real.
- 2. Optimización:** Ajuste del formato y calidad en función del dispositivo y red del usuario.
- 3. Gestión:** Cuenta con un único repositorio todos los archivos.
- 4. Compatibilidad:** Cuenta con numerosas bibliotecas para distintos lenguajes de programación.

¿Para que se utiliza?

Cuando tuvimos más avanzado el proyecto, decidimos separar las imágenes de la página a un servicio más especializado en su manejo, a fin de mejorar los tiempos de respuesta de la página, al reducir el esfuerzo de carga. Strapi contaba con una integración mediante un complemento con Cloudinary por lo que su implementación fue simple y directa.

Expandiendo los Problemas del Desarrollo del Software

Durante el desarrollo del proyecto, se identificaron diversos desafíos que afectaron el progreso del equipo. A continuación, se detallan los principales problemas enfrentados:

1. Dificultades con Git

El manejo de Git fue un desafío constante para algunos miembros del equipo. Hubo problemas recurrentes al clonar repositorios, resolver conflictos de merge y al intentar solucionar errores eliminando partes del repositorio, lo que complicó el flujo de trabajo.

2. Errores de código en Next.js

Fueron problemas, típicos en frameworks que manejan renderizado del lado del cliente y del servidor, causaron inconsistencias en el rendimiento y en la visualización de las páginas.

3. Desafíos con el diseño y CSS

El diseño y la estilización presentaron dificultades, incluyendo problemas para replicar modelos de diseño en herramientas como Canva y falta de unificación en los estilos entre páginas. Además, hubo inconvenientes para localizar y organizar el código CSS, lo que aumentó la complejidad de mantener la coherencia visual del proyecto.

4. Inconvenientes con la API y Strapi

La configuración de Strapi presentó obstáculos importantes, incluyendo la necesidad de compartir configuraciones sin comprometer información sensible. También surgieron problemas con la carga de imágenes, datos incompletos y complicaciones durante el despliegue, reflejando una falta de experiencia en estas áreas específicas.

5. Errores de carga en las páginas

Se encontraron errores de carga debido a datos faltantes, enlaces comentados o problemas de conexión con la API, que generaron experiencias inconsistentes para los usuarios.

6. Dudas sobre el despliegue del proyecto

El equipo mostró incertidumbre respecto al despliegue del proyecto, especialmente sobre cómo configurar y publicar servicios como Strapi. Esto evidenció la necesidad de mayor conocimiento sobre el proceso de implementación y despliegue en aplicaciones web.

CONCLUSIÓN

En retrospectiva, este proyecto resultó ser una experiencia enriquecedora, tanto por su alcance más extenso como por la oportunidad de trabajar en un equipo más grande. La necesidad de investigar, seleccionar y complementar diversas tecnologías, así como de resolver los múltiples problemas que surgieron a lo largo del desarrollo, convirtió este trabajo en un valioso aprendizaje práctico.

A lo largo del proceso, enfrentamos desafíos no solo tecnológicos, sino también relacionados con el trabajo en equipo. Entre ellos, destacan dificultades en la comunicación, la falta de conocimiento sobre algunas herramientas y problemas en la organización del trabajo. Sin embargo, se logró superar estos obstáculos, sumando experiencia a cada integrante del grupo.

BIBLIOGRAFÍA

Node.js. (n.d.). *About Node.js*. Node.js. Recuperado el 26 de Noviembre de 2024, de <https://nodejs.org/en/about/>

React. (n.d.). *React documentation*. React. Recuperado el 26 de Noviembre de 2024, de <https://react.dev>

Next.js. (n.d.). *Next.js documentation*. Vercel. Recuperado el 26 de Noviembre de 2024, de <https://nextjs.org/docs>

Strapi. (n.d.). *Strapi Cloud documentation*. Strapi. Recuperado el 28 de Noviembre de 2024, de <https://docs.strapi.io/cloud/intro>

Render. (n.d.). *About Render*. Render. Recuperado el 28 de Noviembre de 2024, de <https://render.com/about>

Website Planet. (n.d.). *Render Web Hosting Review: 2023 Pros & Cons*. Recuperado el 28 de Noviembre de 2024, de <https://www.websiteplanet.com/web-hosting/render/>

Cloudinary. (n.d.). *About Cloudinary*. Cloudinary. Recuperado el 30 de Noviembre de 2024, de <https://cloudinary.com/about>

Cloudinary. (n.d.). *Frequently Asked Questions*. Cloudinary. Recuperado el 30 de Noviembre de 2024, de <https://cloudinary.com/faq>

Sukalihar, N. (n.d.). *What is NeonDB? A modern, cloud-native PostgreSQL database*. LinkedIn. Recuperado el 1 de Diciembre, de <https://www.linkedin.com/pulse/what-neondb-modern-cloud-native-postgresql-database-n-ilesh-sukalihar-toijf>

Neon. (n.d.). *Why Neon?*. Neon Documentation. Recuperado el 1 de Diciembre, de <https://neon.tech/docs/get-started-with-neon/why-neon>

Vercel. (2021, March 8). *What is Vercel?*. Vercel Blog. Recuperado el 2 de Diciembre, de <https://vercel.com/blog/what-is-vercel>

Gluo. (n.d.). *Vercel: ¿Qué es y para qué sirve?*. Gluo Blog. Recuperado el 2 de Diciembre, de <https://sitio-gluo-2022.vercel.app/blog/vercel-que-es-y-para-que-sirve>