

República Bolivariana de Venezuela
Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación



**Desarrollo de un editor de visualizaciones de
propiedades de historiales de wikis**
Universidad Central de Venezuela

Tutor Prof. Eugenio Scalise
Adrian J. Mejias O. y Jose E. Tirado S.
enero, 2022

Índice general

1. Introduccion	4
2. Marco Teorico	5
2.1. Dominio del problema	5
2.1.1. Wiki, Mediawiki y Wikimedia	5
2.1.2. Filosofía de wiki	6
2.1.3. Moderación de contenido en Wikipedia	6
2.1.4. Watcher	7
2.1.5. Wikipedia como ejemplo práctico	7
2.2. Tecnologías a utilizar	7
2.2.1. ReactJS	7
2.2.2. Next.js	9
2.2.3. MongoDB	9
2.2.4. Fastify	10
2.3. Metodología de desarrollo	11
2.3.1. Desarrollo Rápido de Aplicaciones (RAD)	11
3. Marco Tecnológico	13
3.1. Marco Tecnológico	13
3.1.1. Librerías o frameworks para aplicaciones intensivas de frontend	13
3.2. Arquitectura	13
3.3. Librerías para la visualización de datos	14
3.3.1. Data Driven Documents (D3)	14
3.3.2. Recharts	14
3.3.3.	14

4. Propuesta de Trabajo Especial de Grado	15
4.1. Contexto	15
4.2. Objetivo general	16
4.3. Objetivos específicos	16
4.4. Justificación	16
4.5. Distribución del documento	16

índice de figuras

2.1. Número de peticiones por segundo para distinta cantidad de conexiones	10
2.2. Fases del Desarrollo Rápido de Aplicaciones (enfoque de James Martin)	11

Capítulo 1

Introduccion

Introduccion

Capítulo 2

Marco Teorico

2.1. Dominio del problema

2.1.1. Wiki, Mediawiki y Wikimedia

El término wiki proviene de la raíz Hawaiana “wiki”, que significa “rápido”, y fue propuesto por Ward Cunningham, quien a su vez define los sitios web wiki como “La base de datos más simple que puede existir” [1]. Con el tiempo este concepto fue evolucionando, y en la actualidad cuando hablamos de wiki nos referimos a un sitio web que permite a sus usuarios colaborar en su estructura y contenido. Estos sitios webs son impulsados por el motor wiki, también llamado MediaWiki, el cual es un Sistema Manejador de Contenido (CMS) que permite a los usuarios colaborar en el sitio web sin la necesidad de tener permisos de dueño o líder.

La enciclopedia Wikipedia es el sitio web más popular basado en wiki, que a su vez forma parte del movimiento Wikimedia, el cual incluye otros proyectos interrelacionados, tales como: Wiktionary, Wikiquote, Wikibooks, Wikisource, entre otros, cuyo propósito es usar el poder colaborativo de internet, y el concepto wiki, para compartir conocimiento gratuito de cualquier tipo.

2.1.2. Filosofía de wiki

En la actualidad, gracias a la evolución del internet, la información es considerada virtualmente ubicua y en constante cambio, y lo que realmente ofrece valor es la capacidad individual de sintetizar esa información y relacionarla. Como resultado de esto surge la filosofía wiki, en donde la información se comparte, y el conocimiento no se crea, sino se co-crea de forma colaborativa.

El concepto de la filosofía de wiki y el software utilizado para crear estos sitios web están intrínsecamente relacionados, y no se podría poner en práctica lo primero sin lo segundo. Esto es así debido a que el software debe proporcionar el medio para que pueda existir esa construcción colectiva de conocimiento, que es indispensable en la filosofía wiki.

Algunas de las características de software de los sitios web que hacen uso de esta filosofía son:

1. Cualquiera puede cambiar cualquier cosa
2. Usan un sistema de marcas hipertextuales simplificadas, lo que resulta imprescindible para hacer posible la colaboración.
3. No posee una estructura predefinida a la que se tengan que acomodar los usuarios, lo que otorga flexibilidad.

2.1.3. Moderación de contenido en Wikipedia

El principal problema que maneja Wikipedia en cuanto a moderación de contenido viene como resultado de su propia filosofía "todos pueden editar", lo que conlleva a multiples problemas tales como: vandalismo, escritura pobre, una mala estructura de página, peleas de edición, entre otras cosas. Por esta razón no existe una solución única para acabar con la existencia de "mal" contenido en Wikipedia, y es indispensable el uso de participación humana en procesos de moderación que implican complejos desafíos técnicos y éticos.

Una de las formas que tiene Wikipedia de detectar vandalismo es usando las estadísticas de los artículos para verificar si hay una gran cantidad de ediciones de un articulo en un periodo muy corto de tiempo, o si estas revisiones vienen de la misma dirección IP, bloqueando o baneando las direcciones IP como método de reducción de vandalismo. Sin embargo, el bloqueo de IPs es en sí mismo un método que resulta contradictorio para el núcleo principal

de la filosofía de wiki: "todos pueden editar", por lo que el bloqueo de IPs es utilizado siempre como última alternativa contra el vandalismo.

2.1.4. Watcher

El concepto de watcher en Wikipedia no difiere mucho de las demás paginas que aplican el mismo sistema, siendo este una forma que permite al usuario obtener información de presencia de aquellas paginas que se encuentran en su watchlist.

En el ámbito de Wikipedia la información de presencia de una pagina se refiere a los cambios recientes hechos a dicha pagina, desplegados de forma simplificada en conjunto con todas las demás páginas de interés para el usuario.

2.1.5. Wikipedia como ejemplo práctico

Wikipedia es un sitio web que permite a sus usuarios obtener conocimiento de una gran variedad de temas, siendo la enciclopedia online mas grande que existe; esto se debe a su naturaleza colaborativa que permite a cualquier persona aportar conocimiento escribiendo o editando artículos. Al mismo tiempo, los grupos encargados de realizar la moderación de contenido no son suficientes para asegurar que todo el contenido que existe en el sitio web sea verídico, haciendo de Wikipedia un sitio web con contenido no confiable por las entidades académicas.

2.2. Tecnologías a utilizar

Para la realización de este proyecto se usarán las siguientes tecnologías:

2.2.1. ReactJS

ReactJS es una librería de JavaScript de código abierto desarrollada por Facebook para facilitar la creación de componentes interactivos, reutilizables, para desarrollos de interfaces de usuario, especialmente aplicaciones de una sola página.

React maneja el concepto de “programación reactiva” haciendo uso de un DOM Virtual, lo le permite determinar qué partes del DOM han cambiado comparando contenidos entre la versión nueva y la almacenada den el DOM virtual, para así propagar los datos generando cambios en la aplicación, es decir, los datos “reaccionan” ejecutando una serie de eventos.

Este concepto de reactividad es lo que hace a la librería altamente eficiente, ya que limita la actualización del DOM solamente a los elementos que han cambiado.

Otras características que destacan en React son:

■ Componentes

El código de React es hecho con entidades llamadas componentes. Los componentes pueden ser renderizados en elementos particulares del DOM usando la librería de React DOM. Estos componentes son capaces de recibir parámetros conocidos como ”propiedades del componente” de la siguiente forma:

```
ReactDOM.render(<Greeter greeting="Hello World!" />, document.  
getElementById('myReactApp'));
```

Las 2 formas de declarar componentes en react es mediante el uso de funciones o clases, y generalmente se usa una de las dos opciones de forma situacional.

■ JSX

JSX, tambien llamado Javascript XML, es una extensión a la sintaxis del lenguaje javascript. Este provee una forma de estructurar componentes usando una sintexis familiar para muchos desarrolladores. Los componentes de React son usualmente escritos usando JSX, aunque tambien pueden ser escritos usando Javascript puro.

Un ejemplo de código JSX:

```
class App extends React.Component {  
  render() {  
    return (  
      <div>  
        <p>Header</p>  
        <p>Content</p>  
        <p>Footer</p>  
      </div>  
    );  
  }  
}
```

```
    }  
  }  
</div>  
);
```

■ Hooks

Los hooks son funciones que permiten a los desarrolladores “engancharse” a los estados de React y a ciertos puntos dentro del ciclo de vida de los componentes.

React proporciona algunos hooks integrados tales como: `useState`, `useContext`, `useReducer`, `useMemo` y `useEffect`, los cuales son los mas usados y permiten controlar los estados y eventos respectivamente.

2.2.2. Next.js

Next.js es un framework desarrollado encima de Node.js que permite a las aplicaciones de React usar funcionalidades como el renderizado del lado servidor y la generación de paginas web estáticas.

Por defecto, Next.js pre-renderiza cada pagina. Esto significa que Next.js genera HTML para cada pagina en adelante, en vez de hacerse con Javascript del lado del cliente. Pre-renderizado puede resultar en mejor rendimiento y SEO

Cada HTML generado es asociado con el mínimo código Javascript necesario para que funcione la pagina. Cuando una página es cargada en el explorador, su código javascript se ejecuta y hace la página totalmente interactiva. A este proceso de le conoce como “hydration”

Next.js ofrece 2 formas de pre-renderizado:

- Generación estática: El HTML es generado a tiempo de ejecución y será reutilizado en cada petición.
- Renderizado lado servidor: El HTML es generado en cada petición

2.2.3. MongoDB

MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se

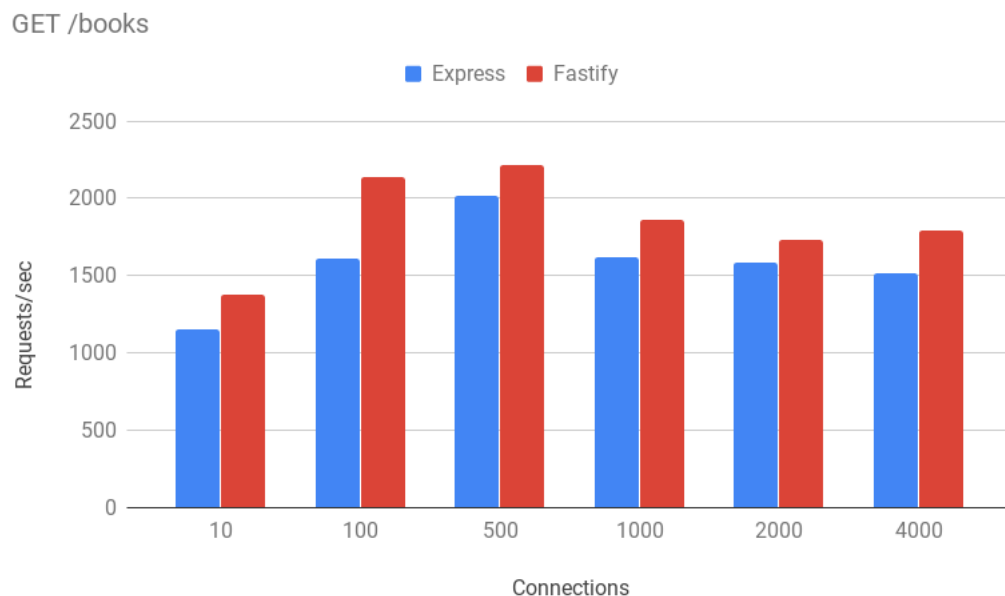


Figura 2.1: *Número de peticiones por segundo para distinta cantidad de conexiones*

hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

2.2.4. Fastify

Fastify es un framework web para Node.js de código abierto concentrado en proporcionar el mejor rendimiento, y una arquitectura flexible.

Si comparamos la velocidad de fastify con otros frameworks web como express, tal como podemos ver en la figura Figura 2.1, notamos que fastify es aproximadamente un 20% más rápido que express.

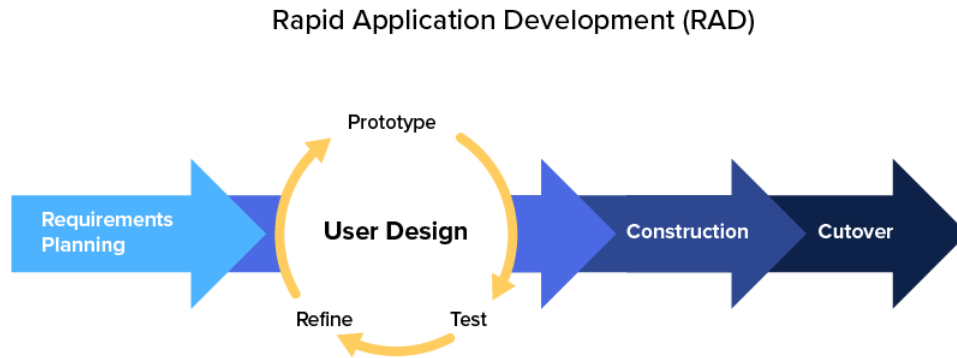


Figura 2.2: *Fases del Desarrollo Rápido de Aplicaciones (enfoque de James Martin)*

2.3. Metodología de desarrollo

2.3.1. Desarrollo Rápido de Aplicaciones (RAD)

El desarrollo rápido de aplicaciones es una metodología de desarrollo que prioriza la creación rápida y prototipos y la retroalimentación rápida sobre ciclos prolongados de desarrollo y prueba. Con el uso de RAD, los desarrolladores pueden realizar múltiples iteraciones y actualizaciones al software rápidamente sin la necesidad de iniciar un cronograma de desarrollo desde cero cada vez.

Esta metodología de desarrollo fue creada por James Martin en el año 1980 en IBM y fue formalizada con la publicación del libro *Rapid Application Development* en 1991.

Fases del Desarrollo Rápido de Aplicaciones

Según el enfoque de James Martin el Desarrollo Rápido de Aplicaciones se divide en las siguientes fases [2]:

- **Fase de planificación:** Esta fase es equivalente a una reunión de alcance del proyecto. Aunque la fase de planificación se condensa en

comparación con otras metodologías de gestión de proyectos.

Durante esta etapa, los desarrolladores, los clientes (usuarios de software) y los miembros del equipo se comunican para determinar los objetivos y las expectativas del proyecto, así como los problemas actuales y potenciales que deben abordarse durante la construcción.

- **Fase de diseño:** Durante esta fase, los usuarios interactúan con analistas de sistemas y desarrollan modelos y prototipos que representan todos los procesos, entradas y salidas del sistema.

Todos los errores y problemas se resuelven en un proceso iterativo. El desarrollador diseña un prototipo, el cliente (usuario) lo prueba y luego se reúnen para comunicar qué funcionó y qué no.

- **Fase de construcción:** Esta etapa toma los prototipos y el sistema en su fase beta resultado de la etapa de diseño y lo convierte en un modelo funcional.

Debido a que la mayoría de los problemas y cambios se abordaron durante la minuciosa fase de diseño iterativo, los desarrolladores pueden construir el modelo de trabajo final más rápidamente que siguiendo un enfoque tradicional de gestión de proyectos.

- **Fase de transición:** Esta es la fase de implementación donde el producto terminado va al lanzamiento. Incluye conversión de datos, pruebas y cambio al nuevo sistema, así como capacitación de usuarios.

Todos los cambios finales se realizan mientras los programadores y los clientes continúan buscando errores en el sistema.

Capítulo 3

Marco Tecnológico

3.1. Marco Tecnológico

3.1.1. Librerías o frameworks para aplicaciones intensivas de frontend

3.2. Arquitectura

Para que una aplicación sea descubierta y usada por internautas es fundamental que tenga una buena relación con los motores de búsqueda.

Sin embargo también para asegurar la larga vida y mantenibilidad de la aplicación y la facilidad de desarrollo se debe tomar en cuenta herramientas extensamente empleadas contemporáneamente como Angular, React y Vue.

El problema entonces recae en que estas tecnologías son meramente para SPA. Lo que implica entonces que no existe una noción real de SEO - En las SPA el enrutamiento ocurre del lado del cliente usando JavaScript, y en consecuencia los crawlers de los motores de búsqueda no saben interpretar estas páginas.

Como remedio surge un nuevo paradigma, que es el que vamos a usar para esta aplicación, conocido como Server Side Rendering; donde se utiliza estas tecnologías SPA como un motor de plantillas para retornar un HTML que los motores de búsqueda puedan entender, y después por un proceso conocido

como hydration, las aplicaciones en el lado del cliente dejan de comportarse como HTML plano y retoman sus funcionalidades de SPA.

Asi entonces llegamos al perfecto balance en el que tenemos herramientas actuales y faciles de usar, que tambien cumplen con los requerimientos de los motores de busqueda para indexar nuestras paginas.

3.3. Librerias para la visualización de datos

3.3.1. Data Driven Documents (D3)

3.3.2. Recharts

3.3.3.

Capítulo 4

Propuesta de Trabajo Especial de Grado

4.1. Contexto

Wikipedia contiene en si una masiva cantidad de datos "base como articulos, eventos, noticias, media y demas. Pero tambien ella misma genera nuevos datos con cada nueva adicion y edicion a su haber documental. Todos estos "astros" que dejan miles de colaboradores dia a dia son conocidos como metadata.

Esta desde luego tiene un inmenso valor por si misma, y es refinada principalmente por analistas de datos y aficionados que quieren buscar patrones, relaciones o informacion que no es facil o posible de distinguir con ojos humanos.

Y son estos mismos grupos que han desarrollado soluciones para extender y analizar las wikis.

Herramientas de extension

- API Wikimetrics
-

Nuestra labor entonces es facilitar este trabajo para ellos dejandoles crear sus propias graficas sobre los articulos que deseen

4.2. Objetivo general

Crear una nueva version del front-end de wikimetrics

4.3. Objetivos específicos

- Implementar una aplicación web responsive que ofrezca las funcionalidades requeridas por un watcher de un wiki y que pueda ser reconocida por los motores de búsqueda.
- Consumir y extender la API de wikimetrics para desarrollar una aplicación web que habilite a sus usuarios construir y visualizar graficas
- Definir los requerimientos de la aplicación
- Utilizar un metodo ´agil para el desarrollo de la aplicación.
- Realizar el despliegue y puesta en producción de la aplicación

4.4. Justificación

4.5. Distribución del documento

Bibliografía

- [1] Ward Cunningham. *What is Wiki*. Media Wiki. 2002. URL: <https://web.archive.org/web/20080416212802/http://www.wiki.org/wiki.cgi?WhatIsWiki>.
- [2] James Martin. *Rapid Application Development*. IBM. 1991. URL: <https://archive.org/details/rapidapplication00mart>.