

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.
marzo, 2022

Índice general

1. Introducción	3
2. Tecnologías para la visualización de datos en web	5
2.1. Marco Tecnológico	5
2.1.1. Librerías o frameworks para aplicaciones intensivas de frontend	5
2.2. Arquitectura	5
2.3. Librerías para la visualización de datos	6
2.3.1. Data Driven Documents (D3)	6
2.3.2. Recharts	6
2.3.3.	6
3. Tecnologías para el desarrollo web	7
3.1. Dominio del problema	7
3.1.1. ReactJS	7
3.1.2. Next.js	9
3.1.3. MongoDB	9
3.1.4. Fastify	10
3.2. Metodologías ágiles	11
3.3. Frameworks	11
4. Propuesta de Trabajo Especial de Grado	13
4.1. Contexto	13
4.2. Objetivo general	14
4.3. Objetivos específicos	14
4.4. Justificación	14
4.5. Distribución del documento	14

índice de figuras

3.1. Número de peticiones por segundo para distinta cantidad de conexiones	10
---	----

Capítulo 1

Introducción

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.

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:

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 múltiples 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 artículo 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.

Capítulo 2

Tecnologías para la visualización de datos en web

2.1. Marco Tecnológico

2.1.1. Librerías o frameworks para aplicaciones intensivas de frontend

2.2. 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.

Así entonces llegamos al perfecto balance en el que tenemos herramientas actuales y fáciles de usar, que también cumplen con los requerimientos de los motores de búsqueda para indexar nuestras páginas.

2.3. Librerías para la visualización de datos

Para la selección de librería se consideran los siguientes factores

1. Debe ser un proyecto open source. 2. De ser posible, debe tener bindings para react para facilidad en el desarrollo. 3. Debe ser extensible para poder implementar aquellas visualizaciones que sean muy específicas. 4. Deben ser longevas y tener cierta garantía de que será mantenida en el tiempo, así se asegura que

2.3.1. Data Driven Documents (D3)

2.3.2. Recharts

2.3.3.

Capítulo 3

Tecnologías para el desarrollo web

3.1. Dominio del problema

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

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

Un ejemplo de codigo JSX:

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

3.1.2. 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 se 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

3.1.3. 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 hace en las bases de datos relacionales, MongoDB guarda estructuras de

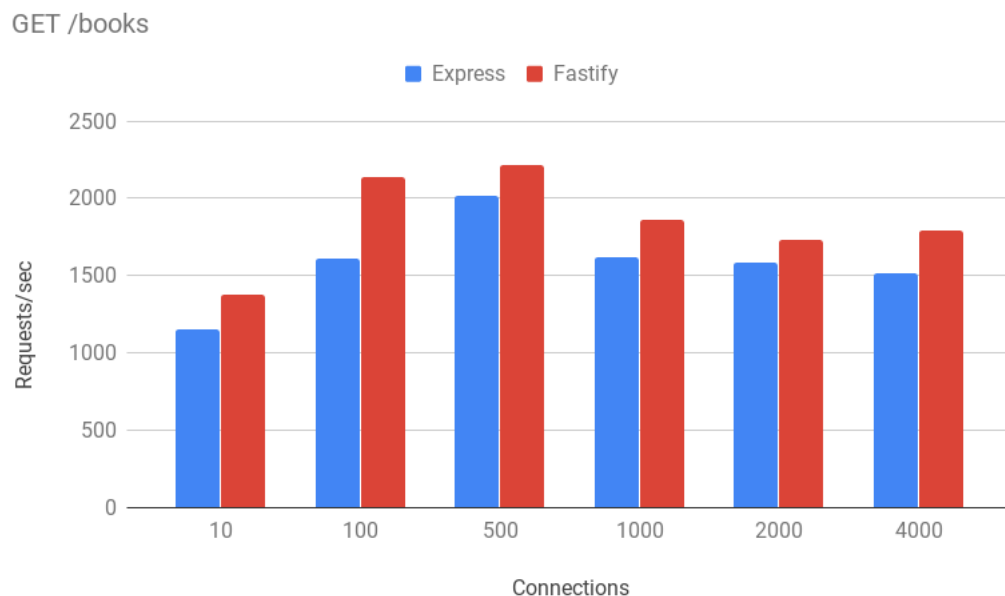


Figura 3.1: *Número de peticiones por segundo para distinta cantidad de conexiones*

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.

3.1.4. 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 3.1, notamos que fastify es aproximadamente un 20 % más rápido que express.

3.2. Metodologías ágiles

3.3. Frameworks

1. Kanban: Tiene como objetivo la mejora continua, la flexibilidad en la gestión de tareas y un flujo de trabajo mejorado. Con este enfoque ilustrativo, el progreso de todo el proyecto se puede comprender fácilmente de un vistazo. Para esto hace uso del tablero Kanban, que es una herramienta que visualiza todo el proyecto para rastrear el flujo de su proyecto. A través de este enfoque gráfico de los tableros Kanban, un miembro nuevo o una entidad externa puede comprender lo que está sucediendo en este momento, las tareas completadas y las tareas futuras.
2. Scrum: Es un framework para desarrollo, entrega, y mantenimiento de proyectos en un ambiente complejo, con un énfasis inicial en el desarrollo de software, aunque también ha sido utilizado en otras áreas como la investigación, ventas, mercadeo y tecnologías avanzadas. Está diseñado para equipos de 10 personas o menos, quienes rompen su trabajo en metas que pueden ser completadas en iteraciones de tiempo fijo, llamadas *sprints*, con duraciones aproximadas de 2 semanas.
3. Lean software development: Es un framework popular basado en optimizar tiempo de desarrollo y recursos, eliminando desperdicios y entregando solamente lo que el producto necesita. El método Lean es usualmente referido como la estrategia del “Producto Mínimo Viable (PMV)”, estrategia, en la que un equipo lanza una versión mínima de su producto al mercado, aprende de los usuarios lo que les gusta, lo que no les gusta y lo que quieren que se agregue, y luego itera en función de estos comentarios.
4. Extreme programming (XP): Es una metodología de desarrollo de software cuyo objetivo es mejorar la calidad del software y la adaptabilidad al cambio de los requerimientos del cliente. Al ser un tipo de metodología ágil, se basa en el uso de ciclos de desarrollo cortos con lanzamientos frecuentes, con el propósito de mejorar la productividad e introducir “checkpoints” en los que se puedan adoptar nuevos requisitos de clientes.
5. Adaptive Software Development (ASD): Es una consecuencia directa

del desarrollo ágil. Su objetivo es permitir que los equipos se adapten rápida y eficazmente a los requisitos cambiantes o las necesidades del mercado mediante la evolución de sus productos con una planificación ligera y un aprendizaje continuo. El enfoque ASD alienta a los equipos a desarrollarse de acuerdo con un proceso de tres fases: especular, colaborar, aprender.

6. Rapid application development (RAD): es una forma de metodología de desarrollo de software ágil que prioriza las versiones e iteraciones rápidas de prototipos. A diferencia del método Waterfall, RAD enfatiza el uso de software y los comentarios de los usuarios sobre la planificación estricta y el registro de requisitos.

Capítulo 4

Propuesta de Trabajo Especial de Grado

4.1. Contexto

Con aproximadamente seis millones cuatrocientos mil artículos (6.400.000) la Wikipedia lidera como la enciclopedia mas extensa del mundo. No solo eso, sino que muy comunmente al googlear sobre algun tema de interes siempre es el primer resultado. Esto es gracias al esfuerzo colaborativo de miles de

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 gráficas
- Definir los requerimientos de la aplicación
- Utilizar un método ágil para el desarrollo de la aplicación.
- Realizar el despliegue y puesta en producción de la aplicación

4.4. Justificación

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 solamente viendo numeros y fechas.

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>.