

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.  
mayo, 2022

# Índice general

<b>1. Introducción</b>	<b>4</b>
<b>2. Tecnologías para la visualización de datos en web</b>	<b>5</b>
2.1. Librerías o frameworks para aplicaciones intensivas de frontend	5
2.2. Librerías para la visualización de datos . . . . .	5
2.2.1. Data Driven Documents (D3) . . . . .	5
2.2.2. Recharts . . . . .	5
2.2.3. . . . .	5
<b>3. Tecnologías para el desarrollo web</b>	<b>6</b>
3.1. tecnologías para el desarrollo . . . . .	6
3.2. Arquitectura . . . . .	6
3.3. Backend . . . . .	7
3.3.1. MongoDB . . . . .	7
3.3.2. Fastify . . . . .	7
3.4. Frontend . . . . .	8
3.4.1. ReactJS . . . . .	8
3.4.2. Next.js . . . . .	10
3.5. Metodologías ágiles . . . . .	11
3.6. Frameworks . . . . .	11
<b>4. Propuesta de Trabajo Especial de Grado</b>	<b>13</b>
4.1. Propuesta de Trabajo Especial de Grado . . . . .	13
4.1.1. Motivación e identificación del problema . . . . .	13
4.1.2. Objetivos del trabajo . . . . .	13
4.1.3. Estrategia de solución y método de desarrollo ágil a utilizar . . . . .	13

4.1.4.	Trabajos similares, diferencias y ventajas de la solución a desarrollar . . . . .	13
4.2.	Contexto . . . . .	13
4.3.	Objetivo general . . . . .	14
4.4.	Objetivos específicos . . . . .	14
4.5.	Justificación . . . . .	14
4.5.1.	Antecedentes . . . . .	15
4.5.2.	Herramientas . . . . .	16
4.5.3.	Prototipos . . . . .	16
4.6.	Distribución del documento . . . . .	19

# índice de figuras

3.1. Número de peticiones por segundo para distinta cantidad de conexiones . . . . .	8
---	---

# Capítulo 1

## Introducción

Un wiki es un sitio web que permite a sus usuarios colaborar en su estructura y contenido. Esta versatilidad que provee el concepto de Wiki es lo que lo convierte en una de las herramientas mas usadas en la actualidad para compartir información. La enciclopedia Wikipedia es el sitio web más popular basado en wiki.

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.

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.

## Capítulo 2

# Tecnologías para la visualización de datos en web

### 2.1. Librerías o frameworks para aplicaciones intensivas de frontend

### 2.2. Librerías para la visualización de datos

Para la selección de librería se consideran los siguiente 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 sera mantenida en el tiempo, asegurando asi

#### 2.2.1. Data Driven Documents (D3)

#### 2.2.2. Recharts

#### 2.2.3.

## Capítulo 3

# Tecnologías para el desarrollo web

### 3.1. tecnologías para el desarrollo

Actualmente existe un catalogo muy amplio de tecnologías de desarrollo web, por lo que escoger las debidas herramientas y sobre todo analizar las compatibilidades entre estas herramientas pueden ser el punto clave para obtener la mejor eficiencia y experiencia de usuario.

Para nuestro proyecto haremos uso de un servidor o backend que se va a encargar de obtener la data y transformarla de forma que el frontend la pueda usar directamente sin usar computo excesivo del lado del cliente, esto con el propósito de darle al usuario una mejor experiencia.

En este capitulo presentaremos las herramientas que nos permitirán realizar este proyecto tanto para backend como para 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 paginas.

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 fáciles de usar, que también cumplen con los requerimientos de los motores de búsqueda para indexar nuestras paginas.

## **3.3. Backend**

### **3.3.1. MongoDB**

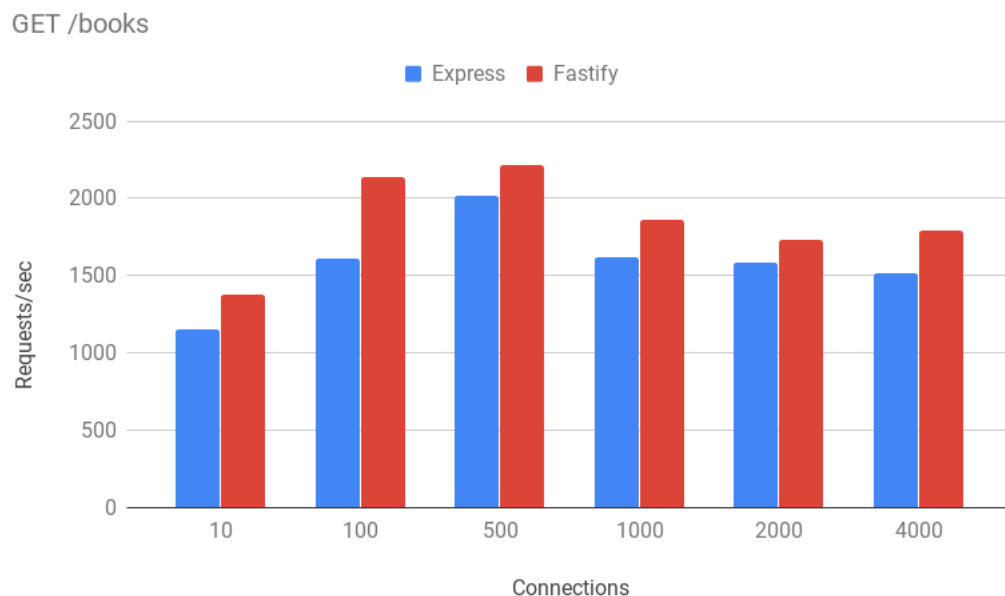
MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

### **3.3.2. 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





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

un 20 % más rápido que express.

## 3.4. Frontend

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

### 3.4.1. 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, también 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 también 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>
      );
    }
  }
```

### ■ 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.

### 3.4.2. 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

## 3.5. Metodologías ágiles

## 3.6. 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. Propuesta de Trabajo Especial de Grado

#### 4.1.1. Motivación e identificación del problema

#### 4.1.2. Objetivos del trabajo

Objetivo General

Objetivos Específicos

#### 4.1.3. Estrategia de solución y método de desarrollo ágil a utilizar

#### 4.1.4. Trabajos similares, diferencias y ventajas de la solución a desarrollar

### 4.2. 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 comúnmente al googlear sobre algún tema de interés 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 dejándoles crear sus propias gráficas sobre los artículos que deseen

### **4.3. Objetivo general**

Crear una nueva version del front-end de wikimetrics

### **4.4. 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

### **4.5. Justificación**

Wikipedia contiene en si una masiva cantidad de datos "base como artículos, eventos, noticias, media y demás. Pero también ella misma genera nuevos datos con cada nueva adición y edición 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 información que no es fácil o posible de distinguir solamente viendo números y fechas.

#### 4.5.1. Antecedentes

Siendo wikimedia y wikipedia las mas grandes comunidades y organizaciones dedicadas a recopilar datos, es lógico pensar que tiene consigo una abundante cantidad de seguidores capacitados y apasionados por aportar lo que puedan. A continuación se detallara el estado actual de las herramientas existentes para explorar la wikipedia.

Estas herramientas cumplen diferentes objetivos,

1. Identificar posibles candidatos para ser administrador
2. Identifica
3. Etc.

<https://wikitech.wikimedia.org/wiki/Portal:Toolforge> [https://www.wikidata.org/wiki/Wikidata:Tools/Visualize\\_data](https://www.wikidata.org/wiki/Wikidata:Tools/Visualize_data) <https://en.wikipedia.org/wiki/Wikipedia:Tools>

[apersonbot.toolforge.org](https://apersonbot.toolforge.org)

Este set de herramientas esta enfocado en supervisar usuarios y sus contribuciones a la wikipedia

- Candidate Search - Aadminsore - Articles for Creation Review History

<https://xtools.wmflabs.org/>

Sin embargo existe una oportunidad para aprovechar mejor la metadata de wikipedia, y explorar y explotar esta sera el problema a resolver de este trabajo especial de grado <https://sigma.toolforge.org/>

<https://interaction-timeline.toolforge.org/>

<https://cosmiclattes.github.io/wikireplay/player>

[https://de.wikipedia.org/wiki/Benutzer:Atlasowa/edit\\_history\\_visualization](https://de.wikipedia.org/wiki/Benutzer:Atlasowa/edit_history_visualization)



Actividad	Tiempo Estimado
Preparar entorno de desarrollo	
Estudiar API existente de Wikimetrix	

### 4.5.2. Herramientas

#### Tecnologías para el ambiente de desarrollo

- Visual Studio Code.
- Git, como control de versiones.
- Navegadores web (Google Chrome, Firefox, etc)
- MongoDB
- 

#### Tecnologías usadas para desarrollar

- Visual Studio Code.
- Git, como control de versiones.
- Navegadores web (Google Chrome)
- MongoDB
- 

### 4.5.3. Prototipos

/watcher Muestra el perfil de un watcher, en el puedes ver todas las contribuciones de este junto con la lista de visualizaciones que ha creado. Se permitirá un mínimo de customización. Y se podrá enlazar el perfil de wikipedia

/home Para usuarios autenticados, les permite ver diferentes secciones relevantes para ellos

1. Watched Muestra una lista y un pequeño abstracto de los artículos que el usuario hace watching y tienen una visualización actualizada

## WikiMetaView

Articulo

Articulo

Articulo

Home

Favoritos

Watched

Recents

## Watch Lists

Watch List name

Watch List name

Watch List name

Watch List name


Watch List name

Watch List name

Watch List name

Settings

Profile

Request an Article 

## Favoritos

Article Title<sup>2</sup>

dated: 28/03/2022

Article Title



dated: 28/03/2022

Unwatch  
Favorite  
Add to list

Article Title



dated: 28/03/2022

Article Title



dated: 28/03/2022

## Recently viewed

Article Title<sup>2</sup>

dated: 28/03/2022

Article Title



dated: 28/03/2022

Article Title



dated: 28/03/2022

Article Title



dated: 28/03/2022

2. Queue Muestra una lista de los artículos que le usuario solicito para hacer una visualización pero que el server no ha podido solicitar

3. Explore 3.1. Favoritos Muestra una lista de aquellas visualizaciones que tienen mucha actividad

3.2 Watched

3.3. Controversial Muestra una lista de aquellas visualizaciones que provocan discusion en la comunidad, caracterizado por muchos comentarios

3.4 Recientemente vistos

3.5 Hot now

/index Esta pagina se encarga de explicar las motivaciones de la App y dar una noción básica del funcionamiento para los watchers. Tiene el objetivo de cap

/settings

/article?title=<title>&url=<url> Muestra un articulo junto con su metadata y las visualizaciones creadas por los usuarios.d

/about Breve resumen del proyecto, incluye el documento de tesis y documento de seminario asi como datos de contacto y repositorios de github para futuros contribuyentes.

Modales

Query Params Globales

?auth

Si este parámetro esta en el url. Presenta la modal de autenticacion

?

La solucion plantada es un avance de la tesis por Leonardo Testa hecha en 20XX.

Actores

Watchers Anonimos

## 4.6. Distribución del documento