

Plataforma Web de recolección de datos del grupo de investigación de ciencias
básicas del sistema músculo-esquelético de la Universidad El Bosque

Autor:

Dr. Rubén Darío Hernández González

Residente de ortopedia y Traumatología Universidad El Bosque, grupo de investigación de
sistema músculo-esquelético Fundación CIMB – Universidad El Bosque

Universidad El Bosque
Facultad de Medicina
Especialización en Ortopedia y Traumatología

Universidad El Bosque

Facultad de Medicina

Título de trabajo de investigación

Plataforma Web de recolección de datos del grupo de investigación de ciencias básicas del sistema músculo-esquelético de la Universidad El Bosque

Tipo de investigación:

Postgrado

Investigadores:

Dr. Rubén Darío Hernández González

Residente de ortopedia y Traumatología Universidad El Bosque, grupo de investigación de sistema músculo-esquelético Fundación CIMB – Universidad El Bosque

Dr. Juan Manuel Herrera Arbeláez

Ortopedista y traumatólogo Universidad El Bosque

Director de grupo de investigación de sistema músculo-esquelético Fundación CIMB - CLEMI

PLATAFORMA WEB DE RECOLECCIÓN DE DATOS DEL GRUPO DE INVESTIGACIÓN DE CIENCIAS BÁSICAS DEL SISTEMA MÚSCULO-ESQUELÉTICO DE LA UNIVERSIDAD EL BOSQUE	1
Introducción	5
Descripción general	7
Oportunidad, y problemática	7
Formulación del problema	7
Justificación del problema	8
Impacto esperado	9
Descripción del proyecto	9
Objetivo general	9
Objetivos específicos	10
Metodología	10
Fase de investigación.	11
Marco teórico	11
Lenguaje de programación C	11
Variables	11
Variables estadísticas	12
Estructuras	13
Unix	13
GNU - Linux	14
Copyleft	14
PHP	14
HTML	14
JavaFX	15
SQL (Structured Query Language, Lenguaje estructurado de consultas)	15
Base de datos:	15
Bases de datos relacionales:	16
Bases de datos no relacionales	16
Marco conceptual	17
Casos de uso	19
Caso de uso: Menú investigación	19
Caso de uso: Crear de investigación	22
Caso de uso: Calidad	26
Arquitectura	29
Vista física	29
Vista de componentes	29
Componentes del servidor	31
Componentes dispositivo móvil	31
Vista de diagramas de clase	31
Procesos del sistema	32
Herramientas y frameworks utilizados	33
Frontend / Reactjs:	33
¿Cómo funciona React?	34
¿Cómo funciona el DOM virtual?	34

Backend / Nodejs:	35
Javascript en el lado del servidor	35
Centrado en la escalabilidad	36
NPM node package manager	38
Un estándar del desarrollo web	38
Conclusión	39
Almacenamiento / MongoDB:	39
¿Cómo funciona MongoDB?	41
¿Dónde se puede utilizar MongoDB?	42
¿Dónde no se debe usar MongoDB?	42
¿Cómo se instala MongoDB?	43
¿Cómo puedo usar la consola?	44
Criterios de elección	44
Análisis de los resultados	44
Resumen	45
Bibliografía	46
Anexos:	47

Introducción

Hoy vivimos en el entorno de las redes sociales, plataformas tecnológicas y nos comunicamos, compramos, nos informamos e interactuamos mayormente a través del internet. Las redes sociales son un mecanismo vigente, valido y real de formas de interacción a través de internet. La principal red social en el mundo es Facebook. Pero ¿que tiene que ver Facebook con la medicina? La respuesta es simple puede recolectar datos de pacientes y sus patologías a una velocidad mucho mayor de la que podemos los investigadores en medicina.

Las series de pacientes más grandes sobre las cuales basamos nuestras decisiones médicas y quirúrgicas por ejemplo en fracturas de Calcáneo suman a lo mucho 200 pacientes, cuyos resultados y discusiones se realizan en ámbitos académicos. Específicamente hablando de fracturas de Calcáneo, en Facebook existe un grupo llamado “yo también me fracturé el Calcáneo”, que tiene unos 1200 miembros quienes comparten las experiencias en sus diferentes tratamientos, con resultados generales entre pobres y muy malos.

Evidentemente no es muy científico pensar que un grupo de Facebook puede aportarnos mucha más información que la que puede aportarnos un cirujano de pie debidamente entrenado con la presentación de sus resultados, sin embargo, hay una diferencia diametralmente opuesta entre lo que evidenciamos científicamente y como se sienten los pacientes.

El problema es que los pacientes se comunican entre ellos de manera más rápida y eficiente que los investigadores. La razón es que los investigadores no poseen una herramienta tecnológica que les

permita comunicarse entre ellos a una mayor velocidad y eficiencia. La ciencia se ha ido quedando atrás de las percepciones de los pacientes. (Herrera)

Por definición, ninguna de estas plataformas es mala; al contrario, representan adelantos tecnológicos, que han creado una nueva economía de mercados y de comportamientos sociales que tarde o temprano llegarán a la medicina bajo una u otra presentación. Esto no es nuevo, ha venido sucediendo a lo largo de los últimos 20 años.

Como investigador, uno necesariamente debe preguntarse si ¿será posible obtener estudios poblacionales utilizando los canales de redes sociales, plataformas tecnológicas, aplicaciones, etc.? Tomó 13 años desarrollar la plataforma y conseguir que se ajuste al rigor metodológico de la investigación.

Muchos investigadores realizan sus investigaciones en papel que después digitan en una tabla de Excel, que después tienen que convertir a un archivo “.csv” para que pueda ser tomado por un programa estadístico, siendo este un proceso engorroso y poco eficiente

Tener una herramienta que contenga los protocolos, validar los momentos de recolección de datos y tener los aplicativos de recolección de datos disponibles, con acceso a las bases de datos a través de dispositivos móviles, contando adicionalmente con una forma eficaz de comunicación entre investigadores, permite realizar de una forma dinámica, eficaz, fácil, versátil y global, la recolección de datos; con lo cual mayores niveles de evidencia, agilizando, actualizando y aumentando la calidad del proceso de investigación.

Descripción general

La plataforma web para recolección de datos de investigación es una herramienta mediante la cual se almacenan datos introducidos por los investigadores y mediante la versatilidad de la plataforma se realizarán los seguimientos de la investigación realizando de forma dinámica control de calidad de esta, adicionalmente la exportación de los datos para realizar análisis de éstos mediante software y la asesoría epidemiológica y estadística de cada investigación.

La forma de acceder a la plataforma es mediante la creación de usuarios, cuyos roles serán asignados por el investigador principal de acuerdo con la investigación a realizar.

Oportunidad, y problemática

Dadas las limitaciones para realizar de forma dinámica la recolección de datos, esta plataforma nos permite en tiempo real alimentar las bases de datos de cada investigación, desde cualquier lugar donde cada investigador se encuentre y tenga conexión a internet, sin importar el medio (computador, tablet o smartphone).

Formulación del problema

¿Hay disponibilidad de una herramienta para recolección de datos de investigación versátil, dinámica y modificable de acuerdo con la necesidad del investigador?

En la actualidad la forma de realizar investigación en medicina se basa en la capacidad de llenar formularios y/o encuestas, en forma física o mediante programas tipo hojas de cálculo. La recolección de datos para investigación genera dificultades en el uso adecuado de la información. Estas dificultades están dadas por diversas variables que van desde la capacidad de recolectar, almacenar, disponibilidad, análisis de la información, estandarización de los datos, extracción de bases de datos.

Este tipo de dificultades retrasan los procesos de investigación y así vez el desarrollo de estudios de mayor impacto y globalización de estos.

Justificación del problema

Al encontrar una brecha en la recolección adecuada de los datos para investigación a nivel institucional, surge la necesidad de crear una herramienta útil y eficaz en la recolección de datos de investigación, y de esta forma poder generar un impacto positivo en la forma de realizar investigación.

Al desarrollar esta plataforma web para la recolección de datos de investigación se busca crear a nivel institucional, empezando por el grupo de investigación de ciencias básicas en el sistema músculo-esquelético, una herramienta que nos permite realizar investigaciones a nivel global, con lo cual se podrán realizar estudios multicéntricos no solo a nivel de la ciudad, sino nacional e inclusive internacional; ya que se podrá generar accesos a diversos investigadores que estén asociados a nuestro grupo y realicen aportes significativos para el desarrollo de los trabajos.

Impacto esperado

Cambiar los paradigmas y la forma de recolección de datos de investigación para ser realizados mediante la plataforma web, con lo cual permite que se alimenten las investigaciones dinámicas y globalmente, logrando de esta manera la realización de estudios multicéntricos, con que las investigaciones tendrán mayor peso y evidencia, siendo pioneros en el desarrollo de investigación con este tipo de herramienta.

Descripción del proyecto

El proyecto se realiza mediante el apoyo de ingenieros desarrolladores de software, quienes se basan en una plataforma previamente desarrollada por el grupo de investigación, siguiendo los lineamientos y necesidades que implican la realización de investigación en medicina.

El trabajo conjunto permite el desarrollo de la plataforma de recolección de datos de investigación, la cual será adicionada a la plataforma existente de software de historia clínica médica, la cual tiene como objetivo la integración de los datos relacionados de cada paciente que serán objeto de investigación por parte del grupo de investigación del sistema músculo-esquelético de la Universidad El Bosque. Permitiendo así el mejoramiento continuo de la plataforma y de la tecnología con la que se desarrolla y se mantiene para un desempeño eficiente y eficaz de la recolección y manejo de las bases de datos.

Objetivo general

Desarrollo de plataforma web para la recolección de datos de investigación, la cual será una herramienta dinámica, eficaz, versátil, fácil de utilizar con un impacto global, con lo cual se

modifica la forma de realizar la recolección de la información, para mejorar la capacidad y la calidad de las investigaciones.

Objetivos específicos

1. Crear investigaciones en servidor web con diversas variables modificables, que se encuentra a disposición de los investigadores para crear bases de datos.
2. Almacenamiento de información en las variables diseñadas para cada investigación, con aportes de los diversos investigadores designados por el investigador principal, de forma rápida, eficaz y dinámica, desde diferentes lugares donde se encuentren los investigadores.
3. Realizar controles de calidad de cada investigación a cargo del director de cada proyecto y a su vez del director general de investigación,
4. Extracción de las bases de datos en diversos formatos los cuales se presente para un análisis fácil de acuerdo con la necesidad de cada investigación y que sea compatible con software de análisis de datos.

Metodología

La forma de acceder a la plataforma es creando los usuarios (investigadores), quienes tendrán roles asignados dentro de la plataforma, cuyos parámetros serán asignados de acuerdo con la relación del usuario dentro de la investigación. Siendo el investigador principal quien puede crear los estudios de investigación, asignando los investigadores quienes intervendrán en la misma, creación de las variables, control de la calidad de dicha investigación. Siendo investigador o colaborador podrá ingresar los datos de cada estudio en el cual sea partícipe y evaluar las estadísticas de su trabajo.

Fase de investigación.

Se encuentra en el momento en fase de implementación, donde se están almacenando datos de acuerdo con unas variables iniciales y extrayendo datos para ser introducidos en software especializado en análisis de datos. Se realizará un estudio de cohorte longitudinal con los datos obtenidos desde el servidor a futuro para realizar mejoras al sistema

Marco teórico

Definición de términos y tipos de lenguajes computacionales utilizados en la plataforma:

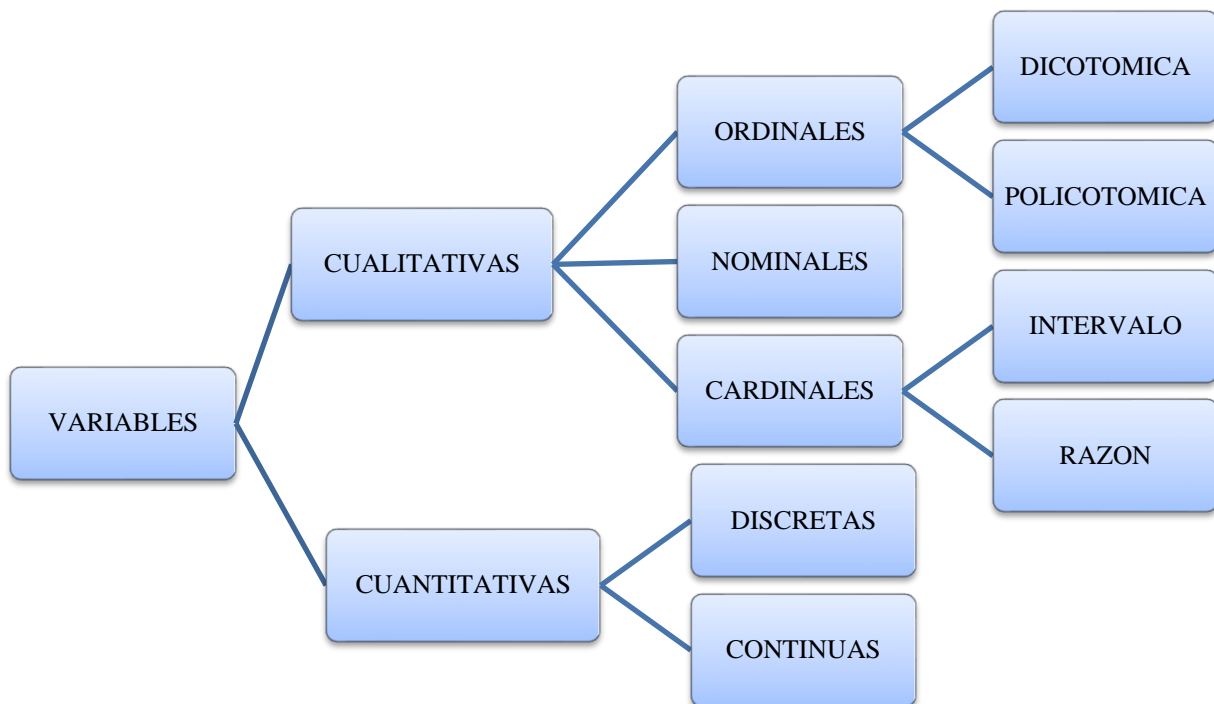
Lenguaje de programación C

C es un lenguaje de programación de propósito general con características de economía de expresión, control de flujo moderno y estructuras de datos, y un rico conjunto de operadores (Kernighan, B. W., & Ritchie, 2006).

Variables

Las variables y las constantes son los objetos de datos básicos manipulados en un programa. Las declaraciones enumeran las variables que se utilizarán y establecen qué tipo tienen y quizás cuáles son sus valores iniciales. Los operadores especifican lo que se les debe hacer. Las expresiones combinan variables y constantes para producir nuevos valores. El tipo de un objeto determina el conjunto de valores que puede tener y qué operaciones se pueden realizar en él (Kernighan, B. W., & Ritchie, 2006).

Variables estadísticas



La distinción entre tipos de variables esta dada en su naturaleza por datos que en su origen son numéricos (cuantitativos) o que describen características (cualitativos), lo cual es importante por lo que la elección de la prueba estadística a aplicar dependerá del tipo de dato recogido. (Henquin, 2013)

Dentro de los datos numéricos los clasificamos dependiendo del uso de números enteros (Variables discretas) o decimales (variables continuas). (Henquin, 2013)

Dentro de los datos categóricos o cualitativos encontramos:

Variables nominales (es decir, categóricas) representan datos clasificados únicamente por el nombre, sin considerar el orden o la magnitud de la variable. (Nordness, 2006)

Variables ordinales pueden considerarse como variables nominales clasificadas en categorías ordenadas o jerarquizadas, pero no necesariamente de acuerdo con la magnitud. Los intervalos (es decir, diferencias) entre las categorías pueden no ser iguales. (Nordness, 2006) Dependiendo de la categorías o resultados posibles se denominan variables dicotómicas o policotómicas. (Henquin, 2013)

Variables cardinales las cuales están jerarquizadas sin embargo se clasifican en variables de intervalo aquellas que están separadas por intervalos o distancias iguales; donde el punto cero no se considera significativo. Y variables de cocientes o de razón, que se clasifican también en grupos ordenados, con intervalos equivalentes entre las variables, pero su punto cero es significativo. (Nordness, 2006)

Estructuras

Una estructura es una colección de una o más variables, posiblemente de diferentes tipos, agrupadas bajo un solo nombre para un manejo conveniente. Las estructuras ayudan a organizar datos complicados, particularmente en programas grandes, porque permiten que un grupo de variables relacionadas se traten como una unidad en lugar de como entidades separadas (Kernighan, B. W., & Ritchie, 2006).

Unix

Unix es un sistema operativo, cuya función es la de organizar las distintas partes de la computadora: el procesador, la memoria integrada, las unidades de disco, los teclados, los monitores de video, etc. para realizar tareas útiles. El sistema operativo es el controlador maestro de la computadora, el pegamento que mantiene unidos todos los componentes del sistema, incluidos los administradores, programadores y usuarios (Byrd, 1997).

GNU - Linux

Linux es un sistema operativo multiplataforma que se ejecuta en muchos modelos de computadora. Su acceso es gratuito y es libre de ser modificado, con lo cual se le permite a los programadores de todo el mundo implementar nuevas funciones y ajustar Linux para mejorar su rendimiento y confiabilidad. Esto es basado en la Free Software Foundation (FSF), una corporación sin fines de lucro que busca promover el software libre y eliminar las restricciones en la copia, redistribución, comprensión y modificación del software (Mccarty, 2000).

Copyleft

Un término muy relacionado con el software libre. Sin embargo, copyleft, se refiere a un tipo de licencias de software libre las cuales permiten la redistribución del software sólo si se garantiza a quien lo recibe libertades iguales a las que otorgó el productor del programa (J. Barahona, E. Rodriguez, N.Rodriguez, M.C.Cagide, N. Fillat Oiz, J. de la Cueva, T.M Torrent, 1999).

PHP

Código diseñado para procesar la información de formularios, generar páginas con contenidos dinámicos, adicionalmente su característica más destacable es la capacidad para proveer soporte para una gran cantidad de bases de datos (Achour et al., 2015).

HTML

HTML es el lenguaje de autoría utilizado para crear documentos de páginas web. Lo que significa que es un sistema para identificar y describir los diversos componentes de un documento, como encabezados, párrafos y listas. El marcado indica la estructura subyacente del documento (Niederst, 2018).

JavaFX

Al hablar de JavaFX necesariamente se debe tener en cuenta JavaFX Mobile el cual es un sistema de software en capas para dispositivos móviles en el que se pueden ejecutar programas desarrollados en JavaFX Script. Siendo este último un lenguaje dirigido a autores de contenido, utilizando scripts simples y declarativos, se pueden crear interfaces de usuario muy ricas. Además de tener secuencias de comandos declarativas, JavaFX Script es un lenguaje totalmente orientado a objetos, completo con métodos y atributos (Weaver, 2007).

SQL (Structured Query Language, Lenguaje estructurado de consultas)

SQL es una herramienta para organizar, administrar y recuperar datos almacenados por una base de datos de computadora. Generando la vinculación de personas, programas y sistemas informáticos a los datos almacenados en una base de datos relacional (Groff & Weinberg, 1999).

Base de datos:

Una base de datos es una colección de datos relacionados entre si, los cuales se interpretan como hechos conocidos que se pueden grabar y que tienen un significado implícito. Creando de esta forma una colección de datos lógicamente coherente cuyo fin es diseñar, construir y alimentar con datos para un propósito específico. De esta forma se dispone de un grupo pretendido de usuarios y algunas aplicaciones en las que esos usuarios están interesados, teniendo a su disposición los datos para el uso en eventos del mundo real. Con lo cual cada usuario puede efectuar transacciones comerciales, producir eventos, cambios en la información almacenada en la base de datos (Elmasri, R. A., & Navathe, 2007).

La información debe organizarse y administrarse para que los usuarios puedan buscar, recuperar

y actualizar los datos que necesiten. Para este propósito es necesario un sistema de administración de datos (DBMS, database management system) el cual es una colección de programas que permite a los usuarios crear y mantener una base de datos. El DBMS es un sistema de software de propósito general que facilita los procesos de definición, construcción, manipulación y compartición de bases de datos entre varios usuarios y aplicaciones (Elmasri, R. A., & Navathe, 2007).

Bases de datos relacionales:

Se basa en el concepto de relación, que se representa físicamente como una tabla, las cuales se usan para contener información acerca de los objetos a representar en la base de datos. Al usar los términos del modelo entidad-relación, los conjuntos de entidades y de relaciones se muestran usando tablas. Una relación se representa como una tabla bidimensional en la que las filas de la tabla corresponden a registros individuales y las columnas corresponden a atributos. Este modelo proporcionar una implementación de sus estructuras de datos y operaciones que permiten tener control de concurrencia, optimización de consulta, gestión de transacción, seguridad e integridad de datos, recuperación, factores humanos e interfaces de usuario.

Bases de datos no relacionales

Las bases de datos NoSQL proporcionan un modelo de datos que encaja mejor con las necesidades de las aplicaciones simplificando así la interacción, lo que resulta en tener que codificar, depurar y evolucionar menos las aplicaciones. adicionalmente la mayoría de las bases de datos NoSQL están diseñadas para ejecutarse sobre clústeres, permitiendo manejar datos a grandes escalas. Operan sin esquema, lo que permite añadir libremente campos a los registros de la base de datos sin tener que redefinir la estructura (Sarasa, 2016).

Marco conceptual

Debido a las características de las investigaciones médicas y en el caso particular del sistema músculo-esquelético en los cuales la cantidad de información a recolectar es de gran tamaño, por los múltiples aportes de los diversos investigadores a participar en cada proyecto, es necesario crear un tipo de plataforma sobre el cual se puede ejecutar de manera apropiada la recolección de los datos.

Siendo el grupo de investigación del sistema músculo-esquelético cuya conformación abarca los residentes de ortopedia y traumatología, medicina física y rehabilitación, y estudiantes de pregrado, es pertinente cimentar la plataforma con un sistema que permita manejar de forma adecuada la información.

Por estas razones anteriormente expuestas se decide realizar el desarrollo de este proyecto bajo la concepción del uso de software libre, dado que permitirá en un futuro realizar modificaciones que van a impactar en el mejor funcionamiento, rendimiento, confiabilidad y evolución de la plataforma. Abriendo paso para que otros grupos de investigación puedan usar esta plataforma para el desarrollo adecuado de las investigaciones.

Teniendo en la cuenta los volúmenes de información a manejar en la plataforma la forma correcta, económica y de mejor desempeño es la utilización de un sistema noSQL, con lo cual permite realizar un escalamiento vertical gracias a la propiedad de estar cargados sobre clusters. Con esto se simplifica la interacción, lo que resulta en tener que codificar, depurar en menor proporción.

Adicionalmente permite añadir nuevos campos de forma libre a los registros de la base de datos, sin la necesidad de modificar la estructura. Lo cual es ventajoso cuando se manejan datos personalizados, ya que por diversos motivos en medicina se tienen que redefinir variables u objetivos de una investigación.

De esta forma se considera utilizar MongoDB que es una base de datos NoSQL orientada al manejo de documentos, cuya organización permite, disponer, identificar, navegar y crear índices, consultas o actualizaciones de forma más satisfactoria. Dando de esta manera mayor versatilidad, y eficiencia en el manejo de los datos, que se almacenan en forma de colecciones en las bases de datos, permitiendo un mejor desempeño al manejar múltiples usuarios que se encuentran en el mismo servidor.

Otra gran ventaja radica en que la Shell de MongoDB que permite interactuar con MongoDB para realizar funciones administrativas, inspeccionar la instancia que se está ejecutando o llevar a cabo determinadas operaciones de gestión sobre la base de datos.

Casos de uso

Caso de uso: Menú investigación

GENERALIDADES CASO DE USO											
Identificador Caso de Uso: Menú investigación						Nombre de Caso de Uso-CU: Menú Investigación					
Propósito: Creación de tablas para bases de datos											
Descripción: Entrada mediante la cual se pueden verificar los datos de la investigación, alimentación de la investigación con nuevos datos aportados por los investigadores. Alimentar las tablas creadas previamente siendo autoincrementales, las cuales están enlazadas en la base de datos y directamente a los pacientes de la investigación con el usuario que alimenta los datos											
Actor (es) Administrador y usuarios											
Prioridad						Complejidad					
Alta	x	Media		Baja		Alta	x	Media		Baja	
Campo		Tipo de Dato		Tamaño	Obligatorio		Valores		Descripción Adicional		
investigaciones					Si				Desplegable con los nombres de las investigaciones registradas		
Ver investigación					Si		Datos de investigación seleccionada previamente		Descripción del estudio de investigación		

Agregar datos			No	Variables en blanco para agregar datos	
Exportar datos			No	Datos de todas las variables	Para descargar en formato de excel y/o CSV
Condiciones					
Previas			Posteriores		
Obligatoriedad de selección de la investigación			permitir continuar agregando datos		
Flujo de Trabajo					
Paso	Actor	Sistema			Sistema de autenticación de credenciales.
1		Despliegue de campo nombre de investigación			
2	Digitar nombre de investigación				
3	Seleccionar siguiente				
4		Despliegue de la investigación seleccionada			
5	seleccionar La investigación buscada	Despliegue del resumen de investigación			
6	Seleccionar siguiente				
		Despliegue de botón de agregar Datos			

	Llenar datos de las variables desplegadas de la investigación		
	Seleccionar Guardar		
		Despliegue de botón Exportar datos	
7	Selección de formato de descarga de las variables de la base de datos		
Excepciones (validaciones)			
Paso	Excepción	Sistema	
		.	
		.	
Requerimientos Especiales (RNF) Y/O Reglas de Negocio			
Historial de Revisiones			
Fecha		Versión	Revisó
Firmas de Elaboración			
Nombre:		Firma:	
Firmas de Aprobación			
Nombre:		Firma:	
Nombre:		Firma:	

Nombre:	Firma:
Firmas de Aceptación	
Nombre: Nombre:	Firma Firma

Caso de uso: Crear de investigación

GENERALIDADES CASO DE USO											
Identificador Caso de Uso: Crear Investigación						Nombre de Caso de Uso-CU: Crear Investigación					
Propósito: Creación de tablas para bases de datos											
Descripción: Entrada mediante la cual se pueden crear las tablas enlazadas en la base de datos los cuales van a ser alimentados por los usuarios											
Actor (es) Administrador											
Prioridad						Complejidad					
Alta	x	Media		Baja		Alta	x	Media		Baja	
Campo	Tipo de Dato	Tamaño	Obligatorio	Valores	Descripción Adicional						
Nombre de investigación			Si		Espacio en blanco para llenar con el nombre de la investigación dado por el grupo						

Fecha de inicio			Si	Fecha	Desplegable con opción de asignar día, mes, año
Investigadores			Si	Datos de los usuarios de la plataforma	Exclusivo para los usuarios que se encuentran inscritos para investigación,
Variables a utilizar			Si		Botón que permita agregar y/o eliminar variables, las cuales definirán los investigadores, que irán a una tabla autoincremental y podrán llenarse con datos alfanuméricos con
Escalas de valoración			no	Alfanuméricos	Datos a ingresar de escalas y clasificaciones utilizadas en ortopedia
Condiciones					
Previas			Posteriores		
Obligatoriedad de completar el campo anterior			permitir continuar agregando datos		
Flujo de Trabajo					
Paso	Actor	Sistema			Sistema de autenticación de credenciales.
1		Despliegue de campo nombre de investigación			
2	Llenar el nombre				
	Agregar resumen de la investigación				

3	Seleccionar siguiente		
4		Despliega botón fecha de inicio de investigación	
5	seleccionar en calendario con fecha (día, mes, año)		
6	Seleccionar siguiente		
7		Despliegue de botón de agregar investigadores	
8	Selección de los investigadores que se encuentran en lista desplegable con los usuarios de la plataforma		
9	Seleccionar siguiente		
10		Despliegue de botón que indica agregar variables	
11	Selección de variable de lista desplegable.		
	Crear variable que no se encuentre en lista		
12	Agregar nueva variable		
13		Crear escala de valoración	

14	introducir valores de la escala a usar		
Excepciones (Validaciones).			
Paso	Excepción	Sistema	
Requerimientos Especiales (RNF) Y/O Reglas de Negocio			
Historial de Revisiones			
Fecha	Versión	Revisó	
Firmas de Elaboración			
Nombre:		Firma:	
Firmas de Aprobación			
Nombre:		Firma:	
Nombre:		Firma:	
Nombre:		Firma:	
Firmas de Aceptación			
Nombre:		Firma	
Nombre:		Firma	

Caso de uso: Calidad

GENERALIDADES CASO DE USO											
Identificador Caso de Uso: Calidad						Nombre de Caso de Uso-CU: Calidad					
Propósito: Permitir la evaluación de la investigación, de los datos ingresados y de los investigadores											
Descripción: Entrada mediante la cual se pueden verificar los datos de la investigación, de los avances, permitir tener datos de fecha, hora de la agregación de los datos de investigación, obtener datos diarios, semanales y mensuales,											
Actor (es) Administrador.											
Prioridad						Complejidad					
Alta	x	Media		Baja		Alta	x	Media		Baja	
Campo	Tipo de Dato	Tamaño	Obligatorio	Valores	Descripción Adicional						
Investigador			Si	Nombre de investigador	Buscador donde se seccione de la lista desplegable el investigador a evaluar						
Nombre de investigación			Si	Nombres de la investigación a evaluar							
Registro de investigación			Si	titulo de la tabla							
Total de pacientes ingresados			Si								

Pacientes ingresados hoy			Si		
total de pacientes programados			Si		
pacientes pendientes por ingresar			Si		
Condiciones					
Previas			Posteriores		
Obligatoriedad de selección del investigador			Desplegar datos		
Flujo de Trabajo					
Paso	Actor	Sistema		Sistema de autenticación de credenciales.	
1		Despliegue de campo nombre de investigador			
2	Digitar /seleccionar nombre del investigador				
3	Seleccionar siguiente				
4		Despliegue campo con las investigaciones en las cuales se encuentra el investigador			
5	Selección de investigación	Llenar los datos de los campos de la tabla Registro de la investigación (Total de pacientes ingresados, Pacientes ingresados hoy, total de pacientes programados, total de pacientes programados)			
Excepciones (Validaciones).					

Paso	Excepción	Sistema	
Requerimientos Especiales (RNF) Y/O Reglas de Negocio			
Historial de Revisiones			
Fecha	Versión	Revisó	
Firmas de Elaboración			
Nombre:		Firma:	
Firmas de Aprobación			
Nombre:		Firma:	
Nombre:		Firma:	
Nombre:		Firma:	
Firmas de Aceptación			
Nombre:		Firma	
Nombre:		Firma	

Arquitectura

Vista física

Actualmente el proyecto se despliega sobre un hosting contratado con godaddy como servicio, el cual provee de un dominio y un servidor manejado por panel.

Por otro lado, las características físicas del servidor dentro del plan contratado proveen un funcionamiento adecuado y robusto con soporte 24/7 que son apropiadas para alojar el desarrollo en cuestión y garantizar el correcto funcionamiento de este, permitiendo escalabilidad de recursos, en caso tal de ser necesario, dichas características son

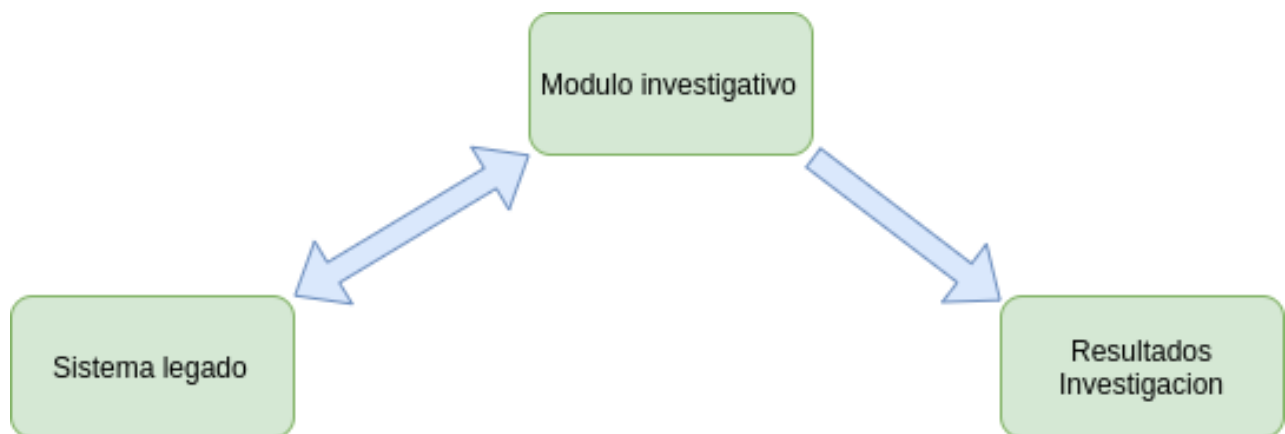
- Sitios web ilimitados
- Almacenamiento ilimitado
- 1 CPU
- 512 MB de memoria RAM
- Monitoreo de seguridad 24/7
- Ancho de banda sin medición
- Sistema operativo LINUX

Vista de componentes

Dentro de la propuesta inicial se contempló obtener información de un sistema legado de atención y agendamiento de pacientes, donde se tienen variadas y acertadas variables tanto de caracterización como de diagnóstico, lo que las hace aprovechables dentro del modulo de investigación.

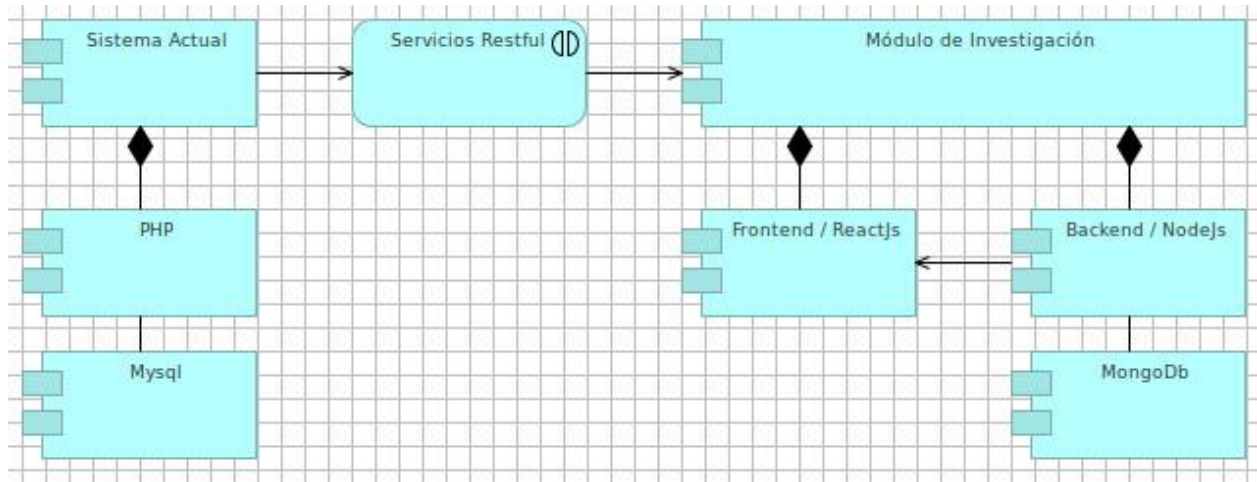
Dentro del módulo de investigación se consideran investigadores, que a su vez dentro del software legado hacen las veces de médicos y se encuentran registrados como tal en el software legado al que se hace referencia, la única variante en cuanto este tipo de actor se refiere se condiciona a que existen investigadores creadores e investigadores asociados, haciendo referencia a estos primeros a los investigadores que crean una investigación con sus diferentes variables y a los asociados a los que van adicionando sujetos de estudio, que en el software legado hacen las veces de pacientes, a la investigación creada y de la cual estos últimos son partícipes.

Por otro lado, los pacientes o sujetos de estudio son parte de la investigación y su vez, así como los investigadores creadores van agregando investigadores asociados a la investigación, los investigadores van agregando sujetos de estudio a esta, que son finalmente la fuente de información de se obtendrán los datos que determinarán los descubrimientos de cada investigación propuesta.



Componentes del servidor

Teniendo en cuenta la necesidad de obtener información de fuentes mixtas donde existe promiscuidad de utilización de bases de datos como de tecnologías se propone el usara esquema de componentes en el servidor:



Componentes dispositivo móvil

Como es habitual, el desarrollo del FrontEnd se realizará en una tecnología vanguardista que es amigables y de tipo responsive que permite tanto el despliegue en navegadores de escritorio como en móviles y adicionalmente el paso amigable y factible a una app, como es el caso de ReactJs.

Vista de diagramas de clase

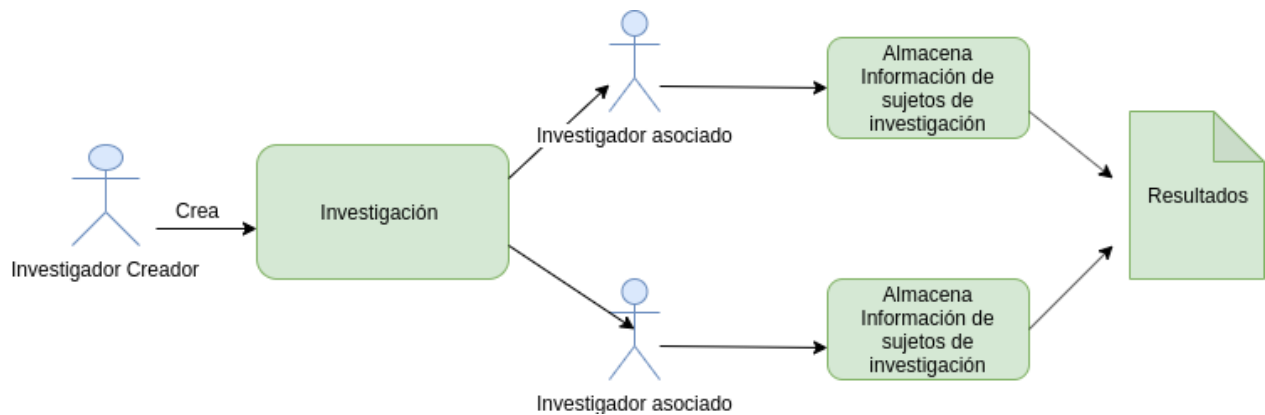
El diagrama de clases de la aplicación se obtendrá del análisis detallado de los casos de uso y de la integración de la aplicación legada desarrollada en php para el uso de esta como fuente de información.

Procesos del sistema

El flujo de información y de como esta viaja y se comporta actualmente en el sistema de citas y agendamiento desarrollado en php aunado a como esta puede influir en el módulo de investigación determinan el proceso del sistema.

Existen varias etapas que pueden ser descritas a continuación:

- **Caracterización:** se realiza desde el software en php, se refiere a la obtención de información característica que define tanto a un paciente como a un médico
- **Segmentación:** Define los roles que adquieren los diferentes actores en la investigación
 - **Investigador creador:** cualquier médico enrolado en el software legado
 - **Investigador asociado:** cualquier médico enrolado en el software legado eh invitado por el investigador creador
 - **Sujeto de estudio:** Cualquier paciente enrolado en el software legado y adicionado a la investigación por el investigador creador o los investigadores asociados.
- **Investigación:** Creada por el investigador creador quien define el núcleo epistemológico de la misma y su razón de ser, adicionalmente define las variables que intervienen en esta.
- **Resultados:** Obtenidos según los criterios de evaluación del investigador creador con base en la información recolectada por los investigadores asociados con base en los resultados de la investigación aplicada a los sujetos de estudio durante el tiempo determinado en la investigación.



Herramientas y frameworks utilizados

Al estudiar el proceso de creación de la herramienta que permita dar vida al módulo de investigativo y al evaluar tanto las fuentes de información como los requerimientos funcionales y no funcionales de las misma se determina utilizar las siguientes herramientas y framework de trabajo citados a continuación:

Frontend / Reactjs:

React.js es una librería JavaScript desarrollada por Facebook. ¿Para qué nos sirve y cómo funciona?

React es una biblioteca escrita en JavaScript, desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario. Se utiliza en Facebook para la producción de componentes, e Instagram está escrito enteramente en React. Uno de sus puntos más destacados, es que no sólo se utiliza en el lado del cliente, sino que también se puede representar en el servidor, y trabajar juntos.

¿Cómo funciona React?

React.js está construido en torno a hacer funciones, que toman las actualizaciones de estado de la página y que se traduzcan en una representación virtual de la página resultante. Siempre que React es informado de un cambio de estado, vuelve a ejecutar esas funciones para determinar una nueva representación virtual de la página, a continuación, se traduce automáticamente ese resultado en los cambios del DOM necesarios para reflejar la nueva presentación de la página.

A primera vista, esto suena como que fuera más lento que el enfoque JavaScript habitual de actualización de cada elemento, según sea necesario. Detrás de escena, sin embargo, React.js hace justamente eso: tiene un algoritmo muy eficiente para determinar las diferencias entre la representación virtual de la página actual y la nueva. A partir de esas diferencias, hace el conjunto mínimo de cambios necesarios en el DOM.

Pues utiliza un concepto llamado el DOM virtual que hace selectivamente subárboles de los nodos sobre la base de cambios de estado, desarrollando esto, con la menor cantidad de manipulación DOM posible, con el fin de mantener los componentes actualizados, estructurando sus datos.

¿Cómo funciona el DOM virtual?

Imagina que tienes un objeto que es un modelo en torno a una persona. Tienes todas las propiedades relevantes de una persona que podría tener, y refleja el estado actual de la persona. Esto es básicamente lo que React hace con el DOM.

Ahora piensa, si tomamos ese objeto y le hacemos algunos cambios. Se ha añadido un bigote, unos bíceps y otros cambios. En React, cuando aplicamos estos cambios, dos cosas ocurren:

- En primer lugar, React ejecuta un algoritmo de "diffing", que identifica lo que ha cambiado.
- El segundo paso es la reconciliación, donde se actualiza el DOM con los resultados de diff.

Lo que hace React, ante estos cambios, en lugar de tomar a la persona real y reconstruirla desde cero, sólo cambiaría la cara y los brazos. Esto significa que si usted tenía el texto en una entrada y un actualización se llevó a cabo, siempre y cuando nodo padre de la entrada no estaba programado para la actualización, el texto se quedaría sin ser cambiado (“¿Cómo funciona React.js?,” n.d.).

Backend / Nodejs:

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google

No ha mejorado mucho la cosa ¿no? Bien, vamos a intentar arreglarlo y ver un resumen de algunas de las características de Node.js que han contribuido al éxito del que disfruta actualmente.

Javascript en el lado del servidor

Es bien sabido que cuando comunicar dos dispositivos (ordenadores, móvil, tablet, IoT) frecuentemente existe una máquina que actúa como servidor, y uno o varios clientes.

Los clientes se comunican con el servidor, y este les devuelve una información, que puede ser desde una página web, datos almacenados en una base de datos, mensaje transmitido por otro cliente... a cualquier cosa.

Tradicionalmente hay lenguajes que se ejecutan en el servidor (PHP, ASP, JSP, Ruby...) y otros que se ejecutan en el cliente (en Web, JavaScript principalmente).

Precisamente una de las primeras características que llaman la atención de Node.js es que "voltea" esta situación, ejecutando Java Script en el lado del servidor.

No es la única tecnología del lado del servidor que funciona con JavaScript (ni siquiera fue el primero) pero sin duda es algo característico de Node.js.

Uno de los motivos para haber elegido JavaScript es que es un lenguaje ampliamente utilizado e íntimamente relacionado con el desarrollo Web. Por este motivo, la curva de aprendizaje resulta sencilla para un gran número de desarrolladores.

Por otro lado, las características de asincronía y diseño orientado a eventos de JavaScript lo hacían apropiado para el propósito principal de Node.js, que veremos justo a continuación.

Centrado en la escalabilidad

Node.js se diseñó con la escalabilidad en mente, en particular, con la capacidad de soportar una gran cantidad de conexiones simultáneas a un servidor.

Muchas tecnologías en el lado del servidor ejecutan el entorno de cada una de las peticiones en un hilo independiente. Cuando el número de peticiones aumentan, aumentan los recursos consumidos en el servidor.

Además de los condicionantes para el rendimiento de un ordenador (RAM, CPU, velocidad de conexión), en un servidor el muchas veces el cuello de botella son los procesos de entradas y salidas (E/S).

Como alternativa, Node.js emplea un único hilo y un bucle de eventos asíncrono. Las nuevas peticiones son tratadas como eventos en este bucle. Este es el motivo por el que las características asíncronas y los eventos de JavaScript encajan tan bien en la filosofía de Node.js

Esto permite que Node.js sea capaz de gestionar múltiples conexiones y peticiones de forma muy eficiente, lo que lo hace apropiado para desarrollo y aplicaciones con un gran número de conexiones simultáneas.

Ejemplos son la elaboración de APIs, aplicaciones web con peticiones Ajax/Websockets, aplicaciones de mensajería push (RabbitMQ, ZeroMQ). Por supuesto, también ocupa un lugar destacado en el "Internet de las cosas" (IoT).

Por el contrario, en general, Node.js no resulta adecuado en aplicaciones que requieran un número reducido de conexiones con un gran consumo de recursos (aplicaciones de cálculo, acceso intensivo a datos, etc).

NPM node package manager

Sin duda, otro de los motivos del éxito de NodeJS es su gestor de paquetes NPM(Node Package Manager). Este nos permite acceder a una enorme cantidad de librerías Open Source desarrolladas por la comunidad (y remarco, ENORME).

Entre los miles de packages que existen tenemos como destacables Express, React, Gulp, Socket.IO, Jade, Mongoose, Browserify, Forever, entre otras muchas. ¡Es todo un ecosistema que deberías conocer!

La lista completa de packages disponibles está en <https://npmjs.org/>.

Un estándar del desarrollo web

¿Ya está, eso todo? ¡Pues no! Aunque Node.js no es ni un servidor web, ni un entorno de desarrollo, lo cierto es que la enorme comunidad y librería de paquetes Open Source disponibles, han hecho que se convierta en una de las herramientas estándar en del desarrollo Web.

Frecuentemente, los desarrolladores Web emplean Node.js, y las múltiples herramientas disponibles, para desarrollar aplicaciones Web de forma rápida. Ejecutar tareas, transcompilar, minificar recursos, sincronización de cambios con el navegador en tiempo real... son sólo algunas de las opciones que tenemos disponibles fácilmente en Node.js y que ahorran una enorme cantidad de tiempo.

Posteriormente, este desarrollo puede ser desplegado en el entorno de producción, bien sea usando el propio Node.js, un servidor Web como Apache o Tomcat, o un incluso introducirlo embebido en un dispositivo IoT.

Conclusión

Node .js es una de las herramientas que marcan tendencia hace años. Si realmente tienes interés en aplicaciones que se comuniquen a través de Internet (incluyendo desarrollo Web e IoT) ya deberías tenerlo instalado.

Sus características lo hacen interesante en sí mismo pero, unido a la enorme cantidad de herramientas y la gran comunidad existente, hace que sea obligado conocer su funcionamiento (“Qué es Node.js y porque ya deberías estar usándolo,” n.d.).

Almacenamiento / MongoDB:

MongoDB es una base de datos orientada a documentos. Esto quiere decir que, en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Los documentos de una misma colección - concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes.

Imaginemos que tenemos una colección a la que llamamos Personas. Un documento podría almacenarse de la siguiente manera:

```
{  
  Nombre: "Pedro",  
  Apellidos: "Martínez Campo",  
  Edad: 22,  
  Aficiones: ["fútbol", "tenis", "ciclismo"],  
  Amigos: [  
    {  
      Nombre: "María",  
      Edad: 22  
    },  
    {  
      Nombre: "Luis",  
      Edad: 28  
    }  
  ]  
}
```

El documento anterior es un clásico documento JSON. Tiene strings, arrays, subdocumentos y números. En la misma colección podríamos guardar un documento como este:

```
{  
  Nombre: "Luis",  
  Estudios: "Administración y Dirección de Empresas",
```


Amigos:12

}

Este documento no sigue el mismo esquema que el primero. Tiene menos campos, algún campo nuevo que no existe en el documento anterior e incluso un campo de distinto tipo.

Esto que es algo impensable en una base de datos relacional, es algo totalmente válido en MongoDB.

¿Cómo funciona MongoDB?

MongoDB está escrito en C++, aunque las consultas se hacen pasando objetos JSON como parámetro. Es algo bastante lógico, dado que los propios documentos se almacenan en BSON. Por ejemplo:

```
db.Clientes.find({Nombre:"Pedro"});
```

La consulta anterior buscará todos los clientes cuyo nombre sea Pedro.

MongoDB viene de serie con una consola desde la que podemos ejecutar los distintos comandos. Esta consola está construida sobre JavaScript, por lo que las consultas se realizan utilizando ese lenguaje. Además de las funciones de MongoDB, podemos utilizar muchas de las funciones propias de JavaScript. En la consola también podemos definir variables, funciones o utilizar bucles.

Si queremos usar nuestro lenguaje de programación favorito, existen drivers para un gran número de ellos. Hay drivers oficiales para C#, Java, Node.js, PHP, Python, Ruby, C, C++, Perl o Scala. Aunque estos drivers están soportados por MongoDB, no todos están en el mismo estado de madurez. Por

ejemplo, el de C es una versión alpha. Si queremos utilizar un lenguaje concreto, es mejor revisar los drivers disponibles para comprobar si son adecuados para un entorno de producción.

¿Dónde se puede utilizar MongoDB?

Aunque se suele decir que las bases de datos NoSQL tienen un ámbito de aplicación reducido, MongoDB se puede utilizar en muchos de los proyectos que desarrollamos en la actualidad.

Cualquier aplicación que necesite almacenar datos semi estructurados puede usar MongoDB. Es el caso de las típicas aplicaciones CRUD o de muchos de los desarrollos web actuales.

Eso sí, aunque las colecciones de MongoDB no necesitan definir un esquema, es importante que diseñemos nuestra aplicación para seguir uno. Tendremos que pensar si necesitamos normalizar los datos, denormalizarlos o utilizar una aproximación híbrida. Estas decisiones pueden afectar al rendimiento de nuestra aplicación. En definitiva, el esquema lo definen las consultas que vayamos a realizar con más frecuencia.

MongoDB es especialmente útil en entornos que requieran escalabilidad. Con sus opciones de replicación y sharding, que son muy sencillas de configurar, podemos conseguir un sistema que escale horizontalmente sin demasiados problemas.

¿Dónde no se debe usar MongoDB?

En esta base de datos no existen las transacciones. Aunque nuestra aplicación puede utilizar alguna técnica para simular las transacciones, MongoDB no tiene esta capacidad. Solo garantiza operaciones

atómicas a nivel de documento. Si las transacciones son algo indispensable en nuestro desarrollo, deberemos pensar en otro sistema.

Tampoco existen los JOINS. Para consultar datos relacionados en dos o más colecciones, tenemos que hacer más de una consulta. En general, si nuestros datos pueden ser estructurados en tablas, y necesitamos las relaciones, es mejor que optemos por un RDBMS clásico.

Y para finalizar, están las consultas de agregación. MongoDB tiene un framework para realizar consultas de este tipo llamado Aggregation Framework. También puede usar Map Reduce. Aún así, estos métodos no llegan a la potencia de un sistema relacional. Si vamos a necesitar explotar informes complejos, deberemos pensar en utilizar otro sistema. Eso sí, esta es una brecha que MongoDB va recortando con cada versión. En poco tiempo esto podría dejar de ser un problema.

¿Cómo se instala MongoDB?

La instalación de una instancia del servidor es un juego de niños. Simplemente tenemos que bajar los binarios para nuestro sistema operativo. Hay versiones par Windows, Linux y MacOS. Una vez bajados podremos arrancar el servicio de MongoDB con un solo comando.

```
mongod --dbpath data
```

Con este comando arrancamos el servicio mongod, que empezará a escuchar peticiones por el puerto 27017. Es importante indicar el parámetro --dbpath, con la ruta dónde se almacenarán los ficheros de nuestra base de datos.

¿Cómo puedo usar la consola?

Si ya tenemos el servidor lanzado en nuestra máquina, bastará con lanzar desde la consola el siguiente comando:

```
mongo localhost
```

Desde ese momento entraremos en la consola y podremos realizar consultas. Si escribimos `help` tendremos un listado con los comandos más comunes y su descripción (“MongoDB. Qué es, cómo funciona y cuándo podemos usarlo (o no),” n.d.).

Criterios de elección

Al realizar un estudio detallado de las diferentes opciones del mercado y evaluar en cuanto a las necesidades de modularidad y escalabilidad se determinó utilizar las herramientas y frameworks anteriormente descritos en el apartado anterior toda vez que permiten en estos ámbitos la mayor versatilidad posible y han sido probados con suficiencia como efectivos para este tipo de desarrollos.

Análisis de los resultados

Se obtiene un sistema completamente operativo de una plataforma de software que permite crear las variables de una investigación, adicionada a una plataforma médica desarrollada por el grupo de investigación.

Resumen

La plataforma web para recolección de datos de investigación, ha tenido un desarrollo y propósito, como una herramienta que permita realizar de una forma dinámica, eficaz, fácil, versátil y global, la recolección de datos de investigación, con lo cual se puede avanzar en el tiempo que toma la recolección, almacenamiento y control de los datos, a su vez poder realizar análisis de las bases de datos, de acuerdo a las variables de cada trabajo, y poder exportar las mismas a los diferentes software de análisis, con lo cual se realizará de una manera más eficiente la investigación.

Adicionalmente realizar estudios multicéntricos, con el aporte diferentes investigadores dentro de la ciudad, del país y fuera del país, con lo cual se avanza en la calidad de los estudios, aumentando el nivel de evidencia e impacto.

La forma de acceder a la plataforma es creando los usuarios (investigadores), quienes tendrán roles asignados dentro de la plataforma, cuyos parámetros serán asignados de acuerdo con la relación del usuario dentro de la investigación. Siendo el investigador principal quien puede crear los estudios de investigación, asignando los investigadores quienes intervendrán en la misma, creación de las variables, control de la calidad de dicha investigación. Siendo investigador o colaborador podrá ingresar los datos de cada estudio en el cual sea partícipe y evaluar las estadísticas de su trabajo.

Con esto resolver los principales inconvenientes que retrasan los procesos de investigación y a su vez el desarrollo de estudios de mayor impacto y globalización de estos, cerrando cada vez más la brecha que existe para realizar investigaciones de forma adecuada, creando un impacto positivo en nuestra forma de investigar.

Bibliografía

¿Cómo funciona React.js? (n.d.). Retrieved November 15, 2019, from

<https://devcode.la/blog/como-funciona-reactjs/>

Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., ... Vrana, J. (2015).

PHP Manual de PHP - Manual.

Byrd, J. (1997). A Basic UNIX Tutorial. In *Computer.*

Elmasri, R. A., & Navathe, S. B. (2007). *Fundamentos de sistemas de bases de datos* (5ta ed.).

Addison Wesley.

Groff, J. R., & Weinberg, P. N. (1999). SQL: The Complete Reference. In *Processing.*

Henquin, R. P. (2013). *Epidemiología y Estadística para principiantes*. Corpus Libros Médicos y Científicos.

J. Barahona, E. Rodriguez, N.Rodriguez, M.C.Cagide, N. Fillat Oiz, J. de la Cueva, T.M Torrent,

E. M. (1999). *Copyleft Manual de uso.*

Kernighan, B. W., & Ritchie, D. M. (2006). *The C programming language.*

Mccarty, B. B. (2000). Learning debian GNU/linux. In *Computers & Mathematics with*

Applications (Vol. 39).

MongoDB. Qué es, cómo funciona y cuándo podemos usarlo (o no). (n.d.). Retrieved November

15, 2019, from <https://www.genbeta.com/desarrollo/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>

Niederst, J. (2018). *Learning Web Design ABegiiers Guide to HTML, CSS, JAVASCRIPT.*

Nordness, R. J. (2006). *Epidemiologia y bioestadística secretos*. Elsevier.

Qué es Node.js y porque ya deberías estar usándolo. (n.d.). Retrieved November 15, 2019, from

<https://www.luisllamas.es/que-es-node-js/>

Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. In *Journal of*

Chemical Information and Modeling (Vol. 53). Editorial UOC.

Weaver, J. L. (2007). *Javafx script: Dynamic java scripting for rich internet/client-side applications*.

Anexos:

Registros de Dirección Nacional de Derechos de Autor [de Plataforma Existente](#):