



UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SOFTWARE



Desarrollo Web Multicapa
José Alfonso Barroso Barajas

Tecnologías actuales en el desarrollo web
Proyecto

301403
Nohemí González Macías

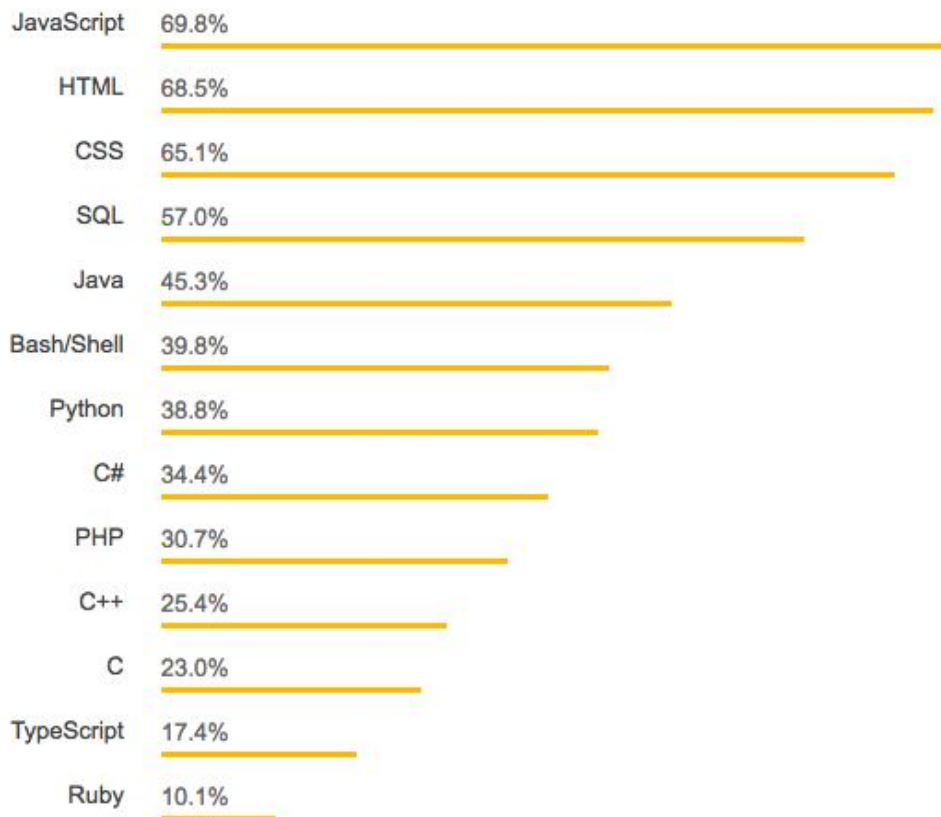
Parciales 1, 2
03 de abril, 2019

Contenido

Tecnologías actuales en el desarrollo web	3
Frontend	4
SPA (Single Page Application)	4
PWAs (Progressive Web App)	4
Frameworks	5
Angular	5
React	6
Vue	6
Backend	8
Frameworks	8
Django	8
Laravel	9
Spring	10
Datos	11
SQLAlchemy	11
Eloquent	12
Hibernate	12
Seguridad	15
Tokens	15
Cookies	16
Indexeddb	16
Referencias	18

Tecnologías actuales en el desarrollo web

Stack Overflow Insights es una propuesta de Stack Overflow con el propósito de obtener información general de los integrantes de la comunidad sobre aspectos como: perfil profesional, condiciones sociales, hábitos, tecnologías entre otros. Los datos más recientes son del periodo enero a junio del año 2018, arrojando los siguientes resultados:



Lenguajes de scripting y de maquetado más populares



Frameworks, librerías y herramientas más populares

Frontend

SPA (Single Page Application)

Las aplicaciones de una sola página (SPA) son aplicaciones web que cargan una sola página HTML y actualizan dinámicamente esa página a medida que el usuario interactúa con la aplicación.

Los SPA utilizan AJAX y HTML5 para crear aplicaciones web fluidas y receptivas, sin recargas de página constantes. Sin embargo, esto significa que gran parte del trabajo ocurre en el lado del cliente, en JavaScript.

PWAs (Progressive Web App)

Son experiencias que combinan lo mejor de la Web y de las apps. Están disponibles para los usuarios a partir de la primera visita en una pestaña del navegador y no requieren instalación.

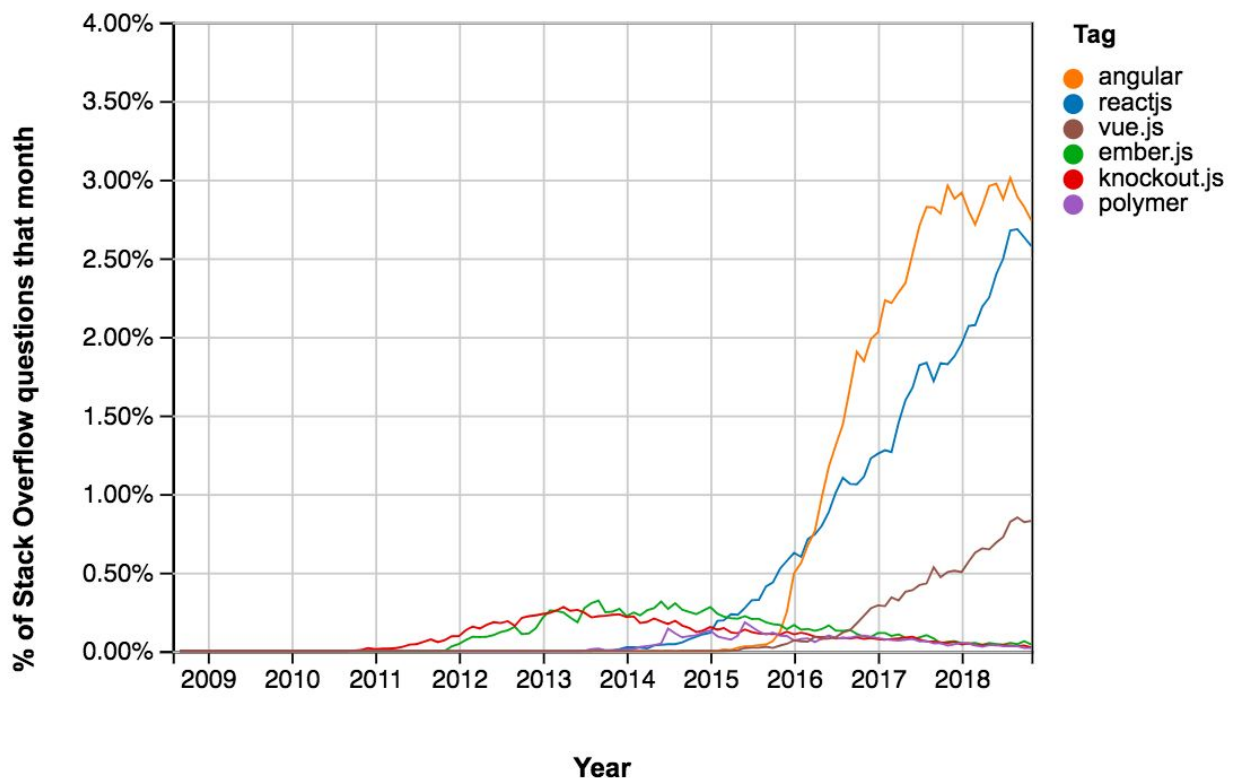
Una Progressive Web App es:

- Progresiva: funciona para todos los usuarios, sin importar la elección de navegador, porque está construida con mejora progresiva como principio central.
- Adaptable: se adapta a cualquier factor de formulario, sea escritorio, móvil, tablet o lo que venga en el futuro.
- Independiente de la conectividad: mejorada con service workers para trabajar sin conexión o con redes de mala calidad.
- Estilo app: al usuario le parece una app con interacciones y navegación estilo app, porque está construida con modelo de shell de app.
- Fresca: siempre actualizada gracias al proceso de actualización de service worker.
- Segura: emitida vía HTTPS para evitar intromisiones y para garantizar que el contenido no se haya manipulado.
- Descubrible: se puede identificar como "app" gracias al manifiesto W3C y al alcance de registro de service worker, lo que permite que los motores de búsqueda la encuentren.

- Posibilidad de volver a interactuar: facilita la posibilidad de volver a interactuar a través de funciones como notificaciones push.
- Instalable: les permite a los usuarios *conservar* las apps que les resultan más útiles en su pantalla principal sin la molestia de una tienda de app.
- Vinculable : se puede compartir fácilmente via URL, no requiere instalación compleja.

Frameworks

En este documento, hablaremos de los tres frameworks para desarrollo frontend más populares.



Frameworks comunes para desarrollar y su popularidad en la comunidad de Stack Overflow

Angular

Angular es una plataforma que facilita la creación de aplicaciones con la web, creada por Google. Combina plantillas declarativas, inyección de dependencias, herramientas de extremo a extremo y mejores prácticas integradas para resolver los

desafíos de desarrollo. Angular permite a los desarrolladores crear aplicaciones que viven en la web, el móvil o el escritorio.

La finalidad de Angular es convertir el desarrollo de una web SPA en un proceso sencillo, limpio, más orientado a objetos así como con una arquitectura basada en componentes y buenas prácticas. También ayuda a la separación completa entre el frontend y el backend en una aplicación web.

React

Es una biblioteca escrita en JavaScript desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario.

React hace que sea fácil crear interfaces de usuario interactivas. Al diseñar vistas simples para su aplicación, React actualizará y representará de manera eficiente los componentes correctos cuando sus datos cambien. No sólo se utiliza en el lado del cliente, sino que también se puede representar en el servidor, y trabajar juntos.

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. Para ello utiliza un DOM virtual que hace selectivamente sub-árboles 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.

Vue

Vue es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde el inicio para ser adoptado incrementalmente. La biblioteca principal se enfoca solo en la capa de la vista, y es

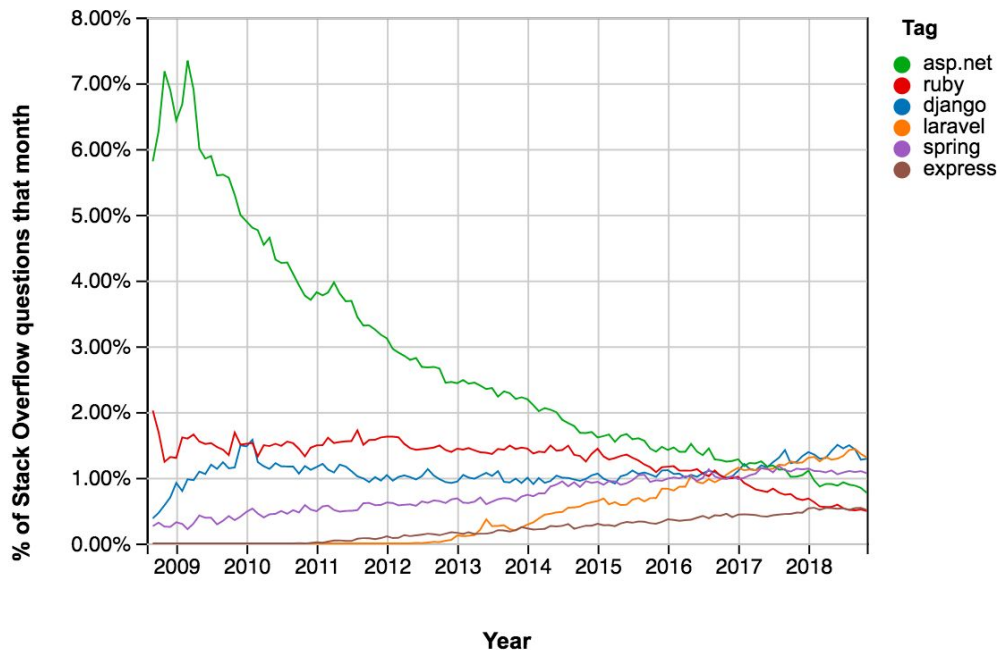
muy simple de utilizar e integrar con otros proyectos o bibliotecas existentes. Por otro lado, Vue también es perfectamente capaz de soportar aplicaciones SPA.

Es un framework que ha tomado de inspiración a otros como React, Angular, Ember, Knockout, Polymer y Riot, buscando hacer de forma más eficiente el trabajo de la actualización de datos y proveyendo de más escalabilidad a las aplicaciones creadas con Vue.

Backend

Frameworks

En este documento, hablaremos de los tres frameworks para desarrollo frontend más populares.



Frameworks comunes para desarrollar y su popularidad en la comunidad de Stack Overflow

Django

Django es un marco web de Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Según documentación oficial, es un framework para desarrollar todo tipo de aplicaciones, desde sistemas de administración, redes sociales y plataformas de computación científica, de manera rápida así con grandes opciones de escalabilidad. Por otro lado, también facilita la seguridad de las aplicaciones, como inyecciones SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking.

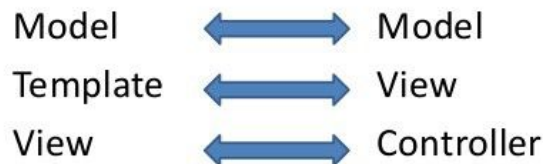
Otras características que lo diferencian de otros frameworks es que Django se jacta de usar una arquitectura de *nada compartido*, lo que significa que puede agregar

hardware a cualquier nivel: servidores de bases de datos, servidores de almacenamiento en caché o servidores de aplicaciones/Web.

Por otro lado, puede parecer un poco confuso, ya que Django cambia los términos MVC por MTV:

Is it MVC or MTV??

- In Django it is called MTV rather than MVC.



Models	Describes your data
Views	Controls what users sees
Templates	How user sees it
Controller	URL dispatcher

6/4/2015

6

Arquitectura Web propuesta por Django

Laravel

Laravel es un framework para construir aplicaciones web basadas en el modelo MVC con PHP, el cual presume de brindar la posibilidad de crear aplicaciones de forma rápida, con una sintaxis expresiva, legible y elegante, permitiendo concentrarse en la aplicación y no tanto en el lenguaje de programación.

Otros recursos que ofrece laravel son también:

- Laravel cashier

Facilita la facturación de suscripciones con las integraciones integradas de Stripe y Braintree. Cupones, intercambio de suscripciones, cancelaciones e incluso facturas en PDF listos para usar.

- Valet

Un entorno de desarrollo Laravel para Mac minimalistas. No vagabundo, no apache, sin problemas.

- Mix
Facilita el frontend para usar SASS y Webpack en minutos.
- Lumen
Si todo lo que necesitas es una API y una velocidad de rayo, prueba Lumen.
Es Laravel super-light.
- Spark
Potentes andamios de aplicación SaaS.

Spring

Spring Framework proporciona un completo modelo de programación y configuración para aplicaciones empresariales modernas basadas en Java, en cualquier tipo de plataforma de implementación.

Un elemento clave de Spring es el soporte de infraestructura a nivel de aplicación: Spring se enfoca en la "plomería" de las aplicaciones empresariales para que los equipos puedan enfocarse en la lógica de negocios a nivel de aplicación, sin vínculos innecesarios con entornos de implementación específicos. Entre sus características se encuentran:

- Tecnologías principales
Inyección de dependencias, eventos, recursos, i18n, validación, enlace de datos, conversión de tipos, SpEL, AOP.
- Pruebas
Objetos simulados, marco TestContext, Spring MVC Test, WebTestClient.
- Acceso a datos
Transacciones, soporte DAO, JDBC, ORM, Marshalling XML.
- Spring MVC y los marcos web de Spring WebFlux.
- Integración
Remoto, JMS, JCA, JMX, correo electrónico, tareas, programación, caché.
- Lenguajes
Kotlin, Groovy, lenguajes dinámicos.

Datos

Para los frameworks de backend citados anteriormente, sus respectivas tecnologías para manejar la capa de datos en bases de datos relacionales son los siguientes.

SQLAlchemy

SQLAlchemy es el kit de herramientas SQL de Python y el Mapeador relacional de objetos que ofrece a los desarrolladores de aplicaciones el máximo poder y flexibilidad de SQL.

Proporciona un conjunto completo de patrones de persistencia conocidos a nivel empresarial, diseñados para el acceso a bases de datos eficientes y de alto rendimiento, adaptados a un lenguaje de dominio simple y de Pythonic.

Las bases de datos SQL se comportan menos como las colecciones de objetos, mientras más tamaño y rendimiento comienzan a importar; las colecciones de objetos se comportan menos como tablas y filas, mientras más abstracción comienza a importar. SQLAlchemy tiene como objetivo dar cabida a estos dos principios.

SQLAlchemy considera que la base de datos es un motor de álgebra relacional, no sólo una colección de tablas. Las filas se pueden seleccionar no solo de tablas sino también de uniones y otras declaraciones de selección; cualquiera de estas unidades se puede componer en una estructura más grande. El lenguaje de expresión de SQLAlchemy se basa en este concepto desde su núcleo.

SQLAlchemy es más famoso por su mapeador de objetos relacional (ORM), un componente opcional que proporciona el patrón del mapeador de datos, donde las clases se pueden mapear en la base de datos de forma abierta, de múltiples maneras, lo que permite que el modelo de objetos y el esquema de la base de datos se desarrollen en una limpiamente desacoplado camino desde el principio.

Eloquent

En Laravel podemos hacer uso de un ORM llamado Eloquent, un ORM es un Mapeo Objeto-Relacional por sus siglas en inglés (Object-Relational mapping), que es una forma de mapear los datos que se encuentran en la base de datos almacenados en un lenguaje de script SQL a objetos de PHP y viceversa, esto surge con la idea de tener un código portable con el que no tengamos la necesidad de usar lenguaje SQL dentro de nuestras clases de PHP.

Eloquent es el ORM que incluye Laravel para manejar de una forma fácil y sencilla los procesos correspondientes al manejo de bases de datos en nuestro proyecto, gracias a las funciones que provee podremos realizar complejas consultas y peticiones de base de datos sin escribir una sola línea de código SQL.

```
1 public function store(Request $request)
2 {
3     $movie = new Movie;
4     $movie->name = $request->name;
5     $movie->description = $request->description;
6     $movie->save();
7     return redirect('movie');
8
9 }
```

Ejemplo de una petición tipo POST

Hibernate

La característica principal de Hibernate es su soporte para el modelado basado en objetos, lo que le permite ofrecer un mecanismo transparente para la persistencia. Se utiliza XML para mapear una base de datos a una aplicación y soporta objetos de grano fino. La versión actual de Hibernate es compatible con anotaciones de Java y, por tanto, satisface la especificación de EJB.

Hibernate incluye un lenguaje de consulta muy poderoso llamado Hibernate Query Language o HQL. HQL es muy similar a SQL, y también define algunos convenios adicionales. HQL es completamente orientado a objetos, lo que le permite

aprovechar la fuerza completa de la orientación a objetos y aprovecharse de la herencia, polimorfismo y asociación. HQL devuelve resultados de la consulta como objetos que se puede acceder directamente y ser manipulados por el programador. HQL también soporta muchas características avanzadas de la paginación y los perfiles dinámico que SQL nunca ha soportado.

Para almacenar y recuperar objetos de la base de datos, el desarrollador debe mantener una conversación con el motor de Hibernate mediante un objeto especial, quizás el concepto clave más importante dentro de Hibernate, llamado Sesión (clase `Session`). Se puede equiparar, a grandes rasgos, al concepto de conexión de JDBC y cumple un papel muy parecido, es decir, sirve para delimitar una o varias operaciones relacionadas dentro de un proceso de negocio, demarcar una transacción y aporta algunos servicios adicionales, como manejo de caché, para evitar interacciones innecesarias contra la BD. En este sentido veremos que la clase `Session` ofrece métodos como `save (Object object)`, `createQuery (String queryString)`, `beginTransaction()`, `close()`, entre otros.

El diseño está pensado para interactuar con la base de datos tal como estamos acostumbrados a hacerlo con una conexión JDBC, pero con una diferencia: mayor simplicidad, es decir, guardar un objeto, por ejemplo, consiste en algo así como `session.save(miObjeto)`, sin necesidad de especificar una sentencia SQL.

Hibernate distingue entre objetos tipo `transient` y tipo `persistent`: los primeros son objetos que sólo existen en memoria y no en un almacén de datos, en algunos casos no serán almacenados jamás en la base de datos y en otros es un estado en el que se encuentran hasta ser almacenados en ella. Los segundos se caracterizan por haber sido ya almacenados y ser, por tanto, objetos persistentes. Dicho de otra manera, los objetos `transient` han sido instanciados por el desarrollador sin haberlos almacenado mediante una sesión, los objetos persistentes han sido creados y almacenados en una sesión o bien devueltos en una consulta realizada con la sesión.

Igual que con las conexiones JDBC, tenemos que crear y cerrar sesiones, aunque no hay una relación 1:1 entre sesiones y conexiones, es decir, no tenemos que abrir y cerrar simultáneamente sesiones y conexiones JDBC, la política a seguir dependerá del contexto del proceso de negocio de cada situación dándonos

Hibernate amplias posibilidades para la implementación de nuestras políticas (conexiones JDBC gestionadas por la aplicación, por Hibernate, por un posible servidor de aplicaciones, etc), siendo solamente necesario en la práctica crear y cerrar explícitamente las sesiones de Hibernate.

```
@Override
public long save(Book book) {
    sessionFactory.getCurrentSession().save(book);
    return book.getId();
}

@Override
public Book get(long id) {
    return sessionFactory.getCurrentSession().get(Book.class, id);
}
```

Ejemplo de llamadas para guardar y consultar un objeto en Hibernate

Seguridad

Tokens

Para evitar que en cada llamada el usuario deba entregar su clave y password se usa un token, que no es más que una cadena de texto bien larga, encriptada con una clave. Según se hace el login, el servidor devuelve al cliente un token y el cliente en futuros accesos entrega ese token para certificar que está autenticado.



En el caso que se realizara un intento correcto de autenticación, y por tanto el servidor nos devolviese el token, en el lado del cliente se deberá almacenar el token, para poder usarlo más tarde.

Lo ideal será almacenar el token en el navegador, en un lugar donde se pueda acceder a él en futuras consultas. Es decir, sería interesante proveer algún sistema de persistencia, para que por ejemplo, si el usuario refresca la página o accede a la misma web pasados unos minutos, se disponga del token generado anteriormente y no se tenga que obligar al usuario a pasar de nuevo por el proceso de login. Lo más normal para almacenar el token sería el LocalStorage del navegador.

Uno de los estándares para producir los token del lado del servidor es JSON Web Token. Básicamente especifica toda una operativa, que se debe implementar del lado del servidor, para producir tales token y verificar su validez cuando sea necesario.

El estándar JSON Web Token indica que los token del lado del servidor se crean mediante una cadena de texto encriptada por una clave, que se mantiene en secreto del lado del servidor. Esa misma clave se debe de usar para desencriptar el token y verificar su validez.

El token en sí, una vez descriptado, permite conocer cosas sobre el usuario que ha realizado la solicitud, es decir, es una manera de mantener los datos de sesión y saber cosas del usuario, como su email, nombre o cualquier otra información que la aplicación juzgue necesaria.

Cookies

Una cookie HTTP, cookie web o cookie de navegador es una pequeña pieza de datos que un servidor envía al navegador web del usuario. El navegador guarda estos datos y los envía de regreso junto con la nueva petición al mismo servidor. Las cookies se usan generalmente para decirle al servidor que dos peticiones tiene su origen en el mismo navegador web lo que permite, por ejemplo, mantener la sesión de un usuario abierta. Las cookies permiten recordar la información de estado en vista a que el protocolo HTTP es un protocolo sin estado.

Las cookies se utilizan principalmente con tres propósitos:

- **Gestión de Sesiones**
Inicios de sesión, carritos de compras, puntajes de juegos o cualquier otra cosa que el servidor deba recordar.
- **Personalización**
Preferencias de usuario, temas y otras configuraciones.
- **Rastreo**
Guardar y analizar el comportamiento del usuario.

Las cookies se usaron una vez para el almacenamiento general del lado del cliente. Si bien esto era legítimo cuando eran la única forma de almacenar datos en el cliente, hoy en día se recomienda preferir las API de almacenamiento modernas. Las cookies se envían con cada solicitud, por lo que pueden empeorar el rendimiento (especialmente para las conexiones de datos móviles).

Indexeddb

IndexedDB es una manera de almacenar datos dentro del navegador del usuario. Debido a que permite la creación de aplicaciones con habilidades de consulta

enriquecidas, con independencia de la disponibilidad de la red, sus aplicaciones pueden trabajar tanto en línea como fuera de línea.

El patrón básico que indexedDB propone es:

1. Abrir una base de datos.
2. Crear un objeto de almacenamiento en la base de datos.
3. Iniciar una transacción y hacer una petición para hacer alguna operación de la base de datos, tal como añadir o recuperar datos.
4. Espere a que se complete la operación por la escucha de la clase correcta de eventos DOM .
5. Hacer algo con el resultado (El cual puede ser encontrado en el objeto de la petición).

Referencias

Stack Overflow Developer Survey 2018. (2019). Recuperado desde

<https://insights.stackoverflow.com/survey/2018>

Wasson, M. (2019). ASP.NET - Single-Page Applications: Build Modern, Responsive Web Apps with ASP.NET. Recuperado desde:

<https://msdn.microsoft.com/en-us/magazine/dn463786.aspx?f=255&MSPPErr=-2147217396>

LePage, P. (2019). Tu primera Progressive Web App | Web | Google Developers. Recuperado desde:

<https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=es>

Grados Caballero, J. (2019). ¿Cómo funciona React.js?. Recuperado desde

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

Robles, V. (2019). ¿Que es Angular y para qué sirve? - Victor Robles. Recuperado desde:

<https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>

Introducción — Vue.js. (2019). Recuperado desde:

<https://es-vuejs.github.io/vuejs.org/v2/guide/>

Django overview | Django. (2019). Recuperado desde:

<https://www.djangoproject.com/start/overview/>

Django documentation | Django documentation | Django. (2019). Recuperado desde:

<https://docs.djangoproject.com/en/2.1/>

Otwell, T. (2019). Laravel - The PHP Framework For Web Artisans. Recuperado desde:

<https://laravel.com/>

Spring Projects. (2019). Recuperado desde

<https://spring.io/projects/spring-framework>

Ochoa, J. (2019). Aprende a usar Eloquent el ORM de Laravel. Recuperado desde:

<https://styde.net/aprende-a-usar-eloquent-el-orm-de-laravel/>

Capítulo 7. Modelos y uso de Eloquent · laravel-5. (2019). Recuperado desde:

<https://richos.gitbooks.io/laravel-5/content/capitulos/chapter7.html>

Referencia a Hibernate | Marco de Desarrollo de la Junta de Andalucía. (2019).

Recuperado desde:

<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/178>

Alvarez, M. (2019). Autenticación por token. Recuperado desde:

<https://desarrolloweb.com/articulos/autenticacion-token.html>

Usando IndexedDB. (2019). Recuperado desde:

https://developer.mozilla.org/es/docs/IndexedDB-840092-dup/Usando_IndexedDB

HTTP cookies. (2019). Recuperado desde:

<https://developer.mozilla.org/es/docs/Web/HTTP/Cookies>