

AllMarket.net

Servicio Web: Proyecto de sistemas distribuidos

Omar Montero

CI: V-24.035.179

omarjo94@gmail.com

Alexis Maita

CI: V- 22.574.538

ajmaitaf@gmail.com

Resumen— En el siguiente Artículo, se vera todo lo necesario para desarrollar la pagina web AllMarket.net, la cual funcionara, haciendo uso de servicios web, la principal forma en que estos son implementados, es haciendo uso de una API (Application Programming Interface) de arquitectura REST(Representational State Transfer). AllMarket.net es un sistema distribuido, debido a esto el siguiente articulo, esta muy enfocado en describir que tecnolgias se usan para poder implementarlo, asi como la arquitectura del sistema y como funciona.

Palabras Clave: *Sistemas Distribuidos, Frameworks, API's, Servidor, Cliente*

1 Introducción

En el mundo del desarrollo de software, siempre ha existido la necesidad, de evolucionar a la hora de poder prestar un servicio al usuario, con la introducción de la web 2.0 en el mercado, se introdujeron conceptos y arquitecturas los cuales permiten desarrollar sistemas no solo de manera mas rápida y eficiente, ademas permiten implementar conceptos de sistemas distribuidos, de manera mas fácil.

Entre uno de estas formas de implementar un sistema distribuido, son los servicios web, estos a pesar de tener bastante tiempo en el mercado, con la introducción de la web 2.0 logran mejorar de manera significativa, brindando así herramientas las cuales serán de gran utilidad para el desarrollador.

Por eso en el siguiente articulo se hará una descripción de la pagina web AllMarket.net tanto a nivel de funcionamiento, así como a nivel de arquitectura y desarrollo, ademas AllMarket.net sera un proyecto el cual permitirá implementar un sistema distribuido, utilizando el servicio web como tecnología

principal.

2 Revisión Teórica

Sistemas Distribuidos

Un sistema distribuido es aquel donde sus componentes localizados en los computadores unidos mediante una red, comunican y coordinan sus acciones solo por pase de mensajes. (Coulourous G, 2012)

Interfaz de Programación de Aplicaciones

Una **API** es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro *software*. Las siglas API vienen del inglés *Application Programming Interface*. En español sería Interfaz de Programación de Aplicaciones.

Una API nos permite implementar las funciones y procedimientos que engloba en nuestro proyecto sin la necesidad de programarlas de nuevo. En términos de programación, es una **capa de abstracción**. [1]

Arquitectura de Transferencia de Estado Representacional

El término **Transferencia de Estado Representacional** (Representational State Transfer) o **REST**, fue introducido y definido en el año 2000 por Roy Fielding en su tesis doctoral. Fielding es uno de los principales autores del Protocolo de transferencia de hipertexto (HTTP) versiones 1.0 y 1.1.

Una API REST es una biblioteca apoyada totalmente en el estándar HTTP. Visto de una forma más sencilla, una API REST es un servicio que nos provee de funciones que nos dan la capacidad de hacer uso de un servicio web que no es nuestro, dentro de una aplicación propia, de manera segura.

El estilo REST hace énfasis en que las interacciones entre los clientes y los servicios se mejoran al tener un número limitado de operaciones (verbos). La flexibilidad se obtiene asignando recursos a sus propios identificadores de recursos universales únicos (URI). Debido a que cada verbo tiene un significado específico (GET, POST, PUT y DELETE), REST evita la ambigüedad. [2]

1. Se usa GET para obtener un recurso
2. Se usa POST para crear un recurso en el servidor
3. Se usa PUT para cambiar el estado de un recurso o actualizarlo
4. Se usa DELETE para eliminar un recurso

Base de datos Mongo

MongoDB(Mongo Data Base) 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(Binary Script Object Notation), que es una representación binaria de JSON (JavaScript Object Notation).

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.[3]

Servicio Web

Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Una posible sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.[4]

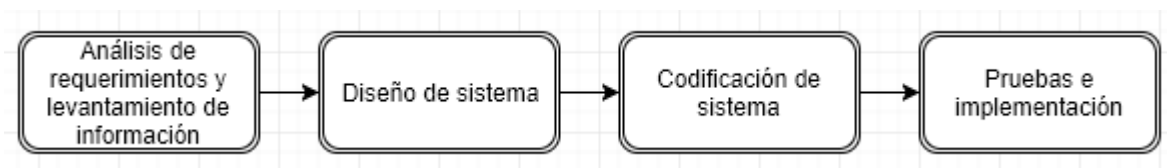
¿Para qué sirven?

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar

interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.
[4]

2.1 Metodología

Como metodología de desarrollo se utilizo una versión de modificación propia de la metodología de desarrollo cascada, debido a esto las fases de desarrollo se verán afectadas, esto podran ser presenciado en el siguiente gráfico.



Análisis de requerimientos y levantamiento de información: En esta fase se procede a determinar cuales son los requerimientos del sistema, para esto es necesario, hacer un levantamiento de información del sistemas

Diseño de sistema: Durante esta fase se debe realizar el diseño del sistema ,esto incluye, el diseño de su arquitectura, así como el diseño del comportamiento y funcionalidades del mismo.

Codificación de sistema: En la Fase de codificación de sistema, se procede a realizar el proceso de desarrollo a nivel de programación.

Pruebas e implementación: En la fase final, se procede a programar las pruebas para conocer si el sistema funciona de manera correcta, y así proceder con su implementación como producto.[5]

3 Diseño del Sistema

Requerimientos

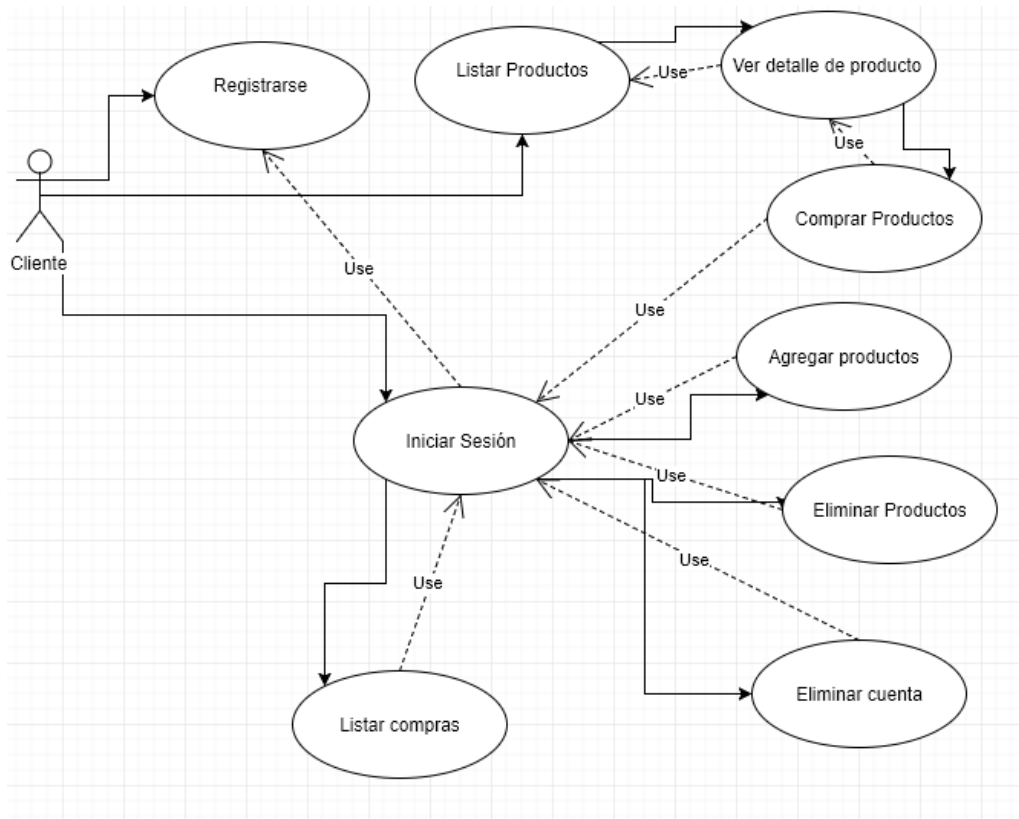
Requerimientos físicos: para la realización del sistema, haciendo énfasis en sus pruebas, es necesario dos laptops, donde una servirá como servidor y cliente al mismo tiempo, y otra será solo para las funciones de cliente, esto con el fin de poder probar que el sistema sea funcional, tanto en un entorno seguro (Laptop Cliente/Servidor), así como su funcionalidad distribuida (Laptop cliente).

Requerimiento de software:

1. Sistema Operativo Linux, para alojar el servidor, el cliente puede funcionar en cualquier sistema operativo, debido a que se trata de una página web
2. Framework NodeJs
3. Brackets IDE y Sublime Text
4. MongoDB, como base de datos.
5. Módulos de externos de NodeJS: Nodamón, Express, EJS y Mongoose

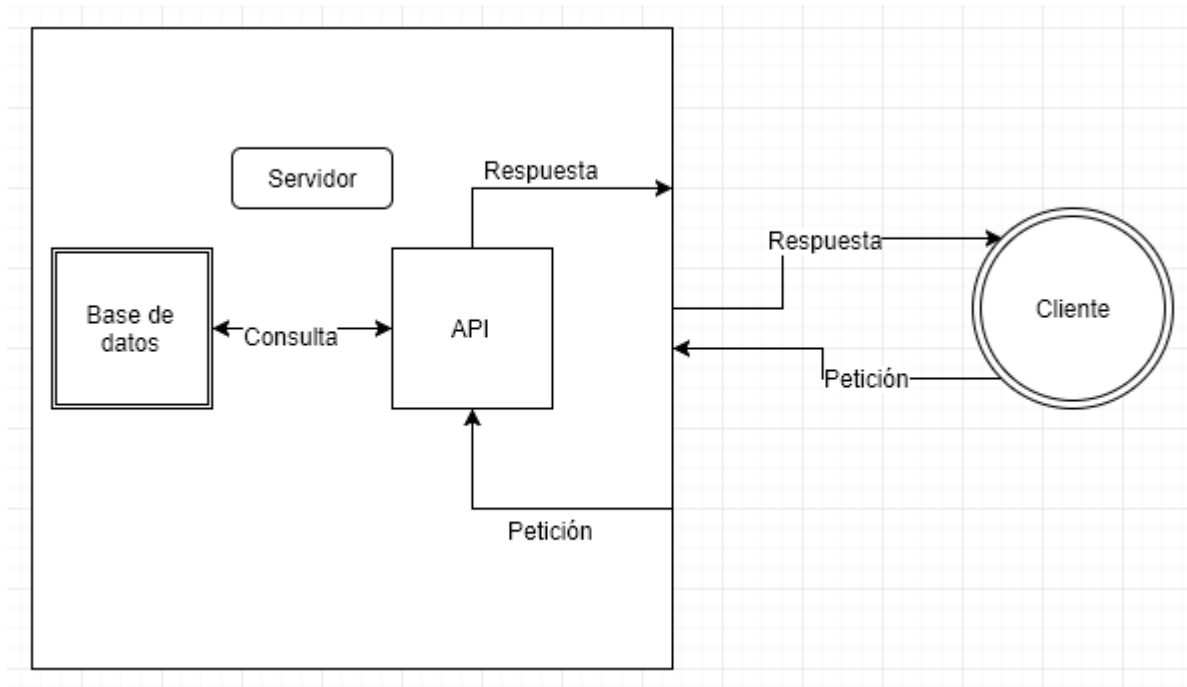
Casos de uso y comportamiento del cliente

El diagrama de caso de usos permitira ver la funcionalidad el sistema, ademas a su vez permitirá ver la funcionalidad del lado del cliente



Si se analiza el diagrama anterior, podemos notar , que el cliente puede realizar varias acciones en el sistema, entre estas esta, la realización de ver un listado de productos, ver detalles específicos de un producto, comprarlo, agregar productos los cuales se pueden vender, ademas de visualizar las compras hechas, esta ultima función sirve para visualizar las compras realizadas, así como las compras realizadas de productos del cliente.

4 Arquitectura del software



Para poder entender la arquitectura del sistema, no se puede usar el UML(Unified Modeling Language) en su totalidad y menos aun en lo que arquitectura respecta, ya que no se trata de un software orientado a objetos, convencional, sino de una pagina web, por lo tanto se decidió incluir la siguiente imagen, con el fin de poder representar la arquitectura del software.

Como Podemos notar en la imagen anterior describe la arquitectura y la forma en que se comporta el sistema. El proceso Consiste en que el cliente realiza una petición al servidor, el API toma esta petición, realiza consultas a la base de datos de ser necesario, y por ultimo API enviá una respuesta al cliente.

Acá podemos notar que la API es de gran importancia para el funcionamiento del sistema, ya que esta es la que se encarga de coordinar todo lo relacionado a recibir, interpretar y responder los mensajes del cliente.

Modelo de fallas

A pesar de los beneficios de la arquitectura REST, el sistema puede llegar a fallar, debido a esto es necesario hacer un modelo de fallas, las principales fallas sera las siguientes:

1. Fallas a nivel de Servidor: El host del servidor puede fallar de forma que afecte al funcionamiento del acceso de la pagina web, la mejor forma para solucionar esto, es tener servidores alternos, o al menos servidores, donde sea posible acceder a los servicios de mayor utilidad por el cliente.
2. Fallas a nivel de base de datos: Debido a que la base de datos es orientada a documentos (MongoDB) los problemas a nivel de limite de espacio de almacenamiento, son bastante menores, pero al igual que las base de datos relacionales, esta puede sufrir fallas donde la base de datos esta caída, produciendo así que sea posible acceder a la pagina web pero las operaciones no podrán ser realizadas. Para solucionar esto, la mejor opción es tener base de datos replicadas.

5 Conclusiones

Es importante entender que, los la web 2.0 y los sistemas distribuidos van de la mano, esto es fácil notar si analizamos conceptos, como los servicios web usando las API REST en un servidor, estos son el núcleo de AllMarket.net , la forma de funcionamiento de la pagina web , en lo que arquitectura respecta, muestra que la arquitectura REST, permite solventar en gran medida, la problemática respecto al pase de mensaje, entre servidor y cliente, ya que es una arquitectura bastante simple, tanto de entender, como implementar, para finalizar es importante hacer notar que las tecnologías REST y los servicios web son una alternativa excelente al momento de desarrollar una pagina web dinámica, se ve un futuro prometedor para el mundo de desarrollo web.

6 Referencias

- [1] Andrearrs, ¿Qué es una API? , 2014, Extraído de: <https://hipertextual.com/archivo/2014/05/que-es-api/>
- [2] Andrearrs, ¿Qué es una API? , 2014, Extraído de: <https://hipertextual.com/archivo/2014/05/que-es-api/>
- [3] Rubenfa , MongoDB: qué es, cómo funciona y cuándo podemos usarlo (o no), 2014, Extraído de: <https://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>
- [4] World Wide Web Consortium, Guía Breve de Servicios Web, Extraído de: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>
- [5] Roger S. Pressman, Ingeniería del software: Un enfoque práctico, 7ma ed., McGrawHill, 2010, pp.33–35.