**PETRY**

ANDERSON LOPEZ

INSTITUCION UNIVERSITARIA POLITECNICO GRANCOLOMBIANO

FACULDAD INGENIERIA Y CIENCIA BASICAS

INGENIERIA DE SOFTWARE II

2018

**Prologo**

1. **Introducción……………………………………………………………………………………………………………..1**
2. Objetivo SMART……………………………………………………………………………………………………….…..1
3. Marco Teórico………………………………………………………………………………………………………….…..2
   1. **Programación Web……………………………………………………………………………………………………2**
   2. **Conceptos básicos de HTML………………………………………………………………………………………2**
   3. **Lenguajes de script……………………………………………………………………………………………………2**
   4. **Modelo de Objetos de Documento…………………………………………………………………………..3**
   5. **Patrón de arquitectura MVC……………………………………………………………………………………..3**
   6. **JavaServer Faces……………………………………………………………………………………………………….3**
   7. **Framework…………………………………………………………………………………………………………….….4**
   8. **Tipos de Framework………………………………………………………………………………………………….4**
   9. **Fronten y Backend…………………………………………………………………………………………………….4**
   10. **Base de Datos……………………………………………………………………………………………………..5**
   11. **SQL Server…………………………………………………………………………………………………….…….5**
   12. **Arquitectura……………………………………………………………………………………………….……….6**
   13. Blockchain y Bitcoin………………………………………………………………………………………………..6
4. Modelado del negocio (Canvas)……………………………………………………………………………..…….6
5. Problema de investigación………………………………………………………………………………………..….7
6. Justificación………………………………………………………………………………………………………………….8
7. Referencias……………………………………………………………………………………………………………..……9
8. **Introducción**

En el presente documento se aborda el desarrollo e implementación de una aplicación Web para la adopción de mascotas aplicando técnicas de Ingeniera de software, los componentes y tecnologías existentes para desarrollo web, las características y alcance del proyecto, en el cual se referencia las tecnologías que se usaran y sus conceptos básicos para tener una idea general de cuales elementos usaremos para el desarrollo del producto final.

1. **Objetivo SMART**

Específico: **Desarrollar una aplicación, basada en tecnologías tradicionales para servicios web, con el fin de registrar el historial, el seguimiento del usuario que adopta una mascota, y calificar del proceso de adopción de mascotas usando tecnologías Blockchain.**

Medible: el objetivo es medible, dado que se verificará que el aplicativo cumpla con las funciones para las cuales es diseñado (es decir, registrar, hacer seguimiento y calificar el proceso de adopción, además que sea una aplicación web), con el frontend en HTML y CSS. El Backend en PHP, y una base SQL para almacenar todos los datos de los usuarios, mascotas y registros asociados, Blockchain solo se usará para puntuar a los usuarios que adopten mascotas, con el propósito que en el futuro les permita adoptar o no adoptar otras mascotas.

Alcanzable: el objetivo es alcanzable, teniendo en cuenta que se implementará una aplicación web usando un Stack Cliente Servidor tradicional para la mayoría de casos de uso, solo un caso de uso se implementara en Blockchain, para esta última parte solo se desarrollara un nodo con Blockchain con las mínimas características requeridas de un sistema descentralizado que genera una cadena de bloques, usando el algoritmo diseñado y publicado por Satoshi Nakamoto, a quien se le atribuye la creación de esta tecnología. El nodo solo almacenará y gestionará la puntuación (Score: calificación del cuidado de la mascota. Bueno o Malo) de los usuarios que adopten mascotas en las Fundaciones y Refugios Autorizados.

Realista: este objetivo es realista ya que es viable la aplicación porque la implementación está basada en las tecnologías tradicionales de servicios web. Lo más complejo es el desarrollo del nodo de la red descentralizada implementando Blockchain, lo que supone el gran reto de este proyecto.

Encuadrado en el Tiempo: por ser un proyecto entregable en el trascurso de un semestre, solo se usarán 4 meses, para todo el proceso de estudio de las tecnologías requeridas, implementación y pruebas de uso. Fecha en la que se realizará el cierre del proyecto: 30 de Mayo de 2018.

1. **Marco Teórico**

**3.1 Programación Web**

Según el autor Mora Sergio, 2002 (1). Una aplicación Web es un mecanismo en el cual se realiza una solicitud a un navegador el cual realiza peticiones a un servidor a través de internet.

Existen muchas tecnologías para desarrollar aplicaciones web, entre las más conocidas son basadas en HTML, JavaScript.

En el autor describe el concepto cliente servidor, como una relación entre dos procesos que solicitan servicios, el cliente y procesos que responden a estas solicitudes, el servidor. La arquitectura Cliente/Servidor permite la creación de aplicaciones distribuidas las cuales separan las funciones de servicio, permitiendo la escalabilidad de la aplicación.

El autor nos comenta que la arquitectura Cliente/Servidor separa entres funciones o niveles, las cuales enumera como Lógica de presentación, Lógica de negocio (o aplicación) y Lógica de datos.

La descripción de un sistema Cliente/Servidor no la plantea el autor como la unión de partes visibles para el cliente denominado Frontend o interfaz de usuario. La parte del Servidor o Backend que administra con los recursos, las cuales tienen diferentes necesidades de capacidad computacional y almacenamiento físico.

En este sistema el cliente se comunica con el servidor a través de una interfaz denominada API y un procedimiento de llamada remoto RPC para acceder a bases de datos. La comunicación entre el Cliente y Servidor se realiza usando el protocolo HTTP el cual hace parte de la familia de protocolos de comunicación TCP/IP empleados por Internet.

**3.2 Conceptos básicos de HTML**

el autor Mora Sergio, 2002 (1). Describe y nos explica los conceptos básicos de HTML en el cual describe como un grupo de etiquetas la cuales delimitan las partes del documento HTML se va a afectar por su acción, las cuales posen atributos los cuales se les asigna un valor, algunos obligatorios.

Cuando un Usuario solicita una página HTML a un servidor Web, le es enviada la página tal cual, el explorador interpreta las etiquetas que contiene la página mostrándola al usuario Final.

**3.3 Lenguajes de script**

Según el Autor Mora Sergio, 2002 [1], los leguajes de Script o Guiones permiten incluir programación en las páginas web, es similar a macros o ficheros por lotes, línea de comandos que se pueden ejecutar con o sin la interacción de usuario. Suele emplearse como contexto y no permite crear aplicaciones individuales con ficheros ejecutables independientes, suelen ser interpretados.

Existe una variedad de leguajes Script que se pueden emplear en páginas web, tales como, JavaScript, VBScript, etc. El más empleado es el primero.

JavaScript deriva de la familia de leguajes formada por C y C++ y Java, cabe aclarar que JavaScript y Java son totalmente diferentes desarrollados por distintas compañías, solo comparten la sintaxis y el nombre, también se puede usar en el Servidor Web

Según el Autor, en aplicaciones de Script en páginas HTML sirve para validar los datos en el cliente y consistencia de los valores antes de mandar un formulario al servidor web, también actualizar campos relacionados en formularios, realizar procesamientos que no requieran la utilización de información centralizada, como servir de base para la utilización de otras tecnologías tales como (DHTML, Java, ActiveX entre otras

Estos leguajes también pueden actuar sobre el navegador a través de objetos integrados que representan el documento, la ventana activa y controles del formulario usando el modelo DOM modelo objeto del documento, también tiene restricciones que no permiten tener acceso a los recursos de la maquina en la que se ejecutan.

**3.4 Modelo de Objetos de Documento**

Según el Autor Mora Sergio, 2002 [1], DOM es una interfaz que permite acceder y modificar la estructura y contenido de una página web, para acceder a enlaces, formularios y como modificarlos, permite acceder a los elementos HTML desde leguajes Script, para esto se crean objetos que los representan, y guardan entre ellos una relación de parentesco que representa la estructura lógica de una página web HTML, la página se representa en una ventana con un documento que contiene formularios, botones, casillas se texto.

**3.5 Patrón de arquitectura MVC**

Según el Autor Gaguana Robinson, 2015 [2], MVC es una arquitectura para el desarrollo de aplicaciones web el cual separa el código del proyecto en tres capas, conocidas como modelo, vista y controlador, patrón muy utilizado en el desarrollo web por medio del Frameword MVC para crear aplicaciones de mayor calidad.

3.6 **JavaServer Faces**

Según el Autor Gaguana Robinson, 2015 [2], la tecnología que ha permitido el desarrollo de aplicaciones web es Java el cual permite la utilización de Servlets junto con JSP para la creación de interfaces cliente, los Servlets han sido de ayuda pero no son suficientes para nuevos requerimientos y tecnologías; por esta razón JavaServer Faces es una tecnología o framework que permite simplificar el desarrollo de las interfaces web de usuario, por otro lado JFS permite que el Backend sea desarrollado sin tener en cuenta los detalles HTML, permite la encapsulación de toda la lógica de interacciones del usuario permitiendo que la interfaz no contenga lógica del programa, como tal JavaServer Faces es un conjunto de componentes UI con una API para la representación de los componentes de la interfaz, también se caracteriza por tener un modelo de eventos en el servidor, permitir la administración de estado.

**3.7 Framework**

Según el Autor Gaguana Robinson, 2015 [2], antes que todo clasifica los programadores en dos tipos, los encargados de escribir código que permita el desarrollo de lenguajes Java, C, PHP entre otros, y los programadores que usan estos leguajes, los cuales se centran en los proyectos sin preocuparse por las mecánicas y técnicas de bajo nivel, los cuales usan librerías, bibliotecas o herramientas que son llamados frameworks, estos últimos son desarrollados por los programadores del sistema.

Los frameworks permiten reutilizar código ahorrando tiempo al desarrollar aplicaciones de manera eficaz en los cuales se adapten a la necesidad sin invertir la rueda de nuevo y todos estos frameworks web son basados en el patrón MVC.

**3.8 Tipos de Framework**

Según el Autor Gaguana Robinson, 2015 [2], existen muchos tipos de Framework, enfocados en el control de eventos, a la publicación de elementos y orientados a la interfaz de usuario como JavaServer Faces.

Los cuales son basados en el patrón MVC, permitiendo una capa de control para permitir la integración con otras herramientas de forma fácil.

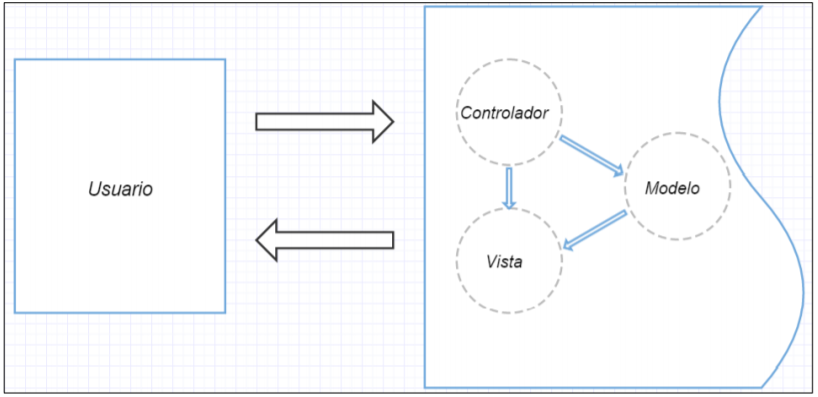


Figura:Tipos de frameworks

Realizado por: Robinson Paguay

**3.9 Fronten y Backend**

Según el autor Diaz Ivan, 2014 [3], el desarrollo web tiene las tecnologías se divide en dos, Backend y Frontend.

El Frontend, es la parte visual que interactúa con el usuario final, la cual se divide en tres partes generalmente, HTML, CSS y JavaScript, lo cual permite separar la interfaz gráfica, del control de la información y subrutinas que se deben ejecutar, para mejorar la contribución y diseño de la página web.

El Backend, esta implementado del lado del servidor, contiene la parte lógica, encargado de interactuar con el servidor, manejo de sesiones, montar la página en el servidor, el cual esta implementado generalmente con PHP¨, Python.Net, Java, entre otros, el programador generalmente debe conocer sobre Bases de datos, Frameworks y Librerías, es el encardo de enviar las peticiones del Frontend y comunicarlas con el servidor.

**3.10 Base de Datos**

Según el autor Rivera Fray, 2008 [4], un sistema de manejo de bases de datos es un conjunto de elementos interrelacionados con programas que permiten tener acceso a los archivos para editarlos, los cuales tiene una visión abstracta de la información, que permite crear estructuras complejas de los datos.

Hay tres niveles de abstracción, físico, conceptual y visión. Un modelo Entidad Relación el cual es la abstracción del mundo real (Entidad) y como se relacionan entre si (Relación), se usa para determinar la lógica de la base de datos.

Los Usuarios solicitan información a la Base de Datos a través de un lenguaje de consulta formal, los cuales pueden ser procedimental o no procedimental, en el procedimental el usuario especifica las operaciones necesarias para obtener un resultado, en la no procedimental el Usuario describe la información que necesita sin indicar procedimiento alguno.

**3.11 SQL Server**

Según el autor Rivera Fray, 2008 [4], SQL Server es una plataforma para base de datos, se utiliza en procesamiento en línea a gran escala, en las bodegas de datos, aplicaciones de comercio electrónico, plataformas de inteligencia de negocios para soluciones de integración, análisis y creación de informes.

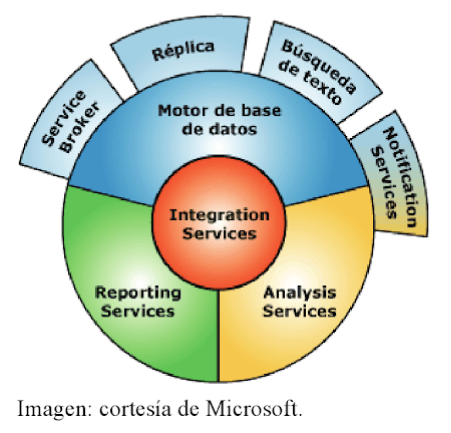


Figura Integración de Servicios, realizado por Microsoft

**3.12 Arquitectura**

Según el autor Rivera Fray, 2008 [4], la base SQL Server está dividida en varios elementos lógicos, tablas, vistas entre otros, los cuales son trasparentes para el Usuario, también puede contener varias bases de datos de diferentes clientes, los cuales tiene permisos, pudiendo quedar separadas de los usuarios que no tiene autoridad para ver dato, sean limitados solo a Usuarios específicos.

La creación de la base de datos asigna características físicas a la base de datos, como tamaño, tasa de crecimiento, identificación del propietario y el grupo de servidores al que pertenece, la base de datos está compuesta por el Archivo Primario, Segundario y Log de Transacciones.

**3.13 Blockchain y Bitcoin**

Según el Autor Satoshi Nakamoto,2009. [5], Bitcoin es un caso de uso de blockchain usado para eliminar el problema del doble pago a través de una versión electrónica de dinero, el cual no dependería de una institución financiera, las transacciones son de usuario a usuario, no dependería de un tercero de confianza, el cual funciona bien para la mayoría de los casos.

Nakamoto nos comenta que una versión puramente electrónica de efectivo permitiría no depender de terceros de confianza, evitaría el doble gasto, con firmas digitales como parte de la solución, con transacción de usuario a usuario usando marca de tiempo que se introducen en una cadena continua de pruebas de trabajo basadas en cálculo de hashes, formando un registro que no pueden ser cambiado sin volver a recrear la prueba de trabajo completa, con lo cual la cadena más larga sirve como testigo y prueba de secuencia de eventos, también asegura que esta vino desde la agrupación con procesamiento CPU que está bajo el control de nodos, la red en si misma requiere de una estructura mínima, los mensajes son enviados bajo la premisa de menor esfuerzo, los nodos se pueden conectar y desconectar en cualquier momento aceptando la cadena más larga de prueba de trabajo.

1. **Modelado del negocio (Canvas)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Socios clave**  Empresas de producción de alimento y accesorios para animales.  Tiendas de mascotas y veterinarias. | **Actividades clave**  Mantener la aplicación funcional. | **Propuesta de valor**  Permite nuestro aplicativo conectar amantes de los animales y el poder usar nuestro aplicativo como medio para adoptar o dar en adopción a las mascotas.  Encontrar hogares a los animales sin hogar. | | **Relaciones con clientes**  Asistencia en redes sociales, asistencia personal con clientes de venta (comida y accesorios). | **Segmentos de cliente**  Personas interesadas en el cuidado animal de entre 20 a 60 años. |
| **Recursos clave**  Persona de desarrollo.  Persona encargada de asociación comercial. | **Canales**  Redes sociales  Posters en centros de animales o veterinarias. |
| **Estructura de costos**  Personal.  Diseño.  Publicidad. | | | **Fuentes de ingreso**  Publicidad.  Venta de alimento y accesorios para mascotas.  Comisiones de recomendación en tiendas de animales y veterinarias. | | |

1. **Problema de investigación.**

Las personan abandonan en la calla sus mascotas sin un responsable directo que responda por esta crueldad animal.

No hay control Único en el sistema de adopción de mascotas, cada fundación o refugio de mascotas usas sus propios sistemas de información de los usuarios que adoptan una mascota, cada persona que adopte mascotas y no cumpla con sus debidas obligaciones de cuidado y obligaciones mínimos, podrías quedar libre de buscar a un tercero para que le dé una mascota sin ningún problema, pero si todos los que donen una mascota usan una aplicación para firmar digitalmente un acuerdo de adopción las dos partes que interviene en el proceso, se podría asegurar el bienestar de animal en adopción y un control único.

Las faltas de cultura ciudadana, control y seguimiento a los dueños de mascotas permite la impunidad agravando el problema de abandono de mascotas.

Con la creación de una única fuente de información que integre todos los procesos de adopción en la cual aparece, quien, cuando y que mascota fue adoptada, con un seguimiento en el tiempo, de la adopción y cuidado del animal el cual genera una calificación, todo gestionado desde una aplicación web e integrando blockchain para llevar un registro único nacional de la calidad de la adopción, el cual sería descentralizado e inalterable.

¿Es posible implementar e integrar tecnología Blockchein en una aplicación básica Cliente Servidor, partiendo de cero conocimientos técnicos del funcionamiento de la tecnología e implementación, en un tiempo limitado?

1. **Justificación.**

Hacer de la adopción de mascotas un mecanismo controlado desde una única fuente de control y verificación de información el cual está siendo calificado al usuario final de su responsabilidad directa con las mascotas que adquiere y se compromete con su cuidado y trato humanitario.

Con el desarrollo de una aplicación web que capture todo el proceso y quede registrados en una base de datos gestionada por las entidades que interviene en el proceso de adopción.

Integrando tecnología Blockchain la cual permite tener un control público de la información de la calificación de los usuarios del sistema, información que no puede ser alterada ni falsificar, la cual esta descentralizada y se puede verificar desde una única fuente en todo el territorio nacional.

**7. Referencias**

[1] Mora Sergio, Programación de aplicaciones web: historia, principios básicos y clientes web, 2002. <https://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf>

[2] Gaguana Robinson, “DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE LA DOCUMENTACIÓN PÚBLICA EN EL GOBIERNO AUTÓNOMO DESCENTRALIZADO MUNICIPAL DE RIOBAMBA, 2015. <http://dspace.espoch.edu.ec/bitstream/123456789/4337/1/18T00599.pdf>

[3] Diaz Ivan, 2014, <https://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>

[4] Rivera Fray, 2008, Base de Datos Relacional, Teoría y Practica. <https://books.google.es/books?hl=es&lr=&id=buM5rlZME-cC&oi=fnd&pg=PA9&dq=base+de+datos+relacional&ots=6L3CssJBJH&sig=xQmhqU2CGYo67kza1pmMBw0Z2jQ#v=onepage&q&f=false>

[5] Satoshi Nakamoto,2009. <https://bitcoin.org/bitcoin.pdf> <https://blog.bit2me.com/es/paper-original-bitcoin-en-espanol/>