# TITULO

## Titulo del proyecto de intervencion

DESARROLLO DE UNA REINGENIERIA DE UN SITIO WEB CORPORATIVO

PARA EL DEPARTAMENTO DE MARKETING DEL HOSPITAL PRIVADO SANTA FE

## Autor

## Resumen

## Resumen, palabras clave

## Indice numerado

TITULO 1

Titulo del proyecto de intervencion 1

Autor 1

Resumen 1

Resumen, palabras clave 1

Indice numerado 2

1. INTRODUCCION 4

1.1 PRESENTACION 4

1.2 ANALISIS SITUACIONAL 4

A. UBICACIÓN ORGANIZACIONAL 4

1.2.1 EMPRESA 4

1.2.2 UBICACIÓN GEOGRAFICA 4

ILUSTRACION 1. UBICACIÓN GEOGRAFICA DE LA EMPRESA. 5

1.2.3 MISION 5

1.2.3.1 MISION DEL HOSPITAL PRIVADO SANTA FE 5

1.2.4. VISION DEL HOSPITAL PRIVADO SANTA FE 5

1.2.4.1 VISION DEL HOSPITAL PRIVADO SANTA FE 5

1.2.5 ORGANIGRAMAS 5

1.2.5.1 ORGANIGRAMA DE LA INSTITUCION 5

B. DESCRIPCION FUNCIONAL Y/O OPERATIVA 5

1.2.6. DESCRIPCION FUNCIONAL 5

1.3. DESCRIPCION GENERAL DEL PROBLEMA. 7

1.4. DEFINICION DEL PROBLEMA SELECCIONADO. 8

2. BASES TEORICAS. 9

2.1. MARCO HISTORICO Y CONTEXTUAL 9

2.2 MARCO REFERENCIAL 11

2.3 MARCO TEORICO 13

I. COMO FUNCIONAN LAS APLICACIONES DE INTERNET 13

1. Aplicaciones web 13

2. Historia de las Aplicaciones web 14

3. Consideraciones técnicas 16

4. Estructura de las Aplicaciones web 16

5. Ventajas 17

6. Servidor web 18

7. Lenguaje de marcado de Hipertexto 20

II. COMO SE DESARROLLA UNA APLICACIÓN WEB 21

1. ¿Qué es el diseño y desarrollo de las aplicaciones web? 21

2. Framework 21

8. Content management system 20

9. Drupal 22

10. Desarrollo ágil de software 22

III. ENTORNO DE TRABAJO 24

1. Sistema operativo: Lubuntu 24

2. Editor de texto: SublimeText2 24

3. Sistema controlador de versiones: GIT 24

4. Computación en la nube: Heroku 26

2.6 TERMINO-DEFINICION 27

3. PROYECTO DE INTERVENCION 29

3.1 ENUNCIADO DEL OBJETIVO 29

3.2 OBJETIVO GENERAL 29

3.3 OBJETIVOS ESPECIFICOS 29

3.4 ANALISIS DE LOS FACTORES RESTRICTIVOS 29

3.5 JUSTIFICACION 30

4. ESTRATEGIAS DE IMPLEMENTACION 30

4.1. ESTRATEGIAS PARA PRESENTACION DEL PROYECTO 30

4.2. ANALISIS INVOLUCRADOS 31

4.3. DIAGRAMA UML 31

4.4. REALIZACIONN DE LOS PASOS DEL PROYECTO 31

5. ADMINISTRACION DEL PROYECTO 31

5.1. PLAN DE ACCION 31

5.2. DIAGRAMA DE GANTT 32

5.3. RECURSOS 32

5.4. PRESUPUESTO. 33

5.4.1. COSTO DEL PROYECTO POR TAREA 34

6. CONCLUSIONES Y RECOMENDACIONES 36

6.1 Presentación de los resultados 36

6.2 Conclusiones y las recomendaciones 36

Referencias Documentales 37

Referencias Documentales 37

Bibliografia complementaria 37

ANEXOS 38

ESTRUCTURA DE BASE DE DATOS 38

REQUERIMIENTOS DEL USUARIO 38

PANTALLAS DEL SISTEMA 38

# 1. INTRODUCCION

## 1.1 PRESENTACION

El presente trabajo describe el análisis de la fiabilidad, planeación y desarrollo de un proyecto de intervencion para el desarrollo del sitio web del hospital privado Santa Fe.

Se describen los conceptos fundamentales de una aplicación web, y las tecnologías que lo componen en dos partes: cliente y servidor. Se relata el estándar mundial HTML5 cuya característica más impresionante es la escalabilidad que proporciona al acceso del lado del cliente.

Se detallan los patrones de reingeniera e ingeniera inversa que son indispensables en la aplicación del análisis a los procesos en los que se desea intervenir para su mejora.

Además se describe el diseño de la solución, la seleccion de un framework y sistema manejador de contenidos de licencia libre, para un desarrollo convencional, configurable, adaptable y sostenible a las necesidades del cliente.

Otros estándares que seguí en el desarrollo de esta aplicación web pertenecen a la metodología de desarrollo ágil mismo que se explica en este documento.

## 1.2 ANALISIS SITUACIONAL

La empresa Hospital privado Santa Fe promociona su empresa mediante su sitio web. Históricamente este sitio web fue desarrollado por una empresa externa en el año 2012, y no se le han hecho cambios desde su entrega. Actualmente el único proceso que mantiene esta empresa con su sitio es el de pagar anualmente la renta de su hospedaje en un servidor compartido.

Sin embargo este sitio ha quedado desactualizado por un espacio de dos años lo que proporciona perdidas a la empresa. La empresa Hospital privado santa fe pierde clientes potenciales en el ciberespacio si descuida el proceso de mantener su página web.

Esto es causado por que la infraestructura y políticas de la empresa no cuentan con un departamento de TI en su infraestructura que permita un desarrollo responsable y pertinente. Para sus procesos con tecnologías de la información deben solicitar a empresas externas que realicen todo tipo de trabajo informático.

### A. UBICACIÓN ORGANIZACIONAL

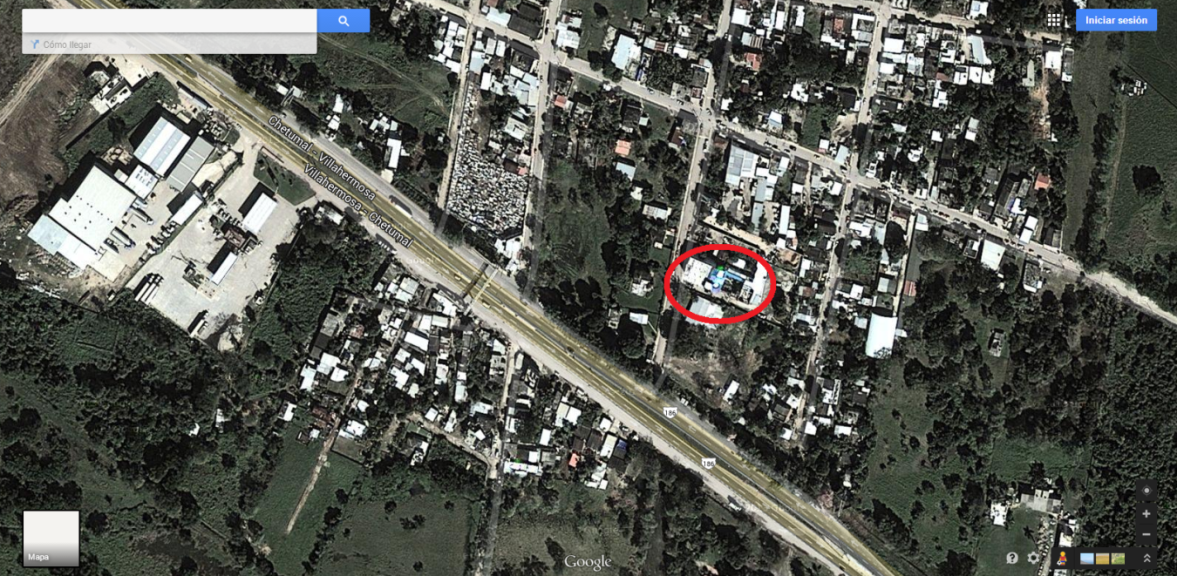
### 1.2.1 EMPRESA

El nombre de la empresa donde se implementara el proyecto será el Hospital Privado Santa Fe S.A. DE C.V.

### 1.2.2 UBICACIÓN GEOGRAFICA

El Hospital Privado Santa Fe S.A. de C.V. está ubicado en Calle Juárez #106, Poblado Dos Montes, Centro, Tabasco.

### ILUSTRACION 1. UBICACIÓN GEOGRAFICA DE LA EMPRESA.

****

**FUENTE: GOOGLE MAPS.**

El hospital privado Santa fe se encuentra ubicada en las afueras de Villahermosa. Se sitúa a pocos kilómetros de la carretera federal que va de Villahermosa hacia Macuspana. Muestra atención por imprimir calidez humana hacia sus clientes, intervenciones médicas, instalaciones espaciales y arquitectónicas, la administración empresarial y la distribución del trabajo. Desde 2012 hasta el día de hoy mantienen su sitio web: hospitalprivadosantafe.com.mx.

### 1.2.3 MISION

### 1.2.3.1 MISION DEL HOSPITAL PRIVADO SANTA FE

Brindar salud mediante la prestación de servicios de excelencia en consulta, cirugía y hospitalización.

### 1.2.4. VISION DEL HOSPITAL PRIVADO SANTA FE

### 1.2.4.1 VISION DEL HOSPITAL PRIVADO SANTA FE

Para el 2018 ser la referencia médica y hospitalaria en Tabasco.

### 1.2.5 ORGANIGRAMAS

### 1.2.5.1 ORGANIGRAMA DE LA INSTITUCION

**DIAGRAMA 1. ORGANIGRAMA GENERAL DEL HOSPITAL PRIVADO SANTA FE.**

### B. DESCRIPCION FUNCIONAL Y/O OPERATIVA

### 1.2.6. DESCRIPCION FUNCIONAL

El hospital Santa Fe es una empresa Mexicana especializada en salud y procesos hospitalarios. Entre las funciones principales del Hospital Santa fe se encarga de:

* Realizar cirugías de apendicetomía, colecistectomía, cesáreas, hernias, partos, RTUP, histerectomía, hemorroides.
* Consultas medicas y cirugías de las siguientes especialidades: cirugía general, ginecología, coloproctología, pediatría, oncología, neurocirugía, traumatología, urología, otorrinolaringología, angiología, cirugía plástica, medicina interna, otras.
* Consulta de medicina general las 24 horas del día.
* Estudios especiales: endoscopía, videorectosigmoidoscopía, colonoscopía.
* Rayos X.
* Farmacia.
* Laboratorio.
* Ultrasonido.
* Urgencias.

**ILUSTRACCION 2. UBICACIÓN DE LA DIRECCION ADMINISTRATIVA DEL HOSPITAL PRIVADO SANTA FE.**

El proceso en el que desea intervenir es en el mantenimiento de la página web del hospital privado santa fe.

“Si quieren seguir siendo competitivos, los productores de bienes y servicios tendrán que desarrollar nuevas oportunidades de canal, como internet y otros canales directos. No obstante, el desarrollo de estos nuevos canales a menudo enfrenta a la compañía en competencia directa contra sus canales establecidos, y origina conflictos. “(Pagina 309. Fundamentos de marketing, Kotler.)

En este contexto surgen los límites de un espacio para el proyecto de intervención a través de un proyecto de reingeniería para sus procesos de marketing digital.

Este proyecto es utilizado por:

* Departamento administrativo
* Clientes habituales
* Clientes potenciales

Teniendo un promedio de 13 visitas al mes por tales. Estas estadísticas se han calculado conforme a las graficas generadas por el panel de control proporcionado por el hosting hospitalprivadosantafe.com.mx

El departamento administrativo es un pilar de esta empresa ya que tiene una multiplicidad de funciones y responsabilidades

Entre las funciones que desempeña se encuentra:

* Administrar los recursos, y procesos financieros de la empresa.
* Contratar nuevos empleados y capacitarlos
* Presupuestar servicios especiales a los clientes.
* Promocionar a la empresa mediante diferentes canales y procesos.

**ILUSTRACION 4. SITIO WEB CORPORATIVO DEL HOSPITAL PRIVADO SANTA FE.**

**FUENTE: SERVIDOR HTTP.**

A la página web se accede por medio de la url http://www.hospitalprivadosantafe.com.mx.

Si el usuario hace clic sobre los menús será redirigido a otras páginas, si analiza la pagina principal vera una noticia sobre quirófanos nuevos en el hospital y la foto de estos nuevos quirófanos, sin embargo al dirigirse a la noticia mediante su enlace se dará cuenta que la noticia solo tiene el titulo y no tiene un cuerpo de texto que le de mas información acerca de estos nuevos quirófanos.

A la izquierda de la pagina vemos un menú con 4 links: Inicio, Noticias, Galeria de fotos, Directorio Médico y Contacto. El primer menú nos redirige a la pagina principal, el segundo a la sección de noticias que comentamos anteriormente, el tercer a nueva pagina, Al acceder a la galería de fotos el sistema publica una galería de fotos, aunque es de notar que tarda demasiado (de 3 a 5 minutos) en cargar la galería y eso puede impacientar a un usuario.

Una vez hemos vista la galería de imágenes del hospital, al acceder al link de “Directorio Medico” el servidor mediante el lenguaje PHP nos muestra un filtro del personal que trabaja en el hospital, el sistema los filtra en: personal administrativo, doctores generales, cirujanos, y especialistas. Si le damos clic a algunos de estos filtros la vista de la pagina cambiara para mostrarnos el nombre del tipo de personal que hayamos elegido.

Finalmente, si el usuario entra al link de contacto, el sistema despliega un formulario de contacto pidiendo nuestro nombre, correo, y mensaje, una vez que hayamos acabado apretamos el botón de enviar y el sistema validara los campos que hayamos escrito, si la validación no pasa nos pedirá que corrijamos los datos, de otra forma enviara nuestra información para contacto y recargara la pagina informándonos que el proceso ha concluido correctamente.

## 1.3. DESCRIPCION GENERAL DEL PROBLEMA.

Hospitalprivadosantafe.com.mx es una página que se encarga de abrir un canal de información mediante la cual promociona, informa y organiza diferentes intereses de los rubros del Hospital Privado Santa Fe.

Actualmente en el encabezado identificamos la imagen corporativa, y descubrimos que el tema de la página es de salud por la foto superior de unos cirujanos trabajando en un paciente. Cuenta con un menú de navegación que está orientado verticalmente del lado izquierdo de la página. Al ir explorando los diferentes enlaces que nos ofrece el menú vamos a ir descubriendo diferentes segmentos de la página web. Una sección distingue la especialización del Hospital en cirugías, y en otro apartado nos ofrece paquetes de estudios especiales e intervenciones médicas. También hay una galería de fotos que tarda de tres a cinco minutos en empezar a mostrar las fotos. Para finalizar encontramos: dirección del Hospital, teléfonos, y un formulario de contacto.

El problema es que la página web ha quedado muy atrás de estar a la par con los planes de negocios del Hospital.

Una de los factores a tomar en cuenta es que su desarrollo se ha vuelto nulo. Por ejemplo; hay secciones de la página no tienen información para clientes, y en su lugar solo tienen datos de prueba. También hay enlaces vacíos dentro del sitio.

¿Pero cómo se generó este problema a esta página? Para verificar los detalles tendremos que analizar detalles tecnológicos que corresponden mas al administrador del sitio que al cliente.

El sitio está hospedado en un host remoto de renta a mipaginaya.com. Mipaginaya.com nos ofrece un dominio junto a otras características muy importantes para la gestión de una página de internet. También nos ofrece un panel de control con el que podemos configurar la mayoría de estas características, estas claves del panel de control, y el acceso al servidor remoto te las envía a tu correo tras hacer el contrato con el hosting.

Entramos al panel de control y encontramos que el documento desde el cual nos redirige al entrar a la página web, se llama index.php desde el cual iniciaremos el análisis.

El archivo index.php tiene embebida la lógica de negocios en el documento con llamadas a la base de datos escritas en PHP. Y la Vista con una mezcla de tecnologías: HTML, CSS, FLASH y JAVASCRIPT. Ahí encontramos el problema, el código de los documentos web está muy acoplado, muy unido, muy frágil en la comparación de que un cambio en un requisito puede romper por completo la funcionalidad de otro.

En conclusión, veo una brecha bastante grande con el sistema que se tiene actualmente y los planes de negocio. En la que le sale más barato al Negocio parar el desarrollo que obtener beneficios del mantenimiento del actual sistema de información.

La falta de desarrollo del sitio web genera insatisfacción a la empresa propietaria. El sitio sigue funcionando pero, quienes gestionan y administran el sitio poco saben del producto que les han dejado. Surge entonces el problema de que producto entregar a una empresa que no tiene manera de gestionar y administrar su sitio web actual.

No se tiene un control de los datos que aparecen en el sitio, no se sabe como agregar, modificar, eliminar u ocultar estos datos.

Se tiene una dependencia muy grande con el desarrollador del sitio web.

## 1.4. DEFINICION DEL PROBLEMA SELECCIONADO.

“No se cuenta con un sitio web que se adapte a las decisiones e inteligencia de negocio del Hospital privado Santa fe”.

El problema central del procedimiento de inteligencia de negocios es causado por no poder poseer un sistema que procese los datos en información.

Aunque existe un tipo de análisis y requerimientos que pretende satisfacer al propósito de la página web este ha quedado obsoleto. Además para ajustarse a cualquier cambio este no toma en cuenta al Negocio, puesto que para modificar cualquier parte del sistema se necesita demasiado conocimiento técnico y se describe a continuación (Conocimiento de diseño de base de datos, PHP, en fin de áreas de tecnologías de la información y que se apartan de los objetivos principales del negocio, y de su forma, evolución y cambio).

1. Analizar y modelar el nuevo cambio.

2. Programar el cambio guiándose del Modelo hecho anteriormente (si el Modelo presenta alguna dificultad que no se previo en el modelado, volver al paso uno).

3. Hacer pruebas al cambio hecho (si las pruebas no pasan volver al paso dos).

4. Entregar el cambio

El sistema actual posee una base de datos, siendo el sistema de gestión de base de datos Mysql el único capaz de generar información para el desarrollo y evaluación. Pero una vez más una consulta directa requiere un conocimiento alto de este programa. Difícil para la integración a la toma de decisiones rápidas que tanto caracteriza el flujo de la inteligencia de negocios.

La página web nueva que se desea implementar tiene el objetivo de incluir la inteligencia de negocios con un sistema que reemplazara al primero.

El tipo de sistema que se ajusta a lo anterior es el denominado “Sistema manejador de contenidos”.

El sistema manejador de contenidos más extendido y de uso abierto es Drupal.

Drupal está constituido por una comunidad de código abierto que además ha sido utilizado en proyectos actuales para organizaciones importantes como la pagina web para la “Casa Blanca” en Estados Unidos.

Probado en miles de servidores, es flexible también para los sistemas operativos. Funciona con los requerimientos básicos que usualmente vienen con la renta de un dominio.

Drupal es construido por una Comunidad de ingenieros de una rama que cambia a lo largo de los años. Una de las características fuertes de Drupal, es la capacidad para extender su sistema y personalizarlo hacia las decisiones que afectan el negocio; a través de lo que son los Módulos. Estos Módulos también son la historia de diferentes requerimientos que se les han presentado a otros desarrolladores en diferentes proyectos, y que se han ido reutilizando y remodelando por otros a través de los años en nuevos proyectos que se encontraron similares.

En conclusión es necesario cambiar la situación actual de la empresa con su sitio web, mediante una reingeniería que ataque los puntos que han hecho difícil el desarrollo y cumplimiento de las expectativas de esta para promocionarse mediante su sitio. Esto es buscando una forma de integrar una comunicación y relación eficaz entre el sitio web y la empresa, mediante procesos que le permiten interactuar con la administración y gestión del sitio, así como el desarrollo y el manejo de contenido de una forma sencilla distribuyendo responsabilidades que permitan un mantenimiento de los procesos a bajo costo.

# 2. BASES TEORICAS.

## 2.1. MARCO HISTORICO Y CONTEXTUAL

EL hospital privado Santa Fe se encuentra localizado en la carretera de Villahermosa a Macuspana del estado de Tabasco, Mexico. Esta privilegiado por estar muy bien ubicado cerca del Hotel Hilton y el aeropuerto de Villahermosa, principales lugares de extranjeros y reuniones de negocios. Esta ubicada en la colonia dos montes, una colonia segura libre de inundaciones, en proceso de modernización. Al ser un hospital privado de especialidades multiple y realizar cirugias tiene todos los servicios de máxima seguridad lo que le da mas ventaja en su infraestructura. Electricidad, internet, agua, gas, aire acondicionado, teléfonos, arquitectura, todo en el hospital esta normalizado teniendo una gran inversión y por lo mismo siendo un lugar con todos los servicios para partir de un proyecto. Ademas esta muy controlado en la parte del hospital, por las enfermedades, curación de pacientes, asi que se trabajaría en la area administrativa que por suerte esta desacoplada para ejercer a una distancia segura de la parte operativa del hospital.

Al estar localizado fuera de la ciudad para llegar al hospital desde la ciudad de villaherosa de tres formas:

Por auto propio. (15 minutos de distancia aproximadamente, cuenta con estacionamiento).

Por colectivo. (Combis en el mercado pasan muy pocas cada media hora. Autobuses de escala a Macuspana pasan cada 45 minutos. No hay transporte después de las 21:00 horas.)

Servicio vehicular particular (Taxi: En cualquier momento pero cobran caro, mas de noche).

No tienen cerca consultorias de Tecnologias de la información, no es una zona de negocios sino de reuniones y de llegadas.

El personal administrativo esta de 9:00 a 2:00 pm, esta saturado en sus ocupación siendo que atiende clientes, proveedores, empleados, maneja los servicios hospital, contrata servicios externos, negocios, etc. Por lo tanto es difícil conseguir su apoyo para estar en el proyecto.

El director del hospital se le tiene que reservar, y usualmente esta de tarde.

Necesidad de normalización con los procesos de tecnologías de la información.

Falta de un equipo de tecnologías de la información en su infraestructura.

La comunidad en la que influye esta empresa mas de cerca es a sus empleados, en sus inversores, clientes antiguos y actuales, el aeropuerto y el hospital, y la colonia dos montes.

Las personas que podrían colaborar en la ejecución de un proyecto de intervención serian el Director del hospital, la administradora, otros administradores, y el encargado de marketing.

Diferentes puntos de vista de cada uno del proyecto.

Director: Quiero una pagina que me muestre lo que he invertido en este hospital, profesional que comunique y refleje estatus.

Administrativo: Expandir las operaciones del hospital, sus activos fijos, la ganancia, el potencial de ganancias, el aumento de otros activos mediante este desarrollo del proyecto.

Encargado de marketing: Que muestre todo, se ve bonito, normalizado, estándar, permita al hospital relacionarse y comunicarse, ampliar sus expectativas y oportunidades de canal.

Personal es consultivo o decisivo. Director es decisivo. Marketing es decisivo. Administracion y encargado de cuentas consultivo.

Utilidad a que personas complementaria en sus funciones de la empresa (interesados) un proyecto de intervención: este proyecto apoyaría al director porque es el complemento del todo al administritativo para que tenga mas activos. Encargado de marketing y publicidad.

Otras partes que se (accederían por el beneficio) beneficiarían con el recurso que le falta al hospital (o con la mejora de los problemas del hospital) (y no son tan importantes). Clientes ocasionales, otras colonias de Villahermosa, municipios y estados de los alrededores, comunidad de doctores, sección amarilla.

Antecedentes: El hospital ha estado bajo remodelación los últimos 5 años, siendo este un gasto muy fuerte y que requería toda su atención. Esta remodelación trajo al hospital el poder ofrecer una serie de servicios de ultima punta, que lo hace destacar como los primeros 5 mejores hospitales privados de tabasco estando a la altura de hospitales privados de renombre como el air, del sureste, angeles.

La empresa tiene doctores que tienen experiencia tanto fuera como dentro de Villahermosa, siendo la visión de sus especialistas muy amplia. El dueño de la empresa es un reconocido doctor de Tabasco, que además tiene una buena visión por los hospitales en los que ha estado, una visión de negocios e integral en las caracteristicas del servicio.

Se promocionan por llamadas, carteles, folletos, radio, periódicos, y sobre todo por el renombre de los médicos que trabajan en ese hospital, siendo doctores muy solicitados y que usualmente tenían una relación de trabajo o reconocimiento de sus habilidades previa por el dueño del hospital. Siendo su personal ejecutivo, después el coste de sus instalaciones y el servicio responsable de sus empleados del que no se ha tenido queja lo que lo mantiene publicitado.

Surge pues la necesidad de un proyecto que encauze esas areas que por su mismo contexto no han tenido relación y que reflejan los problemas que en estos momentos tiene esta empresa.

La empresa mueve a sus miembros mediante las responsabilidades delegadas para buscar el arreglo de las faltas y necesidades en base a su prioritizacion y urgencia definida. Usualmente dejando aquellas que no pueden ser resueltas por su propia estructura organizacional, una vez que se han definido presupuestos y generación de juntas para aprobarlas, contratando expertas externas por políticas de la empresa muy raramente considerando añadir eso a su estructura organizacional, por políticas de ahorro de la empresa.

Los miembros inconformes con la situación o que ven problemas, lo expresan verbalmente cumpliendo su misión de atender específicamente el area que les corresponde.

Indirectamente se han visto estos problemas en:

Falta de Proyectos o dificultad para iniciar cambios en sus procesos de internet. Años sin tocar su pagina web. Desactualizacion de la información.

Disconformidad con su web actual

Pocas visitas web. Pocos beneficios del rendimiento de la web. Desantender el desarrollo de la web, o postergación por la falta de ver un camino, o saber en que punto están, por la misma falta de identificación del problema.

¿Qué hacen? Contratan a otros en base a sus necesidades especificas sin hacer un análisis de las relaciones en las que forman los problemas. Por lo mismo ellos especifican estas necesidades y los servidores las resuelven, pero no atacan a los factores que las originaron y el problema sigue apariciendo en diferentes aspectos pero al final en la misma dimensión.

¿En que plazo se debería dar una respuestas? En cuanto se haga un análisis detallado de la ubicación actual ya que ellos mismos están buscando que sus necesidades particulares se resuelvan, pero lo que se necesita es una intervención especifica que interrumpa este ciclo de incidencias en el que han caído por su propio contexto. Un proyecto de esta magnitud dura alrededor de 4 meses, siendo necesario una determinación del problema. Del que se propondrá un modelo o patrón para salir de este.

El nudo actual: su pagina web ha quedado estatica, obsoleta, por un espacio grande de años y no hay nadie que la atienda o sepa como funciona y no se puede hacer uso o beneficie de ella al no haber documentación o departamento experto en su estructura organizacion que intuya cómo funciona y le aplique una forma de sacar beneficios. Se esta pagando un mantenimiento que se traduce como gasto en la empresa en vez de inversión ya que no se tiene expectativa de que mejore.

## 2.2 MARCO REFERENCIAL

“Los clientes fueron Nokia (Finlandia), Daimler-Benz (Alemania), Sema Group (España), Forschungzentrum Informatik Karlsruhe (FZI, Alemania), y la universidad de Bern (Suiza). Nokia y Daimler-Benz era recientes adaptadores de tecnologías orientadas a objetos, y habían esperado cosechar significantes beneficios de esta táctica. Sin embargo, ahora ellos estaban experimentando varios de los típicos problemas de los sistemas legados: ellos tenían muy grandes, muy valiosos, sistemas de software orientados a objetos que eran muy difíciles de adaptar a cambios de requerimientos. La finalidad del proyecto “FAMOOS” era desarrollar herramientas y técnicas que rejuvenecieran estos sistemas legados orientados a objetos, así continuarían de ser útiles, y fueran mas dóciles a futuros cambios en requerimientos. De cualquier manera, lo mas sorprendentes para nosotros, fue el hecho de que aunque cada uno de los casos de estudios que vimos necesitaba ser rediseñado para muy diferentes razones -tales como la desagregación, la ampliación de los requisitos, la migración a los nuevos ambientes, y así sucesivamente- los problemas técnicos reales con estos sistemas eran extrañamente similares. Esto nos sugiere que tal vez algunas simples técnicas podrían recorrer un largo camino corrigiendo algunos de los problemas más comunes.”

La cita proviene de patrones de reingeniería. En el mismo libro se refieren que los patrones de diseño para los mas experimentados es algo que todos ellos saben, algo que aprenden con la experiencia, y con respecto a los mas nuevos algo que consideran deseable que sepan para poder comunicarse. Los patrones de reingeniería con un conjunto de técnicas conocidas o comunes en el desarrollo de proyectos de software, de buenas practicas, de experiencias y de direcciones que responden a un determinado problema, integrando una solución y también un contra. Para los autores los patrones de diseño son indispensables para todo proyecto de software, ya que describen situaciones comunes, pero deben aplicarse conforme al proyecto tomando en cuenta los análisis de la solución y sus contras.

El proceso de publicidad por la web para una empresa comercial debe entregar resultados a la empresa en base a lo que la misma empresa o la inteligencia de marketing analice, para esto la analista del marketing de empresas en internet apunta:

“¿Cómo mediamos el éxito? Indudablemente necesitábamos apoyarnos en la tecnología. Todavía no conocíamos la verdadera potencia de nuestra herramienta de Analítica, Webtrends, que usábamos básicamente para extraer unas cuántas métricas: visitantes únicos, visitas, páginas vistas, tiempo en el sitio, páginas más consultadas y poco más (sí, sí, sé lo que estáis pensando: todavía me sigo dando cabezazos contra la pared por ello) y siempre tomábamos nuestras decisiones en base a los datos reportados por el Adserver de la agencia de medios, las propias estadísticas de Google y demás colaboradores.”

Los analistas de marketing observan la propia web en donde intervienen para analizar el desarrollo de sus técnicas aplicadas al sitio y tanto el impacto de ellas. Es un proyecto que debe medirse y modificarse de acuerdo a su impacto, se desarrolla pero también se corrige, aunque hay una fase de diseño principal, el plan necesita ser puesto a prueba, y ser modificado si se lo requiere, alcanzar el éxito del marketing se basa en medir los parámetros con los que el propio departamento o la consultoría ha definido el éxito e ir haciendo modificaciones para llegar a el. El departamento de marketing utiliza la tecnología para medir el sitio de internet de la compañía, que tanto es usado, que tanto es visto, etc., los dirigentes del marketing observan su rendimiento en que tanto interactúa el usuario con su pagina web. Ya no se trata de leer solo logos, o misiones y visiones, es decir establecer una pagina que solo comunique la imagen corporativa en un solo sentido. Los planes modernos de marketing en internet buscan la interacción bilateral, y la utilización de elementos dinámicos, y no solo esto, sino que este tipo de web moderno que opera este enfoque, desde hace algunos años se le ha dado el nombre de Web 2.0.

“En los últimos tiempos ha tomado mucha importancia el análisis de todo lo que tenga que ver con la web 2.0 y como podemos determinar cuando tenemos éxito en redes sociales, que se comenta sobre nosotros y nuestra web. El poder medir la distribución de nuestro contenido o como influye en el beneficio que persigue nuestra web es vital para poder decidir la estrategia mas adecuada en la red. No hay una herramienta capaz de medir el impacto en la totalidad del mundo de las redes sociales por lo que la solución puede ser el definir claramente los objetivos que perseguimos en cada una de las redes sociales y tomar decisiones sobre dicha base.”

El implemento de la Web 2.0 no es un concepto nuevo, y viene específicamente dado mas que por una implementación técnica, por un cambio en la función de los sitios web. La web 2.0 es el resultado de un nuevo tipo de enfoque hacia la web, representado por un sistema abierto donde el cliente puede colaborar con la empresa proveyéndole su información, comentarios y hasta extendiendo la propia información de la empresa. Las empresas donde las diferencias entre las que usan y las que no el enfoque Web 2.0 son mas notorios es donde las empresas desarrollan sus servicios principalmente o completamente por internet, como en el caso de la empresa de internet Amazon.com una web corporativa que se dedica a vender libros por internet, la cual fue citada en un articulo por la empresa Oreilly el que define cuales son las Web 2.0.

“Contrasta por ejemplo la posición de Amazon.com. Con competidores como Barnesandnoble.com, su base datos original vino del proveedor de registros ISBN R.R. Bowker. Pero diferente a MapQuest, Amazon se esforzó en expandir la informacion, añadiendo información suplida por el publicador como imágenes de portada, tabla de contenidos, y material de ejemplo. Incluso mas importante ellos alentaron a sus usarios a anotar la informacion, tanto que después de 10 años, Amazon, no Bowker, es la fuente primaria para información bibliografía en libros, una referencia para estudiantes y libreros tanto como conumidores. Amazon también introujo su propio identificador propietario, teh ASIN, que correponde al ISBN donde uno esa presente, y crea un equivalente espacio de datos para productos sin uno. De manera efectiva, Amazon “introdujo y extendió” a sus proveedores de datos.” (Oreilly 2006.)

En la Web 2.0 según Oreilly, trata de informar, pero también de aumentar esa informacion. La información debe cambiar, aumentarse, servirse fácilmente al cliente, inclusive que este la modifique. Es indispensable tomar en cuenta que la creación y mantenimiento de los sitios Web 2.0 estan apoyados sobre las nuevas arquitecturas de software que utilizan elementos como la computación distribuida, la orientacion a servicios, cliente-servidor, middleware, patrones de diseño entre otras.

Tambien hay muchos tipos de Web 2.0, mas que nada la Web 2.0 es un conjunto de tecnologías y practicas que enfatizan en la comunicación y la interacción en internet en dos direcciones, tanto para leer como escribir o sobreescribir. Algunos de los enfoques variados que han surgido de la Web 2.0 son el Marketing 2.0, Health 2.0, Enterprise 2.0.

Sin embargo las empresas buscan obtener los beneficios o el esquema de comunicacion que tienen la web 2.0 al menor tiempo posible, es un gasto para ellos sobre la configuracion o el aspecto tecnico que sustenta los sitios web 2.0. puesto que en el mundo de los negocios se requieren acciones rapidas, flexibles, y adaptables, como lo demuestra un caso de estudio de la empresa Chess Media al implementar un sitio Enterprise 2.0 en una empresa de salud de tamaño medio.

“La compañia no quiso gastar un año de planeacion y diseño del proyecto; en su lugar, prefirió una implementacion basica que podria ser progresivamente desplegada y lo suficientemente agil para adaptarse a lo largo del camino.”

Para este proyecto el caso mas parecido es el de la empresa Hospitales Florida, ubicada en estados unidos .

Debido a crecimientos, renovación del sitio y a el final de soporte para el sistema manejador de contenido que usaban la empresa decidio cambiar por el framework open-source Drupal.

“El equipo de intranet luego comenzó a verificar soluciones open source. Las tecnologías Open Source no habían sido usadas entensivamente en Florida Hospital (fuera de la arena del sistema operativo de nuestro servidor) , asi que el equipo considero sus elecciones cuidadosamente. Despues de buscar en las plataformas lideres Open Source, el equipo recomendó implementar Drupal. El manager del equipo de Intranet, Rick Mann, cito la razón para ir con Drupal, ‘Drupal ofrece un amplio rango de herramientas, una comunidad de desarrollo activa, y nosotros determinamos que podríamos tener el nucleo de la plataforma funcionando relativamente rápido y fácil’”.

En años anteriores tenían problemas con las desactualizaciones de su framework anterior, también dificultades entre los mas de cien sitios para los departamentes internos del mismo hospital ya que eran bastante heterogéneos, estaban dispersos, y no había una comunicacion o forma de conectar entre sus diversos sitios, lo que aumentaba los cuestiones para la administracion y gestion en conjunto.

Tambien Acquia una empresa de consultoria apoyo a la empresa en la planeación, diseño, adaptación y migración al nuevo framework, apoyándolo en sus dudas, puesto que tiene muchos casos de éxito apoyando a empresas similares, siendo consultoria, y no una empresa de TI se puede dedicar mas a servir a la empresa y sus dudas ejecutivas, siendo que TI se ocupa mas de los detalles tecnicos. Esto permitio hacer el cambio, o bien la intervención con un margen muy grande de éxito.

“Acquia Drupal, o la distribución de Drupal de Acquia que incluye el nucleo Drupal y un set de modulos esenciales, ayudo al equipo de intranet del Hospital Florida a mantenerse en el tramo correcto. Empezando con una selección de herramientas disponibles de Drupal, el equipo podría continuar su descentralizado estilo de manejo del contenido que permitiera a los departamenots manejar su propios sitios. Esta seria la clave para la habilidad del equipo para dar soporte a un largo numero de sitios. En teoría, el equipo crearía y hospedaría sitios Drupal usando una plantilla standard y un consistente set de herramientas (modulos), y pasaría el manejo del dia a dia de los sitios a sus respectivos dueños del sitio y creadores de contenido. Esto a su vez liberaría al equipo de intranet para subsecuentemente investigar, desarrollar y producir herramientas adicionales y funcionalidad, para enriquecer por completo la experiencia drupal.”

Con la nueva implementación del framework pudieron migrar a la nueva tecnología, tomando en cuenta que eran cientos de departamentos en el hospital, sin mucha dificultad, y pudiéndose concentrar en el desarrollo de mas funciones para sus aplicaciones y su sitio web y la capacitación de su personal a la nueva plataforma.

Se usara Drupal puesto que permite un desarrollo desacoplado para poder entregar cambiar, mientras se desarrolla o permite su facilidad por el mismo tipo de framework para la actualización y la extensión frecuente sin que esta sea intrusiva, y deje a la red desarrollarse por si misma, mientras se desarrollan o integran nuevas capacidades desde el lado técnico sin que interrumpa y mejor aun, este abierto a los cambios de requerimiento, modificaciones o añado de características que puedan surgir de las renovaciones en las operaciones o planes de marketing.

En suma la herramienta utilizada para desarrollar fue drupal, y lenguaje php. Esta será la base utilizada para este proyecto dado que los recursos instalados en el servidor de produccion son LAMP, que es un entorno básico en el que se desarrollan proyectos Drupal.

Entre los casos de éxito de esta framework y sistema manejador de contenidos, esta una revista de 145 años de edad que ahora publica online con un sistema manejador de contenidos que “esta mas en sincronía con sus políticas” (the nation.com), el gobierno de los estados unidos (sba.gov y whitehouse.gov, entre otros), la campaña del primer senador republicano de Massachustts en 35 años (scottbrown.com), una asociación de interacción internacional de diseñadores (IxDA.org), cientos de miles de sitios de todos los tamaños y propósitos, incluyendo decenas de cientos de sitios hospedados gratuitamente en Drupal 7 como servicio (drupalgardens.com).

## 2.3 MARCO TEORICO

### I. COMO FUNCIONAN LAS APLICACIONES DE INTERNET

### 1. Aplicaciones web

Una [aplicaci](http://www.alegsa.com.ar/Dic/aplicacion%20web.php)ón web es cualquier aplicación que es accedida vía [we](http://www.alegsa.com.ar/Dic/web.php)b por una [red](http://www.alegsa.com.ar/Dic/red.php) como [internet](http://www.alegsa.com.ar/Dic/internet.php) o una [intranet](http://www.alegsa.com.ar/Dic/intranet.php). En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno [del navegador](http://www.alegsa.com.ar/Dic/navegador.php) o codificado con algún lenguaje soportado por el navegador; confiándose en el navegador web para que reproduzca ([renderice](http://www.alegsa.com.ar/Dic/renderizacion.php)) la aplicación.

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de

usuarios potenciales.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo. Desde la perspectiva de un usuario, puede ser difícil percibir la diferencia entre un sitio web y una aplicación web. Según el Diccionario Oxford en línea, una aplicación es "un programa o conjunto de programas para ayudar al usuario de un ordenador para procesar una tarea específica". Una aplicación web es básicamente una manera de facilitar el logro de una tarea específica en la Web, a diferencia de un sitio web estático que es más bien una herramienta de difusión menos personalizada.

El término más decisivo de esta definición es "tarea específica". La aplicación web por lo tanto permite al usuario interactuar directamente contigo y tus datos, todo en forma personalizada, para llevar a cabo esa tarea específica. Si ya has puesto en marcha una búsqueda en Google, has usado una aplicación web. Esto se aplica incluso si: lees el correo desde el correo electrónico de Gmail, pagas tus cuentas en línea, compras en la Web, te estás comunicando con tus amigos y, por último, mantienes tu jardín virtual. El denominador común sigue siendo el mismo: el uso de una aplicación web para llevar a cabo todas estas tareas específicas.

### 2. Historia de las Aplicaciones web

El concepto de la aplicación web no es nuevo. De hecho, uno de los primeros lenguajes de programación para el desarrollo de aplicaciones web es el "Perl". Fue inventado por Larry Wall en 1987 antes de que internet se convirtiera en accesible para el público en general. Pero fue en 1995 cuando el programador Rasmus Lerdorf puso a disposición el lenguaje PHP con lo que todo el desarrollo de aplicaciones web realmente despegó. Hoy en día, incluso muchas de estas aplicaciones se han desarrollado en PHP, como Google, Facebook y Wikipedia.

Unos meses más tarde, Netscape, el navegador web más antiguo y popular, anunció una nueva tecnología, JavaScript, lo que permite a los programadores cambiar de forma dinámica el contenido de una página Web que había sido hasta el momento texto estático. Esta tecnología permite un nuevo enfoque para el desarrollo de aplicaciones Web, que eran, y aún hoy, mucho más interactivas para los usuarios. Por ejemplo, la instantánea de Google, que muestra los resultados de búsqueda en un momento en que la palabra se escribe, hace un uso intensivo de JavaScript. Las actualizaciones del sitio web de productos de Microsoft también utiliza esta tecnología. Al año siguiente, en 1996, dos desarrolladores, Sabeer Bhatia y Jack Smith lanzaron Hotmail (no fue un desarrollo original de microsoft), un servicio de correo en línea que permite (por primera vez) para el público en general para acceder y consultar el correo electrónico siempre que sea los usuarios pudieran estar en cualquier sitio lejos desde su ordenador.

Luego vino la famosa plataforma Flash utiliza para añadir contenido interactivo para sitios Web. Flash hizo su aparición en 1997, conocido como Shockwave Flash. Más tarde, después de ser adquirido por Macromedia y Adobe, Flash se convirtió en una plataforma para desarrollar aplicaciones web interactivas.

El año siguiente marcó un punto de inflexión para los medios de comunicación en línea. De hecho, el 17 de enero de 1998, el sitio web The Drudge Report anunció por primera vez un informe de noticias antes de que se difundiera en los medios de televisión y la prensa tradicional. Se informó el escándalo Clinton/Lewinsky. Este evento fue el detonante del periodismo en línea tal como lo conocemos hoy en día. Antes de esa fecha, internet nunca había sido considerado un medio de comunicación importante.

El mismo año, la compañía Google desarrolló su primer motor de búsqueda en línea que, por su nueva forma de indexar páginas web, facilita enormemente la búsqueda de información en internet. Google sigue innovando y se convirtió en uno de los más prolíficos en cuanto a las aplicaciones Web, con indicación del muy popular Google Maps, Google Docs, Gmail y en aumento. A principios de 2001, poco después de la explosión de la burbuja de internet,

Wikipedia se lanzó como un sub-proyecto de Nupedia, una enciclopedia en línea tradicional.

Para desarrollar su plataforma, se utiliza un tipo de Wikipedia de la aplicación web denominada "wiki", que permite a cualquier usuario agregar contenido. Las contribuciones no se hicieron esperar, y al final del primer año de funcionamiento, Wikipedia ya contaba con 20000 páginas en 18 idiomas. Hoy en día, casi 21 millones de artículos en 285 idiomas conforman el sexto sitio más visitado en el mundo, siendo el primero Google. En 2003, MySpace fue fundado y más tarde, de

2005 a 2008, el sitio se convirtió en el medio de comunicación social más visitado. MySpace fue una plataforma de lanzamiento para otras aplicaciones web conocidos como YouTube, y Slide.com! RockYou, todos los cuales comenzaron como módulos adicionales para los usuarios de MySpace antes de convertirse en sus propios sitios web en su propio derecho. Entonces, tres acontecimientos muy importantes ocurrieron en 2004. En primer lugar, en una conferencia de la Web 2.0 a cargo de John Battelle y Tim O'Reilly, el concepto de "web como plataforma" fue mencionado por primera vez.

Esta innovación allanó el camino para futuras aplicaciones web, es decir, un software que aprovecha las ventajas de la conexión a internet y que se desvían del uso tradicional del escritorio. En segundo lugar, el sitio interactivo de Digg se puso en marcha. Propuso una forma innovadora de crear y encontrar contenido en internet mediante la promoción de noticias y enlaces democráticamente votado por los usuarios. Y, por último, el tercer gran evento, pero no menos importante, fue el lanzamiento de Facebook, que estaba entonces en su infancia, abierto sólo a los estudiantes. Con un millón de suscriptores a finales de 2004, Facebook se ha convertido en el medio de comunicación social más utilizado con más de 900 millones de usuarios. Este es el segundo sitio más visitado en el planeta y tiene la mayor cantidad de fotos compartidas por los usuarios con un total de casi 500000 millones de fotos subidas a la plataforma.

En 2005, YouTube fue lanzado oficialmente, permite a los usuarios compartir vídeos en línea. De simple sitio para compartir vídeos en internet a una plataforma madura que se conoce hoy en día, YouTube ahora ofrece cerca de 4000 millones de videos al día, además de un servicio de alquiler de películas en línea, y, finalmente, episodios de emisión para las empresas o las películas de MGM, Lions Gate Entertainment y CBS. Twitter, por su parte, se puso en marcha en 2006. Con los años, la popularidad de Twitter ha aumentado de 1,6 millones de 'tweets' en

2007 con la impresionante cifra de 340 millones de dólares por día en marzo de

2012 (equivalente a casi 4000 'tweets' por segundo). El año 2007 estuvo marcado por la aparición del iPhone, que fue sin duda responsable de la llegada de la nueva moda para las plataformas móviles y aplicaciones web. Ahora son accesibles por teléfono inteligente. A principios de 2011, la empresa Kickstarter, que facilita la financiación de

proyectos en línea de forma participativa, ha llegado a los 4000 proyectos con más de 30 millones de dólares en donaciones. Por otra parte, casi el 44% de los proyectos se han iniciado con éxito desde esta plataforma. En resumen, los desarrolladores de aplicaciones web de hoy están dando forma al futuro digital.

### 3. Consideraciones técnicas

Una ventaja significativa es que las aplicaciones web deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux y otros sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes. Sin embargo, hay aplicaciones inconsistentes escritas con HTML, CSS, DOM y otras especificaciones estándar para navegadores web que pueden causar problemas en el desarrollo y soporte de estas aplicaciones, principalmente debido a la falta de adhesión de los navegadores a dichos estándares web (especialmente versiones de Internet Explorer anteriores a la 7.0). Adicionalmente, la posibilidad de los usuarios de personalizar muchas de las características de la interfaz (tamaño y color de fuentes, tipos de fuentes, inhabilitar Javascript) puede interferir con la consistencia de la aplicación web. Otra aproximación es utilizar Adobe Flash Player o Java applets para desarrollar parte o toda la interfaz de usuario. Como casi todos los navegadores incluyen soporte para estas tecnologías (usualmente por medio de plug-ins), las aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Dado que ignoran las configuraciones de los navegadores, estas tecnologías permiten más control sobre la interfaz, aunque las incompatibilidades entre implementaciones Flash o Java puedan crear nuevas complicaciones, debido a que no son estándares. Por las similitudes con una arquitectura cliente-servidor, con un cliente "no ligero", existen discrepancias sobre el hecho de llamar a estos sistemas “aplicaciones web”; un término alternativo es “Aplicación Enriquecida de Internet”.

### 4. Estructura de las Aplicaciones web

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa, y un motor capaz de usar alguna tecnología web dinámica (ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python (programming language) o Ruby on Rails) que constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa.

El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

### 5. Ventajas

Una de las [ventajas](http://www.alegsa.com.ar/Dic/aplicacion%20web.php) de las aplicaciones web cargadas desde internet (u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. También la posibilidad de ser ejecutadas en múltiples [plataformas](http://www.alegsa.com.ar/Dic/plataforma.php).

- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.

- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.

- No ocupan espacio en nuestro disco duro.

- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.

- Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.

- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.

- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.

- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.

- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.

- Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios

usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios

u oficina.

- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas (RIAs).

### 6. Servidor web

Imagínense a los servidores web como computadoras especialmente preparadas para estar prendidas las 24 horas, los 365 días del año. La principal función de un servidor Web es almacenar los archivos de un sitio y emitirlos por Internet para poder ser visitado por los usuarios. Básicamente, un servidor Web es una gran computadora que guarda y transmite datos vía Internet. Cuando un usuario entra en una página de Internet su navegador se comunica con el servidor enviando y recibiendo datos que determinan qué es lo que ve en la pantalla. Por eso decimos que los servidores Web están para almacenar y transmitir datos de un sitio según lo que pida el navegador de un visitante. Cada servidor Web y cada computadora conectada a Internet tiene asignado una dirección de IP irrepetible que lo identifica en la red incluyendo la PC que tenés en tu casa. La dirección de IP vendría a ser como los datos del remitente en una carta postal. Cuando llegas a un sitio Web, se envía un pedido desde tu dirección de IP hacia la dirección IP del servidor. El servidor Web responde mandando datos a la dirección IP que los pide. Esto es lo que pasa cada vez que estamos navegando un sitio en Internet.

Más del 90% de los sitios en Internet utilizan un servidor alquilado a través de una empresa de Web hosting.

La capacidad de un servidor depende del tipo de servidor que sea y de los componentes que lo conforman. Sin los servidores Web la Internet tal como la conocemos, no existiría. Los servidores son como la columna vertebral de la estructura de Internet. La industria del Web hosting es simplemente la forma de alquilar esos espacios de memoria y administración de datos.

Funcionamiento

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, al teclear [www.google.o](http://www.google.org)rg en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP.

Hay que distinguir entre:

 Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

 Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre

en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier

cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

El hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un lenguaje de marcas y HTTP es un "protocolo".

Aplicación del lado del Servidor

Una aplicación del lado del servidor es cualquier programa o conjunto de instrucciones diseñadas con la finalidad de que un Servidor Web las procese para realizar alguna acción. Las aplicaciones del lado del servidor están escritas

mediante algún lenguaje de programación, entre los que destacan:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lenguaje | Año | Sistema Operativo |  | Versión |
| PHP | 1995 | Multiplataforma | 5.3.5 |  |
| ASP.Net | 1998 | Windows (Algunas versiones) 4.0 | | |
| Perl | 1987 | Multiplataforma 5.12.3 | | |
| Python |  | 1991 Multiplataforma 3.2.0 | | |
| Ruby | 1995 | Multiplataforma 1.9.3-p125 | | |

El 75% de las aplicaciones del lado del servidor están escritas en PHP, seguido de ASP y las demás opciones usadas de forma alternativa y muy casual.

En 2009 Node.js fue creado por Ryan Dahl, abriendo públicamente la oportunidad de usar javascript del lado del servidor. Evento que revoluciona la web, y da inicio a una nueva era del desarrollo en internet. Node.js, dado a sus usuarios,

programadores web de todo el mundo, está creciendo como la mejor opción para

desarrollo de aplicaciones del lado del servidor, desplazando lenguajes

tradicionales en esta tarea. Node.js trabaja sobre el rápido motor V8 que usa

Google Chrome para la interpretación de javascript. Procesamiento del lado del servidor

Un servidor web tiene la función de procesar los scripts del lado del servidor para dar una salida en HTML y otros lenguajes del lado del cliente al Navegador Web del cliente. La información a procesar podrá ser cedida por el cliente al script mediante cualquier aplicación en el entorno del Navegador. Para ello pueden utilizarse formularios web, enlaces con los valores implícitos en la cadena o cualquier otro método.

### 7. Lenguaje de marcado de Hipertexto

Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, etc.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tablets, etc.). Sin embargo, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Usualmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Un navegador no actualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que

permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores aún mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, aún existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

### II. COMO SE DESARROLLA UNA APLICACIÓN WEB

### 1. ¿Qué es el diseño y desarrollo de las aplicaciones web?

El diseño y desarrollo de aplicaciones web consiste en implementar sus necesidades, objetivos o ideas en Internet utilizando las tecnologías más idóneas según su proyecto.

### 2. Framework

La palabra inglesa "framework" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Por ejemplo, un equipo que usa Apache Struts para desarrollar un sitio web de un banco, puede enfocarse en cómo los retiros de ahorros van a funcionar en lugar de preocuparse de cómo se controla la navegación entre las páginas en una forma libre de errores. Sin embargo, hay quejas comunes acerca de que el uso de frameworks añade código innecesario y que la preponderancia de frameworks competitivos y complementarios significa que el tiempo que se pasaba programando y diseñando ahora se gasta en aprender a usar los frameworks.

Fuera de las aplicaciones en la informática, puede ser considerado como el conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para facilitar una solución dada.

### 8. Content management system

Es un sistema que consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio web. Su función es manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio web sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior (moderador o administrador) que permite que estos contenidos sean visibles a todo el público (los aprueba).

Esto permite gestionar, bajo un formato estandarizado, la información del servidor, reduciendo el tamaño de las páginas para descarga y reduciendo el coste de gestión del portal con respecto a un sitio web estático, en el que cada cambio de diseño debe ser realizado en todas las páginas web, de la misma forma que cada vez que se agrega contenido tiene que maquetarse una nueva página HTML y subirla al servidor web.

Un sistema de administración de contenidos siempre funciona en el servidor web en el que esté alojado el portal. El acceso al gestor se realiza generalmente a través del navegador web, y se puede requerir el uso de FTP para subir contenido. Cuando un usuario accede a una URL, se ejecuta en el servidor esa llamada, se selecciona el esquema gráfico y se introducen los datos que correspondan de la base de datos. La página se genera dinámicamente para ese usuario, el código HTML final se genera en esa llamada. Normalmente se predefinen en el gestor varios formatos de presentación de contenido para darle la flexibilidad a la hora de crear nuevos apartados e informaciones.

Gestión de usuarios

Dependiendo de la plataforma elegida se podrán escoger diferentes niveles de acceso para los usuarios; yendo desde el administrador del portal hasta el usuario sin permiso de edición, o creador de contenido. Dependiendo de la aplicación podrá haber varios permisos intermedios que permitan la edición del contenido, la supervisión y reedición del contenido de otros usuarios,etc.

El sistema de gestión de contenidos controla y ayuda a manejar cada paso de este proceso, incluyendo las labores técnicas de publicar los documentos a uno o más sitios. En muchos sitios con estos sistemas una sola persona hace el papel de creador y editor, como por ejemplo en los blogs personales.

Ventajas y Oportunidades

El gestor de contenidos facilita el acceso a la publicación de contenidos a un rango mayor de usuarios. Permite que sin conocimientos de programación ni maquetación cualquier usuario pueda añadir contenido en el portal web.

Además permite la gestión dinámica de usuarios y permisos, la colaboración de varios usuarios en el mismo trabajo, la interacción mediante herramientas de comunicación.

Los costes de gestión de la información son mucho menores ya que se elimina un eslabón de la cadena de publicación, el maquetador. La maquetación es hecha al inicio del proceso de implantación del gestor de contenidos.

La actualización, backup y reestructuración del portal son mucho más sencillas al tener todos los datos vitales del portal, los contenidos, en una base de datos estructurada en el servidor.

### 9. Drupal

**DESARROLLO DE UN SITIO WEB CORPORATIVO 2.0 PARA EL DEPARTAMENTO DE PROMOCION Y MARKETING DEL HOSPITAL PRIVADO SANTA FE**

**Web 2.0**

El concepto Web 2.0 inició en el año 2004 con una lluvia de ideas entre O’Reilly y MediaLive International en una conferencia. El análisis era que con las múltiples crisis de las que no se reponían varios sitios en internet se esperaba un efecto domino que anunciara un desastre descomunal en la web; lejos de eso, nuevas vivificantes aplicaciones y la integración de nuevos sitios con sorprendente regularidad habían vuelto el ecosistema de internet más importante que nunca. ¿Quiénes eran y porque compartían similitudes las compañías de internet que estaban sobreviviendo al colapso? ¿Era posible que el colapso dot-com estaba marcando una ruptura para la web? ¿Tendría sentido que esta nueva forma de actuar tuviera un nombre como “Web 2.0”? O’Really convino que así era, y por eso la conferencia Web 2.0 comenzó.

En el 2005 que significa Web 2.0

Como diferenciar la Web 1.0

La web 1.0 como plataforma

Si Netscape era el contendiente estándar por la Web 1.0, Google era el más ciertamente contendiente para la Web 2.0.

El viejo paradigma del software

Netscape

Google

La web como servicio

El valor subio sobre los navegadores web, y los servidores web.

Autoservicio. Algoritmos.

Arquitecturas de servicios.Arquitectura de participación

Tags, colaborar. Wikipedia, Flickr, BitTorrent, Google, Yahoo.

El servicio es major entre mas personas lo usan.

La infraestructura de la web es colaborativa. Ambientes colaborativos.

Los efectos en red de contribuciones de usuarios son la clave de mercado dominante en la era Web 2.0.

En muchos casos donde hay un costo significante para crar data, puede haber una oportunidad para una Web 2.0. En otro casos el ganador será la compañía que introduzca añadimiento de usuarios, y convierta la información capturada en un sistema de servicio.

Esta claro que los standars y soluciones efectivamente convierten ciertas clases de datos en subsistemas usable del sistema operativo de internet, habilitaran la siguiente generación de aplicaciones.

Final del ciclo de vida del software.

No es un accidente que el sistema de administración de google, su red y sus técnicas de balanceo de cargas sean a lo mejor un secreto mas guardado que sus algoritmos de búsqueda. El éxito de google al automatizar esos procesos es una parte clave del costo de ventaja sobre sus competidores.

Los lenguajes para sistemas que requieren constante cambio.

Usuarios deben tratarse como co-desarrolladores en una refleccion de las practicas en open source. El lema del open source “libera temprano, y libera usualmente” es un hecho que se ha convertido en una posición mas radicalñ en la cual los productos son desarrollados mientras están abiertos, con nuevas características integradas en una base mensual, semanal o incluso diaria. No es un accidente que servicios como Gmail, Google Maps, Flickr, del.icio.us, y como ellos parecen haber mantenido un logo “Beta” a la ves por varios años.

El modelo de negocios de Microsoft depende en todos actualizando sus entornos de cómputo cada dos o tres años. Google depende de todos explorando que hay de nuevo en sus entornos de cómputo cada día.

Modelos ligeros de programación

Software sobre el nivel de un único dispositivo

Experiencias de usuario ricas.

Experiencia de internet ricas.

La oportunidad competitiva para los nuevos entrantes es abrazar completamente el potencial de la Web 2.0. Compañías exitosos crearan aplicaciones que aprendan de sus usuarios, usando una arquitectura de participación para construir una ventaja que no solo este en la interfaz del software, sino también en la riqueza de la información compartida.

Apoya modelos ligeros de programación que permitan sistemas débilmente acoplados. La complejidad de los web servicios anunciados para corporaciones está diseñada para permitir débil acoplamiento. Mientras sea necesario en varios casos, varios de las aplicaciones mas interesantes pueden permanecer débilmente acopladas, incluso frágilmente. El enfoque de la Web 2.0 es muy diferente del enfoque tradicional de las tecnologías de la información.

Piensa en delegar, no coordinar. Simple servicios web, como RSS y REST-based web services son acerca d delegar data, no controlar lo que pasa hasta que obtiene el fin de la conexión. Esta idea es fundamental al internet mismo, una reflexión de lo que es conocido como un principio end-to-end.

Diseña para “Hackeabilidad” y remixeabilidad. Sistemas como el original web, RSS, y AJAX tienen esto en común: las barreras para reusarlo son extremadamente bajas. Mucho de lo usable en software actualmente es open source, pero inclus cuando no lo es, tiene poca en la forma de protección de la propiedad intelcetual. La opción del navegador web “Ver código fuente” hace posible para cualquier usuario copiar cualquier otra pagina web de otro usuario; RSS fue diseñado para empoderar al usario a ver el contenido que el o ella quiera, cuando quiera, no al mejor momento del proveedor de la información; los mas exitosos web services son aquellos que han sidofacilmente llevados en nuevas direcciones inimaginados por el creador. La frase “algunos derechos reservados” fue popularizada por Creative Commons para contrastar con el mas típico “Todos los derechos reservados”, es una muy útil guía.

En pocas palabras para resumir lo que es el núcleo de competencias de las compañías Web 2.0:

Servicios, no software empacado, con bajo costo de escalabilidad.

Control sobre únicos, difíciles de recrear fuentes de datos que se enriquezcan entre mas gente los uso.

Ver a los usuarios como co-desarrolladores.

Implementar inteligencia colectiva.

Elevar los lados en los que el cliente se auto-sirva

Software arriba de los niveles de un simple dispositivo

Ligereza en las interfaces de usuario, modelos de desarrollo, y modelos de negocios.

Además excelencia en un área, puede decirte más que pequeños pasos en las siete áreas.

**COMPUTO DISTRIBUIDO Y ARQUITECTURAS DE SISTEMAS DISTRIBUIDOS**

Computo distribuido es acerca de diseñar algoritmos para un set de procesos que tratan de lograr una forma de cooperacion. Detrás de ejecutarlos concurrentemente, alguno de los procesos del sistema distribuido podrian parar de operar, o por la instancia de un error ser desconectados, mientras otros podrias seguir vivos y continuar operando. Esta es una nocio muy particular de fallas parciales que es una caracteristica de un sistema distribuido. En hechos, esto puede ser muy util si uno realmente siente que necesita diferenciar un sistema distribuido de un sistema de concurrencia.

“Un sistema distribuido es uno donde una falla en una computadora que tu ni siquiera sabias que existia puede mostrar tu propia computadora inusuable.”

Cuando un elemento del elemento de procesos ha fallado, o ha sido desconectado, el reto es por los procesos que aun estan operando por sincronizar sus actividades en una forma consistente. En otras palabras, la cooperacion debe estar hecha robusta a tolerar fallas parciales. Esto hace que el computo distribuido muy dificil, aunque extremadamente estimulante problema.

Debido a los procesos en sincronia que deben mantener entre la infraestructura, puede incluso ser imposible detectar acertadamente si una falla en particula proviene de una falla de un proceso o de la falla de red. Esto hace incluso mas dificil el problema de la aseguracion de una consistente cooperacion. El reto de los investigadores en computo dsitribuido es precisasamente divisar algoritmos que provean a los procesos que permanezcan en operación con suficiente informacion consistente de tal manera que puedan cooperar correctamente y resolver tareas comunes.

De hecho mucho de los programas que usamos hoy son programas distribuidos. Simple rutinas diaras, como leer email o navegar en la web, invuelven alguna forma de computo distribuido. Igualmente cuando usamos estas aplicaciones estamos tipicamente enfrentados con la forma mas simple de computo distribuido: computo tipo cliente-servidor. En el computo de tipo cliente servidor, un procesos centralizado, el servidor, provee un servicio a varios clientes remoto. Los clientes y el servidor se comunican intercambiando mensajes, usualmente siguiente la forma de requerimient-respuesta de interaccion. Por ejemplo, en orden para desplegar una pagina web a un usuario, un navegador manga una solicitud al servidor WWW y expera obtener una respuesta con la informacion para ser desplegada. La dificultad del computo distribuido, lograr una forma de cooperacion en presencia de fallas parciales puede incluso aparecer en esta forma simple de interaccion.

Volviendo a nuestro ejemplo del navegador, es razonable esperar que el usuario continue navegando en la web si el sitio que consulta falla (como si automaticamente cambio a otros sitios) y es mas razonable que los procesos del servidor mantienen en proveer informacion aotros procesos de clientes e incluso cuando alguno de ellos o se va desconectado.

Los problemas arriba señalados son dificiles de abordar cuando el computo distribuido es limitado a la interracion entre 2 dispositivos, tales como el caso cliente-servidor. Igualmente hay mas tipos de computo distribuido que computo cliente-servidor. Usualmente, no solo dos pero muchos procesos necesitan que cooperar y sincronizar sus acciones para lograr una meta en comun. La existencia de solo dos, pero procesos multiples no hacen la tarea de computo distribuido de ninguna manera mas simples. Algunos hablamos acerca de interacciones multi-party en el caso general. En general ambos patrones podrian coexistan de una manera cercanamente natural. Actualmente una aplicación real distribuido deberia tener partes siguiente la interaccion del patron cliente servidor y otras partes siguiente una interaccion multi-party. Esto podria ser incluso una forma de ver. Por ejemplo, cuando un cliente contacta un servidor para obtener un servicio, esto podria ser de otra forma, en la forma para proveer un servicio, el servidor mismo necesita requerir la asistencia de varios otros servidores cuya participiacion necesita coordinar para satisfacer el requerimiento del cliente.

Justo como el acto de reir, el acto de abstraer esta restringido a poca especies naturales. Capturar propiedades que son comunes a una larga y significante rango de sistemas, ayudar a distinguir abstracciones fundamentales de las accesorias y prevenir a diseñadores e ingenieros de reinventar una y otra vez las mismas soluciones para los mismos problemas.

**DESARROLLO ORIENTADO A COMPONENTES**

En el contexto de Ingenieria de software un activo reutilizable es un producto diseñado de forma recurrente en el desarrollo de muchos sistemas y aplicaciones. Ejemplos de activos reutilizables son: algoritmos, patrones de diseño, esquemas de base de datos, arquitectura de software, especificaciones de requerimientos, de diseño y de prueba, entre otros.

En los ultimos años, como resultado de presiones crecientes sobre la industria del software orientadas a reducir drasticamente el costo y tiempo de desarrollode sistemas y aplicaciones, sin afectar los niveles de calidad del producto ha surgido un nuevo activo reutilizable denominado Componente de Software. Se han propuesto numerosas definiciones a este termino, entre las cuales destacan las siguientes:

Un componente es una parte no trivial casi independiente y reemplazable de un sistema que cumple una funcion dentro del contexto de una arquitectura bien definida. Un componente cumple con un conjunto de interfaces y provee la realizacion fisica de ellas.

Un componente de software es una unidad de composicion con interfaces especificadas contractualmente y solamente dependencias explicitas de contexto. Un componente de software puede ser desplegado independientemente y esta sujeto a composicion por terceros.

Un componente es un artefacto de software autocontenido y claramente identificable que describe o ejecuta funciones especificas; que tiene, ademas una interfaz claramente establecida, una documentacion apropiada y un status de uso recurrente bien definido.

Un componente se considera una implementacion opaca debido a que su distribucion predominantemente es un formato bnario y sus consumidores lo utlizan como una “caja negra” a traves de su interfaz. Dicho aspecto esta alineado con el principio de encapsulamiento de la programacion orientada a objetos. Por otra parte la composicion por terceros implica que los componentes son intrisecamente reutilizables debido a que un sistema basado en componentes puede ser ensamblado con relativa facilidad por un integrador empleando componentes suministrador por multiples proveedores independientes. Finalmente la coordinacion e interaccion entre componentes exige un marco regulario estandarizado (modelo de componentes) que establece la infraestructura de software reqerida (Framework) y las convenciones y restricciones de diseño de los mismos.

Tal como lo refleja la definicion anterior un componente pude ser visto desde dos perspectivas distintas, como.

1. Implementacion: lo scomponeentes se pueden ensamblar y desplegar para crear sistemas y aplicaciones que se ejecutan en un computador
2. 2. Abstraccion de arquitectura: los componentes expresan las reglas de diseño que impone el modelo de componentes.

Estan disponibles de tecnologias de componentes tipicamente asociadas con infraestructuras de procesamientro distribuidos (por ejemplo Enterprise javaBeans, CORBA Componenten Model, y .NET), y aplicaciones de escritorio (ejemplo ActiveX y JavaBeans)

Una de las caracteristicas mas importantes de los componentes es que son reutilizables. Para ellos los componentes deben satisfacer como minimo el siguiente conjunto de caracteristicas:

* identificable: un componente debe tener una identificacion clara y consistente que facilite su catalogacion y busqueda en repositiorios de componentes.
* Accesible solo a traves de su interfaz: el componente debe exponer al publico unicamente el conjunto de operaicones que lo caracteriza (interfaz) y ocultar sus detalles de implementaicon. Esta caracteristica permite que un componente sea reemplaza por otro que implemente la misma interfaz.
* Sus servicios son invariantes. Las poeraciones que ofrece un componente a traves de su interfaz, no deben variar. La implementacion de estos servicios puede ser modificado, pero no debeb afectar la interfaz.
* Documentado: un componente debe tener una documentacion adecuada que facilite su busqueda en repositorios de componentes, evaluacion, adaptacion a nuevos entrnos, integraicon con otros componentes y acceso a informacion de soporte.

Adicionalmente para favorecer su reutilizacion es desable que un componente sea:

-generico: sus servicios pueden ser usados en una gran variedad de aplicaciones.

-autocontenido: es conveniente que un componente dependa lo menos posible de otros componentes para cumplir su funcion e forma tal qe pueda ser desarrollado, probado, optimizado, utilizado, entendido y modificado individualmente.

-mantenido: es deable que un componentente (como toda pieza de software) este inmerso en un proceso de mejoramiento continuo que le garantice al integrador nuevas versiones que incluyan correctivos, optimizaciones y nuevas caracteristicas. Esto contribuye a que dicho componente sea seleccionado con mayor frecuencia para formar parte sistemas de software.

-independiente de la plataforma (hardware y sistema operativo) del lenguaje de programacion y de las herramientas de desarrollo: existen diversas plataformas de computo de uso frecuente (ejemplo Windows/Intel, Solaris/Sparc, OSX/PPC, Linux/intel) y es deable que un componente pueda ejecutarse en todas ellas. Asimismo ya que existe una amplia gama de lenguajes depgroamacion y herramientas de desarrollo es natural que encontremos componentes escritos empleando lenguajes y herramientas de la preferencia del programador, por lo tanto es deable que dichas preferencias no limiten el uso de los componentes.

-puede ser reutilizado dinamicamente: puede ser cargado en tiempo de ejecucion en una aplicación.

-caccedido uniformemente sin importar su localidad: la forma de invocar los servicios ofrecidos por los componentes debiese ser independiente de su ubicación (local o remota). Para ello el modelo de componentes debera estar basado en tecnologias d eprocesamiento distribuido tales ocmo corba, RMI, .NET Remoting.

Una interfaz define el conjunto de operaciones que un componente puede realizar, estas operaciones son llamadas tambien servicios o responsabilidades. Las interfaces proveen un mecanismo para interconectar componentes y controlar las dependencias entre ellos.

La naturaleza de la interfaz varia dependiendo del lenguaje programacion empleado para implementar el componente.

En general una interfaz de progrmacion de aplicaciones (API, Application Programming interface) es una especificaicon, en un lenguaje de programacion, de las propiedades de un modulo de software. Los clientes del modulo solo deben depender exclusivamente de las propiedades definidas por el API de forma explicita.

Los parrafos anteriores solo descirben a las interfaces como una manera de especificar el flujo unidireciconal de dependencia que tiene un cliente respecto a un componente. Sin embargo es mejor decirq ue un cliente y un componente dependen el uno del otro, un cliente depende de la forma en que un componente provee sus servicios, y un componente depende de cómo los clientes utilizan los servicios que este ofrece. Esta interdependencia ha llevado a acuñar el termino Contrato de Interfaz en al literatura de investigacion acerca de sistemas basados en componentes.

Bajo el modelo de desarrollo de software baado en componentes las nuevas aplicaciones se construyen mediante la integracion o composicion de componentes. “El procesos de construir aplicaciones mediante la interconexion de componentes de software a traves de sus interfaces (de composicion” Notese que se hace especial enfasis en las interfaces como elementos fundamentales para lograr la composicion de componentes.

La composicion puede concebirse como una relacion cliente-servidor entre dos componentes. El componente cliente solicita un servicio (operación del componente servidor, el cual ejecuta la operaicon solicitada y devuelve los resultados al cliente. El servidor produce un resultado que es consumido por el cliente.

Ademas de los componentes, los frameworks se consideran entidades sujetas a composicion. En consecuencia existen tres clases principales de interacicon en los sistemas basados en componentes.

Somerville clasifica los procesos de desarrollo de software basados en componentes:

-Desarrollo de componentes: este proceso implica la adaptacion o desarrollo de componentes con el proposito expreso de ser reutilizados en futuras aplicaciones. Su objetivo es producir repositorios de activos que puedan ser reutilizados en el desarrollo de software.

-Desarrollo de software conr eutilizacion de componentes: Es un proceso en el cual el desarrollo de una nueva aplicación involucra la reutilizacion de un conjunto de componentes existentes. Este enfoque maximiza la reutilizacion de componentes de software existentes este enfoque maximiza la reutilizaicon de componentes de software existentes y reduce el numero de componentes que requieren ser desarrollados en su totalidad (desde cero). Para ser exitoso este proceso demanda dos condiciones minimas: 1) la existencia de repositorios o bases de ocmponentes reutilizables, y 2) que los ocmponentes sean confiables y actuen de acuerdo a sus especificaicones.

Las tres primeras fases del modelo son similares al os modelos de procesos tradicionales. La fase de analisis del contexto permite que el grupo de desarrollo adquiera un conocimiento adecuado del dominio o contexto del sistema en desarrollo. Las fases de Descubrimiento, Analisis y especificaicon de requermientos se encargan de identificar las necesidades y requerimientos de los usuarios, asi como analizarlos especificarlos graficamente y documentarllos.

La fase de diseño del sistema establece y describe la arquteictura del software. Describe cada uno de los componentes qe requiere tal estructura y como esos componentes se interconectan para interactuar. El grupo de desarrollo debe a partir de esta arquitecuta deterinar cuales componentes se pueden reutilizr y cuales requieren ser desarrollados en su totalidad. Los componentes reutilizados deben ser adaptados, para satisfacer los requerimientos del sistema; mientras que los ocmponenetes nuevos, deben ser disñeados, codificados y probados separadmente durante la fase de implementacion. Las pruebas del sistema permiten detectar errores en la integracion de los componentes. Finalmente la fase de entre se encarga de la instalacion, adiestramiento de usuarios y puesta en operación del sistema.

Finalmente, estamos asistiendo al nacimiento de los WebServices como una forma de facilitar el acceso a las aplicaciones a traves de internet. Pero no solo eso, sino que permiten llevar el adjetivo componente a los servicios, mas alla de los componentes. En general los WebServices ofrecen imporantes ventajas com son el acceso a los servicios desde cualquier sitio en la red (pervasiveness), permiten simplificar el acceso y facilitar la interoporabilidad y tambien posibilitan pasar de las tipicas aplicaicones Web, que son 2-tier, a aplicaciones mas complejas, que requieren una arquitectura n-tier. Y por ultimo suponen un gran avance frente a los componentes desde un punto de vista del marketing pues permiten realizar un alquiler de servicios externos frente al desarrollo o a la compra de componentes. Quiza el ambito de aplicación actual de los WebServices sea e comercio electronico pero sin duda ocuparan un nivel similar a los componentes de software en un futuro muy cercano. De ahí que sea necesario tambien comenzar a trabajar en modelos de calidad para este tipo de servicios.

En general el desarrollo de software basado en componentes puede verse como una extension natural del a programacion orientada a objetos dentro del ambito de los sistemas abiertos y sitribuidos. Este paradigma se basa en el uso de los componentes de software como entidades basicas del modelo, entendiendo por componente “una unidad de composiciomm de aplicaciones de software que posee un conjutno de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros ocmponentes de forma independiente en tiempo y espacio”.

Numerosas son las caracteristicas que aporta la programacion orientada a comonentes frente a la programacion orientada a objetos tradicional. Entre llas, el desarrollo de los componentes de forma independiente del contexto en donde seran ejecutados, la reutilizacion por composicion (Frente a herencia, al tratarse de entidades “binarias” o “cajas negras”), la introspeccion (facilidad para interrogar al componente sobre sus propiedades y metodos de forma dinamica, normalmente mediente el uso de reflexion) neuvas formas de comunicación (como los “eventos” y las comunicaciones asincronas frente a los rudimentarios de los objetos), el enlazado dinamico y composicion tardia, la extensio de las IDLs tradicionales, etc.

Una de las principales ventjas del desarrollo de software basado en componentes de basa en la reutilizacion de los mismos. De esta forma, los comonentes se diseñan y desarrollan con el objetivo de poder ser reutilizados en ortas aplicaciones, reduciendo el tiempo de desarrollo, mejorando la fiabilidad del producto final ((al usar componentes ya probados previamente), y siendo mas competitivos en costes.

**-DISEÑAR UNA APLICACIÓN DE SERVIDOR**

Necesitaremos diseñar una aplicación de servidor.

La meta del diseño de software de negocios es crear herramientas de informacion que soporten las actividades de negocio de la organización. El desarrollador de software tiene que trabajar de cerca con los ususarios finales y ser capaz de comunicarse en lenguaje de negocios tanto en lenguaje de computo. Al mismo tiempo los usuarios tienen que educarse para entender las capacidades y limitaciones de una computadora. Las necesidades de negocio cambian rapidamente, asi que una metodologia efectiva de diseño debe ser flexible y adaptarse facilmente a cambiantes requirimiento de negocios.

El computo N-tier complica el diseño de los procesos incuso mas. Mientras en los tradicionales two-tier cliente-servidor se enfatizan aplicación e interfaces de usuarios, diseño de n—tier requiere tanto el diseño de software orientado a la aplicación como el diseño de objetos orientados a los procesos de negocios. La presion esta en producir aplicaciones en tiempos cortos mientras crean con cada vez mas robustos, reusables objetos de negocios. Los dias de la programacion “cowboy” se han acabado.

Afortunadamente hay metodologias y herramientas para apoyar estos requirimientos. El desarrollo unido de la aplicación (JAD) le brinda al equipo un puente entre desarrolladores y usuarios finales para hacer el diseño juntos. Iterativo, desarrollo incremental provee ciclos cortos de diseño y programacion para asegurar que el proyecto se quede en la pista y conosca las necesidades del usuario final. El modelo unificado de lenguaje (UML) puede ser usado tanto por desarrolladores de software y finales usuarios para comunicar ideas de diseño. Herramientas CASE como Rational Rose pueden ayudar en este proceso creando diagramas UML, generando un esqueleto de codigo y luego actualizar tanto los modelos UML como los procesos de diseño.

El diseño de aplicaciones de servidor aun esta en su infancia. Las metodologias estan evolucionando y proveedores estan constantemente introduciendo nuevas herramientas.

Alguna de estas son:

Diseño de aplicaciones colaborativo.

Diseño de objetos de negocio

Desarrollo iterativo

Notacion UML

Juntas de necesidades de usuarios.

Por muchos años el metodo aceptado para el diseño de software fue el metodo en cascada. Mientras el metodo en cascada tiene sus adherentes, el desarrollo iterativo o ciclico se ha vuelto mas prominente en los ultimos pocos años. El desarrollador y los usuarios finales forman un desarrollo colaborativo en el que trabajan juntos a traves del proceso entero.

Este proceso tienen ventajas al desarrollo oranizacional. Los usuarios finales ven que sus contribuciones hacen una diferencia y sienten que los otros le s estan apoyando en su trabajo. La administracion obtiene resultados rapidos y tangibles de su inversion, y sobre el tiempo estaran a comprometerse mas con los recursos de tecnologia de la informacion. Incluso los desarrolladores de software se benefician, reciviendo mas reconocimiento de sus esfuerzos. Los proyectos se mueven mas alla del desarrollo de software y hasta se vuelven una oportunidad para mejorar los prcoesos de negocios.

Al mismo tiempo algunos contras existen que tu debes monitorear y manejar. Tu debes poner limitaciones desde el comienzo del proyecto para limitar ambos contexto y tiempos. Un proyecto abierto-cerrado puede desarrollar vida en si mismo y nunca ser completado. Los proyectos pueden ser facilmente desviarse o llevados en direcciones erroneas, resultando en fallas para resolver el problema reginal. Poner juntos a los desarrolladores de software y los usuarios finales puede causa dificultades de comunicación y conflictos personales, pero con liderazgo de equipo efectivo y un manejo vigilante deberia prevenir estos problemas.

Para hacer que todo los objetos trabajen consistentemente, se necesita colocar un compresivo y flexible standard. En varios casos el middleware requerira adherencia a los estandares de la arquitectura o te generara un esqueleto de codigo. En otros casos el equipo de desarrollo debe acordar colocar standares que puedan evolucionar tanto como la aplicación de servidor crezca. Estandares deben incluir convenciones de nombres, consistentes formatos de data, jerarquia de clases, objectos de excepciones, formatos de documentacion, entre otras consideraciones.

La unica manera de saber si un diseño puede ser trasladado a codigo es dejar a los programadores empezar tan pronto como sea posible. La llave para el éxito es cortos, pequeños cicos de revision de diseño (de codigo). Un programador puede usualmente descubrir debilidades del diseño, inconsistencias, y resolver problemas de programacion que facilmente saturan al diseñador. Un dificultoso constructo de programacion puede ser reparado facilmente si se le detecta un error temprano, pero despues que otros objetos empiecen a depender de este constructo, los problemas seran de lejos mas dificiles de reparar.

Diseño por capas.

Middleware que importa

Diseño efectivo uML

Casos de uso

Conociendo las necesidades de los usuarios finales.

Sin embargo Antes de trasladar las abstracciones de una aplicación de servidor a codigo es necesario que antes que programemos establezcamos un framework general que pueda mantener las interfaces del servidor, objetos de negocio y los objetos de persistencia que implemente el servidor.

La ventaja de usar una arquitectura en capas es que cada capa puede ser examidad en si misma, vista independientemente.

Pero antes de que el diseño pueda ser complatado tu necesitas una fase de integracion, poniendo las capas de vuelta juntas. Tu debes diseñar un framework que permita al servicio de interfaz localizar los objetos de negocio y determinar cuando cargar y cuando cagar objetos de las bases de dato a la memoria y devolverlos. Este es un trabajo de diseño, datallado, un diseño de bajo nivel que es altamente dependiente en la arquitectura, middleware, y elecciones del lenguaje de programacion. Cada uno de estos tiene un fuerte impacto en como el framework es creado. En algunos casos el framework proviene como parte un midleware asi que hay un pequeño diseño del framework. En otros casos la complejidad del ambiente de operación y las limitaciones del lenguaje de programacion puede dictar una solucion unica, personalizada, hecha en la empresa.

Para iniciar el servidor de la aplicación debe haber un programa ejecutable que pueda lanzarse por la linea de comandos u otro programa que lo administre. Este programa es simplemente llamado el servidor de la aplicación desde que este contiene enlaces a cada objeto especificado en el diseño. Este tiene la responsabilidad de colocar el framework en memoria.

En la mayoria de los casos el middleware y el lenguaje de programacion dictaran el tipo de framework necesario y que tanto sera construido o comprado. La mayoria de los servidores de aplicación comerciales proveen de un framework de aplicaciones robusto pero requerira objetos que se adapten a estandares rigidos del componente. Otros productos de middleware como CORBA requeriran que el equipo de desarrollo cree su propio framework. Esto permite mas flexibilidad pero requiere mas trabajo departe de los desarrolladores.

La eleccion depende del tamaño y el contexto del proyecto y la cantidad de flexibilidad necesitada. Proyectos simples pueden solo necesitar un simple framework que vaya creciendo en la empresa. Proyectos mas largos probablemente necesitaran las capacidades de escalabilidad y administracion de un framework comercial. Proyecto de escala empresarial necesitar un hibrido, usando framework comercial integrada junto a una logica personalizada.

Cuando evaluamos aplicaciones de framework de servidor o cuando diseñamos una para apoyar nuestro proyecto de aplicación de serivodr los siguientes requerimientos deben ser considerados. Este framework sera el fundamente para la aplicación del servidor a desarrollar, una vez implementada sera dificil de cambiarla. Estas funciones son:

Escalabilidad

Concurrencia

Seguridad

Tolerancia a la falla

Para finalizar otro framework que incluso es mas importante que el programa de framework, es el framework que involucra la estructura organización que apoya el desarrollo de la aplicación de servidor. La mayoria de las organizaciones tiene solidas estrategias de desarrollo de software de forma que pongan las estructuras de su organización, sus metodologias y una suite de herramientas para desarrollo de software tradcional. Estas estrategias han evolucionado con el timpo y se han modifivcado a traves de la necesidades de la cultura de la organización. Cuando consideres estrategias para el desarrollo de software multi-tiered, tu debes examinar las estrategias de desarrollo y revisa cuales se adaptan alas nuevas necesidad, aunque usualmente no es necesario hacer cambios radicales. Tu puedes adoptar neuvas metolodigas para direccionar el desarrollo orientado a objetos y comprar nuevas herramientas para apoyar el middleware o la arquitectura de los componentes, pero los procesos de desarrollo aun deben relacionarse con las necesidades y cultura de tu organización.

Las siguientes curstiones ye standares deberian ser considerados cunado organizes los procesos de desarrollo de la aplicación de servidor.

Soporte de comunicación

Ambiente de desarrollo

Herramientas

Capacitacion

Metricas.

Herramientas:

Manejo del proyecto

Herramientas de modelado

Controlador de versiones.

Lenguajes de programacion y herramientas

Herramientas para desarrollo rapido de aplicaicones

Debuggers

Herramiento de testeo

Bug tracking y reporting

Como no es una arquitectura compleja, podemos usar el shared hosting compartido que tiene contratado la empresa. Este shared hosting llamado tal usa PHP, Mysql, Apache y Linux, y llegado el caso que las necesidades del sitio web y el negocio evolucione y se vuelva mas complejo se puede mover con facilidad a otras infraestructuras mas aplicables como Vps.

Necesitaremos un tipo de middleware del tipo CMS (sistema manejador de contenidos) para poder tener una web social.

La opcion que se selecciono fue drupal porque facilmente cumple con el middleware y el lenguaje de programacion necesitado. Ademas es un caso de éxito en la DGTIC, estado de Tabasco, pues se usa como framewrok para los sitios web del hospital de alta especialidad juan graham, y muchos otros tanto de hospitales como paginas de gobierno.

Ahora bien analizaremos cuales son las ventajas de Drupal para que haya sido elegido por la DGTIC.

**-IMPLEMENTANDO UN FRAMEWORK DE APLICACIÓN DE SERVIDOR**

Conectar la discusion entre como pasar del diseño a la programacion, describiendo como establecer este framework.

En adicion al programa de framework, debemos crear un framework organizacional que gestione y estructure el proceso de desarrollo. Esto incluye canales de comunicación, herramientas de programacion, estrategias de pruebas y otros detalles administrativos.

Integrar capas.

Inicializar el framework.

Elijiendo una strategia de framework

Estrategias de desarrollo.

**-QUE ES DRUPAL**

Drupal es un gran sistema manejador de contenidos, un poderoso framework para aplicaciones web, y una de las mejores plataformas sociales para publicaciones. Sobre eso, Drupal es mas que software- una vibrante comunidad de desarrolladores, diseñadores, managers de proyectos, innovadores de negocio, estrategas de tecnologia, profesionales de la experiencia del usuario, estandares y accesabilidad.

Drupal es un CMS para construir sitios web dinamicos.

Con drupal tu obtienes todas las caracteristicas de un poderosos sistema manejador de contenido, o Cms –login de usuario y registro; definicion de tipos de usuarios y contenido; diferentes niveles de permisos; creacion de contenido, editar, categorizar, y manejar; delegacion y agregacion fuera de la caja metaforica. Y en adicion a la funcionalidad basica, hay expandiendose un universo de funcionalidad adicional disponible del flujo alzante de las contribuciones de la comunidad.

Hay una gran cantidad de extensiones disponibles para drupal a traves de modulos contribuidos. De temas para mejorar el diseño del sitio a herramientas de lineas de comando, caracteristicas de busqueda, drupal es una comunidad de participacion en comunidad y hasta para contribuir al ecosistema tu mismo.

Drupal esta escrito en PHP con una gran parte de JavaScript (especialmente usando la librería Jquery) para la experiencia front-end, y usa bases de dato como MariaDB / Mysql o PostgreSQL para guardar ambos contenidos y configuracion. Por supuesto que puedes hacer suficiente codigo con esto u otros lenguajes de programacion y bases de datos, un desarrollador puede hacer cualquier cosa que un sitio de drupal puede hacer. Pero porque? Usar drupal te guarda de reinventar la rueda, permitiendote concentrarte en alcanzar tus objetivos. Drupal te lleva a donde vayas, sin que tengas que inventar un carro primero.

Drupal es un framework de aplicaicones.

Drupal se ha vuelto solido en su nucleo, tan poderoso y extensible para construir diferentes tipos de sitios web que es mas que un CMS: es una plataforma para desarrollar aplicaciones web serias. Cada actualizacion mayor incluye mejores APIs (Interfaces para programar la aplicación, como el codigo habla al codigo) y otras poderosas caracteristicas que sobrepasan ser un CMS.

Drupal es usada en una base de diferentes tipos de aplicaicones desde smart phone y aplicaciones de facebook hasta sitios web con una compleja logica de negocio (nysenate.gov/mobile, data.gov.uk, zagat.com) a social media y renta de software como servicio (buzzr.com). Drupal puede incluso encontrarse en roles que no son del CMS como el front para aplicaicones basadas en Java y el back end para front ends dirigidos por Ajax o Flash.

Donde se distingue entre framework y CMS u otros productos que puedan significar mas para el desarrollo es su creciente cantidad de distribuciones construido en Drupal para resolve especificos casos de uso. Ejemplos incluyen openatrium.com para equipos de intranet, Drupal Commons para negocios sociales, Openpublish para publicadores online, y OpenScholar para academicos personales y web sites de investigacion.

Drupal es una plataforma web social y semantica

El idea de web semantica y social atraviesa una vision para un futuro donde la informacion no esta atrapado en una sola web o compañía. En vez tu informaicon y eso que algunos tros comparten puede estar bajo tu control y disponible a traves de una cantidad multiple de plataformas y dispositivos. Sitios trabajando juntos ofrecen una manera distopica del mundo donde el control de las conexiones entre las personas y los datos es todo o nada. Drupal y su soporte para RDF (Framework de descripcion de recursos) ayudan a hacer este futuro posible.

RDF ayuda a etiquetar la informacion en una forma que las computadoras universalmente entienda, de tal manera que uedan hacer inteligentes cosas con la informacion de diversas fuentes. Al construir herramientas directamente en drupal que hagan facil compartir data estructurada, estamos ayudando a la Web Semantica, la edad de los datos conectados, cuando los sitios web y otros dispositivos conectados a inernet pueden resolver automaticamente complicadas preguntas basadas en la data compartirda por toda la internet.

Otra razon por la que elegir drupal es la larga cantidad de principiantes y expertos con un increible arreglo de contribuciones para llevar el desarrollo adelante. No muchos software libres pueden decir eso, y por supuesto que los productos propietarios menos pueden decir esto. De esta menra muchas compañía que usan Drupal estan creciendo junto con Drupal, no dejando la escena.

Incluso es una comunidad a nivel de masas, pues organizan las convenciones llamadas DrupalCon.

La razon numero uno para usar Drupal no es la funcionalidad, la extensibilidad, el poder, la flexibilidad, o cualquier otra cosa relacionada con el codigo. La razon numero uno para usar Drupal esta relacionada con la personas, en el respiro y lo profundo de su comunidad.

### III. ENTORNO DE TRABAJO

### 1. Sistema operativo: Lubuntu

Lubuntu es una distribución oficial del proyecto Ubuntu que tiene por lema "menos recursos y más eficiencia energética", usando el gestor de escritorio LXDE. El nombre Lubuntu es una combinación entre LXDE y Ubuntu.

El gestor LXDE usa el administrador de ventanas Openbox e intenta ser un sistema operativo que demande pocos recursos de RAM, CPU y otros componentes, especialmente ideados para equipos portátiles de recursos limitados como netbooks, dispositivos móviles y computadores antiguos. Con este ideal y esquema de desarrollo que posee, se presenta como una competencia a Xubuntu.

Características:

Los requerimientos de hardware de Lubuntu/LXDE son similares a los de

Windows XP. Un viejo CPU Pentium II o III entre 400-500 MHz es suficiente.

Una vez iniciado X.org y LXDE, el uso total de memoria aproximado es de 60 MB en máquinas i386. Este valor puede ser más alto o más bajo dependiendo de las diferentes configuraciones de sistema.

Aunque Lubuntu/LXDE no requiere hardware sofisticado, otras aplicaciones bajo

X.org sí pueden necesitarlo. Por ejemplo, Firefox y OpenOffice consumen niveles relativamente más altos de memoria. Así que es recomendable tener mínimo entre

192-256 MB de memoria RAM o más.

### 2. Editor de texto: SublimeText2

Multiplataforma. Sublime Text es un editor de texto y editor de código fuente creado en Python desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode. Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto,se puede obtener una licencia para su uso ilimitado, pero el no disponer de esta no genera ninguna limitación más allá de una alerta cada cierto tiempo.

### 3. Sistema controlador de versiones: GIT

El diseño de Git resulta de la experiencia del diseñador de Linux, Linus Torvalds, manteniendo una enorme cantidad de código distribuida y gestionada por mucha gente, que incide en numerosos detalles de rendimiento, y de la necesidad de rapidez en una primera implementación.

Fuerte apoyo al desarrollo no-lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no-lineal. Una presunción fundamental en Git es que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente, conforme se pasa entre varios programadores que lo revisan.

Gestión distribuida. Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.

Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de

cifrado SSH. Git también puede emular servidores CVS, lo que habilita el uso de clientes CVS pre-existentes y módulos IDE para CVS pre-existentes en el acceso de

repositorios Git.

### 4. Computación en la nube: Heroku

Heroku es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación.

Heroku es propiedad de Salesforce.com.Heroku, es una de las primeras plataformas de computación en la nube, que fue desarrollada desde junio de 2007, con el objetivo de soportar solamente el lenguaje de programación Ruby, pero posteriormente se ha extendido el soporte a Java, Node.js, Scala, Clojure y Python y (no documentado)PHP. La base del sistema operativo es Debian o, en la nueva plataforma, el sistema basado en Debian Ubuntu.

James Lindenbaum, Adam Wiggins, y Orion Henry fundaron Heroku para dar soporte a proyectos compatibles con Rack.El 8 de diciembre de 2010

Salesforce.com adquirió Heroku como una subsidiaria. El 12 de julio de

2011 Yukihiro "Matz" Matsumoto, el creador de Ruby se unió a la empresa como Arquitecto Jefe. Ese mismo mes, Heroku incorporó el soporte para Node.js y Clojure. El 15 de septiembre de 2011 Heroku y Facebook presentaron Heroku para Facebook

Actualmente Heroku soporta Cloudant, Couchbase Server, MongoDB y Redis, además de la conocida PostgreSQL, tanto como parte de la plataforma o como servicio independente.

Las aplicaciones se corren desde un servidor Heroku usando Heroku DNS Server para apuntar al dominio de la aplicación (típicamentenombreaplicacion.herokuapp.com). Cada aplicación corre sobre un motor a traves de una "red de bancos de prueba" que consta de varios servidores. El servidor Git de Heroku maneja los repositorios de las aplicaciones que son subidas por los usuarios.

### 2.6 TERMINO-DEFINICION

Framework: La palabra inglesa "framework" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar

HTML: siglas del inglés HyperText Markup Language, "lenguaje de etiquetado de hipertexto": es el lenguaje que se utiliza para crear las páginas Web, un lenguaje muy sencillo que permite combinar gráficos, textos y enlaces

Http: siglas del inglés HyperText Transfer Protocol, "protocolo de transferencia de hipertexto": es el protocolo de comunicación que utiliza la WWW.

Interfaz: sistema de comunicación de un programa con su usuario; la interfaz comprende las pantallas y los elementos que informan al usuario sobre lo que puede hacer, o sobre lo que está ocurriendo.

Internet: conjunto de ordenadores, o servidores, conectados en una red de redes mundial, que comparten un mismo protocolo de comunicación, y que prestan servicio a los ordenadores que se conectan a esa red; debe decirse siempre "la Internet".

Link: palabra inglesa para enlace; hay quien usa incluso linkar, porque no se ha enterado de que existe "enlazar"

Sitio Web: traducción del inglés Web site, conjunto de páginas de una institución o persona; también se dice sólo sitio: ("he visitado el sitio de Sí, Spain", "¿has estado en mi sitio?")

W3: forma abreviada de referirse a la World Wide Web

WWW: forma abreviada de referirse a la World Wide Web

Web (siempre en femenino: "la Web") 1 forma abreviada de referirse a la World Wide Web. 2...hay quien usa esta palabra --bastante horrendamente-- como abreviatura de "página Web", por ejemplo: "¿te gusta mi web?"

World Wide Web: (en español "Telaraña Mundial") interfaz de comunicación en la Internet, que hace uso de enlaces de hipertexto en el interior de una misma página, o entre distintas páginas.

CMS: Un sistema de gestión de contenidos (o CMS, del inglés Content Management System) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás roles.

Git: Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

Petición GET: Un servidor web opera mediante el protocolo HTTP, de la capa de aplicación del Modelo OSI. Al protocolo HTTP se le asigna habitualmente el puerto TCP 80. Las peticiones al servidor suelen realizarse mediante HTTP utilizando el método de petición GET en el que el recurso se solicita a través de la url al servidor web.

Por ejemplo para: GET /index.html HTTP/1.1 HOST: [www.host.co](http://www.host.com)m. En la barra de URL de un navegador cualquiera la petición anterior sería análoga a la siguiente dirección Web: [www.host.com/index.](http://www.host.com/index.html)html

Petición POST: Es el segundo tipo de petición HTTP más utilizado. Los datos a enviar al servidor se incluyen en el cuerpo de la misma petición con las cabeceras HTTP asignadas correspondientemente respecto al tipo de petición. Generalmente se asocia con los formularios web en el que los datos suelen ser cifrados para enviarlos de manera segura al servidor.

El navegador recopila la información del formulario para crear la petición y enviarla. Las cabeceras las envía junto a la petición POST, y se recopilan en base a metatags definidos en el código, los automáticos del navegador y los Cookies. Es el navegador, también, el que codifica los datos si es necesario.

Petición WEB: El navegador por medio de la interfaz de usuario permite al usuario realizar una o varias peticiones web. La interfaz de usuario o entorno de usuario es el conjunto de elementos del navegador que permiten realizar la petición de forma activa. Una petición Web no sólo puede ser realizada mediante un navegador sino con cualquier herramienta habilitada para tal fin, como una consola de comandos Telnet.

# 3. PROYECTO DE INTERVENCION

## 3.1 ENUNCIADO DEL OBJETIVO

El sistema manejador de contenidos Drupal proporcionará un margen necesario para integrar eficazmente los requerimientos de la Inteligencia de Negocios en el sitio web, de esta manera el desarrollo generará valor para el Negocio.

## 3.2 OBJETIVO GENERAL

Desarrollar un sitio web para la empresa Hospital Privado Santa fe S.A. de C.V que genere valor continuamente al Negocio utilizando un sistema manejador de contenidos que se apegue a los requerimientos de la inteligencia de negocios

## 3.3 OBJETIVOS ESPECIFICOS

Investigación de proyectos similares que hayan tenido casos de éxito.

Análisis de requerimientos del sitio

Análisis de los módulos que cumplen las características que el sistema manejador de contenidos debe tener para facilitar el desarrollo del mismo.

Instalación y configuración básica de Drupal en un servidor de prueba.

Estructuración de un sistema Drupal mediante la inclusión de diversos módulos que contengan las características que cumplen el análisis de requerimientos.

Desarrollo e integración del contenido al sitio web.

Aplicación del diseño gráfico personalizado a la identidad corporativa y estética de la página.

Migración del sitio web del servidor de prueba al ambiente de producción

Retroalimentación e integración de últimos cambios y/o solicitudes mínimas por parte de esta.

Entrega final del proyecto, instrucciones de uso y claves, y documentación del análisis y el desarrollo del sitio web al Hospital Privado Santa Fe S.A. de C.V.

## 3.4 ANALISIS DE LOS FACTORES RESTRICTIVOS

El desarrollo de este proyecto representa una mejora considerable al sistema actual, que representa una mejora de la comunicación y retroalimentación entre el sitio web y los responsables de este.

Un factor que puede influir gravemente sobre el tiempo es que el hosting tenga los requerimientos necesarios para correr el sistema manejador de contenidos Drupal 7, y entonces nos atenemos a la espera de que ellos nos hagan la migración a un servidor que si tengan estos requerimientos.

De acuerdo a la competencia de mercado Drupal es una comunidad con muchos casos de éxito con muchos desarrolladores web en el mundo.

Uno de los motivos principales de este proyecto es que la misma empresa hospital privado santa fe está solicitando este proyecto y tiene nuevas adquisiciones, nuevas instalaciones, y servicios que quiere promocionar. El área necesita una publicidad de todo esto y cuenta ya con un dominio en renta, pero que no está actualizado con todo esto.

## 3.5 JUSTIFICACION

Los sistemas de manejo de contenido y las tecnologías de información ha ido expandiéndose en todas las aéreas de las empresas, debido a las necesidades y requerimientos de información que se ha incrementado, y el costo, por lo que los primeros sistemas que podían hacer algo para desarrollar web se han hecho de libre código o vendidos los servicios para no tener que volver a escribir el código, código que la multitud de tecnologías ha permitido reutilizar.

Los sistemas de información son útiles siempre y cuando los datos sean presentados a manera de reportes a los directivos que se encargan de evaluarlos.

Drupal maneja los datos de tal forma que tiene módulos para utilizar esos datos y convertirlos en información de forma resumida. Por lo anterior y el hecho de que le modelo para la información en la web de una empresa necesita de estas características surge la idea de este proyecto cuyas principales funciones serán:

-Llevar un historial de visitas tanto al portal como páginas específicas y al mismo tiempo llevar un control de las páginas específicas que visitan. Para así mostrar al departamento las páginas más vistas y derivar en mejores servicios y promoción.

Es importante decir que este software que se está aplicando a una empresa de medicina también puede implementarse en cualquier otro tipo de empresa donde se requiera un sistema personalizado o básico para manejar contenido, y que cuenta con miles de casos de éxito y de usuarios.

# 4. ESTRATEGIAS DE IMPLEMENTACION

## 4.1. ESTRATEGIAS PARA PRESENTACION DEL PROYECTO

El proyecto será presentado a la dirección del hospital privado santa fe, donde estará el dueño del hospital para evaluar las ventajas que da el software nuevo.

Las ventajas que tendrá este nuevo sistema es que tendrá mejor control el contenido que ya se dispone y una plataforma para la publicación y subir tipos específicos de contenido.

La dirección del hospital santa fe tendrá reporte mensual del uso de la pagina web que beneficio a la toma de decisiones posteriores.

Por lo que este sistema beneficiara acercando la tecnología de la información involucrándose a la inteligencia de negocio de la empresa y brindando servicio rápido y eficiente.

La implementación del proyecto tomara 3 meses, y la migración final se hará en horario nocturno para evitar que el público se vea afectado por el funcionamiento del sitio web. La capacitación de los responsables se hará en un día junto con sus opiniones, sugerencia y comentarios que puedan surgir durante su utilización.

Se realizara una presentación donde se resaltaran los beneficios que tendrá este sistema al ser implementado. También se contempla una presentación técnica para los usuarios finales y explicar el funcionamiento general.

## 4.2. ANALISIS INVOLUCRADOS

## 4.3. DIAGRAMA UML

## 4.4. REALIZACIONN DE LOS PASOS DEL PROYECTO

# 5. ADMINISTRACION DEL PROYECTO

## 5.1. PLAN DE ACCION

A continuación se presenta el cronograma de actividades junto con los responsables de cada actividad y los días estimados para el desarrollo de cada una.

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre de tarea | Duración | Comienzo | Fin |
| Definir los nuevos requerimientos | 1 día | lun 03/02/14 | lun 03/02/14 |
| Elegir IDE de desarrollo y lenguaje de Programacion | 1 día | mar 04/02/14 | mar 04/02/14 |
| Modelo | 1 día | mié 05/02/14 | mié 05/02/14 |
| Manejador de modelo | 1 día | jue 06/02/14 | jue 06/02/14 |
| Creacion de modelo | 1 día | vie 07/02/14 | vie 07/02/14 |
| Pruebas de modelo | 1 día | lun 10/02/14 | lun 10/02/14 |
| Diseño de interfaz | 1 día | mar 11/02/14 | mar 11/02/14 |
| Diseño de reportes | 1 día | mié 12/02/14 | mié 12/02/14 |
| Pruebas locales del sistema | 1 día | jue 13/02/14 | jue 13/02/14 |
| Deteccion y correcion de bugs | 1 día | vie 14/02/14 | vie 14/02/14 |
| Instalacion de base de datos en ambientes de producción | 1 día | lun 17/02/14 | lun 17/02/14 |
| Instalacion de la aplicación | 1 día | mar 18/02/14 | mar 18/02/14 |
| Pruebas por parte del usuario final | 1 día | mié 19/02/14 | mié 19/02/14 |
| Retroalimentacion | 1 día | jue 20/02/14 | jue 20/02/14 |
| Exposicion del sistema a la Directiva de la empresa | 1 día | vie 21/02/14 | vie 21/02/14 |
| Capacitacion a usuarios | 1 día | lun 24/02/14 | lun 24/02/14 |
| Entrega definitiva | 1 día | mar 25/02/14 | mar 25/02/14 |

## 5.2. DIAGRAMA DE GANTT

## 5.3. RECURSOS

En la siguiente tabla se detallan los recursos que se utilizarán durante el desarrollo del proyecto. Se contemplan diferentes especialistas para el diseño de la base de datos, análisis y programación del software.

RECURSOS NECESARIOS PARA ELABORACION DEL PROYECTO

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Cantidad** | **Precio** |
| Paquete renta Telcel | 2 rentas por mes | $400.00/mes |
| Internet Banda ancha de 2GB descarga con línea de teléfono local Telmex | 1 renta por mes | $450.00/mes |
| Responsable de proyecto |  | $180.00/hr |
| Analista |  | $90.00/hr |
| Lider de proyecto |  | $120.00/hr |
| Desarrollador web |  | $90.00/hr |
| Diseñador Grafico |  | $90.00/hr |
| Energia eléctrica |  | $600.00/mes |
| Laptop  Windows 7, Intel Core 2 Duo 1.5GHz, 2Gb RAM | 2 | $10,000.00 |
| Desktop para Servidor  Windows 7, Intel Core 2 Duo 1.5GHz, 2Gb RAM | 1 | $9,500.00 |
| Impresora Laser B/N | 1 | $1,050.00 |
| Herramientas y Papeleria en Gral. |  | $1,000.00 |

## 5.4. PRESUPUESTO.

En la tabla siguiente se detallan los costos que tendrá el proyecto, estimando el pago de los especialistas con el tiempo considerado del proyecto.

COSTO DE RECURSOS DEL PROYECTO

|  |  |
| --- | --- |
| **Recurso** | **Costo** |
| Paquete renta Telcel | $2400.00 |
| Internet Banda ancha de 2GB descarga con línea de teléfono local Telmex | $1350.00 |
| Responsable de proyecto | **$52,800.00** |
| Analista | **$33,600.00** |
| Lider de proyecto | **$43,200.00** |
| Desarrollador web | **$33,600.00** |
| Diseñador Grafico | **$33,600.00** |
| Energia eléctrica | $1800.00 |
| Laptop  Windows 7, Intel Core 2 Duo 1.5GHz, 2Gb RAM | $20,000.00 |
| Desktop para Servidor  Windows 7, Intel Core 2 Duo 1.5GHz, 2Gb RAM | $9,500.00 |
| Impresora Laser B/N | $1,050.00 |
| Herramientas y Papeleria en Gral. | $1,000.00 |

## 5.4.1. COSTO DEL PROYECTO POR TAREA

En la siguiente tabla se detalla los costos estimados de cada tarea según los recursos utilizados en cada una.

COSTOS DE PROYECTOS POR ETAPA

|  |  |
| --- | --- |
| **Tarea** | **Costo** |
| 1 Analisis de los nuevos requerimientos |  |
| Definir los nuevos requerimientos |  |
| 2 Diseño del programa |  |
| Definicion modulos del sistema |  |
| Elegir IDE de desarrollo y lenguaje de Programacion |  |
| Diseño de ventanas del programa |  |
| Definir roles de usuario |  |
| 3 Diseño de Base de Datos |  |
| Modelo |  |
| Manejador de modelo |  |
| Creacion de modelo |  |
| Pruebas de modelo |  |
| 4 Desarrollo de la aplicación |  |
| Diseño de interfaz |  |
| Diseño de reportes |  |
| Pruebas locales del sistema |  |
| Deteccion y correcion de bugs |  |
| 5 Implementacion |  |
| Instalacion de base de datos en ambientes de producción |  |
| Instalacion de la aplicación |  |
| Pruebas por parte del usuario final |  |
| Retroalimentacion |  |
| 6 Cierre del proyecto |  |
| Exposicion del sistema a la Directiva de la empresa |  |
| Capacitacion a usuarios |  |
| Entrega definitiva |  |

# 6. CONCLUSIONES Y RECOMENDACIONES

Las conclusiones deben de responder a la pregunta **¿Qué resultados se encontraron y sus beneficios?**

Las conclusiones deben estar redactadas con: rigor lógico, claridad y concisión de estilo, originalidad, precisión, amplitud, compatibilidad con la ética, significancia y pertinencia.

## 6.1 Presentación de los resultados

Es una argumentación fundamentada con claridad y rigor científico, donde se dan a conocer los resultados dando respuesta a cada una de las preguntas u objetivos de la investigación.

## 6.2 Conclusiones y las recomendaciones

Es la argumentación lógica y fundamentada, dando a conocer los beneficios o alcances del proyecto de intervención general.

Las aplicaciones Web interactivas poco a poco han revolucionado la forma de utilizar internet, aumentando el contenido de las páginas con texto estático (texto que no evoluciona, sino que permanecen como es) a un contenido rico e interactivo, por lo tanto escalable.

Con los resultados de esta investigación se comprueba que una aplicación web que centraliza los recursos informativos que ellos usualmente toman de diferentes partes, y que además de esto corra bajo los últimos estándares de navegadores web, personalización y rapidez.

# Referencias Documentales

## Referencias Documentales

## Bibliografia complementaria

Babsoftware, web applications. (2011). Diseño y desarrollo de aplicaciones web. Recuperado el 30 de Octubre de 2013, de

<http://www.bab-soft.com/es/diseno_desarrollo_aplicaciones_web.php>

Barzanallana, Rafael (10 de Agosto de 2010). Universidad de Murcia. Recuperado el 30 de Octubre de 2013, de http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia- desarrollo-aplicaciones-web.html

Duplica Internet Solutions (20 de Septiembre 2010). Servidor web Recuperado el 30 de Octubre de 2013, de http://www.duplika.com/blog/que-son-los-servidores-web-y-por-que-son- necesarios

Sublime Text Unofficial Documentation (2012).

Sublime Text 2, Recuperado el 30 de Octubre de 2013, de

http://sublime-text-unofficial-documentation.readthedocs.org/en/sublime-text-2/

Wikipedia (8 de Noviembre de 2013). HTML Recuperado el 15 de Noviembre de 2013, de http://es.wikipedia.org/wiki/HTML

Wikipedia (30 Octubre de 2013). Framework Recuperado el 15 de Noviembre de 2013, de http://es.wikipedia.org/wiki/Framework

Wikipedia (7 de Noviembre de 2013).Ruby on Rails Recuperado el 15 de Noviembre de 2013, de http://es.wikipedia.org/wiki/Ruby\_on\_Rails

Wikipedia (3 de Noviembre de 2013). Lubuntu Recuperado el 15 de Noviembre de 2013, de http://es.wikipedia.org/wiki/Lubuntu

Wikipedia (7 de Noviembre de 2013). Git Recuperado el 15 de Noviembre de 2013, de http://en.wikipedia.org/wiki/Git\_(software)

Wikipedia (4 de Septiembre de 2013). Heroku

Recuperado el 30 de Octubre de 2013, de http://es.wikipedia.org/wiki/Heroku

Wikipedia (15 de Septiembre de 2013). Refinery CMS Recuperado el 30 de Octubre de 2013, de http://en.wikipedia.org/wiki/Refinery\_CMS

# ANEXOS

## ESTRUCTURA DE BASE DE DATOS

## REQUERIMIENTOS DEL USUARIO

## PANTALLAS DEL SISTEMA