UNIVERSIDAD TECNOLÓGICA DE LEÓN CAMPUS ACÁMBARO

Ingeniería en Tecnologías de la Información y Comunicación

ACTIVIDAD 4

Planeación de las actividades de un proyecto.

PARA LA EMPRESA

MATERIA

Administración de Proyecto de T.I.

PROFESORA

Mayra Neli Rivera Pizaña

PRESENTA

|  |  |  |
| --- | --- | --- |
| 15000033 | Jonathan Jair Alfaro Sanchez | jonathan\_jair\_01@hotmail.com |
| 14001794 | Mary Carmen Crescencio Bernal | marylupe58@hotmail.com |
| 15000028 | Miguel Ángel Mandujano Barragán | miguel\_dark97@hotmail.com |
| 15002186 | Giovanni Misael Alfaro Sánchez | giovannimisael007@hotmail.com |

GRUPO

ITI-801

GENERACIÓN: 2017-2019

ACÁMBARO, GUANAJUATO. 28/02/2018

**ÍNDICE**

[1. ANTECEDENTES 1](#_Toc507601149)

[1.1. FILOSOFÍA EMPRESARIAL 1](#_Toc507601150)

[1.1.1. Visión 1](#_Toc507601151)

[1.1.2. Misión 1](#_Toc507601152)

[1.2. ORGANIGRAMA 2](#_Toc507601153)

[1.3. ANTECEDENTES DEL PROYECTO 2](#_Toc507601154)

[2. PROBLEMÁTICA 3](#_Toc507601155)

[3. JUSTIFICACIÓN 4](#_Toc507601156)

[4. OBJETIVOS 5](#_Toc507601157)

[4.1. OBJETIVO GENERAL 5](#_Toc507601161)

[4.2. OBJETIVOS METODOLÓGICOS 5](#_Toc507601162)

[4.3. ALCANCE 6](#_Toc507601163)

[5. MARCO TEÓRICO 8](#_Toc507601164)

[5.1. CICLOS DE VIDA DE SOFTWARE 8](#_Toc507601166)

[5.2. HERRAMIENTAS DE DESGLOSE DE TRABAJO (EDT) 14](#_Toc507601167)

[5.3. TÉCNICAS PARA RECOLECCIÓN DE INFORMACIÓN 19](#_Toc507601168)

[5.4. ESPECIFICACIÓN DE REQUERIMIENTOS 23](#_Toc507601169)

[5.4.1. Definición 23](#_Toc507601170)

[5.4.2. Requerimientos funcionales 25](#_Toc507601171)

[5.4.3. Requerimientos no funcionales 25](#_Toc507601172)

[5.5. HERRAMIENTAS DE DISEÑO 26](#_Toc507601173)

[5.5.1. Software para modelado UML 26](#_Toc507601174)

[5.5.2. Software de diseño 31](#_Toc507601175)

[5.5.3. Software para maquetado 36](#_Toc507601176)

[5.6. HERRAMIENTAS DE PROGRAMACIÓN 40](#_Toc507601177)

[5.6.1. Lenguajes de programación 40](#_Toc507601178)

[5.6.2. Entornos de desarrollo 52](#_Toc507601179)

[5.6.3. Frameworks de desarrollo 57](#_Toc507601180)

[5.7. HERRAMIENTAS DE BASE DE DATOS 65](#_Toc507601181)

[5.7.1. Sistemas gestores de bases de datos 65](#_Toc507601182)

[5.7.2. Herramientas para gestionar Bases de Datos 71](#_Toc507601183)

[CONCLUSIONES 79](#_Toc507601184)

[BIBLIOGRAFÍAS 83](#_Toc507601185)

**INTRODUCCIÓN**

Este aparatado se redacta al final del Informe, porque se debe conocer todo lo que se desarrolló como proyecto de Estadía.Este apartado pretende describir brevemente: ¿Qué se hizo?, ¿cómo se hizo?, ¿qué se obtuvo? Es importante mencionar que NO debe ser una trascripción de los elementos mencionados, ni una copia resumida; tampoco se deberán enlistar resultados, ni presentar tablas, figuras o gráficas.

Por otro lado, y si fuese necesario, se puede mencionar aquello que estuvo fuera del alcance del trabajo, justificando el por qué.

La extensión de este apartado se recomiendo sea máximo de una cuartilla con una redacción en **tiempo pasado** y de manera impersonal., el título es de primer orden, por lo tanto debe ir centrado, con mayúsculas, Arial, negrita, en tamaño 14 y específicamente este título no lleva numeración.

Para el desarrollo del contenido debes consultar la Guía de Contenido (Palomares, 2014)

Algunos comentarios personales ayudan a que se desarrolle mejor la introducción, ya que se puede platicar sobre la evolución de la investigación, las experiencias que se obtuvieron durante la elaboración o bien si se desea indicar instrucciones o recomendaciones sobre la lectura de algunas partes del informe final de estadía como son: gráficos, tablas, imágenes, entre otros.

1. **ANTECEDENTES**

Empresa La Mexicana es un PYME orientada a la elaboración de salsas tradicionales y gourmet, estas son elaboradas con ingredientes naturales y frescos, la intensión del proyecto es destacar la gran variedad y particularidad de México destacando en las salsas, la sede es en la ciudad de Acámbaro en el Estado de Guanajuato.

En México la mayoría de las personas acompañan su comida con salsas, esto genera una demanda alta en el producto, al ser un producto de consumo popular es fácil entrar a este mercado, aunque la competencia es variada y posicionada en cuestión de marcas, la ventaja del proyecto es que son salsas caseras, esto nos da una oportunidad para poder incursionar sin problemas, además la materia prima es de fácil acceso y con un precio accesible ya que al ser parte de la alimentación diaria de los mexicanos ésta se encuentra dentro de la canasta básica.

* 1. FILOSOFÍA EMPRESARIAL

Ofrecer la mejor atención a nuestros clientes, enfocándonos en cumplir la satisfacción de los mismos, cubriendo sus necesidades y cumpliendo sus deseos. Creando en ellos un ambiente de confort y confianza.

Visión

Ser una empresa líder en la fabricación y distribución de salsas y aderezos, logrando abastecer la mayor cantidad de nichos de mercado, satisfaciendo los gustos y necesidades los nuestros clientes.

Misión

Ofrecer salsas y aderezos al mercado nacional, que satisfagan las expectativas del cliente en precio y calidad, elaborados bajo estrictas normas de higiene y seguridad, generando además fuentes de empleo.

* 1. **ORGANIGRAMA**
  2. ANTECEDENTES DEL PROYECTO

Existe el precedente de un sitio web desarrollado por la misma empresa, pero dicho sitio era muy básico y no cubría todos los requerimientos que ahora la empresa tiene.

1. PROBLEMÁTICA

En la ciudad de Acámbaro, Gto. se encuentra ubicada la empresa La Mexicana S.A. DE C.V. con la intención de desarrollar un proyecto para la misma se llevó a cabo una reunión el mes de febrero del 2018. En dicha reunión se observó que, al ser una empresa que surgió hace poco tiempo, no cuentan con una herramienta tecnológica que les permita dar mayor publicidad y difusión a sus productos y servicios.

Actualmente la empresa no cuenta con medios publicitarios ya que surgió hace poco tiempo, pero pretenden implementar medios y estrategias de publicidad ya conocidas como spots de radio, volanteo y material pop en busca de mayor difusión y alcance.

Al contar solo con los medios convencionales de publicidad como folletos, volates, trípticos, etc. tiene la limitante de cubrir un menor alcance geográfico, además de que requiere contar con personal para distribuirlos en el área que se quiere cubrir. Además de todo esto, sabemos que elaborar medios impresos conlleva un mayor tiempo de elaboración y costos por su impresión.

Los productos relacionados con la comida difícilmente tienen auge en el Internet, sin embargo, es una gran oportunidad que le dará un valor agregado a sus servicios, a su publicidad para aumentar los posibles clientes y reducirá costos de la misma, siempre que se le dé el enfoque adecuado, tomando como la gran ventaja de brindar la comodidad al público de conocer los productos que la empresa ofrece, sin la necesidad de ellos se tengan que trasladar y dejando a su disposición la información sin distinción de horarios.

1. JUSTIFICACIÓN

El ritmo de los avances tecnológicos requiere de herramientas modernas, comunicaciones eficientes, actualización permanente de información, estrategias y metodologías. La velocidad de comunicación que ofrece hace que sea un medio ideal para cubrir esas necesidades. Internet brinda la capacidad de estar siempre disponible en cualquier lugar y en cualquier momento.

Entre los beneficios que un sitio web podría darle a la empresa podemos destacar los siguientes:

* Permite aumentar el número de posibles clientes de su empresa.
* Un sitio es la publicidad más económica en relación al tiempo de publicación, 365 días del año, las 24 hrs del día.
* Brinda comodidad al público pues no tienen que trasladarse a ningún lugar para conocer de su producto.
* Le permite ahorrar en publicidad, evitando tener que reimprimir o rehacer, como sería el caso con guías y folletos impresos.
* La información publicada en Internet es instantánea.
* Da la posibilidad de ampliar su mercado desde un nivel local hasta un nivel nacional o internacional.
* Puede dar a conocer y difundir la dirección de su empresa, oficinas, locales o tiendas, sus números de teléfono, fax y su dirección de correo electrónico, para que sus clientes lo contacten fácilmente.

De no tomarse en cuenta la propuesta presentada en este proyecto a un mediano plazo, la empresa no lograra un mejor posicionamiento, además de seguir invirtiendo más en los medios convencionales de publicidad.

1. OBJETIVOS

4. 1. OBJETIVO GENERAL

Desarrollar un sitio web para la empresa La Mexicana S.A. de C.V. ubicada en Acámbaro Guanajuato durante el periodo Febrero-Abril del 2018, con el objeto de brindar un medio que sirva para dar a conocer su empresa, así como los productos y servicios que ofrecen.

* 1. OBJETIVOS METODOLÓGICOS

Seleccionar un ciclo de vida de desarrollo de software para el desarrollo del proyecto.

Planear las actividades para el desarrollo del proyecto utilizando una herramienta de desglose de trabajo.

Monitorear las actividades, el presupuesto y los tiempos a lo largo del desarrollo del proyecto.

Definir un método de recolección de datos.

Recolectar la información para la construcción del proyecto aplicando el método definido.

Analizar la información obtenida de la entrevista en conjunto con los interesados en el proyecto.

Determinar los requerimientos funcionales y no funcionales del proyecto utilizando la ingeniería de requerimientos.

Diseñar los diagramas UML para la construcción del proyecto utilizando software de modelado UML.

Diseñar la estructura de la Base de Datos utilizando software de diagramado.

Diseñar los bocetos necesarios para la construcción de sitio web utilizando software de maquetado.

Codificar el proyecto mediante un entorno de desarrollo utilizando un lenguaje de programación WEB en conjunto con un gestor de base de datos.

Realizar las pruebas al sitio web utilizando pruebas de software.

Implementar el sitio web en un hosting gratuito.

* 1. ALCANCE

EL sitio WEB que será desarrollado durante un periodo de 4 meses con fecha de terminación y entrega el 5 de abril del 2018 para la empresa “La Mexicana”, empresa orientada a la elaboración de salsas tradicionales y gourmet, deberá ser completamente funcional cumpliendo con los requerimientos funcionales y no funcionales obtenidos por medio de un método de recolección de datos aplicado a los clientes durante las reuniones de trabajo realizadas 2 veces por mes en la Unidad Académica del Sureste del estado plantel Acámbaro, en las cuales, se llegó a un acuerdo por parte del equipo de trabajo, dicho acuerdo consiste en presentar avances, opiniones y recomendaciones acerca del proyecto en cada una de las reuniones.

Para determinar el desarrollo del proyecto desde su inicio hasta su final y garantizar el cumplimiento de los requerimientos acordados se elegirá un ciclo de vida de software apropiado al tiempo límite de entrega del proyecto, así mismo, para llevar el control de las actividades necesarias para completar satisfactoriamente el proyecto se elegirá una herramienta de desglose de trabajo en la cual se definirán las tareas, los tiempos, recursos necesarios y presupuestos de cada actividad, además, durante el desarrollo el equipo de trabajo deberá monitorear las tareas, presupuestos y tiempos para asegurar el éxito del proyecto.

Respecto a la construcción del proyecto, se elegirá un software de modelado UML para plasmar la relación e interacción entre el sitio WEB con el usuario final de acuerdo a los requerimientos funcionales, así como también un software para el maquetado de los bocetos de cada una de las páginas que deberá contener el sitio WEB, las cuales son: Inicio, productos, platillos, pedidos y conócenos. Cada una de las paginas mencionadas anteriormente deberá ser desarrollada de acuerdo a los colores, estilo, fuentes, imágenes, funciones y características específicas acordadas durante las reuniones con los clientes.

Para la construcción de la estructura de la base de datos clara, así como cada una de sus entidades y relaciones necesarias para la correcta funcionalidad del proyecto se deberá usar un software de diagramado de base de datos y posteriormente realizar la base de datos por medio de un SGBD.

Para el desarrollo de cada una de las páginas del sitio WEB se elegirá un entorno de desarrollo que pueda ser usado con las especificaciones de las máquinas de trabajo, así como los lenguajes de programación y frameworks que permitan en conjunto desarrollar la variedad de funciones suficientes y necesarias para cumplir con los requerimientos para el proyecto, además de otorgar las herramientas necesarias para desarrollar los módulos que contendrá el proyecto.

Se deberán realizar pruebas unitarias y pruebas de modulo para verificar y validar que cada uno de los módulos cumplan con su función, así como también probar su funcionalidad en conjunto.

El proyecto deberá alojarse en un hosting gratuito, el cual se elegirá de acuerdo a las especificaciones propias de cada uno, así como el más recomendado para el proyecto.

Finalmente, en la entrega del proyecto, se le deberá de proporcionar a los clientes capacitación acerca de cómo usar el sitio WEB para ser administrado correctamente.

1. MARCO TEÓRICO
2. 1. CICLOS DE VIDA DE SOFTWARE

**Modelo de cascada retroalimentado**

En Ingeniería de software el desarrollo en cascada retroalimentado es idéntico al modelo de cascada puro, pero con una característica importante, la cual es la retroalimentación, esta ofrece la oportunidad de realizar cambios o evoluciones durante el ciclo de vida del software, permitiendo retroceder de una etapa a la anterior o incluso poder saltar a otras anteriores si es requerido. (Rodriguez, 2012)

* **Etapas**
* **Análisis de los requisitos del software:** El proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.
* **Diseño:** El diseño del software se enfoca en cuatro atributos distintos del programa; la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.
* **Implementación:** El diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
* **Pruebas:** Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.
* **Mantenimiento:** El software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se haya encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o a que el cliente requiera ampliaciones funcionales o del rendimiento.

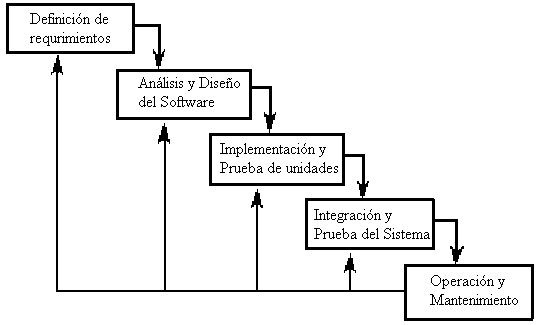


Figura 1 Modelo de cascada retroalimentado

* **Ventajas**
* Permite la departamentalización y control de gestión.
* El horario se establece con los plazos normalmente adecuados para cada etapa de desarrollo.
* Este proceso conduce a entregar el proyecto a tiempo.
* Es sencillo y facilita la gestión de proyectos.
* Permite tener bajo control el proyecto.
* Limita la cantidad de interacción entre equipos que se produce durante el desarrollo.
* **Desventajas**
* Los resultados y mejoras no son vivibles progresivamente.
* Poco tiempo para corregir dallas de depuración complicada.
* Los cambios introducidos durante el desarrollo pueden confundir al equipo profesional en las etapas tempranas del proyecto.
* El proceso es lento y pesado.

**Modelo en espiral**

Es un modelo de proceso de software evolutivo. Tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por la de riesgo más asumible y se hace un ciclo de la espiral. En cada giro se construye un nuevo modelo del sistema completo. Mejor modelo para el desarrollo de grandes sistemas. No hay número definido de iteraciones. (PHP, 2009)

* **Etapas**
* **Planificar:** En esta etapa se realizan las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.
* **Determinar objetivos:** En esta etapa se determina los objetivos que se necesitan para el desarrollo del proyecto.
* **Evaluar riesgos:** En esta etapa se realizan las tareas requeridas para evaluar riesgos técnicos y de gestión.
* **Desarrollar y probar:** En esta etapa se realizan las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

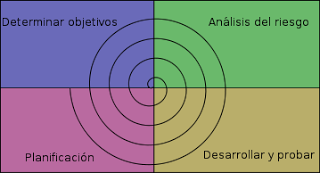


Figura 2 Ciclo de vida en espiral

* **Ventajas**
* Posibilidad adaptarse y aplicarse a lo largo de la vida del software.
* Provee de un enfoque realista del desarrollo del software.
* Permite aplicar el enfoque de construcción de prototipos en cualquier momento para reducir riegos.
* Reduce los riesgos antes de que se conviertan en problemáticos.
* Controla muy bien los riesgos y mientras más iteraciones se realicen, menos riesgos habrá.
* **Desventajas**
* Solo resulta aplicable para proyectos de gran tamaño.
* Resulta difícil convencer a los clientes de que el enfoque evolutivo es controlable
* Supone una carga de trabajo adicional, no presente en otros ciclos de vida.
* El modelo no se ha utilizado tanto como otros.
* Debido a su elevada complejidad no se aconseja utilizarlo en pequeños sistemas.

**Modelo de prototipos**

El modelo de prototipos permite que todo el sistema, o algunos de sus partes, se construyan rápidamente para comprender con facilidad y aclarar ciertos aspectos en los que se aseguren que el desarrollador, el usuario, el cliente estén de acuerdo en lo que se necesita así como también la solución que se propone para dicha necesidad y de esta forma minimizar el riesgo y la incertidumbre en el desarrollo, este modelo se encarga del desarrollo de diseños para que estos sean analizados y prescindir de ellos a medida que se adhieran nuevas especificaciones, es ideal para medir el alcance del producto, pero no se asegura su uso real. (María, s.f.)

* **Etapas**
* **Investigación preliminar:** En esta etapa lo esencial es determinar el problema y su ámbito, la importancia y los efectos potenciales que tendrán sobre la organización, identificar una idea general de la solución para realizar un estudio de factibilidad que determine la factibilidad de una solución software.
* **Definición de los requerimientos del sistema:** Esta es la fase más importante de todo el ciclo de vida del método de prototipos, el objetivo en esta fase es determinar todos los requerimientos y deseos que los usuarios tienen en relación con el proyecto que se está deseando implementar.
* **Diseño técnico:** En esta etapa el sistema debe ser rediseñado y tener la respectiva documentación guiándose en los estándares que tiene la organización la cual servirá como ayuda en mantenciones futuras del mismo.
* **Programación y prueba:** En esta etapa es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos. Las pruebas serán de realizarse tantas veces sea necesarias para verificar cualquier tipo de anomalía en el sistema.
* **Operación y mantención:** En esta fase se realiza ya la instalación y mantención del software, la complejidad en este caso resulta menor ya que en las etapas anteriores los usuarios han trabajado con el sistema al momento de hacer las pruebas de prototipos, además la mantención también debería ser una fase menos importante, ya que se supone que el refinamiento del prototipo permitiría una mejor claridad en los requerimientos.

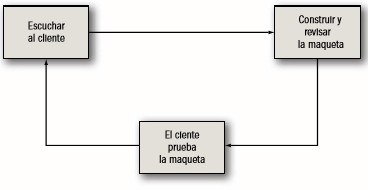


Figura 3 Modelo de prototipos

* **Ventajas**
* Es un excelente ciclo cuando existe incertidumbre en los requerimientos o soluciones de diseño.
* Es muy útil para el diseño de la interfaz de usuario final.
* Favorece a que los usuarios finales se involucren bastante a lo largo del proceso de desarrollo del sistema.
* **Desventajas**
* Posibilidad de que el prototipo sea muy lento, muy grande, no muy amigable en uso, o incluso, que este escrito en lenguaje de programación inadecuado.
* Posibilidad de desilusión y desacuerdo de parte del cliente cuando ve funcionando por primera vez el prototipo que ha sido construido con las primeras herramientas.
  1. HERRAMIENTAS DE DESGLOSE DE TRABAJO (EDT)

**Microsoft Project**



Figura 4 Logotipo de Microsoft Project

* **Definición**: Microsoft Project es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. Simplifica el trabajo y ayuda a mantenerse organizado. (Microsoft)
* **Características:**
* Permite el aprendizaje rápido con el planeamiento y la administración guiados, organización y seguimiento del rendimiento.
* Posibilidad de generar informes predefinidos.
* Permite compartir planes de proyecto, colaboración entre grupos de trabajo.
* Presenta diagramas como: Diagrama de Gantt y Diagrama de Pert (diagrama de red).
* **Permite la integración de la ruta crítica:** Permite analizar las tareas y la secuencia en qué deben realizarse, para reconocer cuáles son esenciales y las relaciones de dependencia entre ellas.
* **Controlar la sobrecarga de recursos:** El programa controla la cantidad de recursos asignados a cada persona y señala si, en algún caso, éstos son excesivos.
* **Controlar cálculo de costos:** Calcula los costes de los recursos y la mano de obra, una vez que han sido distribuidos por tareas.
* **Muestra un resumen de proyecto:** Muestra la información clave del proyecto, como las fechas de inicio y fin, duración, horas totales de trabajo, costes, estado de las tareas y recursos necesarios.
* **Ventajas:**
* Mejorar la inteligencia empresarial.
* Colaborar y coordinar con facilidad.
* Administrar los recursos eficazmente.
* Administrar el ciclo de vida del proyecto.
* Mejorar constantemente los procesos.
* Recuperación real de la inversión.
* **Desventajas:**
* No se puede medir la productividad.
* Su precio es algo excesivo.
* No es un programa multiplataforma.

**CELOXIS**



Figura 5 Logotipo de Celoxis

* **Definición:** Celoxis es una plataforma completa online para la gestión de proyectos y la gestión de equipos de trabajo. Ayuda a los responsables a realizar el seguimiento del proyecto, los recursos, el presupuesto, el beneficio, los riesgos, y las solicitudes y aprobaciones de cambios. Adicionalmente, su capacidad para generar informes ayuda a los directores de proyectos a tomar decisiones efectivas basadas en datos objetivos.
* **Características:**
* Es una plataforma online para la gestión de proyectos, por lo cual necesita de internet.
* Posibilidad de generar informes de ayuda para los directores de un proyecto.
* Contiene múltiples herramientas para gestionar un proyecto a fondo.
* **Ventajas**
* Celoxis es un software que opera en línea, así que únicamente se necesita contar con acceso a internet para poder administrar los proyectos.
* Proporciona herramientas de colaboración como las carpetas de internet las cuales muestran los archivos adjuntos a cada actividad de manera clasificada.
* Incluye todas las herramientas necesarias para gestionar proyectos del mundo real como los gráficos de Gantt, sub tareas, dependencias etc., además cuenta con compatibilidad con MS Project.
* **Desventajas**
* Cuenta con soporte al cliente limitado.
* No permite que el entregable sea entregado el mismo día que es terminada la actividad, lo asigna al día siguiente de la actividad.

**WBS Chart Pro**

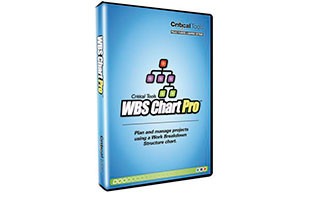


Figura 6 Logotipo de WBS Chart Pro

* **Definición:** WBS Chart Pro es un software basado en Windows para el manejo de proyectos que se utiliza para crear y visualizar proyectos utilizando diagramas WBS (Work Breakdown Structure). Un diagrama WBS visualiza la estructura de un proyecto mostrando cómo está organizado el proyecto en un resumen y varios niveles de detalles. Utilizar un diagrama WBS es un acercamiento más intuitivo para la planeación y visualización de su proyecto.
* **Características:**
* Posibilidad de crear diagramas WBS desde Microsoft Project.
* Muestra Visualizaciones de Alto Nivel o de Resumen de sus Proyectos.
* Permite visualizar secciones de su Plan.
* Impresión a Escala.
* Soporte MPX.
* Exportación como Página Web.
* Exportación como Imagen.
* Capacidad Ilimitada.
* Soporte a usuarios en otros idiomas.
* Compatible con Project KickStart.
* **Ventajas:**
* Los planes creados en WBS Chart Pro pueden ser administrados y manejados separadamente, transferidos directamente a Microsoft Project o a cualquier programa compatible con archivos MPX.
* Transferir un diagrama WBS a Microsoft Project en cualquier momento con un simple clic de un botón.
* Toda la información que ingrese a WBS Chart Pro será transmitida automáticamente a Microsoft Project.
* WBS Chart Pro puede funcionar integrado a Microsoft Project, usted no necesita tener Microsoft Project para utilizar WBS Chart Pro como aplicación única.
* **Desventajas:**
* El software actualmente se encuentra descontinuado.
* La interfaz es más difícil de utilizar.
* No hay suficiente información o documentación.
  1. TÉCNICAS PARA RECOLECCIÓN DE INFORMACIÓN

Entrevista

* **Definición:**

La entrevista es una conversación dirigida por el periodista, casi siempre con un fin predeterminado, que después se trasladará a los lectores u oyentes para su información u orientación; es un encuentro que no se suele dejar que transcurra al azar, pues el periodista busca llevar las riendas para obtener el máximo provecho de tal plática (Cantavella, 2017).

Las entrevistas se utilizan para recabar información en forma verbal, a través de preguntas que propone el analista. Quienes responden pueden ser gerentes o empleados, los cuales son usuarios actuales del sistema existente, usuarios potenciales del sistema propuesto o aquellos que proporcionarán datos o serán afectados por la aplicación propuesta.

Una entrevista para recabar información es una conversación dirigida con un propósito específico que utiliza un formato de preguntas y respuestas. Las metas son información importante que se pueden recabar de las entrevistas. (Kenneth E. Kendall , Julie E. Kendall, 2005)

* **Ventajas:**
* Las elaboraciones de preguntas son seguras para los que van a responder.
* Fácil de administrar y evaluar.
* Es muy compleja en la información dada a conocer.
* Se necesita un limitado tiempo de entrenamiento para el analista.
* Pueden que existan entrevistas más pequeñas y eficaces.
* Tienes la oportunidad de usar tus propias palabras para presentarte a un empleador potencial.
* **Desventajas:**
* Algunos solicitantes pueden sentirse incómodos durante las entrevistas personales, haciendo que se olviden de los puntos clave.
* Las entrevistas consumen una gran cantidad de tiempo, incluyendo el tiempo de preparación.
* Las entrevistas se basan en el juicio propio del entrevistador, y no garantizan el mejor resultado.

Observación

* **Definición:**

La observación es otra técnica útil para el analista en su proceso de investigación, consiste en observar a las personas cuando efectúan su trabajo. La observación es una técnica de observación de hechos durante la cual el analista participa activamente como espectador de las actividades llevadas a cabo por una persona para conocer mejor su sistema.

El propósito de la observación es múltiple, permite al analista determinar que se está haciendo, como se está haciendo, quien lo hace, cuando se lleva a cabo, cuánto tiempo toma, donde se hace y porque se hace.

* **Características:**
* **Tipos de observación:** El analista puede observar de tres maneras básicas:
* Puede observar a una persona o actividad sin que el observado se dé cuenta y sin interactuar por parte del propio analista.
* El analista puede observar una operación sin intervenir para nada, pero estando la persona observada enteramente consiente de la observación.
* Se puede observar y estar en contacto con las personas observadas. La interrogación puede consistir simplemente en preguntar respecto a una actividad específica, pedir una explicación, etc.
* **Pasos de la observación:**
* Determinar y definir aquello que se va a observar.
* Estimar el tiempo necesario de observación.
* Obtener la autorización para llevar a cabo la observación.
* Explicar a las personas que van a ser observadas lo que se va hacer y las razones para ello.
* **Ventajas:**
* Permite al analista determinar que se está haciendo, como se está haciendo, quien lo hace, cuando se lleva a cabo, cuánto tiempo toma, donde se hace y porque se hace.
* Ciertos tipos de datos sólo pueden obtenerse mediante la observación, incluyendo los patrones de conducta que el participante desconoce o no es capaz de comunicar.
* Permite la medición de conducta real.
* No hay un sesgo de informe, y el sesgo potencial causado por el entrevistador y el proceso de la entrevista se elimina o se reduce.
* **Desventajas:**
* Probable que las razones de la conducta observada no estén definidas
* Otra limitación de la observación es que la percepción selectiva (sesgo en la percepción del investigador) puede sesgar los datos
* La observación suele ser costosa y prolongada; y resulta difícil observar ciertas formas de conducta, como las actividades personales.
* En algunos casos el uso de técnicas de observación se vuelve poco ético, como sucede al vigilar el comportamiento de la gente sin su conocimiento o consentimiento.

Cuestionarios

* **Definición:** El uso de cuestionarios permite a los analistas reunir información proveniente de un grupo grande de personas.
* **Características:**

Los cuestionarios proporcionan una alternativa muy útil para la entrevista; sin embargo, existen ciertas características que pueden ser apropiada en algunas situaciones e inapropiadas en otra. Al igual que la entrevistas, deben diseñarse cuidadosamente para una máxima efectividad.

Existen dos formas de cuestionarios para recabar datos: cuestionarios abiertos y cerrados, y se aplican dependiendo de si los analistas conocen de antemano todas las posibles respuestas de las preguntas y pueden incluirlas. Con frecuencia se utilizan ambas formas en los estudios de sistemas.

* **Cuestionario Abierto:** El formato abierto proporciona una amplia oportunidad para quienes respondan escriba las razones de sus ideas. Algunas personas, sin embargo, encuentran más fácil escoger una de un conjunto de respuestas preparadas que pensar por sí mismas.
* **Cuestionario Cerrado:** El cuestionario cerrado limita las respuestas posibles del interrogado. Por medio de un cuidadoso estilo en la pregunta, el analista puede controlar el marco de referencia. Este formato es el método para obtener información sobre los hechos. También fuerza a los individuos para que tomen una posición y forma su opinión sobre los aspectos importantes.
* **Ventajas:**
* El empleo de formatos estandarizados para las preguntas.
* Puede proporcionar datos más confiables que otras técnicas.
* Su amplia distribución asegura el anonimato de los encuestados, situación que puede conducir a respuestas más honestas.
* **Desventajas:**
* El inconveniente es que la respuesta puede ser limitada, ya que es posible que no tenga mucha importancia para los encuestados llenar el cuestionario.
* Es recomendable conseguir apoyo de la alta dirección para solicitar a las personas de la organización que contesten el cuestionario.
* Falta de profundidad en las respuestas y no se pueden ir más allá del cuestionario.
  1. ESPECIFICACIÓN DE REQUERIMIENTOS

Definición

“Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas” (Sommerville, 2005).

Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema (o un sistema de software) para que tenga valor y utilidad para el usuario. En otras palabras, los requerimientos muestran qué elementos y funciones son necesarias para un proyecto.

* **Etapas:**
* **Obtención de requerimientos:** Búsqueda y obtención de los requerimientos desde los grupos de interés.
* **Análisis:** Comprobación de la consistencia y completitud de los requerimientos.
* **Verificación:** Constatación de que los requerimientos especificados son correctos.
* **Características:**
* **Actual:** El requerimiento no debe volverse obsoleto con el paso del tiempo.
* **Cohesión:** El requerimiento debe dirigirse a solo una única cosa
* **Completo:** El requerimiento debe estar completamente declarado en un único lugar, sin información faltante.
* **Consistente:** El requerimiento no debe contradecir ningún otro requerimiento y debe ser completamente consistente con toda la documentación.
* **Correcto/necesario:** El requerimiento debe cumplir con la necesidad declarada por los interesados en el sistema/software.
* **Factible/viable:** El requerimiento debe poder ser implementado.
* **No ambiguo:** El requerimiento debe estar concisamente declarado. Debe expresar hechos objetivos, no opiniones subjetivas. Debe poder ser interpretado de una única manera.
* **Obligatorio:** El requerimiento debe representar una característica definida por el grupo interesado en el desarrollo del sistema/software, su ausencia no puede ser reemplazada.
* **Observable externamente:** El requerimiento debe especificar una característica observable externa o experimentable por el usuario del producto.
* **Verificable/demostrable:** La implementación del requerimiento debe poder ser resuelta en alguno de estos cuatro métodos: inspección, análisis, demostración o prueba.

Requerimientos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005).

Requerimientos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Sommerville, 2005).

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como, por ejemplo, el rendimiento, interfaces de usuario, fiabilidad, mantenimiento, seguridad, portabilidad, estándares, flexibilidad, confiabilidad, estabilidad, recuperación de fallas, etc.

* 1. HERRAMIENTAS DE DISEÑO

Software para modelado UML

**StarUML**

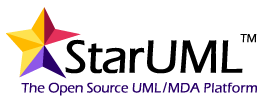


Figura 7 Logotipo de StarUML

* **Descripción:** Es una herramienta UML de licencia gratuita, desarrollada en 1996 y posteriormente en el 2005 modificada por la GLP para el modelamiento de software, basándose en estándares UML (Unified Modeling Language) y DMA (Model Driven Arquitecture). (Jotatsu, 2006)
* **Características:** A lo largo de su trayectoria, StarUML heredo las características de su versión comercial, así mismo ha ido mejorando sus características entre las cuales se encuentran:
* Soporte completo para la creación de diseño UML mediante el uso de:
* Diagrama de casos de uso.
* Diagrama de clase.
* Diagrama de secuencia.
* Diagrama de colaboración.
* Diagrama de estados.
* Diagrama de actividad.
* Diagrama de componentes.
* Diagrama de despliegue.
* Diagrama de composición estructural.
* Definir elementos propios para los diagramas, que no necesariamente son UML.
* Capacidad de generar código a partir de los diagramas y viceversa, los lenguajes que actualmente están funcionando son C++, C# y Java.
* Generar documentación en formatos: Word, Excel y PowerPoint.
* Generar plantillas de proyectos.
* Posibilidad de crear plugins para el programa.
* **Ventajas**
* Es un software con el más amplio catálogo de herramientas para el modelamiento UML.
* Mejoramiento continuo.
* Facilidad de creación de diagramas, debido a la simplicidad y rápida percepción de sus objetos, funciones y características.
* Es software libre.
* **Desventajas**
* Problemas al utilizar símbolos tildes.
* No es posible utilizarlo en Linux únicamente con la distro Wine.
* No es muy usado y mucho menos conocido

**Dia**

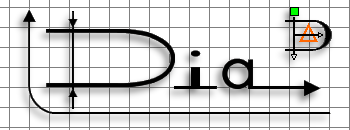


Figura 8 Logotipo de Dia

* **Descripción:** Dia es una aplicación informática de propósito general para crear diagramas y mapas conceptuales. Se trata de una aplicación de código abierto desarrollada como parte del Proyecto GNOME. Dia está diseñado como un sustituto de la aplicación comercial Visio de Microsoft. Se puede utilizar para dibujar diferentes tipos de diagramas, es un programa intuitivo y ligero, que pone el acento en la simplicidad y la eficiencia. Para utilizarlo no se requiere ni un ordenador potente ni grandes conocimientos técnicos. (McCann, 2013)
* **Características:**
* Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades.
* Posibilidad de creación de diferentes tipos de diagramas como:
* Diagramas entidad relación.
* Diagramas de UML.
* Diagramas de flujo.
* Diagramas de redes.
* Diagramas de circuitos electrónicos.
* Etc.
* Permite agregar diferentes componentes para diagramas desde archivos tipo SVG.
* Posibilidad de almacenar archivos en diferentes formatos como: EPS, SVG, PNG, JPG, entre otros.
* **Ventajas**
* Es software gratuito.
* Es multiplataforma.
* Posee una importante colección de imágenes y símbolos de inmediata aplicación.
* Posibilidad de exportar en diferentes formatos.
* **Desventajas**
* Carece de un diseño profesional en la interfaz.
* Carece de un diseño profesional en los diagramas.
* No existen tanta documentación sobre el software.

**ArgoUML**



Figura 9 Logotipo de ArgoUML

* **Descripción:** ArgoUML es una aplicación de diagramado de UML publicada bajo la licencia BSD (Berkeley Software Distribution), dado que la aplicación está hecha con el lenguaje de programación Java, está disponible en cualquier plataforma soportada por java.
* **Características:**
* Exportación de diagramas de diferentes formatos.
* Generación de código a partir de los diagramas.
* Soporte para bases de datos.
* Es necesario tener java instalado.
* Permite crear diagramas como:
* Diagrama de casos de uso.
* Diagrama de clases.
* Diagrama de secuencia.
* Diagrama de colaboración.
* Diagrama de estado.
* Diagrama de actividades.
* Diagrama de despliegue.
* **Ventajas**
* Soporte de múltiples idiomas como: inglés, alemán, francés, español y ruso.
* Es de código abierto.
* Interfaz de módulos extensible.
* La mayoría de las funciones soportan la selección múltiple de los elementos del modelo.
* **Desventajas**
* No existe la función de deshacer.
* En veces los modelos no pueden ser re-abiertos.
* Debes de crear un diagrama de clases, para crear algún otro diagrama.

Software de diseño

**Adobe Photoshop**



Figura 10 Logotipo de Photoshop

* **Descripción:** Adobe Photoshop es uno de los programas más famosos para la realización de tareas profesionales relacionadas con imágenes digitales como el retoque, diseño gráfico, impresión profesional y dirección artística. Photoshop fue desarrollado por Adobe Systems Incorporated. (Pascual González, 2004)
* **Características:** Como se sabe, Photoshop va mejorando su entorno continuamente, contando con más de 5 versiones, ha evolucionado considerablemente y ha incluido diversas mejoras fundamentales tales como:
* Espacio de trabajo multicapa.
* Inclusión de elementos vectoriales.
* Gestión avanzada de color
* Tratamiento extensivo de tipografías.
* Control y retoque de color.
* Efectos creativos.
* Exportación para sitio web
* Entre otros.
* **Ventajas**
* Posibilidad de utilizar la tecnología HDR.
* Cuenta con la función de correctivos automáticos.
* Flujo mayor de trabajo.
* Obtención de resultados profesionales.
* Soporte multiplataforma.
* **Desventajas**
* Únicamente está disponible para los sistemas operativos Windows y Mac.
* El costo podría ser muy excesivo.
* Es necesario un equipo con altas especificaciones.
* Es muy pesado.

**CorelDraw**



Figura 11 Logotipo de CorelDraw

* **Descripción:** Es una aplicación de diseño gráfico vectorial, poderosa, intuitiva y sencilla de usar, es la aplicación principal del paquete de aplicaciones CorelDraw Suite, está diseñada para cumplir múltiples necesidades como el dibujo, la maquetación de páginas para impresión o publicación web. (Haro, s.f.)
* **Características:** Corel draw utiliza variables matemáticas generalizadas en escalas de medición real (Centímetros, cíceros, milímetros, pulgadas, pixel, entre otros.).
* Permite manejar objetos de forma independiente (puntos, líneas, polilíneas, figuras geométricas, imágenes, texto, entre otros).
* Optimiza el almacenamiento de sus archivos gracias a su manejo de vectores.
* Posee productos relacionados de gran utilidad para los diseñadores como galerías de fuentes, gráficos prediseñados, plantillas entre otros.
* **Ventajas**
* Permite trabajar de manera sencilla con trazados, crear diseños bordados, grabados, estampados etc.
* Cuenta con soporte multi página.
* Posibilidad de importar y exportar documentos de Office hasta Freehand y cuanto formato haya.
* Posibilidad de adaptación del espacio de trabajo.
* **Desventajas**
* Contiene herramientas simples y características estáticas no aptos para codificar fotografías o videos.
* Es necesario un equipo con altas especificaciones.
* No será posible obtener una calidad de 100%.

**GIMP**



Figura 12 Logotipo de GIMP

* **Descripción:** GIMP es un software libre, apropiado para tareas de retoque fotográfico, composición y edición de imagen, comúnmente usado para la creación de logotipos y gráficos para páginas web.
* **Características:**
* **Dibujo:**
* Gran variedad herramientas de dibujo.
* Editor de gradientes.
* Soporta pinceles y diseños personalizados.
* **Manipulación avanzada:**
* Soporta canales alfa completos.
* Capas y canales.
* Múltiples deshacer/rehacer.
* Editar las etiquetas de las capas
* Herramientas de transformación (rotar, escalar, recortar).
* Herramientas de selección (rectángulo, elipse, libre, etc).
* Herramientas de extracción del fondo.
* Herramientas de selección avanzadas haciendo formas poligonales, por color.
* **Extensible:**
* Permite agregar plugins, scripts.
* **Animaciones:**
* Creación de animaciones GIF por medio de capas.
* **Ventajas**
* Es un software libre bajo la licencia GPL.
* Los plugins instalables a GIMP se pueden obtener gratis.
* Posibilidad de ser extensible a través de plugins.
* Disponible para múltiples plataformas y sistemas operativos.
* Interfaz sencilla e intuitiva.
* Ocupa poco almacenamiento en disco duro.
* **Desventajas**
* Contiene menos funcionalidades que otros software.
* Poco conocido.
* Carece de una interfaz gráfica profesional.

Software para maquetado

**Balsamiq mockups**



Figura 13 Logotipo de Balsamiq Mockups

* **Descripción:** Balsamiq Mockups es una aplicación para crear maquetas para interfaces gráficas para usuario. Permite 'pintar' los requerimientos del usuario de tal forma que le presentas bocetos que se acercan mucho a la realidad, contemplando explicaciones o aclaraciones sobre las distintas funcionalidades que se pueden dar en cada caso. (Hernandez, 2010)
* **Características:** Balsamiq Mockups cuenta con:
* **Tramas incompletas de baja fidelidad:** Las tramas incompletas, de baja fidelidad permiten centrar conversaciones de diseño en la funcionalidad.
* **Componentes de interfaz e iconos:** 75 componentes de interfaz de usuario y 187 iconos integrados, además de un montón de componentes generados por la comunidad.
* **Prototipos seleccionables:** La vinculación te permite generar prototipos seleccionables a través de demostraciones y pruebas de usabilidad.
* **Exportación a PNG o PDF:** Comparte o presenta maquetas con incrustación de vínculos mediante la exportación a PDF o utiliza una herramienta de terceros para exportar a código.
* **Adición rápida para velocidad:** Te permite construir prototipos utilizando tu teclado.
* **Bibliotecas de símbolos reutilizables:** Crea plantillas, bibliotecas de componentes reutilizables y maestras.
* **Importación y exportación:** Integración transparente con todas las versiones de maquetas, para cuando estés de nuevo en línea.
* **Posibilidad de trabajar sin conexión:** Trabaja en el avión o en una cafetería sin conexión a Internet.
* **Ventajas**
* Posibilidad de realizar proyectos con gran rapidez.
* Funciones de búsqueda rápida de componentes.
* Gran cantidad de herramientas para generar los bocetos.
* Interfaz gráfica amigable.
* **Desventajas**
* Algunas herramientas se quedan cortas en opciones.
* Falta de diseño en herramientas para bocetos.
* Para algunos clientes el diseño podría parecer infantil o poco profesional.

**Mockflow**



Figura 14 Logotipo de Mockflow

* **Descripción:** Mockflow es una herramienta que permite crear bocetos de la estructura de un sitio web, es decir los prototipos de las páginas que componen el sitio. Permite diseñar de una manera rápida e intuitiva los bocetos, implantando una arquitectura web muy personalizada y con múltiples complementos.
* **Características:**
* Permite crear y editar el sitio web maquetas de interfaz de usuario en línea.
* Posibilidad de compartir y colaborar maquetas con los demás.
* Es fácil de usar, tiene un diseño limpio, y cuenta con una interfaz de arrastrar y soltar interfaz.
* Posibilidad de exportación como una imagen PNG.
* **Ventajas**
* Los wireframes creados con MockFlow son interactivos, simulan la navegación del usuario a través de las páginas web.
* Dispone de una variada colección de elementos GUI.
* Posibilidad de exportación en: ppt, pdf y png.
* **Desventajas**
* Requiere una conexión a internet para poder trabajar.
* Las cuentas gratuitas de MockFlow únicamente permiten construir un wireframe con un máximo de cuatro páginas.

**HotGloo**



Figura 15 Logotipo de HotGloo

* **Descripción:** HotGloo es una herramienta de esquematización y prototipo UX (diseño de experiencia de usuarios) diseñada para construir diseños de páginas para internet, móviles y aparatos electrónicos adaptables al cuerpo (wearables).
* **Características:**
* Cuenta con una amplia librería de componentes a utilizar para desarrollar los diseños.
* Funciones de arrastrar y soltar los componentes a la mesa de trabajo.
* Contiene herramientas que facilitan la colaboración y revisión de proyectos.
* Cuenta con una interfaz intuitiva y completa.
* Es una herramienta de pago, pero tiene 15 días de prueba gratuitos.
* **Ventajas**
* **Fácil de usar:** HotGloo ha sido diseñado pensando en una curva de bajo aprendizaje. Ofrece tutoriales, documentación para comenzar y un confiable y rápido equipo de apoyo.
* **Celulares optimizados:** Edita, revisa y prueba el trabajo realizado en cualquier dispositivo móvil, cuando sea y donde sea.
* **Hecho para equipos:** Posibilidad de crear proyectos en equipo invitando a diferentes personas o compañeros.
* **Full-stack:** Más de 2000 elementos, iconos y herramientas es lo que hace a la biblioteca de elementos de HotGloo una de las más amigables del mercado.
* **Desventajas**
* Requiere de una conexión a internet para poder trabajar.
* Únicamente tiene una versión gratuita de 15 días.
  1. HERRAMIENTAS DE PROGRAMACIÓN

Lenguajes de programación

**Java**



Figura 16 Logotipo de Java

* **Descripción:** Java es un lenguaje de programación de alto nivel con el que se pueden escribir tanto programas convencionales como para internet.

Java incluye dos elementos: un compilador y un intérprete. El compilador (programa traductor) produce un código de bytes que se almacena en un fichero para ser ejecutado por el intérprete Java denominado máquina virtual de Java. (Ceballos Sierra, 2006)

El lenguaje para la programación en Java, es un lenguaje orientado a objetos, de una plataforma independiente.

El lenguaje para la programación en Java, fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas WEB.

* **Características**:
* Simple.
* Orientado a Objetos.
* Tipado estáticamente.
* Distribuido.
* Interpretado.
* Robusto.
* Seguro.
* De Arquitectura Neutral.
* Multihilo.
* Con Recolector de basura (Garbage Collector)
* Portable.
* De Alto Rendimiento: sobre todo con la aparición de hardware especializado y mejor software.
* Dinámico.
* **Ventajas**:
* Es el JDK, una librería de clases bastante completa que se puede aprovechar gracias a un lenguaje perfectamente orientado a objetos que, corriendo sobre la máquina virtual, te permite olvidarte de algo tan engorroso como es la gestión de memoria (ya sabéis, punteros, reserva y liberación de memoria...).
* Gran cantidad de recursos disponibles, tanto en librerías como en documentación y comunidad.
* Más rápido que lenguajes interpretados.
* Manejo automático de la memoria. (para los que vienen de C/C++).
* Sintaxis similar a C/C++, pero más simple.
* Lenguaje Multi-plataforma: El código que escrito en java es leído por un intérprete, por lo que su programa andará en cualquier plataforma.
* Programación Orientada a Objetos: Paradigma muy utilizado hoy en día que facilita y organiza mucho la programación.
* Puede correr en el explorador y en dispositivos móviles.
* Fácil de aprender.
* **Desventajas**:
* Al tratarse de un lenguaje interpretado, el rendimiento en la ejecución de programas suele ser un poco menor.
* Al contrario que los programas compilados a código nativo, sólo podemos ejecutar un programa en Java si disponemos de una máquina virtual (JVM), sin este simulador no podremos ejecutar ningún programa escrito en Java.
* Aunque java es un lenguaje relativamente sencillo de manejar, no es recomendado implementarlo con personas que apenas se unen a este mundo, ya que Java se basa en una filosofía de programación (la orientación a objetos) que es una evolución de otras formas de entender la programación más sencilla de enseñar e implementar.
* Su sintaxis comparada con C# o Python parece para algunos bastante engorrosa y al contrario que su semejante en .NET, C#, es un lenguaje que evoluciona muy lentamente.
* Menos Eficiente, comparado a C/C++.
* Requiere un intérprete.
* Algunas implementaciones y librerías pueden tener código rebuscado.
* Una mala implementación de un programa en java, puede resultar en algo muy lento.
* Algunas herramientas tienen un costo adicional.

**C++**



Figura 17 Logotipo de C++

* **Descripción:** Es un lenguaje de programación muy utilizado mundialmente en la actualidad, surge en la década de los 80"s por parte de Bjarne Stroustrup, solamente con el simple hecho de mejor el lenguaje de programación anteriormente publicado el Lenguaje C, este tipo de lenguaje es un lenguaje híbrido desde el punto de vista de los lenguajes orientados a objetos, una de las mejores particularidades del lenguaje C++ es la de poder redefinir sus operadores, su nombre viene a darse en 1983 por Rick Mascitti, el cual significa (C++) "incremento de C", refiriendo a este como una actualización al lenguaje C.
* **Características**:
* Tiene un conjunto completo de instrucciones de control.
* Permite la agrupación de instrucciones.
* Incluye el concepto de puntero (variable que contiene la dirección de otra variable).
* Los argumentos de las funciones se transfieren por su valor.
* E/S no forma parte del lenguaje, sino que se proporciona a través de una biblioteca de funciones.
* Permite la separación de un programa en módulos que admiten compilación independiente.
* Originalmente el Lenguaje C estuvo muy ligado al sistema operativo UNIX como se había mencionado antes que, en su mayor parte, está escrito en C. Más adelante se comenzó a utilizar en otros sistemas operativos para programar editores, compiladores, etc.
* **Ventajas**:
* El lenguaje C ha sido pensado para ser altamente transportable.
* Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de archivos, proporcionadas por bibliotecas.
* Es un lenguaje muy flexible que permite programar con múltiples estilos
* Usa un lenguaje de reprocesado de C, para tareas como definir macros e incluir múltiples archivos de código fuente.
* Acceso a memoria de bajo nivel mediante el uso de punteros.
* Interrupciones al procesador con uniones.
* Un conjunto reducido de palabras clave.
* Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.
* **Desventajas**:
* Uso de DLLs (librerías dinámicas) muy complejo. Java y .Net han evolucionado estos conceptos manipulando las DLLs mediante los frameworks que proveen. En cambio, en C++ el desarrollador debe encargarse de cargar y liberar de memoria estas librerías, y correr los riesgos por el manejo de esta memoria.
* Elaborar un sistema en C++ es como construir un rascacielos: tiene buen soporte y es robusto, pero si existen errores en los pisos inferiores toda la parte superior se viene abajo terriblemente.
* No es recomendable para desarrollo de páginas Web.
* Es mucho más difícil de usar q otros lenguajes de programación.
* Es difícil programar bases de datos
* Los punteros (es muy complicado su sistema).

**PHP**

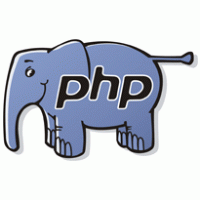


Figura 18 Logotipo de PHP

* **Descripción:** PHP (Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

PHP es un lenguaje de alto nivel que se ejecuta en el servidor.

¿Qué quiere decir que se ejecuta en el servidor?

Un lenguaje de servidor es aquel que se ejecuta en el servidor donde están alojadas las paginas, al contrario que otros lenguajes que son ejecutados en el propio navegador.

¿Qué ventajas tiene el ser un lenguaje de servidor?

La principal ventaja es que, al ejecutarse el código en el servidor, todas nuestras páginas van a poder ser vistas en cualquier ordenador, independientemente del navegador que tenga. En cambio, el gran problema de que se ejecute el código en el navegador es que muchos navegadores no son capaces de entender todo el código, lo que presentaría errores al mostrar el resultado de las páginas. (Puertas, 2004)

* **Características**:
* Autenticación HTTP con PHP.
* Cookies.
* Sesiones.
* Manejo de Xforms.
* Subida de ficheros.
* Subida con el método POST.
* Explicación de los mensajes de error.
* Dificultades comunes.
* Subida de varios ficheros.
* Soporte para el método PUT.
* Empleo de ficheros remotos.
* Manejo de conexiones.
* **Ventajas**:
* “Principalmente, que se trata de un lenguaje de programación gratuito y, por tanto, todo el mundo puede utilizarlo sin ningún coste, frente a otros lenguajes cuyo software es necesario comprar para su utilización”. (Puertas, 2004)
* Es un lenguaje sencillo y fácil de estudiar y aprender.
* Una de sus características es la rapidez.
* Es un lenguaje multiplataforma.
* PHP no requiere ningún tipo de licencia.
* Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
* El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
* Tiene ciertas características de los lenguajes orientados a objetos como la utilización de clases y herencias.
* Puede mezclarse con código HTML, aunque esto dificulta su lectura.
* Puede manejar ficheros y conectarse a distintas bases de datos (MySQL, Oracle, SQL Server, Informix, PostgreSQL, etcétera).
* **Desventajas**:
* El lugar más seguro para ejecutar una aplicación es en un servidor propio, por lo cual, si un cliente o usuario requiere su código en su pc, tendríamos que dejar su código, sin manera de ocultarlo, aunque hay muchas aplicaciones que nos ayuda a encriptar el código fuente.
* Debes saber cuándo menos HTML para poder hacer un trabajo medianamente funcional.
* Si no lo configuras correctamente dejas abiertas muchas brechas de seguridad.
* Se necesita instalar un servidor web.
* Promueve creación de código desordenado y con un mantenimiento complejo.

**JavaScript**



Figura 19 Logotipo de JavaScript

* **Descripción:** JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web capaces de interactuar con el usuario. Las páginas web se consideran estáticas cuando se limitan a mostrar el contenido establecido por su creador sin proporcionar más opciones al usuario que elegir entre los enlaces disponibles para seguir navegando. Cuando un creador incorpora JavaScript a su página web, es decir, cierto dinamismo y por lo tanto de incrementan las prestaciones de la misma al añadir procesos en respuesta a las acciones del usuario. Es importante señalar que estos procesos se ejecutan en la máquina del cliente (en el navegado) y por tanto no implican intercambio de datos con el servidor. Con JavaScript se accede al mundo de las páginas web dinámicas. (Mohedano, Saiz, & Salazar Román, 2012)

Aunque comparte muchas de las características y de las estructuras del lenguaje Java, fue desarrollado independientemente. El lenguaje JavaScript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico. Por ejemplo, hace fácil responder a los acontecimientos iniciados por usuarios (como introducción de datos en formularios) sin tener que utilizar CGI.

* **Características**:
* JavaScript comparte muchos elementos con otros lenguajes de alto nivel.
* Es muy semejante a otros como C, Java o PHP, tanto en su formato como en su sintaxis.
* JavaScript es un lenguaje que diferencia entre mayúsculas y minúsculas.
* Las expresiones se denominan operadores y sirven tanto para encerrar expresiones como para realizar trabajos con ellas, como operaciones matemáticas o de texto. Los operadores que permiten encerrar expresiones deben cerrarse siempre. ‘(‘, ‘{‘ y ‘[‘ deben cerrarse con sus correspondientes ‘)’, ‘}’ y ‘]’, respectivamente.
* Es un lenguaje de formato libre.
* **Ventajas**:
* Excelente herramienta para validación de formularios del lado del cliente.
* Provee mayor diseño dinámico en imágenes y otros elementos en sitios WEB.
* **Velocidad:** Al ser client-side, JavaScript es muy rápido y cualquier función puede ser ejecutada inmediatamente en lugar de tener que contactar con el servidor y esperar una respuesta.
* **Simplicidad:** JavaScript es relativamente simple de aprender e implementar.
* **Versatilidad:** JavaScript encaja perfectamente con otros lenguajes y puede ser usado en una gran variedad de aplicaciones. Al contrario de PHP o scripts SSI, JavaScript puede insertarse en cualquier página independientemente de la extensión del fichero. JavaScript puede también ser usado dentro de scripts escritos en otros lenguajes como Perl y PHP.
* **Carga del servidor:** Al ejecutarse del lado del cliente reduce la carga en el servidor de la página web.
* Se recomienda para la creación de aplicaciones web.
* Útil para el desarrollo de páginas web dinámicas.
* Tiene gran cantidad de efectos visuales.
* Es soportado por la mayoría de los navegadores.
* Puede agregar interactividad a los elementos web.
* **Desventajas**:
* En el FrontEnd sus códigos son visibles, por lo tanto, pueden ser leídos por cualquier usuario.
* Tiende a introducir gran cantidad de fragmentos de código en los sitios web.
* Sus opciones 3D son limitadas, si se quiere utilizar este lenguaje de programación para crear un juego, deben emplearse otras herramientas.
* No es compatible en todos los navegadores de manera uniforme, es decir que algunas veces es interpretado diferente dependiendo en el navegador que sea ejecutado.
* Los usuarios tienen la opción de desactivar JavaScript desde su navegador.
* Vulnerabilidades de seguridad al descargar el código y ejecutarlo en el navegador con posibilidad de ejecutar algún código malicioso.
* Posibilidad de introducir una cantidad enorme de fragmentos de código en sitios Web.

**JSP**

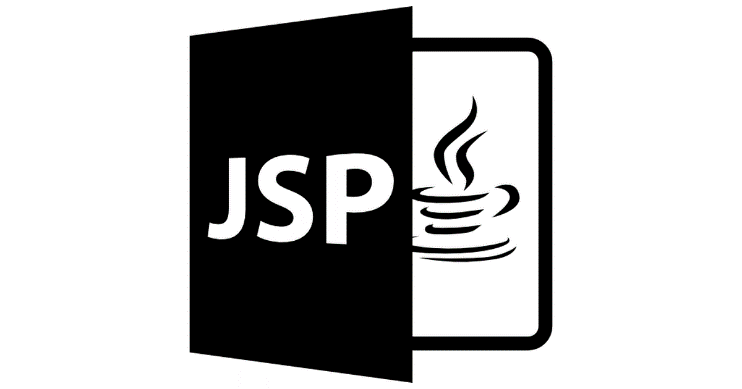


Figura 20 Logotipo de JSP

* **Descripción:** Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor. JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET desarrollado para la creación de aplicaciones web potentes. (Valdés, 2007)
* **Características:** La tecnología de JSP permite a los desarrolladores y a los diseñadores de web desarrollar rápidamente y mantener fácilmente páginas dinámicas, ricas en información como son las que soportan a sistemas de negociación. La tecnología de los JSP separa la interfaz del usuario de la parte lógica del contenido permitiendo a los diseñadores cambiar a su disposición las plantillas de la interfaz sin alterar el contenido dinámico subyacente.
* JSP también permite introducir código para la generación dinámica de HTML dentro de una página web.
* Conjunta el poder de Java en el servidor y la flexibilidad de HTML en el browser.
* No sólo se puede utilizar HTML, sino también XML o WML y la extensión de los archivos de este tipo es “.jsp”.
* Hace más fácil reusar componentes con JavaBeans los cuales realizan tareas más específicas.
* Su función es saber cómo procesar una solicitud para crear una respuesta.
* Soporta contenido dinámico que refleja las condiciones del mundo real.
* Es más rápido y fácil crear aplicaciones de web.
* Capaz de instanciar cualquier clase de Java.
* **Ventajas**:
* El código está bien estructurado y resulta fácil de leer si se conoce bien el lenguaje.
* La parte de JSP dinámico se escribe en Java, permitiendo una integración total con módulos Java y la utilización de un motor de páginas basado en servlets de Java.
* Reusabilidad, robustez y multiplataforma.
* Permite una integración total de módulos Java.
* Permite la utilización de un motor de páginas basado en servlets de Java.
* El código está estructurado, por lo que es fácil de leer si se conoce el lenguaje.
* **Desventajas**:
* Es un lenguaje bastante complejo y que requiere bastante tiempo de aprendizaje.
* El alojamiento web requiere tener instalado un servidor Tomcat.
* No posee una interfaz con muchas posibilidades, ya que la interfaz web es limitada.
* La carga de la interfaz es más lenta que la de una aplicación de escritorio.
* La mayor parte de la lógica de la aplicación se ejecuta en el servidor, por lo que se corre el riesgo de sobrecargarlo de trabajo.
* La aplicación no está disponible si ocurre algún problema con el servidor o con la conexión de red.

Entornos de desarrollo

**Eclipse**



Figura 21 Logotipo de Eclipse

* **Descripción:** Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.
* **Características**:
* **Perspectivas, editores y vistas:** En Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una preconfiguración de ventanas y editores, relacionadas entre sí, y que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.
* **Gestión de proyectos:** El desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente, documentación, ficheros configuración, árbol de directorios, etc. El IDE nos proporcionará asistentes y ayudas para la creación de proyectos. Por ejemplo, cuando creamos uno, se abre la perspectiva adecuada al tipo de proyecto que estemos creando, con la colección de vistas, editores y ventanas preconfigurada por defecto.
* **Depurador de código:** Se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente nos ayuda a mejorar nuestro código. Para ello sólo debemos ejecutar el programa en modo depuración (con un simple botón). De nuevo, tenemos una perspectiva específica para la depuración de código, la perspectiva depuración, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.
* **Extensa colección de plugins:** Están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Al haber sido un estándar de facto durante tanto tiempo (no el único estándar, pero sí uno de ellos), la colección disponible es muy grande. Los hay gratuitos, de pago, bajo distintas licencias, pero casi para cualquier cosa que nos imaginemos tenemos el plugin adecuado.
* **Ventajas**:
* Es de código abierto.
* Eclipse dispone de un Editor de texto con un analizador sintáctico.
* La compilación es en tiempo real.
* Tiene pruebas unitarias con JUnit, control de versiones con CVS.
* Pueden desarrollar proyectos en los lenguajes: Java, ANSI C, C++, JSP, SH, Perl, Php, Sed.
* **Desventajas:**
* El consumo de recursos del sistema.
* Los plugins por lo general no son ni tan potentes ni tan sencillos.
* Carece de mucho soporte para webapps.

**NetBeans**



Figura 22 Logotipo de NetBeans

* **Descripción:** La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.
* **Características:** La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

* Administración de las interfaces de usuario (ej. menús y barras de herramientas).
* Administración de las configuraciones del usuario.
* Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
* Administración de ventanas.
* Framework basado en asistentes (diálogos paso a paso).
* **Ventajas**:
* La plataforma NetBeans puede ser usada para desarrollar cualquier tipo de aplicación.
* Reutilización del Módulos.
* Permite el uso de la herramienta Update Center Module.
* Instalación y actualización simple.
* Incluye Templates y Wizards.
* Posee soporte para PHP.
* **Desventajas**:
* Poca existencia de plugins para esta plataforma.
* hace falta documentación del Rich Client Plataform (RCP).
* No posee un editor de código HTML.

**PhpStorm**



Figura 23 Logotipo de PhpStorm

* **Descripción:** IDE multiplataforma para PHP construida sobre IntelliJ IDEA JetBrains' plataforma. PhpStorm proporciona un editor de PHP, HTML y JavaScript con el análisis de código en la marcha, la prevención de errores y automatizadas refactorizaciones para PHP y JavaScript. PhpStorm de finalización de código es compatible con PHP 5.3, 5.4, 5.5, 5.6 y 7.0 (proyectos modernos y heredados). Incluye una de pleno derecho de editor SQL con resultados de la consulta editables.
* **Características**:
* Permite la gestión de proyectos fácilmente.
* Proporciona un fácil autocompletado de código.
* Soporta el trabajo con PHP 5.5
* Sintaxis abreviada.
* **Ventajas**:
* Posibilidad de instalar temas y plugins para mejorar su funcionamiento.
* Funcionalidades comparables con los demás entornos de desarrollo como soporte de varios lenguajes, prevención de errores etc.
* Reutilización del Módulos.
* Instalación y actualización simple.
* Posee soporte para PHP.
* Código libre.
* **Desventajas**:
* Poca existencia de plugins para esta plataforma.
* Mucho consumo de recursos.
* No es fácil aprender a utilizarlos en su totalidad.
* Es un software de paga.
* Se necesitan complementos para que este completo.

Frameworks de desarrollo

**CodeIgniter**



Figura 24 Logotipo de CodeIgniter

* **Descripción:** CodeIgniter es un entorno de desarrollo de aplicaciones para gente que construye sitios web usando PHP. El objetivo es habilitar el desarrollo de proyectos de forma mucho más rápida de lo que podría si escribiese código desde cero, a través de proveer un rico conjunto de librerías para tareas comúnmente necesarias, tanto como una simple interface y estructura lógica para acceder a estas librerías. CodeIgniter le permite concentrarse creativamente en su proyecto, minimizando el volumen de código necesario para una tarea determinada.
* **Características**:
* Sistema Basado en Modelo-Vista-Controlador.
* Compatible con PHP 4.
* Extremadamente Liviano.
* Soporte de Active Record para Base de Datos.
* Clases de base de datos llenas de características con soporte para varias plataformas.
* Formulario y Validación de Datos.
* Seguridad y Filtro XSS.
* Manejo de Sesión.
* Clase de Envío de Email. Soporta Archivos Adjuntos, email de texto/HTML, múltiples protocolos (sendmail, SMTP, and Mail) y más.
* Librería de Manipulación de Imagen (cortar, redimensionar, rotar, etc.). Soporta GD, ImageMagick, y NetPBM.
* Clase de Carga (upload) de Archivo.
* Clase de FTP.
* Localización.
* Paginación.
* Encriptación de Datos.
* Puntos de referencia.
* Cacheo de páginas enteras.
* Historial de Errores.
* Perfilando la Aplicación.
* Clase de Calendario.
* Clase de Agente del Usuario.
* Clase de Codificación Zip.
* Clase de Motor de Plantillas.
* **Ventajas**:
* **Versatilidad:** Quizás la característica principal de CodeIgniter, en comparación con otros frameworks PHP. CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.
* **Compatibilidad:** CodeIgniter, al menos en el momento, es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Por supuesto, funciona correctamente también en PHP 5.
* **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
* **Flexibilidad:** CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos casos se pueden seguir o no, y sus reglas de codificación muchas veces se pueden saltar para trabajar como más a gusto encontremos. Algunos módulos, como el uso de plantillas, son totalmente opcionales. Esto ayuda muchas veces a que la curva de aprendizaje sea más sencilla al principio.
* **Ligereza:** CodeIgniter es liviano. El núcleo del sistema sólo requiere unas pocas pequeñas librerías. Esto es en duro contraste a muchos entornos de trabajo que requieren significativamente más recursos. Las librerías adicionales son cargadas dinámicamente bajo demanda, basándose en sus necesidades para un proceso dado, así que el sistema base es muy delgado y bastante rápido.
* **Documentación tutorializada:** La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto no facilita mucho la referencia rápida, cuando ya se tienen conocimientos acerca del framework y se quiere consultar sobre una función o un método en concreto, pero para iniciarnos sin duda se agradece mucho.
* **Extensibilidad:** El sistema puede ser fácilmente extendido a través del uso de plugins y librerías asistentes, o a través de extensión de clases o ganchos del sistema.
* **Desventajas**:
* No tiene sistema de plantillas.
* No tiene un layout general.
* No hay módulos.
* No hay un ORM.
* Los controladores no cargan por defecto las listas.

**Zend Framework**



Figura 25 Logotipo de Zend

* **Descripción:** Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones web al combinarse.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y StrikeIron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

* **Características**:
* Basado en PHP.
* Código 100% orientado a objetos.
* Implementa el patrón MVC (modelo-vista-controlador).
* Es open source.
* Amplia documentación.
* Constante desarrollo.
* Cuenta con soporte para internalización y localización de aplicaciones (construir sitios multi idioma, convertir formatos de fechas, monedas, etc. según la región.
* Arquitectura de uso a voluntad con componentes ligeramente acoplados e interdependencias mínimas.
* Soporte para múltiples sistemas de bases de datos y proveedores, incluyendo MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite e Informix Dynamic Server.
* Marcos de validación de formularios.
* Proporciona una gran cantidad de componentes:
* **Zend\_Auth:** Proporciona una API para la autenticación e incluye adaptadores concretos de autenticación para casos de uso comunes.
* **Zend\_Date:** Ofrece una API detallada y sencilla para manipular fechas y horas.
* **Zend\_Db:** una capa de acceso a la base de datos ligera, que proporciona una interfaz para PDO y otras extensiones de base de datos en PHP.
* **Zend\_Json:** Convierte fácilmente las estructuras de PHP en JSON y viceversa para su uso en aplicaciones habilitadas para AJAX.
* **Zend\_Loader:** Carga archivos, clases y recursos dinámicamente en tu aplicación PHP.
* **Zend\_Mail y Zend\_Mime:** Crea mensajes de correo electrónico y los envía.
* **Zend\_Pdf:** Soporta formato de archivo PDF de Adobe.
* **Zend\_Session:** Ayuda a administrar y preservar datos de sesión a través de varias solicitudes de página por el mismo cliente.
* **Ventajas**:
* Está totalmente orientado a objetos.
* Contiene muchos componentes.
* Comunidad y documentación.
* Reduce el "time to market" de las aplicaciones, permitiendo ofrecer presupuestos más ajustados.
* Estandariza los procesos más frecuentes, dotándolos de gran robustez.
* Facilita el mantenimiento de las aplicaciones.
* Ofrece muchas facilidades para el acceso a recursos avanzados, que de otro modo resultan bastante más costosos de desarrollar.
* A diferencia de otros frameworks, es posible utilizarlo en modo "desacoplado", es decir, aquellas clases o componentes que sean necesarios en cada proyecto, sin arrastrar todo el framework detrás para cualquier pequeña necesidad.
* Tiene el respaldo de la propia ZEND, creadora de PHP, lo que asegura su continuidad futura tanto como la del propio lenguaje PHP.
* **Desventajas**:
* Difícil instalación.
* Requiere de una larga curva de aprendizaje.
* Necesita de una gran configuración antes de poder comenzar a trabajar.
* Su estructura de archivos no está definida por sí misma, cualquier desarrollador puede modificarla, esto indica que no sería compatible con otra aplicación desarrollada con Zend.
* Es necesario comprender algunos patrones de diseño y programación orientada a objetos para utilizar todo el potencial de Zend Framework. Y eso no sólo requiere tiempo dedicado al aprendizaje, sino también la experiencia de trabajar con él.
* Es grande, es pesado, gasta mucha memoria y tiene una gran cantidad de inclusiones.

**Yii**



Figura 26 Logotipo de Yii

* **Descripción:** Yii es un framework PHP basado en componentes de alto rendimiento para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo.

Yii es un framework genérico de programación Web que puede ser utilizado para todo tipo de aplicaciones Web. Gracias a que es liviano de correr y está equipado con soluciones de cacheo sofisticadas, es adecuado para desarrollar aplicaciones de gran tráfico como portales, foros, sistemas de administración de contenidos (CMS), Sistemas de comercio electrónico (e-commerce), etc.

Como la mayoría de los frameworks PHP, Yii es un framework MVC (modelo-vista-controlador).

Yii sobresale frente a otros frameworks PHP en su eficiencia, su gran cantidad de características y su clara documentación.

* **Características**:
* Usa el patrón MVC (Modelo Vista Controlador) que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
* Provee un Nivel de Abstracción hacia la Base de Datos (DAO, Data Access Object) y se apoya en el patrón de arquitectura de datos Active Record.
* Incluye soporte para autenticación de usuarios basado en el modelo RBAC role-based access control o control de acceso basado en roles.
* Herramientas para automatización y validación de Formularios
* Integración con jQuery.
* Manejo de Seguridad (Sql injection, XSS, CSRF, cookie tampering, etc.). Las medidas de seguridad incluyen la prevención cross-site scripting (XSS), prevención cross-site request forgery (CSRF), prevención de la manipulación de cookies, etc.
* Generador de código automático, muy útil a la hora de implementar formularios o bien operaciones CRUD. Además de que el código generado cumple XHTML standards.
* Amigable y extensible, por lo cual se pueden usar sin problemas librerías de terceros como Pear, Zend, etc.
* Widgets habilitados para AJAX.
* Pruebas de unidad y funcionalidad.
* **Ventajas**:
* Es un Framework muy ligero.
* Tiene una curva de aprendizaje muy corta comparada con otros frameworks.
* Promueve la organización del código basado en el patrón de diseño MVC.
* Ofrece muchas características que simplifican las tareas al desarrollador.
* Es extensible a través de widgets.
* Ofrece alto rendimiento en aplicaciones de gran envergadura.
* Por defecto utiliza Bootstrap para renderizar elementos HTML (inputs, radios, etc.).
* Es libre y está en constante desarrollo.
* **Desventajas**:
* **Dependencia:** Una vez que escribimos un programa para determinado framework es casi imposible migrar a otro sin tener que reescribir todo el código.
* Con Yii framework se pueden hacer páginas web, pero no es la herramienta más óptima para hacer páginas web, para eso es mejor usar un CMF (content management framework), como lo es Joomla, Drupal o WordPress, el cual supera por mucho el alcance de Yii framework, el cual está orientado a aplicaciones web, mas no tanto a páginas web tradicionales.
* El uso de jQuery y Ajax aún depende mucho del conocimiento que tengas de ambos, si tu conocimiento es bajo puedes pasar días tratando de resolver un problema. El framework como tal no provee un eslabón jquery ajax "maduro, reutilizable y orientado a objetos".
  1. HERRAMIENTAS DE BASE DE DATOS

Sistemas gestores de bases de datos

**MySQL**



Figura 27 Logotipo de MySQL

* **Descripción:** MySQL es un gestor de base de datos completamente desarrollado en lenguaje C/C++, lo que ofrece una estabilidad de trabajo impresionante. También, cuenta entre sus características, con una excelente capacidad de integración con diferentes entornos de desarrollo de software y de aplicaciones cliente/servidor, por lo tanto, es muy popular entre los programadores de aplicaciones web y entre administradores de base de datos en todo el mundo.Es un sistema de gestión de base de datos relacional, multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual.
* **Características**:
* Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
* Soporta gran cantidad de tipos de datos para las columnas.
* Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
* Gran portabilidad entre sistemas.
* Soporta hasta 32 índices por tabla.
* Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
* Condición de open source de MySQL hace que la utilización sea gratuita y se puede modificar con total libertad.
* Se puede descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones.
* Es una de las herramientas más utilizadas por los programadores orientados a Internet.
* Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
* MYSQL, es el manejador de base de datos considerado como el más rápido de Internet.
* Gran rapidez y facilidad de uso.
* Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
* Fácil instalación y configuración.
* **Ventajas**:
* “Al igual que PHP, su principal ventaja reside en que es una base de datos gratuita” (Puertas, 2004)
* MySQL software es OpenSource.
* Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
* Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
* Facilidad de configuración e instalación.
* Soporta gran variedad de Sistemas Operativos.
* Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
* El software MySQL usa la licencia GPL.
* **Desventajas**:
* Un gran porcentaje de las utilidades de MySQL no están documentadas.
* No es intuitivo, como otros programas (ACCESS).
* Otro problema común es, el tamaño de la base de datos. Aunque en teoría es escalable hasta 8 TB, MySQL no puede trabajar de manera eficiente con grandes bases de datos.

**Oracle**



Figura 28 Logotipo de Oracle

* **Descripción:** Fabricado por Oracle Corporation, utiliza la arquitectura cliente/servidor. Ha incorporado en su sistema el modelo objeto-relacional, pero al mismo tiempo garantiza la compatibilidad con el tradicional modelo relacional de datos. Así ofrece un servidor de bases de datos híbrido. Es uno de los más conocidos y ha alcanzado un buen nivel de madurez y de profesionalidad. Se destaca por su soporte de transacciones, estabilidad y escalabilidad.
* **Características**:
* Modelo relacional: los usuarios visualizan los datos en tablas con el formato filas/columnas.
* Herramienta de administración gráfica intuitiva y cómoda de utilizar.
* Control de acceso: tecnologías avanzadas para vigilar la entrada a los datos.
* Protección de datos: seguridad completa en el entorno de producción y de pruebas y gestión de copias de seguridad.
* Lenguaje de diseño de bases de datos muy completo (PL/SQL): permite implementar diseños "activos", que se pueden adaptar a las necesidades cambiantes de negocio.
* Alta disponibilidad: escalabilidad, protección y alto rendimiento para la actividad empresarial.
* Gestión de usuarios: agilidad en los trámites, reducción de costes y seguridad en el control de las personas que acceden a las aplicaciones y a los sistemas.
* **Ventajas**:
* Oracle es el motor de base de datos objeto-relacional más usado a nivel mundial.
* Puede ejecutarse en todas las plataformas, desde una Pc hasta un supercomputador.
* Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
* Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
* El software del servidor puede ejecutarse en multitud de sistemas operativos.
* Existe incluso una versión personal para Windows 9x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa.
* Oracle es la base de datos con más orientación hacía INTERNET.
* **Desventajas**:
* Las versiones más recientes de Oracle son la 11g, 10g, 9g, 8g, desde el lanzamiento original de la 8 se sucedieron varias versiones con correcciones, hasta alcanzar la estabilidad en la 8.0.3. El motivo de tantos fallos fue, al parecer, la remodelación del sistema de almacenamiento por causa de la introducción de extensiones orientadas a objetos.
* El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras, en mi opinión. Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.
* También es elevado el costo de la información, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.

**SQL Server**

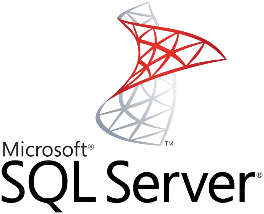


Figura 29 Logotipo de SQL Server

* **Descripción:** Es un sistema de gestión de bases de datos relacionales basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.
* **Características**:
* Soporte de transacciones.
* Escalabilidad, estabilidad y seguridad.
* Soporta procedimientos almacenados.
* Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
* Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
* Además, permite administrar información de otros servidores de datos.
* **Ventajas**:
* SQL Server nos permite olvidarnos un poco de los ficheros que forman la base de datos. En Access tenemos que abrir el fichero .mdb para ver el contenido. Aquí abrimos la consola de SQL Server y tenemos todas las bases de datos de todos nuestros programas.
* Si trabajamos en una red local nos permite agregar otros servidores de SQL Server.
* Ofrece una potente forma de unir SQL e Internet.
* Seguridad: SQL permite administrar permisos a TODO. Permisos a nivel de servidor, seguridad en tablas, permitir o no lectura, escritura, ejecución, todo se puede configurar. O sea, que podemos permitir que alguien conecte su SQL al nuestro, pero sin embargo podemos decirle que no puede ver esta base de datos, pero otra sí.
* **Desventajas**:
* Utiliza mucho la memoria RAM para las instalaciones y utilización de software.
* No se puede utilizar como practicas porque se prohíben muchas cosas, tiene restricciones en lo particular.
* La relación, calidad y el precio está muy debajo comparado con Oracle.
* Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño, una pésima implementación de los tipos de datos variables.

Herramientas para gestionar Bases de Datos

**phpMyAdmin**



Figura 30 Logotipo de phpMyAdmin

* **Descripción:** Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet, está disponible bajo la licencia GPL (General Public License y en más de 50 idiomas este proyecto se encuentra vigente desde el año 1998.

Con esta herramienta puedes crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos.

* **Características**:
* Interfaz sobre web intuitiva.
* Proporciona herramientas de gestión de la base de datos: Edición, creación, modificación y eliminación de bases de datos, tablas, vistas, campos, relaciones e índices.
* Mantenimiento de usuarios y sus privilegios.
* Mantenimiento de procedimientos almacenados.
* Importación de datos desde CSV y SQL.
* Multiplataforma.
* Multilenguaje (más de 50).
* Exportación a varios formatos: CSV, SQL, XML, PDF, SO/IEC 26300 - OpenDocument Text y Spreadsheet, Word, LATEX y otros.
* Administración de múltiples servidores.
* Creación del despliegue de la base de datos en un gráfico exportado a PDF.
* Creación de consultas complejas haciendo uso QBE (Query By Example).
* **Ventajas**:
* Posee una interfaz web intuitiva.
* Es desarrollada en PHP.
* Sirve de referencia para la creación de las herramientas como PHPgAdmin.
* Se encuentra bajo licencia GNU GPL, que nos permite la libre distribución, modificación y uso.
* Se pueden importar datos de archivos CVS y SQL.
* Capacidad para ejecutar la mayoría de características de MySQL.
* **Desventajas**:
* Su código se basa solo en PHP.
* La cantidad de bases de datos manejadas se ven muy limitadas, comparando con DataBase Master que maneja, Oracle, SQL Server, MySQL, PostgreSQL, FireBird y SQLite.
* Es más pesado que la herramienta Adminer, ya que este último es único archivo de PHP que también trabaja sobre bases de datos MySQL.
* En cuestión de procedimientos almacenados y disparadores, no puede crear, modificar ni buscar dentro de estos.
* Solo administra bases en MySQL, quedando fuera otras opciones como PostgreSQL y Microsoft SQL Server.
* La visualización grafica del constructor de consultar, el esquema, modelo y diseño del diagrama entidad-relación no es posible con esta herramienta.

**Open System Architect**



Figura 31 Logotipo de Opne System Architect

* **Descripción:** Un software libre disponible en Windows, Mac OS X, Linux y Solaris. La herramienta te permite crear modelados de datos (modelo físico y modelo lógico). Se caracteriza por tener una buena organización de proyectos. El acceso a las bases de datos es vía ODBC y no trabaja bajo el entorno Java (es algo habitual en este tipo de utilidades).
* **Características**:
* Posee un repositorio único que integra todas las herramientas, y metodologías usadas.
* Conecta directamente al diccionario de datos, los elementos asociados, comentarios, reglas de validaciones, normalización, etc.
* Posee control automático de diagramas y datos, normalizaciones y balanceamiento entre diagramas "Padre e Hijo", además de balanceamiento horizontal, que trabaja integrado con el diccionario de datos, asegurando la compatibilidad entre el Modelo de Datos y el Modelo Funcional.
* Es considerado un Upper Case, que puede ser integrado a la mayoría de los generadores de código.
* Traduce modelos de entidades, a partir de la enciclopedia, en esquemas para Sybase, DB2, Oracle u Oracle 7, Ingress, SQL Server, RDB, XDB, Progress, Paradox, SQL Base, AS400, Interbase, OS/2, DBMS, Dbase 111, Informix, entre otros.
* Genera también Windows DDL, definiciones de datos para lenguaje C/C++ y estructuras de datos en Cobol.
* Posee esquemas de seguridad e integridad a través de contraseñas que posibilitan el acceso al sistema en diversos niveles, pudiéndose integrar a la seguridad de la red Novell o Windows/NT de ser necesario.
* Posee un módulo específico para Ingeniería Reversa desde las Bases de Datos SQL más populares, incluyendo Sybase, DB2, Infonmix, Oracle y SQL Server (DLL), además de diálogos (DLG) y menúes (MNU) desde Windows.
* Logra leer bases de datos y construir el modelo lógico o físico (diagrama), alimentando su diccionario de datos con las especificaciones de las tablas y de sus elementos de datos, incluyendo las relaciones entre tablas y su cardinalidad.
* **Ventajas**:
* Open System Architect es una herramienta potente, con la cual podrás crear modelos desde cero o desde modelos existentes y todo bajo un entorno multiplataforma.
* Posee esquemas de seguridad e integridad a través de contraseñas que posibilitan el acceso al sistema en diversos niveles, pudiéndose integrar a la seguridad de la red.
* Puede ser integrado a la mayoría de los generadores de código.
* Proporcionan amplio soporte para la construcción de los Modelo Conceptual, Funcional y Operacional.
* Además de tener un módulo específico para customización de informes que emplea los recursos gráficos y facilidad de edición de Windows.
* **Desventajas**:
* Un punto en contra es su interfaz antigua y poco elaborada que hace que los diagramas generados no sean visualmente muy atractivos.
* La herramienta te permite importar tablas, admite herencias, validación ERD, dispone de un editor de SQL, pero no podrás agregar tablas “Padres” o “Relacionadas”.
* Tiene problemas de ejecución cuando se trata de mantener los datos separados del cliente.

**MySQL Workbench**



Figura 32 Logotipo de Workbench

* **Descripción**: MySQL Workbench, es una poderosa herramienta que nos ofrece la posibilidad de diseñar base de datos en forma visual, es decir, podemos “dibujarla” y luego automáticamente crea los scripts necesarios para su creación en el servidor; además, permite documentar y manejar nuestras bases.

MySQL Workbench es el sucesor de DBdesigner 4 que era desarrollado por la empresa fabForce.net, ahora el proyecto lo desarrolla el equipo de Sun Microsystems, pero continúa siendo dirigido por el fundador y creador de DBdesigner, Michael G. Zinner.

* **Características**:
* Edición de diagramas basada en Cairo, con posibilidad de realizar una salida en los formatos como OpenGL, Win32, X11, Quartz, PostScript, PDF.
* Proporciona una representación visual de las tablas, vistas, procedimientos y funciones almacenadas y claves foráneas.
* Permite acceso a bases de datos e ingeniería inversa de las mismas para crear los SQL de creación.
* Ofrece sincronización con la base de datos y el modelo.
* Permite generar los scripts SQL a partir del modelo creado.
* Ofrece una arquitectura extensible.
* Tiene soporte para exportar los datos como script SQL CREATE.
* Permite importar modelos de DBDesigner4.
* Ofrece soporte completo a las características de MySQL 5.
* Permite exportar la base en varios formatos por ejemplo pdf, png, etc.
* Soporta múltiples diagramas para un mismo proyecto.
* La calidad visual de la aplicación es excelente.
* **Ventajas**:
* Libre, distribuida bajo la licencia GPL.
* Multiplataforma. disponible para Windows, GNU/Linux y Mac.
* Permite crear diagramas E-R.
* Importar archivos SQL.
* Permite generar los scripts SQL a partir del modelo creado.
* Ofrece soporte completo a las características de MySQL 5.
* La herramienta dispone de una versión comercial la cual dispone de una validación del modelado o la opción de ingeniaría inversa de base de datos.
* **Desventajas**:
* Características no disponibles en la versión gratuita.
* **Complejidad:** Se debe asegurar que la base de datos sea transparente, se debe lidiar con varios sistemas diferentes que pueden presentar dificultades únicas. El diseño de la base de datos se tiene que trabajar tomando en cuenta su naturaleza distribuida, por lo cual no podemos pensar en hacer consultas JOIN que afecten varios sistemas.
* **Seguridad:** Se debe trabajar en la seguridad de la infraestructura, así como cada uno de los sistemas.
* **Carencia de estándares:** Aún no existen herramientas o metodologías que ayuden a los usuarios a convertir un DBMS (Sistema de Administración de Base de Datos) centralizado en un DBMS distribuido.
* **Mayor probabilidad de errores:** Como los nodos que constituyen el sistema funcionan en paralelo, es más difícil asegurar el funcionamiento correcto de los algoritmos, así como de los procedimientos de recuperación de fallos del sistema.

CONCLUSIONES

**Jonathan Jair Alfaro Sánchez**

**¿Con qué problemas se presentaron en la realización de su proyecto?**

Durante esta actividad se presentaron algunos detalles, el mas grande para mi fue formar la ruta critica en Microsoft Project. Se me hizo algo difícil ya que nunca antes lo había hecho así.

Como equipo también tenemos ciertos fallos en la redacción.

**¿Cuáles fueron las estrategias a seguir para resolver dichos problemas?**

Cada problema que se nos presenta lo resolvemos como equipo, como primer instancia hablamos sobre el tema entre nosotros, si no podemos resolverlo o necesitamos información recurrimos a internet, esto con el fin de tener mas bases, y por ultimo cuando necesitamos más orientación acudimos a los profesores.

**Mary Carmen Crescencio Bernal**

**¿Con qué problemas se presentaron en la realización de su proyecto?**

Durante la realización de la actividad se presentaron algunos problemas, uno de ellos fue la realización de la ruta crítica dentro de Project ya que en proyectos anteriores no la habíamos implementado, otro de los problemas que se nos presentaron fue poder redactar algunos conceptos ya que, sabíamos que era lo que se quería dar a entender, pero no sabíamos cómo plasmarlo en el documento.

**¿Cuáles fueron las estrategias a seguir para resolver dichos problemas?**

Las estrategias que aplicamos para poder resolver los problemas, fueron los siguientes: para la realización de la ruta crítica, fue buscar información en internet como tutoriales para poder realizarlo y para la redacción, fue leer y comprender los objetivos.

**Miguel Ángel Mandujano Barragán**

**¿Con qué problemas se presentaron en la realización de su proyecto?**

Considero que los problemas que se han presentado hasta este punto han sido más que nada la manera de redactar algunos textos necesarios para comenzar a desarrollar el proyecto en cuanto a la página web.

**¿Cuáles fueron las estrategias a seguir para resolver dichos problemas?**

La manera de solucionarlos fue tomar la opinión de todos los integrantes del equipo, de esta manera todos aportábamos algunas ideas en busca de mejorar la redacción o contenido del texto, además de solicitar ayuda por parte de la profesora.

**Giovanni Misael Alfaro Sánchez**

**¿Con qué problemas se presentaron en la realización de su proyecto?**

Tener que hacer el levantamiento de requerimientos necesarios para elaborar el sitio web que se solicita.

**¿Cuáles fueron las estrategias a seguir para resolver dichos problemas?**

Asignar otros días aparte de los permitidos en clases y estar en contacto mediante mensajes con los clientes.

BIBLIOGRAFÍAS

Cantavella, J. (27 de 05 de 2017). *Biblioteca Digital ECEST.* Obtenido de E-libro – La entrevista como conversación dirigida.: http://site.ebrary.com.ezproxy.bibliotecaecest.mx/lib/bidigecestsp/reader.action?docID=11245555

Ceballos Sierra, F. J. (Enero de 2006). *Biblioteca Digital ECEST.* Obtenido de E-brary - Java 2: lenguaje y aplicaciones: http://site.ebrary.com.ezproxy.bibliotecaecest.mx/lib/bidigecestsp/reader.action?docID=11045988

Haro, B. B. (s.f.). */site/brunoharobrayan/corel-draw-x5: sites.google.com*. Obtenido de sites.google.com: https://sites.google.com/site/brunoharobrayan/corel-draw-x5

Hernandez, I. y. (19 de Octubre de 2010). *balsamiq-mockups-como-herramienta-de-prototipado: blogtrw*. Obtenido de blogtrw: http://www.blogtrw.com

Jotatsu. (19 de Agosto de 2006). *review/staruml: black-byte*. Obtenido de black-byte: http://black-byte.com/review/staruml/

Kenneth E. Kendall , Julie E. Kendall. (2005). *Análisis y diseño de sistemas.* México: Pearson Educacion.

María, U. S. (s.f.). *modelo-de-prototipo: gestionrrhhusm*. Obtenido de gestionrrhhusm: http://gestionrrhhusm.blogspot.mx/2011/05/modelo-de-prototipo.html

McCann, W. J. (26 de Noviembre de 2013). *Apps/Dia: Wiki.gnome*. Obtenido de Wiki.gnome: https://wiki.gnome.org

Microsoft. (s.f.). *enterprise project server : products office.* Obtenido de products office: https://products.office.com/es-mx/project/enterprise-project-server

Mohedano, J., Saiz, J. M., & Salazar Román, P. (2012). *Digitalia Hispanica.* Obtenido de http://www.digitaliapublishing.com.ezproxy.bibliotecaecest.mx/a/20926/

Pascual González, F. (2004). *Adobe Photoshp CS y 7.* México: RA-MA Editorial.

PHP, G. E. (08 de Agosto de 2009). *definicion:modeloespiral*. Obtenido de modeloespiral: http://modeloespiral.blogspot.mx/

Puertas, J. P. (2004). *Creación de un portal con PHP y MYSQL.* México: RA-MA Editorial.

Rodriguez, S. (19 de Octubre de 2012). *Prezi.* Obtenido de Prezi: Ciclo de vida en Cascada y Cascada Retroalimentada: https://prezi.com/ey6st376ardk/ciclo-de-vida-en-cascada-y-cascada-retroalimentada/

Sommerville, I. (2005). *Ingeniería del Software.* Madrid: PEARSON EDUCACIÓN.

Valdés, D. P. (02 de Noviembre de 2007). *los-diferentes-lenguajes-de-programacion-para-la-web: maestrosdelweb*. Obtenido de maestrosdelweb: http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/