2.2 Marco teórico

Arquitectura

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende los componentes del software, sus propiedades externas visibles y las relaciones entre ellos.

La arquitectura no es el software operativo. Es una representación que permite 1) analizar la efectividad del diseño para cumplir los requerimientos establecidos, 2) considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y 3) reducir los riesgos asociados con la construcción del software.

Esta definición pone el énfasis en el papel de los “componentes del software” en cualquier representación arquitectónica. En el contexto del diseño de la arquitectura, un componente del software puede ser algo tan simple como un módulo del programa o una clase orientada a objeto, pero también puede ampliarse para que incluya bases de datos y “middleware” que permitan la configuración de una red de clientes y servidores. Las propiedades de los componentes son aquellas características necesarias para entender cómo interactúan unos componentes con otros. En el nivel arquitectónico, no se especifica las propiedades internas (por ejemplo, detalles de un algoritmo). Las relaciones entre los componentes pueden ser tan simples como una invocación de procedimiento de un módulo a otro o tan complejas como un protocolo de acceso a una base de datos. [6]

Base de datos relacional

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional.

Los matemáticos definen las relaciones como subconjuntos del producto cartesiano de la lista de dominios. Esta definición se corresponde de manera casi exacta con la definición de tabla presentada anteriormente. La única diferencia es que aquí se han asignado nombres a los atributos, mientras que los matemáticos sólo utilizan «nombres» numéricos, utilizando el entero 1 para denotar el atributo cuyo dominio aparece en primer lugar en la lista de dominios, 2 para el atributo cuyo dominio aparece en segundo lugar, y así sucesivamente. Como las tablas son esencialmente relaciones, se utilizarán los términos matemáticos relación y tupla en lugar de los términos tabla y fila. Una variable tupla es una variable que representa a una tupla; en otras palabras, una tupla que representa al conjunto de todas las tuplas. [7]

Conceptos y diagramas del lenguaje unificado de modelación (UML)

UML proporciona un conjunto estandarizado de herramientas para documentar el análisis y diseño de un sistema de software. El conjunto de herramientas de UML incluye diagramas que permiten a las personas visualizar la construcción de un sistema orientado a objetos, similar a la forma en que un conjunto de planos permite a las personas visualizar la construcción de un edificio. Ya sea que se trabaje independientemente o con un equipo grande de desarrollo de sistemas, la documentación que se crea con UML proporciona un medio eficaz de comunicación entre el equipo de desarrollo y el equipo de negocios en un proyecto.

UML consiste de cosas, relaciones y diagramas. Los primeros componentes, o elementos principales de UML se denominan cosas. Las cosas estructurales son más comunes, son clases, interfaces, casos de uso y muchos otros elementos que proporcionan una forma de crear modelos, también permiten al usuario describir relaciones. Las cosas de comportamiento describen cómo funcionan las cosas. Las interacciones y las máquinas de estado son ejemplos de cosas de comportamiento. Las cosas de agrupamiento se usan para definir límites. Un ejemplo de una cosa de agrupamiento es un paquete. Por último, se cuenta con las cosas de anotación, que permiten agregar notas a los diagramas.

Las relaciones son el pegamento que une las cosas. Es útil considerar a las relaciones de dos formas. Las relaciones estructurales se usan para enlazar las cosas en los diagramas estructurales. Las relaciones estructurales incluyen dependencias, agregaciones, asociaciones y generalizaciones. Por ejemplo, las relaciones estructurales muestran herencia. Las relaciones de comportamiento se usan en los diagramas de comportamiento. Los cuatro tipos básicos de relaciones de comportamiento son: comunica, incluye, extiende y generaliza.

Hay dos tipos principales de diagramas en UML: diagramas estructurales y diagramas de comportamiento. Por ejemplo, los diagramas estructurales se usan para describir las relaciones entre las clases. Incluyen diagramas de clases, diagramas de objetos, diagramas de componentes y diagramas de despliegue. Por otro lado, los diagramas de comportamiento se pueden usar para describir la interacción entre las personas (denominadas actores en UML) y la cosa a la que se refiere como caso de uso, o cómo usan los actores el sistema. Los diagramas de comportamiento incluyen diagramas de caso de uso, diagramas de secuencias, diagramas de colaboración, diagramas de gráfico de estado y diagramas de actividades.

Modelado del caso de uso

El UML está basado fundamentalmente en una técnica de análisis orientada a objetos conocida como modelado de casos de uso, en la cual la palabra uso se pronuncia como sustantivo en lugar de verbo. Un modelo de caso de uso describe lo que hace un sistema sin describir cómo lo hace; es decir, es un modelo lógico del sistema. El modelo de caso de uso refleja la vista del sistema desde la perspectiva de un usuario fuera del sistema (es decir, los requerimientos del sistema). El UML se puede usar para analizar el modelo de caso de uso para derivar objetos del sistema y sus interacciones entre sí y con los usuarios del sistema. Con el uso de las técnicas de UML, se analiza más a fondo los objetos y sus interacciones para derivar el comportamiento del objeto, sus atributos y relaciones.

Un analista desarrolla casos de uso en colaboración con los expertos del negocio que ayudan a definir los requerimientos del sistema. El modelo de caso de uso proporciona medios eficaces de comunicación entre el equipo del negocio y el equipo de desarrollo. Un modelo de caso de uso divide la funcionalidad del sistema en comportamientos, servicios y respuestas (los casos de uso) que son significativos para los usuarios del sistema.

Desde la perspectiva de un actor (o usuario), un caso de uso debe producir algo que es de valor. Por lo tanto, el analista debe determinar lo que es importante para el usuario y recordar incluirlo en el diagrama de caso de uso.

Diagramas de actividades

Los diagramas de actividades muestran las secuencias de actividades de un proceso, entre las que se encuentran las actividades secuenciales, las actividades paralelas y las decisiones que se toman. Por lo general, un diagrama de actividades se elabora para un caso de uso y podría reflejar los diferentes escenarios posibles.

Diagramas de clase

Las metodologías orientadas a objetos se enfocan en descubrir clases, atributos y métodos y relaciones entre las clases. Puesto que la programación se realiza al nivel de la clase, la definición de clases es una de las tareas más importantes del análisis orientado a objetos. Los diagramas de clases muestran las características estáticas del sistema y no representan ningún procesamiento en particular. Un diagrama de clases también muestra la naturaleza de las relaciones entre las clases.

Diagramas de estados

El diagrama de estados, o de transición de estados, es otra manera de determinar los métodos de una clase. Se usa para examinar los diferentes estados que podría tener un objeto. Un diagrama de estados se crea para una sola clase. Por lo general, los objetos se crean, sufren cambios y se eliminan.

Los objetos existen en cualquiera de estos estados, que son las condiciones de un objeto en un momento específico. [3]

Sistema Gestor de Base de Datos

Un sistema gestor de bases de datos (SGBD o DBMS por sus siglas en inglés) consiste en una colección de datos interrelacionados y una colección de programas para acceder a esos datos. Los datos describen una empresa particular.

El objetivo principal de un SGBD es proporcionar un entorno que sea tanto conveniente como eficiente para las personas que lo usan para la recuperación y almacenamiento de la información.

Los sistemas de bases de datos se diseñan para almacenar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para el almacenamiento de la información como la provisión de mecanismos para la manipulación de la misma. Además, los sistemas de bases de datos deben proporcionar la seguridad de la información almacenada, en caso de caídas del sistema o intentos de accesos sin autorización. Si los datos están compartidos por varios usuarios, el sistema debe evitar posibles resultados anómalos.

Un propósito principal de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo los datos se almacenan y mantienen.

Por debajo de la estructura de la base de datos está el modelo de datos: una colección de herramientas conceptuales para describir los datos, las relaciones entre los mismos, su semántica y restricciones. [7]

Diseño web adaptable

El diseño web adaptable (o responsive web design) corresponde a una tendencia de creación de páginas web que pueden ser visualizadas perfectamente en todo tipo de dispositivos, desde ordenadores de escritorio hasta smartphones o tablets. Con este tipo de diseño no se debe tener una versión para cada dispositivo, ya que una sola web se adapta a todos ellos.

En lugar de construir una aplicación para cada tipo de dispositivo (smartphone, tablet, ordenador desktop, laptop e incluso, hoy en día, para smart TV), se crea un solo sitio web con CSS3 media queries y un layout con imágenes flexibles. De esta forma, el la aplicación detecta qué clase de dispositivo utiliza el usuario y muestra la versión más optimizada para ese medio, a través de la reorganización de los elementos de la web e incluso la discriminación de algunos de ellos (menos imágenes, más ligeras, redistribución de las columnas en el diseño, menos texto, entre otros).

HTML5 permite una experiencia excelente para los usuarios, sin el coste de desarrollar una aplicación nativa para cada dispositivo.

Es fácil de manejar ya que solamente se trabaja sobre un sitio web y los resultados serán uniformes en todas las plataformas sin distorsiones en las imágenes o texto. [18]

HTML—Lenguaje de Marcado de Hipertexto

En la actualidad las páginas Web se escriben en un lenguaje llamado HTML (HiperText Markup Language o traducido como Lenguaje de Marcado de Hipertexto). HTML permite a los usuarios producir páginas web que incluye texto, gráficos y apuntadores a otras páginas web. HTML es un lenguaje de marcado que sirve para describir cómo se van a formatear los documentos. El término “marcado” proviene de la época en que los correctores de estilo realmente marcaban los documentos para indicar a la imprenta —en aquellos tiempos, un humano— qué fuentes utilizar y cosas por el estilo. Por lo tanto, los lenguajes de marcado contenían comandos explícitos para formatear. Por ejemplo, en HTML, <b> significa iniciar modo en negritas y </b> significa abandonar modo en negritas. La ventaja de un lenguaje de marcado sobre uno con marcado no explícito es que escribir un navegador para él es directo: el navegador simplemente tiene que entender los comandos de marcado. TeX y troff son ejemplos de otros lenguajes de marcado bien conocidos.

Al integrar todos los comandos de marcado dentro de cada archivo HTML y al estandarizarlos, se hace posible que cualquier navegador Web lea y reformatee cualquier página de éste tipo. La capacidad de reformatear las páginas tras su recepción es crucial porque una página pudo haberse producido en una ventana de 1600 × 1200 con colores de 24 bits pero tal vez se vaya a desplegar en una de 640 × 320 con colores de 8 bits. [9]

Modelos de proceso prescriptivo: modelo de la cascada

Hay veces en las que los requerimientos para cierto problema se comprenden bien: cuando el trabajo desde la **comunicación** hasta el **despliegue** fluye en forma razonablemente lineal. Esta situación se encuentra en ocasiones cuando debe hacerse adaptaciones o mejoras bien definidas a un sistema ya existente (por ejemplo, una adaptación para software de contabilidad que es obligatorio hacer debido a cambios en las regulaciones gubernamentales). También ocurre en cierto número limitado de nuevos esfuerzos de desarrollo, pero sólo cuando los requerimientos están bien definidos y tienen una estabilidad razonable.

El modelo de la cascada, a veces llamado ciclo de vida clásico, sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de la planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado. [6]

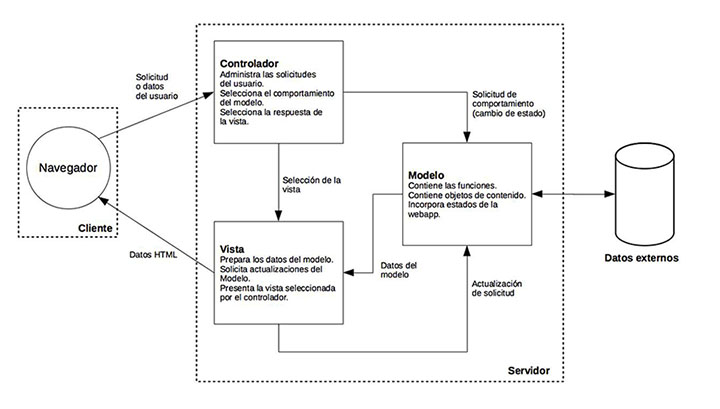
Modelo-Vista-Controlador

La arquitectura de una aplicación web describe una infraestructura que permite que un sistema o aplicación basados en web alcance sus objetivos empresariales. Jacynto et al. [2] describe las características básicas de esta infraestructura del modo siguiente:

*Las aplicaciones deben construirse con el empleo de capas en las que se tomen en cuenta distintas preocupaciones; en particular, deben separarse los datos de la aplicación de los contenidos de ésta (nodos de navegación), y éstos, a su vez, deben separarse con toda claridad del aspecto y la sensación de la interfaz (páginas).*

Los autores sugieren una arquitectura del diseño en tres capas que desacopla la interfaz de la navegación y del comportamiento de la aplicación. Plantean que mantener separadas las interfaz, la aplicación y la navegación, simplifica la implementación y mejora la reutilización.

La arquitectura *modelo-vista-controlador* (MVC) [4] es uno de varios modelos sugeridos para la infraestructura de aplicaciones web que desacoplan la interfaz de usuario de sus funciones y contenido informativo. El ***modelo*** (a veces denominado “objeto de modelo”) contiene todo el contenido y la lógica de procesamiento específicos de la aplicación, incluso todos los objetos de contenido, acceso a fuentes de datos o información externos y todas las funciones de procesamiento que son específicas de la aplicación. La ***vista*** contiene todas las funciones específicas de la interfaz y permite la presentación del contenido y lógica del procesamiento, incluidos todos los objetos de contenido, el acceso a fuentes de datos o información del exterior y todas las funciones de procesamiento que requiere el usuario final. El ***controlador*** administra el acceso al modelo y la vista, y coordina el flujo de datos entre ellos. En una aplicación web, “la vista es actualizada por el controlador con datos del modelo, basándose en las entradas que da el usuario” [10]. En la figura 2.1 se muestra una representación de la arquitectura MVC.



**Figura 2.1.** La arquitectura CVM [2].

En relación con la figura, el controlador maneja las solicitudes o datos del usuario. El controlador también selecciona el objeto de vista que sea aplicable con base en la solicitud del usuario. Una vez determinado el tipo de solicitud, se transmite al modelo un pedido de comportamiento, que implementa la funcionalidad o recupera el contenido requerido para dar acomodo a la solicitud. El objeto de modelo accede a los datos almacenados en una base de datos corporativa, como parte de un almacén de datos locales o como una colección de archivos independientes. El objeto de vista apropiado debe dar formato y organizar los datos desarrollados por el modelo para luego transmitirlos desde el servidor de la aplicación hacia el navegador del cliente para que se desplieguen en la máquina de éste. [6]

Navegador web

En esencia, un navegador es un programa que puede desplegar una página web y atrapar los clics que se hacen en los elementos de la página desplegada. Cuando se selecciona un elemento, el navegador sigue el hipervínculo y obtiene la página seleccionada. Por lo tanto, el hipervínculo incrustado necesita una manera de nombrar cualquier página que se encuentre en web. Las páginas se nombran con el uso de URLs (Uniform Resource Locator o traducido como Localizadores Uniformes de Recursos). Un URL típico es http://www.abcd.com/productos.html

Cuando un usuario hace clic en un hipervínculo, el navegador lleva a cabo una serie de pasos para obtener la página a la que se está apuntado. Suponga que un usuario navega en la web y encuentra un vínculo sobre telefonía de Internet que apunta a la página de inicio de la ITU (International Telecommunication Union o traducido como Unión Internacional de Telecomunicaciones), que es http://www.itu.org/home/index.html. Los pasos que se dan cuando se selecciona este vínculo son:

1. El navegador determina el URL (enviando lo que se seleccionó).
2. El navegador pide al DNS (Domain Name System o traducido como Sistema de Nombres de Dominio) la dirección IP de www.itu.org.
3. DNS responde con 156.106.192.32.
4. El navegador realiza una conexión TCP (Transmission Control Protocol o en español “Protocolo de control de transmisión”) con el puerto 80 en 156.106.192.32.
5. Después envía un mensaje en el que solicita el archivo /home/index.html.
6. El servidor www.itu.org envía el archivo /home/index.html.
7. Se libera la conexión TCP.
8. El navegador despliega todo el texto de /home/index.html.
9. El navegador obtiene y despliega todas las imágenes del archivo.

Para poder desplegar la nueva página (o cualquiera), el navegador tiene que entender su formato. Para permitir que todos los navegadores entiendan todas las páginas web, éstas se escriben en un lenguaje estandarizado (HTML), el cual las describe.

Aunque un navegador es básicamente un intérprete HTML, la mayoría de los navegadores tiene varios botones y características para facilitar la navegación en la web. La mayoría tiene un botón para regresar a la página anterior, uno para ir a la siguiente (el cual sólo funciona una vez que el usuario ha regresado de esa página) y uno para ir directamente a la página de inicio del usuario. La mayoría de los navegadores tiene un botón o elemento para establecer un marcador y otro para desplegar la lista de marcadores, lo que hace posible volver a visitar cualquiera de ellos con sólo algunos clics del ratón. Las páginas también pueden guardarse en disco o imprimirse. Por lo general, hay varias opciones disponibles para controlar el diseño de la pantalla y establecer varias preferencias de usuario.

Además de tener texto ordinario (que no está subrayado) e hipertexto (texto subrayado), las páginas web también pueden tener iconos, dibujos de líneas, mapas y fotografías. Cada uno de éstos puede (opcionalmente) vincularse a otra página. Hacer clic en alguno de estos elementos causa que el navegador obtenga la página vinculada y la despliegue en pantalla, al igual que al hacer clic en el texto. En el caso de imágenes como fotos y mapas, la página que se obtenga a continuación depende de en cuál parte de la imagen se hizo clic. [9]

Mapeo Objeto-Relacional

El mapeo objeto-relacional (ORM siglas en inglés de Object Relational Mapping) es un mecanismo que hace posible direccionar, acceder y manipular objetos sin tener que considerar cómo aquellos objetos se relacionan con sus orígenes de datos. Permite a los programadores mantener una vista consistente de los objetos en el tiempo, incluso si las fuentes que los entregan, los recipientes que los reciben y las aplicaciones que acceden a ellos cambian.

Basado en la abstracción, ORM administra los detalles de mapeo entre un conjunto de objetos y bases de datos relacionales subyacentes, repositorios XML (siglas en inglés de eXtensible Markup Language o “Lenguaje de marcas Extensible”) u otros orígenes de datos y recipientes, al tiempo que oculta los a menudo cambiantes detalles relacionados con las interfaces de los desarrolladores y el código que ellos crean.

ORM oculta y encapsula cambios en el mismo origen de datos, así que cuando los orígenes de datos o sus APIs (del inglés Application Programming Interface o “Interfaz de programación de aplicaciones”) cambian, solo el ORM necesita cambiar para mantenerse actualizado - no las aplicaciones que usan ORM para aislarse a sí mismas de ésta clase de esfuerzo. Esta capacidad permite a los desarrolladores tomar ventaja de nuevas clases a medida que estén disponibles y también hace que sea fácil extender las aplicaciones basadas en ORM. En muchos casos, los cambios en ORM pueden incorporar nueva tecnología y capacidad sin requerir cambios en el código de las aplicaciones relacionadas. [20]

Servidor web

Cuando el usuario teclea un URL o hace clic en una línea de hipertexto, el navegador lo analiza e interpreta la parte entre http:// y la siguiente diagonal como un nombre DNS a buscar.

Una vez que el navegador tiene la dirección IP del servidor, establece una conexión TCP con el puerto 80 de ese servidor. A continuación envía un comando que contiene el resto del URL, que es el nombre del archivo que se encuentra en ese servidor. Éste regresa el archivo para que el navegador lo despliegue.

Los pasos que da el servidor en su ciclo principal son:

1. Acepta una conexión TCP de un cliente (un navegador).
2. Obtiene el nombre del archivo solicitado.
3. Obtiene el archivo (del disco).
4. Regresa el archivo al cliente.
5. Libera la conexión TCP.

Los servidores Web actuales tienen más características, pero en esencia, esto es lo que hacen.

Los servidores web modernos hacen más que sólo aceptar y regresar nombres de archivos. De hecho, el procesamiento real de cada solicitud puede ser muy complicado. Por esta razón, en muchos servidores cada módulo de procesamiento realiza una serie de pasos. El front end pasa cada solicitud entrante al primer módulo disponible, que después la transporta mediante algunos de los siguientes pasos, según sean necesarios para esa solicitud en particular.

1. Resuelve el nombre de la página web solicitada.
2. Autentica al cliente.
3. Realiza control de acceso en el cliente.
4. Realiza control de acceso en la página web.
5. Verifica el caché.
6. Obtiene del disco la página solicitada.
7. Determina el tipo MIME (Multipurpose Internet Mail Extensions o en español “extensiones multipropósito de correo de internet”) que se incluirá en la respuesta.
8. Se encarga de diversos detalles.
9. Regresa la respuesta al cliente.
10. Realiza una entrada en el registro del servidor. [9]

CSS3

Hojas de estilo en cascada. Son usadas para dar formato al diseño de páginas web. Pueden ser usadas para definir estilos de texto, tamaños de tabla, y otros aspectos de las páginas web que previamente sólo podían ser definidos en el código HTML de una página.

CSS ayuda a los desarrolladores web a crear una vista uniforme a través de muchas páginas de un sitio web. En lugar de definir el estilo de cada tabla y cada bloque de texto dentro del HTML de una página, comúnmente los estilos utilizados deben ser definidos una sola vez en un documento CSS. Una vez que el estilo es definido en una hoja de estilo en cascada, puede ser utilizado por cualquier página que haga referencia al archivo CSS. Además, CSS hace fácil cambiar estilos a través de muchas páginas web al mismo tiempo. [12]

**CSS3** es la última evolución del lenguaje de hojas de estilo en cascada y aspira a extender CSS2.1. Trae una gran cantidad de novedades esperadas por mucho tiempo, como esquinas redondeadas, sombras, gradientes, transiciones o animaciones, así como también nuevos diseños como multi-columnas, cuadros flexibles o diseños de cuadrícula. [13]

HTML5

HTML 5 es una revisión del lenguaje de marcado de hipertexto, el lenguaje de programación estándar para describir los contenidos y apariencia de páginas web.

HTML5 fue desarrollado para resolver problemas de compatibilidad que afectan el actual estándar, HTML4. Una de las más grandes diferencias entre HTML5 y versiones previas del estándar es que las versiones antiguas de HTML requieren plugins propietarios y APIs (esta es la razón por la cual una página Web que fue construida y probada en un navegador puede no cargar correctamente en otro navegador.) HTML5 provee una interfaz común para hacer la carga de elementos más fácil. Por ejemplo, no hay necesidad de instalar un plugin Flash porque el elemento funcionará por sí mismo. [19]

PHP

PHP acrónimo de "*PHP: Hypertext Preprocessor*", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. La meta principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas. [15]

Apache web server

El servidor HTTP apache o web server apache es una implementación de código abierto de un servidor para el protocolo HTTP/1.1 multiplataforma, modular, extensible, ampliamente soportado y documentado. [11]

Ventajas

1. Instalación/Configuración. Software de código abierto.
2. Coste. El servidor web Apache es completamente gratuito.
3. Funcional y Soporte. Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.
4. Multi-plataforma. Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
5. Rendimiento. Capacidad de manejar más de un millón de visitas/día.
6. Soporte de seguridad SSL y TLS. [14]

MySQL

MySQL es un sistema gestor de bases de datos relacionales (DBMS por sus siglas en inglés) propiedad de Oracle Corporation. Es desarrollado como software libre con dos tipos de licenciamiento: libre y propietario, lo que permite utilizarlo tanto en proyectos de software de código abierto y cerrado.

Este DBMS provee características como multihilo, multiusuario, facilidad de uso, instalación y configuración, lenguajes DDL y DML basados en el estándar ANSI/ISO SQL, una gran cantidad de documentación y soporte de parte de la comunidad y de la empresa desarrolladora. [21]

Doctrine ORM

Doctrine ORM es un Mapeador Objeto-Relacional (ORM) escrito en y para PHP que se sitúa sobre una poderosa Capa de Abstracción de Base de datos (DBAL por sus siglas en inglés). Una de sus características clave es la opción de escribir consultas de base de datos en un dialecto SQL orientado a objetos llamado Doctrine Query Language (DQL), inspirado por Hibernates HQL. Esto provee a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad sin duplicación de código. [16]

Symfony

Es un marco de trabajo (framework) para el desarrollo de aplicaciones web dinámicas con lenguaje de programación PHP, basado en el patrón Modelo-Vista-Controlador, con el cual la interacción con la base de datos, la lógica del negocio y la presentación se encuentran separadas. [22]

Twitter Bootstrap

Twitter Bootstrap es un framework HTML, CSS y javascript especialmente diseñado para la construcción de sitios web que pueden visualizarse en cualquier dispositivo, y cuya prioridad son los dispositivos móviles. Para dicho fin provee a los desarrolladores de una gran cantidad de plantillas y componentes adicionales. [17]

Arquitectura

En una aplicación web, el diseño arquitectónico del contenido se relaciona con la forma en que las distintas páginas de las que se compone el programa, se estructuran para la presentación y la navegación.

Arquitectura centrada en los datos

“*En el centro de esta arquitectura se halla un almacenamiento de datos (como un archivo o base de datos) al que acceden con frecuencia otros componentes que actualiza, agregan, eliminan o modifican datos de cierto modo dentro del almacenamiento.*

*Las arquitecturas centradas en datos promueven la integrabilidad. Es decir, los componentes del software pueden ser cambiados y agregarse a otros nuevos, del cliente, a la arquitectura sin problemas con otros clientes (porque los componentes del cliente operan en forma independiente).*” [6]

Arquitectura en capas

“*Se define un número de capas diferentes; cada una ejecuta operaciones que se aproximan progresivamente al conjunto de instrucciones de máquina. En la capa externa, los componentes atienden las operaciones de la interfaz de usuario. En la interna, los componentes realizan la interfaz con el sistema operativo. Las capas intermedias proveen servicios de utilerías y funciones de software de aplicación.*” [6]