Universidad Internacional de las Américas



Estructuras y Algoritmos

IS-21

Proyecto del Curso:

Atención al Cliente

Estudiantes:

María Fernanda Salas Arce

Carolina Vega

Roger Rivera

Diego Méndez

Profesor:

Lic. David Quesada Chacón

Fecha de Entrega: martes 16 de agosto del 2016

Valor del proyecto: 40%

INDICE

[Introducción 4](#_Toc458983938)

[Objetivo General 5](#_Toc458983939)

[Objetivos Específicos 5](#_Toc458983940)

[Marco Teórico 6](#_Toc458983941)

[Programación Orientada a Objetos 6](#_Toc458983942)

[Diagrama de Clases 6](#_Toc458983943)

[HTML 6](#_Toc458983944)

[CSS 6](#_Toc458983945)

[JavaScript 6](#_Toc458983946)

[JQuery 6](#_Toc458983947)

[JSON 7](#_Toc458983948)

[Programación Orientada a Eventos 7](#_Toc458983949)

[AJAX 7](#_Toc458983950)

[Alcances 8](#_Toc458983951)

[Limitaciones 8](#_Toc458983952)

[Desarrollo 9](#_Toc458983953)

[Programación Orientada a Objetos 9](#_Toc458983954)

[Origen 9](#_Toc458983955)

[¿Qué es la POO? 9](#_Toc458983956)

[Características de la POO 9](#_Toc458983957)

[Ventajas 10](#_Toc458983958)

[Desventajas 10](#_Toc458983959)

[Diagrama de Clases 11](#_Toc458983960)

[Clases 11](#_Toc458983961)

[Portafolio Personal 12](#_Toc458983962)

[Vista Usuario 12](#_Toc458983963)

[Vista Administrativa 12](#_Toc458983964)

[Programación Implementada 13](#_Toc458983965)

[Mejoras 13](#_Toc458983966)

[Conclusión 14](#_Toc458983967)

[Glosario 15](#_Toc458983968)

[Bibliografía 16](#_Toc458983969)

[Anexos 17](#_Toc458983970)

[Ilustración 1 17](#_Toc458983971)

[Ilustración 2 17](#_Toc458983972)

[Ilustración 3 18](#_Toc458983973)

[Ilustración 4 18](#_Toc458983974)

# Introducción

El trabajo realizado consiste en la creación de un portafolio personal, donde, se muestra la información de cada persona, que desee mostrar, por ejemplo, los pasatiempos, una galería de fotos, una sección de información personal, logros personales, entre otras cosas. Básicamente el portafolio es dinámico, ya que los efectos empleados en su mayoría son de las galerías de JavaScript, las cuales son realmente útiles, gráficamente hablando.

# Objetivo General

* Evaluar los conocimientos adquiridos durante el cuatrimestre para la elaboración de un portafolio personal.

# Objetivos Específicos

* Analizar la importancia de un portafolio personal.
* Utilizar la lógica necesaria para la implementación de cada una de las funcionalidades especificadas, que permitan a los usuarios visualizar y entender apropiadamente el portafolio personal creado.
* Implementar de manera local con un servidor web configurado correctamente.
* Implementar los temas vistos en clase de HTML, CSS, JavaScript y PHP.
* Gestionar la información desde una base de datos MySQL.

# Marco Teórico

Para la realización de este trabajo es necesario el conocimiento de ciertos conceptos de suma importancia, los cuales deben ser implementado para obtener resultados eficientes, algunos conceptos serían los siguientes:

## Programación Orientada a Objetos

Entre los más importantes y principales se encuentra la Programación Orientada a Objetos (POO), la cual es un paradigma de programación que define los programas en términos de “clases de objetos”, donde los objetos son entidades que combinan estados (es decir, datos), comportamientos (esto es, procedimientos o métodos) e identidades (propiedades del objeto que lo diferencia del resto). Algunas características de la POO serían: Abstracción, Encapsulamiento, Herencia y Polimorfismo. (Garita, 2015)

## Diagrama de Clases

Algo importante es la utilización de los Diagramas de Clases, que sirve para visualizar las relaciones entre las clases que involucra el sistema, las cuales pueden ser asociativas, de herencia, de uso e incluso de contenimiento. Algunos de sus elementos más importantes son las clases, que incluyen atributos y métodos y luego están las relaciones entre clase, que usan la Herencia, Composición, Agregación, Asociación y Uso. (Salinas, 1996)

## HTML

Otro concepto que hay que tener en cuenta es el de HTML, que significa Lenguaje de Marcado para Hipertextos (HyperText Markup Language), el cual es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. (Muñoz, 2016)

## CSS

El concepto de HTML va de la mano con el concepto CSS que es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. (Hawke)

## JavaScript

JavaScript, a veces es abreviado como JS, este es un lenguaje ligero e interpretado, orientado a objetos con [funciones de primera clase](https://en.wikipedia.org/wiki/First-class_functions), más conocido como el lenguaje de script para páginas web, pero también [usado en muchos entornos sin navegador](http://en.wikipedia.org/wiki/JavaScript#Uses_outside_web_pages), tales como  [node.js](http://nodejs.org/)o[Apache CouchDB](http://couchdb.apache.org/)**.** Además, se dice que es un lenguaje script multi-paradigma, [basado en prototipos](https://en.wikipedia.org/wiki/Prototype-based),  dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. (Jsx, 2016)

## JQuery

Este concepto es uno de los complementos más esenciales para el desarrollo web, facilita el desarrollo de aplicaciones enriquecidas del lado del cliente, en JavaScript, compatibles con todos los navegadores. JQuery no es un lenguaje, sino una serie de funciones y métodos de JavaScript. Por lo tanto, JavaScript es el lenguaje y jQuery es una librería que se puede usar opcionalmente si se quiere facilitar la programación en JavaScript. A veces se puede referir a jQuery como framework o incluso como un API de funciones, útiles en la mayoría de proyectos web. (Anonimo, desarroloweb.com, 2012)

## JSON

Este concepto se entiende como un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. Está constituido por una colección de pares de nombre/valor y por una lista ordenada de valores. (Anonimo)

## Programación Orientada a Eventos

Es un paradigma de la programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen. Para que un programa se pueda llamar orientado a eventos debe haber sido creado en un lenguaje de programación orientado a objetos, ya que cada objeto espera algún evento que realice el usuario sobre él. Se debe definir los eventos que manejarán su programa y las acciones que se realizarán al producirse cada uno de ellos, lo que se conoce como el administrador del evento. Los eventos soportados estarán determinados por el lenguaje de programación utilizado, por el sistema operativo e incluso por eventos creados por el mismo programador. Al comenzar la ejecución del programa se llevarán a cabo las inicializaciones y demás código inicial y a continuación el programa quedará bloqueado hasta que se produzca algún evento. (Garcia, 2011)

## AJAX

El término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". Ajax no es una tecnología en sí mismo, en realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes. Las tecnologías que forman AJAX son las siguientes:

* XHTML y CSS, para crear una presentación basada en estándares.
* DOM, para la interacción y manipulación dinámica de la presentación.
* XML, XSLT y JSON, para el intercambio y la manipulación de información.
* XMLHttpRequest, para el intercambio asíncrono de información.
* JavaScript, para unir todas las demás tecnologías.

La explicación anterior de las tecnologías que forman AJAX se muestra de manera gráfica en la Ilustración 1.

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario. El modelo tradicional se ilustra en la Ilustración 2 y en la Ilustración 3 se puede observar el nuevo modelo propuesto por AJAX.

(Eguiluz, 2006-2016)

# Alcances

Al realizar este proyecto se logró reforzar los conocimientos adquiridos durante el cuatrimestre, tanto en HTML, PHP, CSS, además se aumentaron otros en algunos temas que también fueron vistos durante las clases y necesarios para la realización de este trabajo, tales como JQuery, JSON, JavaScript, AJAX.

Además, se consiguió obtener nuevos conocimientos como lo es la creación de un carrusel de fotos y así realizar una galería de imágenes con un diseño y efecto diferente a los estudiados.

# Limitaciones

Una de las limitaciones que se presentaron fue el hecho de que no sé pudo terminar la parte lógica del botón para ocultar información que fue creado con ese fin, debido a la falta de tiempo que se presentó para poder terminarlo.

# Desarrollo

## Programación Orientada a Objetos

### Origen

Las etapas que se llevaron a cabo previamente al origen de la POO son las siguientes 4:

1. Lenguajes Ensambladores: La unidad de programación es la instrucción, compuesta de un operador y el operando. El nivel de abstracción que se aplica es muy bajo.
2. Lenguajes de Programación: Fortran, Algol, Cobol. Los objetos y operaciones del mundo real se podían modelar mediante datos y estructuras de control separadamente. En esta etapa el diseño del software se enfoca sobre la representación del detalle procedimental y en función del lenguaje elegido. Conceptos como: refinamiento progresivo, modularidad procedimientos y programación estructurada son conceptos básicos que se utilizan en esta etapa. Existe mayor abstracción de datos.
3. Se introducen en esta etapa los conceptos de abstracción y ocultación de la información.
4. A partir de los años setenta se trabaja sobre una nueva clase de lenguajes de simulación y sobre la construcción de prototipos tales como Simula-70 y basado en parte de éste, el Smalltalk. En estos lenguajes, la abstracción de datos tiene una gran importancia y los problemas del mundo real se representan mediante objetos de datos a los cuales se les añade el correspondiente conjunto de operaciones asociados a ellos. Términos como Abstracción de datos, objeto, encapsulación entre otros, son conceptos básicos sobre la que se fundamenta la POO.
5. Se llama así a veces a los lenguajes de la inteligencia artificial, se desarrolló a principios de os 80. Tiene mayor legibilidad de los programas, portabilidad, facilidad de aprendizaje y facilidad de modificación.

(Universidad de Murcia)

### ¿Qué es la POO?

La POO es un paradigma de programación que define los programas en términos de “clases de objetos”, objetos que son entidades que combinan estados (es decir, datos), comportamientos (esto es, procedimientos o métodos) e identidades (propiedades del objeto que lo diferencia del resto). (Garita, 2015)

### Características de la POO

Algunas características de la POO son las siguientes:

* Abstracción.
* Encapsulamiento.
* Herencia.
* Polimorfismo.

Abstracción: Se entiende por este término la capacidad de manejar un objeto como un concepto general, sin considerar la enorme cantidad de detalles que pueden estar asociados con dicho objeto. Lo realmente importante es destacar los aspectos importantes de un objeto y descartar los irrelevantes. (Garita, 2016)

Encapsulamiento: Este concepto significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la [cohesión](http://es.wikipedia.org/wiki/Cohesi%C3%B3n) de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente. (Gonzalez, 2008)

Herencia: Es la capacidad que tiene una clase de derivar las propiedades y métodos de otra. La herencia en programación nos permite, entre otras cosas, evitar tener que escribir el mismo código una y otra vez, puesto que al definir que una clase pertenece a otra, automáticamente estamos atribuyéndoles las características generales de la primera, sin tener que definirlas de nuevo. (Fabian, 2016)

Polimorfismo: Es la capacidad que tienen los objetos de una [clase](http://es.wikipedia.org/wiki/Clase_%28inform?tica%29) de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación. Un objeto polimórfico es una entidad que puede contener valores de diferentes tipos durante la ejecución del programa. El concepto de polimorfismo se puede aplicar tanto a funciones como a tipos de datos. Así nacen los conceptos de funciones polimórficas y tipos polimórficos. Las primeras son aquellas funciones que pueden evaluarse o ser aplicadas a diferentes tipos de datos de forma indistinta; los tipos polimórficos, por su parte, son aquellos tipos de datos que contienen al menos un elemento cuyo tipo no está especificado. (Fabian, 2016)

Por otro lado, se sabe que, durante años, programadores se han dedicado a construir aplicaciones muy parecidas a la POO que resolvían una y otra vez los mismos problemas. Para conseguir que los esfuerzos de los programadores pudieran ser utilizados por otras personas se creó la POO, la cual es bien conocida por ser una serie de normas para realizar las cosas de una forma que otras personas pudieran utilizarlas y adelantar su trabajo, de manera que se logre conseguir que el código sea reutilizable. La POO no es difícil, pero es una manera especial de pensar, a veces subjetiva de quien la programa, de manera que la forma de hacer las cosas puede ser diferente según el programador. (Alvarez, 2001)

### Ventajas

Algunas ventajas que tiene la utilización de la POO son las siguientes:

* Los objetos pueden poseer atributos inherentes de la clase a la que pertenecen.
* Escribir códigos es más fácil porque se pueden conceptualizar como algo que le sucede a un objeto.
* La POO hace que los programas grandes sean más manejables.
* Si todas las ventanas pertenecen a una jerarquía de clases de ventanas y todo el código que se refiere a una ventana particular está dentro de esa ventana, todas las manipulaciones de ventana se pueden escribir como una sencilla transferencia de mensajes.

(Mano, 2014)

### Desventajas

Algunas desventajas que conlleva la POO son las siguientes:

* No todos los programas pueden ser modelados con exactitud por el modelo de objetos.
* Si lo que deseas es leer algunos datos, hacerles algo simple y escribir de nuevo, no tienes necesidad de definir clases y objetos. Sin embargo, en algunos lenguajes de POO, puede que tengas que realizar este paso extra.
* Si se fuerza el lenguaje en el concepto de programación orientada a objetos, se pierden algunas de las características de lenguajes útiles, como los "lenguajes funcionales".
* El concepto que un programador tiene de lo que constituye un objeto abstracto puede no coincidir con la visión de otro programador.
* Los objetos a menudo requieren una extensa documentación.

(Mano, 2014)

## Diagrama de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucra el sistema, las cuales pueden se asociativas, de herencia, de uso y de contenimiento. Un diagrama de clases está compuesto por los siguientes elementos:

* Clase4: Atributos, métodos y visibilidad.
* Relaciones entre clases: Herencia, Composición, Agregación, Asociación y Uso.

(Salinas, 1996)

### Clases

El conocimiento de las clases en Java es fundamental, ya que todo gira entorno a ellas. Un buen conocimiento de ellas permitirá realizar programas que cumplan con lo que se desea. Existen 4 tipos de clases:

* Clases Públicas: Son accesibles desde cualquier otra clase, lo que permite trabajar con ellas fácilmente. Sin embargo, para acceder a ellas se debe tener acceso al “package” al que pertenece. Se puede acceder a ellas directamente cuando están en el mismo paquete o por herencia importándolas con la sentencia “import”.
* Clases Abstractas: Estas clases se dejan incompletas para después ser refinadas mediante el mecanismo de la herencia, se puede fijar un conjunto de métodos y atributos, que serán especializados más adelante. Trabajan con los métodos abstractos, los cuales están vacíos. Esto es muy útil cuando la implementación es específica, pero cada usuario puede implementarlos como le parezca, sin embargo, sus métodos deben tener los mismos nombres que se dieron antes.
* Clase Final: es la que acaba con una cadena de herencia. De la clase final no se puede heredar nada. Al decir que termina la cadena de herencia significa que ya no se sub-divide en más clases, pues indica que es el final delas divisiones. Algunos motivos para crear este tipo de clases son los siguiente:
  1. El motivo de seguridad: Cuando no se quiere que otra persona haga sub-clases en base a la clase que se ha creado y haga nuevas instancias de ésta. Da un control sobre la clase evitando que le introduzcan funcionamientos anormales. Las librerías de java usan mucho el “final”, como las clases Math, InetAddressand y Socket. Es una forma de seguridad de java para que no las alteren.
  2. Eficiencia: Se trabaja solo con instancias de la clase sin tener que ir a las sub-clases, así se puede optimizar esa clase. Si la clase es “final”, entonces implícitamente todos sus métodos son “final” también y no hay sobrecarga de métodos. Además, los programas pueden correr hasta dos veces más rápido si la clase es “final”.
* Clase Synchronizable: Los métodos que estén definidos dentro de esta clase serán sincronizados. Se debe saber que la computadora puede llevar a cabo varios threads a la vez. Porque al ejecutar, diversas partes del mismo se pueden estar llevando acabo al mismo tiempo. Un thread puede interrumpir a otro en cualquier momento, se debe sincronizar todas las partes del programa que se están llevando a cabo. Al estar sincronizados los métodos dentro de esta clase, mientras uno se lleva acabo, el otro tiene que esperar a que acabe el primero antes de comenzar a ejecutarse.

(Cueva, 2007)

## Portafolio Personal

### Vista Usuario

En el portafolio personal creado se muestra información básica, pasatiempos, logros, hasta una galería de imágenes de la persona a la cual pertenece la administración de la misma. En el portafolio se puede viajar de una ventana a otra, por medio de un menú de opciones para navegar. La ventana Inicio es donde se muestra la información principal del portafolio, la cual tiene como objetivo dar la bienvenida a los usuarios que ingresen a ver el sitio. Desde dicha ventana de inicio se puede navegar y ver otra de Pasatiempos, donde, se le muestra al usuario los pasatiempos, de los cuales la persona del portafolio disfruta realizar, por medio de imágenes representativas de cada uno de ellos.

Otra ventana es la de Escolaridad, en donde, se le muestra al usuario una cantidad de información sobre las escolaridades que la persona ha tenido, empezando desde la escuela y colegio, a los que asistió, luego la Universidad a la que asiste o asistió. Por otro lado, se encuentra una venta Logros, la cual muestra al usuario todos aquellos logros que la persona ha cumplido en su vida, los cuales le parecen de importancia en su vida y que ha decidido compartir con los usuarios visitantes en el sitio.

Otra ventana es Información, donde se presenta la información básica y personal de la persona administradora del portafolio, la cual consiste en su nombre, apellidos, edad, trabajo, dirección, correo, entre otros datos, algunos de estos tienen un sentido obligatorio para llenar sus espacios correspondientes y esos son los que siempre se le van a mostrar al usuario, los demás se mostraran solo si el administrador así lo desea, este puede decidir entre si llenar el campo con la información o no.

Finalmente, está la ventana Galería, en la cual se muestra un carrusel de imágenes, en el cual se muestran fotos que la persona desea hacer públicas, en dicho carrusel se muestra la foto que el usuario selecciones en la parte inferior del mismo.

### Vista Administrativa

Por otro lado, desde la ventana Inicio, en caso de que el usuario sea administrador, puede dar clic al final del sitio en una etiqueta “Aquí” y luego debe iniciar sesión, en caso de que los datos sean erróneos muestra un mensaje al igual que si intenta abrir una página de administración sin iniciar sesión antes, es decir, si alguien tiene conocimiento del nombre asignado a cierta página administrativa e intenta ingresar desde el URL, se le muestra un mensaje de acceso denegado.

Una vez ingresado a la parte administrativa, se le da la bienvenida a la persona que ingresó desde ese index el administrador puede viajar a través del navegar al igual que el usuario en el portafolio, sin embargo, siendo administrador se muestra diferente, ya que en estas ventanas administrativas se ve con los espacios necesarios para que el administrador digite la información, imágenes y demás que desea mostrar, o en su defecto modificar.

Por ejemplo, las ventanas Información y Dirección se muestran una serie de espacios para digitar texto, otro caso es el de la ventana Escolaridad y Logros, en las cuales la dinámica es diferente, ya que para insertar información nueva hay que indicar que es nueva y luego agregar la información, pero si es modificar o eliminar se utilizan los botones que están después de la información.

Otra ventana es la Galería hay dos botones, dos opciones, uno para buscar y seleccionar las imágenes y el otro para realizar los cambios en la base de datos y lo que se muestra en el portafolio público. Finalmente, hay una opción en el navegador llamada Logout, que sirve para cerrar la sesión de administrador una vez realizados los cambios.

## Programación Implementada

Para el proyecto realizado se utilizó los componentes vistos en clase y para su funcionamiento básico se da la obtención de la información guardada desde la base de datos, esto mediante el id de la persona dueña de la administración del portafolio abierto. Para la parte grafica del proyecto se utilizaron diseño propiamente de JavaScript.

En la parte administrativa del portafolio se utiliza dos tipos de transmisiones el sincrónico, el cual realiza un request a otras páginas, cada vez que hace updates (modificaciones) de la información en la base de datos, luego devuelve todo lo relacionado al id de la persona administradora del portafolio o un error, en caso de que lo haya, por ejemplo, la ventana Información Básica y Dirección tiene una transmisión sincrónica y el asincrónico no realiza request, este lo hace todo a tiempo real, por explicarlo de cierta forma, es decir, lo emplea mediante JavaScript, el cual tiene los métodos necesarios para la obtención de la información y la muestra sin necesidad de un request, por ejemplo, las ventanas Pasatiempos, Logros y Escolaridad utilizan transmisión asincrónica . Por otro lado, la ventana Galería usa ambos tipos de transmisión, ya que usa AJAX para cargar las imágenes en la ventana de Configuración de la Galería y usa transmisión sincrónica (requests) para subir la foto, debido a que primero la sube a la base de datos y luego valída para mostrarla en el portafolio.

## Mejoras

Algunas mejoras que se podrían realizar sería un mejor manejo de los mensajes de errores a mostrar, en caso de que existan, además en la parte administrativa del portafolio se podría completar la parte lógica para un botón de ocultar información que se encuentra en varias de las ventanas de configuraciones, el cual el administrador podrá utilizar en caso de que quiera ocultar alguna información al usuario.

# Conclusión

En el trabajo realizado y explicado en este documento se logró cumplir con todos los objetivos planteados desde el comienzo, se analizó la importancia del portafolio personal que se quería crear, se utilizó la lógica necesaria para las funcionalidades que posee el portafolio, además, se implementaron los temas vistos en clase durante todo el cuatrimestre, tales como HTML, CSS, JavaScript, PHP , JSON, AJAX y también se consiguió la gestión de la información desde una base de datos creada en MySQL Workbench.

# Glosario

* POO: Abreviación para Programación Orientada a Objetos.
* Principio de Ocultación: El objeto está aislado del exterior, así protege las propiedades del objeto contra su modificación de quien no tenga derecho de acceder a ellas.
* Objeto: Es una instancia de una clase. Todo aquellos a lo que se le puede sacar características.
* Clase: Es la unidad básica que encapsula toda la información de un Objeto. A través de ella podemos modelar el entorno en estudio.
* Atributos: So las características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno.
* Métodos: Son la forma en como ésta interactúa con su entorno. Pueden ser de tipo public, private, protected.
* Herencia: Indica que una subclase hereda los métodos y atributos especificados por una Super Clase, por ende, la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (public y protected).
* Composición: Tipo de relación estática, donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye.
* Agregación: Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye.
* Asociación: Permite asociar objetos que colaboran entre sí. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.
* Thread: Es un flujo de control para controlar la ejecución de un programa.
* Nodo: Elemento físico que existe en tiempo de ejecución y representa un recurso computacional con memoria y capacidad de procesamiento.
* Estereotipos: Anotaciones que acompaña a cualquier parte del diagrama de los componentes, sirve para dar mayor entendimiento de qué rol, función o tipo cumple con el nodo en el diagrama.
* Conexiones: Diferentes asociaciones entre los nodos, de tal manera que dan una explicación más detallada de como los diversos nodos del diagrama se comunican entre sí.
* Componentes: Parte física que se puede reemplazar, usada para modelar elementos que pertenecen a un nodo tales como archivos, ejecutables, librerías, etc.

# Bibliografía

Alvarez, M. A. (24 de 7 de 2001). *Desarrollo Web*. Obtenido de http://www.desarrolloweb.com/articulos/499.php

Anonimo. (19 de 9 de 2012). *desarroloweb.com*. Obtenido de http://www.desarrolloweb.com/manuales/manual-jquery.html

Anonimo. (s.f.). *Introducción a JSON*. Obtenido de http://www.json.org/json-es.html

Cueva, D. (8 de 2007). *Scribd.* Obtenido de https://www.scribd.com/doc/17041197/2-Tipos-de-Clases

Eguiluz, J. (2006-2016). *Introducción a AJAX.*

Fabian. (2016). *Scribd.* Obtenido de https://www.scribd.com/doc/15354850/CARACTERISTICAS-POO

Garcia, V. (13 de 12 de 2011). *Programación en JAVA*. Obtenido de https://programarjava.wordpress.com/2011/12/13/programacion-orientada-a-eventos/

Garita, C. (2015). *Cursos de la carrera Ingeniería en Sistemas*. Obtenido de http://moodle.realdesigncr.com/

Garita, C. (2016). *Cursos de la carrera de Ingenieria en Sistemas*. Obtenido de http://moodle.realdesigncr.com/

Gonzalez, A. (2008). *Programación Orientada a Objetos*. Obtenido de https://algonzalezpoo.wordpress.com/caracteristicas-de-poo/

Hawke, S. (s.f.). *W3C España*. Obtenido de http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo

Jsx. (12 de 5 de 2016). *Mozilla Developer NetWork*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/JavaScript

Mano, C. (2014). *eHow en Español*. Obtenido de http://www.ehowenespanol.com/ventajas-desventajas-programacion-orientada-objetos-info\_184901/

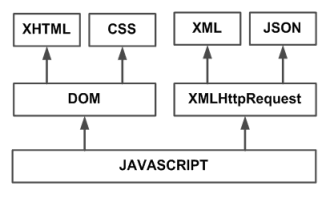
Muñoz, I. (13 de 7 de 2016). *Mozilla Developer NetWork*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/HTML

Salinas, P. (30 de 10 de 1996). *Universidad de Chile*. Obtenido de http://users.dcc.uchile.cl/~psalinas/uml/modelo.html

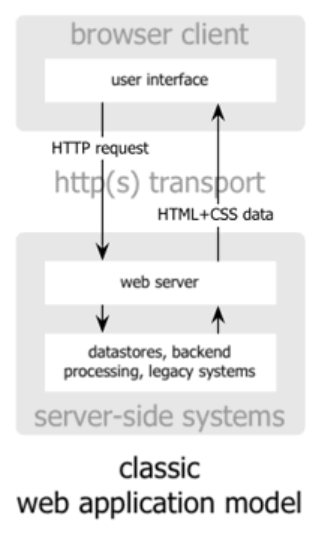
*Universidad de Murcia*. (s.f.). Obtenido de http://dis.um.es/~jfernand/0506/dai/poo.pdf

# Anexos

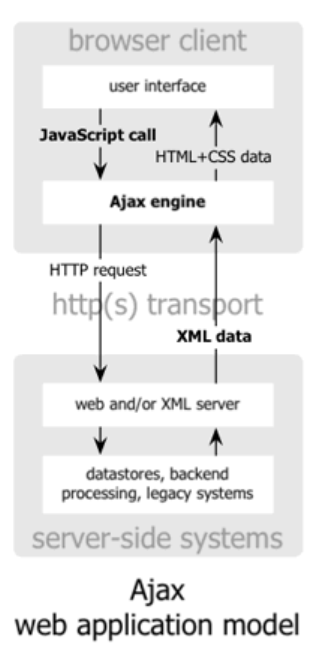
## Ilustración 1



## Ilustración 2



## Ilustración 3



## Ilustración 4