



Universidad Autónoma
del Estado de México

UNIVERSIDAD AUTONOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO
LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA

ANTOLOGÍA DE PROGRAMACIÓN DECLARATIVA
LIA D8-2022A

UNIDAD DE APRENDIZAJE:
PROGRAMACIÓN DECLARATIVA

AUTORES:
CRISTIAN BARRERA ROJAS
OSWALDO MERCADO ROJAS
ITZEL NICOLAS FLORES
ELIZABETH PÉREZ FLORES
LUIS FELIPE PÉREZ GARDUÑO
DANIELA RAMÍREZ CHIMAL
EMMANUEL RAMÍREZ LUCAS
JAIME RUBIO MARTÍNEZ
ALMA JOSELIN SÁNCHEZ RAMÍREZ

COMPILADORES:
MAN. CARLOS ALBERTO BALTAZAR VILCHIS
M.I. ELIZABETH EVANGELISTA NAVA
DRA. EN A. YENIT MARTÍNEZ GARDUÑO
DR. EN C.F. ALBERTO GARDUÑO

FECHA DE ELABORACIÓN: FEBRERO - JUNIO DE 2022



Licenciatura en Informática Administrativa 2022A

INDICE

INTRODUCCIÓN	3
APRENDIZAJES ESPERADOS	5
MAPA CURRICULAR: LICENCIATURA INFORMÁTICA ADMINISTRATIVA	6
MATERIAL REFERENCIAL AL MODULO DE LA UNIDAD DE APRENDIZAJE	8
UNIDAD I. INTRODUCCIÓN	9
REFERENCIAS	38
UNIDAD II. TECNOLOGÍAS DE FRONTEND	40
REFERENCIAS	69
UNIDAD 3. TECNOLOGÍAS DE BACK-END	70
REFERENCIAS	87
UNIDAD 4. TECNOLOGÍAS CONEXIÓN Y TRANSFERENCIA DE DATOS	88
REFERENCIAS	125
UNIDAD 5. INTEGRACIÓN DE TECNOLOGÍAS Y ENTORNOS DE DESARROLLO WEB	127
REFERENCIAS	154
ANEXO I. EJERCICIOS PROPUESTOS	155
ANEXO II. IDENTIFICACION DE LOS AUTORES	170



Licenciatura en Informática Administrativa 2022A

INTRODUCCIÓN

La presente Antología de Programación Declarativa LIA D8-2022 pretende exponer e introducir temas de gran importancia para un Licenciado en Informática Administrativa de igual forma para aquellas personas como estudiantes o maestros lleguen a adquirir la Antología, el presente documento brinda un listado de contenidos con temas relacionados con la programación declarativa, tanto en el desarrollo de sistemas y definiciones básicas de lenguaje de programación utilizados en el campo de análisis y diseño de sistemas, así como la identificación de los requerimientos del sistema incluyendo los métodos para recolectar requisitos relacionados con los datos, interacción con los usuarios, así como herramientas para finalmente documentar los detalles del sistema con ayuda de diversos métodos plasmados en los distintos softwares, también se estudiara el diseño de las características de los sistemas operativos cuya utilidad es importante hoy en día como software de base para la instalación de software de producción, esto deriva en crear programas computacionales con diferentes herramientas de acuerdo al área donde se ubique, por ejemplo una tarea simples como escribir un código fuente que imprima en pantalla un mensaje, este programa puede ser hecho en programas como Visual Studio, Java, C++, PHP, etc, y al mismo tiempo utilizar virtualizaciones en sistemas operativos como Ubuntu.

Adicionalmente, existen métodos útiles en la generación del código fuente para detectar errores en los datos de entrada y salida de esa forma evitar pérdidas de datos por parte del usuario que puedan producir resultados inesperados. Los sistemas operativos en la programación son importantes ya que en la actualidad los sistemas están basados en computadoras para la realización de las actividades cotidianas y objeto de gran consideración en la toma de decisiones. El desarrollo de sistemas de información involucra a los analistas de sistemas, programadores, así como a todos aquellos que harán uso de las aplicaciones que se desarrollen, es decir, es importante considerar a todos los stakeholders. De lo anterior el desarrollo de los contenidos de la presente antología se so organiza de acuerdo a las unidades de competencia que integran la unidad de aprendizaje Programación Declarativa; en la unidad de competencia I, se analizan los conceptos básicos, la importancia de los requerimientos y su relación con las categorías de software, concluyendo con un estudio de las herramientas utilizadas en el desarrollo de software enfatizando la forma de trabajo respecto a la programación; en la unidad de competencia II se observan las distintas formas de cómo se puede programar hasta la descripción de herramientas más utilizadas, sin embargo se complementan alternativas de desarrollo online que están diseñadas para asistir a los trabajadores que ocupan las Tecnologías de Frontend; en la unidad de competencia III se analizan recursos digitales relacionados con los lenguajes de programación tomando en cuenta cada una de las características de los lenguajes y controladores integrados en área de administración junto con distribución de código; en la unidad de competencia IV se analiza la configuración, revisión de la conectividad de los distintos lenguajes de programación; en la unidad de competencia V se



Licenciatura en Informática Administrativa 2022A

analiza la administración ,estructura y desarrollo de proyectos en la web con aplicación de conocimientos, habilidades, herramientas y técnicas aplicables a las actividades de un proyecto para satisfacer los requisitos del proyecto, resaltando los beneficios en la toma de decisiones al estudiar un sistema, a través de sus subsistemas con énfasis en programación.



Licenciatura en Informática Administrativa 2022A

APRENDIZAJES ESPERADOS

La programación declarativa ha evolucionado hasta el grado de permitir que las máquinas realicen procesos, simulando habilidades humanas implementadas mediante la codificación de programas en una gran diversidad de lenguajes de programación, se pretende que los alumnos obtengan los fundamentos teóricos necesarios para la comprensión y creación de programas de software. La antología como estrategia de enseñanza aprendizaje facilita la observación de avances sistemáticos con complejidad creciente sobre la temática específica, permite el reconocimiento de información especializada con sustento científico, en el caso de la programación permite identificar de lo general a lo particular el paradigma de la programación declarativa, identificar, caracterizar y relacionar las herramientas y técnicas computacionales-administrativas avanzadas. La Unidad de aprendizaje Programación Declarativa, tiene la finalidad de probar que toda administración debe tener información útil que permita a los programadores e Informáticos transformar los datos en información de calidad, que permitan a tomar decisiones, resolver problemas y cumplir con sus funciones/operaciones en forma eficiente y eficaz. La información es la base de la función de control, cualquier programa debe brindar información exacta, en el momento oportuno y para las personas adecuadas, si el objetivo de estas es la ejecución eficaz de la tarea. El término programación declarativa se refiere a que es un estilo de programación en el que el programador no dice “cómo” llegar a obtener la solución de un problema (es decir, las instrucciones necesarias para llegar a ella) sino “qué” define esa solución. Es lo contrapuesto a la programación imperativa, en la que el programador indica los pasos o instrucciones que deben realizarse para obtener la solución. Por lo contrario, la antología de programación declarativa LIA D8-2022 pretende reforzar y dar un amplio panorama sobre la importancia de tener información útil sobre los resultados de programar, planear, apoyar y tomar medidas adecuadas.



Licenciatura en Informática Administrativa 2022A

MAPA CURRICULAR: LICENCIATURA INFORMÁTICA ADMINISTRATIVA

La Unidad de Aprendizaje (UA) Programación declarativa, se imparte en el plan de estudios de la Licenciatura en Informática Administrativa, en su reestructuración f18 en el cuarto periodo.

Mapa curricular de la Licenciatura en Informática Administrativa, 2018

	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5	PERIODO 6	PERIODO 7	PERIODO 8	PERIODO 9	
O B J E T I V O S	Administración 3 1 4 7	Habilidades directivas 3 1 4 7	Modelos de emprendimiento Informático 2 2 4 6	Administración de las pymes y empresa familiar 3 1 4 7	Diseño por computadora 1 5 6 7	Administración de sistemas de capital social 2 4 6 8	Administración de proyectos informáticos 2 2 4 6	Administración Informática 2 2 4 6		P r á c t i c a p r o f e s i o n a l 30
	Contabilidad 3 1 4 7	Estructura de datos 2 4 6 8	Bases de datos 2 2 4 6	Software de base 2 4 6 8	Plataformas de aprendizaje virtual 2 4 6 8	Modelos de evaluación de software 2 2 4 6	Integrativa profesional* ** ** 8	Auditoría informática 2 2 4 6		
	Economía 3 1 4 7	Legislación informática 3 1 4 7	Análisis y planeación financiera 3 1 4 7	Ingeniería del software 2 4 6 8	Plataforma de comercio digital 2 4 6 8	Dirección de proyectos informáticos 2 2 4 6	Ética Profesional 2 2 4 6	Prospectiva Informática 2 2 4 6		
	Matemáticas aplicadas a la informática 3 1 4 7	Algoritmos computacionales 2 4 6 8	Programación imperativa 2 4 6 8	Programación declarativa 2 4 6 8	Riesgos de Tecnologías de la Información 2 4 6 8	Instalaciones y seguridad informática 2 4 6 8	Gestión de seguridad informática 2 4 6 8	Calidad de los servicios de Tecnologías de la Información 2 4 6		
	Gobierno de Tecnologías de la Información 3 1 4 7		Sistemas operativos 2 4 6 8	Comunicación entre computadoras 2 4 6 8	Análisis y diseño de sistemas 2 4 6 8	Sistemas de información administrativos 2 2 4 6	Sistemas de información del conocimiento 2 2 4 6	Sistemas de información estratégicos 2 2 4 6		
	Lógica computacional 3 1 4 7	Arquitectura computacional 2 4 6 8								
		Inglés 5 2 2 4 6	Inglés 6 2 2 4 6	Inglés 7 2 2 4 6	Inglés 8 2 2 4 6					
						Optativa 1 1 3 4 5	Optativa 2 1 3 4 5	Optativa 3 1 3 4 5		

Figura 1. Mapa Curricular (2018)



Licenciatura en Informática Administrativa 2022A

DISTRIBUCIÓN DE LAS UNIDADES DE APRENDIZAJE OPTATIVAS

	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5	PERIODO 6	PERIODO 7	PERIODO 8	PERIODO 9
O P T A T I V A S						Projects based on PMBok i 1 3 4 5	Gobierno de TI basados en COBIT 1 3 4 5	Gestión y análisis de BIG DATA 1 3 4 5	
						Desarrollo de proyectos complejos basados en SCRUM 1 3 4 5	Servicios de IT basados en ITIL 1 3 4 5	Arquitectura empresarial basada en TOGAF 1 3 4 5	
						Inteligencia de negocios BI 1 3 4 5	Lenguaje extensible de informes de negocios XERL 1 3 4 5	Sistemas de planificación de recursos empresariales ERP 1 3 4 5	

SIMBOLOGÍA		PARAMETROS DEL PLAN DE ESTUDIOS																	
Unidad de aprendizaje	HT: Horas Teóricas HP: Horas Prácticas TH: Total de Horas CR: Créditos	Núcleo básico obligatorio: cursar y acreditar 15 UA 36 28 66 104	Núcleo sustantivo obligatorio: cursar y acreditar 20 UA 41 63 104 145	Núcleo integral obligatorio: cursar y acreditar 9 UA + 2* 18** 18** 94	Núcleo integral optativo: cursar y acreditar 3 UA 3 9 12 15	Total del núcleo básico: acreditar 15 UA para cubrir 104 créditos	Total del núcleo sustantivo: acreditar 20 UA para cubrir 145 créditos	Total del núcleo integral: acreditar 12 UA + 2* para cubrir 109 créditos	<table border="1"> <thead> <tr> <th colspan="2">TOTAL DEL PLAN DE ESTUDIOS</th> </tr> </thead> <tbody> <tr> <td>UA obligatorias</td> <td>44 + 2 Actividades académicas</td> </tr> <tr> <td>UA optativas</td> <td>3</td> </tr> <tr> <td>UA a acreditar</td> <td>47 + 2 actividades académicas</td> </tr> <tr> <td>Créditos</td> <td>358</td> </tr> </tbody> </table>	TOTAL DEL PLAN DE ESTUDIOS		UA obligatorias	44 + 2 Actividades académicas	UA optativas	3	UA a acreditar	47 + 2 actividades académicas	Créditos	358
TOTAL DEL PLAN DE ESTUDIOS																			
UA obligatorias	44 + 2 Actividades académicas																		
UA optativas	3																		
UA a acreditar	47 + 2 actividades académicas																		
Créditos	358																		
→ 5 líneas de seriación. * Actividad académica. ** Horas de las actividades académicas. Créditos mínimos 20 y máximos 45 por periodo escolar.																			
■ Núcleo básico obligatorio. ■ Núcleo sustantivo obligatorio. ■ Núcleo integral obligatorio. ■ Núcleo integral optativo																			

Figura 2. Parámetros de Plan de estudios LIA (2018)



Licenciatura en Informática Administrativa 2022A

MATERIAL REFERENCIAL AL MODULO DE LA UNIDAD DE APRENDIZAJE

Tabla 1. Módulo de aprendizaje

UNIDAD DE APRENDIZAJE	TEMA
UNIDAD 1. INTRODUCCIÓN	1.1 Paradigma de la programación declarativa y lenguajes usados. 1.2 Arquitectura de las aplicaciones web. 1.2.1 Modelo Cliente-Servidor de 3 capas (datos, lógica de negocio y presentación) 1.2.2 Modelo Vista Controlador 1.3 Conceptos de Interfaces (APIs) 1.4 Modelado y diseño 1.4.1 UML 1.4.2 Diagramas de diseño (Wireframe, Mock-up)
UNIDAD 2. Tecnologías de Frontend	2.1 Etiquetado y estructuración con HTML. 2.2 Presentación de información con CSS. 2.3 Programación de funcionalidad con JavaScript. 2.4 Frameworks CSS y JavaScript. 2.5 Alternativas de desarrollo on-line
UNIDAD 3. Tecnologías de Back-end	3.1 Lenguajes y entornos de desarrollo integrados (IDE) 3.2 Controladores de versiones, administración y distribución de código. 3.3 Instalación y configuración de servidores de aplicaciones y entornos preconfigurados para la ejecución. 3.4 Desarrollo de programas para el back-end
UNIDAD 4. Tecnologías conexión y transferencia de datos	4.1 Revisión de la instalación y configuración de un DBM 4.2 Estándares y tecnologías de conectividad a DB (ODBC, JDBC, OLE DB, etc.) 4.3 Lenguaje de consulta SQL y aplicación. 4.4. Lenguaje de etiquetado de datos (JSON, XML) y aplicación.
UNIDAD 5. Integración de tecnologías y entornos de desarrollo web	5.1 Entornos preconfigurados para el desarrollo web. 5.2 Frameworks de desarrollo y estructura del proyecto. 5.3 Desarrollo de soluciones integradas.



Licenciatura en Informática Administrativa 2022A

UNIDAD I. INTRODUCCIÓN

OBJETIVO:

Comprender los conceptos del paradigma de la programación declarativa, los elementos de arquitectura de aplicaciones web y los fundamentos para la administración de su desarrollo.

1.1 Paradigma de la programación declarativa y lenguajes usados.

Para Gómez (2021) un paradigma de programación es una forma de escribir implementaciones formales de algoritmos específicos (programas de computador) que sigue un conjunto coherente de principios, posiblemente tomando como base teorías matemáticas (Van Roy, 2009 citado por Gómez, 2021). Por ejemplo, si se basan en lógica formal, los algoritmos son considerados deducciones (Clinger, 1989 citado por Gómez, 2021). El paradigma de programación declarativo incluye los lenguajes de programación donde los programadores expresan el qué de un algoritmo mientras el lenguaje implementa el cómo (Lloyd 1994 citado por Gómez, 2021). Este incluye los paradigmas de programación funcional y lógico.

Por su parte Cardona (2015) afirma que la programación declarativa es un paradigma de programación que está basado en el desarrollo de programas especificando o declarando un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución. En la programación imperativa se describe paso a paso un conjunto de instrucciones que deben ejecutarse para variar el estado del programa y hallar la solución. En la programación declarativa, en cambio, las sentencias que se utilizan lo que hacen es describir el problema que se quiere solucionar, pero no las instrucciones imperativas necesarias para hacerlo. Por todo ello, los programas como PROLOG tienen menos detalles y son más concisos, evitan controlar la secuencia de ejecución del programa.

¿Qué es la programación declarativa?

No hay consenso sobre la definición de este paradigma, pero todas las explicaciones coinciden en algo: lo que destaca de los lenguajes de programación declarativa es que siempre se describe el resultado final deseado, en lugar de mostrar todos los pasos de trabajo. Para alcanzar el objetivo, en la programación declarativa se determina automáticamente la vía de solución. Esto funciona siempre y cuando las especificaciones del estado final se definan claramente y exista un procedimiento de ejecución adecuado. Si se dan las dos condiciones, la programación declarativa es muy eficiente. (Ionos, 2020)

Como la programación declarativa no determina el “cómo”, sino que funciona a un nivel de abstracción muy alto, este paradigma deja margen para la optimización. Si se ha desarrollado un procedimiento de ejecución mejor, el algoritmo integrado lo encuentra y lo aplica. En este sentido, el paradigma está muy preparado para el futuro porque, al escribir el código, no es necesario determinar el procedimiento según el cual se alcanza el resultado.



Licenciatura en Informática Administrativa 2022A

Los lenguajes de programación declarativa más conocidos son:

- Prolog
- Lisp
- Haskell
- Miranda
- Erlang
- SQL (en un sentido amplio)

Los distintos lenguajes declarativos se pueden subdividir, a su vez, en dos paradigmas, el de la programación funcional y el de la programación lógica.

Sin embargo, es habitual que en la práctica los límites se difuminen y que, a la hora de solucionar problemas, se apliquen tanto elementos de la programación imperativa, con sus subtipos de programación procedimental, modular y estructurada, como de la programación declarativa.



Figura 3. Paradigmas de programación

Fuente: Ionos (2022)

Programación declarativa e imperativa

El paradigma de programación imperativa (paradigma orientado a instrucciones) es el más antiguo de los dos paradigmas fundamentales. A diferencia de en la programación declarativa, en este caso, el desarrollador determina con precisión en el código fuente los pasos que debe dar el ordenador para alcanzar el resultado. Se prioriza, pues, el "cómo" de la vía de solución. Este enfoque se puede encontrar, por ejemplo, en Java, Pascal o en



Licenciatura en Informática Administrativa 2022A

C. Por el contrario, en la programación declarativa se describe directamente el “qué” de la solución. (Ionos, 2022).

Tomemos como ejemplo la construcción de muebles: mientras que la programación imperativa ofrece las instrucciones de montaje, la programación declarativa ofrece una imagen del mueble ya montado como muestra.

En lugar de dejar abierta la ejecución con funciones, en la programación imperativa existen variables que se modifican durante el tiempo de ejecución. Esto hace que el código sea más largo, pero también más fácil de entender que la forma abreviada y abstracta del lenguaje declarativo.

Ejemplo de programación declarativa

Un punto fuerte de la programación declarativa es su capacidad para describir problemas de forma más corta y precisa que el lenguaje imperativo.

Si se tiene que crear una lista con nombres, con la programación declarativa se puede describir en PHP con solo una línea de código, como vemos aquí abajo, mientras que el procedimiento imperativo necesita cinco líneas. (Ionos, 2022)

Programación imperativa:

```
1 $listaparticipantes = [1 => 'Peter', 2 => 'Hans', 3 => 'Sarah'];  
2 $nombres = [];  
3 foreach ($listaparticipantes as $id => $apellido) {  
4     $nombres[] = $apellido;  
5 }
```

Figura 4. Programación imperativa.

Fuente: Ionos (2022)

Programación declarativa:

```
1 $nombres = array_values($listaparticipantes);
```

Figura 5. Programación declarativa.

Fuente: Ionos (2022)



Licenciatura en Informática Administrativa 2022A

Ventajas e inconvenientes de los lenguajes de programación declarativa

El estilo de programación declarativa se utiliza hoy en día en una multitud de casos y, en ocasiones, en estilo puro. Sin embargo, el enfoque no es apropiado para todos los usos.

El código declarativo destaca por su alto nivel de abstracción. Esto permite a los desarrolladores representar programas complejos de forma comprimida. Sin embargo, cuanto más extensa sea la ejecución, mayor será el riesgo de que el código se vuelva tan enrevesado que solo sea legible para el desarrollador que lo escribió. Para las empresas que quieren aplicar el mantenimiento y desarrollo de las aplicaciones con independencia de la persona que las desarrolló, esta situación puede suponer un peligro. En estos casos, los desarrolladores externos deberán esforzarse por leer el código declarativo hasta que hayan entendido la estructura y hayan resuelto los problemas.

No obstante, el nivel de abstracción de la programación declarativa también ofrece ventajas. Siempre que la ejecución se encuentre claramente separada del sistema mediante un algoritmo, el mantenimiento se podrá realizar con independencia del desarrollo de la aplicación. Las averías en el funcionamiento diario se reducen al mínimo. Al mismo tiempo, la optimización resulta más fácil, ya que el algoritmo utilizado es suficientemente abierto como para acceder a nuevos métodos. La desventaja del uso del algoritmo es que, dependiendo de determinadas características de los casos concretos de aplicación, una solución típica de este tipo a menudo resulta insuficiente.

Aunque de por sí no puede considerarse una desventaja, el modelo de la programación declarativa sí que presenta un desafío. Pensar en estados de solución es contrario al pensamiento natural del ser humano. Las personas pensamos en procesos que llevan a un objetivo, en lugar de partir de un objetivo e ir hacia atrás. En este sentido, se requiere de los desarrolladores un cambio de perspectiva y una familiarización que puede ralentizar, en un primer momento, las soluciones de los problemas. Sin embargo, una vez que se haya entrenado esta nueva forma de pensar, el enfoque declarativo puede desarrollar sus puntos fuertes.

Tabla 2. Ventajas y desventajas de la programación declarativa

Ventajas	Desventajas
Código más corto y eficiente.	En parte, difícil de comprender para personas ajenas.
Realizable con métodos no conocidos en el momento de la programación.	Basado en una forma de pensar no habitual en las personas (estado de solución).
Optimización sencilla, ya que la ejecución se gestiona mediante un algoritmo.	Las características de casos de aplicación individuales se pueden considerar en la programación, pero únicamente de forma compleja.
Es posible el mantenimiento independiente del desarrollo de la aplicación.	

Fuente: Ionos (2022)

Licenciatura en Informática Administrativa 2022A

Además, partir del desarrollo de la descripción del problema tiene la ventaja de que los equipos pueden esquematizar realmente rápido los modelos de soluciones. Por último, la programación concreta de la ejecución puede realizarse en un paso posterior. Para ello, el estilo declarativo de prototyping en el desarrollo ágil de software es muy propicio.

En la práctica, hoy en día se utilizan principalmente formas mixtas de paradigmas y los lenguajes de programación declarativa se complementan con métodos imperativos. Sin embargo, estas fórmulas son más propensas a contener errores y pueden perjudicar la legibilidad del código.

1.2 Arquitectura de las aplicaciones web

La arquitectura de una aplicación web se basa en el modelo cliente/servidor, pues en esta existe una lógica de negocios sensible a las interacciones del usuario, esta es considerada como la arquitectura clásica. (Conallen, 1999 citado por García, 2010).

La arquitectura de las aplicaciones web consta de máquinas conectadas a una red, por lo general, Internet o una Intranet corporativa que sigue el esquema cliente-servidor en nuestro caso de servidores web. Surgió a mediados de la década de 1990, durante la etapa de la Web 1.0 con la aparición de las primeras conexiones de acceso conmutado (RTC, RDSI, GSM, GPRS) y de las etiquetas multimedia del estándar HTML y la incorporación de pequeños programas realizados en Java, llamados applets.

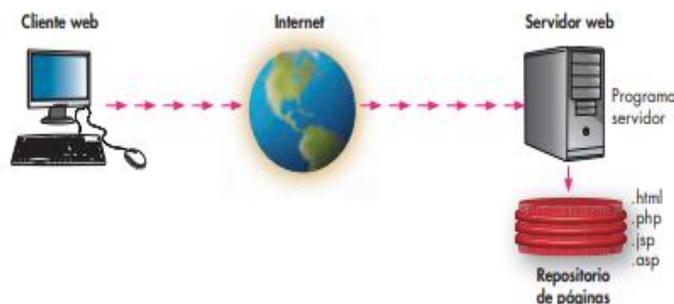


Figura 6. Esquema básico del servidor web

Fuente: Lerma et al (2013)

Cuando un cliente realiza una petición de una URL a un servidor, teniendo en cuenta que gracias al servicio DNS conocemos su IP, le pide una página web. Esta se busca en el repositorio de páginas del servidor y es devuelta al cliente. Toda esta comunicación se lleva a cabo mediante el protocolo HTTP, y el código de las páginas es básicamente HTML junto con código CSS y JavaScript (u otros complementos como animaciones Flash que se

Licenciatura en Informática Administrativa 2022A

ejecutan en el cliente), que surgieron en la etapa de la web 1.5, como se indica en la figura 4.

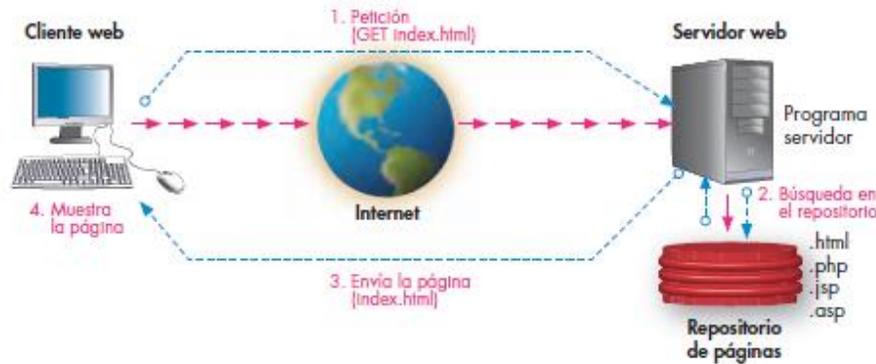


Figura 7. Esquema básico de funcionamiento del servicio web con páginas estáticas

Fuente: Lerma et al (2013)

Cuando se trata de páginas web dinámicas, escritas en lenguajes de programación para el servidor como PHP, ASP.net o JSP, el código HTML que se envía al cliente se construye de forma dinámica dentro del programa servidor en el momento en que se procesa la petición. Las páginas se crearán a partir de la información recibida en la misma petición, o mediante consultas a bases de datos (Lerma Et al (2013)).



Figura 8. Esquema básico del funcionamiento del servicio web con páginas dinámicas

Fuente: Lerma et al (2013)

Ahora que las aplicaciones web interactivas han evolucionado para los humanos, el siguiente paso se centra en el Internet de las cosas, es decir, que todo lo que nos rodea que no es humano pueda conectarse entre sí y nos haga la vida más fácil. Esta funcionalidad se denomina servicio web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus

Licenciatura en Informática Administrativa 2022A

servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web (Lerma *Et al* (2013)).

El funcionamiento básico de los servicios web es el siguiente:

1. Una empresa, como proveedor, implementa su servicio y lo registra en el agente o UDDI que hace la función de listín público de servicios web.
2. El cliente o consumidor busca el servicio web que necesita a través del UDDI.
3. El cliente puede acceder al servicio web del proveedor y decidir si lo utiliza. (Lerma *Et al* (2013)).

1.2.1 Modelo Cliente-Servidor de 3 capas (datos, lógica de negocio)

Para Marini (2012) el modelo Cliente-Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas.

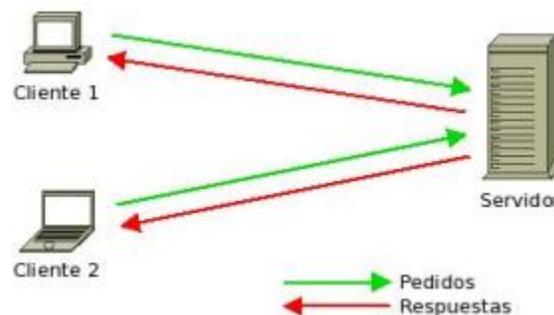


Figura 9. El Modelo Cliente/Servidor.

Fuente: Marini et al (2012)

Por tanto, Marini (2012) menciona que esta arquitectura permite que la capacidad de proceso este repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes. Es decir:

- Cliente: Programa ejecutable que participa activamente en el establecimiento de las conexiones. Envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo.

Licenciatura en Informática Administrativa 2022A

• Servidor: Es un programa que ofrece un servicio que se puede obtener en una red. Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante. Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación. El servidor comienza su ejecución antes de comenzar la interacción con el cliente.

Una representación gráfica de lo anterior señalado podría ser lo siguiente:

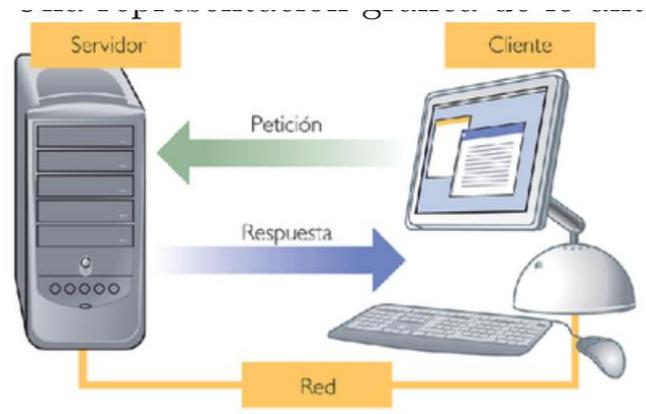


Figura 10. Representación del Modelo Cliente/ Servidor

Fuente: Lizama *et al* (2016)

Las ventajas de este modelo respecto de otras posibles arquitecturas de red serían:

- Centralizado
- Es de Fácil mantención
- Escalabilidad

Por otro lado, la mayor desventaja es que a mayor número de clientes, más problemas para el servidor, debido a la congestión en el tráfico que se genera (Lizama *et al* (2016).

El Modelo de Servicios

Según Marini (2012) un modelo es una vista abstracta que establece las definiciones, reglas y relaciones entre las estructuras relacionadas con la aplicación. Sirve de base para el intercambio de ideas durante el desarrollo lógico de la aplicación y determina cómo será la aplicación resultante.

El modelo de servicios establece tres grandes conjuntos de funcionalidades, en cada uno de los cuales se encuadran las distintas tareas en las que se ve involucrado cualquier tipo

Licenciatura en Informática Administrativa 2022A

de proyecto de desarrollo. Dicho modelo establece los siguientes conjuntos que comúnmente denominamos lógicas.

1. **Lógica de presentación:** Esta lógica es la responsable del control de todos los aspectos relacionados con la interacción entre el usuario y la aplicación.
2. **Lógica de negocio:** Es la lógica de la aplicación que controla la secuencia de acciones y fuerza el cumplimiento de las reglas del negocio propias de cada empresa. La lógica del negocio también transforma una serie de datos en información útil para el usuario mediante la aplicación de las reglas apropiadas. El objetivo que debe cumplir esta lógica es el de aislar las reglas del negocio, así como las transformaciones de datos de los consumidores (usuarios y otros componentes de esta misma capa) y de los sistemas de gestión de datos.
3. **Lógica de datos:** En este conjunto entran los procesos encargados de la gestión de los datos propiamente dicha, es decir, los procesos encargados del mantenimiento de los datos, de garantizar las reglas de integridad referencial establecidas, así como de la gestión de las transacciones. Estas tareas son realizadas, generalmente, por un Sistema de Gestión de Bases de Datos Relacionales.



Figura 11. El Modelo de Servicios

Fuente: Marini et al (2012)

Modelo Cliente/Servidor 2 capas

Según Marini (2012) uno de los objetivos de las aplicaciones de 2 capas es separar la lógica de acceso a los datos de lo que es la interfaz de usuario y trasladarla al servidor. ¡Habitualmente, se implementan servicios como procedimientos almacenados en el sistema gestor de datos; con esto se pretende reducir la carga de los clientes y centralizar las

Licenciatura en Informática Administrativa 2022A

operaciones comunes de acceso a los datos. El Sistema Gestor de Datos también suele incorporar la funcionalidad necesaria para trabajar en entornos multiusuarios.

En este modelo intervienen únicamente dos entidades: El Cliente y El Servidor:

El papel de Cliente lo desempeña la aplicación final del usuario, que implementará todas las funciones correspondientes a la lógica de presentación, más algunas de las funciones relacionadas con la lógica del negocio, como pueden ser determinadas validaciones de datos y condiciones de recuperación.

El papel de Servidor lo desempeña el propio SGBD, el cual se ocupará de todas las funciones correspondientes a la lógica de datos, más las restantes funciones correspondientes a la lógica del negocio, mediante la codificación de Procedimientos Almacenados.

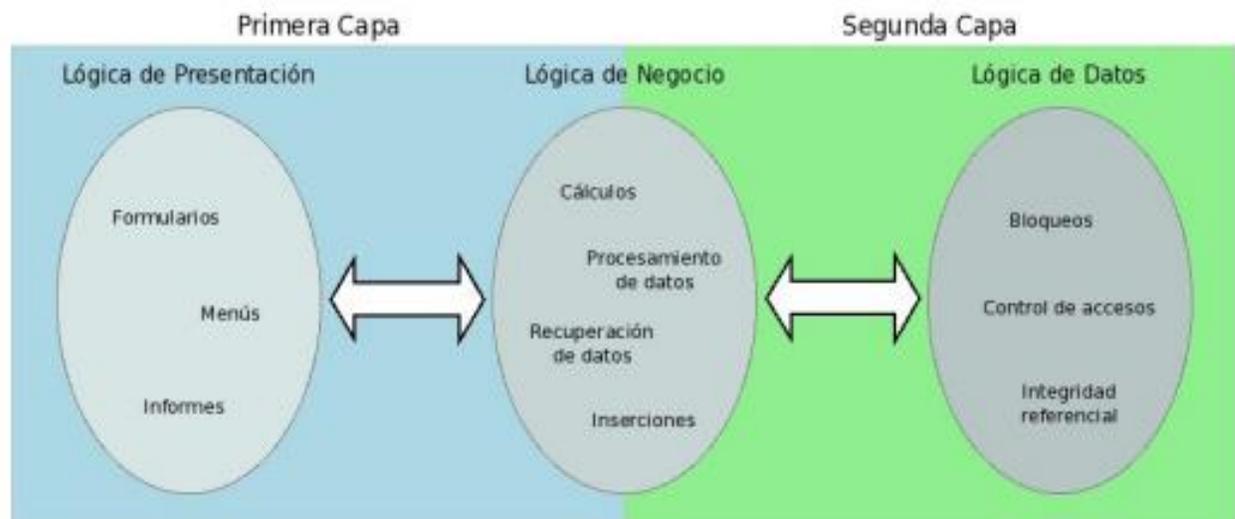


Figura 12. El Modelo Cliente/Servidor de 2 capas

Fuente: Marini et al (2012)

Modelo Cliente/Servidor multicapa

Para Marini (2012) la arquitectura cliente/servidor genérico tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o dos capas. Algunas redes disponen de tres tipos de nodos:

Licenciatura en Informática Administrativa 2022A

- Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.
- Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Esta configuración se llama una arquitectura de tres- capas, es decir, este modelo aporta una flexibilidad adicional en la construcción de aplicaciones cuando éstas aumentan su complejidad. Influye tanto en el modelo de aplicación (lógicas de presentación, del negocio y de datos) como en la distribución de los servicios. El modelo conceptual de una aplicación establece sus definiciones, reglas y relaciones, así como su estructura. Hay partes de la lógica que residen en el cliente, normalmente las que se refieren a la interfaz de usuario, mientras que las del negocio y de datos suelen residir en los servidores, que proporcionan los mecanismos necesarios para el trabajo en entornos multiusuarios. En este tipo de modelo se aplica íntegramente el modelo de servicios ya que, cada una de las capas se corresponde con cada una de las lógicas descritas.



Figura 13. El Modelo Cliente/Servidor multicapa.

Fuente: Marini et al (2012)

Este modelo hace que la aplicación final sea completamente independiente del origen de los datos que procesa, tarea que pasa a ser competencia directa del componente especializado.



Licenciatura en Informática Administrativa 2022A

1.2.2 Modelo Vista Controlador

El patrón Modelo/Vista/Controlador o MVC (Model/View/Controller) fue descrito por primera vez en 1979 por Trygve Reenskaug e introducido como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk.

Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas (Fernández Y. et al (2012)).

Este modelo de arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- Interfaz de Programación de Aplicaciones API (Application Programming Interface) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones.

- El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.
- El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Licenciatura en Informática Administrativa 2022A

Elementos del Patrón

- Modelo: datos y reglas de negocio.
- Vista: muestra la información del modelo al usuario.
- Controlador: gestiona las entradas del usuario.

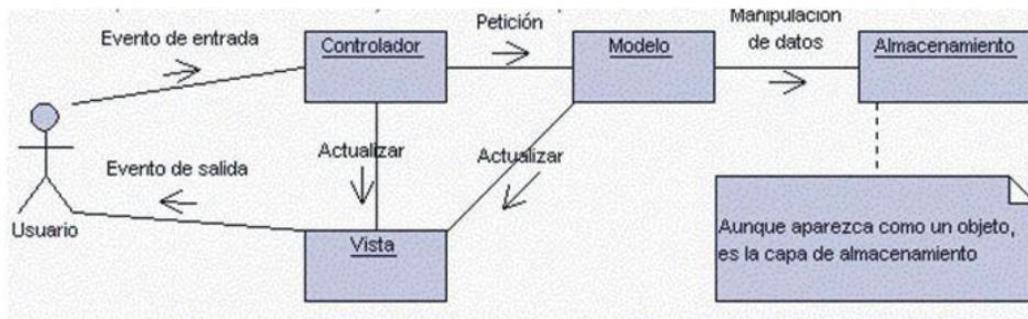


Figura 14. Interrelación entre los elementos del patrón MCV

Fuente: Fernández Y. *et al* (2012)

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador, aunque es posible definir múltiples vistas para un único controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de tarta, de barras, tabular, etc. A continuación, se detalla cada elemento:

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos.
- Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Es opcional, pues las reglas de negocio pueden estar también en los controladores, directamente en las acciones.
- Notificará a las vistas los cambios que en los datos pueda producir un agente externo si se está ante un modelo activo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El controlador es responsable de:

1. Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
2. Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas



Licenciatura en Informática Administrativa 2022A

peticiones a las vistas puede ser una llamada al método "Actualizar ()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega (nueva_orden_de_venta)".

Las vistas son responsables de:

- Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado.
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes (Fernández Y. *et al* (2012)).

1.3 Conceptos de Interfaces (APIs)

(Plata, 2022) nos dice que una interfaz de programación de aplicaciones (API) es un conjunto de herramientas, definiciones y protocolos que se utiliza para integrar los servicios y el software de aplicaciones. Es lo que permite que sus productos y servicios se comuniquen con otros, sin tener que diseñar permanentemente una infraestructura de conectividad nueva.

Las API pueden ser privadas (para uso interno únicamente), compartidas (con partners específicos para brindar flujos de ingresos adicionales) o públicas (entidades externas pueden desarrollar aplicaciones que interactúen con sus API para fomentar la innovación). Si opta por compartir sus API, podrá disfrutar de las siguientes ventajas:

- Crear nuevos canales de ingresos o ampliar los actuales
- Expandir el alcance de su marca
- Facilitar la innovación abierta o lograr un mayor nivel de eficiencia gracias a la colaboración y el desarrollo externos

Para Altamirano (2012) la interfaz de programación de aplicaciones (API) es un conjunto de herramientas, definiciones y protocolos que se utiliza para integrar los servicios y el software de aplicaciones. Por ejemplo, permite que los productos y servicios se comuniquen con otros, sin tener que diseñar permanentemente una infraestructura de conectividad nueva (Altamirano, (2012)). Este mismo pueden puede ser privada es decir para uso interno únicamente, compartidas con partners específicos para brindar flujos de ingresos adicionales o bien públicas, entidades externas pueden desarrollar aplicaciones que interactúen con sus API para fomentar la innovación.

Según Plata, (2022) las ventajas de compartir las interfaces de las API se encuentran

- Crear_nuevos canales de ingresos o ampliar los actuales
- Expandir el alcance de su marca
- Facilitar la innovación abierta o lograr un mayor nivel de eficiencia gracias a la colaboración y el desarrollo externos

Licenciatura en Informática Administrativa 2022A

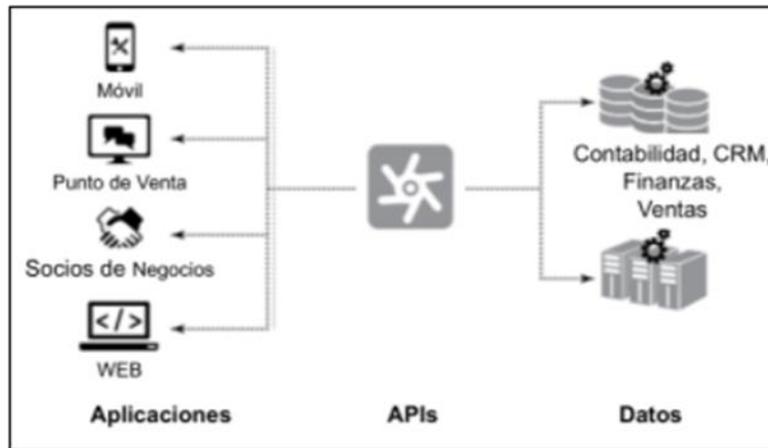


Figura 15. Interfaces (APIs)

Fuente: Altamirano (2012)

Para Eames, (2017) cuando un cliente abre una incidencia relacionada con, por ejemplo, impresoras en ServiceTonic, la aplicación accede a la API de la herramienta del proveedor y crea un ticket. Cuando el proveedor finaliza la incidencia, su herramienta de help desk accede a ServiceTonic vía API, introduce la solución y cambia el estado de la incidencia. Así, cada empresa puede utilizar su propia herramienta de help desk. Lo único que hay que hacer es conectar ambas herramientas mediante sus respectivas API.

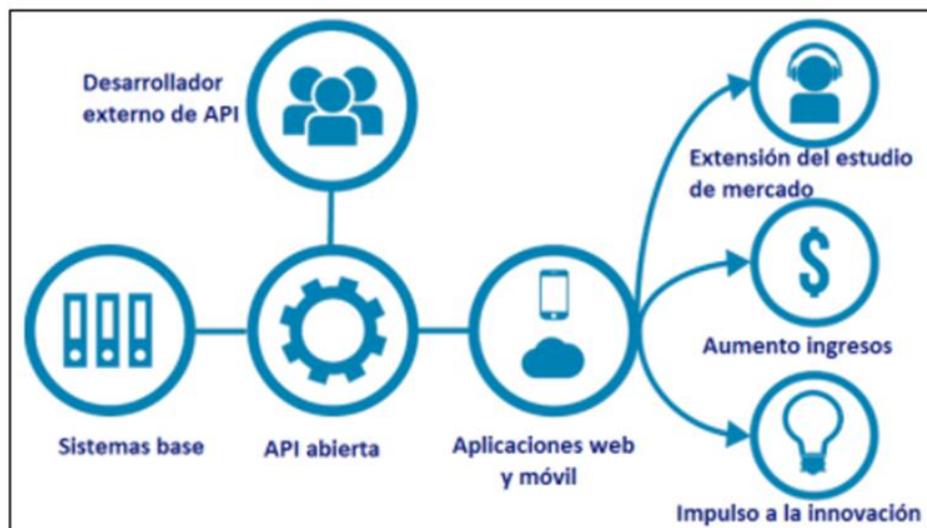


Figura 16. El desarrollo Interfaces (APIs)

Fuente: Altamirano (2012)



Licenciatura en Informática Administrativa 2022A

Una de las pocas claves eficaces del problema de diseño (la capacidad del diseñador de reconocer la mayor cantidad de limitaciones posible), su predisposición y entusiasmo para trabajar dentro de estas limitaciones. Las limitaciones de precio, de tamaño, de solidez, de equilibrio, de superficie, de tiempo, y así sucesivamente.

APIS como herramienta de plataformas

Para Pérez (2012) Las APIs de nubes son bibliotecas de software con un conjunto de procedimientos, funciones y objetos que permiten a los desarrolladores de aplicaciones el acceso y control a la plataforma de la nube y a sus recursos. Estas herramientas son esenciales para obtener acceso completo a las ventajas de la nube. Las APIs son desarrolladas en algún lenguaje de programación particular, siendo los más comunes: Java, Python y C#.

Tabla 3. Aplicación –A (implementación única)

Aplicación-A (implementación única)				SaaS
jClouds				
Implementación API-1	Implementación API-2		Implementación API-n	PaaS
Nube-1	Nube-2		Nube-n	IaaS

Fuente: Pérez (2012)

JCLOUDS es una implementación en Java de un paquete estándar para unificar múltiples proveedores de nubes. Soporta las nubes más populares y da a los desarrolladores la posibilidad de mantener sus aplicaciones libres de lock-in. Puede que éste sea lo más cercano a lo que se propone este documento, pero su estrategia es diferente. En la tabla 1 se muestra la ubicación de jClouds en la arquitectura de una nube. JClouds es un API no estándar colocada entre la aplicación y el API de la nube. Sus desarrolladores se dan a la tarea de incorporar soporte para cada nube popular. Debido a esto, jClouds tiene un problema potencial de crecimiento; entre más nubes surjan, el API se volverá más y más grande hasta volverse demasiado pesada o tendrá que ignorar algunas nubes para mantenerse ligera. Y como no hay compromiso de las nubes a jClouds, un proveedor podría



Licenciatura en Informática Administrativa 2022A

cambiar su API y afectar el resultado de esta biblioteca. jClouds es la única manera de crear aplicaciones portables en la nube en este momento.

Es una buena estrategia, pero puede no ser suficiente a futuro. A diferencia de jClouds, el modelo del API que se está desarrollando no pretende ser una biblioteca adicional a la del proveedor, sino que la de él sea implementada basándose en un estándar para que sea compatible con las de otras nubes.

1.4 Modelado y Diseño

Un modelo de datos puede ser pensado como un diagrama de flujo que ilustra las relaciones entre los datos. A pesar de que la captura de todas las posibles relaciones en un modelo de datos puede consumir mucho tiempo, es un paso importante que no debería ser apresurado. (Kenneth, 2011).

El modelado de datos es el proceso de documentar un diseño de sistema de software complejo como un diagrama de fácil comprensión, usando texto y símbolos para representar la forma en que los datos necesitan fluir. (Kendall, 2011).

Los modelos de datos se construyen durante las fases de análisis y diseño de un proyecto, para asegurar que los requisitos para una nueva aplicación se entienden completamente. (Kendall, 2011).

A través del modelado conseguimos cuatro objetivos:

- Los modelos nos ayudan a visualizar cómo es, o queremos que sea un sistema.
- Los modelos nos permiten especificar la estructura o el comportamiento de un sistema.
- Los modelos nos proporcionan plantillas que nos guían en la construcción de un sistema.
- Los modelos documentan las decisiones que hemos adoptado.

Existen tres tipos diferentes de modelos de datos: conceptual, lógico y físico. Cada modelo de datos tiene un propósito específico, que se define principalmente por el nivel de detalles operativos. (Kenneth, 2011).

Modelo de Datos Físico

Un modelo de datos físicos es un modelo específico de bases de datos que representa objetos de datos relacionales (por ejemplo, tablas, columnas, claves principales y claves externas) y sus relaciones. (Fernández, 2006).

Un modelo de datos físicos se puede utilizar para generar sentencias DDL que, después, se pueden desplegar en un servidor de base de datos.



Licenciatura en Informática Administrativa 2022A

Mediante el entorno de trabajo, puede crear un modelo de datos físicos de diferentes maneras:

- Cree un modelo físico en blanco mediante un asistente
- Cree un modelo de datos físicos a partir de una plantilla mediante un asistente
- Reverta la ingeniería de un modelo físico de una base de datos o un archivo DDL mediante un asistente o arrastrando objetos de datos del Explorador de fuentes de datos
- Importe un archivo de modelo de datos físicos del sistema de archivos
- Cree un modelo físico mediante la transformación de un modelo de datos lógicos.

Modelo de Datos Lógico

Un modelo de datos lógico contiene representaciones de entidades y atributos, relaciones, identificadores exclusivos, subtipos y supertipos y restricciones entre relaciones. Un modelo lógico también puede contener objetos de modelo de dominio o referirse a uno o varios modelos de dominio o de glosario. (Fernández, 2006).

Una vez definidas las relaciones y los objetos lógicos en un modelo lógico de datos, se utiliza el área de trabajo para transformar el modelo lógico en una representación física específica de la base de datos en forma de modelo físico de datos.

Modelo de Datos Conceptual

Debe emplearse un modelo de datos conceptual para definir y comunicar relaciones de alto nivel entre conceptos/entidades. En otras palabras, ayudan a una empresa a ver sus datos y las relaciones entre diferentes tipos de datos en contexto. (Fernández, 2006).

Un modelo conceptual de datos identifica las relaciones de más alto nivel entre las diferentes entidades.

Las características del modelo conceptual de datos incluyen:

- Incluye las entidades importantes y las relaciones entre ellas.
- No se especifica ningún atributo.
- No se especifica ninguna clave principal.

Los modelos de datos físicos, lógicos y conceptuales bien documentados permiten que las partes interesadas identifiquen errores y hagan cambios antes de que cualquier código de programación se haya escrito. (Kenneth, 2011).

Los modeladores de datos suelen utilizar varios modelos para ver los mismos datos y garantizar que todos los procesos, entidades, relaciones y flujos de datos han sido identificados. Hay varios enfoques diferentes para el modelado de datos, incluyendo:



Licenciatura en Informática Administrativa 2022A

Modelado conceptual de datos

Identifica las relaciones de más alto nivel entre diferentes entidades.

Modelado de datos empresariales

Similar al modelado de datos conceptuales, pero se dirige a los requisitos únicos de un negocio específico.

Modelado lógico de datos

Ilustra las entidades, atributos y relaciones específicas que participan en una función de negocios. Sirve como base para la creación del modelo de datos físico.

Modelado de datos físicos

Representa una aplicación e implementación específica de base de datos de un modelo de datos lógicos.

Existe una herramienta eficaz para el modelado y diseño de software la cual lleva por nombre UML. El Lenguaje Unificado de Modelado (UML, Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables. (Fernández, 2006).

El modelado es la espina dorsal del desarrollo de software de calidad. Se construyen modelos para poder comunicarnos con otros, para explicar el comportamiento del sistema a desarrollar, para comprender mejor ese sistema y poder atacar problemas que sin el modelado su resolución sería imposible, tanto desde el punto de vista de los desarrolladores como desde el punto de vista del cliente. (Fernández, 2006).

1.4.1 UML (Lenguaje unificado de modelado)

El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos.

Una parte del UML detalla, una abstracción con significado de un lenguaje para expresar otros modelos (es decir, otras abstracciones de un sistema, o conjunto de unidades conectadas que se organizan para conseguir un propósito).

UML surge como una herramienta de gran aceptación cuando es necesario soportar el diseño y la implementación de una solución automatizada, que subyace en un modelo de gestión de cualquier sistema. Para ello se debe tener la documentación apropiada para su desarrollo y su mantenimiento subsiguiente o eventuales modificaciones.



Licenciatura en Informática Administrativa 2022A

Lo anterior resulta deseable y debe tenerse en cuenta en las representaciones visuales del sistema para su adecuada operación y un mejor entendimiento de los diseños (Basile et al., 2009).

Diagramas UML:

El UML está compuestas por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. El modelo UML de un sistema es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice como implementar dicho sistema. (Schmuller.J,1998).

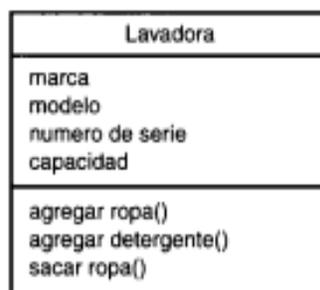
Diagrama de clase:

Es una categoría o grupo de cosas que tienen atributos y acciones similares. He aquí un ejemplo: cualquier cosa dentro de la clase lavadoras tiene atributos como son la marca, el modelo, el número de serie y la capacidad. Entre las acciones de las cosas de esta se encuentra “agregar ropa”, “agregar detergente”, “activarse” y “sacar ropa”

La figura 1.1 muestra un ejemplo de la notación del UML. Que captura los atributos y acciones de una lavadora. Un rectángulo es el símbolo que representa a la clase, y se divide en tres áreas.

El área superior contiene el nombre, el área central contiene los atributos. Y el área inferior las acciones.

Tabla 4. El símbolo UML de una clase



Fuente: Schmuller (1998)

Los diagramas de clase facilitan las representaciones a partir de las cuales los desarrolladores podrán trabajar. A su vez los diagramas colaboran en lo referente al



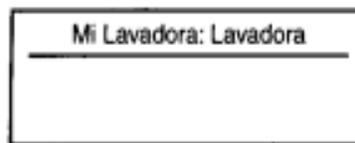
Licenciatura en Informática Administrativa 2022A

análisis. Permiten al analista hablarles a los clientes en su propia terminología, lo cual hace posible que los clientes indiquen importantes detalles de los problemas que requieren ser resueltos.

Diagrama de Objetos:

Es una instancia de clase (Una entidad que tiene valores específicos de los atributos y acciones).

Tabla 5. El símbolo UML del Objeto



Fuente: Schmuller (1998)

La figura 1.2 muestra la forma que el UML representa a un objeto. El símbolo es un rectángulo, como una clase, pero el nombre este subrayado. El nombre de la instancia específica se encuentra a la izquierda de los dos puntos (:), y el nombre de la clase a la derecha.

Diagrama de casos de uso:

Es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, esta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general (no solo por expertos en computación).

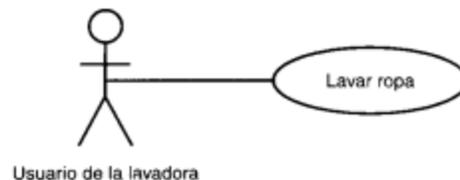


Figura 17. Diagrama de casos de un UML

Fuente: Schmuller (1998)

A la figura correspondiente al usuario de lavadora se le conoce como actor. La elipse representa caso de uso. por lo cual la entidad que inicia el caso de uso puede ser una persona u otro sistema.



Licenciatura en Informática Administrativa 2022A

Diagrama de Estados:

Una lavadora podrá estar en la fase de remojo grabado, enjuague, centrifugada o apagada.

El diagrama de Estados UML que aparece en la figura 1.2 captura esta pequeña realidad. la pintura muestra las acciones de la lavadora de un estado al otro.

El símbolo que está en la parte superior de la estructura representa el estado inicial y el de la parte inferior el estado final.

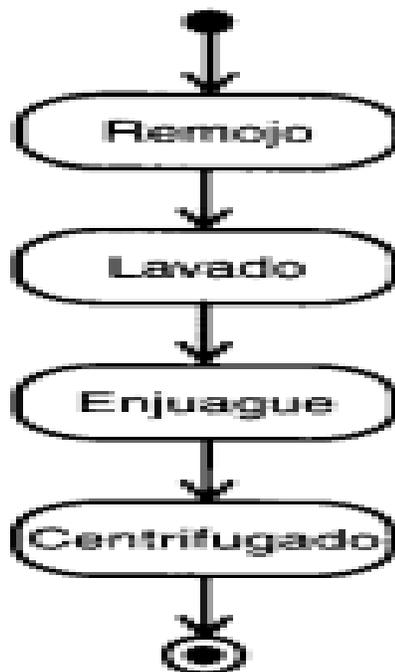


Figura 18. Diagrama de estados UML.

Fuente: Schmuller (1998)

Diagrama de secuencias:

Los diagramas de clase y los de objeto representan información estática. No obstante, en un sistema funciona el Rosal gestos e interactúan entre sí, tales interacciones suceden con el tiempo. el diagrama de secuencias UML muestra la metálica de la interacción con la base en tiempos.

La al final 1.4 representa un diagrama de secuencias que captura grandes interacciones que realizan a través del tiempo entre el establecimiento de agua, el tambor y entrenar

Licenciatura en Informática Administrativa 2022A

(representados como un rectángulo en la parte superior del diagrama). en este diagrama el tiempo se da de arriba hacia abajo.

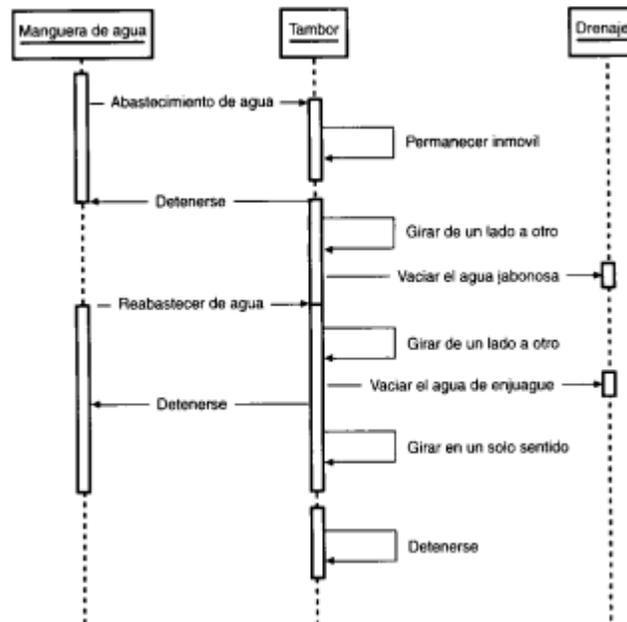


Figura 19. Diagrama de secuencias UML.

Fuente: Schuller (1998)

Diagrama de actividades

Las actividades que ocurren dentro de un caso de uso o dentro del comportamiento de un objeto se dan, normalmente en secuencia, como en los 11 pasos de la sesión anterior. la figura 1.3 Muestra la forma en el que diagrama de actividades UML representa los pasos del cuatro al 6 de tal secuencia.

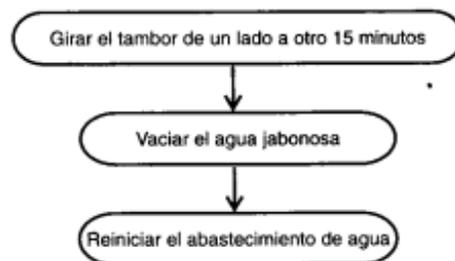


Figura 20. Diagrama de actividades UML.

Fuente: Schuller (1998)

Licenciatura en Informática Administrativa 2022A

Diagrama de colaboraciones:

Los elementos de un sistema trabajan en conjunto para cumplir con los objetivos del sistema y un lenguaje de modelo deberá contar con una forma de representar esto. el diagrama de colaboraciones UML, Diseñado con este fin, se muestra en la tabla 1.6 Este ejemplo agrega un cronómetro interno al conjunto de clases que constituyen a una lavadora. luego de cierto tiempo, el cronómetro detendrá el flujo de agua y el tambor comenzará a girar de un lado a otro.



Figura 21. Diagrama de colaboraciones UML.

Fuente: Schmuller (1998)

Diagrama de componentes:

Está íntimamente ligado a los sistemas informáticos, el moderno desarrollo de software se realiza mediante componentes, lo que es particularmente importante en los procesos de desarrollo en equipo. sin extenderse mucho en este punto te mostrará en la tabla 1.7 la manera en que el UML representa un componente de software.

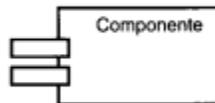


Figura 22. Diagrama de componentes UML.

Fuente: Schmuller (1998)

Diagrama de distribución:

Muestra la arquitectura física de un sistema informático. puede representar los equipos y los dispositivos, mostrar sus interconexiones y el software que se encontrará en cada máquina. cada computadora está representada por un cubo y las interacciones entre las computadoras están representadas por líneas que conectan a los cubos.



Licenciatura en Informática Administrativa 2022A

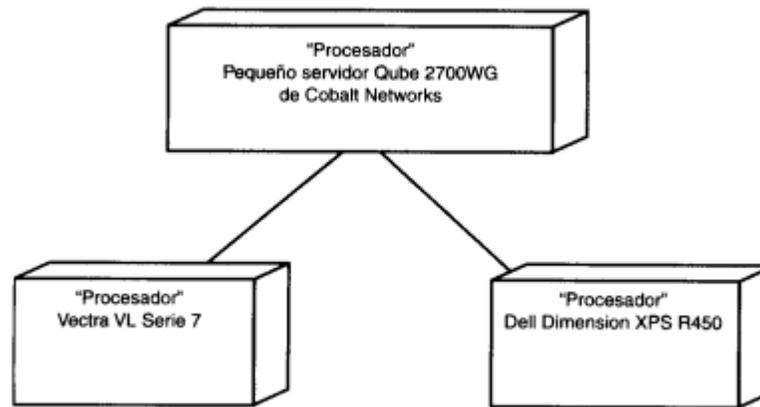


Figura 23. Diagrama de distribución UML.

Fuente: Schmuller (1998)

1.4.2 Diagramas de diseño (WIREFRAME, MOCK-UP)

El Lenguaje Unificado de Modelado (UML) desempeña un rol importante no solo en el desarrollo de software, sino también en los sistemas que no tienen software en muchas industrias, ya que es una forma de mostrar visualmente el comportamiento y la estructura de un sistema o proceso. El UML ayuda a mostrar errores potenciales en las estructuras de aplicaciones, el comportamiento del sistema y otros procesos empresariales.

UML se implementó por primera vez en la década de los 90 gracias a tres ingenieros de software: Grady Booch, Ivar Jacobson y James Rumbaugh. Querían desarrollar una forma menos caótica de representar el cada vez más complejo desarrollo de software, a la vez que separaban la metodología del proceso.

¿Cuáles son las ventajas del UML?

- Simplifica las complejidades
- Mantiene abiertas las líneas de comunicación
- Automatiza la producción de software y los procesos
- Ayuda a resolver los problemas arquitectónicos constantes
- Aumenta la calidad del trabajo
- Reduce los costos y el tiempo de comercialización

Licenciatura en Informática Administrativa 2022A

Tipos de diagramas UML

Existen dos tipos principales de diagramas UML: diagramas de estructura y diagramas de comportamiento (y dentro de esas categorías se encuentran varios otros). Estas variaciones existen para representar los numerosos tipos de escenarios y diagramas que usan los diferentes tipos de personas.

Desde clientes y gestores de proyectos hasta autores técnicos, diseñadores, analistas, codificadores y encargados de pruebas y control de calidad, cada rol utilizará un diagrama específico que se adapte a sus necesidades. Cada disposición requiere un enfoque y nivel de detalle diferente. El objetivo es que el UML exprese visualmente diagramas que sean fáciles de entender para todos. (Microsoft, 2019)

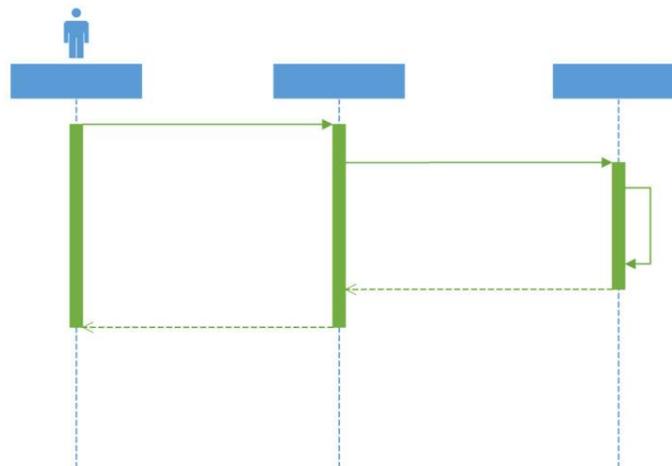


Figura 24. Ejemplo de secuencia básica del UML.

Fuente: Microsoft, (2019)

Wireframe

El wireframe es un prototipo de página web o de una aplicación. el wireframe sirve como una guía para auxiliar para la diagramación de la página.

La creación de bocetos también se conoce como wireframe o schematic y pretende organizar, de manera esquemática, la representación de una página web. En esa representación no se suelen incluir imágenes, contenido ni comportamiento de las páginas, ya que lo que pretenden es dar una visión global de los elementos que tendrá una página y de cómo se ubicarán dentro de ésta. No debería tener colores, ser lo más simple posible,



Licenciatura en Informática Administrativa 2022A

suficientemente explicativo en cuanto a las áreas que se definan y no combinar diferentes tipografías.

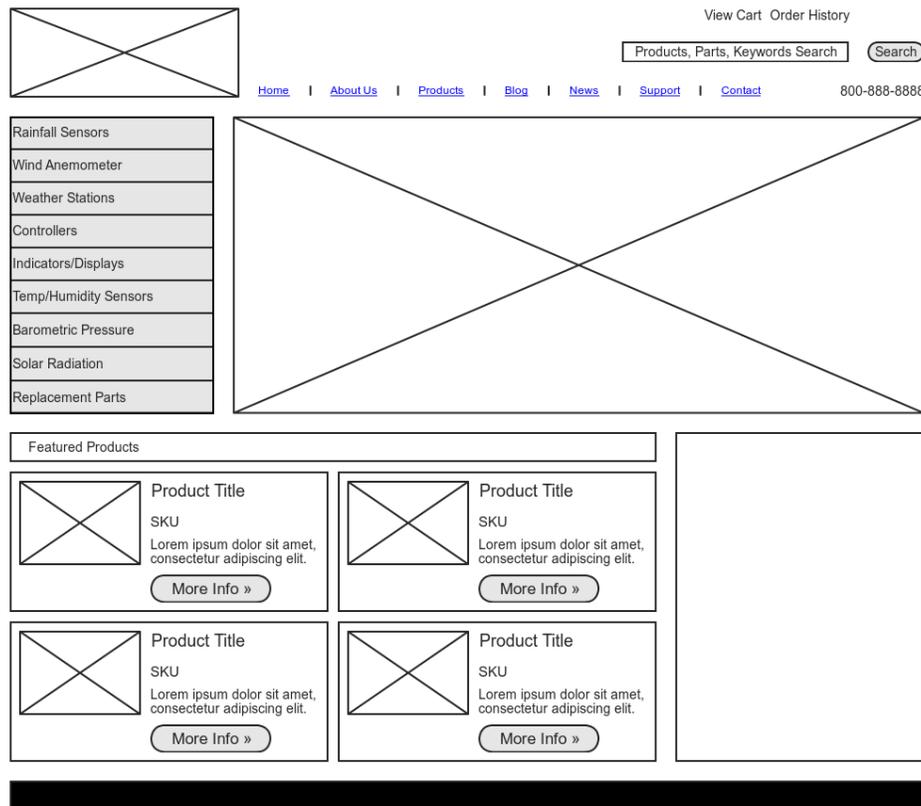


Figura 25. Creación de Wireframe

Fuente: Dext, et al (2018)

Los principios que se declaran en el wireframe deben ser repetidos en el resto de las páginas, de tal forma que, si existe un equipo trabajando en el desarrollo de la web, todos tengan claro lo que hacer. El objetivo fundamental es diseñar contenidos, no el aspecto visual del sitio, por lo que en la creación del wireframe nos debemos centrar en organizar la información a nivel de página. La información que debe contener un wireframe (según StrangeSystems) es:

- Contenido: qué contenido debe estar en la página.
- Elementos: cabeceras, pies de página, imágenes, formulario, etiquetado.
- Diseño: dónde se coloca cada elemento y cómo se agrupan los datos dentro de ellos. Por ejemplo: dónde poner exactamente el pie de página y qué datos debe tener (fecha, derechos de autor, datos de contacto...). Además, incluye el bloque

Licenciatura en Informática Administrativa 2022A

dedicado a barras laterales, barras de navegación (tanto principal como secundaria) y área de contenido.

- Comportamiento: no se debe emular el comportamiento (esto no es un prototipo), sino más bien indicar la cantidad de elementos que se deben mostrar en cada área o qué pasará cuando se active un enlace determinado.

Existe una variante de los wireframes denominada wireflows que, además de incluir el wireframe, añade información sobre el comportamiento y el flujo de navegación de la página. De ahí su nombre: la suma de wireframe y flowcharts.

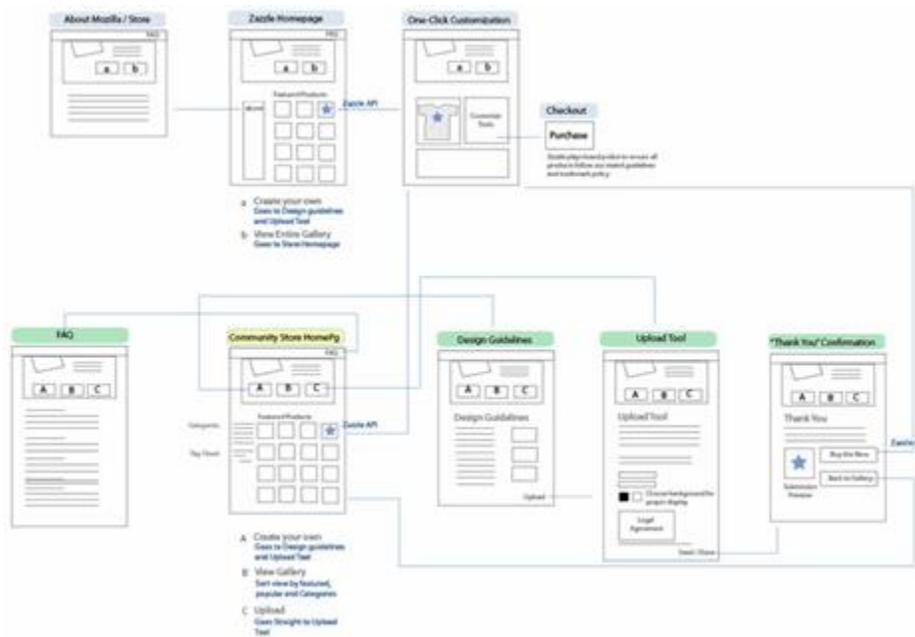


Figura 26. Wireframe.

Fuente: Dext, et al (2018)

¿Cómo hacer un wireframe?

Aunque existe bastante literatura al respecto, yo suelo recomendar una que es algo antigua (2004), pero que marca muy bien los pasos a seguir. En ella se dan seis pasos recomendables para la creación de un wireframe:

- Haz una lista numerada con toda la información que quieres incluir en la página.
- Luego deberías agrupar todos los elementos que tengan relación entre sí, y asignarles algún número para relacionarlas.

Licenciatura en Informática Administrativa 2022A

- A continuación, decidir qué grupo contiene información importante, necesaria, y complementaria.
- Diseña cada grupo: su ubicación, tanto dentro de la página como dentro del mismo grupo.
- Junta los grupos en función al diseño y al equilibrio de la página en global
- implementa todo esto por medio de una herramienta. De esto se infiere que los puntos anteriores deberías hacerlos en papel para, a continuación, pasarlos a ordenador.

Mockup

El wireframe conecta la estructura conceptual con la visual y para hacer el diseño visual del sitio se suelen emplear mockups. El mockup es un diseño de calidad media o alta. Hay más colores, diferentes tipos de letra, fondos... en definitiva, todo aquello que permita representar la estructura de la información y demostrar las funcionalidades básicas. Se trata de un modelo, generalmente a escala, de un diseño determinado. (Senso, 2015).

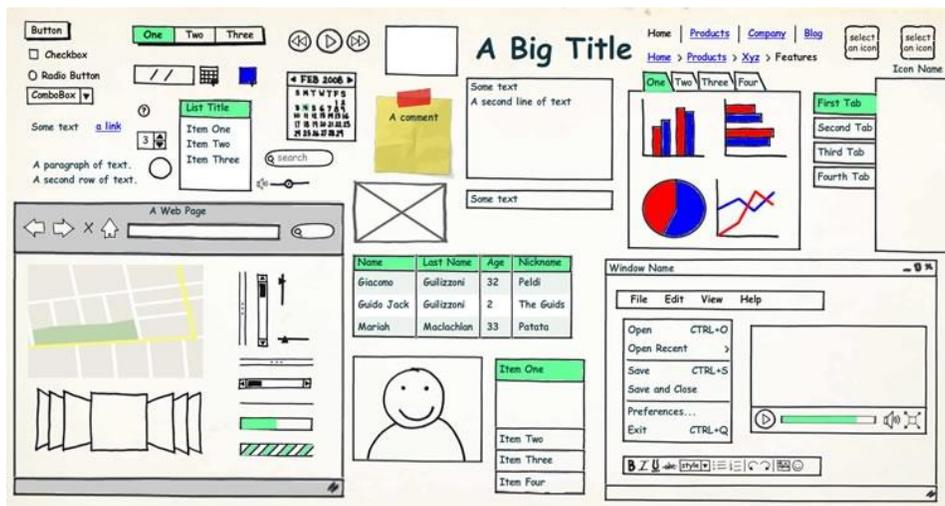


Figura 27. Ejemplo de creación de mockup.

Fuente: Senso, (2015)

Herramientas para crear wireframe y mockup.

- Adobe Illustrator.
- Microsoft Visio.
- Apache OpenOffice Draw.
- Omnigraffe (para Mac).



Licenciatura en Informática Administrativa 2022A

- Balsamiq.
- Gomockingbird.
- Hotgloo.
- Invisionapp.
- Mockflow.

REFERENCIAS

Altamirano, J. (2012). *Aplicaciones (APIs) usadas en cómputo* (2.^a ed.). México: 75-99. <https://bit.ly/3pd6pKI>

Cardona Taltavull, J. A. (2015). *PROGRAMACIÓN LÓGICA*. Dipòsit Digital de la Universitat de Barcelona. Treballs Finals de Grau (TFG) - Enginyeria Informàtica. <https://bit.ly/3vepmRb>

Clinger, W. [1989]. Multitasking and macscheme+toolsmith, MacTech: The Journal of Apple Technology 3(12). <https://bit.ly/35hdq6e>

Developer Mozilla (2021). *Entendiendo los frameworks de JavaScript del lado del cliente*. Aprende sobre desarrollo web | MDN. <https://mzl.la/3urUCKo>

Eames, C. (2017). *Una guía para el diseño de API y REST* (1.^a ed., pp. 22–69). España. <https://bit.ly/35tSkBK>

Elena, D. A., Avila, S., & Riojas Martínez, A. (n.d.). *Utilidad del Lenguaje Unificado de Modelado (UML) en el desarrollo de software profesional dentro del sector empresarial y educativo*. Uadec.Mx. <https://bit.ly/3liqCp4>

Equipo Geek (2020). *Los mejores JavaScript Frameworks*. <https://bit.ly/37M4Opd>

Fernández Alarcón, V. (2006). *Desarrollo de sistemas de información : una metodología basada en el modelado*. UPC.

Fernández Y. & Díaz Y. (2012), Patrón Modelo-Vista-Controlador: *Revista Telem@tica*. Vol. 11. No. 1, p. 47-57. <https://bit.ly/3lhtOC1>

G. Assayag & A. Gerzso (eds), *New Computational Paradigms for Computer Music*, IRCAM/Delatour, chapter 2

Gómez, J. (2021). *Semántica de Lenguajes de Programación Declarativos: Una Breve Introducción*. A Computational Solution for the software refactoring problem View project Self-healing model for Distributed Environments Based on Artificial Life Technique View project. <https://bit.ly/3JL2MmX>

Joseph Schmuller. *Aprendiendo UML en 24 horas*. (n.d.). Google.com. <https://bit.ly/3vblStO>

Kendall, K. E., & Kendall, J. E. (2011). *Systems analysis and design*. Pearson Prentice Hall.



Licenciatura en Informática Administrativa 2022A

Lerma-Blasco, R. V., Murcia, J. A., & Talón, A. E. M. (2013). *Aplicaciones web*. Segunda Edición. McGraw-Hill/ Interamericana de España.

Lizama O., Kindley G. & Jeria M. I. J. (2016) *Redes de computadores Arquitectura Cliente – Servidor*, Universidad Técnica Federico Santa María. <https://bit.ly/33FvUN8>

Lonos. (2020, 24 de febrero). *Programación declarativa: cuando el qué es más importante que el cómo*. Digital Guide IONOS. <https://bit.ly/35keBS7>

Lloyd, J. [1994]. Practical advantages of declarative programming, Proceedings of the Joint Conference on Declarative Programming, pp. 3 – 17.

Marini, E. (2012). *El modelo cliente/servidor*. Obtenido de <https://bit.ly/3lo4TwE>

Microsoft. (24 de 09 de 2019). Obtenido de La guía sencilla para la diagramación de UML y el modelado de la base de datos: <https://bit.ly/34llrZ>

Pérez.M.F (2012). Desarrollo de un modelo estándar para los servicios ofrecidos por las interfaces de programación de aplicaciones (APIs) usadas en cómputo en la nube, Benemérita Universidad Autónoma de Puebla. <https://bit.ly/3CldjwU>

Pérez.M.F (2012). Desarrollo de un modelo estándar para los servicios ofrecidos por las interfaces de programación de aplicaciones (APIs) usadas en cómputo en la nube, Benemérita Universidad Autónoma de Puebla. <http://148.228.16.10/archivo/TES1599.pdf>

Plata, J. (2022). *¿Qué son las API?* (1.ª ed.). México: 79-90. <https://bit.ly/35o7icr>

Senso, J. A. (03 de 06 de 2015). *UNIVERSIDAD DE GRANADA* . Obtenido de Diseño web: wireframes y mockups: <https://bit.ly/3q6GZPq>

Van Roy, P. [2009]. *Programming paradigms for dummies: What every programmer should know in*.



Licenciatura en Informática Administrativa 2022A

UNIDAD II. TECNOLOGÍAS DE FRONTEND

OBJETIVO:

Aplicar las diferentes tecnologías de Front-end en sus versiones más actualizadas para el desarrollo y programación de soluciones.

2.1 Etiquetado y estructuración con HTML.

Aplicar las diferentes tecnologías de front-end en sus versiones más actualizadas para el desarrollo y HTML es un lenguaje de marcas. Como tal, debemos conocer las diferentes etiquetas y elementos que utiliza. Las etiquetas por lo general vienen emparejadas. Una etiqueta de apertura significará que el navegador debe tratar el nuevo texto utilizando las propiedades de esa etiqueta y una etiqueta de cierre indica donde terminan esas propiedades.

Cuando escribes código en HTML, estás escribiendo etiquetas HTML. Todas las etiquetas HTML están hechas con un número de partes específicas, incluyendo:

- El carácter "menor que" <
- Una palabra o carácter que determina qué etiqueta se está escribiendo
- Cualquier número de atributos HTML que se quiera usar, escritos de la forma nombre = "valor"
- El carácter "mayor que" >

La parte esencial de una etiqueta HTML es lo que se denomina la etiqueta de apertura. Se trata de escribir el nombre de la etiqueta en cuestión, colocándola entre los caracteres

< y >. Aunque no es terminantemente obligatorio, se recomienda y considera una buena costumbre escribir las etiquetas siempre en minúsculas.

La estructura de las etiquetas HTML tiene el siguiente formato:

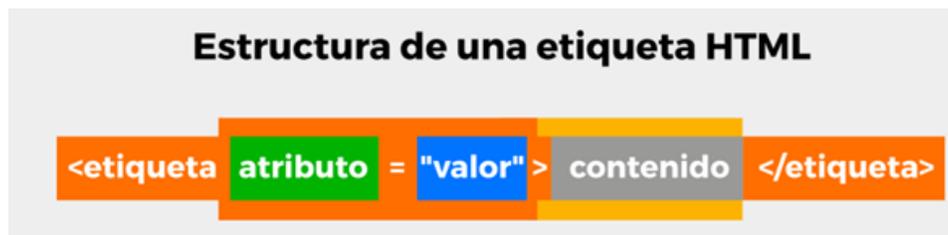


Figura 28. Estructura de una etiqueta HTML

Fuente: Roma, J. (s.f).



Licenciatura en Informática Administrativa 2022A

En HTML5 no se puede colocar cualquier palabra como etiqueta, sino que existen una serie específica de etiquetas, cada una con una misión y objetivo diferente. Por ejemplo, la etiqueta ``:

`Contenido`

Como se puede ver, la mayoría de las etiquetas requieren que se especifique un cierre de etiqueta para saber dónde termina de actuar. Se caracteriza en que se escribe igual que la etiqueta de apertura, pero con la barra / inmediatamente después del <.

¿Dónde deben ir las etiquetas?

Cualquier contenido que sea visible en tu página web debe ir entre la etiqueta de `<body>` que abre y la etiqueta de `</body>` que cierra, como se muestra a continuación:

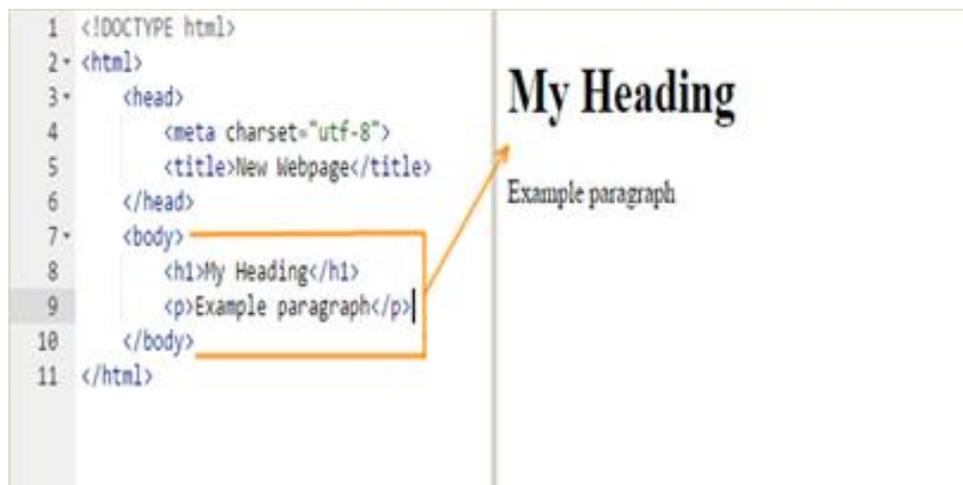


Figura 29. Ejemplo de una etiqueta.

Fuente: Roma, J. (s.f).

Atributo

En algunas etiquetas HTML, existen algunos atributos específicos (que pueden ser opcionales u obligatorios). Los atributos determinan cierta información sobre la etiqueta (o su modo de actuar) y generalmente van asociados a un valor determinado. Este par atributo-valor se escribe después del nombre de la etiqueta, separándola por espacio y antes del carácter > de la etiqueta de apertura:



Licenciatura en Informática Administrativa 2022A

```
<strong id="dato">Contenido</strong>
```

Figura 30. Etiqueta Strong con atributo.

Fuente: Roma, J. (s.f).

En este caso, la etiqueta sería strong, el atributo id, el valor de id sería dato, mientras que el contenido de la etiqueta es Contenido. Por otro lado, y al igual que las etiquetas, cada atributo tiene una misión y comportamiento concreto. Aunque los valores pueden ir rodeados por comillas simples, se recomienda escribir el valor siempre entre comillas dobles.

Existen 3 tipos de atributos dependiendo de sus valores:

Conjunto finito de valores: Son los atributos en los que puedes especificarle una serie de valores limitada. Cualquier otro valor que no sea uno de ellos, no será válido.

Valores libres: Son los atributos en los que puedes especificar un valor libremente, como una dirección URL o un texto, y no existe una serie de valores específicos para escribir.

Valores booleanos: Son los atributos que deben tener un valor verdadero (true) o un valor falso (false). En HTML5 estos atributos se escriben sin valor (solo el atributo) si son verdaderos y se omiten para indicar que son falsos.

Una etiqueta puede contener desde un fragmento de texto hasta un grupo de etiquetas. Esto depende mucho de la etiqueta a utilizar, y se verá más adelante en el apartado de etiquetas. Un posible ejemplo sería el siguiente:

```
<div id="pagina">  
  <strong>Contenido importante</strong>  
</div>
```

Figura 31. Estructura de una etiqueta de dos fragmentos

Fuente: Roma, J. (s.f).



Licenciatura en Informática Administrativa 2022A

Nótese que dentro de la etiqueta <div> vemos que no sólo hay un fragmento de texto, sino que además incluye otra etiqueta, en este caso. Esto ocurre en las etiquetas de agrupación.

2.2 Presentación de información con CSS.

Para Ermerson (2014) CSS es el lenguaje que permite describir la presentación de páginas Web, incluidos los colores, el diseño y las fuentes y permite adaptar la presentación a los diferentes tipos de dispositivos como pantallas grandes, pequeñas o impresoras.

CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML con la separación de HTML de CSS facilita el mantenimiento los sitios web compartir las hojas de estilo a través de páginas y la medida para diferentes entornos. Esto se conoce como la separación de la estructura (contenido) y la presentación.

Esta especificación define las Hojas de Estilo en Cascada, Nivel 2 revisión 2

(CSS 2.2 “2014”). Con CSS podemos definir la forma de presentación aplicada a una página web entera, un documento HTML es una porción del documento o una etiqueta en concreto en CSS.

Puede definir:

- Definir la distancia entre líneas del documento
- Aplicar (sangría) a las primeras líneas del párrafo
- Colocar elementos en la página con mayor precisión
- Definir la visibilidad de los elementos, márgenes, subrayados, tachados.

Algunos problemas

Las especificaciones anteriores del lenguaje tenían muchas utilidades para aplicar estilos a las webs, pero los desarrolladores aún continúan usando trucos diversos para conseguir efectos tan comunes o deseados como los bordes redondeados o el sombreado de elementos en la página.

Estos trucos pasaban por cambiar el contenido de las páginas web lo que rompía con la filosofía de CSS de separar el contenido de la forma.



Licenciatura en Informática Administrativa 2022A

CSS3

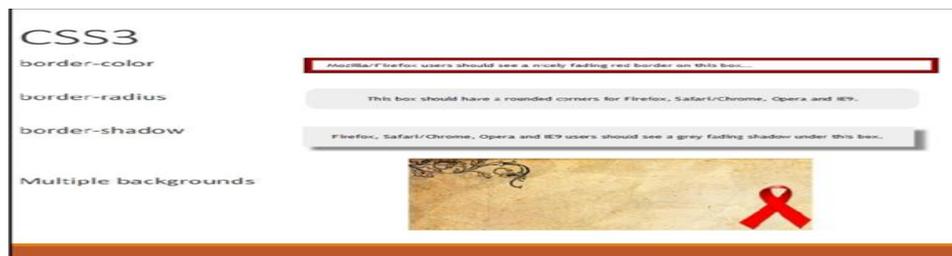


Figura 32. CSS3

Fuente: Cordovall,E(2014)

EJEMPLO HTML + CSS

Paso 1. Escribir el código

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Mi primera página con estilo</title>
</head>
<body>

<!-- Menú de navegación del sitio -->
<ul class="navbar">
<li><a href="indice.html">Página principal</a>
<li><a href="meditaciones.html">Meditaciones</a>
<li><a href="ciudad.html">Mi ciudad</a>
<li><a href="enlaces.html">Enlaces</a>
</ul>

<!-- Contenido principal -->
<h1>Mi primera página con estilo</h1>

<p>¡Bienvenido a mi primera página con estilo!

<p>No tiene imágenes, pero tiene estilo.
También tiene enlaces, aunque no te lleven a
ningún sitio..

<p>Debería haber más cosas aquí, pero todavía no sé
qué poner.

<!-- Firma y fecha de la página, sólo por cortesía! -->
<address>Creada el 17 de febrero de 2012<br>
para curso de Programación Web. <br>
ETSI en Topografía, Geodesia y Cartografía - UPM
</address>
</body>
</html>
```

Figura 33. Código

Fuente: Gonzalez,R (2017)



Licenciatura en Informática Administrativa 2022A

Paso 2. Añadir colores

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Mi primera página con estilo</title>
<style type="text/css">
body {
color: purple;
background-color: #d8da3d }
</style>
</head>
<body>
<!-- Menú de navegación del sitio -->
<ul class="navbar">
<li><a href="indice.html">Página principal</a>
<li><a href="meditaciones.html">Meditaciones</a>
<li><a href="ciudad.html">Mi ciudad</a>
<li><a href="enlaces.html">Enlaces</a>
</ul>
<!-- Contenido principal -->
<h1>Mi primera página con estilo</h1>
<p>iBienvenido a mi primera página con estilo!
<p>No tiene imágenes, pero tiene estilo.
También tiene enlaces, aunque no te lleven a
ningún sitio...
<p>Debería haber más cosas aquí, pero todavía no sé
qué poner.
```

Figura 34. Asignación de color

Fuente: Gozalez,R (2017)

Paso 3. Añadir tipo de letra

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Mi primera página con estilo</title>
<style type="text/css">
body {
font-family: Georgia, "Times New Roman",
Times, serif;
color: purple;
background-color: #d8da3d }
h1 {
font-family: Helvetica, Geneva, Arial,
SunSans-Regular, sans-serif }
</style>
</head>
<body>|
<!-- Menú de navegación del sitio -->
<ul class="navbar">
<li><a href="indice.html">Página principal</a>
<li><a href="meditaciones.html">Meditaciones</a>
<li><a href="ciudad.html">Mi ciudad</a>
<li><a href="enlaces.html">Enlaces</a>
</ul>
<!-- Contenido principal -->
<h1>Mi primera página con estilo</h1>
<p>iBienvenido a mi primera página con estilo!
<p>No tiene imágenes, pero tiene estilo.
```

Figura 35. Tipo de letra

Fuente: Gonzalez,R (2017)

Licenciatura en Informática Administrativa 2022A

Paso 4. Añadir una barra de navegación

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Mi primera página con estilo</title>
<style type="text/css">
body {
padding-left: 11em;
font-family: Georgia, "Times New Roman",
Times, serif;
color: purple;
background-color: #d8da3d }
ul.navbar {
position: absolute;
top: 2em;
left: 1em;
width: 9em }
h1 {
font-family: Helvetica, Geneva, Arial,
SunSans-Regular, sans-serif }
</style>
</head>
<body>
<!-- Menú de navegación del sitio -->
<ul class="navbar">
<li><a href="indice.html">Página principal</a>
<li><a href="meditaciones.html">Meditaciones</a>
<li><a href="ciudad.html">Mi ciudad</a>
<li><a href="enlaces.html">Enlaces</a>
</ul>
```

Figura 36. Barra Navegación

Fuente: Gonzalez,R (2017)

Paso 5. Dar estilo a los enlaces

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Mi primera página con estilo</title>
<style type="text/css">
body {
padding-left: 11em;
font-family: Georgia, "Times New Roman",
Times, serif;
color: purple;
background-color: #d8da3d }
ul.navbar {
list-style-type: none;
padding: 0;
margin: 0;
position: absolute;
top: 2em;
left: 1em;
width: 9em }
h1 {
font-family: Helvetica, Geneva, Arial,
SunSans-Regular, sans-serif }
ul.navbar li {
background: white;
margin: 0.3em 0;
padding: 0.3em;
border-right: 1em solid black }
ul.navbar a {
text-decoration: none }
a:link {
color: blue }
a:visited {
color: purple }
</style>
</head>
```

Figura 37. Enlaces

Fuente: Gonzalez,R (2017)



Licenciatura en Informática Administrativa 2022A

Paso 6. Añadir una línea horizontal

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Mi primera página con estilo</title>
<style type="text/css">
body {
padding-left: 11em;
font-family: Georgia, "Times New Roman",
Times, serif;
color: purple;
background-color: #d8da3d }
ul.navbar {
list-style-type: none;
padding: 0;
margin: 0;
position: absolute;
top: 2em;
left: 1em;
width: 9em }
h1 {
font-family: Helvetica, Geneva, Arial,
SunSans-Regular, sans-serif }
ul.navbar li {
background: white;
margin: 0.5em 0;
padding: 0.3em;
border-right: 1em solid black }
ul.navbar a {
text-decoration: none }
a:link {
color: blue }
a:visited {
color: purple }
address {
margin-top: 1em;
padding-top: 1em;
border-top: thin dotted }
</style>
</head>
```

Figura 38. Código para una línea horizontal

Fuente: Gonzalez,R (2017)

Paso 7. Poner la hoja de estilo en un archivo separado

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Mi primera página con estilo</title>
<link rel="stylesheet" href="miestilo.css">
</head>
<body>
<body>
<!-- Menú de navegación del sitio -->
<ul class="navbar">
<li><a href="indice.html">Página principal</a>
<li><a href="meditaciones.html">Meditaciones</a>
<li><a href="ciudad.html">Mi ciudad</a>
<li><a href="enlaces.html">Enlaces</a>
</ul>
<!-- Contenido principal -->
<h1>Mi primera página con estilo</h1>
<p>¡Bienvenido a mi primera página con estilo!
<p>No tiene imágenes, pero tiene estilo.
También tiene enlaces, aunque no te lleven a
ningún sitio...
<p>Debería haber más cosas aquí, pero todavía no sé
qué poner.
<!-- Firma y fecha de la página, sólo por cortesía! -->
<address>Creada el 17 de febrero de 2012<br>
```

Figura 39. Hoja de estilo

Fuente: Gonzalez,R (2017)



Licenciatura en Informática Administrativa 2022A

Paso 8. Presentación de CSS



Figura 40. Presentación

Fuente: Gonzalez,R (2017)

2.3 Programación de Funcionalidad con JavaScript

La programación funcional es un paradigma de programación o un estilo de programación que se basa en gran medida en el uso de funciones puras y aisladas. (Azaustre, 2016).

El uso de funciones es el componente principal de la programación funcional, pero el mero uso de funciones no se traduce en programación funcional. En la programación funcional, utilizamos funciones puras, que son funciones que no tienen efectos secundarios.

Hay cuatro tipos principales de funciones:

- **Funciones de primera clase:** En JavaScript, todas las funciones son funciones de primera clase. Eso significa que pueden tratarse como cualquier otra variable. Las funciones de primera clase son funciones que pueden asignarse como valores a variables, devolverse desde otras funciones y pasarse como argumentos a otras funciones.
- **Funciones Callback:** Las funciones callback son funciones que se pasan a otras funciones como argumentos y son llamadas por la función en la que se pasan. Simplemente, las funciones callback son funciones que escribimos como argumentos en otras funciones. No podemos invocar funciones callback, se invocan cuando se llama a la función principal en la que se pasaron como argumentos.



Licenciatura en Informática Administrativa 2022A

- **Funciones de orden superior (High Order Functions):** Las funciones de orden superior son funciones que reciben otras funciones como argumentos o devuelven una función.
- **Funciones asíncronas:** Las funciones asíncronas son funciones que no tienen nombre y no se pueden reutilizar. Estas funciones se escriben normalmente cuando necesitamos realizar algo una vez y en un solo lugar.

Principios de la programación funcional:

Evite mutaciones y efectos secundarios

El primer principio de la programación funcional es evitar cambiar las cosas. Una función no debería cambiar nada, como una variable global. Esto es muy importante porque los cambios a menudo provocan errores. Si una función cambia una variable global, por ejemplo, podría dar lugar a un comportamiento inesperado en todos los lugares donde se utiliza esa variable.

El segundo principio es que una función debe ser pura, lo que significa que no tiene efectos secundarios. En la programación funcional, los cambios que se realizan se denominan mutaciones y los resultados se denominan efectos secundarios. Una función pura no hace ninguna de las dos. Una función pura siempre tendrá la misma salida para la misma entrada.

Si una función depende de una variable global, esa variable debe pasarse a la función como argumento. Esto nos permite obtener la misma salida para la misma entrada.

Abstracción

Las abstracciones ocultan detalles y nos permiten hablar sobre problemas en un nivel superior sin describir todos los detalles de implementación del problema.

Las funciones nos permiten lograr algo similar. Podemos crear funciones para tareas que es más probable que repitamos una y otra vez. Las funciones nos permiten crear nuestras propias abstracciones.

Además de crear nuestras propias abstracciones, ya se han creado algunas funciones para que abstraigamos tareas que es más probable que hagamos una y otra vez.

Filtrando Arreglos

Cuando trabajamos con estructuras de datos como arreglos, es más probable que nos encontremos en una situación en la que solo estemos interesados en ciertos elementos del arreglo. Para obtener estos elementos podemos crear fácilmente una función para realizar la tarea.



Licenciatura en Informática Administrativa 2022A

Por ejemplo:

```
function arregloFiltro(arreglo, prueba) {  
  const arregloFiltrado = [];  
  for (let elemento of arreglo) {  
    if (prueba(elemento)) {  
      arregloFiltrado.push(elemento);  
    }  
  }  
  return arregloFiltrado;  
};  
  
const arregloMezclado = [1, true, null, "Hola", undefined, "Mundo", false];  
const soloCadenas = arregloFiltro(arregloMezclado, elemento => typeof elemento ===  
'cadena');  
onlyCadenas; // ['Hola', 'Mundo']
```

arregloFiltro es una función que acepta un arreglo y una función callback. Recorre el arreglo y agrega los elementos que pasan la prueba en la función callback en un arreglo llamada arregloFiltrado.

Con esta función podemos filtrar un arreglo y devolver los elementos que nos interesan, como en el caso de arregloMezclado.

Transformar Elementos del arreglo con Map

Imagine otro escenario en el que tenemos una serie de elementos, pero nos gustaría realizar una determinada operación en todos los elementos. Podemos escribir una función para hacer esto por nosotros:

```
function transformarArreglo(arreglo, prueba) { const arregloTransformado = []; for (let  
elemento of arreglo) { arregloTransformado.push(prueba(elemento)); } return  
arregloTransformado;};const edades = [12, 15, 21, 19, 32];const doblarEdades =  
transformarArreglo(edades, edad => edad * 2);doblarEdades; // [24, 30, 42, 38, 64];
```



Licenciatura en Informática Administrativa 2022A

Así, hemos creado una función que recorre cualquier arreglo dada y transforma todos los elementos del arreglo en función de la función callback que proporcionamos.

Pero nuevamente, esto se volvería tedioso si tuviéramos que reescribir la función en 20 programas diferentes. Afortunadamente los arreglos tienen un método estándar llamado `map` que hace exactamente lo mismo. Aplica la función callback en todos los elementos del arreglo dada y luego devuelve un nuevo arreglo.

Reducir arreglos con `Reduce`

El método `reduce` se utiliza para reducir un arreglo a un solo valor en función de la función callback que proporcionamos. También toma un segundo argumento opcional que especifica desde dónde queremos que comience la operación callback.

La función callback que proporcionamos en la función de reducción (`reduce`) tiene dos parámetros. El primer parámetro es el primer elemento del arreglo de forma predeterminada.

De lo contrario, es el segundo argumento que proporcionamos en el método de reducción (`reduce`), y el segundo parámetro es el elemento actual del arreglo.

Todos estos conceptos están relacionados con la programación funcional. Pese a no ser un concepto nuevo, en los últimos tiempos se puede ver cómo hay vientos de cambio que apuntan directamente hacia este nuevo paradigma, como por ejemplo Java 8 o Redux y sus `reducers` con funciones puras.

La programación funcional es un paradigma de programación declarativa, es decir, una forma de afrontar el mismo problema con un punto de vista diferente; cuyo objetivo es centrarse en describir qué hacer, y no en cómo hacerlo.

Un ejemplo de programación declarativa es SQL, donde decimos qué queremos obtener con las consultas y no en qué manera hay que hacerlo. Uno de los principales beneficios es la legibilidad, al centrarse en qué pasos hay que hacer y no en cómo hacerlos, eliminando el código repetitivo que no aporta valor al producto; y que, junto con alguno de los conceptos, lo que nos permite obtener un código menos propenso al error, facilitar los test unitarios y convertir en predecible su comportamiento.

First Class Functions

En JavaScript las funciones son valores, como cualquier otro tipo de dato. Por ejemplo, pueden ser almacenadas en variables, objetos o arrays, así como ser pasadas como argumentos a otras funciones.



Licenciatura en Informática Administrativa 2022A

```
var plus1 = a => a + 1;

plus1(2); // 3

var myArray = [0, 1, 'a', null, plus1];
myArray[4](2); // 3

var obj = {
  increment: plus1
}
obj.increment(2); // 3

function incrementBy2(fnArg, num) {
  return fnArg(num);
}
incrementBy2(plus1, 2); // 3
```

Figura 41. Ejemplo de variables en JavaScript

Fuente: Sapetti (2022)

Vale la pena destacar en este punto que a veces, por prisa o falta de atención, se pueden ver el siguiente tipo de expresiones:

```
var incNum = (num) => {
  return plus1(num);
}
```

Figura 42. Ejemplo en JavaScript

Fuente: Sapetti (2022)

La función plus1 ya recibe un número como argumento, así que podríamos eliminar un nivel y dejar de esta manera la expresión:

```
var incNum = plus1
```

Figura 43. Ejemplo de función Plus1

Fuente: Sapetti (2022)

Como, por ejemplo, con las Promesas de ES6:



Licenciatura en Informática Administrativa 2022A

```
getArtists
.then( artists => {
  return parseArtistsResponse(artists);
});
```

Figura 44. Promesas de ES6

Fuente: Sapetti (2022)

Podríamos reemplazarlo por:

```
getArtists
.then(parseArtistsResponse);
```

Figura 45. Promesas de ES6

Fuente: Sapetti (2022)

Otro caso común, que se ve mucho, es con las llamadas AJAX. Puede que resulte excesivo visto aquí como ejemplo, pero podemos encontrar código como el siguiente:

```
var getArtists = function(parseFn) {
  return artistsAjaxCall(function(json) {
    return parseFn(json);
  });
};
```

Figura 46. Llamadas AJAX

Fuente: Sapetti (2022)

No es complejo, es importante revisarlo de nuevo ya que este punto es muy importante para lo que viene por delante:

La función `artistsAjaxCall` recibe como argumento una función que, a su vez, recibirá un argumento (`json`). En este caso `parseFn` ya es una función que espera un argumento (`json`), así que podemos eliminar ese nivel y este sería el resultado:

```
var getArtists = function(parseFn) {
  return artistsAjaxCall(parseFn);
};
```

Figura 47. Ejemplo en JavaScript

Fuente: Sapetti (2022)



Licenciatura en Informática Administrativa 2022A

Ahora tenemos `getArtists`, es una función que espera una función como argumento (callback) para hacer la llamada de AJAX. De nuevo `artistsAjaxCall` ya es una función que recibe otra función (callback) como argumento. Así que eliminamos nivel y así quedaría el resultado:

```
var getArtists = artistsAjaxCall;
```

Figura 48. Ejemplo en JavaScript

Fuente: Sapetti (2022)

Pure functions

Una función pura es aquella que, en sucesivas llamadas para el mismo input, genera siempre el mismo output (es determinista). Además, no tiene efectos secundarios o side effects. Entendiendo esto último como un cambio en el sistema o una visible interacción con fuentes externas.

Ejemplos de side effect serían: cambiar los valores a los argumentos de entrada o variables externas a nuestra función, leer de disco, hacer llamadas http, acceder a base de datos, hacer logging en pantalla o fichero, leer el input del usuario, obtener la fecha actual... es decir, un cambio que nos limita a no hacer nada no relacionado con la propia función. Obviamente, un programa que no haga ninguna de las acciones consideradas impuras no aporta utilidad alguna, por ello, no es que se vayan a eliminar, pero sí que favoreceremos el uso de funciones puras.

En programación funcional se tiende a acotar el uso de side effects a puntos determinados, de forma que buscar el origen del error puede ser mucho más sencillo y el comportamiento del programa ser más predecible.

```
// Impure
var x = 1
var y = 2
...
var add = () => x + y
...
add() // 3?

// Pure
var add = (x, y) => x + y
...
add(1, 3) // 3
```



Licenciatura en Informática Administrativa 2022A

Figura 49. Ejemplo 1 Pure functions

Fuente: Sapetti (2022)

```
// Impure
var getDayOfWeek = () => {
  var d = new Date();
  return d.getDay();
}
getDayOfWeek()

// Pure
var getDayOfWeek = d => {
  return d.getDay();
}
getDayOfWeek(new Date())
```

Figura 50. Ejemplo 2 Pure functions

Fuente: Sapetti (2022)

```
//Impure
var someValue = [0];
...
var impureFn = (varByRef) => {
  ...
  varByRef.push(1); // Estamos alterando una variable No podemos saber si el valor que se ha modificado se esta
  // usando en otra parte del programa y estamos alterando el resultado esperado (causando algún efecto
  // secundario)
  ...
}
...
impureFn(someValue)
//someValue ahora es [0, 1]

//Pure
var someValue = [0];
...
var pureFn = (varByRef) => {
  ...
  var brandNewArray = varByRef.concat(1); //Concat a diferencia de push devuelve un nuevo array
  ...
  return brandNewArray;
}
...
var returnedArray = pureFn(someValue)
//someValue sigue siendo [0]
//returnedArray es [0, 1]
```

Figura 51. Ejemplo 3 Pure functions



Licenciatura en Informática Administrativa 2022A

Fuente: Sapetti (2022)

Referential Transparency

Las funciones puras nos dan transparencia referencial, es decir, que una función siendo invocada con el mismo input puede ser intercambiada por su output directamente sin alterar el funcionamiento del programa.

```
var add (a, b) => a + b
add(add(1,3), add(6, 2)) // 12
add(4, add(6, 2)) // 12
add(4, 8) // 12
12
```

Figura 52. Ejemplo Referential Transparency

Fuente: Sapetti (2022)

Por tanto, es más sencillo de interpretar tanto para los desarrolladores como para el compilador; nos da una mayor sencillez para testing; y puede ser un punto en el que podemos realizar micro optimizaciones usando la memorización (guardar el resultado para los mismos argumentos de entrada de una función, para que no se tenga que volver a ejecutar).

Shared state

También llamado por internet: "the root of evil". las funciones puras, el estado compartido es algo a erradicar de nuestras aplicaciones ya que hace que el comportamiento de las aplicaciones no sea determinista. Por lo tanto, provocan que sean más difíciles de comprender, de testear y solucionar posibles errores que vayan surgiendo.

De la mano con lo que vamos comentando, está el término inmutable (Immutability). Los datos deben ser inmutables para evitar shared state, side effects, impure functions... Todo lo que hagamos será sobre copias y no sobre los datos compartidos con más partes de la aplicación.

Higher Order Functions

las funciones en Javascript son valores, podemos enviarlas como argumentos a otras funciones o podrían ser el valor de retorno. Pues eso mismo define las Higher Order Functions: funciones que reciben funciones como argumentos y/o devuelven funciones como valor de retorno.

Licenciatura en Informática Administrativa 2022A

```
var ajaxCall = cb => {  
  ...  
  cb();  
}  
  
var thingsFactory = () => {  
  ...  
  return function() {  
    ...  
  }  
}
```

Figura 53. Ejemplo Higher Order Functions

Fuente: Sapetti (2022)

2.4 Frameworks CSS y JavaScript.

En general el término frameworks se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

Según Gutiérrez (2021), un frameworks se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Un frameworks de CSS es una biblioteca de estilos genéricos que puede ser usada para implementar diseños web. Aportan una serie de utilidades que pueden ser aprovechadas frecuentemente en los distintos diseños web.



Figura 54. Concepto de Frameworks.

Fuente: Yuste. A, (2013)

FRAMESWORKS CSS

Ventajas de un frameworks de CSS

Un frameworks de CSS, si está bien diseñado e implementado, proporciona las siguientes ventajas:

- Sustituir elementos predeterminados del navegador.



Licenciatura en Informática Administrativa 2022A

- Ayudar a hacer el texto accesible en cualquier soporte.
- Crear cualquier diseño robusto fácilmente.



Figura 55. Ventajas de un Frameworks de CSS.

Fuente: Yuste. A, (2013)

Clasificación de frameworks de CSS

Para Gutiérrez (2021), existe una clasificación específica de frameworks en CSS, pues esta tiende a tener un uso configurable para construir una aplicación concreta, la clasificación de frameworks de CSS que existen son:

- Multipropósito: Proporcionan un conjunto de utilidades variadas útiles para distintos aspectos del desarrollo de un diseño web, por ejemplo, Cascade Frameworks.
- Propósito específico: Están orientados a proporcionar herramientas para un propósito determinado. Por ejemplo, hay frameworks de CSS orientados a proporcionar herramientas para el desarrollo de rejillas (Ej. Responsive Grid System, Skeleton, Semantic Grid System), orientadas al control de la parte gráfica de la interfaz de usuario (Ej. Accordion, tabs, slideshow), orientados a la tipografía, etc.

Algunos tipos de Frameworks en el lenguaje de programación CSS:

Tailwind CSS

Este Frameworks es muy personalizable, te permite crear una interfaz de usuario única, permite jugar con los colores, tamaños de borde, tamaño de la fuente, paddings, márgenes, sombras y otros elementos y propiedades.



Licenciatura en Informática Administrativa 2022A

Tailwind CSS se centra en el nombre de clases, es difícil dominarla si no tiene nociones de CSS y de la herramienta Atomic CSS.



Figura 56. Interfaz gráfica de Tailwind CSS.

Fuente: Gutiérrez et al, (2021)

Bootstrap

Este Frameworks es uno de los más populares en la actualidad, se encuentra en su versión 4.4.1 y cuenta con características muy importantes como esquemas de color, modificadores, clases útiles para usar formularios, cards, listas, tablas, videos, carrusel, ventanas modales, tooltips, etc. Está construida sobre SASS y es compatible con LESS y SASS.



Figura 57. Interfaz gráfica de Bootstrap CSS.

Fuente: Gutiérrez et al, (2021)



Licenciatura en Informática Administrativa 2022A

Foundation

Este Frameworks junto con Bootstrap son uno de los más utilizados en la actualidad, Foundation es más sofisticado, asimismo es muy flexible y fácil de personalizar. Este Frameworks CSS te permite crear aplicaciones y páginas web adaptables a los dispositivos, empresas como Mozilla, HP, Cisco, eBay, Facebook, Adobe, Disney, entre otras.

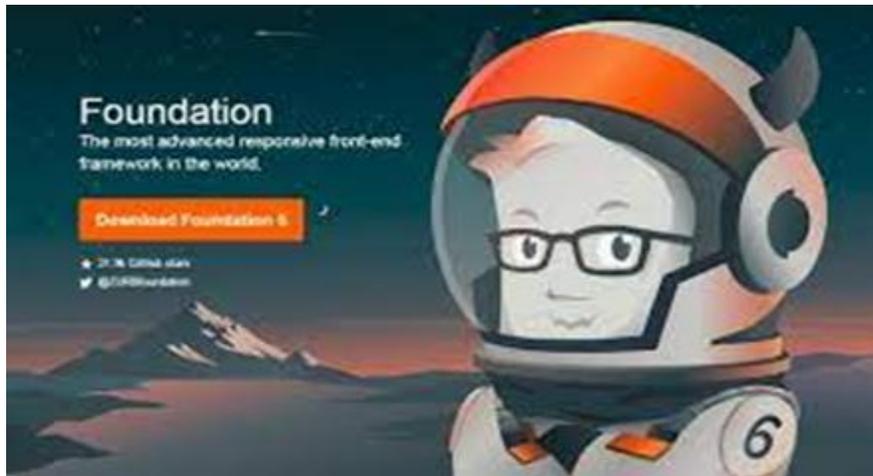


Figura 58. Interfaz gráfica de Foundation CSS.

Fuente: Gutiérrez et al, (2021)

Pure CSS

Este Frameworks es adaptable a los dispositivos y fue creado por la empresa Yahoo por el año 2014, fue construido sobre la herramienta Normalize CSS. A diferencia de Bootstrap, no permite crear diseños fijos y es muy ligero, pesa alrededor de 3.5 Kb solamente cuando se comprime y se minimiza. Este Frameworks cuenta con menús verticales, horizontales y desplegados, asimismo con formularios, botones, tablas, etc.



Licenciatura en Informática Administrativa 2022A



Figura 59. Interfaz gráfica de Pure CSS.

Fuente: Gutiérrez et al, (2021)

Bulma

Este Frameworks CSS es open source y esta creado para trabajar con el diseño de cajas flexibles o Flexbox. Bulma es muy ligero, liviano y con soporte para móviles cuenta con más de 200 mil usuarios hasta la actualidad y se encuentra entre los Frameworks más populares del área, junto con Bootstrap y Foundation.



Figura 60. Interfaz gráfica de Bulma CSS.

Fuente: Gutiérrez et al, (2021)



Licenciatura en Informática Administrativa 2022A

FRAMEWORKS DE JAVASCRIPT

Los frameworks de JavaScript son una parte esencial del desarrollo web front-end moderno, los cuales proveen a los desarrolladores herramientas probadas y testeadas para la creación de aplicaciones web interactivas y escalables. Muchas empresas modernas utilizan frameworks como parte estándar de sus herramientas, por lo que muchos trabajos de desarrollo front-end en la actualidad requieren experiencia en frameworks.

Como aspirante a desarrollador front-end, puede resultar difícil saber por dónde empezar cuando se trata de aprender sobre frameworks; hay muchos frameworks diferentes para elegir, nuevas opciones surgen todo el tiempo. En su mayoría funcionan de forma similar, pero hacen algunas cosas de manera diferente, y hay algunas cosas específicas con las que se debe tener cuidado a la hora de usar frameworks.

Prerequisitos

Antes de intentar aprender sobre los frameworks del lado del cliente, es recomendable que conozcamos los conceptos básicos de los principales lenguajes de la web: HTML, CSS, y especialmente JavaScript.

Como resultado, el código será más consistente y profesional, por lo que podrás solucionar problemas con mayor confianza si comprende las características fundamentales de la plataforma web sobre las que se basan los frameworks. (Developer Mozilla, 2021)

JavaScript es un lenguaje robusto que no solo se utiliza para crear aplicaciones web, también de escritorio y móviles con características y funcionalidades enriquecidas. Con él, podemos construir un sitio web que sea más fácil de usar y altamente interactivo. Se pueden implementar mejoras en la interfaz de usuario que incluyen animaciones, mapas interactivos, gráficos y mucho más mediante JavaScript.

Debido a su popularidad muchas empresas comenzaron a demandar desarrolladores expertos que pudieran adoptar y aprovechar cada una de sus características para crear aplicaciones de todo tipo para sus clientes. Y así se gestó una pequeña gran revolución ya ya dura más de diez años en donde empresas, startups y PYMES han podido ver cómo su negocio no ha parado de crecer a cada nueva versión.

En este escenario, como desarrollador, mantenerse actualizado es clave. Saber usar las mejores herramientas JavaScript se ha convertido en un requisito indispensable y conocer qué librerías o marcos de desarrollo tienen más potencial en 2020 nos permite tomar una decisión rentable, evitando dedicar tiempo a un framework JS que ya no es popular y, finalmente, elegir una mejor alternativa según las necesidades de nuestra aplicación.

En el ámbito del desarrollo del software, el término “JavaScript Framework” significa una biblioteca que proporciona a los desarrolladores plantillas preconstruidas y código JavaScript prescrito para tareas de programación estándar. Millones de desarrolladores usan este tipo de frameworks para acelerar el flujo de trabajo de desarrollo y aplicar las mejores prácticas de una manera fluida y fácil.



Licenciatura en Informática Administrativa 2022A

Debido a su gran variedad su uso depende de los objetivos principales, la funcionalidad general de la plataforma, los requisitos del proyecto y cómo se puede implementar dentro de cada escenario específico.

Al usar este tipo de marcos desarrollo para JavaScript, se puede ahorrar una gran cantidad de tiempo y esfuerzo en el desarrollo de sitios web y aplicaciones basados en este lenguaje. Simplifica todo el procedimiento y permite a los desarrolladores crear aplicaciones web a gran escala de manera eficiente.

Framework más populares de JavaScript

Angular.Js

Creado y mantenido por Google, Angular.Js es uno de los marcos de JavaScript más potentes y conocidos que ofrece a los desarrolladores las mejores respuestas para consolidar JavaScript con HTML y CSS. Es un framework pensado para trabajar en el front-end y de código abierto que permite a los diseñadores de programación ensamblar aplicaciones web intuitivas de una sola página (SPA) como ninguna otra.

React.Js

Creado por Facebook, ReactJS es una biblioteca de JavaScript de código abierto muy efectiva utilizada para crear interfaces de usuario intuitivas. Los administradores del sitio web utilizan este marco para ocuparse de la capa de visualización de aplicaciones y sitios complejos. React.js es más rápido que Angular.js - Sí, lo es - y es una decisión ideal para las personas que necesitan un marco básico, versátil y rápido.

ReactJS es fundamentalmente una biblioteca de estructura de interfaz de usuario, sin embargo, también es una combinación sólida para crear aplicaciones receptivas de una sola página y aplicaciones de etapas cruzadas. Destaca en diferentes aspectos, por ejemplo, poco tamaño de registro, adaptabilidad, similitud con numerosas bibliotecas y una curva de aprendizaje moderada.

Vue.Js

Es uno de los frameworks JavaScript más populares. Fue creado por Evan You en 2014 y cada vez cuenta con más seguidores. La principal razón es que sirve como alternativa a ReactJS o Angular, es sumamente liviana y versátil al estar basada en código abierto se utiliza para crear y mejorar UIs inventivas.

En realidad, es una alternativa a tener muy en cuenta pues se espera que en algún momento supere a sus competidores en el mercado.

Meteor.Js

Es un marco de JavaScript fullstack creado por Meteor Development Group. Toma como base Node.js y permite la creación rápida de prototipos.



Licenciatura en Informática Administrativa 2022A

La innovación clave de Meteor es que mientras las demás aplicaciones viven en el lado servidor o en el cliente Meteor permite operar ambos lados de la aplicación.

Mithril.Js

Es un framework JavaScript avanzado desarrollado para trabajar principalmente en el lado del cliente que generalmente se usa para crear aplicaciones de página única (SPA). Además, ofrece un tiempo de carga de la biblioteca y ejecución más rápidos que otros framework.

Mithril.js permite además un aprendizaje fácil y sencillo evitando en lo posible aportar funcionalidades duplicadas ya presentes en JavaScript que solo agregarían sobrecarga en nuestro aprendizaje. Esto permite centrarnos en lo importante y no perdernos en nuevos conceptos.

Otra de sus ventajas es que se integra bien con código procedente de otros framework y cuenta con un ecosistema extremadamente rico de bibliotecas JavaScript para ayudar con todo tipo de proyectos.

Ember.Js

Un framework JavaScript front-end de código abierto que permite a desarrolladores de software crear aplicaciones versátiles, además tiene la particularidad de no seguir la regla del diseño MVVM (Model-View-ViewModel) y destacar por ser un framework muy flexible.

En contrapartida Ember tiene un diseño moderadamente complejo, aunque los desarrolladores más avanzados encontrarán similitud con otros framework como Angular.js.

Node.js

Un entorno de tiempo de ejecución de JavaScript para trabajar del lado servidor, que tiene la particularidad de funcionar en plataformas cruzadas y es de código abierto. El marco es capaz de manejar E/S asíncronas con su arquitectura basada en eventos.

Funciona en el entorno de tiempo de ejecución de JavaScript y, por lo tanto, muestra propiedades similares de JAVA como subprocesos, paquetes, formación de bucles.

Polymer.Js

Biblioteca JavaScript de código abierto utilizada en múltiples sitios. Google lo utiliza de manera intensiva y está presente en YouTube Gaming, Google Play Music, Google Earth, etc.

Su principal punto fuerte reside en que posibilita la creación de aplicaciones web entre programas con segmentos web reutilizables. En otras palabras, Polymer.js en la actualidad la biblioteca principal de JS para tener en cuenta para la creación de aplicaciones intuitivas.



Licenciatura en Informática Administrativa 2022A

Aurelia.js

Marco de JavaScript de código abierto destinado al desarrollo de aplicaciones móviles, de escritorio y web para diversos fines, incluido el data binding. Aurelia nace con la idea de ser un framework que solucione problemas habituales que se encuentran los desarrolladores y, de esta forma, convertir el desarrollo de aplicaciones en algo sencillo.

A través de una gran variedad de módulos JS permite a los desarrolladores crear aplicaciones rápidamente tanto si estamos utilizando JavaScript vainilla o TypeScript.

Backbone.Js

Es uno de los marcos JavaScript más populares. Diferentes marcas como Hulu, USA Today y Disqus utilizan este framework de manera intensiva debido a que es sumamente ligero y funcional. Es fácil de entender y aprender. Proporciona un sistema MVP y se puede usar para crear todo tipo de aplicaciones. El desarrollo de este framework implica la idea de que todas las funciones del lado del servidor deben fluir a través de una API, lo que ayuda a lograr funcionalidades complejas escribiendo menos código.

Deno.Js

Creado por el mismo desarrollador que NodeJs (Ryan Dahl), comparte con NodeJs algunas de sus principales características como la programación asíncrona y el 'event driven'. Al igual que NodeJs, Deno es multiplataforma y está basada en el motor V8 de Chrome, pero a diferencia de NodeJS, Deno está construido sobre Rust, un lenguaje de desarrollo conocido por su robustez y seguridad, además de su eficiencia. (Equipo Geek, 2020).

2.5 Alternativas de desarrollo on-line.

Para Leover A.G (2011) las alternativas de desarrollo son herramientas que permiten crear y mantener una web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores de las webs. Es decir, es un sistema de gestión de contenidos el cual sirve para que la gestión de un sitio web, por pequeño que sea, no se descontrola: permite tener una apariencia y navegación uniforme en todo el sitio, y actualizar y gestionar el contenido fácilmente.

Gustavo B. (2022) menciona en esta definición se incluyen aplicaciones tan simples como una WIKI hasta tan complejas como un gestor de contenidos avanzado como Drupal, PostNuke, etc o un LMS (Learning Management System) para la implementación de plataformas de aprendizaje mediante la gestión e implementación de cursos on line. Moodle es un ejemplo de esto.

Para Trabajar o realizar cualquier aplicación en la Web existen muchas opciones como son los Frameworks, las librerías de desarrollo (generalmente escritas en java script) y los Sistemas Gestores de Contenidos, a continuación, explicaremos que es cada uno de ellos según Leover A.G (2011).



Licenciatura en Informática Administrativa 2022A

Frameworks: La palabra inglesa "framework" define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Librería: En todos los lenguajes de programación existen librerías de funciones que sirven para hacer cosas diversas y muy repetitivas a la hora de programar.

Las librerías de los lenguajes de programación ahorran la tarea de escribir las funciones comunes que por lo general pueden necesitar los programadores y estas librerías ya traen muchas funciones dentro de sí. Un lenguaje de programación bien desarrollado tendrá una buena cantidad de ellas. En ocasiones es más complicado conocer bien todas las librerías que aprender a programar en el lenguaje.

Sistemas Gestores de Contenidos (CMS) Un Sistema de gestión de contenidos (CMS) permite la creación y administración de contenidos principalmente en páginas Web. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño.

Porque usar un CMS

- a) Creación de Contenido.
- b) Gestión de Contenido.
- c) Administración.
- d) Publicación.
- e) Presentación.
- f) Usabilidad.
- g) Aumento de la productividad del trabajo.
- h) Poner el producto final en manos no expertas.
- i) Eliminar los FTP.
- j) Código estándar a lo largo de la aplicación.



Licenciatura en Informática Administrativa 2022A

k) Disminución de costo y esfuerzo.

Necesidad de un CMS

a) Inclusión de nuevas funcionalidades en la web.

b) Mantenimiento de gran cantidad de páginas.

c) Reutilización de objetos o componentes.

d) Páginas interactivas.

e) Cambios del aspecto de la web.

f) Control de acceso.

Características principales que nos aportan los CMS

ü Independencia del contenido respecto a la presentación.

} Facilita futuros cambios de diseño en la página Web.

Posibilita publicar un mismo contenido a través de varios canales a la vez.

Organización de la información en tipos de contenido.

Ejemplos de Frameworks.

Gustavo B.(2022) propone ejemplos entre los cuales se encuentran:

- **Symfony** es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.
- **CodeIgniter** es un entorno de desarrollo abierto que permite crear webs dinámicas con PHP. Su principal objetivo es ayudar a que los desarrolladores, puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, además su principal característica es que es muy rápido y ligero. Facilita la escritura de código repetitivo, y a comparación de otros Frameworks cómo CakePHP, Symphony o Zend Framework, Codeigniter es más rápido, pero menos fácil ya que carace de algunas librerías que los otros frameworks tienen, pero aun así no deja de ser un



Licenciatura en Informática Administrativa 2022A

buen framework además de que es totalmente extensible y altamente compatible con gran variedad de versiones y configuraciones de PHP.

- **Django** es un framework de desarrollo web de código abierto, escrito en Python, que cumple en cierta medida el paradigma del Modelo Vista Controlador. Su meta fundamental es facilitar la creación de sitios web complejos, pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, del desarrollo rápido y del principio de DRY (del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.
- **jQuery** es una biblioteca o framework de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

Ejemplos de CMS:

- **Joomla!** es un sistema de gestión de contenidos, y entre sus principales virtudes está la de permitir editar el contenido de un sitio web de manera sencilla. Es una aplicación de código abierto programada mayoritariamente en PHP bajo una licencia GPL. En Joomla! se incluyen características como: mejorar el rendimiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, polls (encuestas), calendarios, búsqueda en el sitio web e internacionalización del lenguaje.
- **WordPress** es un sistema de gestión de contenido enfocado a la creación de blogs (sitios web periódicamente actualizados). Desarrollado en PHP y MySQL, bajo licencia GPL y código modificable, En principio, está configurado para usar una bitácora o weblog por sitio o instalación, pero también es posible, sin "hacks" o añadidos, tener varios blogs -varias instalaciones en realidad- con varias o una única base de datos.
- **TYPO3** es una herramienta de gestión de contenidos abierta y flexible que permite diseñar espacios web a medida de las necesidades y personalidad de cada empresa. Esta tecnología abre un nuevo concepto de website funcional estructurada sobre tres sólidos pilares: Máxima facilidad y rapidez en la gestión y actualización de contenidos, Integración de diferentes idiomas en un mismo proyecto y multiplicidad de usuarios, páginas y recursos. Se integra con varias bases de datos como MySQL, PostgreSQL y Oracle, Typo3 se puede ampliar usando el plugin (Extensiones de llamada). Estas extensiones permiten utilizar muchas aplicaciones (por ejemplo: Noticias, comercio electrónico, foros, etc) sin tener que programar

Todas estas tecnologías mencionadas son una alternativa, el éxito o fracaso final depende de la persona, no de la herramienta.



Licenciatura en Informática Administrativa 2022A

REFERENCIAS

- Azaustre, Carlos. (2016). Aprendiendo JavaScript, Desde cero hasta ECMAScript 6+. Edición Kindle. Carlosazaustre. <https://bit.ly/36GJ79B>
- Cordovall , E. (2014). Manuales de CSS. <https://bit.ly/3NinZat>
- DuvalL,R.(2015). <https://bit.ly/3NdCegK>
- González, L.(2011). Alternativas para el desarrollo de Aplicaciones Web. Informática y las Comunicaciones. <https://bit.ly/3tEErtK>
- Gutiérrez, A. (2021). Los 5 mejores Framesworks CSS. ¿Qué es un Frameworks?. <https://bit.ly/36K9Odn>
- IDEVIK, X. (2021). ¿Qué es una etiqueta HTML y cuáles son las más importantes? Barcelona, IEBSCHOOL. <https://bit.ly/3iA4hZD>
- Gonzalez, R. (2017). Ejemplos de CSS3. <https://bit.ly/3NqI7d8>
- Roma, J. (s.f). Estructura de etiqueta HTML- HTML en español. Lenguaje HTML. <https://bit.ly/3iDcuw4>
- Sapetti, C. (n.d.). Programación funcional con Javascript (I). 26, Septiembre de, 2022. <https://bit.ly/3uk4TYY>
- Yuste Torregrosa, Á., & Luján-Mora, S. (2013). iDESWEB. Frameworks CSS. Tipos de Framesworks en la actualidad. <https://bit.ly/3lwLjxz>



Licenciatura en Informática Administrativa 2022A

UNIDAD 3. TECNOLOGÍAS DE BACK-END

OBJETIVO:

Identificar las tecnologías actuales de back-end en sus versiones más recientes para diseñar programas de cómputo ejecutándose en el servidor.

3.1 Lenguajes y entornos de desarrollo integrados (IDE)

Según Hernández (2022) Un entorno de desarrollo integrado (IDE, por siglas en inglés de Integrated Development Environment) no es más que un programa que se encuentra compuesto por un conjunto de herramientas que se usan por un programador.

Los IDE se crearon para facilitar un programa de forma única en el cual se puede acometer todo el desarrollo de un software o sistema, y así aumentar la productividad de los desarrolladores, proporcionando componentes necesarios para el desarrollo.

Los IDES pueden funcionar como aplicaciones por sí solas también ser parte de aplicaciones existentes, en muchos casos se dedican a un solo lenguaje de programación y otros múltiples lenguajes que estos mismos permiten opciones como editar, compilar, depurar y desarrollar interfaces gráficas de usuario. En los lenguajes de programación clásicos por ejemplo C++, Object Pascal, Velneo, entre otros, además dota de un marco de trabajo muy accesible.

Su objetivo principal de las IDES es permitir simplificar la configuración de múltiples componentes de desarrollo, además suministra una misma estructura de servicios, los cuales reduce los tiempos de ajustes, permitiendo aumentar la productividad de desarrollo.

Características de una IDE

Un IDE es una aplicación de software que une múltiples herramientas y servicios en un solo programa.

1. Son multiplataforma.
2. Son capaces de sustentar múltiples lenguajes de programación.
3. Permiten la integración con sistemas de control de versiones.
4. Identificación del modelo de sintaxis en la cual se desarrolla.
5. Cuentan con extensiones y componentes para el IDE.
6. Cuentan con la integración con frameworks conocidos.
7. Permiten crear, modificar, compilar, implementar y depurar un software.
8. Tienen capacidad de importar y exportar proyectos.
9. Trabajan en múltiples idiomas.
10. Tiene manual de usuarios y ayuda.



Licenciatura en Informática Administrativa 2022A

Ventajas de un Entorno de Desarrollo Integrado (IDE)

- El uso de un IDE permite desarrollar nuevos programas y aplicaciones que permiten al desarrollador realizar sus tareas.
- Tienen una curva de aprendizaje baja.
- Posibilita crear tareas básicas en el desarrollo (guardar los datos, compilar-enlazar ejecutar los datos) los cuales se encuentran en la barra de herramientas del desarrollo.
- El código fuente que se muestra está compuesto con el formato del lenguaje de programación correspondiente.
- Permiten al desarrollador seleccionar opciones de auto completado de código.
- Proporcionan mensajes de error de sintaxis, durante el desarrollo o la compilación del programa.
- Facilitan lista de métodos sobre cómo realizar las correcciones de errores durante la compilación.
- Explican método para desarrollar refactorización de archivos y códigos fuente.

¿Componentes de un Entorno de Desarrollo Integrado (IDE)?

- Editor de texto
- Compilador
- Intérprete
- Herramienta para la automatización
- Depurador

¿Cómo usan los programadores de aplicaciones web los IDE?

Si un programador desarrolla solamente web estático (HTML, CSS y a lo mejor un poco de JavaScript), quizás se encuentre pensando «¡No necesito ningún IDE!», y estarían en lo correcto porque un IDE es desmesurado para aquellos que solo desarrollan sitios web estáticos. Pero si lo que quiere es desarrollar aplicaciones web, o convertir sus aplicaciones en aplicaciones móviles, quizás debe utilizar un IDE.

¿Cómo hallar un excelente IDE?

Debido que está desarrollando páginas web, lo primero es averiguar si el IDE que se está pensando en utilizar es compatible con HTML, CSS y JavaScript.

Si se encuentra deseando desarrollar una aplicación web, necesitará algo de HTML y CSS. Es posible que pueda arreglárselas sin JavaScript, pero eso es poco probable.

Entonces debería considerar en el lenguaje para el que necesita el IDE, este podría ser: Java, C / C ++ / C #, Perl, entre otros. El IDE debería poder compilar o interpretar el idioma que desea utilizar, así como depurarlo.

Licenciatura en Informática Administrativa 2022A

¿En realidad los desarrolladores de aplicaciones web necesitan utilizar un IDE?

Los desarrolladores de aplicaciones web no necesitan utilizar un IDE. Para el mayor número de los diseñadores, un IDE sumará más complejidad sin adicionar mucho valor. Puesto que la mayoría de las páginas web e inclusive la mayor cantidad de aplicaciones web se desarrollan usando lenguajes de programación que no es necesario compilar.

Entonces, un compilador también es innecesario. Excepto que el IDE pueda depurar JavaScript, el depurador tampoco es de mucha utilidad. Las herramientas de automatización de compilación son basadas en el depurador y el compilador, por lo que no adicionan mucho valor. Entonces, la solución de la mayoría de los diseñadores web utilizan en un IDE es el editor de código fuente, con el fin de escribir HTML. Y en la mayoría de los casos, hay editores de texto que ofrecen más funciones y son útiles.

Entornos de Desarrollo Integrado (IDE) para JavaScript, HTML y CSS

Visual Studio Code

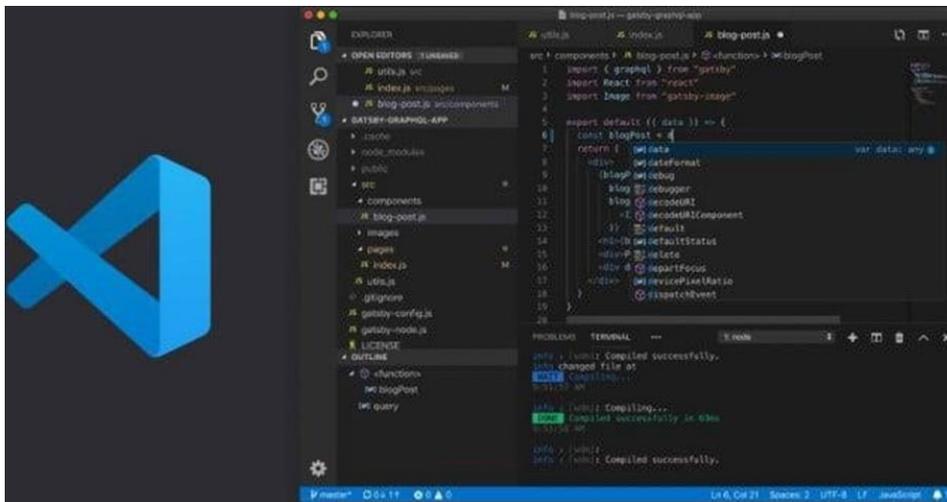


Figura 61. Visual Studio

Fuente: Hernández (2022)

Es un editor de código fuente ligero pero muy potente, y se encuentra disponible para los principales sistemas operativos tales como Windows, Mac y Linux. Contiene un excelente soporte incorporado para JavaScript, TypeScript y Node.js, y tiene un rico ecosistema de extensiones para otros lenguajes como C ++, C #, Python, PHP, y tiempos de ejecución.

Además, este editor que proporciona un resaltado de sintaxis y el autocompletado con IntelliSense, proporcionando terminaciones en tipos de variables, definiciones de funciones y módulos importados. Además, permite depurar código directamente desde el editor.

Licenciatura en Informática Administrativa 2022A

Atom by Github



Figura 62. App Atom

Fuente: Hernández (2022)

Es un editor de texto que es moderno, accesible, pero modificable hasta el núcleo, permitiendo personalizarlo para cualquier cosa. Este editor de texto contiene un administrador de paquetes incorporado, permitiendo buscar e instalar nuevos paquetes o comenzar a crear los tuyos desde Atom.

Atom viene preinstalado con cuatro interfaces de usuario y ocho temas de sintaxis en colores claros y oscuros. Si no puede encontrar lo que está buscando, también puede instalar temas creados por la comunidad Atom o crear los tuyos tu mismo.

Brackets

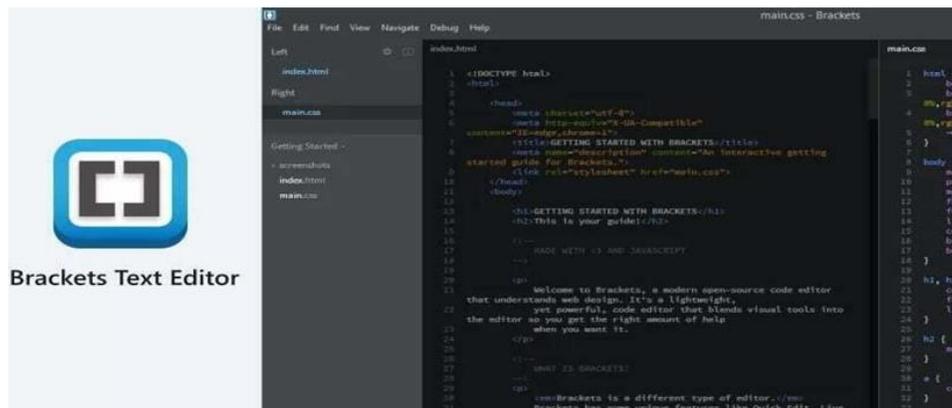


Figura 63. Brackets

Fuente: Hernández (2022)

Es un editor de texto de código abierto muy actual que entiende el diseño web. Contiene herramientas visuales y soporte de preprocesador, Brackets es un moderno editor de texto que facilita el diseño en el navegador. Se encuentra diseñado desde cero para diseñadores web y desarrolladores de aplicaciones para usuario. Brackets es un proyecto de código abierto, apoyado por una comunidad activa y apasionada.

Licenciatura en Informática Administrativa 2022A

WebStorm es un excelente Entorno de Desarrollo Integrado para el desarrollo de aplicaciones web

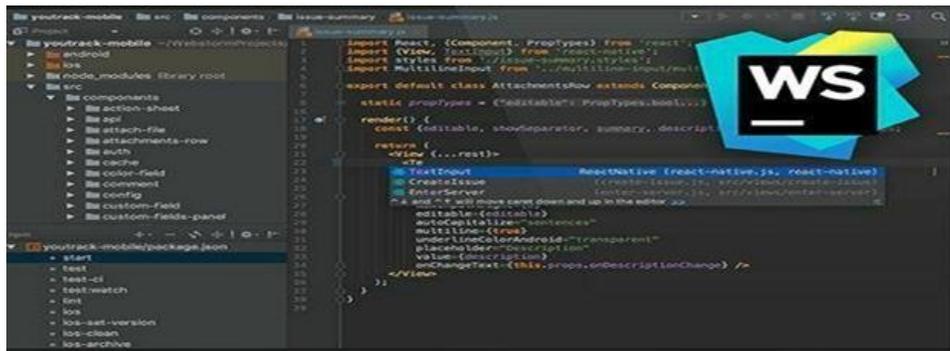


Figura 64. WebStorm

Fuente: Hernández (2022)

WebStorm tiene un excelente IDE, ayudando al desarrollador a escribir código mejor gracias a la finalización de código inteligente, sobre el desarrollo permite la detección de errores, navegación de gran alcance y refactorización. El IDE proporciona soporte para JavaScript, Node.js, HTML y CSS, así como sus sucesores modernos. Marcos compatibles incluyen Angular, React, Meteor y más.

Netbeans

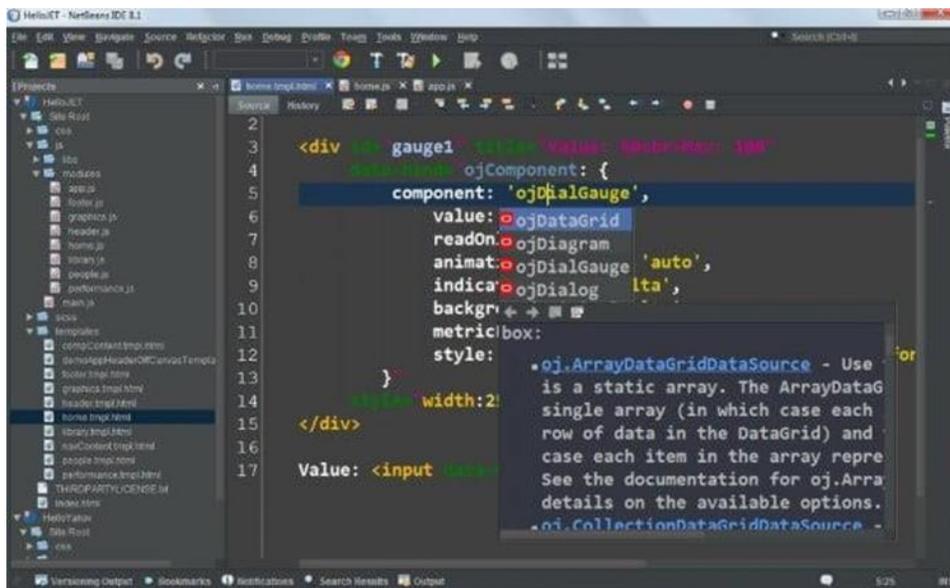


Figura 65. Netbeans

Fuente: Hernández (2022)



Licenciatura en Informática Administrativa 2022A

Netbeans es un IDE que permite desarrollar aplicaciones de escritorio, móviles y web de manera rápida y sencilla con Java, JavaScript, HTML5, PHP, C / C ++, entre otros. Además, es gratuito, de código abierto y tiene una gran comunidad de desarrolladores y usuarios.

3.2 Controladores de versiones, administración y distribución de código

Control de versiones

Se llama control de versiones a los métodos y herramientas disponibles para controlar todo lo referente a los cambios en el tiempo de un archivo.

Difícilmente un archivo de código o un documento de texto está terminado con la primera escritura; necesita cambios o reescrituras para corregir errores, modificar su contenido... A medida que el documento cambia existen dos opciones, mantener un historial de cambios o dejar que evolucione sin memoria. El control de versiones es un método estándar para mantener esta memoria haciendo además que sea útil para el desarrollo futuro.

En documentos sencillos como un ensayo o un pequeño programa la memoria no es algo esencial, pero en la escritura de un libro o un programa con centenares de páginas y una docena de manos involucradas no hay otra manera de trabajar. Esta es precisamente la palabra clave, mantener un control de las versiones de todos los archivos de un proyecto es una manera de trabajar completamente estandarizada; todas las prácticas tienen un nombre.

La buena noticia es que todo este formalismo es generosamente recompensado por el uso de sistemas de control de versiones automáticos como CVS, Subversión, git..

¿Cómo se especifican las versiones?

El método más habitual de asignar una versión a un documento es mediante un número o un grupo de números. No existe un modo fijo de numerar una versión, se deja al criterio de cada desarrollador. Sí existen ciertas prácticas habituales en la numeración como el uso de tres cifras o la numeración decimal.

Versiones X.Y.Z: Un método bastante habitual de numerar las versiones es utilizando dos o tres cifras decimales para indicar la importancia de los cambios realizados. El cambio de la primera cifra indica cambios más importantes que el de la segunda. El criterio más habitual es seguir las siguientes normas:

- **La primera cifra (X) indica la versión mayor del documento.** Si empieza con un cero significa que el documento aún no está listo o no cumple con los requerimientos mínimos. Cada cambio en esta cifra denota una reescritura o la incompatibilidad con versiones mayores anteriores.
- **La segunda cifra (Y) indica la versión menor del documento.** Denota cambios en el contenido o en la funcionalidad del documento, pero no lo suficientemente importantes



Licenciatura en Informática Administrativa 2022A

como para decir que ya no es el mismo. Cuando se estrena una versión mayor se deja la versión menor a cero, pero aun así se incluye de modo que la segunda versión mayor sería la 2.0

- **La tercera cifra (Z) indica la segunda versión menor.** Indica que el documento se ha corregido pero que no se ha añadido ni eliminado nada relevante. Cuando se estrena una versión menor, es decir, cuando la segunda versión menor es igual a cero; suele omitirse.

- Pueden añadirse recursivamente versiones menores, algunos proyectos hacen uso de ellas cuando los criterios de selección de versiones no corresponden con los definidos en los puntos anteriores. El núcleo del sistema operativo Linux utiliza una numeración de cuatro cifras.

- Otra práctica común es definir versiones especiales como las versiones Alpha, beta o las reléase candidate. Estas versiones suelen utilizarse antes de llegar a un hito como una nueva versión menor. Por ejemplo, antes de llegar a la segunda versión mayor, la versión 2.0, puede publicarse una primera versión previa Alpha numerada como 2.0a o 1.99. La segunda versión previa se llamaría beta y se numeraría como 2.0b o (por ejemplo) 1.999. Si aun así son necesarias más versiones previas, en vez de optar por más letras griegas se numeran versiones candidatas; la primera sería la release candidate1 numerada como 2.0-rc1.

Versiones numeradas: Un método de numeración compatible con el anterior es numerar las versiones mediante un número decimal. La primera versión es la 1, la segundo es la 2 y así sucesivamente. Esta numeración suele utilizarse en el desarrollo cuando el control de versiones se lleva de forma automática mediante algún programa de control de versiones. En algunos casos ambas numeraciones se combinan. Por ejemplo, se está desarrollando la versión 2.1 a partir de la 2.0.5 y para el software de control de versiones (se toma como ejemplo Subversión) la última versión es la 671. Entonces la versión de desarrollo podría ser la 2.0.5-svn671.

Repositorios y copias de trabajo

Todas las herramientas de control de versiones se basan en la típica comunicación servidor-cliente. Desde el punto de vista de los documentos se podría decir que el repositorio es el código está en el servidor y la copia de trabajo en el cliente.

La copia de trabajo es entonces una imagen del contenido en el repositorio de código y es lo que se utiliza para el trabajo diario (modificar archivos existentes crear documentos nuevos...). Todos los cambios que se hagan en dicha imagen no son definitivos hasta que se suban al repositorio. El control de versiones se ve influido en gran parte por cómo se realiza esta comunicación entre copia de trabajo-repositorio.



Licenciatura en Informática Administrativa 2022A

El lector puede plantearse la necesidad de duplicar los datos del proyecto. Esta arquitectura tiene la ventaja de que pueden existir tantas copias de trabajo como sean necesarias, normalmente tantas como participantes tenga el proyecto. Cada participante tiene su copia de trabajo y es el servidor el encargado de que los cambios aportados por los desarrolladores no entren en conflicto.

¿Cómo se organizan los documentos dentro de la copia de trabajo?

Los documentos sujetos al control de versiones deben estar organizados de un modo determinado. No sirve utilizar un directorio distinto para cada versión porque se supone que el control de versiones es automático.

La estructura de documentos suele dividirse en tres grandes directorios llamados trunk, branches y tags. (Borrell, G., 2006).

Controladores de administración

Según David-Engel (2021) este es una biblioteca que administra la comunicación entre las aplicaciones y los controladores. Por ejemplo, en las plataformas Microsoft® Windows®, el Administrador de controladores es una biblioteca de vínculos dinámicos (DLL) escrita por Microsoft y que pueden redistribuir los usuarios del SDK redistribuible de MDAC 2.8 SP1.

Por tanto, existen principalmente para los escritores de aplicaciones y resuelven una serie de problemas comunes a todas las aplicaciones. Esto incluye determinar qué controlador cargar en función de un nombre de origen de datos, cargar y descargar controladores y llamar a funciones en los controladores.

Como menciona David-Engel (2021) la función principal de los controladores de administración es resolver el problema proporcionando un único lugar para llamar a cada función. La aplicación está vinculada al mismo y llama a funciones ODBC, no al controlador. La aplicación identifica el controlador de destino y el origen de datos con un identificador de conexión. Cuando carga un controlador, el Administrador de controladores genera una tabla de punteros a las funciones de ese controlador. Usa el identificador de conexión pasado por la aplicación para buscar la dirección de la función en el controlador de destino y llama a esa función por dirección.

El rol principal final de los controladores de administración es cargar y descargar controladores. Cuando desea usar un controlador determinado, llama a una función de conexión (SQLConnect, SQLDriverConnect o SQLBrowseConnect) en el Administrador de controladores y especifica el nombre de un origen de datos o controlador determinado, como "Accounting" o "SQL Server". Con este nombre se busca en la información del origen de datos el nombre de archivo del controlador, como Sqlsrvr.dll. A continuación, carga el controlador (suponiendo que no esté ya cargado), almacena la dirección de cada función



Licenciatura en Informática Administrativa 2022A

en el controlador y llama a la función de conexión en el controlador, que se inicializa a sí mismo y se conecta al origen de datos.

Distribución de Código

Se conoce como código fuente al texto desarrollado en un lenguaje de programación y que debe ser compilado o interpretado para poder ejecutarse en un ordenador, también llamado computadora. El término código tiene diferentes usos y acepciones. Puede tratarse de una combinación de símbolos que, en el marco de un sistema ya establecido, cuente con un cierto valor. En la informática encontramos distintos tipos de códigos. Un código, por otra parte, es una combinación de caracteres que se emplea para crear y entender mensajes.

El término código tiene diferentes usos y acepciones. Puede tratarse de una combinación de símbolos que, en el marco de un sistema ya establecido, cuente con un cierto valor. (Martínez, 2019).

El control de código fuente (o el control de la versión) es la práctica de seguimiento y administración de los cambios en el código. Los sistemas de administración de código fuente (SCM) mantienen un historial del desarrollo de código, y ayudan a resolver conflictos a la hora de combinar las contribuciones de diferentes orígenes.

Git es un sistema de administración de código fuente distribuido de código abierto, Git permite crear una copia de su repositorio, conocida como rama. Con esta rama, puede trabajar en su código con independencia de la versión estable de la base de código. Cuando los cambios estén listos, puede almacenarlos como un conjunto de diferencias, conocido como confirmación. Puede insertar confirmaciones de otros colaboradores en su repositorio, enviar confirmaciones a otras personas y combinar sus confirmaciones de nuevo en la versión principal del repositorio. Para obtener más información sobre Git.

La noción de código fuente también puede tomarse de manera más amplia, para incluir código de máquina y notaciones en lenguajes gráficos, ninguno de los cuales es de naturaleza textual. Un ejemplo de un artículo presentado en la conferencia anual IEEE y sobre análisis y manipulación de código fuente.

En aras de la claridad, se entiende por "código fuente" cualquier descripción completamente ejecutable de un sistema de software. Por lo tanto, se considera que incluye código de máquina, lenguajes de muy alto nivel y representaciones gráficas ejecutables de sistemas.

A menudo, hay varios pasos de traducción o minificación de programas entre el código fuente original escrito por un humano y un programa ejecutable. Mientras que algunos, como la FSF., argumentan que un archivo intermedio "no es un código fuente real y no cuenta como código fuente", otros encuentran conveniente referirse a cada archivo intermedio como el código fuente para los siguientes pasos. (Perales G., 2015).



Licenciatura en Informática Administrativa 2022A

3.3 Instalación y configuración de servidores de aplicaciones y entornos preconfigurados para la ejecución.

Servidor de Aplicaciones: Se trata de un dispositivo que proporciona los servicios de aplicación a las computadoras cliente. (Joyanes, 2008.)

Características:

- Control parcial o total de las funciones lógicas del negocio.
- Control de acceso de datos a la aplicación.
- Centralización y disminución de la complejidad en el desarrollo de las aplicaciones.

Herramientas de desarrollo:

- Java 7.
- Eclipse Kepler.
- Jboss 7.0.1 S.E.
- Servidor (Local o Red).

Configuraciones

- Dependencias jboss.
- Variables de entorno.
- Pool de conexión.
- Entrada y salida de puertos.

Configuración del árbol de desarrollo

Se requiere hacer la estructuración del árbol de desarrollo de la siguiente forma:

C:\java/Jboss

Agregar las variables de entorno.

Instalación de las herramientas

- Instalación de Java 7.
- Instalación de Eclipse Kepler.
- Instalar Jboss 7.0.1.
- Agregar dependencias de Jboss.

Agregar sitios a Jboss

- Asignar el nombre del sitio.
- Path de ubicación del sitio.



Licenciatura en Informática Administrativa 2022A

- Configuración de DNS o IP (Local o Red).
- Arrancar sitio.

Configurar el Pool de conexión

Name: SqlServer_pool

JNDI: java:jboss/datasource/sqljdbc

Driver de conexión.

Path driver: C:\Java\jboss\standalone\deployments

Url de conexión: jdbc:sqlserver://ipServer:puerto;

JNDI: sqlserver

Driver: h2

Usuario: usuarioServer

Password: passwordServer

Reiniciar el servidor

Configurar puertos de conexión

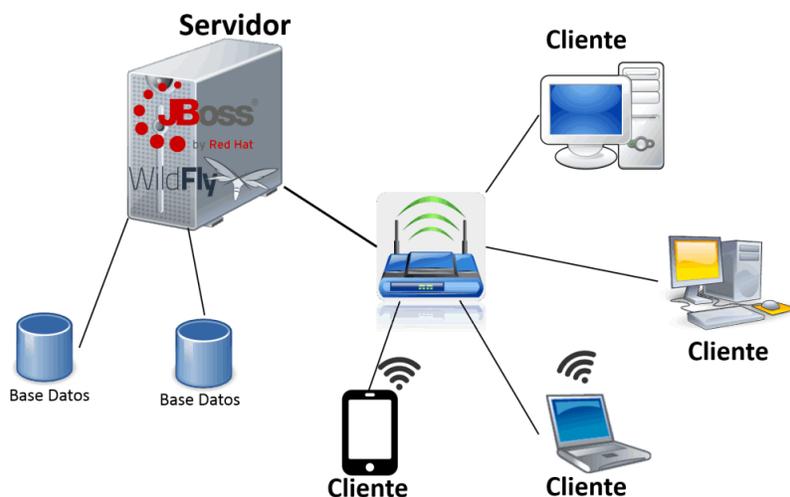


Figura 66. Instalación y Configuración de Servidores de Aplicaciones.

Fuente: Angexim, (2020).



Licenciatura en Informática Administrativa 2022A

Pruebas unitarias del sistema

- Visualizar el servidor de aplicaciones en ejecución
- Verificar el pool de conexiones
- Generar un proyecto web-jsp
- Correr el servidor de aplicaciones con el proyecto
- Visualizar el resultado del servidor

A continuación, se muestra un ejemplo de los pasos de la instalación de un servidor de aplicación llamado Gerónimo:

- Descomprimos el archivo descargado en un directorio (Ej: C:\geronimo-tomcat6-jee5-2.0.2). Al directorio de instalación le llamaremos a partir de ahora: GERONIMO_HOME.
- Editamos el archivo "GERONIMO_HOME/bin/geronimo.bat" y añadimos nuestra JAVA_HOME en la primera línea después de los comentarios.
- Para iniciar el servidor ejecutamos el archivo "GERONIMO_HOME/bin/startup.bat".
-
- Para parar el servidor ejecutamos el archivo "GERONIMO_HOME/bin/shutdown.bat".
- Introducimos el UserName: "system" y el Password: "manager" y vemos como el servidor se para.
- Finalizando de ambas formas debemos ver que la señal de finalización se ha recibido.

3.4 Desarrollo de programas para el back-end

Según Álvarez *et.al.*, (2016) El back-end es el encargado del procesamiento de los datos de entrada desde el front-end, siendo este el controlador de la vista y realizando la conexión con la base de datos. En este apartado se detallarán las tecnologías empleadas para los controladores y la creación de la base de datos.

Lenguajes de programación recomendados para backend

- PHP. Se trata del lenguaje de programación más extendido para desarrollar páginas web. La gran mayoría de CMS parten de la base de PHP y este lenguaje dispone de varios frameworks como Laravel o Symfony para cohesionar el contenido con la base de datos.



Licenciatura en Informática Administrativa 2022A

Ejemplo de PHP.Net manual (2020):

```
<!DOCTYPE html>
<html>
  <head>
<title>Ejemplo</title>
  </head>
  <body>

    <?php
echo "¡Hola, soy un script de PHP!";
    ?>

  </body>
</html>
```

- Java. Java es el lenguaje todoterreno, uno de los más utilizados en muchos ámbitos. La programación backend en Java es más segura que en el caso de PHP, por ejemplo, puesto que cuenta con un sistema de validación que otros lenguajes no poseen. En el caso de Java, se aplican sistemas de seguridad a nivel de servidor de aplicaciones y en frameworks externos. En PHP, todos estos procesos cuentan con menos automatizaciones.

```
public class Mi_Clase_1
{
  public static void main (String args[])
  {
    for (int i = 0; i <= 12; i++)
    {
      System.out.print("12 * "+ i + " = " + 12 * i + "\n");
    }
  }
}
```

Figura 67. Código

Fuente: Álvarez et. al. (2008)

Ejemplo de Programarya, (2010):



Licenciatura en Informática Administrativa 2022A

- Node.js. Además de ser un lenguaje de programación, Node.js es un entorno JavaScript basado en eventos junto con la velocidad con la que ejecuta JavaScript es una de sus mayores bondades. Además, programa de forma asíncrona, cosa que permite asignar variables en cualquier momento de la ejecución. En el caso de la programación síncrona, las variables se ejecutan en orden, primero una y después la siguiente, cosa que ralentiza los procesos.

```
extends layout
block content
  h1= title
  p Welcome to #{title}
  script.
    var f1 = function() { document.getElementById('myImage').src='#{data.item1}' }
  script.
    var f2 = function() { document.getElementById('myImage').src='#{data.item2}' }
  script.
    var f3 = function() { document.getElementById('myImage').src='#{data.item3}' }

  button(onclick='f1()') One!
  button(onclick='f2()') Two!
  button(onclick='f3()') Three!
  p
  a: img(id='myImage' height='300' width='300' src=")
```

Figura 68. Código

Fuente: Álvarez et. al. (2008)

Ejemplo de Microsoft. visualstudio (2022):

- Python. Python es un lenguaje ideal para aprender a programar. Los usuarios suelen recomendar el uso del framework Django en el contexto web. Además, es compatible con una gran variedad de servidores. Al tratarse de un lenguaje tan extendido, los desarrolladores están optimizando continuamente sus librerías y frameworks. Hay que tener en cuenta que su flexibilidad le permite funcionar en programas simples y en operaciones altamente complejas.

```
text = text.split()
words = {}
for i in text:
```



Licenciatura en Informática Administrativa 2022A

```
if i in words:
    words[i] += 1
else:
    words[i] = 1
return words
print(count_words(text))
print(most_repeated(count_words(text)))

def most_repeated(words):
    max_word = ""
    max_freq = 0
    for word, freq in words.items():
        if freq > max_freq:
            max_word = word
            max_freq = freq
    return max_word, max_freq
```

Figura 69. Código

Fuente: Álvarez *et. al.* (2008)

Ejemplo de aprendeconalf.es (2004)

- Ruby. Este lenguaje, combinado con su framework Ruby on Rails conforma un conjunto ganador. Ruby on Rails se basa en Ruby, HTML, CSS y JavaScript para generar aplicaciones que se ejecutan en un servidor web. Los expertos aseguran que es más recomendable aprender Ruby mientras se trabaja en Ruby on Rails.

- BEGIN: representa un bloque que se ejecuta al inicio del programa.
- #END: representa un bloque que se ejecuta al final del programa.
- #__END__: representa el final del código, no se ejecuta nada después de __END__.
- #__FILE__: es el nombre del archivo que se está ejecutando.
- #__LINE__: es el número de líneas que se está ejecutando en el archivo actual.

Figura 70. Código de Ruby

Fuente: Álvarez *et. al.* (2008)

Ejemplo de hostgator (2021)

Bases de datos para backend

- Bases de datos relacionales. Estas bases de datos presentan la información organizada en un conjunto de tablas descritas formalmente. Permite acceder y tratar



Licenciatura en Informática Administrativa 2022A

estos datos de distintos modos sin tener que reorganizar las tablas base. Algunos ejemplos de este tipo de base de datos son SQL Server, MySQL y Oracle.

- Bases de datos no relacionales. Las bases de datos no relacionales o NoSQL guardan los datos en documentos en lugar de en registros. Una de las más populares es MongoDB, que requiere de JavaScript, puesto que la consola desde la que podemos ejecutar sus comandos está construida sobre este lenguaje.

El desarrollador backend debe garantizar la seguridad en todos sus procesos, puesto que interactúa de forma constante con estas bases de datos, sensibles a ciberataques. Todas las herramientas que hemos presentado son solo algunas de las más recomendables, pero existe una infinidad de soluciones al alcance de todo desarrollador. Te animamos a probar con los distintos lenguajes para averiguar cuál es el que más se adapta a tus necesidades y a los requisitos del proyecto.

MongoDB



Figura 71. Logo Mongo DB

Fuente: Álvarez *et. al.* (2008)

MongoDB (de la palabra en inglés 'humongous' que significa enorme) es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto. MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL ('no sólo SQL, sino que es una base de datos con open source').

En lugar de guardar los datos en tablas como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos en documentos similares a archivos JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.



Licenciatura en Informática Administrativa 2022A

Mongo DB fue fundada en 2007 por "DoubleClick, ShopWiki y Gilt Groupe". En DoubleClick había desafíos diarios con el tratamiento, el almacenamiento y con el escalado de los datos, por este motivo, DoubleClick escribió su propio software para solventar dichos problemas.

En 2009 MongoDB fue lanzado como un producto independiente y publicado bajo la licencia de código abierto AGPL (Affero General Public License, es una licencia derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red). En marzo de 2011, se lanzó la versión 1.4 y se consideró ya como una base de datos lista para su uso en producción.

Express

Express.js está basado en Connect, el cual es un framework extensible de manejo de servidores HTTP, provee plugins de alto rendimiento conocidos como middleware. Middleware es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones móviles y web, facilitando un rápido desarrollo basado en nodos.

Características de Express:

- Permite configurar middleware para responder a las peticiones HTTP.
- Define una tabla de enrutamiento que se utiliza para llevar a cabo diferentes acciones, basadas en el método HTTP y URL.
- Permite representar dinámicamente páginas HTML basadas en el envío de argumentos de las plantillas.

NodeJS



Figura 72. Logotipo de NodeJs

Fuente: Álvarez *et. al.* (2008)



Licenciatura en Informática Administrativa 2022A

Node.js es una librería y entorno de ejecución de E/S (Entrada / Salida) dirigida por eventos y por lo tanto asíncrona que se ejecuta sobre el intérprete de JavaScript creado por Google V8. Trabaja con un único hilo de ejecución que es el encargado de organizar todo el flujo de trabajo que se deba realizar. Para poder trabajar de una forma óptima Node.js delega todo el trabajo en un pool de threads (hilos). Este pool de threads está construido con la librería libuv. Esta librería dispone de su propio entorno multithread (multitarea) asíncrono.

Node.js envía el trabajo que hay que realizar al pool. Ilustración 10. Logotipo NodeJS. Jorge Nieto Rodrigo 18/47 Cuando se trabaja con Node.js prácticamente toda la programación es asíncrona y se parece tanto a las clásicas llamadas AJAX, además dispone de una función de callback ya que prácticamente todo el trabajo se delega. Actualmente, por defecto tiene un límite de memoria de 512 MB en sistemas de 32 bits, 1gb en sistemas de 64 bits.

El límite se puede aumentar, pero se recomienda dividir el proceso en varios 'workers' si se llega a los límites. La última versión usada en producción es la v0.12.7 (versión estable) liberada el 9 Julio de 2015. Recientemente se ha liberado la versión v4.0, uniendo Node.js e io.js

REFERENCIAS

Aprendeconalf.es. (2004). Ejercicios de Programación con Python. <https://bit.ly/3PgX4wY>

Borrell, G. (2006). Control de versiones. Revista. <https://torroja.dmt.upm.es/media/files/cversiones.pdf>

David-Engel. (2021). Controladores de Administración. ¿Qué es y cuáles son sus características? <https://bit.ly/3JJJqhy>

Hernández, J. (2022) ¿Qué es Entorno de Desarrollo Integrado (IDE)? <https://bit.ly/3yFdjh8>

Joyanes, L. (2008). Fundamentos de la programación. Algoritmos y Estructura de Datos, 4ª Edición. Madrid: McGraw-Hill.

Álvarez, E., Andrade, S., De la Mora, M., Nieto, O., Trevilla, C., Valera, A., y Wals, J. (2006). Introducción a la Programación. UACM. ISBN: 968-9037-31-5 <https://bit.ly/39XmgZ6> .

Martínez, M. (2019). Definición de código, Definición, <https://bit.ly/3EdOg5H>

Perales G. (2015). Código Fuente - Concepto, usos y qué es código abierto. Concepto. <https://bit.ly/3viF7oh>

EduIt. (2010). Programación. <https://bit.ly/3ysaArw>



Licenciatura en Informática Administrativa 2022A

UNIDAD 4. TECNOLOGÍAS CONEXIÓN Y TRANSFERENCIA DE DATOS

OBJETIVO:

Identificar las principales tecnologías de conexión a datos para su uso e integración en entornos de desarrollo web, así como los estándares más recientes de etiquetado de datos para su transferencia.

4.1 REVISIÓN DE LA INSTALACIÓN Y CONFIGURACIÓN DE UN DBM.

Según Vázquez R, (2013) menciona que el modelo Mecanicista Basado en Datos (DBM), es un método híbrido que puede emplearse para la predicción de caudales, es decir, el modelo DBM es el que describe de mejor manera la ventaja ante una base de datos completa como MySQL ya que esta es mucho más ligera, y mucho más fácil de crear en un binario distribuido, ya que no hacen falta instalaciones de servidor de base de datos independientes.

Por otro lado, la base de datos DBM permite almacenar estructuras de datos de diferentes tamaños, usando un índice, y recuperando después la estructura ya sea mediante un índice o escaneando consecutivamente la base de datos. La base de datos DBM es mejor para los datos a los que se acceden con frecuencia pero que no se actualiza casi nunca, ya que tiende a ser bastante lenta creando entradas, pero es muy rápida recuperándolas.

Localización de problemas y reinstalación de DBM

La implementación GNU de GDBM instalada, junto con las bibliotecas NDBM compatibles con las distribuciones Linux es posible que se tenga que instalar explícitamente el paquete de desarrollo de biblioteca con el fin de compilar los archivos usando las rutinas NDBM. La biblioteca incluida y la de enlace necesarias varían dependiendo de la distribución, de manera que, aunque esté instalada, es posible que tenga que experimentar un poco para saber cómo compilar los archivos fuente usando NDBM. El caso más común es que GDBM esté instalada y que acepte, por defecto, el modo de compatibilidad NDBM.

Las distribuciones son las siguientes:

Licenciatura en Informática Administrativa 2022A



Figura 73. Distribuciones de DBM

Fuente: Vázquez (2013)

Rutinas DBM

La prestación DBM está compuesta por un archivo de cabecera y una biblioteca y debe vincularse cuando se compila el programa. La biblioteca se denomina DBM, pero como, normalmente, usamos la implementación GNU de Linux, tenemos que vincularla con esta implementación usando -LGDBM en la línea de compilación. El archivo de cabecera es NDBM.H

Según Vázquez R, (2013) el elemento básico de la base de datos DBM es un bloque de datos que hay que guardar, junto con un bloque acompañante de datos que actúa como la clave para recuperar los datos. Cada base de datos DBM debe tener una clave única para cada uno de los bloques de datos que ha de ser almacenado. El valor clave actúa como un índice de los datos guardados. No existe restricción alguna sobre las claves o sobre los datos, ni se han definido errores por usar datos o claves demasiado extensas. Las especificaciones permiten una implementación para limitar el tamaño de la pareja clave/datos en 1023 bytes, pero normalmente no hay límite, porque las implementaciones han sido más flexibles de lo que debían ser.

Para manipular estos bloques de datos, el archivo incluido ndbm.h define un nuevo tipo denominado datum. El contenido exacto de este tipo depende de cada implementación, pero debe tener como mínimo los siguientes miembros:

```
void *dptr;
size_t dsize
```



Licenciatura en Informática Administrativa 2022A

Datum es el tipo definido por typedef. En el archivo ndbm.h aparece también una definición de tipo para dbm, que es una estructura usada para acceder a la base de datos, al igual que se usa FILE para acceder a los archivos. Los internos de dbm typedef dependen de cada implementación y no deben usarse nunca.

Para referirse a un bloque de datos mientras esté usando la biblioteca dbm, debe declarar un datum, configurar dptr para que indique la parte inicial de los datos, y configurar dsize para que contenga su tamaño. El tipo datum hace referencia siempre tanto a los datos que hay que almacenar como al índice usado para acceder a dichos datos.

El tipo dbm se comprende mejor como análogo del tipo FILE. Cuando abre una base de datos dbm, se suelen crear dos archivos físicos: uno con extensión .pag y otro con extensión .dir. Se reenviará un único indicador dbm, que se usa para acceder a estos archivos directamente, ya que están pensados para acceder a ellos únicamente de las rutinas dbm.

Ejemplo de una base de datos en DBM

En primer lugar, están los archivos #includes, #defines, la función main, y la declaración de la estructura test_data:

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>

#include <gdbm-ndbm.h>

/* En algunos sistemas tendra que reemplazar lo anterior con
    #include <ndbm.h>
*/
#include <string.h>

#define TEST_DB_FILE "/tmp/dbm1_test"
#define ITEMS_USED 3

struct test_data {
    char misc_chars[15];
    int any_integer;
    char more_chars[21];
}; /* fin struct */

int main()
{
```

Figura 74. Estructura de una base de datos en DBM

Fuente: Vázquez (2013)



Licenciatura en Informática Administrativa 2022A

Por tanto, las bases de trabajo de una biblioteca DBM necesitan ciertas funciones con todo a detalle. Los prototipos para las principales funciones DBM son:

- dbm_open
- dbm_store
- dbm_fecth
- dbm_close

Configuración de un DBM

Según IBM Corporation (2014), una configuración DBM local no necesita una conexión de instancia, pero dicha conexión es necesaria para efectuar operaciones de configuración DBM remotas. Para actualizar la configuración del gestor de bases de datos para una instancia remota, es necesario conectarse primero a dicha instancia, es decir, para

actualizar en línea un parámetro de configuración, también es necesario conectarse primero a la instancia.

Por lo tanto, IBM Corporation (2014) menciona que es necesario utilizar la siguiente configuración:

MEMBER número-miembro

La cláusula MEMBER especifica a qué miembro debe aplicarse el cambio. La omisión de esta cláusula da como resultado la aplicación de un cambio a todos los miembros.

USING palabraclave-config valor

Especifica el parámetro de configuración del gestor de bases de datos que se debe actualizar. Para obtener una lista de parámetros de configuración, consulte el resumen de parámetros de configuración. valor especifica el valor que se debe asignar al parámetro.

AUTOMATIC

Algunos parámetros de configuración se pueden establecer en AUTOMATIC, lo que permite a Db2 ajustar automáticamente estos parámetros para reflejar los requisitos de recurso actuales. Para obtener una lista de parámetros de configuración que den soporte a la palabra clave AUTOMATIC, consulte el resumen de parámetros de configuración. Si se



Licenciatura en Informática Administrativa 2022A

especifica un valor junto a la palabra clave AUTOMATIC, es posible que influya en los cálculos automáticos. Para obtener detalles específicos sobre este comportamiento, consulte la documentación para el parámetro de configuración.

MANUAL

Inhabilita el ajuste automático para el parámetro de configuración. El parámetro se establece en su valor interno actual y ya no se actualiza automáticamente.

IMMEDIATE

Efectuar los cambios ahora, dinámicamente, mientras se ejecuta la instancia. IMMEDIATE es el valor por omisión, pero se necesita una conexión de instancia para que entre en vigor.

DEFERRED

Efectuar los cambios solo en el archivo de configuración, de forma que los cambios entren en vigor cuando se reinicie la instancia. Es la cláusula por omisión cuando funciona en la interfaz CLPPlus. Si se utiliza el procesador CLPPlus, no es necesario invocar DEFERRED.

```
>>-UPDATE--+DATABASE MANAGER+--+CONFIGURATION+----->
      +-DB MANAGER-----+  +-CONFIG-----+
      '-DBM-----'      '-CFG-----'

>----->
      '-MEMBER--número-miembro-'
      .
      V                                     I
>>-USING---palabraclave-config--+valor----->
      +-valor--AUTOMATIC--+
      +-AUTOMATIC-----+
      '-MANUAL-----'

      .- IMMEDIATE - .
>-----<
      '-DEFERRED--'

```

Figura 75. Configuración de un DBM

Fuente: IBM Corporation (2014)

Ejemplo de instalación y configuración de un DBM

En primer lugar, están los archivos #includes, #defines, la función main, y la declaración de la estructura test_data:



Licenciatura en Informática Administrativa 2022A

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>

#include <gdbm-ndbm.h>

/* En algunos sistemas tendra que reemplazar lo anterior con
    #include <ndbm.h>
*/
#include <string.h>

#define TEST_DB_FILE "/tmp/dbm1_test"
#define ITEMS_USED 3

struct test_data {
    char misc_chars[15];
    int any_integer;
    char more_chars[21];
}; /* fin struct */

int main()
{
```

Figura 76. Función Main

Fuente: Vázquez (2013)

Dentro de main, configuramos las estructuras items_to_store e items_received, la cadena key y los tipos datums:

```
struct test_data items_to_store[ITEMS_USED];
struct test_data item_retrieved;

char key_to_use[20];
int i, result;

datum key_datum;
datum data_datum;

DBM *dbm_ptr;
```

Figura 77. Estructura Items

Fuente: Vázquez (2013)



Licenciatura en Informática Administrativa 2022A

Una vez que se ha declarado el indicador de una estructura tipo dbm, abra su base de datos de prueba con acceso de lectura y escritura, creándola si fuese necesario:

```
dbm_ptr = dbm_open(TEST_DB_FILE, O_RDWR | O_CREAT, 0666);  
if (!dbm_ptr) {  
    fprintf(stderr, "Fallo al abrir base de datos\n");  
    exit (EXIT_FAILURE);  
} /* final de if */
```

Figura 78. Tipo de DBM

Fuente: Vázquez (2013)

Añada ahora algún dato a la estructura items_to_store:

```
memset(items_to_store, '\0', sizeof(items_to_store));  
strcpy(items_to_store[0].misc_chars, "Primero!");  
items_to_store[0].any_integer = 47;  
strcpy(items_to_store[0].more_chars, "foo");  
strcpy(items_to_store[1].misc_chars, "bar");  
items_to_store[1].any_integer = 13;  
strcpy(items_to_store[1].more_chars, "¿Mala suerte?");  
  
strcpy(items_to_store[2].misc_chars, "Tercera");  
items_to_store[2].any_integer = 3;  
strcpy(items_to_store[2].more_chars, "baz");
```

Figura 79. Estructura items_to_store

Fuente: Vázquez (2013)

Para cada entrada, ha de crear una clave para las futuras referencias. Se trata de la primera letra de una cadena y del entero. La clave será identificada después con ky_Datum, mientras que data_datum se refiere a la entrada items_to_store. después debe guardar los datos en una base de datos:



Licenciatura en Informática Administrativa 2022A

```
for (i = 0; i < ITEMS_USED; i++) {
    sprintf(key_to_use, "%c%c%d",
            items_to_store[i].misc_chars[0],
            items_to_store[i].more_chars[0],
            items_to_store[i].any_integer);

    key_datum.dptr = (void *)key_to_use;
    key_datum.dsize = strlen(key_to_use);
    data_datum.dptr = (void *)&items_to_store[i];
    data_datum.dsize = sizeof(struct test_data);

    result = dbm_store(dbm_ptr, key_datum, data_datum, DBM_REPLACE);
    if (result != 0) {
        fprintf(stderr, "dbm_store failed on key %s\n", key_to_use);
        exit(2);
    } /* final de if */
} /* final del for */
```

Figura 80. Referencias y guardar datos

Fuente: Vázquez (2013)

A continuación, y, para terminar, observe si puede recuperar los nuevos datos, y después, finalmente, debe cerrar la base de datos:

```
    sprintf(key_to_use, "bu%d",13);
    key_datum.dptr = key_to_use;
    key_datum.dsize = strlen(key_to_use);

    data_datum = dbm_fetch(dbm_ptr, key_datum);
    if (data_datum.dptr) {
        printf("Datos recuperados\n");
        memcpy(&item_retrieved, data_datum.dptr, data_datum.dsize);
        printf("Elemento obtenido - %s %d %s\n",
                item_retrieved.misc_chars,
                item_retrieved.any_integer,
                item_retrieved.more_chars);
    } /* Final if */
    else {
        printf("No se encuentran datos con la clave %s\n",key_to_use);
    } /* Final else */

    dbm_close(dbm_ptr);
    exit(EXIT_SUCCESS);
} /* Final main */
```

Figura 81. Cerrar la base de datos

Fuente: Vázquez (2013)



Licenciatura en Informática Administrativa 2022A

4.2 Estándares y tecnologías de conectividad a DB (ODBC, JDBC, OLE DB, etc.).

Según IBM, (2013) el estándar internacional para SQL define un lenguaje de alto nivel para acceder y manipular los datos que están almacenados en una base de datos relacional. El estándar internacional para SQL CLI define una interfaz de programación de aplicaciones (API) que utilizan los programas para emitir las sentencias SQL de una forma estandarizada a través de una serie de funciones API bien definidas.

OPEN DATABASE CONNECTIVITY (ODBC):

Según Nevárez, (1998) plantea que un API para acceso a bases de datos ampliamente utilizado. Está basado en las especificaciones Call-Level Interface (CLI) de X/Open e ISO/IEC y utiliza SQL como su lenguaje de acceso a bases de datos. ODBC está diseñado para permitir a una aplicación acezar diferentes bases de datos utilizando el mismo código fuente.

Implementación:

Estas aplicaciones para bases de datos llaman funciones de ODBC, las cuales están implementadas en controladores específicos a una base de datos. Las funciones del API de ODBC son implementadas por los desarrolladores de los controladores específicos para cada base de datos. Las aplicaciones llaman las funciones en estos controladores para acezar las bases de datos en forma independiente. El Driver Manager de ODBC administra la comunicación entre las aplicaciones y los controladores.

ODBC utiliza dos requerimientos en su arquitectura para estandarizar el acceso a bases de datos:

- Las aplicaciones deben ser capaces de acezar múltiples bases de datos utilizando el mismo código sin necesidad de recompilación o reenlace de este.
- Las aplicaciones deben ser capaces de acezar múltiples bases de datos simultáneamente.

Arquitectura:

ODBC tiene cuatro componentes:

- **Aplicación:** Es el componente que ejecuta llamadas a funciones ODBC para enviar sentencias SQL y recuperar resultados.
- **Driver Manager:** El Driver Manager carga y elimina controladores para la aplicación. Procesa las llamadas a las funciones ODBC o pasa estas al controlador.
- **Controlador:** El controlador procesa las llamadas a funciones ODBC, envía la petición SQL a una fuente específica de datos y regresa los resultados a la aplicación. Si es necesario, el controlador modifica la petición de la aplicación de tal forma que esta cumple con la sintaxis soportada por la base de datos asociada.

Licenciatura en Informática Administrativa 2022A

- **Fuente de datos:** La fuente de datos consiste en los datos que el usuario desea accezar además del manejador de base de datos, sistema operativo y, en algunos casos, red necesarios para accezar los datos.

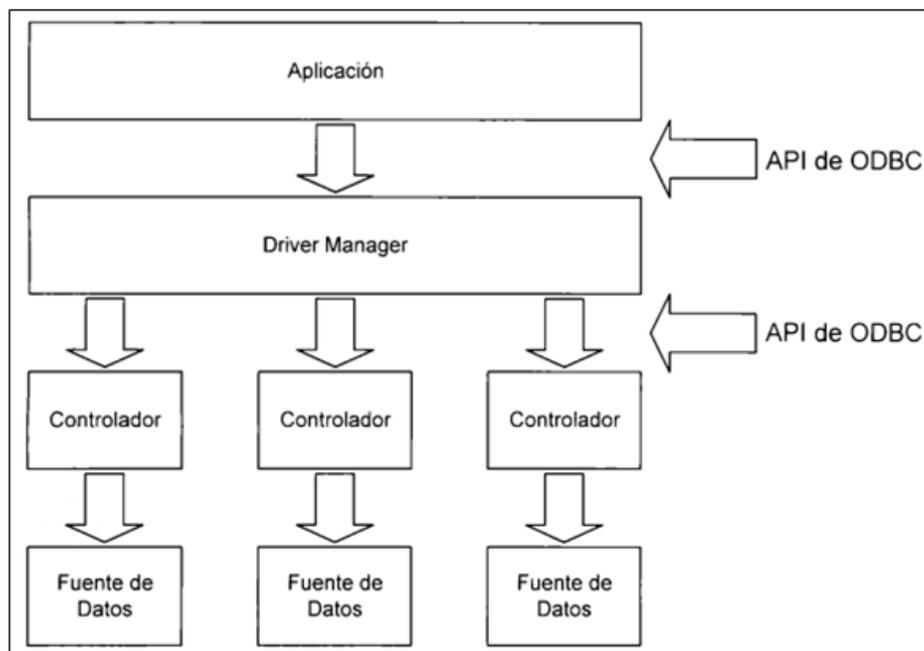


Figura 82. Arquitectura de ODBC.

Fuente: Nevárez, (1998)

La figura 82 muestra la relación entre estos cuatro componentes. La figura muestra cómo pueden existir múltiples controladores y fuentes de datos, lo cual permite a la aplicación el acceso simultáneo a datos de más de una fuente de datos.

Entre las tareas realizadas por un controlador se incluyen:

- Conectarse y desconectarse de la fuente de datos.
- Encontrar errores de funciones no revisados por el Driver Manager.
- Iniciar Transacciones.
- Enviar sentencias SQL a la fuente de datos para su ejecución.
- Enviar y recuperar datos de una fuente de datos incluyendo la conversión de tipos de datos especificados por la aplicación.

OLE DB / ADO:

Licenciatura en Informática Administrativa 2022A

OLE DB es la interface de programación a nivel sistema de Microsoft diseñada para proporcionar acceso a cualquier tipo de dato sin importar su tipo, formato o ubicación. OLE DB especifica un conjunto de interfaces COM que encapsulan u ocultan varios servicios de administración de bases de datos. Mientras OLE DB es una interface de programación a nivel sistema.

ADO es una interface de programación a nivel aplicación. es un modelo de programación de bases de datos que permite al programador escribir aplicaciones sobre datos OLE DB desde cualquier lenguaje incluyendo Visual Basic, Java, VBScript, JavaScript y C/C++.

Arquitectura:

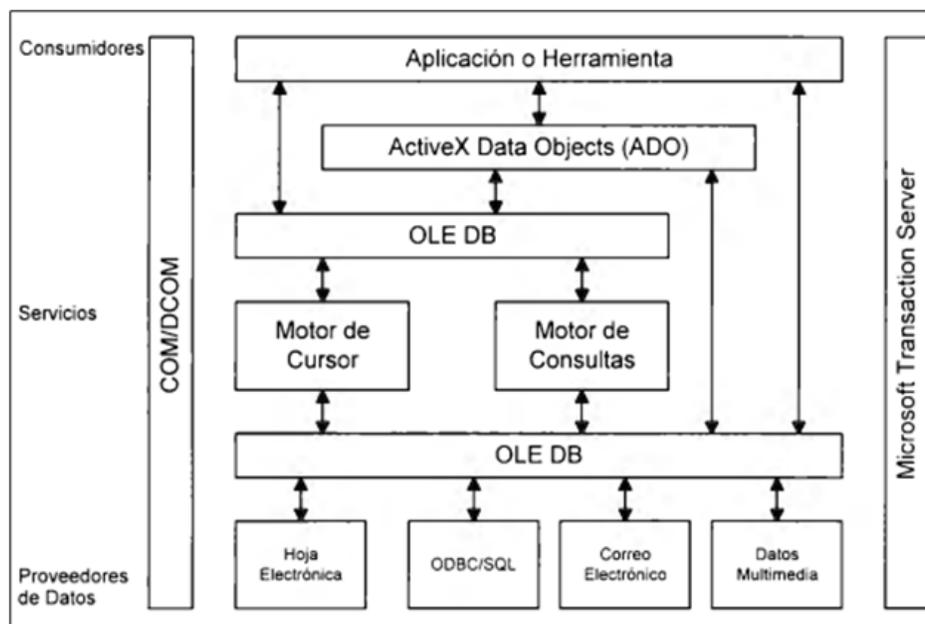


Figura 83. Arquitectura de UDA.

Fuente: Nevárez, (1998)

La figura 83 muestra la arquitectura de UDA. En esta arquitectura existen tres tipos generales de componentes de bases de datos:

- **Proveedores de datos:** Los proveedores de datos son componentes que representan las fuentes de datos como pueden ser bases de datos relacionales, hojas electrónicas y datos multimedia.



Licenciatura en Informática Administrativa 2022A

- **Servicios:** Los servicios son componentes que consumen y producen datos OLE DB.
- **Consumidores de datos:** Los consumidores de datos son componentes que consumen datos OLE DB. Estos pueden ser modelos de acceso a datos de alto nivel como ADO o aplicaciones escritas en lenguajes como C++ o Java.

JAVA DATABASE CONNECTIVITY (JDBC)

- **JDBC** es un API para la ejecución de sentencias SQL. Consiste en un conjunto de clases e interfaces escritas en el lenguaje de programación Java. JDBC proporciona un API estándar para desarrolladores de herramientas y aplicaciones para bases de datos.
- Utilizando JDBC es posible enviar sentencias SQL a diversas bases de datos. En otras palabras, usando JDBC, no es necesario escribir un programa para acceder una base de datos Oracle, otro para acceder una base de datos Sybase y otro para acceder una base de datos Informix.
-
- Con JDBC solo será necesario la escritura de un programa el cual enviará las sentencias SQL a la base de datos apropiada.

En resumen: JDBC hace posible tres cosas:

- Establece una conexión con la base de datos
- Envía sentencias SQL
- Procesa los resultados de estas sentencias.

```
Connection conexion = DriverManager.getConnection(  
    "oracle.jdbc.driver.OracleDriver", "scott", "tiger")  
Statement sentencia = conexion.createStatement();  
ResultSet resultado = sentencia.executeQuery("SELECT titulo, precio FROM  
album");
```

Figura 84. Ejemplo de Arquitectura de JDBC.

Fuente: Nevárez, (1998)

JDBC es una interface de bajo nivel, lo que significa que es utilizada para llamar sentencias SQL directamente. Puede utilizarse de esta forma o puede usarse también como una base sobre la cual construir interfaces y herramientas de más alto nivel.

LIVEWIRE DATABASE SERVICE

Las aplicaciones JavaScript corriendo en el Netscape Enterprise Server pueden utilizar el LiveWire Database Service para acceder bases de datos de Oracle, Sybase, Informix y 082,



Licenciatura en Informática Administrativa 2022A

además de servidores utilizando el estándar ODBC. Las aplicaciones corriendo en el Netscape FastTrack Server pueden acesar únicamente bases de datos a través de ODBC.

Generalmente, para interactuar con una base de datos usando LiveWire se siguen los siguientes pasos:

- Utilizar el objeto database o crear un objeto DbPool para establecer un pool de conexiones de bases de datos: Esto se realiza típicamente en la página inicial de la aplicación a menos que se requiera que los usuarios utilicen una conexión especial.
- Conectar el pool a la base de datos: Al igual que el anterior, este paso es realizado típicamente en la página inicial de la aplicación.
- Recuperar una conexión del pool: Esto es realizado implícitamente cuando se utiliza el objeto database o explícitamente cuando se utiliza el método connection del objeto DbPool.
- En los casos en que se desee actualizar información en la base de datos se deberá iniciar una transacción.
- Crear un cursor o llamar un procedimiento almacenado en la base de datos para trabajar con información de la base de datos. Esto puede implicar desplegar resultados de una consulta o actualizar el contenido de la base de datos. Será necesario cerrar cualquier cursor, conjunto de resultados o procedimientos almacenados que se haya utilizado.
- Completar (commit) o cancelar (rollback) cualquier transacción abierta.
- Liberar la conexión con la base de datos (en el caso de que se esté utilizando objetos Connection).

4.3 Lenguaje de consulta SQL y aplicación.

López, J. (2022) nos menciona que es un lenguaje estandarizado que sirve para definir y manipular los datos de una base de datos relacional y de acuerdo con el modelo relacional de datos, la base de datos se crea como un conjunto de tablas, las relaciones se representan mediante valores en las tablas y los datos se recuperan especificando una tabla de resultados que puede derivarse de una o más tablas base.

Las sentencias de SQL las ejecuta un gestor de bases de datos, Una de las funciones del gestor de bases de datos es transformar la especificación de una tabla resultante en una secuencia de operaciones internas que optimicen la recuperación de los datos, Esta transformación se produce en dos fases: preparación y vinculación, El método de preparación de una sentencia de SQL y la persistencia de su formato operativo diferencian SQL estático de SQL dinámico.



Licenciatura en Informática Administrativa 2022A



Figura 85. Funcionamiento de SQL

Fuente: López (2022)

Moreno, G. (2005) resalta que es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros y no a registros individuales permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros.

Características de SQL:

- Lenguaje de definición de datos: El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.
- Lenguaje interactivo de manipulación de datos: El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.
- Integridad: El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.
- Definición de vistas: El LDD incluye comandos para definir las vistas.
- Control de transacciones: SQL tiene comandos para especificar el comienzo y el final de una transacción.

Mendoza, (2006) indica que este lenguaje se convirtió en estándar en 1986 por la ANSI, American National Standards Institute, y posteriormente por ISO, International Organización for Standardization, en 1987. No obstante, existen numerosos dialectos de SQL implementados por diversas alternativas de motores de bases de datos. Estos dialectos pueden no cubrir en estándar completo o tener variaciones leves como el tratamiento de mayúsculas y minúsculas, los índices, etc.

IBM (2018) el lenguaje SQL es un lenguaje estandarizado que sirve para definir y manipular los datos de una base de datos relacional. De acuerdo con el modelo relacional de datos, la base de datos se crea como un conjunto de tablas, las relaciones se representan



Licenciatura en Informática Administrativa 2022A

mediante valores en las tablas y los datos se recuperan especificando una tabla de resultados que puede derivarse de una o más tablas base.

Las sentencias de SQL las ejecuta un gestor de bases de datos. Una de las funciones del gestor de bases de datos es transformar la especificación de una tabla resultante en una secuencia de operaciones internas que optimicen la recuperación de los datos. Esta transformación se produce en dos fases: preparación y vinculación. Todas las sentencias de SQL ejecutables deben prepararse antes de su ejecución. El resultado de esta preparación es el formato operativo o ejecutable de la sentencia. El método de preparación de una sentencia de SQL y la persistencia de su formato operativo diferencian SQL estático de SQL dinámico.

Por otro lado, para Ramos (2018) el Lenguaje de Consulta Estructurado popularmente conocido por sus siglas en inglés como SQL, es un tipo de lenguaje de programación que ayuda a solucionar problemas específicos o relacionados con la definición, manipulación e integridad de la información representada por los datos que se almacenan en las bases de datos. Algunos aspectos de SQL están basados en el cálculo relacional, algunos en el álgebra relacional que provienen del modelo relacional y otros a ninguno de los dos, sino que son parte de SQL.

Pérez, L. (2019) da a conocer que el SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática de tipo relacional. Cuando se necesita recuperar datos de una base de datos, se utiliza al lenguaje SQL para efectuar la petición, ya sea en modo consola o desde el código programado inmerso en una aplicación tipo HMI que ha sido desarrollado desde un lenguaje anfitrión (GAMBAS BASIC, VISUAL BASIC, etc). El DBMS, sea MySQL, Oracle, SQL Server, etc, procesa la petición SQL, recupera los datos solicitados y los devuelve. Este proceso de solicitar datos de la Base de datos y de recibir los resultados se denomina consulta (Query) a la base de datos, de aquí su nombre Structured Query Language. Por lo tanto, el SQL es un lenguaje completo de control e interacción con un sistema de gestión de bases de datos.

EL PAPEL DEL LENGUAJE DE CONSULTA ESTRUCTURADO SQL

Pérez (2019) indica que SQL no es en sí mismo un DBMS, ni un producto autónomo, no se puede ir a una tienda de informática y comprar al SQL solo. El SQL es parte integral de un DBMS (ORACLE; MySQL, etc), Es un lenguaje que funciona como una herramienta que permite comunicar al DBMS con la base de datos.

Todo DBMS este compuesto de diferentes partes virtuales programadas que engranadas cumplen con todas sus funciones, pero existe una parte en específico a la que se le llama motor de la base de datos, que es la que estructura, almacena y recupera realmente los datos desde y hacia el disco duro. Acepta peticiones SQL ya sea modo consola, gráfico o desde el código de un lenguaje anfitrión. El motor es el verdadero corazón del DBMS.



Licenciatura en Informática Administrativa 2022A

Papeles de SQL:

- El SQL es un lenguaje de consultas interactivas. Los usuarios escriben órdenes para recuperar datos y mostrarlos en la pantalla, proporcionando una herramienta conveniente y fácil de utilizar para consultas rápidas a la base de datos.
- El SQL es un lenguaje de programación de base de datos. Los programadores insertan ordenes SQL en sus programas de aplicación para acceder a los datos de la base de datos.
- El SQL es un lenguaje de administración de bases de datos. El administrador de la base de datos responsable de gestionar una base de datos en un computador utiliza al SQL para definir la estructura de la base de datos y para controlar el acceso a los datos almacenados.
- El SQL es un lenguaje cliente / servidor. Los programas del computador utilizan al SQL para comunicarse sobre una LAN con servidores de bases de datos que almacenan los datos compartidos (Strings de enlaces y conexión a nivel de código). Las aplicaciones tipo Cliente / Servidor minimizan el tráfico por la red y permite que tanto los PC como los servidores efectúan mejor su trabajo.
- El SQL es un lenguaje de bases de datos distribuidas. Los DBMS utilizan al SQL para ayudar a distribuir los datos a través de muchos sistemas informáticos conectados. El software DBMS de cada DBS utiliza la SQL para comunicarse con los otros DBS enviando peticiones para acceso a datos.
- El SQL es un lenguaje de puerta de enlace a otras bases de datos. En una red informática con una mezcla de diferentes productos DBMS, el SQL se utiliza a menudo como puerta de enlace que permite que un DBMS se comunique con otros DBMS.

Ejemplos:

Cómo usar una simple cláusula de UNION en SQL en la instrucción select

Según Jayaram, P. (2019) En este ejemplo, el conjunto de resultados incluye un conjunto distinto de filas del primer conjunto y en el segundo conjunto. El siguiente ejemplo se basa en las reglas 1, 3 y 5.

```
1 SELECT *  
2 FROM
```



Licenciatura en Informática Administrativa 2022A

```
3 (
4 (
5 SELECT 1 A
6 UNION
7 SELECT 2
8 UNION
9 SELECT 3
10 )
11 UNION
12 (
13 SELECT 3 B
14 UNION
15 SELECT 4
16 UNION
17 SELECT 5
18 )
19 UNION ALL
20 (
21 SELECT 8 c
22 UNION
23 SELECT 9
24 UNION
25 SELECT 1
26 )
27 ) T;
```

Figura 86. Uso de SQL

Fuente: Jayaram (2019)

La salida es una combinación de operadores Union y Union All los mismos que usan paréntesis.



Licenciatura en Informática Administrativa 2022A

	A
1	1
2	2
3	3
4	4
5	5
6	1
7	8
8	9

A UNION B

UNION ALL C

Figura 87. Uso de SQL

Fuente: Jayaram (2019)

¿Cómo usar SQL Union con las consultas que tienen la cláusula WHERE?

El siguiente ejemplo muestra el uso de Union en dos instrucciones SELECT con una cláusula WHERE y una cláusula ORDER BY.

El siguiente ejemplo se basa en las reglas 1,2 y 3.



Licenciatura en Informática Administrativa 2022A

```
1 SELECT P1.ProductModelID,  
2     P1.Name  
3 FROM Production.ProductModel P1  
4 WHERE ProductModelID IN(3, 4)  
5 UNION  
6 SELECT P2.ProductModelID,  
7     P2.Name  
8 FROM Production.ProductModel P2  
9 WHERE P2.ProductModelID IN(3, 4)  
10 ORDER BY P1.Name;
```

Figura 88. Uso de Select

Fuente: Jayaram (2019)

```
SELECT P1.ProductModelID, P1.Name  
FROM Production.ProductModel P1  
WHERE ProductModelID IN (3, 4)  
UNION  
SELECT P2.ProductModelID, P2.Name  
FROM Production.ProductModel P2  
WHERE P2.ProductModelID IN (3, 4)  
ORDER BY P1.Name;  
GO
```

	ProductModelID	Name
1	3	Full-Finger Gloves
2	4	Half-Finger Gloves

Figura 89. Uso de SQL

Fuente: Jayaram (2019)

¿Cómo usar la cláusula SELECT INTO con SQL Union?

El siguiente ejemplo crea una nueva tabla dbo.dummy utilizando la cláusula INTO en la primera instrucción SELECT que contiene el conjunto de resultados finales de la Unión de las columnas ProductModel y el nombre de dos conjuntos de resultados diferentes. En este



Licenciatura en Informática Administrativa 2022A

caso, se deriva de la misma tabla, pero en un caso y una situación del mundo real, esto también puede dar origen a que sean dos tablas diferentes. El siguiente ejemplo se basa en las reglas 1, 2 y 4.

```
1 DROP TABLE IF EXISTS dbo.dummy;  
2 SELECT P1.ProductModelID,  
3     P1.Name  
4 INTO dummy  
5 FROM Production.ProductModel P1  
6 WHERE ProductModelID IN(3, 4)  
7 UNION  
8 SELECT P2.ProductModelID,  
9     P2.Name  
10 FROM Production.ProductModel P2  
11 WHERE P2.ProductModelID IN(3, 4)  
12 ORDER BY P1.Name;  
13 GO  
14  
15 SELECT *  
16 FROM dbo.Dummy;
```

Figura 90. Uso de SQL

Fuente: Jayaram (2019)



Licenciatura en Informática Administrativa 2022A

```
SELECT * FROM dbo.Dummy
```

	ProductModelID	Name
1	3	Full-Finger Gloves
2	4	Half-Finger Gloves

Figura 91. Uso de SQL

Fuente: Jayaram (2019)

¿Cómo usar SQL Union con las consultas que tienen una cláusula WHERE y una cláusula ORDER BY?

Esto solo es posible cuando utilizamos funciones TOP o funciones agregadas en cada instrucción select del operador Union. En este caso, se enumeran las 10 filas principales de cada conjunto de resultados y luego las mismas se combinan con las filas usando la cláusula Union para obtener un resultado final. También usted verá que el orden por cláusula se coloca en toda la instrucción select.

```
1 SELECT a.ModelID,  
2     a.Name  
3 FROM  
4 (  
5     SELECT TOP 10 ProductModelID ModelID,  
6         Name  
7     FROM Production.ProductModel  
8     WHERE ProductModelID NOT IN(3, 4)  
9     ORDER BY Name DESC  
10 ) a  
11 UNION
```



Licenciatura en Informática Administrativa 2022A

```

12      SELECT b.ProductModelID,
13              b.Name
14      FROM
15      (
16          SELECT TOP 10 ProductModelID,
17              Name
18          FROM Production.ProductModel
19          WHERE ProductModelID IN(5, 6)
20          ORDER BY Name DESC
21      ) b;

```

Figura 92. Uso de SQL

Fuente: Jayaram (2019)

	ModelID	Name
1	5	HL Mountain Frame
2	6	HL Road Frame
3	34	Touring-1000
4	35	Touring-2000
5	36	Touring-3000
6	37	Women's Mountain Shorts
7	38	Women's Tights
8	43	Touring Rear Wheel
9	91	Touring Tire
10	94	Touring Tire Tube
11	111	Water Bottle
12	120	Touring-Panniers

Figura 93. Uso de SQL

Fuente: Jayaram (2019)



Licenciatura en Informática Administrativa 2022A

¿Cómo usar SQL Union y SQL Pivot?

En el siguiente ejemplo, estamos tratando de combinar más de un conjunto de resultados. En una situación del mundo real, puede tener números financieros de varias regiones o departamentos y como las tablas pueden tener las mismas columnas y tipos de datos, pero usted desea colocarlos en un conjunto de filas y en un solo informe. En tal escenario, usaría la cláusula Union y es muy fácil combinar resultados y transformar los datos en un informe más significativo.

En este ejemplo, ProductModel se clasifica en Top10, Top100, Top 100 y transforma las filas como un conjunto de valores agregados en las columnas respectivas. El siguiente ejemplo se basa en la regla 2.

```
1 SELECT MAX(Top10) Top10,
2     MAX(Top100) Top100,
3     MAX(Top1000) Top100
4 FROM
5 (
6     SELECT COUNT(*) Top10,
7           0 Top100,
8           0 Top1000
9     FROM Production.ProductModel
10    WHERE ProductModelID < 10
11    UNION
12    SELECT 0,
13           COUNT(*),
14           0
15    FROM Production.ProductModel
16    WHERE ProductModelID > 11
17           AND ProductModelID < 100
18    UNION
19    SELECT 0,
20           0,
21           COUNT(*)
```



Licenciatura en Informática Administrativa 2022A

```
22 FROM Production.ProductModel
23 WHERE ProductModelID > 101
24 ) T;
```

Figura 94. Uso de SQL

Fuente: Jayaram (2019)

Los valores NULL son muy importantes con los Operadores de conjuntos y se tratan como de bases de datos de ciudadanos de segunda clase. Debido a que los NULL se consideran únicos y si dos filas tienen un NULL en la misma columna, entonces se considerarían idénticos, por lo que, en ese caso, en realidad está comparando un NULL con un NULL y obteniendo igualdad. En el siguiente ejemplo, verá el uso de valores NULL. En este caso, usted puede observar que se opera con la función de agregado, máx.

```
1 SELECT MAX(Top10) Top10,
2 MAX(Top100) Top100,
3 MAX(Top1000) Top100
4 FROM
5 (
6 SELECT COUNT(*) Top10,
7 NULL Top100,
8 NULL Top1000
9 FROM Production.ProductModel
10 WHERE ProductModelID < 10
11 UNION
12 SELECT NULL,
13 COUNT(*),
14 NULL
15 FROM Production.ProductModel
16 WHERE ProductModelID > 11
17 AND ProductModelID < 100
18 UNION
19 SELECT NULL,
```



Licenciatura en Informática Administrativa 2022A

```

20          NULL,
21          COUNT(*)
22          FROM Production.ProductModel
23          WHERE ProductModelID > 101
24      ) T;

```

Figura 95. Uso de SQL

Fuente: Jayaram (2019)

Results		Messages	
	Top10	Top100	Top100
1	9	88	27

Figura 96. Uso de SQL

Fuente: Jayaram (2019)

¿Cómo usar SQL Union con Instrucciones Group y Have?

Los siguientes ejemplos usan el operador Unión para combinar el resultado de la tabla que tiene en su totalidad la cláusula condicional definida usando la cláusula Group by y Having clause.

El apellido se analiza especificando las condiciones en la cláusula have.

El siguiente ejemplo se basa en la regla 5.

```

1      SELECT pp.lastname,
2          COUNT(*) repeatedtwice,
3          0 Repeatedthrice
4      FROM Person.Person AS pp

```



Licenciatura en Informática Administrativa 2022A

5	JOIN HumanResources.Employee AS e ON
6	e.BusinessEntityID = pp.BusinessEntityID
7	GROUP BY pp.lastname
8	HAVING COUNT(*) = 2
9	UNION
10	SELECT pp.LastName,
11	0,
12	COUNT(*) NtoZRange
13	FROM Person.Person AS pp
14	JOIN HumanResources.Employee AS e ON
15	e.BusinessEntityID = pp.BusinessEntityID
	GROUP BY pp.LastName
	HAVING COUNT(*) > 2;

Figura 97. Uso de SQL

Fuente: Jayaram (2019)

Podemos ver que los apellidos se derivan en dos columnas diferentes usando el operador Unión.

	lastname	repeatedtwice	Repeatedthrice
1	Baker	2	0
2	Brown	0	3
3	Campbell	2	0
4	Chen	2	0
5	Hamilton	2	0
6	Hill	0	3
7	Johnson	0	3
8	Kane	2	0
9	Li	2	0
10	Liu	0	3
11	Martin	2	0
12	Miller	0	3
13	Randall	2	0
14	Smith	2	0
15	Valdez	2	0

Figura 98. Uso de SQL

Fuente: Jayaram (2019)



Licenciatura en Informática Administrativa 2022A

4.4. LENGUAJE DE ETIQUETADO DE DATOS (JSON, XML) Y APLICACIÓN

JSON

JavaScript Object Notation (JSON) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Es comúnmente utilizado para transmitir datos en aplicaciones web (por ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o viceversa).

Aunque es muy parecido a la sintaxis de objeto literal de JavaScript, puede ser utilizado independientemente de JavaScript, y muchos entornos de programación poseen la capacidad de leer y generar JSON.

Los JSON son cadenas - útiles cuando se quiere transmitir datos a través de una red. Debe ser convertido a un objeto nativo de JavaScript cuando se requiera acceder a sus datos. Esto no es un problema, dado que JavaScript posee un objeto global JSON que tiene los métodos disponibles para convertir entre ellos.

Un objeto JSON puede ser almacenado en su propio archivo, que es básicamente sólo un archivo de texto con una extensión .json, y una de application/json. (Mozilla, 2020)

Características

- JSON es sólo un formato de datos — contiene sólo propiedades, no métodos.
- JSON requiere usar comillas dobles para las cadenas y los nombres de propiedades. Las comillas simples no son válidas.
- Una coma o dos puntos mal ubicados pueden producir que un archivo JSON no funcione. Se debe ser cuidadoso para validar cualquier dato que se quiera utilizar (aunque los JSON generados por computador tienen menos probabilidades de tener errores, mientras el programa generador trabaje adecuadamente). Es posible validar JSON utilizando una aplicación como JsonLint.
- JSON puede tomar la forma de cualquier tipo de datos que sea válido para ser incluido en un JSON, no sólo arreglos u objetos. Así, por ejemplo, una cadena o un número único podrían ser objetos JSON válidos.
- A diferencia del código JavaScript en que las propiedades del objeto pueden no estar entre comillas, en JSON, sólo las cadenas entre comillas pueden ser utilizadas como propiedades.

Estructura

Es posible incluir los mismos tipos de datos básicos dentro de un JSON que en un objeto estándar de JavaScript - cadenas, números, arreglos, booleanos, y otros literales de objeto. Esto permite construir una jerarquía de datos, como ésta:



Licenciatura en Informática Administrativa 2022A

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Supertuman reflexes"
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

Figura 99. Jerarquía de datos en Json

Fuente: Mozilla (2020)

Si se carga este objeto en un programa de JavaScript, convertido (parseado) en una variable llamada `superHeroes` por ejemplo, se podría acceder a los datos que contiene utilizando la misma notación de punto/corchete que se revisó en el artículo JavaScript object basics. Por ejemplo:

```
superHeroes.homeTown
```

```
superHeroes['active']
```

Para acceder a los datos que se encuentran más abajo en la jerarquía, simplemente se debe concatenar los nombres de las propiedades y los índices de arreglo requeridos. Por ejemplo, para acceder al tercer superpoder del segundo héroe registrado en la lista de miembros, se debería hacer esto:

```
superHeroes['members'][1]['powers'][2]
```



Licenciatura en Informática Administrativa 2022A

1. Primero el nombre de la variable — superHeroes.
2. Dentro de esta variable para acceder a la propiedad members utilizamos ["members"].
3. members contiene un arreglo poblado por objetos. Para acceder al segundo objeto dentro de este arreglo se utiliza [1].
4. Dentro de este objeto, para acceder a la propiedad powers utilizamos ["powers"].
5. Dentro de la propiedad powers existe un arreglo que contiene los superpoderes del héroe seleccionado. Para acceder al tercer superpoder se utiliza [2].

Arreglos como Json

Anteriormente se mencionó que el texto JSON básicamente se parece a un objeto JavaScript, y esto es en gran parte cierto. La razón de esto es que un arreglo es también un JSON válido, por ejemplo:

```
{
  "name": "Molecule Man",
  "age": 29,
  "secretIdentity": "Dan Jukes",
  "powers": [
    "Radiation resistance",
    "Forming time",
    "Radiation blast"
  ]
},
{
  "name": "Madame Oppercut",
  "age": 39,
  "secretIdentity": "Jane Wilson",
  "powers": [
    "Million tonne punch",
    "Damage resistance",
    "Superhuman reflexes"
  ]
}
```

Figura 100. Arreglos como Json

Fuente: Mozilla (2020)

Éste es un JSON perfectamente válido. Para acceder a esta versión convertida se debe comenzar con un índice de arreglo, por ejemplo[0][["powers"]][0].

Ejemplo

A continuación, se muestra un ejemplo de cómo podemos utilizar algunos datos JSON en un sitio web.

Haga una copia local de los archivos heroes.html y style.css. El último contiene un CSS simple para dar estilo a la página, mientras el primero contiene un HTML muy sencillo:



Licenciatura en Informática Administrativa 2022A

```
<header>  
</header>  
  
<section>  
</section>
```

Figura 101. Inicio de etiquetas

Fuente: Mozilla (2020)

Además de un elemento `<script>` que contiene el código JavaScript que se escribirá en este ejercicio. En este momento sólo contiene dos líneas, que hacen referencia a los elementos `<header>` y `<section>` y los almacena en variables:

```
const header = document.querySelector('header');  
const section = document.querySelector('section');
```

Figura 102. Contenido de la etiqueta script

Fuente: Mozilla (2020)

Los datos se cargarán en esta página y se desplegarán a través de la manipulación del DOM de esta forma:

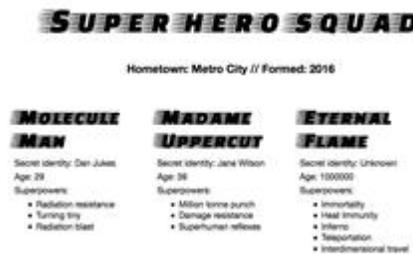


Figura 103. DOM

Fuente: Mozilla (2020)



Licenciatura en Informática Administrativa 2022A

Obteniendo el JSON

Para obtener el JSON se utilizará un API llamado XMLHttpRequest (a menudo llamado XHR). Éste es un objeto JavaScript muy útil que permite realizar solicitudes de red para recuperar recursos desde un servidor vía JavaScript (por ejemplo: imágenes, texto, JSON, incluso código HTML), con lo que es posible actualizar pequeñas secciones de contenido sin tener que volver a cargar la página entera. Con ello se obtienen páginas web más interactivas, pero está fuera del alcance de este artículo entrar en detalle.

1. Para empezar, se debe almacenar la URL del JSON que se quiere recuperar en una variable. Agregue lo siguiente al final del código JavaScript:

```
const requestURL = 'https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json';
```

Figura 104. Almacenamiento de la URL del JSON

Fuente: Mozilla (2020)

2. Para crear una solicitud, se necesita crear una nueva instancia de objeto de solicitud desde el constructor XMLHttpRequest, utilizando la palabra clave new. Agregue lo siguiente a continuación de la última línea:

```
const request = new XMLHttpRequest();
```

Figura 105. Creación de la solicitud

Fuente: Mozilla (2020)

3. Ahora es necesario abrir una nueva solicitud utilizando el método open (). Agregue la siguiente línea:

```
request.open('GET', requestURL);
```

Figura 106. Abrir solicitud

Fuente: Mozilla (2020)



Licenciatura en Informática Administrativa 2022A

Esto requiere al menos dos parámetros — Existen otros parámetros opcionales disponibles. Sólo se requieren los dos obligatorios para este ejemplo:

- El método HTTP a usar cuando se hace una solicitud en red. En este caso GET es adecuado, dado que sólo se están recuperando algunos datos simples.
- La URL a la que se realiza la solicitud — esta es la URL del archivo que se almacenó antes.

4. Luego, agregue las siguientes dos líneas — establecemos el responseType a JSON, de esta forma ese XHR sabe que el servidor estará retornando JSON y que esto debería ser convertido en segundo plano en un objeto JavaScript. Entonces se envía la solicitud con el método send():

```
request.responseType = 'json';  
request.send();
```

Figura 107. Envío de solicitud

Fuente: Mozilla (2020)

5. La última parte de esta sección comprende la espera por la respuesta a retornar desde el servidor y luego, manejarla. Agregue el siguiente código bajo el código previo:

```
request.onload = function() {  
  const superHeroes = request.response;  
  populateHeader(superHeroes);  
  showHeroes(superHeroes);  
}
```

Figura 108. Espera y manejo de la respuesta

Fuente: Mozilla (2020)

En este punto se está almacenando la respuesta a la solicitud (disponible en la propiedad response) en una variable llamada superHeroes; esta variable ahora contendrá el objeto JavaScript basado en el JSON. Luego se pasa el objeto como argumento a dos funciones — la primera llenará el <header> con los datos correctos, mientras la segunda creará una tarjeta de información para cada héroe en el equipo y la insertará en <section>.

Se ha contenido el código en un manejador de eventos que se activa cuando se dispara el evento de carga (ver onload) — esto es porque el evento de carga se dispara cuando la



Licenciatura en Informática Administrativa 2022A

respuesta ha sido retornada de forma exitosa; de esta manera se garantiza que request.response estará disponible cuando se intente hacer algo con ella.

Poblando el encabezado

Se han obtenido los datos desde el JSON y convertidos en un objeto de JavaScript. Ahora, se utilizarán estos datos escribiendo las dos funciones que fueron referenciadas previamente. Antes que todo, agregue la siguiente definición de función a continuación del código previo:

```
function populateHeader(jsonObj) {  
  const myH1 = document.createElement('h1');  
  myH1.textContent = jsonObj['squadName'];  
  header.appendChild(myH1);  
  
  const myPara = document.createElement('p');  
  myPara.textContent = 'Hometown: ' + jsonObj['homeTown'] + ' // Formed: ' +  
  jsonObj['formed'];  
  header.appendChild(myPara);  
}
```

Figura 109. Población del encabezado

Fuente: Mozilla (2020)

Se ha llamado al parámetro jsonObj, para recordar que este es un objeto JavaScript originado desde un JSON. Primero se crea un elemento <h1> (en-US) con createElement(), se asigna su textContent igual a la propiedad squadName del objeto, luego se agrega al encabezado utilizando appendChild(). A continuación, se realiza una operación muy parecida en un párrafo: se crea, se asigna su contenido de texto y se agrega al encabezado. La única diferencia es que su texto se asigna a una cadena concatenada que contiene las propiedades homeTown y formed del objeto.

Creación de las tarjetas de información del héroe Luego, agregue la siguiente función al final del código, que crea y muestra las tarjetas de los superhéroes:

Licenciatura en Informática Administrativa 2022A

```
function showHeroes(jsonObj) {
  const heroes = jsonObj['members'];

  for (var i = 0; i < heroes.length; i++) {
    const myArticle = document.createElement('article');
    const myH2 = document.createElement('h2');
    const myPara1 = document.createElement('p');
    const myPara2 = document.createElement('p');
    const myPara3 = document.createElement('p');
    const myList = document.createElement('ul');

    myH2.textContent = heroes[i].name;
    myPara1.textContent = 'Secret identity: ' + heroes[i].secretIdentity;
    myPara2.textContent = 'Age: ' + heroes[i].age;
    myPara3.textContent = 'Superpowers: ';

    const superPowers = heroes[i].powers;
    for (var j = 0; j < superPowers.length; j++) {
      const listItem = document.createElement('li');
      listItem.textContent = superPowers[j];
      myList.appendChild(listItem);
    }

    myArticle.appendChild(myH2);
    myArticle.appendChild(myPara1);
    myArticle.appendChild(myPara2);
    myArticle.appendChild(myPara3);
    myArticle.appendChild(myList);

    section.appendChild(myArticle);
  }
}
```

Figura 110. Creación de las tarjetas de información del héroe

Fuente: Mozilla (2020)

Para empezar, se almacena la propiedad members del objeto JavaScript en una nueva variable. Este arreglo contiene múltiples objetos que contienen la información para cada héroe.

A continuación, se utiliza un ciclo [for](#) para recorrer cada objeto en el arreglo. Para cada uno:

1. Se crean varios elementos nuevos: un <article>, un <h2>, tres <p>s, y una .
2. Se asigna al <h2> para que muestre el name del héroe.
3. Se completan los tres párrafos con su secretIdentity, age, y una línea que diga "Superpowers:" para introducir la información de la lista.
4. Se almacena la propiedad powers en otra variable nueva llamada superPowers — que contiene un arreglo que lista los superpoderes del héroe actual.



Licenciatura en Informática Administrativa 2022A

5. Para recorrer los superpoderes del héroe, se utiliza otro ciclo for — para cada uno se crea un elemento , se asigna el superpoder a él y luego se pone el listItem dentro del elemento (myList) utilizando appendChild().
6. Lo último es agregar los <h2>, <p>s, y dentro del <article> (myArticle), luego se agrega <article> dentro de <section>. El orden en que las cosas son agregadas es importante, dado que este es el orden en el que aparecerán dentro del HTML. (Mozilla, 2020).

XML

XML es un lenguaje de marcado similar a HTML. Significa Extensible Markup Language (Lenguaje de Marcado Extensible) y es una especificación de W3C como lenguaje de marcado de propósito general. Esto significa que, a diferencia de otros lenguajes de marcado, XML no está predefinido, por lo que debes definir tus propias etiquetas. El propósito principal del lenguaje es compartir datos a través de diferentes sistemas, como Internet.

Para que un documento XML sea correcto, debe ser un documento bien formado, cumpliendo todas las reglas de sintaxis de XML, y válidas, acorde a las reglas de un lenguaje específico. Un ejemplo de un documento que no está bien formado es uno que tiene una etiqueta de apertura y no tiene una de cierre, ni se cierra en si misma.

En el ejemplo siguiente, vemos un documento en el que una etiqueta que no se cierra a sí misma, no tiene etiqueta de cierre.

```
<message>
  <warning>
    Hola, mundo
  <!--missing </warning> -->
</message>
```

Figura 111. Documento XML mal formado

Fuente: Mozilla (2021)

Versión correcta del mismo documento:

```
<message>
  <warning>
    Hola, mundo
  </warning>
</message>
```

Figura 112. Documento XML bien formado

Fuente: Mozilla (2021)



Licenciatura en Informática Administrativa 2022A

Para ser válido, un documento XML necesita cumplir ciertas reglas de semántica que son generalmente definidas en un esquema XML o en una Definición de Tipo de Documento (en-US) (DTD). Un documento que contiene una etiqueta no definida es inválido. Por ejemplo, si nunca definimos la etiqueta <warning>, el documento anterior no sería válido.

Entidades

Al igual que HTML, XML ofrece métodos (llamados entidades) para referir a algunos caracteres especiales reservados (como el signo de mayor que, usado para las etiquetas). Hay cinco de estos caracteres que debes conocer:

Entidad	Caracter	Descripción
<	<	Menor que
>	>	Mayor que
&	&	Ampersand
"	"	Comilla doble
'	'	Apóstrofe (o comilla sencilla)

Figura 113. Entidades en XML

Fuente: Mozilla (2021)

Aunque solo hay cinco entidades declaradas, se pueden añadir más usando el DTD (en-US) del documento. Por ejemplo, para crear una nueva entidad &warning;, puedes hacer esto:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE body [
  <!ENTITY warning "Advertencia: Algo malo pasó... porfavor intente de
nuevo.">
]>
<body>
  <message> &warning; </message>
</body>
```

Figura 114. Creación de entidad

Fuente: Mozilla (2021)



Licenciatura en Informática Administrativa 2022A

Presentando XML

XML es generalmente para propósitos descriptivos, pero hay maneras de presentar datos XML. Si no defines una manera específica de renderizar XML, el contenido puro de XML es mostrado en el navegador.

Una forma de estilizar la salida de XML es especificando CSS que aplique al documento usando la instrucción de procesamiento `xml-stylesheet`.

```
<?xml-stylesheet type="text/css" href="stylesheet.css">
```

Figura 115. Especificación de CSS

Fuente: Mozilla (2021)

También hay otra manera más poderosa de presentar XML: Extensible Stylesheet Language Transformations (XSLT) que puede ser usado para transformar XML a otros lenguajes como HTML. Esto hace a XML increíblemente versátil.

```
<?xml-stylesheet type="text/xsl" href="transform.xsl"?>
```

Figura 116. Transformación de XML a HTML

Fuente: Mozilla (2021)

Ejemplos:

```
1 <persona>
2   <nombres>Elsa</nombres>
3   <apellidos>Zambrano</apellidos>
4   <fecha-de-nacimiento>
5     <día>18</día>
6     <mes>6</mes>
7     <año>1996</año>
8   </fecha-de-nacimiento>
9   <ciudad>Pamplona</ciudad>
10</persona>
```

Figura 117. Ejemplo básico de documento XML para la representación de un objeto de tipo Persona.

Fuente: Palermo (2018).



Licenciatura en Informática Administrativa 2022A

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CatalogoLibros xmlns="http://perso.wanadoo.es/kishimaru/Ejemplo">
  <libro isbn="84-473-0224-5">
    <titulo>Los dragones del Edén</titulo>
    <autores>
      <autor>Carl Sagan</autor>
      <traductor>Jaume Josa Llorca</traductor>
    </autores>
    <editorial>RBA</editorial>
    <materia>Divulgación</materia>
    <resumen>Relato de los hechos fundamentales en el desarrollo de la vida
      inteligente sobre la Tierra, en el que se analiza desde la cantidad
      de información contenida en un cromosoma, hasta posibles líneas de
      evolución del cerebro humano.
    </resumen>
  </libro>
</CatalogoLibros>
```

Figura 118. Ejemplo de documento XML
Fuente: Peis (2008)

REFERENCIAS

- IBM. (2013). IBM docs. Información general sobre ODBC, JDBC y OLE D.
<https://ibm.co/3yvzwOP>
- IBM Corporation (2014). Mandato Update Database Manager Configuration.
<https://ibm.co/3Nays6T>
- IBM .(2021) Lenguaje de consulta estructurada (SQL) <https://ibm.co/3FWRNq6>
- Jayaram, P. (2019). Uso de SQL. <https://bit.ly/3PtIKX>
- López, J. (2022). Curso de SQL. SQL. <https://bit.ly/3vOI9B7>
- Mendoza, J. (2006). SQL Basic. SQL Basic. <https://bit.ly/3KFk2dH>
- Moreno, G. (2005). Consultas SQL. <https://bit.ly/3y5sEr8>
- Mozilla. (2020). Trabajando con JSON - Aprende sobre desarrollo web | MDN.
<https://mzl.la/3svMVm2>
- Mozilla. (2021). Introducción a XML - XML: Extensible Markup Language | MDN.
<https://mzl.la/3MkIGTi>
- Nevárez, B. (1998). Análisis de tecnologías de bases de datos en el World Wide Web. Repositorio del Instituto Tecnológico y de Estudios Superiores de Monterrey. Tesis de Maestría. <https://bit.ly/3MadHbl>
- Palermo, R. (2018), Ejemplo básico de documento XML para la representación de un objeto de tipo, ResearchGate, <https://bit.ly/3I2uvVX>



Licenciatura en Informática Administrativa 2022A

Peis, E. (2008), Ejemplo de documento XML, ResearchGate, <https://bit.ly/3yAqfvA>

Ramos, P.(2018), ¿Qué es SQL? <https://bit.ly/38rcjTy>

Vázquez, R. (2013). Revisión de la instalación y configuración de un DBM. Las bases de datos empleadas en DBM. <https://bit.ly/3L7YAhi>



Licenciatura en Informática Administrativa 2022A

UNIDAD V. INTEGRACIÓN DE TECNOLOGÍAS Y ENTORNOS DE DESARROLLO WEB

OBJETIVO:

Integrar todas las tecnologías web para la propuesta de soluciones desarrolladas en este entorno.

5.1 Entornos preconfigurados para el desarrollo web.

¿Qué es un entorno de Desarrollo?

Dharma Web Studio (2018) menciona que igual a un servidor web, con ciertos recursos asignados, software y librerías instalados, y una configuración determinada, podríamos decir que un servidor es una aplicación que es capaz de atender las peticiones generadas por un cliente (en la mayoría de los casos se puede entender esto como un navegador web como Firefox, Google Chrome, Safari, u otro) y devolverle una respuesta en concordancia.

Dicho claramente, un servidor es el entorno de desarrollo que un programador de aplicaciones web requiere de forma obligatoria y que le permite escribir su código, diseñar y maquetar una interfaces, generar y consultar resultados de diversa índole, y servir o visualizar los cambios.

Un entorno de desarrollo en su versión más básica se suele conformar de:

1. Un servidor web, que usualmente es Apache o Nginx.
2. Librerías de PHP y módulos necesarios, que es el lenguaje de programación utilizado para ejecutar Wordpress y OpenCart.
3. Una base de datos, que suele ser MySQL pero que podría ser cualquier otra.

Sus características técnicas y los softwares, librerías y versiones de estas, que utilizamos para estructurar un entorno, pueden variar en función de las características de cada proyecto.

Esto nos permite desarrollar aplicaciones web de una forma segura, libre de errores, con entornos diferenciados para realizar la programación, realizar pruebas, compartir resultados con los clientes y permitirles realizar pruebas y prácticas, y finalmente publicar una aplicación al alcance de cualquier usuario de internet, libre de errores, robusta, estable.



Licenciatura en Informática Administrativa 2022A

Niveles de un entorno de desarrollo.

Es útil y adecuado definir diferentes “niveles de entorno” o servidores utilizados en el desarrollo de una aplicación web. Cada uno de ellos son utilizados con un fin específico, y presentan ciertas ventajas sobre los otros en determinado momento del proceso de trabajo.

1. Servidor de desarrollo:

El desarrollador prueba el código y comprueba si la aplicación se ejecuta correctamente con ese código. Una vez que la implementación ha sido probada y el desarrollador considera que el código trabaja de forma correcta, la aplicación se mueve entonces al servidor intermedio de montaje. Su principal ventaja es que los ficheros de programación no requieren ser subidos a un servidor en la nube; sino que son manipulados directamente en su entorno. De esta forma, consultar los cambios realizados en pantalla, o las operaciones que realiza determinadas líneas de código, pueden realizarse de forma rápida.

2. Entorno o servicio de pruebas:

Permite minimizar incidencias en etapas posteriores; y que la aplicación no sufra ninguna fricción al ser desplegada en su entorno final. La principal ventaja es la posibilidad de permitir a los administradores de la aplicación, a los clientes, u otros miembros del equipo a interactuar con está en etapas tempranas del desarrollo a fin de conocer el comportamiento de la aplicación, formación, y realizar pruebas y testing por el equipo de pruebas. De esta forma tener una visión acerca de si la aplicación que se está desarrollando, se encuentra libre de errores y si cumple con las necesidades planteadas para el proyecto.

El servidor de desarrollo no solo permite tener un entorno en el cual otros miembros del equipo pueden probar la aplicación, también permite consultar el comportamiento de la aplicación.

3. Entorno o servidor de Pruebas:

Un entorno de pruebas suele estar ubicado en un servidor en la nube y debería tener características similares al entorno local y al resto de entornos a ser utilizados en el ciclo de trabajo. Permite minimizar incidencias en etapas posteriores; y que la aplicación no sufra ninguna fricción al ser desplegada en su entorno final.

La principal ventaja es la posibilidad de permitir a los administradores de la aplicación, a los clientes, u otros miembros del equipo a interactuar con está en etapas tempranas del desarrollo a fin de conocer el comportamiento de la aplicación, formación, y realizar pruebas por el equipo de pruebas.

4. Entorno o servidor de pre-producción:

También llamado entorno de integración o staging, posee una configuración técnica idéntica a la que nos encontraremos en el entorno de producción. El propósito principal de este



Licenciatura en Informática Administrativa 2022A

entorno es emular al entorno de producción con el fin de testear las actualizaciones y asegurar que las mismas no corromperá la aplicación en los servidores en producción cuando estas sean desplegadas.

5. Entorno o servidor de producción:

Se ejecuta la aplicación donde acceden los usuarios finales y donde se trabaja con los datos de negocio en sí mismos. Es un servidor que posee las mismas características y configuración que tendrá el servidor de pre-producción; aunque pudiera en este caso estar configurado por más de un servidor, para efectos de balanceo de carga en aplicaciones que requieren una infraestructura en capacidad de manejar un tráfico de usuarios pesado y miles de conexiones concurrentes.

Movimientos de la aplicación entre los niveles de entornos.

Lo habitual para desplazar el código de las aplicaciones, entre los diferentes niveles de entornos de desarrollo, es utilizar sistemas de integración continua basados en GIT. Git es un sistema de subversiones de código. Una vez configurado e implementado en el entorno de desarrollo, permite rastrear todos y cada uno de los cambios que se realizan en el código, y definir puntos de anclaje en esos cambios, los cuales se denominan "commits". Por otro lado, un sistema de integración continua permite entre otras cosas, facilitar el despliegue de los cambios en cada entorno, desde un repositorio GIT, de forma ordenada y predecible.

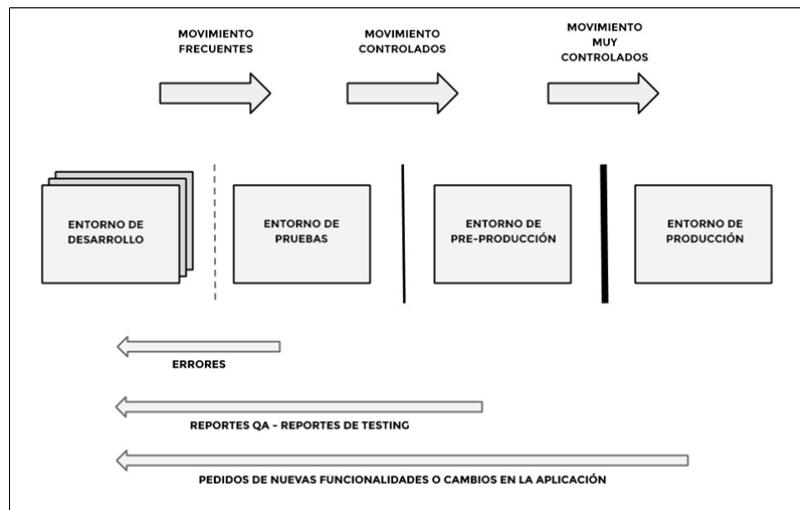


Figura 119. Principios de un entorno de desarrollo

Fuente: Dharma Web Studio (2018)



Licenciatura en Informática Administrativa 2022A

Lo habitual es que los desarrolladores de aplicaciones web escribamos código en el entorno de desarrollo y desplazamos ese código mediante "commits" a entornos de pruebas (o de pre-producción, cuando el entorno de pruebas sea innecesario o excesivo para la aplicación que estamos trabajando). El entorno de pruebas permite realizar pruebas avanzadas en la aplicación y también a otros miembros de equipo, realizar pruebas.

¿ Cómo crear un entorno de desarrollo web virtual y un servidor?

Windows y Mac OSX han progresado como sistemas operativos amigables para el desarrollador durante los últimos años, pero cada desarrollador web realmente necesita trabajar en el entorno nativo de la web, Linux.

Gran parte de la web funciona con lo que los desarrolladores llaman "Pila LAMP". Este acrónimo significa Linux, Apache, MySQL, PHP agrupados y funcionando como lo que normalmente entiende como un servidor web. Las otras tecnologías, Apache (o Nginx), MySQL y PHP, tienen diferentes herramientas e implementaciones disponibles para las diferentes plataformas, pero por razones históricas y técnicas, las configuraciones basadas en Linux dominan el espacio del servidor y tienden a ser más predecibles.

Las máquinas virtuales son programas que se ejecutan en su sistema operativo principal. (C. Alex, 2019).

De manera efectiva, le permiten ejecutar un sistema operativo completamente separado dentro de una ventana, totalmente contenido y aparte de cualquier otra cosa que esté sucediendo con su máquina física, pero a costa de algunos gastos generales.

Una herramienta llamada Vagrant te ayudará a implementar "cajas base" que contienen sistemas operativos Linux preconfigurados en una variedad de sabores.

Una herramienta llamada PuPHPet condensa y GUI-une el proceso de selección y configuración del software comúnmente utilizado en LAMP, e incluso va más allá al agregar soporte para lenguajes del lado del servidor como Ruby y Node.js

Node.js es una forma de ejecutar JavaScript en el servidor; PuPHPet hace archivos de configuración base para Vagrant.

El asistente incluye más de veinte dimensiones diferentes para configurar, algunas de las más destacadas son:

- Destino de despliegue: Aquí puede elegir si desea crear una imagen adecuada para VirtualBox, VMWare y similares, o una que sea adecuada para la infraestructura de computación en la nube como AWS o Digital Ocean, entre otras.



Licenciatura en Informática Administrativa 2022A

- Sistema – Paquetes: Aquí puede incluir cualquier software que instalaría como lo haría en una instalación normal. En particular, debe incluir los paquetes que utilice para el desarrollo que se incluyen en la base de la distribución del sistema operativo.
- Para incluir dependencias de desarrollo para Ubuntu, especifique elementos esenciales de construcción
- Para incluir lo mismo para CentOS 7, especifique "Herramientas de desarrollo"
- Servidores web: Elija Apache o Nginx para formar la columna vertebral de su pila L (A | E) MP.
- Idiomas: PHP, Ruby, Node.js, Python.

Bases de datos: Quizás una de las características más atractivas de este enfoque es la capacidad de construir máquinas virtuales para jugar con las diferentes variedades de bases de datos disponibles. Si bien MySQL es un valor predeterminado, es posible que desee jugar con algo más nuevo como el software de código abierto MariaDB y Forking: The Good, The Great and The Ugly Software de código abierto y Forking: The Good, The Great and The Ugly A veces, el usuario final Beneficios en gran medida de las horquillas. A veces, la bifurcación se hace bajo un velo de ira, odio y animosidad.

El resto de las opciones son un tanto exóticas, y si no sabe cuáles son, siempre se pueden instalar más tarde. Lo que tienes, después de todo, es un sistema operativo completo a tus órdenes.

PuPHPet, al final de este viaje de configuración, producirá un archivo. Descomprima eso en el directorio que creó antes de configurar los parámetros de su nuevo servidor.

Ahora, ejecuta lo siguiente:

\$ vagabundo hasta

Y observa los resultados:



Licenciatura en Informática Administrativa 2022A

```
Bringing machine 'fulcrum' up with 'virtualbox' provider...
=> fulcrum: Box 'puphpet/ubuntu1604-x64' could not be found. Attempting to find and install...
    fulcrum: Box Provider: virtualbox
    fulcrum: Box Version: >= 0
=> fulcrum: Loading metadata for box 'puphpet/ubuntu1604-x64'
    fulcrum: URL: https://atlas.hashicorp.com/puphpet/ubuntu1604-x64
=> fulcrum: Adding box 'puphpet/ubuntu1604-x64' (v20161102) for provider: virtualbox
    fulcrum: Downloading: https://atlas.hashicorp.com/puphpet/boxes/ubuntu1604-x64/versions/20161102/providers/virtualbox.box
    fulcrum: Progress: 100% (Rate: 4882k/s, Estimated time remaining: -: -: -)
=> fulcrum: Successfully added box 'puphpet/ubuntu1604-x64' (v20161102) for 'virtualbox'!
=> fulcrum: Importing base box 'puphpet/ubuntu1604-x64'...
=> fulcrum: Matching MAC address for NAT networking...
=> fulcrum: Checking if box 'puphpet/ubuntu1604-x64' is up to date...
=> fulcrum: Setting the name of the VM: Working_fulcrum_1485568363945_26448
=> fulcrum: Clearing any previously set network interfaces...
=> fulcrum: Preparing network interfaces based on configuration...
    fulcrum: Adapter 1: nat
    fulcrum: Adapter 2: hostonly
=> fulcrum: Forwarding ports...
    fulcrum: 22 (guest) => 7905 (host) (adapter 1)
    fulcrum: 22 (guest) => 2222 (host) (adapter 1)
=> fulcrum: Running 'pre-boot' VM customizations...
=> fulcrum: Booting VM...
```

Figura 120. Resultado de la creación de un entorno de desarrollo web

Fuente: Ephesos Software

Como no tiene el cuadro base en su directorio local, vagrant recuperará la imagen del Atlas, un repositorio de cuadros de vagrant preconfigurados.

En este punto, su máquina virtual se inicia y está básicamente en línea. Sólo queda una cosa: emitir el comando \$ vagrant ssh para pasar a una sesión SSH adecuada con su VM (sin cabeza) que actúa como servidor LAMP.

5.2 Frameworks de desarrollo y estructura del proyecto.

Según Nicolás (2019) Un frameworks de desarrollo es una herramienta que pretende facilitar la creación de aplicaciones web o móviles. Esto se logra de diferentes formas. Al usar un framework, se tiene como base un esqueleto que el desarrollador debe seguir. Este esqueleto incluye, entre otras cosas, archivos de configuración básica que siempre son iguales, y componentes previamente creados por los desarrolladores de la herramienta con el fin de que sean reutilizados. En otras palabras, es una receta básica de lo que debe tener un proyecto y que incluye ingredientes de todo tipo que el desarrollador puede elegir usar o no, dependiendo de sus necesidades.

Los frameworks más comunes y usados son los de desarrollo web, los cuales empezaron a tener acogida a principios del 2010 y han venido mejorando con el tiempo. Entre los más usados están React, desarrollado por Facebook y Angular, desarrollado por Google.



Licenciatura en Informática Administrativa 2022A

Tipos de Frameworks

Ionic es un framework que utiliza Córdoba para encargarse de la parte móvil y Angular para encargarse de la parte web. Aunque la mayoría del código se escribe en TypeScript y HTML, que es la base de Angular, es Cordova el que se encarga de crear el marco de ejecución para las aplicaciones en los dispositivos.

React-Native es un framework desarrollado por Facebook y tiene como fundamentos el framework de desarrollo web React. El equipo que desarrollo React Native decidió basarse en React por la facilidad y buen acogimiento que ha tenido entre la comunidad de desarrolladores. El desarrollo de una aplicación de React-Native y React es prácticamente el mismo. Hay dos cambios importantes. Primero, no se usa el lenguaje de React jsx como tal, sino una adaptación de este a lo que necesita típicamente un dispositivo y, además. Es necesario lidiar con particularidades típicas de un dispositivo como permisos que se le soliciten al usuario.

Flutter es el framework que cambio todo para un desarrollador que tenía un antecedente web, migrar a móvil por medio de estos framework, la forma en la que se escribe una aplicación de Flutter es muy diferente. Los ingenieros de Google decidieron crear una herramienta que funciona a base de "Widgets", los cuales son componentes gráficos que vienen incluidos con Flutter, y son escogidos basado en lo que típicamente tiene una aplicación móvil. Por ejemplo, si se necesita un botón flotante, hay un widget para eso en Flutter. No hay que escribirlo y posicionarlo como se hace normalmente. La arquitectura de Flutter también es algo nuevo, pues ellos se inventaron una llamada BloC, que traduce por sus siglas en ingles Business Logic Component y consiste en tener dos capas, una que es puramente el UI de la aplicación y otra que se encarga de la lógica y se comunican por medio de streams. Es decir, suponiendo que existe un botón que realiza una acción X, en vez de realizar esta acción y cambiar la interfaz directamente, esta manda un streams a la lógica y este devuelve una acción que debe ejecutar la interfaz si es necesario.

Ejemplo de Diseño de la aplicación

La aplicación que se decidió hacer es una típica aplicación de To-Do, donde básicamente se pueden agregar tareas y tienen dos estados, completada o no completada.

Las dos imágenes son las principales vistas de la aplicación, en donde se muestran los dos estados en los que puede estar, 'To-DO' son las tareas que se agregaron, pero aún no se han completado, y "Done" las tareas que ya se completaron. Además, el botón flotante sirve para agregar una tarea nueva a la aplicación.



Licenciatura en Informática Administrativa 2022A

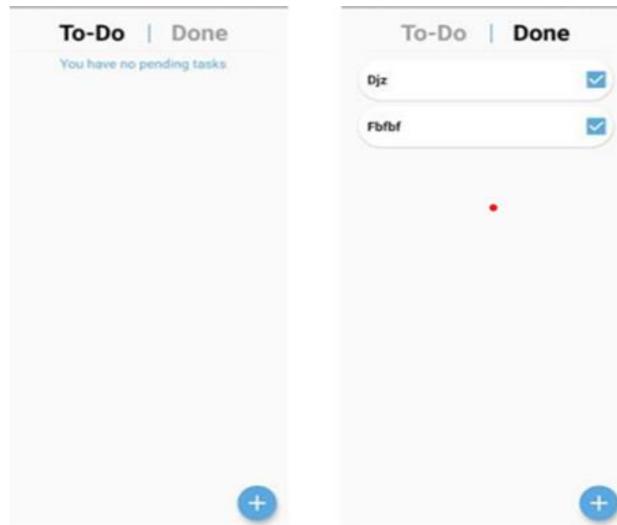


Figura 121. Pantalla inicial

Fuente: Nicolás (2019)

Funcionalidades

Las funcionalidades que ofrece la aplicación son las siguientes:

1. Agregar una tarea
 - Agregar descripción tarea
 - Agregar imagen, cámara o galería
 - Agregar fecha
 - Agregar ubicación (Se agrega automáticamente con la ubicación actual)
2. Eliminar una tarea
3. Cambiar estado de "To-Do" A "Done" y viceversa
4. Intercambiar posición de las tareas
5. Ver información de la tarea
6. Notificaciones push

Licenciatura en Informática Administrativa 2022A

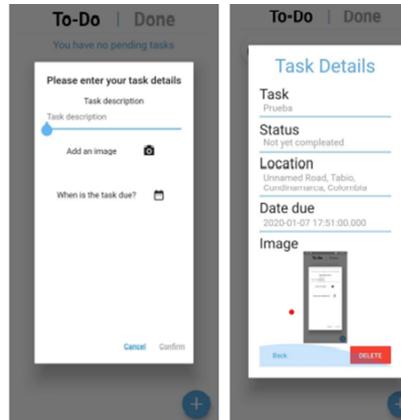


Figura 122. Funciones

Fuente: Nicolás (2019)

La imagen de la izquierda representa el modal que aparece cuando el usuario quiere agregar una tarea nueva. Como se ve, se puede agregar una descripción, una imagen ya sea de la cámara o de la galería de fotos, y una fecha. Por otro lado, en la imagen de la derecha está el modal que se muestra cuando el usuario hace clic sobre una tarea ya existente y muestra la información de esta. Esta información es: la descripción, el estatus de la tarea, la cual puede ser completada o no completada, la ubicación de la persona cuando creó la tarea, la fecha límite de la tarea, la imagen que se escogió y unas acciones, que pueden ser ir hacia atrás o eliminar la tarea. Además, como parte de las funcionalidades hay un sistema de notificaciones push, que se va a explicar en detalle más adelante, pero cuyo funcionamiento se puede ver en la figura 123.

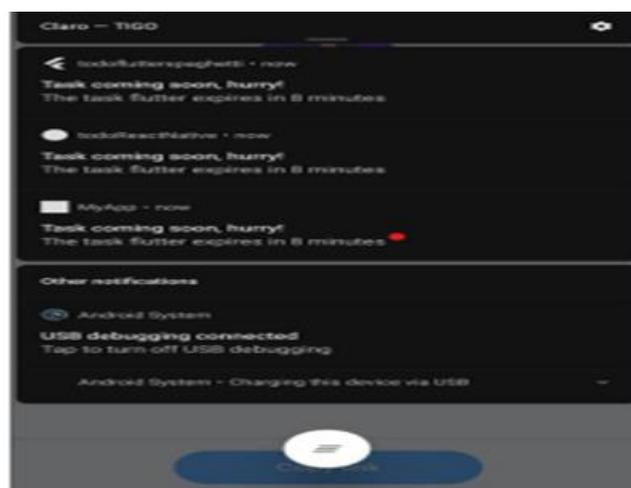


Figura 123. Notificación de Push

Fuente: Nicolás (2019)



Licenciatura en Informática Administrativa 2022A

Descripción Backend Para esta aplicación se decidió hacer un solo backend que las tres aplicaciones pudieran compartir. Esto se decidió así ya que lo importante era analizar el cada framework como tal, y no una aplicación robusta. Además, en el proceso de desarrollo se evidencio que esta decisión ayudo a ver la reactividad de las aplicaciones al agregar una tarea a la vez, y ver si automáticamente la cargaba en otra o era necesario volver a lanzarla. Dicho esto, el backend esta creado con Node.JS, y MongoDB. En términos generales permite hacer un CRUD sobre los datos guardados. A continuación, se presentan los components.

```
const dataSchema = new mongoose.Schema(  
  {  
    task: String,  
    active: Boolean,  
    picture: String,  
    date: String,  
    location: String,  
    position: Number,  
  },  
  { timestamps: true },  
);
```

Figura 124. Esquema base de datos

Fuente: Nicolás (2019)

En la imagen anterior se presenta el esquema de datos que se definió. Como es de esperarse incluyen todos los datos que la aplicación permite guardar además de un timestamp que genera mongo automáticamente para saber la última fecha de modificación.

```
router.get('/data', dataController.index);  
router.get('/data/:id', dataController.show);  
router.post('/data', dataController.create);  
router.put('/data/:id', dataController.update);  
router.delete('/data/:id', dataController.delete);
```

Figura 125. Operaciones Posibles

Fuente: Nicolás (2019)

Esta imagen representa las definiciones de las rutas de las operaciones CRUD que se pueden hacer sobre la base de datos. Como se puede ver, se puede hacer un get de todas



Licenciatura en Informática Administrativa 2022A

las tareas, un get por id, un post para crear una tarea, un put para modificar una tarea dado un id y un delete para eliminar una tarea dado un id.

Descripción Firebase API

El API Cloud Functions de Firebase, es una herramienta muy poderosa que permite ejecutar una función o funciones, generalmente escritas en JavaScript, cada cierto tiempo determinado. Hay dos formas de crearlas. La primera es generarla directamente desde la consola de Google donde uno escribe la función que se quiere ejecutar y el intervalo de tiempo que se desee. La segunda, y es la que se implementó en este proyecto, es por medio del CLI de Firebase. En este caso, se instala este cliente en la máquina que el usuario desee y se escribe un programa que se quiere ejecutar cada cierto tiempo. La ventaja de esta alternativa es que permite generar un set de instrucciones más robusto que por medio de la consola de Google.

Cloud Messaging El API Cloud Messaging es otra herramienta que proporciona Firebase la cual permite mandar notificaciones push a todo tipo de aplicaciones y segmentarlas por diferentes reglas predefinidas para apuntar a cierto tipo de usuarios. En este caso, se creó un servicio de Cloud Messaging por cada aplicación y este es el segmento al que el API apunta.

Integración Lo que se hizo en el proyecto fue integrar estas dos APIS para poder generar un servicio en el que cada cierto tiempo, si se cumplen ciertas condiciones, se mande una notificación push al dispositivo. Como demostración, se muestra a continuación un fragmento de código de este servicio hecho para la aplicación de Flutter.

```
exports.taskDue = functions.pubsub.schedule("*/5 * * * *").onRun(() => {
  gettoken()
  .then(token => {
    const headers = {
      Authorization: "Bearer " + token
    };
  });
});
```

Figura 126. Declaración de cloud

Fuente: Nicolás(2019)

En la imagen anterior se puede apreciar la creación de la función cloud. En este caso, se ejecuta cada 15 minutos y este tiempo se define en formato CRON. Luego se pide el token y se crea el header que se va a mandar posteriormente en la notificación push.



Licenciatura en Informática Administrativa 2022A

```
const dataAfter = {
  message: {
    topic: "push",
    notification: {
      title: "Task Due Date Passed",
      body: `The task ${task} is past it's due date`
    }
  }
};
axios.post(
  "https://fcm.googleapis.com/v1/projects/flutterpaghetti-cd14f/messages:send",
  dataAfter,
  {
    headers: headers
  }
)
```

Figura 127. Mandar llamar Notificación

Fuente: Nicolás (2019)

En el cuerpo de la función se hace un get de toda la base de datos y se calcula la fecha guardada y la fecha actual. En caso de cumplirse que falten 15 minutos o menos, o se haya vencido la tarea, se procede a enviar un mensaje push al dispositivo. Este proceso es el que se ejemplifica, Se crea el cuerpo del mensaje como un objeto y luego se hace una petición POST al API de Cloud Messaging, apuntando a la aplicación que se desee, que en este caso es la de Flutter.

Análisis por framework (Flutter)

Estructura del proyecto En la imagen de la derecha se puede ver la estructura general del proyecto de Flutter. Lo importante a analizar es lo que se encuentra dentro de la carpeta "lib" pues es donde el desarrollador escribe todo el código de la aplicación. El resto es lo que Flutter genera automáticamente y que anteriormente se mencionó como el esqueleto o "receta". Dentro de la carpeta lib hay dos archivos: main.dart y listComponent.dart. El primero es donde esta lo principal de la aplicación y el segundo archivo es widget personalizado de una sola tarea. Este widget se renderiza por cada tarea que se encuentre en la base de datos y se pasan por parámetro la información de la tarea.



Licenciatura en Informática Administrativa 2022A

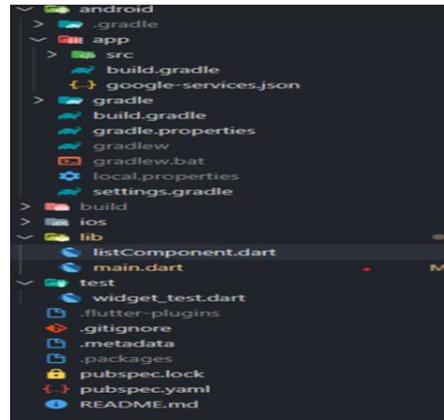


Figura 128. Estructura Proyecto Flutter

Fuente: Nicolás (2019)

Librerías externas usadas por defecto Flutter trae muchas librerías que el desarrollador puede usar, pero en algunos casos es necesario instalar algunas externas para lograr ciertas funcionalidades. A continuación, se describen este tipo de librerías.

```
geolocator: ^5.1.4+1
geocoder: ^0.2.1
image_picker: ^0.6.0+9
firebase_core:
firebase_messaging:
flutter_datetime_picker: ^1.2.8
```

Figura 129 . Librerías

Fuente: Nicolás (2019)

Geolocator: Esta librería es usada para saber la ubicación actual del dispositivo, siempre y cuando haya conexión a internet. Esta ubicación se da como un arreglo [latitud, longitud].



Licenciatura en Informática Administrativa 2022A

```
_getCurrentLocation() async {  
  final Geolocator geolocator = Geolocator().forceAndroidLocationManager;  
  
  geolocator  
    .getCurrentPosition(desiredAccuracy: LocationAccuracy.best)  
    .then((Position position) {  
      _getLocationName(position);  
    }).catchError((e) {  
      throw e;  
    });  
}
```

Figura 130. Uso de Geolocator

Fuente: Nicolás (2019)

Geocoder: Por medio de esta librería se puede ingresar las coordenadas previamente extraídas y convertirlas a una dirección real, por ejemplo, si se ingresan las coordenadas [4.602835, -74.065039], por medio del Geocoder se obtiene la dirección Cra. 1 #18a 12, Bogotá, Cundinamarca, Colombia.

```
_getLocationName(Position pos) async {  
  final coordinates = new Coordinates(pos.latitude, pos.longitude);  
  var addresses =  
    await Geocoder.google('API-KEY-GOOGLE')  
      .findAddressesFromCoordinates(coordinates);  
  var first = addresses.first;  
  setState(() {  
    location = "${first.addressLine}";  
  });  
}
```

Figura 131 . Uso de Geolocator

Fuente: Nicolás (2019)

Image_picker: Esta librería permite otorgar acceso a ver archivos guardados en el dispositivo y a la cámara. Entonces, es la que permite seleccionar una imagen guardada en el dispositivo o tomar una foto para agregarla a una tarea.



Licenciatura en Informática Administrativa 2022A

```
picker(bool action, BuildContext context) async {  
  File img = action  
    ? await ImagePicker.pickImage(source: ImageSource.camera)  
    : await ImagePicker.pickImage(source: ImageSource.gallery);  
  if (img != null) {  
    Navigator.of(context).pop();  
    setState(() {  
      _image = img.path;  
    });  
  }  
}
```

Figura 132. Uso de imagen

Fuente: Nicolás (2019)

firebase_core y firebase_messaging: Estas dos librerías permiten hacer la conexión a los servicios de Firebase y en este caso, recibir notificaciones push.

```
firebaseMessaging.requestNotificationPermissions(  
  const IosNotificationSettings(sound: true, alert: true, badge: true));  
firebaseMessaging.onIosSettingsRegistered  
  .listen((IosNotificationSettings settings) {  
    debugPrint('ios');  
  });  
  
firebaseMessaging.getToken().then((onValue) {  
  debugPrint(onValue);  
});  
  
firebaseMessaging.subscribeToTopic('push');  
}
```

Figura 133. Uso de Firebase

Fuente: Nicolás (2019)

Flutter_datetime_picker: Por medio de esta librería se puede declarar un widget que da la opción al usuario de seleccionar una fecha y hora.

Licenciatura en Informática Administrativa 2022A

```
_selectDate() {  
  DatePicker.showDateTimePicker(  
    this.context,  
    showTitleActions: true,  
    minTime: DateTime.now(),  
    onChanged: (e) {  
      setState(() {  
        date = e.toString();  
      });  
    },  
    onConfirm: (e) {  
      setState(() {  
        date = e.toString();  
      });  
    },  
    currentTime: DateTime.now(),  
  );  
}
```

Figura 134. Uso de Futtler

Fuente: Nicolás (2019)

Análisis

Facilidad de desarrollo Desarrollar una aplicación en Flutter es relativamente sencillo. La forma en la que se crea la interfaz gráfica es intuitiva, pues está basado en lo que el equipo de desarrollo de Google llama árbol de widgets. Esto significa que se van anidando widgets para ir construyendo un elemento. Por ejemplo, si se quiere construir un contenedor que tenga una imagen y un botón, primero se declara un widget Container, y en los hijos de ese container se declara al mismo nivel de anidación el widget Text y FlatButton. Cada uno de estos widgets tiene un sinfín de opciones de personalización que van desde estilos hasta el tipo de comportamiento que se quiere. A partir de esto, se pueden crear también widgets personalizados, que es el caso del ListElement.dart. Este es un widget que se creó que tiene por dentro muchos otros widgets, pero se puede reutilizar las veces que quiera, lo que aumenta significativamente la reutilización de código A continuación, se presenta un fragmento de código que muestra un árbol de widgets.

Licenciatura en Informática Administrativa 2022A

```
class ListElementState extends State<ListElement> with WidgetsBindingObserver {  
  @override  
  Widget build(BuildContext ctx) {  
    bool _checked = widget.active;  
    return InkWell(  
      onTap: () => widget.displayInfo(this.widget),  
      child: Container(  
        padding: EdgeInsets.symmetric(horizontal: 20),  
        child: SizedBox(  
          height: 75,  
          child: Card(  
            shape: RoundedRectangleBorder(  
              borderRadius: BorderRadius.circular(100.0),  
            ), // RoundedRectangleBorder  
            elevation: 2,  
            child: Row(  
              mainAxisAlignment: MainAxisAlignment.spaceBetween,  
              children: [
```

Figura 135. Árbol de Widgets

Fuente: Nicolás (2019)

Se ve como se inicia declarando un widget InkWell que da la funcionalidad de reaccionar a clics del usuario, y adentro de ese un Container, y así sucesivamente.

Para el desempeño se van a analizar dos métricas, uso de CPU y uso de memoria (RAM). En cuanto al uso de CPU, se puede ver que es bajo, pues el promedio es de 1.3% y en algunos momentos alcanzo hasta 2.3%.

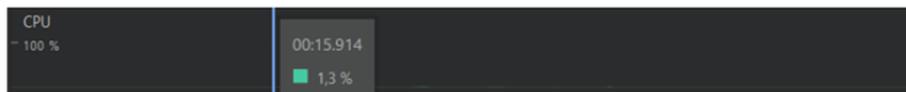


Figura 136 . Uso de CPU

Fuente: Nicolás (2019)



Licenciatura en Informática Administrativa 2022A

React-Native

Estructura del proyecto La estructura del proyecto de React-Native es la típica estructura de un proyecto de React. Se tienen 4 principales carpetas. Componentes, donde hay elementos que se reutilizan varias veces en el código. Containers, que archivos que sirven de contenedores, pero no representan las vistas principales de la aplicación, en este caso, dichos contenedores son el modal donde se muestra la información de una tarea existente y el modal para agregar una tarea nueva. En la carpeta Styles está el archivo de los estilos que se manejan a lo largo de la aplicación. En este caso es un archivo js por que se usan StyledComponents, pero podría ser cualquier archivo de estilos compatibles con React-Native como css o scss. La carpeta utils aloja funciones que manejan la lógica de la aplicación. Por un lado, el archivo api.js funciona como intermediario entre la aplicación y el backend para hacer todas las peticiones y la carpeta Functions, aloja funciones de lógica general como pedir permisos. Por último, el archivo App.js es el núcleo de la aplicación. Ahí se maneja la vista y se renderizan los containers y los componentes necesarios para que funcione correctamente la aplicación, además del manejo del estado.

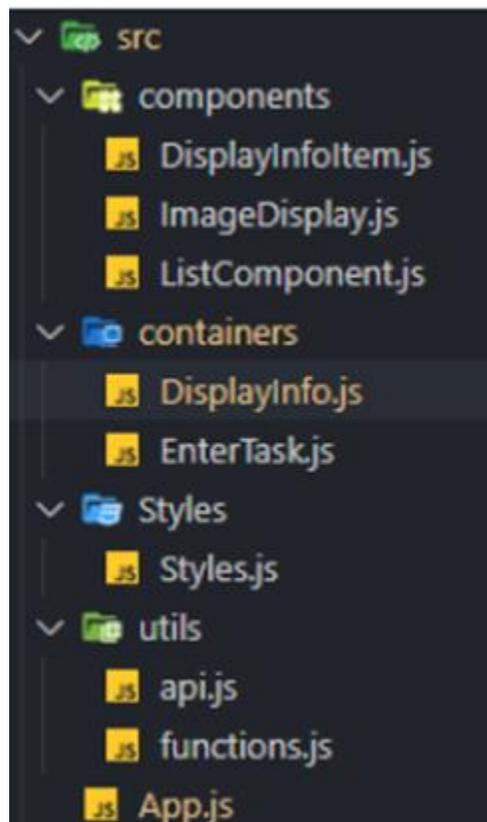


Figura 137 . Estructura del proyecto React.

Fuente: Nicolás (2019)



Licenciatura en Informática Administrativa 2022A

Librerías externas usadas, así como Flutter, React-Native incluye librerías necesarias para que corra la aplicación, así como algunas básicas que el usuario puede elegir usar o no. En el archivo package.json se pueden ver las librerías externas que se tuvieron que agregar.

```
"react-native-elements": "^1.2.7",  
"react-native-geocoding": "^0.4.0",  
"react-native-geolocation-service": "^3.1.0",  
"react-native-image-picker": "^1.1.0",  
"react-native-modal-datetime-picker": "8.x.x",  
"react-native-paper": "^3.1.1"
```

Figura 138. Librerías externas.

Fuente: Nicolás (2019)

Draggable-flatlist: Esta librería sirve para agregar la funcionalidad de intercambiar elementos en una lista.

```
<DraggableFlatList  
  data={tasks2}  
  renderItem={renderItem}  
  keyExtractor={({item}) => `draggable-item-${item.key}`}  
  scrollPercent={5}  
  onMoveEnd={({ data }) => { setTasks2(data); reorder(data); }}  
</>
```

Figura 139. Implementación

Fuente: Nicolás (2019)

Se declara el componente y se le agregan los parámetros necesarios para que la funcionalidad sirva correctamente. El ejemplo total es bastante largo por lo que no se incluye todo el código, pero se puede ver en el proyecto de React-Native en el enlace.

Elementos: Esta librería se usa para agregar componentes visuales como checkbox personalizados. En este caso se usó para agregar el checkbox de cada lista, e iconos de cámara y calendario.



Licenciatura en Informática Administrativa 2022A

```
<CheckBox
  checked={item.active}
  onPress={() => onChangeEvent(item.position)}
  checkedColor={'rgba(92, 168, 224, 1)'}
  checkedIcon="check-square"
  size={35} />
```

Figura 140. Implementación

Fuente: Nicolás (2019)

Geolocation-Service: Esta librería es usada para saber la ubicación actual del dispositivo, siempre y cuando haya conexión a internet. Esta ubicación se da como objeto al que se pueden acceder a muchas propiedades entre las que están la latitud y la longitud.

Geocoding: Por medio de esta librería se puede ingresar las coordenadas previamente extraídas y convertirlas a una dirección real, por ejemplo, si se ingresan las coordenadas [4.602835, -74.065039], por medio del Geocoder se obtiene la dirección Cra. 1 #18a 12, Bogotá, Cundinamarca, Colombia.

```
Geolocation.getCurrentPosition(e => {
  Geocoder.from(e.coords.latitude, e.coords.longitude).then(res => {
    const results = res.results[0].address_components;
    results.forEach(addresses => {
      if (addresses.types.includes('route') && !dir.includes(addresses.long_name)) {
        dir.unshift(addresses.long_name);
      }
      if (addresses.types.includes('street_number') && !dir.includes(addresses.long_name)) {
        dir.push(addresses.long_name);
      }
      if (addresses.types.includes('country') && !dir.includes(addresses.long_name)) {
        dir.push(addresses.long_name);
      }
      if (addresses.types.includes('administrative_area_level_1') && !dir.includes(addresses.long_name)) {
        dir.push(addresses.long_name);
      }
    });
    setLocation(dir);
  });
});
```

Figura 141. implementación services

Fuente: Nicolás (2019)

En la imagen anterior se puede ver el uso de las dos últimas librerías pues se usan en conjunto, primero se le pide a Geolocation que, de la posición actual, y luego a partir de las coordenadas se va construyendo la dirección actual por medio de la respuesta que da Geocoder.

Image-picker: Esta librería permite otorgar acceso a ver archivos guardados en el dispositivo y a la cámara. Entonces, es la que permite seleccionar una imagen guardada en el dispositivo o tomar una foto para agregarla a una tarea

Licenciatura en Informática Administrativa 2022A

```
const chooseImage = () => {  
  requestStorage();  
  const options = {  
    title: 'Please choose between camera or gallery',  
    storageOptions: {  
      skipBackup: true,  
      path: 'images',  
    },  
  };  
  ImagePicker.showImagePicker(options, (response) => {  
    if (!response.didCancel && !response.error) {  
      setImage(response.path);  
    }  
  });  
};
```

Figura 142 . Implementación de imagen

Fuente: Nicolás (2019)

Modal-datetime-picker: Por medio de esta librería se puede declarar un widget que da la opción al usuario de seleccionar una fecha y hora.

```
<DateTimePickerModal  
  isVisible={isDatePickerVisible}  
  mode="datetime"  
  onConfirm={handleConfirm}  
  onCancel={hideDatePicker}  
>
```

Figura 143. Implementación

Fuente: Nicolás (2019)

Paper: Esta librería sirve para agregar diferentes elementos de UI inspirados en material. En este caso se uso para implementar el botón flotante para agregar una tarea.

Facilidad de desarrollo Si se viene de un entorno de desarrollo web, y el desarrollador está familiarizado con React, aprender este framework es relativamente fácil. Sin embargo, partiendo del punto de que no tiene este conocimiento, es de una complicidad media. Empezando por el ciclo de vida de React, es algo a lo que los programadores no están acostumbrados y aprender puede ser un poco tedioso. Además, al ser un lenguaje basado en JavaScript, implica todos los problemas conocidos que vienen con ese lenguaje, aunque se puede mitigar si se decide implementarlo con TypeScript. Por otro lado, crear el markup



Licenciatura en Informática Administrativa 2022A

(jsx), no es algo difícil de aprender, por lo que en un principio se podría empezar construyendo interfaces estáticas e ir agregándole interacciones para que la curva de aprendizaje se haga más fácil, Para el desempeño se van a analizar dos métricas, uso de CPU y uso de memoria (RAM).



Figura 144. Implementación Cpu

Fuente: Nicolás (2019)

Ionic

Estructura del proyecto La estructura del proyecto del proyecto de ionic es extensa pues, como se mencionó anteriormente, al crear componentes genera varios archivos por cada uno. Hay dos componentes, el primero addtask es el que se encarga de agregar una nueva tarea. El rol de cada archivo se explicó Luego esta display-info, que es el componente donde se renderiza la información de la tarea ya creada.

List-element es otro componente y es donde se trabaja todo lo relacionado con una tarea creada. Después sigue home, que no es un componente sino una página, y en este caso donde está la vista principal de la aplicación. Los otros archivos son generados por el framework, pero hay uno muy importante, que es app.module.ts. En este archivo se declaran todas las importaciones de librerías externas.



Licenciatura en Informática Administrativa 2022A

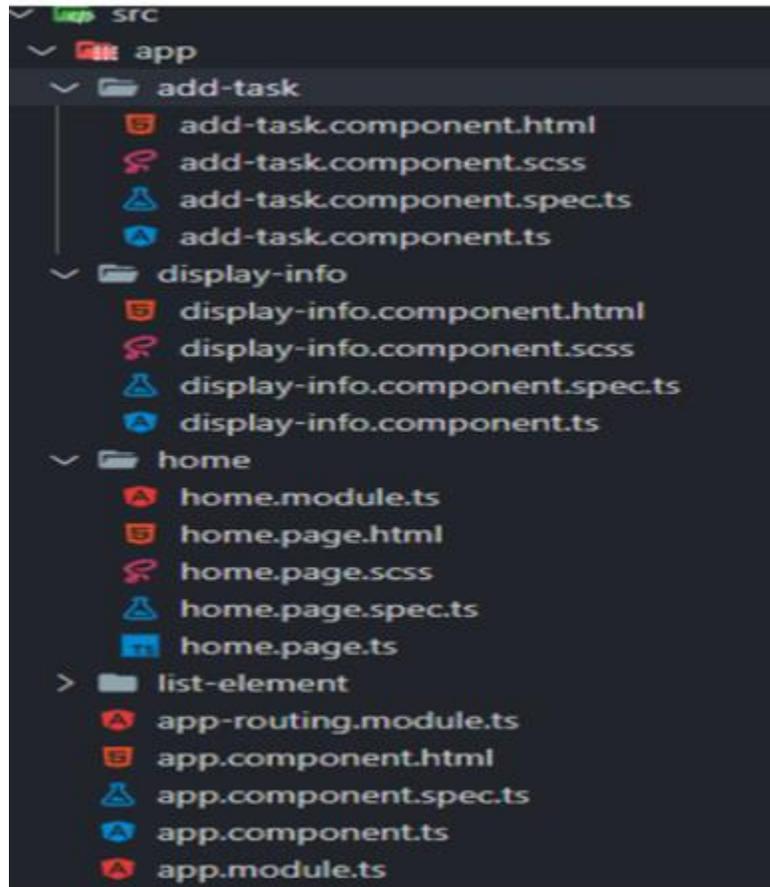


Figura 145. Estructura de Proyecto IONIC

Fuente: Nicolás (2019)

Camera: Esta librería es la que permite seleccionar una imagen guardada en el dispositivo o tomar una foto para agregarla a una tarea.

Licenciatura en Informática Administrativa 2022A

```
openDeviceCamera(option: number) {  
  const options: CameraOptions = { ...  
};  
this.camera.getPicture(options).then(  
  (imageData: string) => {  
    if (imageData.includes('?')) {  
      this.image = imageData.substring(7, imageData.indexOf('?'));  
    } else {  
      this.image = imageData.substring(7);  
    }  
    this.cameraOptionsMenu = false;  
  },  
  err => {  
    console.log(err);  
  }  
);  
}
```

Figura 146. Implementación camer

Fuente: Nicolás (2019)

Date-picker: Esta librería permite otorgar acceso a ver archivos guardados en el dispositivo y a la cámara. Entonces, es la que permite seleccionar una imagen guardada en el dispositivo o tomar una foto para agregarla a una tarea.

```
showDatePicker() {  
  this.datePicker.show({  
    date: new Date(),  
    mode: 'datetime',  
    androidTheme: this.datePicker.ANDROID_THEMES.THEME_HOLO_LIGHT,  
    is24Hour: false,  
    minDate: new Date().getTime()..  
  }).then(  
    dateTime => {  
      this.date = dateTime;  
    },  
    err => console.log('Error occurred while getting dateTime: ', err)  
  );  
}
```

Figura 147. Implementación Date

Fuente: Nicolás (2019)

Fcm: Esta librería permite hacer la conexión a los servicios de Firebase y en este caso, recibir notificaciones push.



Licenciatura en Informática Administrativa 2022A

```
this.fcm.getToken().then(token => {  
  console.log(token);  
});  
this.fcm.subscribeToTopic('push');  
}
```

Figura 148. Implementación Fcm

Fuente: Nicolás (2019)

File: Esta librería permite otorgar acceso a la aplicación a leer y escribir archivos en el dispositivo del usuario. Como tal no hay una implementación en específico, pero a continuación se muestra un ejemplo.

```
await this.file  
  .readAsDataURL(path, imageName)  
  .then(data => {  
    this.imageUrl = data;  
  })  
  .catch(err => {  
    this.imageUrl = undefined;  
  });
```

Figura 149. Implementación File

Fuente: Nicolás (2019)

5.3 Desarrollo de soluciones integradas.

Según, EDS Robotics (2022) menciona que el desarrollo de soluciones integradas son una implementación tecnológica que involucra las operaciones de una industria de manera horizontal, abarcando al nivel estratégico, logístico y de ejecución. Por tanto, estos tipos de sistemas trabajan en coordinación con la maquinaria y el personal, tanto operativo como de administración estratégica, para proporcionar una gestión amplia de los procesos.

Por otro lado, BEETRACK (s.f.) indica que son un conjunto de estrategias que incluyen modernas herramientas digitales que involucran todos los procesos logísticos de una



Licenciatura en Informática Administrativa 2022A

empresa, tanto internos como externos. Es decir, son soluciones integradas porque mediante la tecnología numérica integran todas las actividades asociadas a la gestión logística de una empresa. Y a su vez hacen parte de un grupo más diversificado de alternativas de soluciones digitales enfocadas al desarrollo empresarial.

En algunas aplicaciones avanzadas, una solución integral puede incluso regular las operaciones de la maquinaria según ciertas variaciones de desempeño. Este concepto puede adaptarse a una amplia gama de herramientas, muy valiosas para la implementación de estrategias y proyectos, medición de operatividad, integración de nuevos equipos, monitorización de políticas y la planificación a largo plazo.

Sin embargo, EDS Robotics (2022) menciona que las soluciones integradas pueden llegar a ser muy amplias, debido a las peculiaridades que se presentan en el entorno y es por ello que deben tener en cuenta lo siguiente:

Planeación estratégica: Las soluciones integradas juegan un rol fundamental en cuanto a la ejecución o puesta en marcha de las estrategias, por lo cual, requieren de planes estratégicos previos para poder establecer los estándares de calidad bajo los cuales van a operar. Retomando lo siguiente:

1. Establecimiento de metas
2. Límites de operación segura

Nivel estratégico: El desarrollo de soluciones integradas permiten difundir estrategias claras a todos los departamentos, adecuando el lenguaje y la información a las competencias de cada uno para facilitar la gestión de cada tarea. Para ello, estos sistemas cuentan con un entramado de KPI's (medidores de desempeño o variables de desempeño), acciones a ejecutarse y el personal a cargo de cada tarea.

Nivel logístico: Las operaciones logísticas dependen casi exclusivamente de la planificación y la coordinación de trabajo para que las operaciones de final de línea de producción, almacenamiento y despacho se ejecuten correctamente.

Nivel de ejecución: El desarrollo de soluciones integradas se basan en las funciones colaborativas, ya sea entre dispositivos autónomos con inteligencia artificial o maquinaria-persona para gestionar la puesta en marcha de cualquier proyecto, mientras miden sus niveles de eficiencia.

Por lo tanto, BEETRACK (s.f.) dice que la implementación del desarrollo de soluciones integradas, está asociada directamente a la contratación de operadores logísticos, que ofrecen estos servicios, para delegar en ellos la responsabilidad total o parcial de las operaciones logísticas, dichos operadores ponen al servicio del contratante su infraestructura física, su personal logístico especializado y herramientas de software de control y seguimiento.



Licenciatura en Informática Administrativa 2022A

1. **Los operadores logísticos:** son empresas de soluciones integradas, que funcionan como proveedores que ponen al servicio del contratante, tanto su infraestructura como su personal logístico para la realización total o parcial de las actividades asociadas a la distribución y entrega de los productos al receptor final.
2. **Empresas de desarrollo tecnológico:** posibilitan la transformación digital de las compañías, al proveer soluciones integrales consistentes en sistemas de gestión empresarial, y en general tecnologías de última generación, que optimizan la planeación logística.

Ejemplos de desarrollo de soluciones integradas

Los principales ejemplos del desarrollo de soluciones integradas incluyen, entre otras, las siguientes herramientas:

- **PlannerPro by Beetrack:** software de planeación, diseño y optimización de rutas de despacho.
- **LastMile by Beetrack:** software de monitoreo de entregas en tiempo real, control logístico de la última milla: realizan un seguimiento del producto hasta la puerta del cliente.
- **LastMile Contactless by Beetrack:** software de trazabilidad de entregas, gestión de rutas, visibilidad y control, conjunto de herramientas que permiten conocer la ubicación, la trayectoria y el histórico de un producto.
- **LastMile+Collect:** funcionalidad de aprovechamiento de los camiones de recolección para distribuir ventas con retiro en tienda, sin costo adicional.
- **Soluciones basadas en Big Data:** hacen parte de las soluciones integrales de optimización y predicción.

Ejemplo de empresas que se dedican a el desarrollo de soluciones integradas:

- La empresa Sistemas y Soluciones Integradas SAS, es una empresa que desarrolla soluciones de software innovadoras
- DiverGética es una empresa que se dedica a implementar y desarrollar soluciones tecnológicas.
- BlueStar incorpora soluciones de Software y Middleware al portafolio tradicional de Hardware que permitan a los clientes (VARs) incrementar de manera sustancial su oferta de negocio, permitiéndoles entrar a nuevas oportunidades y aumentar las ventas con sus clientes cautivos.

BlueStar (2022) cuentas con dos herramientas que son las siguientes:

ISV Connect: Que tiene como objetivo registrar cada una de las aplicaciones que nuestros ISVs de la región han desarrollado. Si tu empresa cuenta con alguna aplicación accede a este sitio para registrarla.



Licenciatura en Informática Administrativa 2022A

Appware Solutions: Es un sitio enfocado en presentar soluciones integrales de Software y Hardware clasificados por tipo de aplicación o por mercado vertical. Permite a los interesados en alguna solución en particular encontrarla de manera ágil y obtener la información clave para determinar su funcionalidad. Algunos de los nichos que se pueden encontrar son:

- Punto de Venta
- Almacenes, Inventarios y Activos Fijos
- Servicios en Ruta
- Señalización Digital

REFERENCIAS

Dharma Web Studio (2018) ¿Conoces los entornos de trabajo utilizados en el desarrollo de una aplicación Web? <https://bit.ly/3Pvd1PV>

Arimetrics. (2016). Qué es Entorno de desarrollo. <https://bit.ly/3wrcqal>

BEETRACK. (s.f.). Soluciones Integrales en empresas. ¿Qué son y cómo implementarlas? Powered by Beetrack. <https://bit.ly/39Wh3kh>

BlueStar (2022), Soluciones de Software <https://bit.ly/39ThITw>

DiverGética. (s.f). Divergética Soluciones de alta tecnología. <https://bit.ly/3MPytxC>

EDS Robotics. (2022). ¿Qué son las soluciones integrales y cómo aplicarlas? EDS Robotics. <https://bit.ly/38mIDYA>

C. Alex. (2019). Cómo crear un entorno de desarrollo web virtual y un servidor. Ephesos Software. Disponible en: <https://bit.ly/3NBywgu>

Nicolás, S. (2019) Análisis de frameworks para desarrollo de aplicaciones móviles y web.

<https://bit.ly/3sXNPrT>

Sistemas y Soluciones Integradas (2022). Sistemas y Soluciones Integradas SAS. <https://bit.ly/3GnNT9L>



Licenciatura en Informática Administrativa 2022A

ANEXO I. EJERCICIOS PROPUESTOS

UNIDAD I. INTRODUCCIÓN

CONTEXTUALIZACIÓN: Esta unidad se relaciona con el tema de definiciones de uno de los subtemas de la introducción. El contenido facilitará la comprensión de los conceptos difíciles, así como los conceptos básicos de la programación que integran a la misma.

EVALUACIÓN: Una vez finalizada la lectura de la unidad 1 Introducción la persona quien adquirió la antología programación declarativa discutirá y analizará el tema en el que se encuentra centrado en discusión de las siguientes preguntas como guía de la unidad:

1. Son algunas de las ventajas de la programación declarativa, EXCEPTO:

- A) Código más corto y eficiente
- B) Realizable con métodos no conocidos en el momento de la programación.
- C) Basado en una forma de pensar no habitual en las personas (estado de solución).
- D) Optimización sencilla, ya que la ejecución se gestiona mediante un algoritmo

2. ¿En qué se basa la arquitectura de las aplicaciones web?

- A) Modelo Cliente/Servidor
- B) Modelo TCP/IP
- C) Modelo Relacional

3. El papel de _____ lo desempeña el propio SGBD, el cual se ocupará de todas las funciones correspondientes a la lógica de datos, más las restantes funciones correspondientes a la lógica del negocio, mediante la codificación de Procedimientos Almacenados.

- A) Servidor
- B) Programador



Licenciatura en Informática Administrativa 2022A

4. ¿Son elementos de la interfaz de programación de aplicaciones?

- A) Modelo
- B) Almacenamiento
- C) Controlador
- D) Expandir el alcance de su marca

5. ¿En qué año surgió la arquitectura de las aplicaciones web?

- A) 1980
- B) 1930
- C) 1990

6. ¿Qué elementos básicos componen al código de las páginas web?

- A) HTML, CSS, JAVA Script
- B) C++, Bootstrap, Python
- C) Visual Code, HTML, JAVA

7. Identifica las relaciones de más alto nivel entre diferentes entidades, dicho enfoque pertenece al modelado:

- A) Evaluativo
- B) Conceptual
- C) Analítico
- D) Lógico

8. ¿Cuáles son los tres tipos de modelos de datos?

- A) Conceptual, lógico y físico
- B) Analítico, evaluativo y monitoreado
- C) Conceptual, monitoreado y evaluativo
- D) Analítico, físico y evaluativo

9. ¿Qué es la interfaz de programación de aplicaciones?

- A) Es un patrón de arquitectura de software, que separa los datos.



Licenciatura en Informática Administrativa 2022A

- B) Sirve como pauta para ser imitada, reproducida o copiada.
- C) Persona que merece ser imitada por sus buenas cualidades.
- D) Es un conjunto de herramientas, definiciones y protocolos

10. De los elementos que emplea el Patrón del Modelo Vista-Controlador, el MODELO es el encargado de:

- A) Acceder a la capa de almacenamiento de datos.
- B) Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- C) Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.

RESPUESTAS Y REFLEXIONES DE LOS EJERCICIOS:

En sentido amplio, conocer los conceptos básicos puede ser como un conjunto de elementos interrelacionados entre los mismos temas., conjunto de elementos, que no contengan más que la simple suma de estos mismos, conjunto para el cual es válido el principio de crear nuevas cosas.

CLAVE DE LAS PREGUNTAS

- 1- C
- 2- A
- 3- A
- 4- D
- 5- C
- 6- A
- 7- B
- 8- A
- 9- D
- 10-A



Licenciatura en Informática Administrativa 2022A

UNIDAD II. TECNOLOGÍAS FRONTEND

CONTEXTUALIZACIÓN: Esta unidad se relaciona con el tema de un desarrollador FRONTEND se encarga de ir más allá del diseño gráfico, ya que en su campo de acción también cuida la usabilidad, la legibilidad y el funcionamiento en general desde el punto de vista del cliente, como se muestra en los subtemas de la unidad 2. El contenido facilitará la comprensión de los conceptos difíciles, así como los conceptos básicos de la programación que integran a la misma.

EVALUACIÓN: Una vez finalizada la lectura de la unidad 2 Tecnologías FRONTEND la persona quien adquirió la antología programación declarativa discutirá y analizará el tema en el que se encuentra centrado en discusión de las siguientes preguntas como guía de la unidad:

1. Ocultan detalles y nos permiten hablar sobre problemas en un nivel superior sin describir todos los detalles de implementación del problema. Es el concepto de:

- A) Abstracción
- B) Callback
- C) Reduce
- D) Programación Funcional

2. ¿Cuál es la parte esencial de una etiqueta HTML?

- A) Etiqueta de apertura
- B) Atributos
- C) Fragmentos

3. ¿Qué son las alternativas de desarrollo?

- A) Servidores web
- B) Programas de un correcto desarrollo
- C) Son herramientas que permiten crear y mantener una web con facilidad
- D) Archivos de extensión de una computadora

4. Es el lenguaje que permite describir la presentación de páginas Web.



Licenciatura en Informática Administrativa 2022A

- A) Css
- B) Java
- C) Ubuntu

5. Existe una clasificación de frameworks de CSS que permiten manipular múltiples herramientas para que estas puedan ser editadas en el diseño web, EXEPTO:

- A) Multipropósito
- B) Propósito específico
- C) Propósito único

6. Son funciones que se pasan a otras funciones como argumentos y son llamadas por la función en la que se pasan. Es un concepto de las funciones:

- A) Callback
- B) Asíncronas
- C) De primera clase
- D) De orden superior

7. Es una función que espera una función como argumento (callback) para hacer la llamada de AJAX.

- A) Pure functions
- B) AJAX
- C) getArtists
- D) Referential Transparency

8. ¿Qué conceptos son recomendables conocer antes de intentar aprender sobre los frameworks de JavaScript?

- A) HTML, CSS y JavaScript
- B) Modelos de capa de red
- C) Lenguaje ensamblador

9. ¿Cuáles son los 3 tipos de atributos dependiendo de sus valores?

- A) Conjunto finito de valores. Valores libres. Valores booleanos.



Licenciatura en Informática Administrativa 2022A

B) Atributos de valores, atributos finitos, atributos por etiqueta.

C) Valores libres, Atributos de valores, Valores booleanos

10. Son algunos frameworks de JavaScript:

A) Python, Java, Node.js

B) Css, HTML, C++

C) Node.js, Meteor.js, React.js

RESPUESTAS Y REFLEXIONES DE LOS EJERCICIOS:

En sentido amplio, conocer los elementos básicos puede ser como un conjunto de software interrelacionados entre los mismos lenguajes de programación, conjunto de elementos, que no contengan más que la simple suma de los mismos, conjunto para el cual es válido el principio de crear programas.

CLAVE DE LAS PREGUNTAS

- 1- A
- 2- A
- 3- C
- 4- A
- 5- C
- 6- A
- 7- C
- 8- A
- 9- A
- 10- C



Licenciatura en Informática Administrativa 2022A

UNIDAD III. TECNOLOGÍAS BACK-END)

CONTEXTUALIZACIÓN: Esta unidad se relaciona con el tema de definiciones de uno de los subtemas de las tecnologías como el contenido facilitará la comprensión de los conceptos difíciles, así como los conceptos básicos de la programación que integran a la misma.

EVALUACIÓN: Una vez finalizada la lectura de la unidad 3 Tecnologías BACK-END la persona quien adquirió la antología programación declarativa discutirá y analizará el tema en el que se encuentra centrado en discusión de las siguientes preguntas como guía de la unidad:

1. Java 7, Eclipse Kepler, Jboss 7.0.1 S.E, Servidor (Local o Red). Son:

- A) Instalación de herramientas
- B) Herramientas de desarrollo
- C) Servidor de aplicaciones
- D) Pruebas unitarias del sistema

2. ¿Cómo se debe hacer la estructuración del árbol de desarrollo?

- A) C:\java\Jboss
- B) C:\java
- C) C:\Jboss
- D) C:\java-Jboss

3. Asignar el nombre del sitio, Path de ubicación del sitio, Configuración de DNS o IP (Local o Red), Arrancar sitio. Son opciones para:

- A) Instalación de las herramientas
- B) Configurar el Pool de conexión
- C) Pruebas unitarias del sistema
- D) Asignar sitios a Jboss



Licenciatura en Informática Administrativa 2022A

4. Visualizar el servidor de aplicaciones en ejecución, Verificar el pool de conexiones, Generar un proyecto web-jsp, Correr el servidor de aplicaciones con el proyecto, Visualizar el resultado del servidor. Son o se utilizan para:

- A) Configurar el Pool de conexión
- B) Instalación de las herramientas
- C) Pruebas unitarias del sistema

D) Asignar sitios a Jboss

5. Método para asignar la versión de un documento utilizando dos o tres cifras decimales para indicar la importancia de los cambios realizados.

- A) Versiones X.Y.Z
- B) Versiones constantes
- C) Versiones lineales

6. Se trata de un dispositivo que proporciona los servicios de aplicación a las computadoras cliente. Es un concepto de:

- A) Instalación de herramientas
- B) Herramientas de desarrollo
- C) Servidor de aplicaciones
- D) Pruebas unitarias del sistema

7. El control de código son los métodos y herramientas disponibles para controlar todo lo referente a los cambios en el tiempo de un archivo.

- A) Verdadero
- B) Falso

8. La función principal de los controladores de administración es resolver el problema proporcionando un único lugar para llamar a cada función.

- A) Verdadero
- B) Falso

9. Node.js es un entorno JavaScript basado en eventos junto con la velocidad con la que ejecuta JavaScript es una de sus mayores bondades.

- A) Verdadero



Licenciatura en Informática Administrativa 2022A

B) Falso

10. La Bases de datos relacionales presentan la información organizada en un conjunto de tablas descritas formalmente.

A) Verdadero

B) Falso

RESPUESTAS Y REFLEXIONES DE LOS EJERCICIOS:

En sentido amplio, conocer los elementos básicos puede ser como un conjunto de software interrelacionados entre los mismos métodos de programación, conjunto de lenguajes, que no contengan más que la simple suma de los mismos, conjunto para el cual es válido el principio de crear programas.

CLAVE DE LAS PREGUNTAS

- 1- B
- 2- A
- 3- D
- 4- C
- 5- A
- 6- D
- 7- B
- 8- A
- 9- A
- 10- A



Licenciatura en Informática Administrativa 2022A

UNIDAD IV. TECNOLOGÍAS CONEXIÓN Y TRANSFERENCIA DE DATOS

CONTEXTUALIZACIÓN: Esta unidad se relaciona con el tema de tecnologías conexión y transferencia de datos, El contenido facilitará la comprensión de los conceptos difíciles, así como los conceptos básicos de la programación que integran a la misma.

EVALUACIÓN: Una vez finalizada la lectura de la unidad 4 Tecnologías conexión y Transferencia de datos la persona quien adquirió la antología programación declarativa discutirá y analizará el tema en el que se encuentra centrado en discusión de las siguientes preguntas como guía de la unidad:

1. ¿Qué es el Open Database Connectivity (ODBC)?

- A) Está basado en las especificaciones Call-Level Interface (CLI) de X/Open e ISO/IEC y utiliza SQL como su lenguaje de acceso a bases de datos.
- B) Programación a nivel sistema de Microsoft diseñada para proporcionar acceso a cualquier tipo de dato sin importar su tipo, formato o ubicación.
- C) Consiste en un conjunto de clases e interfaces escritas en el lenguaje de programación Java.

2. ¿Qué es el Java Database Connectivity (JDBC)?

- A) Programación a nivel sistema de Microsoft diseñada para proporcionar acceso a cualquier tipo de dato sin importar su tipo, formato o ubicación.
- B) Consiste en un conjunto de clases e interfaces escritas en el lenguaje de programación Java.
- C) Está basado en las especificaciones Call-Level Interface (CLI) de X/Open e ISO/IEC y utiliza SQL como su lenguaje de acceso a bases de datos.

3. ¿Qué es el OLE DB?

- A) Es un modelo de programación de bases de datos que permite al programador escribir aplicaciones sobre datos OLE DB desde cualquier lenguaje.
- B) Datos consiste de los datos que el usuario desea acesar además del manejador de base de datos, sistema operativo y, en algunos casos, red necesarios para acesar los datos.
- C) Programación a nivel sistema de Microsoft diseñada para proporcionar acceso a cualquier tipo de dato sin importar su tipo, formato o ubicación.



Licenciatura en Informática Administrativa 2022A

4. ¿Es una tarea que realiza un controlador?

- A) Conectarse y desconectarse de la fuente de datos.
- B) Se deberá iniciar una transacción.
- C) Implicar desplegar resultados de una consulta o actualizar el contenido de la base de datos.

5. ¿Cuáles son las características de SQL?

- A) Lenguaje de definición de datos e interactivo de manipulación de datos
- B) Integridad y Definición de vistas
- C) Control de transacciones
- D) Todas las anteriores

6. ¿Qué es el modelo Mecanicista Basado en Datos (DBM)?

- A) Método híbrido que puede emplearse para la predicción de caudales.
- B) Estructura lógica que adopta la base de base datos, incluyendo las relaciones y limitaciones
- C) Modelo que describe aspectos relacionados con las necesidades de una organización para recopilar datos y las relaciones entre estos aspectos.

7. La prestación DBM está compuesta por:

- A) Un archivo de cabecera y una biblioteca.
- B) Datos o claves demasiado extensas
- C) Un datum que es el tipo definido por typedef

8. Prototipos para las principales funciones DBM:

- A) dbm_open, dbm_class, dbm_main y dbm_close
- B) dbm_open, dbm_main, dbm_fecth y dbm_dml
- C) dbm_open, dbm_store, dbm_fecth y dbm_close

9. Abreviación de “World Wide Web Consortium”:

- A) WWWC
- B) W3C



Licenciatura en Informática Administrativa 2022A

C) WOWC

10. Es un formato basado en un subconjunto de la sintaxis de JavaScript.

A) JAVA

B) XML

C) JSON

RESPUESTAS Y REFLEXIONES DE LOS EJERCICIOS:

En sentido amplio, conocer los elementos básicos puede ser como un conjunto de software interrelacionados entre los mismos métodos de programación, conjunto de lenguajes, que no contengan más que la simple suma de estos, conjunto para el cual es válido el principio de crear programas.

CLAVE DE LAS PREGUNTAS

- 1- A
- 2- B
- 3- C
- 4- A
- 5- D
- 6- A
- 7- A
- 8- C
- 9- B
- 10- C



Licenciatura en Informática Administrativa 2022A

UNIDAD V. INTEGRACIÓN DE TECNOLOGÍAS Y ENTORNOS DE
DESARROLLO WEB

CONTEXTUALIZACIÓN: Esta unidad se relaciona con el tema de definiciones de uno de los entornos de desarrollo web, El contenido facilitará la comprensión de los conceptos difíciles, así como los conceptos básicos de la programación que integran a la misma.

EVALUACIÓN: Una vez finalizada la lectura de la unidad 5 Integración de tecnologías y entornos de desarrollo web la persona quien adquirió la antología programación declarativa discutirá y analizará el tema en el que se encuentra centrado en discusión de las siguientes preguntas como guía de la unidad:

1. Es una aplicación que es capaz de atender las peticiones generadas por un cliente:

- A) Servidor
- B) Entorno
- C) Desarrollo web

2. El desarrollador prueba el código y comprueba si la aplicación se ejecuta correctamente con ese código, pertenece al nivel de:

- A) Servidor de producción
- B) Servidor de desarrollo
- C) Servidor de pre-producción

3. Permite minimizar incidencias en etapas posteriores; y que la aplicación no sufra ninguna fricción al ser desplegada en su entorno final pertenece al nivel:

- A) Servicio de pruebas
- B) Servidor de producción
- C) Servidor de desarrollo

4. Es una forma de ejecutar JavaScript en el servidor:

- A) PHP
- B) Mac OSX
- C) Node.js



Licenciatura en Informática Administrativa 2022A

5. Herramienta que ayuda a implementar cajas base que contienen sistemas operativos Linux preconfigurados:

- A) LAMP
- B) Vagrant
- C) MySQL

6. ¿Qué es el desarrollo de soluciones integradas?

- A) Son una implementación tecnológica que involucra las operaciones de una industria de manera horizontal, abarcando al nivel estratégico, logístico y de ejecución.
- B) Funcionan como proveedores que ponen al servicio del contratante, tanto su infraestructura como su personal logístico para la realización total o parcial de las actividades asociadas a la distribución y entrega de los productos al receptor final.
- C) Requieren de planes estratégicos previos para poder establecer los estándares de calidad bajo los cuales van a operar.

7. ¿Qué son los operadores logísticos?

- A) Empresas de soluciones integradas, que funcionan como proveedores que ponen al servicio del contratante, tanto su infraestructura como su personal logístico para la realización total o parcial de las actividades asociadas a la distribución y entrega de los productos al receptor final.
- B) Soluciones integradas se basan en las funciones colaborativas, ya sea entre dispositivos autónomos con inteligencia artificial o maquinaria-persona para gestionar la puesta en marcha de cualquier proyecto, mientras miden sus niveles de eficiencia.
- C) Implementación tecnológica que involucra las operaciones de una industria de manera horizontal, abarcando al nivel estratégico, logístico y de ejecución.

8. PlannerPro by Beetrack es un software de planeación, diseño y optimización de rutas de despacho.

- A) Verdadero
- B) Falso

9. Los desarrollos de soluciones integradas permiten difundir estrategias claras a todos los departamentos, adecuando el lenguaje y la información a las competencias de cada uno para facilitar la gestión de cada tarea.

- A) Verdadero
- B) Falso



Licenciatura en Informática Administrativa 2022A

10. El desarrollo de una aplicación de React-Native y React es el mismo.

- A) Verdadero
- B) Falso

RESPUESTAS Y REFLEXIONES DE LOS EJERCICIOS:

En sentido amplio, conocer los elementos básicos puede ser como un conjunto de software interrelacionados entre los mismos métodos de programación, conjunto de lenguajes, que no contengan más que la simple suma de estos, conjunto para el cual es válido el principio de crear programas en el uso de la web junto con su entorno de desarrollo.

CLAVE DE LAS PREGUNTAS

- 1- A
- 2- B
- 3- A
- 4- C
- 5- B
- 6- A
- 7- A
- 8- A
- 9- A
- 10- A



Licenciatura en Informática Administrativa 2022A

ANEXO II. IDENTIFICACION DE LOS AUTORES



Cristian Barrera Rojas, Estudiante de la Licenciatura en Informática Administrativa del Centro Universitario UAEM Atlacomulco.

Apasionado por las nuevas tecnologías cuenta con certificación en Microsoft Excel.

No le gustan las traiciones.

Es apasionado y visionario.

Lucha por un buen futuro el cual se esfuerza para conseguirlo día con día.

Estudia en la licenciatura de Informática Administrativa por lo que le llama la atención las diferentes áreas que maneja. Se identifica con la programación además le gusta estar en constante actualización.



Oswaldo Mercado Rojas, tengo 19 años de edad, soy originario de Santa Rosa de Guadalupe, El Oro, Edo. De México actualmente soy estudiante de la licenciatura de informática administrativa en la universidad Autónoma de estado de México cursando el cuarto semestre.

Mi formación académica inicia en el preescolar “Boreje Flor Negra, Jardín de niños” en Santa Rosa de Guadalupe a los 5 años de edad, posteriormente la Primaria “Lic. Juan Fernández Albarrán” en el mismo lugar de residencia, terminando acudí a la Secundaria “Escuela Secundaria Agropecuaria Num.0017 Gustavo Vaz Prada” en Santa Rosa de Lima, posteriormente Ingresé a la “Escuela Preparatoria Oficial Num.302” en Santa Rosa de Guadalupe. actualmente participación en un proyecto de investigación. de una antología de las unidades de aprendizaje de la materia de programación declarativa.



Licenciatura en Informática Administrativa 2022A



Itzel Nicolás Flores nació en Atlacomulco de Fabela, Estado de México en el año 2002 desde el comienzo me ha gustado asistir a la escuela y tener buenas notas en todos los niveles de escolaridad primaria, secundaria, preparatoria y actualmente la universidad. Soy una persona con mucho potencial para aprender nuevas cosas y así mismo compartir mis conocimientos con los demás.

Mis primeros estudios los realice en mi pueblo natal, es decir, la primaria la curse en el instituto "Benito Juárez", seguido de ello estude en la secundaria 0358 "Gral. Ignacio Zaragoza" en la cual destaque como el mejor promedio de ese año, posteriormente entre a estudiar una carrera como "Técnico en Informática" en el Centro de Bachillerato Tecnológico CBT Timilpan que de igual forma destaque como el mejor promedio de aprovechamiento de ese año. Actualmente soy estudiante de cuarto semestre de la licenciatura en "Informática Administrativa" en la Universidad Autónoma del Estado de México UAEM Atlacomulco. Durante mi formación profesional he culminado dos cursos en línea de nombre Entrepreneurship y NDG Linux Unhatched del Cisco Networking Academy cuya finalidad es enriquecer más mis conocimientos. De igual forma obtuve un título como "Técnico en Informática" del Centro de Bachillerato Tecnológico. Sin embargo, aún me quedan muchos proyectos por culminar, pero considero que con dedicación, tiempo y potencial podre lograrlos.



Elizabeth Pérez Flores soy estudiante de la Licenciatura en informática Administrativa. Soy egresada de la preparatoria Cecytem San Felipe del progreso como técnica en programación y cuento con una certificación en Microsoft, soy una persona que le gusta superarse y ayudar a los demás, soy divertida y entusiasta. Me gusta mucho viajar y espero en un futuro cumplir cada una de mis metas que me proponga.



Licenciatura en Informática Administrativa 2022A



Luis Felipe Pérez Garduño

27 de febrero de 2002

Técnico en Administración

Desarrollo de Proyecto de Incubadora de Empresas

Seis Meses de Servicio Social en SADER Atlacomulco

Certificación CISCO por el Curso Entrepreneurship



Daniela Ramírez Chimal. Mexicana, nacida el 11 de octubre de 2002 en Temascalcingo, Estado de México. Técnica en Informática desde el 2021, formada por el CBT LIC. Mario Colín Sánchez, Atlacomulco. Actualmente estudiante de cuarto semestre de la Licenciatura en Informática Administrativa en el Centro Universitario UAEM Atlacomulco, la elección de estudiar esta licenciatura es debido a que la misma es muy importante en la formación de profesionales que pueden llevar a cabo la gestión de sistemas de información, la administración de proyectos informáticos, auditar sistemas de gestión en seguridad de la información, crear y asesorar negocios electrónicos entre muchos otros puntos destacables.

Posee gran interés en continuar desarrollando habilidades referentes a la programación y el diseño para poder desempeñarse en el campo del desarrollo web y en base a los conocimientos adquiridos en la licenciatura poder desarrollar estrategias para la optimización de diferentes procesos.

Actualmente su enfoque está en aprender inglés, adquirir nuevos conocimientos y habilidades informáticas para poder obtener certificaciones en un futuro y así tener más oportunidades dentro del área laboral.



Licenciatura en Informática Administrativa 2022A

Emmanuel Ramírez Lucas

Mexicano, nacido en Atlacomulco Estado de México, en el año 2002. Alumno inscrito en el cuarto semestre de la Licenciatura en Informática Administrativa en el Centro Universitario UAEM Atlacomulco.

Cuenta con certificados que acreditan sus conocimientos en materias como, "Instalación y configuraciones básicas del software Linux, comprensión de los aspectos básicos sobre la interfaz de línea de comandos (CLI) de Linux", como de igual manera "Adquisición de confianza y construcción de una buena ambición, determinación que se carece para iniciar distintos tipos de negocio y la mejora de las destrezas de liderazgo como de sus conocimientos financieros".

Por otro parte cuenta con constancias que acreditan sus conocimientos en materias como "Habilidades en el uso de las tecnologías de la información y comunicación, manejo del sistema operativo Windows, procesador de textos, hoja de cálculo, presentaciones electrónicas, internet, redes y programas antivirus" como de igual manera "Haber realizado Prácticas Profesionales en sus tres modalidades: observación, ejecución, y estadía de la carrera de Técnico en Informática".



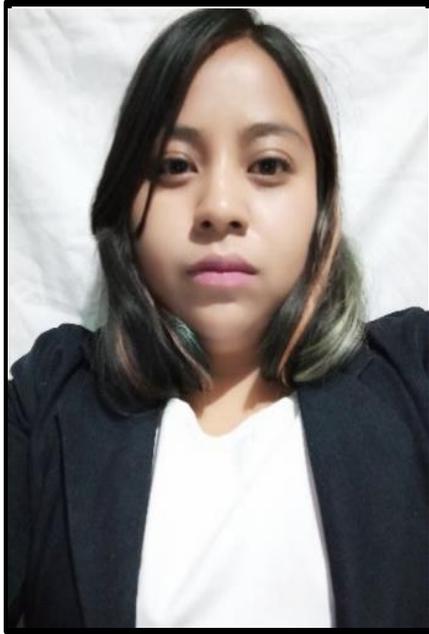
Jaime Rubio Martínez, lugar de nacimiento: Toluca Edomex

Mi Formación académica va desde la primaria en la escuela miguel hidalgo, la secundaria 0093 José Martí, La preparatoria CBT Atlacomulco, y estudio actualmente la universidad en UAEM Atlacomulco. El área disciplinaria de desarrollo académico y campo de especialización es en el área de la informática ya que estudie en CBT para técnico y estudio informática administrativa.





Licenciatura en Informática Administrativa 2022A



Alma Joselin Sánchez Ramírez, tengo 20 años de edad, soy originaria de Santa María Canchesda, Temascalcingo, Edo. De México actualmente soy estudiante de la Licenciatura de Informática Administrativa en la Universidad Autónoma del Estado de México cursando el cuarto semestre.

Mi formación académica inicio en el preescolar “Antonio José de Sucre” en Santa María Canchesda a los 5 años, posteriormente la primaria en “José María Morelos y Pavón” terminando la escuela Telesecundaria “Benito Juárez” en el mismo lugar estudiando así en la Preparatoria “CBT NO.3 TEMASCALCINGO” egresando como Técnico en Informática.

En mi experiencia con la Licenciatura tengo una gran satisfacción hasta el momento porque he intentado sacar adelante mis trabajos de igual forma arriesgarme como en vincular materias para ciertas complicaciones como alumnos, con áreas relacionadas en la informática.

Mi objetivo es poder brindar mi ayuda a los compañeros en fomentar la parte esencial de algunos temas que se habla en mi Licenciatura junto consigo aprender e identificar conocimientos que requieren.