
	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	1/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería	Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica		
La impresión de este documento es una copia no controlada			


Manual de prácticas de Temas Selectos de Programación I y II

Elaborado por:	Revisado por:	Autorizado por:	Vigente desde:
M.F. Gabriel Hurtado Chong M.A. Luis Yair Bautista Blanco M.I. Jorge Armando Rodríguez Vera Ing. Miguel Serrano Reyes M.I. Gersaín Barrón Velázquez Ing. Ricardo Fernando Abela Posada	M.F. Gabriel Hurtado Chong M.I. Jorge Armando Rodríguez Vera Ing. Miguel Serrano Reyes	Dr. Francisco Javier Solorio Ordaz	24 de enero de 2020

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	2/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Índice de prácticas

Práctica 1: Programación concurrente.....	3
Práctica 2: Implementación de una Interfaz de Programación de Aplicaciones (API)	8
Práctica 3: Base de Datos: Manejo y conexión.....	11
Práctica 4: Interfaz Software Hardware	15
Práctica 5: Servicios web	19

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	3/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Práctica #1

Programación concurrente

1. Seguridad en la ejecución

	Peligro o Fuente de energía	Riesgo asociado
1	Tensión alterna	Electrocución

2. Objetivos de aprendizaje


OBJETIVO GENERAL: Entender el concepto de programación concurrente y ser capaz de estructurar el código para su implementación.

OBJETIVOS ESPECÍFICOS:

- Realizar un código que considere la implementación de hilos.
- Evaluar las diferencias de código de programas secuenciales en comparación con programas distribuidos.
- Identificar el código para sincronizar el acceso de hilos a objetos en un programa.

3. Introducción

Un programa concurrente es aquél en el que un cálculo puede avanzar sin esperar a que se completen otros cálculos, ejecuta procesos en paralelo. Su desarrollo fue motivado originalmente por el deseo de generar sistemas operativos más confiables, aunque actualmente su alcance ha llegado a muchas áreas de desarrollo tecnológico por los beneficios que ofrece respecto a la programación tradicional.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	4/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Un algoritmo es un texto, o una representación gráfica, que describe sentencias que deben ejecutarse; un proceso, por otro lado, es un "texto en acción", una entidad dinámica generada por la ejecución de un algoritmo. Un subproceso (también conocido como hilo) es un proceso definido por un solo flujo de control (esto puede ser un programa individual, una computadora o una red) y, en la mayoría de los casos, es un componente de un proceso superior en donde se ha generado.


Pueden existir múltiples hilos dentro de un proceso, ejecutándose simultáneamente y compartiendo recursos, como la memoria del sistema, mientras que diferentes procesos podrían no compartir este tipo de recursos.

La implementación de hilos difiere entre los distintos sistemas operativos. Los sistemas con un solo procesador generalmente implementan hilos mediante un mecanismo de división de tiempo: la unidad central de procesamiento (CPU) cambia entre diferentes hilos de software. Este cambio de contexto generalmente ocurre suficientemente rápido como para que los usuarios perciban que los hilos o tareas se ejecutan en paralelo. En un sistema multiprocesador o multinúcleo, varios hilos pueden ejecutarse en paralelo, con cada procesador o núcleo ejecutando un subproceso separado simultáneamente.

El uso de hilos es un modelo generalizado de programación y ejecución que permite que existan múltiples subprocesos en el contexto de un proceso dado. Sin embargo, en un ambiente de programación donde se han generado varios hilos trabajando sobre elementos o controles en común del proceso padre, debe controlarse el acceso a estos recursos de manera secuencial, con el fin de evitar la generación de excepciones debido a que dichos elementos pueden estar siendo usados por otros hilos en un momento dado. Regularmente los distintos lenguajes de programación proveen mecanismos para evitar este tipo de problemas.

La programación concurrente tiene, entre otras, las siguientes ventajas:

Aumento del rendimiento del programa ya que la ejecución paralela de un programa concurrente permite que la cantidad de tareas completadas en un tiempo determinado aumente proporcionalmente a la cantidad de procesadores del sistema.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	5/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Permitir que una aplicación siga respondiendo a la interacción con el usuario pues si en un programa el subproceso de ejecución principal se bloquea en una tarea de ejecución prolongada, la aplicación completa puede aparecer como congelada. Al mover tales tareas de larga ejecución a un subproceso que se ejecuta simultáneamente con el proceso de ejecución principal, es posible que la aplicación siga respondiendo a las entradas del usuario mientras ejecuta tareas en segundo plano.


Al usar subprocesos, una aplicación puede servir a múltiples clientes al mismo tiempo, utilizando menos recursos de los que necesitaría al usar múltiples copias del proceso total requerido.

Muchos lenguajes de programación soportan hilos. Los lenguajes C, C++, Java, Python y .NET Framework, por mencionar algunos, exponen los subprocesos a los desarrolladores y permiten diferentes grados de usabilidad. Algunos lenguajes están diseñados primordialmente para la programación paralela, especialmente mediante la unidad de procesamiento de gráficos (GPU, por sus siglas en inglés), sin requerir concurrencia o hilos.

4. Material y equipo



Computadora

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	6/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

5. Desarrollo

Actividad 1. Implementación de hilos

Escribir el código que involucre un algoritmo de ejecución prolongada, con la intención de que un programa principal quede congelado mientras se ejecuta dicho algoritmo (programación secuencial). Una vez hecho lo anterior, realizar un código para enviar la ejecución del algoritmo a un hilo (subproceso) y discutir las diferencias que existe entre cada uno de los códigos generados (programación concurrente).

Actividad 2. Evaluación de tiempos de ejecución


Modifique el código de la actividad anterior de tal manera que se ejecute N veces el algoritmo de manera secuencial y concurrente, posteriormente repita para 2N veces, también para 3N, 4N y 5N. Genere una comparativa respecto al código empleado y con base en la linealidad o no linealidad esperada por la respuesta del mismo.

Actividad 3. Sincronización de hilos

Haga una modificación del algoritmo que se ha estado usando para que escriba un dato en un archivo determinado, genere el código para enviar el algoritmo a los hilos necesarios y generar un error de acceso al archivo (debido a que otro hilo lo esté usando). Posteriormente genere el código de un mecanismo para sincronizar el acceso de los hilos al archivo y determine si esta acción elimina el error de accesibilidad.

6. Bibliografía

- BELL, Douglas y PARR, Mike. **C# para estudiantes**. México, Pearson, 2010.
- BRINCH, Hansen. **The origin of concurrent programming**. New York, Springer, 2002.
- RAYNAL, Michel. **Concurrent Programming: Algorithms, Principles, and Foundations**. New York, Springer, 2013.
- LAAKMANN MCDOWELL, Gayle. **Cracking the Coding Interview: 189 Programming Questions and Solutions**. Palo Alto, CA., CareerCup, 2016.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	7/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería	Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica		
La impresión de este documento es una copia no controlada			

Práctica #2

Implementación de una Interfaz de Programación de Aplicaciones (API)

1. Seguridad en la ejecución

	Peligro o Fuente de energía	Riesgo asociado
1	Tensión alterna	Electrocución

2. Objetivos de aprendizaje


OBJETIVO GENERAL: El alumno comprenderá el concepto de una interfaz de programación de aplicaciones (API) y lo integrará a un proyecto que aproveche algunas de las características ofrecidas por una API en particular.

OBJETIVOS ESPECÍFICOS:

- Implementar una API
- Entender las ventajas de ocupar una API respecto a la generación del código específico para atender un problema en particular.

3. Introducción

Una API es un conjunto de herramientas, programas y protocolos que se integran a un proyecto de software con la intención de aprovechar las ventajas de comunicarse con productos propios o de terceros.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	8/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Generalmente el uso de una API permite ahorrar tiempo y costo de desarrollo debido a que las aplicaciones se ajustan en variadas ocasiones a las necesidades de los negocios e instituciones en general.


Un ejemplo de una API que se ocupa de manera muy recurrente es la de Google Maps que permite trazar rutas y estimar tiempos de traslado por parte del usuario dependiendo de la geolocalización.

Generalmente una API requiere de un registro por parte del programador para poder ocuparla y dependiendo de la licencia hasta un cobro por derecho de uso que establece los límites de uso. La manera en que se puede ocupar una API es generando una cadena de código HASH que se trata de una secuencia aleatoria de caracteres de una longitud definida y única que permite identificar al usuario programador. Si se llegase a perder dicho HASH ya no se podría acceder con el proveedor y la API deja de funcionar correctamente.

4. Material y equipo



Computadora

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	9/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

5. Desarrollo

Actividad 1. Implementar una API.


Con base en las necesidades de un programa en particular, elegir una API gratuita que permita atender al menos una de las particularidades de éste. Realizar la el código respectivo para el programa en cuestión destacando el elemento obtenido a través de la API.

Actividad 2. Ventajas de una API.

Investigar el tiempo que lleva en el mercado la API, versiones que ha tenido y mejoras en cada una de ellas, así como los módulos mínimos necesarios para su uso. Con base en lo anterior, brindar un estimado de horas de programación que se requerirían para sustituir las funciones que atiende la API programando específicamente la función que se utilizó en la actividad anterior.

6. Bibliografía

- JACOBSON, Daniel, BRAIL, Greg y WOODS, Dan. **APIs A Strategy Guide**. USA, O'Reilly, 2012.
- HOLLANDER, Ami (Ed.). **Learning Firebase**. <https://riptutorial.com/Download/firebase.pdf>, 20-06-2019
- LEWIN, Marck. **Google Maps API**. USA, Syncfusion, 2018.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	10/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Práctica #3

Base de Datos: Manejo y conexión

1. Seguridad en la ejecución

	Peligro o Fuente de energía	Riesgo asociado
1	Tensión alterna	Electrocución

2. Objetivos de aprendizaje

OBJETIVO GENERAL: El alumno comprenderá el manejo de bases de datos y realizará una conexión con un equipo para modificar una en particular.


OBJETIVOS ESPECÍFICOS:

- Realizar el código para las operaciones básicas en una base de datos.
- Desarrollar un código que utilice los elementos disponibles en una base de datos local y muestre algunos tipos de datos que contenga un registro solicitado.

3. Introducción

Una base de datos es un conjunto de información estructurada bajo un contexto particular que puede ser accedida desde una computadora, dispositivo móvil o cualquier equipo que requiera manejarla.

En general las bases de datos pueden clasificarse como relacionales y no relacionales.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	11/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Las bases de datos relacionales son aquéllas que se pueden equiparar con una tabla de Excel, donde los datos son básicamente integrados a celdas y cada columna representa un tipo de dato distinto, mientras que cada fila distingue a un registro.

Por ejemplo: Una tienda de mascotas donde se almacenan los datos de cada animal como nombre, especie, fecha de nacimiento y vacunas. La información por cada mascota la podemos almacenar en una tabla como se muestra a continuación:


Nombre	Especie	Fecha de nacimiento	Vacunas
Fígaro	Gato	14/noviembre/2017	Parásitos, Viral
Cofi	Perro	18/Enero/2016	Viral, Triple

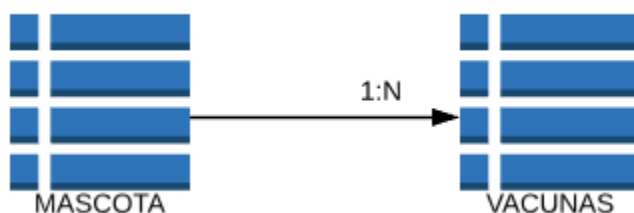
De la tabla anterior se puede observar que los datos que se insertan en Vacunas pueden ser múltiples por lo que en las bases relacionales se hace la separación en más tablas y se ocupa un identificador (ID) para reconocer la mascota de la que se trata.

ID	Nombre	Especie	Fecha de nacimiento
100	Fígaro	Gato	14/noviembre/2017
200	Cofi	Perro	18/Enero/2016

ID_Vacunas	Vacunas
10	Parásitos
20	Viral
30	Triple

Para poder relacionar ambas tablas se acostumbra utilizar un esquema que nos permita visualizar el comportamiento de ambas a la vez:

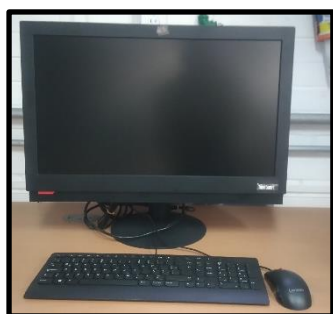
	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	12/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			




Para manipular una base de datos relacional se ocupa, generalmente, lenguaje SQL que contiene instrucciones definidas para su manipulación como puede ser: eliminar tablas, ingresar datos a la tabla, eliminar datos de la tabla y hacer cambios en los datos de la tabla.

Las bases de datos no relacionales son generalmente representadas en un esquema de árbol lineal donde el ID de cada objeto es el que nos permite distinguirlo de los demás. En este caso, a diferencia del otro tipo de bases de datos, todos los objetos son independientes y no se pueden agrupar de acuerdo a características en particular, lo que generalmente se conoce como clusters.

4. Material y equipo



Computadora

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	13/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería	Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica		
La impresión de este documento es una copia no controlada			

5. Desarrollo

Actividad 1. Manejar base de datos


Elegir un manejador de base de datos y manipularlo para crear una base de datos que contenga al menos 4 tipos de datos distintos y 5 registros diferentes. Realizar al menos dos cambios en los registros.

Actividad 2. Acceso a la base de datos.

Realizar un código que permita al usuario consultar un registro de una base de datos almacenada previamente, así como visualizar al menos dos de los tipos de datos que contenga ese registro.

6. Bibliografía

- BEIGHLEY, Lynn. **Head First SQL**.USA, O'Reilly 2007.
- NEVADO, Victoria. **Introducción a las bases de datos**. Madrid, Vision Libros, 2010.
- CHARLIE, Brooks. **Enterprise NoSQL for Dummies**. USA, John Wiley&Sons, 2014.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	14/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Práctica #4

Interfaz Software Hardware

1. Seguridad en la ejecución

	Peligro o Fuente de energía	Riesgo asociado
1	Tensión alterna	Electrocución

2. Objetivos de aprendizaje


OBJETIVO GENERAL: Entender los protocolos existentes para comunicar un programa (software) con elementos físicos (hardware) que le sean de utilidad.

OBJETIVOS ESPECÍFICOS:

- Entender qué es la comunicación serial.
- Realizar el código para la comunicación entre un programa y un dispositivo externo.

3. Introducción

La interfaz software-hardware es de amplio uso en la vida académica y profesional de las personas involucradas en la industria de la programación. Una de las formas más recurridas para la comunicación entre software y hardware es a través de microcontroladores o microcomputadoras con la intención de tener entradas y salidas de propósito general, tanto digitales como analógicas y a partir de ellos acceder a información de sensores o generar respuestas en actuadores con base en la ejecución de un programa.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	15/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

La manera más sencilla de comunicar una computadora con un microcontrolador es mediante el uso de la comunicación serial. La comunicación serial es un proceso por el cual se puede enviar un bit a la vez, o palabras de bits; entre sus ventajas está la reducción de cables con respecto a un proceso en paralelo, disminuyendo así la posibilidad de introducir ruido de línea en un proceso.

Entre los protocolos más conocidos de comunicación serial se encuentra el USB (Universal Serial Bus, sin embargo, existen más protocolos de comunicación como los que se mencionan a continuación:

- Ethernet
- I²C
- RS-232
- RS-485
- SPI

Un protocolo es un conjunto de reglas y normas que están obligadas a cumplir todos los sistemas con la finalidad de estandarizar la comunicación entre ellos. Sin ese conjunto de reglas, la comunicación requeriría condiciones particulares cada vez que se deseara establecer y eso complicaría la interacción entre desarrolladores. Algunas de las características que norman los protocolos son:

- Detección de la conexión física
- Handshaking
- Cómo iniciar y finalizar un mensaje.
- Procedimientos para dar formato a un mensaje
- Detección de pérdida de información

En el caso específico del uso de microcontroladores, la comunicación serial que se utiliza sigue el protocolo RS-232, por ello se requiere un complemento que permita convertir ese protocolo en USB, para conectarse directamente con la computadora en donde se ejecute el software que interesa interactúe con el hardware.




	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	16/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería	Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica		
La impresión de este documento es una copia no controlada			

Figura 1. Proceso de conversión USB a RS232.

En la figura 1 se observan los pines usados en el protocolo USB, D+ y D-, ambos se utilizan para enviar y recibir información, pero nunca al mismo tiempo, es decir, en algún momento, envían información y en otro reciben; a lo anterior se le conoce como comunicación Half Dúplex.

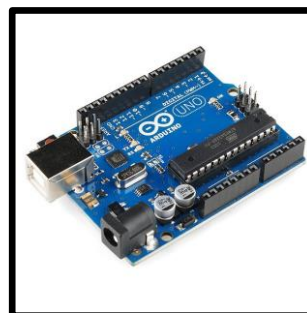
En el caso de los pines entre el convertidor y el microcontrolador se encuentran RX (recepción de datos) y TX (transmisión de datos). Cada pin desempeña una función diferente; el pin TX sólo envía y el RX sólo recibe, de tal forma que al comunicarse con otro dispositivo se realiza una conexión cruzada y la dirección de la información es en un sólo sentido tal cual como se muestra en las flechas.

En el caso de algunas tarjetas de desarrollo, el convertidor USB a RS-232 ya está integrado, por lo que no es necesario conectar un adaptador adicional reduciendo así la cantidad de elementos a utilizar.


4. Material y equipo



Computadora



Microcontrolador o Tarjeta de desarrollo con convertidor USB a RS-232

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	17/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

5. Desarrollo

Actividad 1. Comunicación Computadora a Hardware externo, envío.


Realizar un código en el que a través de una interfaz se pueda enviar una señal de salida digital a una tarjeta de desarrollo, tal salida deberá ser visualizable físicamente a través de un hardware externo.

Actividad 2. Comunicación Computadora a Hardware externo, recepción.

Realizar un código en el que se reciba una señal de entrada analógica leída por una tarjeta de desarrollo, tal dato será visualizable a través de algún elemento en una interfaz por software.

6. Bibliografía

- LAAKMANN MCDOWELL, Gayle. **Cracking the Coding Interview: 189 Programming Questions and Solutions**. Palo Alto, CA., CareerCup, 2016.
- BELL, Douglas y PARR, Mike. **C# para estudiantes**. México, Pearson, 2010

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	18/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

Práctica #5

Servicios web

1. Seguridad en la ejecución

	Peligro o Fuente de energía	Riesgo asociado
1	Tensión alterna	Electrocución

2. Objetivos de aprendizaje


OBJETIVO GENERAL: El alumno comprenderá el concepto de servicio web así como su utilidad e implementará uno.

OBJETIVOS ESPECÍFICOS:

- Generar el código que permita utilizar protocolos de servicios web.
- Identificar los elementos para lograr la compatibilidad entre un programa desarrollado y un servicio web.

3. Introducción

Las aplicaciones web han sufrido una evolución análoga a la que ya padecieron las aplicaciones de escritorio que utilizan los recursos propios de cada sistema operativo para construir su interfaz de usuario. Inicialmente, estas aplicaciones se ejecutaban en una única máquina, que era además la máquina donde se almacenaban los datos requeridos, procesados y entregados.

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	19/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

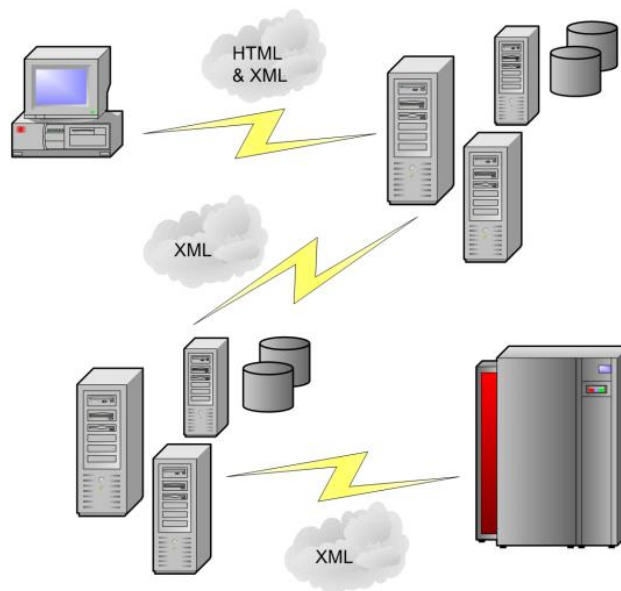



Figura 1. Representación gráfica de la arquitectura de un servicio web.

Un servicio web (web service) es un conjunto de técnicas, protocolos y estándares que sirven para el intercambio de datos entre aplicaciones a través de internet. Al ser un servicio apoyado en HTTP y un envío de datos, el servicio web puede funcionar con diferentes lenguajes de programación.

Esta inter-operatividad e intercambio de mensajes entre aplicaciones a través de la red, están típicamente basados en XML sobre HTTP conjuntamente con otros estándares, lo que lo hace ideal para internet.

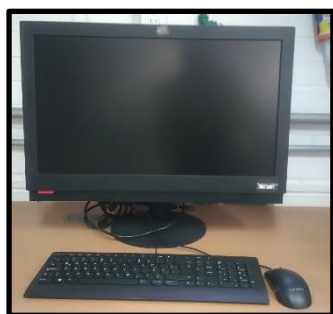
Entre algunos de los estándares utilizados para realizar servicios web están:

- Web Services Protocol Stack: conjunto de servicios y protocolos de los servicios web.
- XML (Extensible Markup Language): formato estándar para los datos que se vayan a intercambiar.
- WSDL (Web Services Description Language): es el lenguaje de la interfaz pública para los servicios web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios web.


	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	20/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería		Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica	
La impresión de este documento es una copia no controlada			

La complejidad de un servicio web se ve muy afectada al tener que crear autenticación de las máquinas que acceden a su servicio, es por ello que la configuración y aprendizaje de los protocolos es muy importante para evitar un mal uso o tener vulnerabilidades en el sistema. Por las razones anteriores, hoy día se ofrece una variedad de proveedores de servicios de almacenamiento y accesibilidad a través de la nube que simplifican las tareas de los desarrolladores; de tal forma que con sólo subir el programa que se desea tener como servicio web, los proveedores se encargan del cifrado, seguridad y acceso al mismo, disminuyendo la incidencia de ataques informáticos y promoviendo un correcto ecosistema de servicios a través de internet. Ejemplos de tales proveedores son: Google Cloud Platform, Amazon Web Services, y Azure.

4. Material y equipo



Computadora

	Manual de prácticas de Temas Selectos de Programación I y II	Código:	MADO-84
		Versión:	01
		Página	21/21
		Sección ISO	8.3
		Fecha de emisión	24 de enero de 2020
Facultad de Ingeniería	Área/Departamento: Laboratorio de Cómputo de Ingeniería Mecatrónica		
La impresión de este documento es una copia no controlada			

5. Desarrollo

Actividad 1. Crear un servicio web simple.

Realizar el código para implementar un servicio web que resuelva operaciones simples, definiendo métodos que encapsulen la funcionalidad del servicio.

Actividad 2. Crear un servicio web basado en base de datos.

Desarrollar el código para una aplicación cuyo funcionamiento requiera de una base de datos, realizando la conexión a través de un servicio web.

6. Bibliografía

- MARTIN, Robert C. **Código limpio: Manual de estilo para el desarrollo ágil de software**. España, Anaya Multimedia, 2012
- CEBALLOS SIERRA, Francisco Javier. **Microsoft C#. Curso de programación**. México, Alfaomega, 2007
- DEITEL, Harvey y Deitel, PAUL. **C# Cómo programar**. España, Pearson, 2007
- LÓPEZ ROMÁN, Leobardo. **Metodología de la programación orientada a objetos**. México, Alfaomega, 2007