**Semana 1 - Fundamentos de las Ciencias Informáticas**

Vamos a ver que las computadoras surgen por la necesidad humana de cuantificar. En la antigüedad era suficiente con contar con los dedos, hasta hace no tantos años atrás era suficiente con llevar controles en un cuaderno. Pero hoy en día por la propia evolución de la humanidad, su desarrollo y el gran volumen de datos que se manejan son necesarias cada vez más máquinas con gran poder de procesamiento.

Vamos a ver que con el transcurrir de los años los equipos fueron aumentando su capacidad de procesamiento y también su capacidad de almacenamiento. Y aunque parece contradictorio, fueron aumentando en capacidades y disminuyendo en costo y tamaño.

Tenemos computadoras cada vez más potentes y más pequeñas, que sirven para múltiples actividades. Un mismo dispositivo puede ser utilizado en ámbitos muy diversos. Podemos considerar la computadora de cualquiera de nosotros que en estos momentos está siendo utilizada para estudiar y seguramente en otro momento o incluso al mismo tiempo la utilizamos para estar en contacto con otras personas, para informarnos para hacer cálculos matemáticos, escuchar música, ver contenido de distintos formatos en la web.

Todo esto ha sido posible gracias a diferentes personas a lo largo de la historia que han investigado y desarrollado nuevos dispositivos posibilitando que las computadoras sean de uso común tanto en lo personal como en la industria.

Al realizar la lectura que indicamos en esta hoja de ruta van a poder conocer un poco más sobre la historia de la informática.

**Objetivo: Comprender los conceptos de hardware y software y poder diferencia para ambos casos entre propietario y abierto o libre.**

La diferencia entre hardware y software puede simplificarse a la diferencia este aquello que podemos tocar, es decir los componentes físicos de la computadora, (HARDWARE) y los componentes lógicos o que no podemos tocar (SOFTWARE).

Ahora bien, eso es bastante simple, pero que sucede cuando hablamos de propietario y abierto o libre.

Tanto el software como el hardware pueden ser o propietario o libre.  En ambos casos, tanto para el hard como para el soft que sean propietarios significa que los usuarios no pueden hacer modificaciones ni estudiar el funcionamiento interno de ese hardware o software. En cambio, cuando hablamos de libre son aquellos casos donde cuando uno adquiere un producto ya sea de hard o de soft tiene acceso a sus configuraciones, datos de base, funcionamiento interno y tiene la posibilidad de poder hacer modificaciones o mejoras al dispositivo o aplicación. Esto ha permito grandes avances y la posibilidad de poder ajustar el producto a la necesidad particular de ese usuario. Cuando nos referimos al usuario no hacemos referencia solo al usuario final, que posiblemente no cuente con los conocimientos para realizar estas modificaciones. sino también al usuario técnico que utiliza esos dispositivos y programas para su trabajo.

**Semana 2 - Fundamentos de las Ciencias Informáticas**

**Objetivo: Conocer un poco más acerca de los sistemas de computación**

Un sistema de computación o sistema informático se compone con una computadora, que incluye el hardware y el software necesarios para que sea funcional para un usuario.

Es el sistema encargado de recoger datos, procesarlos y transmitir la información una vez procesada.

Es decir que un sistema de computación es el conjunto de hardware y software. Estos sistemas deben tener la capacidad de recibir información del usuario (introducir datos), procesar los datos y con los datos procesados, crear información para su almacenamiento y/o para su salida.

En la unidad 3 de esta asignaturas vamos a conocer acerca de la arquitectura y organización de los sistemas de computación.

En este primer acercamiento vamos a conocer los tipos de computadoras:

Las computadoras, por su potencia de cálculo, capacidad de almacenamiento interno y número de periféricos que pueden soportar, se clasifican en los siguientes grandes grupos:

* Supercomputadoras
* Mainframes
* Servidores
* PC (Computadora Personal)
* Computadoras portátiles
* Minicomputadora
* Microcomputadora
* Computadoras de bolsillo
* Computadoras incrustadas
* Computadoras de carácter especifico

**Objetivo: Comprender como se lleva a cabo la resolución de problemas con computadoras**

La resolución de problemas mediante una computadora consiste en dar una adecuada formulación de pasos precisos a seguir. En este sentido la definición del concepto de algoritmo es el tema principal.

Una computadora no puede comprender indicaciones ambiguas. Cada orden que se le da a la computadora debe tener una única interpretación de lo que hay que realizar. Una máquina no posee la capacidad de decisión del ser humano para resolver situaciones no previstas. Si al dar una orden a la computadora se produce una situación no contemplada, será necesario abortar esa tarea y recomenzar todo el procedimiento nuevamente.

Etapas en la resolución de problemas con computadora

* Análisis del problema
* Diseño de una solución
* Especificaciones de algoritmos
* Escritura de programas
* Verificación

**Algoritmo**

La palabra algoritmo deriva del nombre de un matemático árabe del siglo IX, llamado Al-Khuwarizmi, quien estaba interesado en resolver ciertos problemas de aritmética y describió varios métodos para resolverlos. Estos métodos fueron presentados como una lista de instrucciones específicas (como una receta de cocina) y su nombre es utilizado para referirse a dichos métodos.

Un algoritmo es, en forma intuitiva, una receta, un conjunto de instrucciones o de especificaciones sobre un proceso para hacer algo. Ese algo generalmente es la solución de un problema de algún tipo. Se espera que un algoritmo tenga varias propiedades. La primera es que un algoritmo no debe ser ambiguo, o sea que, si se trabaja dentro de cierto marco o contexto, cada instrucción del algoritmo debe significar sólo una cosa.

**Objetivo: Conocer los sistemas de numeración más utilizados en informática**

En el mundo existen distintos sistemas de numeración, el más utilizado es el sistema decimal. Pero cuando se trata de la computadora aparecen otros sistemas de numeración, el sistema binario y el hexadecimal.

Un sistema de numeración es un conjunto de símbolos y reglas que se utilizan para la representación de datos numéricos o cantidades. Se caracterizan fundamentalmente por su base, que es el número de símbolos distintos que utiliza, y además es el coeficiente que determina cuál es el valor de cada símbolo dependiendo de la posición que ocupe.

Los sistemas de numeración actuales son sistemas posicionales, en los que el valor relativo que representa cada símbolo o cifra de una determinada cantidad depende de su valor absoluto y de la posición relativa que ocupa dicha cifra con respecto a la coma decimal.

**Sistema decimal**

Es un sistema posicional, que utiliza un conjunto de 10 símbolos. (0 a 9). Base 10.

**Sistema hexadecimal**

Es un sistema posicional que utiliza dieciséis símbolos para la representación de cantidades. Estos símbolos son los siguientes: 0 1 2 3 4 5 6 7 8 9 A B C D E F. Las letras A, B, C, D, E, F equivalen a 10, 11, 12, 13, 14 y 15 del sistema decimal respectivamente.

**Sistema binario**

Es el sistema de numeración que utiliza internamente el hardware de las computadoras actuales. La base o número de símbolos que utiliza el sistema binario es 2, siendo los símbolos 0 y 1.

**Semana 3 - Conceptos de Lenguajes de programación**

**Lenguajes de Programación**

Un conjunto de letras negras en un fondo blanco

Descripción generada automáticamente con confianza media

Los **lenguajes de programación** son herramientas que permiten construir aplicaciones personalizadas. Un **lenguaje de programación** es un conjunto de instrucciones junto con una serie de reglas que indiquen como pueden combinarse esas instrucciones para representar ideas más complejas. Cada instrucción tiene su propia sintaxis y semántica. La sintaxis hace referencia a la representación simbólica de la instrucción; la semántica hace referencia al significado de la instrucción. Cada lenguaje de programación contiene un conjunto finito y preciso de instrucciones utilizables para especificar la solución buscada. Estos lenguajes de programación tienen como objetivo lograr la comunicación entre el programador y las máquinas.

**Lenguaje máquina**

Como sabemos el hardware solo comprende **lenguaje máquina**. Este lenguaje es propio de cada máquina, si bien todos tienen instrucciones para efectuar las cuatro operaciones aritméticas básicas, para comparar pares de números, instrucciones para formar bucles, etc. cada uno de estos lenguajes máquina son lenguajes diferentes, y las máquinas basadas en uno de ellos no pueden entender los programas escritos en otro.

El lenguaje maquina, utilizado por los primeros programadores, emplea códigos numéricos para representar las distintas instrucciones, esto dificultaba demasiado la tarea de programar, era tedioso tener que identificar errores en extensas combinaciones de 0s y 1s.

**Lenguajes de bajo y alto nivel**

Diagrama

Descripción generada automáticamente

Con la invención del **ensamblador**(un lenguaje funcionalmente similar al lenguaje máquina, pero más sencillo de escribir, leer y comprender por las personas) el proceso de programación se hizo más sencillo. En este lenguaje, los programadores utilizan códigos alfabéticos que se corresponden con las instrucciones numéricas de la máquina. Por ejemplo, la sentencia ensamblador para restar podría ser SUB. Desde luego, SUB no significa nada para la computadora, la cual sólo responde ante comandos del tipo 10110111. Para establecer un lazo de unión entre el programador y la computadora, un programa llamado **ensamblador**traduce cada instrucción de este lenguaje en la sentencia máquina correspondiente.

Los lenguajes ensamblador y máquina son **lenguajes de bajo nivel,**los cuales conllevan procesos repetitivos, tediosos y muy propensos a los errores. El lenguaje ensamblador se continua utilizando para escribir partes de aplicaciones en las que la velocidad y la comunicación con el hardware es un factor crítico.

La programación con lenguajes de bajo nivel obliga al programador a pensar al nivel de la máquina y a incluir una enorme cantidad de detalles en cada programa. Esto conlleva procesos repetitivos, tediosos y muy propensos a los errores. Para complicar aún más las cosas, cualquier programa escrito en uno de estos lenguajes deben ser reescritos por completo antes de poder utilizarlos en una computadora con un lenguaje máquina diferente.

En busca de dar solución a las cuestiones antes mencionadas surgieron los **lenguajes de alto nivel,**que están a medio camino entre el lenguaje natural de los humanos y los lenguajes máquina, fueron desarrollados a principio de la década de los 50 para simplificar y perfilar el proceso de programación. Los lenguajes de alto nivel permiten que los programadores escriban programas usando una terminología y notación familiar en lugar de las enigmáticas instrucciones máquina.

**Intérpretes y compiladores**

Para que una computadora entienda un programa escrito en uno de estos lenguajes, es preciso convertirlo al idioma de las máquinas, es decir, a unos y ceros. Para convertir un programa en lenguaje máquina, necesitamos un software de traducción. Dicho programa puede ser un **intérprete**(un programa que traduce y transmite cada sentencia de forma individual, del mismo modo que en las Naciones Unidas se traduce un discurso del español al inglés), o un **compilador**(un programa que traduce el programa completo antes de pasarlo a la computadora, del mismo modo que un estudiante puede traducir una novela del inglés al español).

Los intérpretes y los compiladores traducen los programas de alto nivel en lenguaje máquina. Una vez interpretada o compilada, una sentencia de uno de estos lenguajes se transforma en varias instrucciones máquina. Un lenguaje de alto nivel oculta al programador la mayoría de los detalles oscuros de las operaciones máquina. Como resultado de ello, resulta más sencillo centrarse en la lógica básica del programa, es decir, en la idea principal. Además de ser más sencillos de escribir y depurar, los programas de alto nivel tienen la ventaja de poder transportarse de una máquina a otra. Un código escrito en un determinado lenguaje estándar puede ser compilado y ejecutado en cualquier computadora que disponga de ese compilador.

[**Paradigmas de programación**](https://campus.uner.edu.ar/fcad/mod/page/view.php?id=35679)

Los [paradigmas de programación](https://campus.uner.edu.ar/fcad/mod/page/view.php?id=35679) representan enfoques fundamentalmente distintos para obtener soluciones a los problemas y, por tanto, afectan al proceso completo de desarrollo software.

El **paradigma imperativo**, también conocido como **paradigma procedimental**, representa el enfoque tradicional del proceso de programación. Como su nombre sugiere, el paradigma imperativo define el proceso de programación como el desarrollo de una secuencia de comandos que, al ser ejecutados, manipula los datos para generar el resultado deseado. Por tanto, el paradigma imperativo nos dice que debemos enfocar el proceso de programación determinando un algoritmo para solucionar el problema que nos traemos entre manos y luego expresando dicho algoritmo como una secuencia de sentencias.

El **paradigma orientado a objetos**, que está asociado con el proceso de programación denominado **programación orientada a objetos**(OOP, *Object-Oriented Programming*). De acuerdo con este paradigma, un sistema software se ve conceptualmente como un conjunto de unidades, denominadas **objetos**, cada uno de las cuales es capaz de llevar a cabo las acciones que le afectan directamente, así como de solicitar acciones a otros objetos. De forma conjunta, estos objetos interactúan para resolver el problema que tengamos entre manos.

El **paradigma orientada a aspectos** (AOP, Aspect-oriented programming) tiene por objetivo proporcionar mecanismos que hacen posible **separar los elementos que son transversales a todo el sistema**. Permite capturar los diferentes**intereses entrecruzados** que componen una aplicación en entidades bien definidas, eliminando las dependencias inherentes entre cada uno de los módulos que la componen. Este paradigma tiene por objetivo**encapsular aspectos transversales**, que atraviesan a diferentes módulos y no pueden encapsularse limpiamente en el POO.

El **paradigma declarativo**pide al programador que describa el problema que hay que resolver, en lugar del algoritmo que hay que aplicar. Para ser más precisos, un sistema de programación declarativo aplica un algoritmo preestablecido para resolución de problemas de propósito general con el fin de solucionar los problemas que se le presenten. En un entorno de este tipo, la tarea del programador consiste en desarrollar un enunciado preciso del problema en lugar de describir un algoritmo para la resolución del problema.

El **paradigma funcional**. En este caso, un programa se ve como una entidad que acepta entradas y genera salidas. Los matemáticos denominan a tales entidades funciones, razón por la que esta técnica recibe el nombre de paradigma funcional. Bajo este paradigma, los programas se construyen conectando entidades predefinidas más pequeñas (funciones predefinidas), tal que las salidas de cada unidad se utilicen como entradas de otras unidades, de tal forma que al final se obtenga la relación entrada-salida global deseada. En resumen, el proceso de programación bajo el paradigma funcional consiste en construir funciones como conjuntos anidados de otras funciones más simples.

**Semana 4 - Conceptos de Arquitecturas de Computadores**

**Comprender las diferencias entre arquitectura y organización de las computadoras.**

La arquitectura de computadores se refiere a los atributos de un sistema que son visibles a un programador, o para decirlo de otra manera, aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa. La organización de computadores se refiere a las unidades funcionales y sus interconexiones, que dan lugar a especificaciones arquitectónicas. Entre los ejemplos de atributos arquitectónicos se encuentran el conjunto de instrucciones, el número de bits usados para representar varios tipos de datos (por ejemplo, números, caracteres), mecanismos de E/S y técnicas para direccionamiento de memoria. Entre los atributos de organización se incluyen aquellos detalles de hardware transparentes al programador, tales como señales de control, interfaces entre el computador y los periféricos y la tecnología de memoria usada.

**Conocer la estructura y funcionamiento de los procesadores**Un procesador consta de tres partes: la unidad aritmético/lógica, que contiene los circuitos que realizan las operaciones con los datos (como por ejemplo sumas y restas), la unidad de control, que contiene los circuitos que coordinan las actividades de la máquina y la unidad de registros, que contiene celdas de almacenamiento de datos (similares a las celdas de la memoria principal), denominadas registros, que se emplean para almacenar temporalmente la información dentro del procesador.  
Para realizar una operación con datos almacenados en la memoria principal, la unidad de control transfiere los datos desde la memoria hasta los registros de uso general, informa a la unidad aritmético/lógica de qué registros son los que contienen los datos, activa los circuitos apropiados dentro de la unidad aritmético/lógica y le dice a esta en qué registro debe almacenar el resultado.  
Para transferir los patrones de bits, el procesador y la memoria principal de una máquina se conectan a través de un conjunto de hilos de conexión, denominado bus. A través de este bus, el procesador extrae (lee) datos de la memoria principal, suministrando la dirección de la celda de memoria pertinente, junto con una señal electrónica que le indica a los circuitos de memoria que debe extraer los datos contenidos en la celda indicada. De forma similar, el procesador coloca (escribe) datos en la memoria proporcionando la dirección de la celda de destino y los datos que hay que almacenar, junto con la señal electrónica apropiada que le dice a la memoria principal que debe almacenar los datos que se le están enviando.  
Basándonos en este diseño, la tarea de sumar dos valores almacenados en la memoria principal implica algunas tareas más que la mera ejecución de la operación de suma. Es necesario transferir los datos desde la memoria principal hasta los registros contenidos en el procesador; luego hay que sumar los valores y colocar el resultado en un registro y, finalmente, el resultado debe almacenarse en una celda de memoria.

**Ciclo de maquina**

El procesador lleva a cabo una instrucción repitiendo continuamente un algoritmo que le hace recorrer un proceso de tres pasos conocido con el nombre de ciclo de máquina. Los pasos del ciclo de máquina son la captación de instrucción, la decodificación y la ejecución.

Vemos que la ejecución de un programa almacenado en la memoria implica el mismo tipo de proceso que cualquiera de nosotros usaría al seguir una lista de instrucciones detallada. Mientras que nosotros podemos saber en qué punto de la lista nos encontramos tachando las instrucciones a medida que las ejecutamos, el procesador sabe dónde se encuentra mediante el contador de programa.

**Aplicar la clasificación de memorias**Diagrama

Descripción generada automáticamente

**Memoria caché:**es una memoria de alta velocidad localizada dentro del propio procesador. En esta área de memoria especial, la máquina trata de mantener una copia de aquella porción de la memoria principal que es de interés en ese momento concreto. Con este tipo de diseño, las transferencias de datos que normalmente se llevarían a cabo entre los registros y la memoria principal se realizan entre los registros y la memoria caché. Todos los cambios realizados en la memoria caché se transfieren luego de manera colectiva a la memoria principal, en algún momento más oportuno. El resultado es un procesador que puede ejecutar su ciclo de máquina más rápidamente, porque no se ve retardado por la comunicación con la memoria principal.

**Memoria Interna**

Un circuito electrónico

Descripción generada automáticamente con confianza media

La RAM (random access memory, memoria de acceso aleatorio) es el tipo más común de almacenamiento primario, o de memoria interna. Los chips de la RAM contienen circuitos que almacenan temporalmente las instrucciones y los datos del programa. La computadora divide cada chip de la RAM en muchas ubicaciones de memoria del mismo tamaño. Las ubicaciones de memoria, como las casas, tienen direcciones únicas para que la computadora pueda indicarles aparte cuándo ha de guardar o recuperar la  información. Puede almacenar un fragmento de información en cualquier ubicación RAM (puede elegir una al azar) y la computadora puede, si se le indica, recuperarla  rápidamente. De aquí el nombre de memoria de acceso aleatorio.

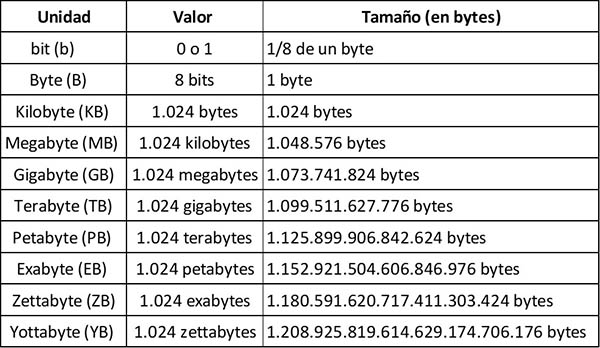
La información almacenada en la RAM no es más que un patrón de corriente eléctrica fluyendo a través de circuitos microscópicos en chips de silicio. Esto significa  que cuando la energía sale de la computadora se olvida instantáneamente de todo lo que se recordaba en la RAM. La RAM se llama a veces memoria volátil, porque la información  almacenada allí no se mantiene permanentemente. Esto podría ser un serio  problema si la computadora no tuviera otro tipo de memoria donde almacenar la información que no se quiere perder. Esta memoria no volátil se llama ROM (read-only memory, memoria de sólo lectura) porque la computadora sólo puede leer la información almacenada en ella; nunca puede escribir ninguna información nueva. Todas las computadoras modernas incluyen ROM que contiene instrucciones de arranque  y otra información crítica. La información de la ROM fue grabada en ella cuando se fabricó el chip, así que está disponible siempre que la computadora está  funcionando, pero no puede cambiarse salvo reemplazando el chip de la ROM.

**Memoria externa**



Las memorias externas son conocidas también como dispositivos o sistemas de almacenamiento masivo. Entre las ventajas de los sistemas de almacenamiento masivo, con respecto a la memoria principal, podemos citar una menor volatilidad, mayores  capacidades de almacenamiento, su bajo coste y, en muchos casos, la capacidad de extraer el medio de almacenamiento de la máquina, con el propósito de archivarlo.

A menudo se emplean los términos en línea y fuera de línea para describir a aquellos dispositivos que pueden estar conectados o no a una máquina. Estar en línea significa que el dispositivo o la información están conectados y la  máquina puede acceder fácilmente a ellos, sin que haya intervención humana. Estar fuera de línea significa que se requiere intervención humana para que  la máquina pueda acceder al dispositivo o la información, quizá porque el dispositivo  debe encenderse o porque el medio en el que se almacena la información debe introducirse en algún tipo de mecanismo o conectarse a la máquina.

**Reconocer las distintas unidades de medidas**

Así como para medir longitudes hay distintas unidades de medida, como por ejemplo milímetros, centímetros y metros, lo mismo sucede cuando queremos medir cantidades de información.

**• Byte:** grupo lógico de 8 bits. Si trabaja sobre todo con palabras, puede considerar un byte como un carácter del texto codificado en ASCII.

**• KB (kilobyte o K):** unos 1.000 bytes de información. Por ejemplo, se necesitan unos 5 K de almacenamiento para contener 5.000 caracteres de texto ASCII. (Técnicamente, 1 K es 1.024 bytes, porque 1.024 es 210, lo que simplifica el cálculo para las computadoras basadas en el sistema binario. Para aquellos de nosotros que no pensamos en binario, 1.000 se aproxima lo bastante.)

**• MB (megabyte o mega):**aproximadamente 1.000 KB, o 1 millón de bytes.

**• GB (gigabyte o giga):** aproximadamente 1.000 MB.

**• TB (terabyte):** aproximadamente 1 millón de MB o 1 billón de bytes. Esta masiva unidad de medida se aplica a los mayores dispositivos de almacenamiento disponibles actualmente.

**• PB (petabyte):** este valor astronómico es el equivalente a 1.024 terabytes, o 1000 billones de bytes. Aunque es improbable que nadie sea capaz de almacenar por ahora 1PB de datos en su PC doméstico, vamos definitivamente en esa dirección.

Las abreviaturas K, MB, GB, TB y PB describen la capacidad de los componentes de almacenamiento y de memoria. Podría, por ejemplo, describir una computadora diciendo que tiene 4GB de memoria (RAM) y un disco duro diciendo que tiene 1 TB de capacidad de almacenamiento.

**Semana 5 - Sistemas Operativos**

**Conocer la definición, evolución y tipos de sistemas operativos.**

Una computadora moderna consta de uno o más procesadores, una memoria principal, discos, impresoras, un teclado, un ratón, una pantalla o monitor, interfaces de red y otros dispositivos de entrada/ salida. En general es un sistema complejo. Si todos los programadores de aplicaciones tuvieran que comprender el funcionamiento de todas estas partes, no escribirían código alguno. Es más: el trabajo de administrar todos estos componentes y utilizarlos de manera óptima es una tarea muy desafiante. Por esta razón, las computadoras están equipadas con una capa de software llamada sistema operativo, cuyo trabajo es proporcionar a los programas de usuario un modelo de computadora mejor, más simple y pulcro, así como encargarse de la administración de todos los recursos antes mencionados.

Un sistema operativo es el software que controla el conjunto de operaciones de una computadora. Proporciona el mecanismo por el cual un usuario puede almacenar y extraer archivos, proporciona la interfaz mediante la que el usuario puede solicitar la ejecución de programas y proporciona también el entorno necesario para ejecutar los programas solicitados.

Podemos ver que en la parte inferior de la imagen se muestra el hardware, por encima del hardware se encuentra el software. La mayoría de las computadoras tienen dos modos de operación: modo kernel y modo usuario. El sistema operativo es la pieza fundamental del software y se ejecuta en modo kernel (también conocido como modo supervisor). En este modo, el sistema operativo tiene acceso completo a todo el hardware y puede ejecutar cualquier instrucción que la máquina sea capaz de ejecutar. El resto del software se ejecuta en modo usuario, en el cual sólo un subconjunto de las instrucciones de máquina es permitido. En particular, las instrucciones que afectan el control de la máquina o que se encargan de la E/S (entrada/salida) están prohibidas para los programas en modo usuario.

Los sistemas operativos actuales son paquetes software de gran tamaño y complejidad que han ido creciendo a partir de orígenes muy humildes. Las computadoras de las décadas de 1940 y 1950 no eran muy flexibles, ni tampoco eficientes, las máquinas podían ocupar una habitación completa. La ejecución de programas requería una tediosa preparación de los equipos, en el sentido de que había que montar cintas magnéticas, colocar tarjetas perforadas en los lectores de tarjetas, configurar una serie de conmutadores, etc. La ejecución de cada programa, denominada trabajo, se gestionaba como una actividad independiente: se preparaba la máquina para ejecutar el programa, se ejecutaba ese programa y luego había que extraer todas las cintas, tarjetas perforadas, etc. antes de que pudiera iniciarse la ejecución del siguiente programa. Cuando varios usuarios necesitaban compartir una máquina se suministraban hojas de petición para que esos usuarios pudieran reservar el uso de la máquina durante ciertos periodos de tiempo. Durante el periodo de tiempo asignado a un usuario, la máquina estaba totalmente bajo el control de dicho usuario. La sesión comenzaba normalmente con la preparación del programa, seguida de cortos periodos de ejecución de ese programa. A menudo, el usuario tenía que completar su tarea a toda prisa, tratando de terminar algún aspecto más de su proyecto (“Solo tardaré un minuto”) mientras que el siguiente usuario se iba impacientando cada vez más mientras preparaba su propio programa.

**Tipos de sistemas operativos**

Para los usuarios ocasionales de computadoras, las diferencias entre los sistemas operativos son más bien de carácter cosmético. Para los profesionales de la computación, sin embargo, los diferentes sistemas operativos pueden incluir variaciones enormemente importantes en las herramientas con las que trabajan o en la filosofía que deben seguir a la hora de diseminar y mantener su trabajo.

De todos modos, en lo que respecta al núcleo fundamental de los sistemas operativos, todos los sistemas operativos más conocidos intentan resolver los mismos tipos de problemas con que los expertos en el campo de la computación se han enfrentado durante más de medio siglo.Durante este tiempo se ha desarrollado una variedad bastante extensa de sistemas operativos, no todos se conocen ampliamente.

A continuación nombramos algunos:

* Sistemas operativos de mainframe
* Sistemas operativos de servidores
* Sistemas operativos de multiprocesadores
* Sistemas operativos de computadoras personales
* Sistemas operativos de computadoras de bolsillo
* Sistemas operativos integrados
* Sistemas operativos de nodos sensores
* Sistemas operativos en tiempo real
* Sistemas operativos de tarjetas inteligentes

**Poder aplicar la clasificación de sistemas operativos**.

**Establecer las diferencias entre sistemas de tiempo real y sistemas embebidos.**

**Sistemas operativos en tiempo real**

Estos sistemas se caracterizan por tener el tiempo como un parámetro clave. Por ejemplo, en los sistemas de control de procesos industriales, las computadoras en tiempo real tienen que recolectar datos acerca del proceso de producción y utilizarlos para controlar las máquinas en la fábrica. A menudo hay tiempos de entrega estrictos que se deben cumplir. Por ejemplo, si un auto se desplaza sobre una línea de ensamblaje, deben llevarse a cabo ciertas acciones en determinados instantes. Si un robot soldador realiza su trabajo de soldadura antes o después de tiempo, el auto se arruinará. Si la acción debe ocurrir sin excepción en cierto momento (o dentro de cierto rango), tenemos un sistema en tiempo real duro. Muchos de estos sistemas se encuentran en el control de procesos industriales, en aeronáutica, en la milicia y en áreas de aplicación similares. Estos sistemas deben proveer garantías absolutas de que cierta acción ocurrirá en un instante determinado.

**Sistemas embebidos**

Los sistemas embebidos son aquellos sistemas operativos dentro de las lavadoras, hornos de microondas, muñecas, radios de transistores, reproductores de MP3, cámaras de video, etc. Es muy probable que sean distintos unos de otros. Como todos los sistemas embebidos (incrustados) sólo ejecutan un número limitado de programas, que se conocen en tiempo de compilación, puede ser posible realizar optimizaciones que no sean posibles en sistemas de propósito general.

**Comprender las funciones y prestaciones de los sistemas operativos.**

Los sistemas operativos realizan dos funciones básicas proporcionar a los programadores de aplicaciones un conjunto abstracto de recursos simples, en vez de los complejos conjuntos de hardware; y administrar estos recursos de hardware.

El hardware presenta interfaces difíciles, enredadas, muy peculiares e inconsistentes para ser utilizados por las personas que tienen que escribir software. Una de las principales tareas del sistema operativo es ocultar el hardware y presentar a los programas (y a sus programadores) abstracciones agradables, elegantes, simples y consistentes con las que puedan trabajar. Los sistemas operativos ocultan la parte fea con la parte hermosa.

El concepto de un sistema operativo cuya función principal es proporcionar abstracciones a los programas de aplicación responde a una perspectiva de arriba hacia abajo. La perspectiva alterna, de abajo hacia arriba, sostiene que el sistema operativo está presente para administrar todas las piezas de un sistema complejo. En esta perspectiva el trabajo del sistema operativo es proporcionar una asignación ordenada y controlada entre los diversos programas que compiten por los recursos. Los sistemas operativos modernos permiten la ejecución simultánea de varios programas. Imagine lo que ocurriría si tres programas que se ejecutan en cierta computadora trataran de imprimir sus resultados en forma simultánea en la misma impresora. Las primeras líneas de impresión podrían provenir del programa 1, las siguientes del programa 2, después algunas del programa 3, y así en lo sucesivo: el resultado sería un caos. El sistema operativo puede imponer orden al caos potencial, guardando en búferes en disco toda la salida destinada para la impresora. Cuando termina un programa, el sistema operativo puede entonces copiar su salida, previamente almacenada, del archivo en disco a la impresora, mientras que al mismo tiempo el otro programa puede continuar generando más salida, ajeno al hecho de que la salida en realidad no se está enviando a la impresora todavía.

Además, los usuarios necesitan con frecuencia compartir no sólo el hardware, sino también la información (archivos o bases de datos, por ejemplo). En resumen, esta visión del sistema operativo sostiene que su tarea principal es llevar un registro de qué programa está utilizando qué recursos, de otorgar las peticiones de recursos, de contabilizar su uso y de mediar las peticiones en conflicto provenientes de distintos programas y usuarios.

**Semana 6 - Bases de Datos**

**Comprender el concepto de bases de datos.**

La tecnología actual es capaz de almacenar cantidades enormemente grandes de datos, pero esas colecciones son inútiles a menos que seamos capaces de extraer aquellos elementos concretos de información que sean pertinentes para la tarea que tenemos entre manos.

Una base de datos es un sistema que convierte un conjunto de datos de gran tamaño en una herramienta abstracta, permitiendo al usuario buscar y extraer elementos pertinentes de información, de una forma cómoda para él. Esos sistemas aplican la abstracción con el fin de convertir grandes conglomerados de datos en fuentes de información útil.

El término base de datos hace referencia a un conjunto de datos multidimensional en el sentido de que los enlaces internos existentes entre los distintos elementos hacen que se pueda acceder a la información con diversas perspectivas.

**Conocer acerca de Lenguajes de bases de datos.**

Un sistema de bases de datos proporciona un lenguaje de definición de datos para especificar el esquema de la base de datos y un lenguaje de manipulación de datos para expresar las consultas a la base de datos y las modificaciones. En la práctica, los lenguajes de definición y manipulación de datos no son dos lenguajes separados; en su lugar simplemente forman partes de un único lenguaje de bases de datos, tal como SQL, ampliamente usado.

**Lenguaje de definición de datos**

Un esquema de base de datos se especifica mediante un conjunto de definiciones expresadas mediante un lenguaje especial llamado lenguaje de definición de datos (LDD). Especificamos el almacenamiento y los métodos de acceso usados por el sistema de bases de datos por un conjunto de instrucciones en un tipo especial de LDD denominado lenguaje de almacenamiento y definición de datos. Estas instrucciones definen los detalles de implementación de los esquemas de base de datos, que se ocultan usualmente a los usuarios.

Los valores de datos almacenados en la base de datos deben satisfacer ciertas restricciones de consistencia. Por ejemplo, supóngase que el saldo de una cuenta no debe caer por debajo de $500. El LDD proporciona facilidades para especificar tales restricciones. Los sistemas de bases de datos comprueban estas restricciones cada vez que se actualiza la base de datos.

**Lenguaje de manipulación de datos**

La manipulación de datos es:

* La recuperación de información almacenada en la base de datos.
* La inserción de información nueva en la base de datos.
* El borrado de información de la base de datos.
* La modificación de información almacenada en la base de datos.

Un lenguaje de manipulación de datos (LMD) es un lenguaje que permite a los usuarios acceder o manipular los datos organizados mediante el modelo de datos apropiado. Hay dos tipos básicamente:

* LMDs procedimentales. Requieren que el usuario especifique qué datos se necesitan y cómo obtener esos datos.
* LMDs declarativos (también conocidos como LMDs no procedimentales). Requieren que el usuario especifique qué datos se necesitan sin especificar cómo obtener esos datos.

La parte de un LMD que implica recuperación de información se llama lenguaje de consultas. Aunque técnicamente sea incorrecto, en la práctica se usan los términos lenguaje de consultas y lenguaje de manipulación de datos como sinónimos.

**Definir Sistemas de Gestión de Bases de Datos (SGBD).**

Un **sistema de gestión de bases de datos**(SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos.

El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos.

Una aplicación de bases de datos típica consta de varias capas de software, que agruparemos en dos capas principales: una capa de aplicación y una capa de gestión de base de datos. El software de aplicación gestiona la comunicación con el usuario de la base de datos y puede ser bastante complejo, como ilustran las aplicaciones en las que los usuarios acceden a una base de datos por medio de un sitio web. En tal caso, la capa completa de aplicación está compuesta por clientes dispersos por Internet y por un servidor que utiliza la base de datos con el fin de satisfacer las solicitudes de los clientes.

El software de aplicación no manipula directamente la base de datos. La manipulación real de la misma se lleva a cabo mediante el sistema de gestión de base de datos (DBMS, Database Management System). Una vez que el software de aplicación ha determinado qué acción está solicitando el usuario, utiliza el DBMS como herramienta abstracta para obtener el resultado. Si la solicitud consiste en añadir o borrar datos, será el DBMS el que modifique en la práctica la base de datos. Si la solicitud consiste en extraer información, será el DBMS el que lleve a cabo las búsquedas requeridas. Esta dicotomía entre el software de aplicación y el DBMS tiene varias ventajas.

* Permite la construcción y uso de herramientas abstractas
* Separar el software de aplicación del DBMS, proporciona un medio de controlar el acceso a la base de datos.
* Independencia de los datos, la capacidad de modificar la organización de la propia base de datos sin tener que modificar el software de aplicación.

**Poder diferenciar entre usuarios y administradores de bases de datos.**

Un objetivo principal de un sistema de bases de datos es recuperar información y almacenar nueva información en la base de datos. Las personas que trabajan con una base de datos se pueden catalogar como usuarios de bases de datos o como administradores de bases de datos.

**Usuarios de bases de datos e interfaces de usuario**

Hay cuatro tipos diferentes de usuarios de un sistema de base de datos, diferenciados por la forma en que ellos esperan interactuar con el sistema.

* **Usuarios normales**. Interactúan con el sistema mediante la invocación de alguno de los programas de aplicación. Por ejemplo, un cajero bancario que necesita transferir $5000 de la cuenta A a la cuenta B invoca un programa llamado transferir. Este programa pide al cajero el importe de dinero a transferir, la cuenta de la que el dinero va a ser transferido y la cuenta a la que el dinero va a ser transferido.
* **Programadores de aplicaciones.** Son profesionales informáticos que escriben programas de aplicación. Los programadores de aplicaciones pueden elegir entre muchas herramientas para desarrollar interfaces de usuario.
* **Los usuarios sofisticados** interactúan con el sistema sin programas escritos. En su lugar, ellos forman sus consultas en un lenguaje de consulta de bases de datos.
* **Usuarios especializados.** Usuarios sofisticados que escriben aplicaciones de bases de datos especializadas que no son adecuadas en el marco de procesamiento de datos tradicional. Entre estas aplicaciones están los sistemas de diseño asistido por computador, sistemas de bases de conocimientos y sistemas expertos, sistemas que almacenan los datos con tipos de datos complejos (por ejemplo, datos gráficos y datos de audio) y sistemas de modelado del entorno.

 Una de las principales razones de usar SGBDs es tener un control centralizado tanto de los datos como de los programas que acceden a esos datos. La persona que tiene este control central sobre el sistema se llama **administrador de la base de datos**(DBA). Las funciones del DBA incluyen las siguientes:

* **Definición del esquema.** El DBA crea el esquema original de la base de datos escribiendo un conjunto de instrucciones de definición de datos en el LDD.
* **Definición de la estructura y del método de acceso**.
* **Modificación del esquema y de la organización física.** Los DBA realizan cambios en el esquema y en la organización física para reflejar las necesidades cambiantes de la organización, o para alterar la organización física para mejorar el rendimiento.
* **Concesión de autorización para el acceso a los datos.** La concesión de diferentes tipos de autorización permite al administrador de la base de datos determinar a qué partes de la base de datos puede acceder cada usuario. La información de autorización se mantiene en una estructura del sistema especial que el sistema de base de datos consulta cuando se intenta el acceso a los datos en el sistema.
* **Mantenimiento rutinario**. Algunos ejemplos de actividades rutinarias de mantenimiento del administrador de la base de datos son:
* Copia de seguridad periódica de la base de datos para prevenir la pérdida de datos en caso de desastres.
* Asegurarse de que haya suficiente espacio libre en disco para las operaciones normales y aumentar el espacio en disco según sea necesario.
* Supervisión de los trabajos que se ejecuten en la base de datos y asegurarse de que el rendimiento no se degrada por tareas muy costosas iniciadas por algunos usuarios.

**Conocer diferentes modelos de bases de datos.**

Bajo la estructura de la base de datos se encuentra el **modelo de datos**: una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

Los diferentes modelos de datos que se han propuesto se clasifican en tres grupos diferentes: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos.

**Modelo entidad-relación**

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados *entidades*, y de *relaciones*entre estos objetos. Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades. Las entidades se describen en una base de datos mediante un conjunto de **atributos**. Por ejemplo, los atributos *número-cuenta*y *saldo*describen una cuenta particular de un banco y pueden ser atributos del conjunto de entidades *cuenta*. Análogamente, los atributos *nombre-cliente*, *calle-cliente*y *ciudad-cliente*pueden describir una entidad *cliente*.

Un atributo extra, *id-cliente*, se usa para identificar unívocamente a los clientes (dado que puede ser posible que haya dos clientes con el mismo nombre, dirección y ciudad. Se debe asignar un identificador único de cliente a cada cliente.

Una **relación**es una asociación entre varias entidades. Por ejemplo, una relación *impositor*asocia un cliente con cada cuenta que tiene. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente **conjunto de entidades**y **conjunto de relaciones**.

Además de entidades y relaciones, el modelo E-R representa ciertas restricciones que los contenidos de la base de datos deben cumplir. Una restricción importante es la *correspondencia de cardinalidades*, que expresa el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones. Por ejemplo, si cada cuenta puede pertenecer sólo a un cliente,  el modelo puede expresar esta restricción.

**Modelo relacional**

En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único.

El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro. No es difícil ver cómo se pueden almacenar las tablas en archivos. Por ejemplo, un carácter especial (como una coma) se puede usar para delimitar los diferentes atributos de un registro, y otro carácter especial (como un carácter de nueva línea) se puede usar para delimitar registros. El modelo relacional oculta tales detalles de implementación de bajo nivel a los desarrolladores de bases de datos y usuarios. El modelo de datos relacional es el modelo de datos más ampliamente usado, y una amplia mayoría de sistemas de bases de datos actuales se basan en el modelo relacional.

El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R. Los diseños de bases de datos a menudo se realizan en el modelo E-R, y después se traducen al modelo relacional.

**Otros modelos de datos**

El **modelo de datos orientado a objetos**es otro modelo de datos que está recibiendo una atención creciente El modelo orientado a objetos se puede observar como una extensión del modelo E-R con las nociones de encapsulación, métodos (funciones) e identidad de objeto.

El **modelo de datos relacional orientado a objetos**combina las características del modelo de datos orientado a objetos y el modelo de datos relacional.

Los modelos de datos semiestructurados permiten la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos. Esto es diferente de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos.

Históricamente, otros dos modelos de datos, el **modelo de datos de red**y el **modelo de datos jerárquico**, precedieron al modelo de datos relacional. Estos modelos estuvieron ligados fuertemente a la implementación subyacente y complicaban la tarea del modelado de datos. Como resultado se usan muy poco actualmente, excepto en el código de bases de datos antiguo que aún está en servicio en algunos lugares.

**Semana 7 - Bases de Datos**

**Conocer diferentes modelos de bases de datos.**

Bajo la estructura de la base de datos se encuentra el **modelo de datos**: una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

Los diferentes modelos de datos que se han propuesto se clasifican en tres grupos diferentes: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos.

**Modelo entidad-relación**

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados *entidades*, y de *relaciones*entre estos objetos. Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades. Las entidades se describen en una base de datos mediante un conjunto de **atributos**. Por ejemplo, los atributos *número-cuenta*y *saldo*describen una cuenta particular de un banco y pueden ser atributos del conjunto de entidades *cuenta*. Análogamente, los atributos *nombre-cliente*, *calle-cliente*y *ciudad-cliente*pueden describir una entidad *cliente*.

Un atributo extra, *id-cliente*, se usa para identificar unívocamente a los clientes (dado que puede ser posible que haya dos clientes con el mismo nombre, dirección y ciudad. Se debe asignar un identificador único de cliente a cada cliente.

Una **relación**es una asociación entre varias entidades. Por ejemplo, una relación *impositor*asocia un cliente con cada cuenta que tiene. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente **conjunto de entidades**y **conjunto de relaciones**.

Además de entidades y relaciones, el modelo E-R representa ciertas restricciones que los contenidos de la base de datos deben cumplir. Una restricción importante es la *correspondencia de cardinalidades*, que expresa el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones. Por ejemplo, si cada cuenta puede pertenecer sólo a un cliente,  el modelo puede expresar esta restricción.

**Modelo relacional**

En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único.

El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro. No es difícil ver cómo se pueden almacenar las tablas en archivos. Por ejemplo, un carácter especial (como una coma) se puede usar para delimitar los diferentes atributos de un registro, y otro carácter especial (como un carácter de nueva línea) se puede usar para delimitar registros. El modelo relacional oculta tales detalles de implementación de bajo nivel a los desarrolladores de bases de datos y usuarios. El modelo de datos relacional es el modelo de datos más ampliamente usado, y una amplia mayoría de sistemas de bases de datos actuales se basan en el modelo relacional.

El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R. Los diseños de bases de datos a menudo se realizan en el modelo E-R, y después se traducen al modelo relacional.

**Otros modelos de datos**

El **modelo de datos orientado a objetos**es otro modelo de datos que está recibiendo una atención creciente El modelo orientado a objetos se puede observar como una extensión del modelo E-R con las nociones de encapsulación, métodos (funciones) e identidad de objeto.

El **modelo de datos relacional orientado a objetos**combina las características del modelo de datos orientado a objetos y el modelo de datos relacional.

Los modelos de datos semiestructurados permiten la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos. Esto es diferente de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos.

Históricamente, otros dos modelos de datos, el **modelo de datos de red**y el **modelo de datos jerárquico**, precedieron al modelo de datos relacional. Estos modelos estuvieron ligados fuertemente a la implementación subyacente y complicaban la tarea del modelado de datos. Como resultado se usan muy poco actualmente, excepto en el código de bases de datos antiguo que aún está en servicio en algunos lugares.

**Bases de datos NoSQL**

Las bases de datos NoSQL (Not Only SQL) difieren de las bases de datos relacionales en su enfoque de almacenamiento y manejo de datos. Una base datos NoSQL permite el almacenamiento y manipulación de datos no estructurados o semi-estructurados, como documentos, grafos, clave-valor y columnas, en contraposición a las tablas relacionales típicas de las bases de datos SQL. Están diseñadas para escalar de manera más eficiente que las bases de datos relacionales, lo que significa que pueden manejar grandes volúmenes de datos distribuidos en múltiples servidores.

Los principales tipos de bases de datos NoSQL incluyen bases de datos de documentos (como MongoDB), bases de datos de grafos (como Neo4j), bases de datos de clave-valor (como Redis) y bases de datos de columnas (como Apache Cassandra). Estas bases de datos ofrecen una alternativa flexible y escalable a las bases de datos relacionales.

En el mosaico de está semana encuentran enlaces de diferentes proveedores de bases de datos donde explican las bases de datos NoSQL para quienes quieran profundizar en el tema.

**Semana 8 - Redes de datos**

Nuestro **objetivo** de esta semana será cono**conocer acerca de:**

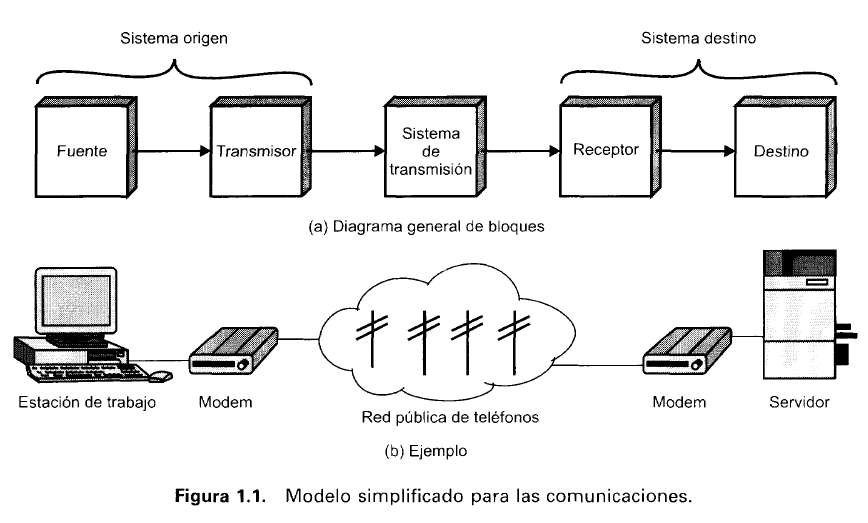
**Conceptos básicos y modelos de referencia**

**Medios de transmisión**

**Dispositivos y cableado**

Para cumplir estos [objetivos](https://campus.uner.edu.ar/fcad/mod/page/view.php?id=24602) vamos a realizar distintas lecturas y discutir algunos puntos en clases.

**Conceptos básicos y modelos de referencia**



* La **fuente**. Este dispositivo genera los datos a transmitir. Ejemplos de fuentes pueden ser un teléfono o un computador personal.
* El **transmisor**. Normalmente los datos generados por la fuente no se transmiten directamente tal y como son generados. Al contrario, el transmisor transforma y codifica la información, generando señales electromagnéticas susceptibles de ser transmitidas a través de algún sistema de transmisión. Por ejemplo, un módem convierte las cadenas de bits generadas por un computador personal y las transforma en señales analógicas que pueden ser transmitidas a través de la red de telefonía.
* El **sistema de transmisión**. Puede ser desde una sencilla línea de transmisión hasta una compleja red que conecte a la fuente con el destino.
* El **receptor**. El receptor acepta la señal proveniente del sistema de transmisión y la transforma de tal manera que pueda ser manejada por el dispositivo de destino. Por ejemplo, un módem captará la señal analógica de la red o línea de transmisión y la convertirá en una cadena de bits.
* El**destino**. Toma los datos del receptor.

**Modelos de referencia**

Los modelos de referencia OSI y TCP/IP tienen mucho en común. Ambos se basan en el concepto de una pila de protocolos independientes. Además, la funcionalidad de las capas es muy similar. A pesar de estas similitudes fundamentales, los dos modelos también tienen muchas diferencias.

**Interfaz de usuario gráfica, Diagrama, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**

**El modelo de referencia OSI**

Este modelo se basa en una propuesta desarrollada por la Organización Internacional de Normas (iso) como el primer paso hacia la estandarización internacional de los protocolos utilizados en las diversas capas (Day y Zimmerman, 1983). Este modelo se revisó en 1995 (Day, 1995) y se le llama Modelo de referencia OSI (Interconexión de Sistemas Abiertos, del inglés Open Systems Interconnection) de la iso puesto que se ocupa de la conexión de sistemas abiertos; esto es, sistemas que están abiertos a la comunicación con otros sistemas.

El modelo OSI tiene siete capas.

Los principios que se aplicaron para llegar a las siete capas se pueden resumir de la siguiente manera:

1. Se debe crear una capa en donde se requiera un nivel diferente de abstracción.
2. Cada capa debe realizar una función bien definida.
3. La función de cada capa se debe elegir teniendo en cuenta la definición de protocolos estandarizados internacionalmente.
4. Es necesario elegir los límites de las capas de modo que se minimice el flujo de información a través de las interfaces.
5. La cantidad de capas debe ser suficiente como para no tener que agrupar funciones distintas en la misma capa; además, debe ser lo bastante pequeña como para que la arquitectura no se vuelva inmanejable.

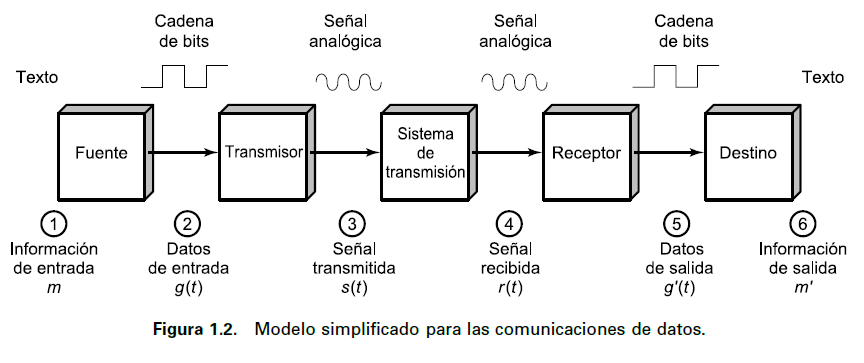
Tenga en cuenta que el modelo OSI en sí no es una arquitectura de red, ya que no especifica los servicios y protocolos exactos que se van a utilizar en cada capa. Sólo indica lo que cada una debe hacer. Sin embargo, la ISO también ha elaborado estándares para todas las capas, aunque no son parte del modelo de referencia en sí. Cada uno se publicó como un estándar internacional separado. Aunque el modelo (en parte) es muy usado, los protocolos asociados han estado en el olvido desde hace tiempo.

**El modelo de referencia TCP/IP**

El Modelo de referencia TCP/IP, lleva este nombre debido a sus dos protocolos primarios, se definió por primera vez en 1974; después se refinó y definió como estándar en la comunidad de Internet (Braden, 1989).  Uno de los [objetivos](https://campus.uner.edu.ar/fcad/mod/page/view.php?id=24602) principales fue que la red pudiera sobrevivir a la pérdida de hardware de la subred sin que se interrumpieran las conversaciones existentes. En otras palabras, se buscaba que las conexiones permanecieran intactas mientras las máquinas de origen y de destino estuvieran funcionando, incluso aunque algunas de las máquinas o líneas de transmisión en el trayecto dejaran de funcionar en forma repentina. Además, como se tenían en mente aplicaciones con requerimientos divergentes que abarcaban desde la transferencia de archivos hasta la transmisión de voz en tiempo real, se necesitaba una arquitectura flexible.

Como dijimos antes, la fortaleza del modelo de referencia OSI es el modelo en sí (excepto las capas de presentación y de sesión), el cual ha demostrado ser excepcionalmente útil para hablar sobre redes de computadoras. En contraste, la fortaleza del modelo de referencia TCP/IP son los protocolos, que se han utilizado mucho durante varios años.

**Medios de transmisión**

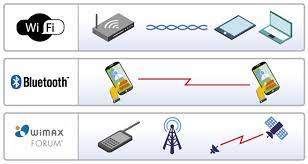
****

La transmisión de datos entre un emisor y un receptor siempre se realiza a través de un medio de transmisión. Los medios de transmisión se pueden clasificar como **guiados** y **no guiados**. En ambos casos, la comunicación se realiza usando ondas electromagnéticas. En los medios guiados, por ejemplo en pares trenzados, en cables coaxiales y en fibras ópticas, las ondas se transmiten confinándolas a lo largo de un camino físico. Por el contrario, los medios no guiados, también denominados inalámbricos, proporcionan un medio para transmitir las ondas electromagnéticas sin confinarlas, como por ejemplo en la propagación a través del aire, el mar o el vacío.

Un medio de transmisión guiado es **punto a punto** si proporciona un enlace directo entre dos dispositivos que comparten el medio, no existiendo ningún otro dispositivo conectado. En una configuración guiada **multipunto**, el mismo medio es compartido por más de dos dispositivos.

**Dispositivos y cableado**

**Escala de tiempo

Descripción generada automáticamente  
**

**Semana 9 - Redes de datos - Fundamentos básicos de la Web**

**Unidad 6: Redes de datos**

* Conceptos básicos y modelos de referencia.
* Medios de transmisión.
* Dispositivos y cableado.
* **Modelo TCP/lP.**
* **Redes LAN, MAN Y WAN.**

Nuestro objetivo de esta semana será **conocer acerca de estos conceptos**y por otra parte comenzaremos a ver algunas cuestiones de la unidad siguiente **Fundamentos básicos de la Web**

**Modelos TCP/IP**

La arquitectura de protocolos TCP/IP es resultado de la investigación y desarrollo llevados a cabo en la red experimental de conmutación de paquetes ARPANET, financiada por la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA, Defense Advanced Research Projects Agency), y se denomina globalmente como la familia de protocolos TCP/IP. Esta familia consiste en una extensa colección de protocolos que se han especificado como estándares de Internet por parte de IAB (Internet Architecture Board).

**LAS CAPAS DE TCP/IP**

Captura de pantalla de un celular con letras

Descripción generada automáticamente

El **nivel de acceso a la red** es responsable del intercambio de datos entre el sistema final (servidor, estación de trabajo, etc.) y la red a la cual está conectado. El emisor debe proporcionar a la red la dirección del destino, de tal manera que ésta pueda encaminar los datos hasta el destino apropiado. El emisor puede requerir ciertos servicios que pueden ser proporcionados por el nivel de red, por ejemplo, solicitar una determinada prioridad. El software en particular que se use en esta capa dependerá del tipo de red que se disponga. Así, se han desarrollado, entre otros, diversos estándares para la conmutación de circuitos, la conmutación de paquetes (por ejemplo, retransmisión de tramas) y para las redes de área local (por ejemplo, Ethernet).  Haciendo esto, el software de comunicaciones situado por encima de la capa de acceso a la red no tendrá que ocuparse de los detalles específicos de la red a utilizar.

El software de las capas superiores debería, por tanto, funcionar correctamente con independencia de la red a la que el computador esté conectado.

El nivel de internet es el encargado de una serie de procedimientos que permiten que los datos atraviesen las distintas redes interconectadas, en aquellas situaciones en las que los dos dispositivos (emisor y receptor) están conectados a redes diferentes. El protocolo IP (Internet Protocol) se utiliza en esta capa para ofrecer el servicio de encaminamiento a través de varias redes. Este protocolo se implementa tanto en los sistemas finales como en los encaminadores intermedios. Un encaminador es un procesador que conecta dos redes y cuya función principal es retransmitir datos desde una red a otra siguiendo la ruta adecuada para alcanzar al destino.

El nivel de transporte es el encargado de  que los datos se intercambien de forma fiable. Es deseable asegurar que todos los datos llegan a la aplicación destino y en el mismo orden en el que fueron enviados. El protocolo para el control de la transmisión, TCP (Transmission Control Protocol), es el más utilizado para proporcionar esta funcionalidad.

El nivel de aplicación contiene toda la lógica necesaria para posibilitar las distintas aplicaciones de usuario. Para cada tipo particular de aplicación, como por ejemplo, la transferencia de archivos, se necesitará un módulo bien diferenciado.

**Redes LAN, MAN Y WAN**

Una red de área local (LAN, Local Area Networks) es una red de comunicaciones que interconecta varios dispositivos y proporciona un medio para el intercambio de información entre ellos. Para las LAN hay muy diversas configuraciones. De entre ellas, las más habituales son las LAN conmutadas y las LAN inalámbricas. Dentro de las conmutadas, las más populares son las LAN Ethernet, constituidas por un único conmutador, o, alternativamente, implementadas mediante un conjunto de conmutadores interconectados entre sí.

Las redes de área metropolitana (MAN, Metropolitan Area Network) están entre las LAN y las WAN. El interés en las MAN ha surgido tras ponerse de manifiesto que las técnicas tradicionales de conmutación y conexión punto a punto usadas en WAN, pueden ser no adecuadas para las necesidades crecientes de ciertas organizaciones. Mientras que la retransmisión de tramas y ATM prometen satisfacer un amplio espectro de necesidades en cuanto a velocidades de transmisión, hay situaciones, tanto en redes privadas como públicas, que demandan gran capacidad a coste reducido en áreas relativamente grandes. Para tal fin se han implementado una serie de soluciones, como por ejemplo las redes inalámbricas o las extensiones metropolitanas de Ethernet. El principal mercado para las MAN lo constituyen aquellos clientes que necesitan alta capacidad en un área metropolitana. Las MAN están concebidas para satisfacer estas necesidades de capacidad a un coste reducido y con una eficacia mayor que la que se obtendría mediante una compañía local de telefonía para un servicio equivalente.

Las redes de área amplia (WAN, Wide Area Networks) son todas aquellas que cubren una extensa área geográfica, requieren atravesar rutas de acceso público y utilizan, al menos parcialmente, circuitos proporcionados por una entidad proveedora de servicios de telecomunicación. Generalmente, una WAN consiste en una serie de dispositivos de conmutación interconectados. La transmisión generada por cualquier dispositivo se encaminará a través de estos nodos internos hasta alcanzar el destino. A estos nodos (incluyendo los situados en los contornos) no les concierne el contendido de los datos, al contrario, su función es proporcionar el servicio de conmutación, necesario para transmitir los datos de nodo en nodo hasta alcanzar su destino final. Tradicionalmente, las WAN se han implementado usando una de las dos tecnologías siguientes: conmutación de circuitos y conmutación de paquetes. Últimamente, se está empleando como solución la técnica de retransmisión de tramas ( frame relay), así como las redes ATM.

**Internet**

Internet tiene su origen en una serie de proyectos de investigación que datan de principios de la década de 1960. El objetivo era desarrollar la capacidad de enlazar diversas redes de computadoras, para que pudieran funcionar como un sistema conectado que no se viera afectado en su operación global por desastres de escala local. Buena parte de este trabajo estaba patrocinado por el gobierno de Estados Unidos a través de la agencia DARPA (Defense Advanced Research Projects Agency, Agencia de Proyectos Avanzados de Investigación para la Defensa). A lo largo de los años, el desarrollo de Internet fue pasando de ser un proyecto patrocinado por el gobierno a un proyecto académico de investigación y actualmente es, en buena medida, un esfuerzo de carácter comercial que enlaza una combinación mundial de redes LAN, MAN y WAN que abarcan a millones de computadoras.

**La World Wide Web (WWW)**

La World Wide Wev es una aplicación Internet que permite diseminar información multimedia a través de Internet. Está basada en el concepto de hipertexto, un término que originalmente hacía referencia a documentos de texto que contenían enlaces, denominados hipervínculos, a otros documentos. Hoy día, el hipertexto se ha ampliado para incluir también imágenes, audio y vídeo, y a causa de esta expansión de su ámbito de actuación en ocasiones se denomina hipermedia.

Cuando se emplea una interfaz gráfica de usuario, el lector de un documento de hipertexto puede seguir los hipervínculos asociados con el mismo apuntando y haciendo clic con su ratón. Por ejemplo, suponga que en un documento de hipertexto aparece la frase “La interpretación del Bolero de Maurice Ravel por la orquesta fue excepcional” y que el nombre Maurice Ravel está vinculado a otro documento, que quizá proporcione información acerca del compositor. Un lector podría elegir ver ese material asociado apuntando al nombre Maurice Ravel con el ratón y haciendo clic en el botón del ratón. Además, si se instalan los hipervínculos apropiados, el lector podría escuchar una grabación de audio del concierto haciendo clic en la palabra Bolero.

De esta forma, un lector de documentos de hipertexto puede explorar documentos relacionados o seguir la línea de pensamiento de un documento a otro. A medida que se vinculan partes de documentos con otros documentos se va desarrollando una especie de tela de araña de información relacionada. Cuando se implementa en una red de computadoras este tipo de sistema, los documentos que componen esa “tela de araña” pueden residir en diferentes máquinas, formando una red de documentos global. Esa red de documentos que ha evolucionado en Internet tiene alcance mundial y se conoce como World Wide Web (también se la denomina WWW, W3 o simplemente Web).

A un documento de hipertexto en la World Wide Web a menudo se le llama página web. Un conjunto de páginas web estrechamente relacionadas se conoce con el nombre de sitio web. La World Wide Web tiene su origen en el trabajo de Tim Berners-Lee, que se dio cuenta del potencial que presentaba la combinación del concepto de documentos vinculados con la tecnología de interredes y que desarrolló el primer software para implementar la WWW en diciembre de 1990.

**Semana 10 - Fundamentos básicos de la web**

**Unidad 7: Fundamentos básicos de la Web**

* Internet.
* World Wide Web (W.W.W.).
* **Modelo computacional de la web.**
* **Aplicaciones web.**
* **Puertos.**
* **Sistema de nombres.**
* **Protocolos.**
* **Lenguajes de programación web.**

**Modelo computacional de la web.**

Desde el punto de vista del usuario, la web consiste en una enorme colección de contenido en forma de páginas web, por lo general, conocidas simplemente como páginas. Cada una puede contener vínculos a otras páginas en cualquier lugar del mundo. Para seguir un vínculo, los usuarios pueden hacer clic en él, y a continuación los llevará a la página apuntada. Este proceso se puede repetir de manera indefinida.

Por lo general, las páginas se ven mediante un programa llamado navegador. Firefox, Internet Explorer y Chrome son ejemplos de navegadores populares. El navegador obtiene la página solicitada, interpreta el contenido y despliega la página en pantalla con el formato adecuado. El contenido en sí puede ser una mezcla de texto, imágenes y comandos de formato, ya sea en forma de un documento tradicional u otras formas de contenido, como un video o programas que produzcan una interfaz gráfica con la que puedan interactuar los usuarios.

Diagrama

Descripción generada automáticamente

En la figura se muestra la imagen de una página. Esta página muestra texto y elementos gráficos (que en su mayoría son muy pequeños como para leerlos).

Algunas partes de la página están asociadas con vínculos a otras. A una pieza de texto, icono, imagen u otro elemento asociado con otra página se le conoce como hipervínculo. Para seguir un vínculo, el usuario coloca el cursor del ratón en la parte vinculada del área de la página (lo cual hace que el cursor cambie de forma) y hace clic. Seguir un vínculo es simplemente una forma de decir al navegador que obtenga otra página.

A continuación, el navegador obtiene la nueva página y la despliega en pantalla. Hay docenas de páginas adicionales a las que se puede acceder mediante vínculos desde la primera página. Cualquier otra página puede estar compuesta de contenido en la(s) misma(s) máquina(s) que la primera página, o en máquinas en alguna parte del mundo. El usuario no puede distinguir esto. El navegador se encarga del proceso de obtención de las páginas, sin ninguna ayuda del usuario. De esta forma, el proceso de desplazarse entre máquinas al momento de ver contenido es transparente para el usuario.

En la figura se muestra el modelo básico de la forma en que se despliegan las páginas en pantalla. El navegador despliega una página web en la máquina cliente. Para obtener cada página, se envía una solicitud a uno o más servidores, los cuales responden con el contenido de la página. El protocolo de solicitud-respuesta para obtener páginas es un protocolo simple basado en texto que se ejecuta sobre TCP, como en el caso de SMTP. Este protocolo se llama HTTP (Protocolo de Transferencia de HiperTexto, del inglés HyperText Transfer Protocol). El contenido puede ser simplemente un documento que se lea de un disco, o el resultado de una consulta en una base de datos y la ejecución de un programa.

La página se considera una página estática si es el mismo documento cada vez que se despliega en pantalla. Por el contrario, si se generó bajo demanda mediante un programa o contiene uno, es una página dinámica. Una página dinámica se puede presentar de manera distinta cada vez que se despliega en pantalla. Por ejemplo, la página principal de una tienda electrónica puede ser distinta para cada visitante. Si el cliente de una librería ha comprado novelas de misterio en sus anteriores visitas, es probable que la próxima vez le aparezcan las nuevas novelas de misterio en la página de inicio; mientras que un cliente enfocado hacia lo culinario podría ser recibido con nuevos libros de cocina. En sí, la respuesta involucra el uso de cookies.

**Aplicaciones web**

Las aplicaciones web son herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. Son programas que se codifican en un lenguaje interpretable por los navegadores web. Las aplicaciones web han ganado popularidad debido a los pocos requerimientos necesarios para acceder a un navegador web, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Estas aplicaciones se ejecutan dentro del navegador, y los datos de usuario se almacenan en servidores en centros de datos de Internet. Utilizan protocolos web para acceder a la información a través de Internet, y el navegador para mostrar una interfaz de usuario. La ventaja de este método es que los usuarios no necesitan instalar programas de aplicación separados, y se puede acceder a los datos de usuario desde distintas computadoras, además de que el operador del servicio respalda la información. Estas aplicaciones tienen tanto éxito que rivalizan con el software de aplicación tradicional. Claro que también ayuda el hecho de que los grandes proveedores ofrezcan estas aplicaciones sin costo. Este modelo es la forma común de computación en nube, en la cual la computación pasa de las computadoras de escritorio individuales hacia grupos compartidos de servidores en Internet.

Para que actúen como aplicaciones, las páginas web ya no pueden ser estáticas. Es necesario el contenido dinámico. Por ejemplo, una página del catálogo de una biblioteca debe reflejar los libros disponibles en un momento dado, además de los libros que están prestados y por lo tanto no se encuentran disponibles.

**Puertos**

Para que se pueda llevar a cabo la comunicación extremo a extremo entre cliente y servidor de aplicaciones web necesitamos usar lo que denomina puerto. Hay una serie de puertos bien conocidos que sirven para aplicaciones específicas, y otros puertos que son utilizados por diferentes software como juegos, servicios online etc.

Cualquier aplicación o servicio que usemos, desde videojuegos, gestores de correo electrónico, mensajería instantánea o incluso el propio sistema operativo, siempre tienen una serie de puertos abiertos transmitiendo o escuchando lo que sucede a su alrededor.

Los puertos pueden ser de dos tipos diferentes, dependiendo del protocolo de la capa de transporte que se utilice. El protocolo TCP es un protocolo conectivo, fiable y orientado a conexión, se encarga de que lleguen todos los segmentos correctamente, y realiza las retransmisiones necesarias en caso de que ocurra algún tipo de problema, además, también garantiza el orden, por lo que las capas superiores no tienen que encargarse de ello. UDP es un protocolo no orientado a conexión, no es fiable, no garantiza que lleguen los paquetes ni tampoco su orden, para ello las capas superiores (capa de aplicación) garantizarán esto.

En el siguiente [enlace](https://www.redeszone.net/tutoriales/configuracion-puertos/puertos-tcp-udp/) podrán ver los puertos más utilizados.

**Sistema de nombres**

Aunque en teoría los programas pueden hacer referencia a páginas web, buzones de correo y otros recursos mediante las direcciones de red (por ejemplo, IP) de las computadoras en las que se almacenan, a las personas se les dificulta recordar estas direcciones. Por este motivo se introdujeron nombres legibles de alto nivel con el fin de separar los nombres de máquina de las direcciones de máquina. De esta manera, el servidor web de la empresa podría conocerse como [www.cs.washington.edu](http://www.cs.washington.edu/) sin importar cuál sea su dirección IP. Sin embargo, como la red sólo comprende direcciones numéricas, se requiere algún mecanismo para convertir los nombres en direcciones de red.

Por esto en 1983 se inventó el DNS (Sistema de Nombres de Dominio, del inglés Domain Name System), el cual ha sido una parte clave de Internet desde entonces. La esencia del DNS es la invención de un esquema jerárquico de nombres basado en dominios y un sistema de base de datos distribuido para implementar este esquema de nombres. El DNS se usa principalmente para asociar los nombres de host con las direcciones IP, pero también se puede usar para otros fines.

El DNS se define en los RFC 1034, 1035, 2181 y se elabora con más detalle en muchos otros. Dicho en forma muy breve, la forma en que se utiliza el DNS es la siguiente. Para asociar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado resolvedor y le pasa el nombre como parámetro. El cual envía una consulta que contiene el nombre a un servidor DNS local, que después busca el nombre y devuelve una respuesta con la dirección IP al resolvedor, que a su vez lo devuelve al solicitante. Los mensajes de solicitud y respuesta se envían como paquetes UDP. Armado con la dirección IP, el programa puede entonces establecer una conexión TCP con el host o enviarle paquetes UDP.

**Protocolos**

Los protocolos de nivel de aplicaciones oficiales de Internet incluyen:

* Domain Name Protocol (Protocolo de nombres de dominio)
* Exterior Gateway Protocol (Protocolo de pasarela exterior)
* File Transfer Protocol (Protocolo de transferencia de archivos)
* Name/Finger Protocol (Protocolo de nombres/finger)
* Telnet Protocol (Protocolo Telnet)
* Trivial File Transfer Protocol (Protocolo de transferencia de archivos trivial)

TCP/IP implementa otros protocolos de nivel superior que no son protocolos oficiales de Internet pero que se utilizan comúnmente en la comunidad de Internet a nivel de programa de aplicación. Estos protocolos incluyen:

* Protocolo de red local DCN (Distributed Computer Network)
* Protocolo de ejecución remota de mandatos
* Protocolo de inicio de sesión remoto
* Protocolo de shell remoto
* Protocolo Wake On LAN
* Protocolo de información de direccionamiento
* Protocolo de servidor horario

[**Lenguajes de programación web**](https://campus.uner.edu.ar/fcad/mod/url/view.php?id=24731)

Los [lenguajes de programación web](https://campus.uner.edu.ar/fcad/mod/url/view.php?id=24731) pueden programar instrucciones y operaciones lógicas complejas. Mientras que con los lenguajes de marcado como HTML solo es posible generar documentos, los lenguajes de programación te permiten crear programas de cualquier tamaño adaptados a tus necesidades.  
**Frontend**

La parte de un sitio web con la que el usuario interactúa directamente se denomina interfaz. También se conoce como el "lado del cliente" de la aplicación. Incluye todo lo que los usuarios experimentan directamente: colores y estilos de texto, imágenes, gráficos y tablas, botones, colores y menú de navegación. HTML, CSS y JavaScript son los lenguajes utilizados para el desarrollo de Front End. Los desarrolladores front-end implementan la estructura, el diseño, el comportamiento y el contenido de todo lo que se ve en las pantallas del navegador cuando se abren sitios web, aplicaciones web o aplicaciones móviles.

**Backend**

El backend es el lado del servidor del sitio web. Almacena y organiza los datos, y también se asegura de que todo en el lado del cliente del sitio web funcione correctamente. Es la parte del sitio web que no se puede ver y con la que no puede interactuar. Es la parte del software que no entra en contacto directo con los usuarios. Los usuarios acceden indirectamente a las partes y características desarrolladas por los diseñadores de back-end a través de una aplicación de front-end. Las actividades, como escribir API, crear bibliotecas y trabajar con componentes del sistema sin interfaces de usuario o incluso sistemas de programación científica, también se incluyen en el backend.