

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**

**Avance 1**

**Diseño y construcción de un sistema electrónico para el control de acceso biométrico de huella dactilar gestionado por plataforma WEB.**

**Alumno : Juan Sotomayor Chacón**

**Carrera : Ingeniería civil industrial en electrónica**

**Profesor Guía : Jorge Gianotti Hidalgo**

**Antofagasta, Mayo de 2016**

**ÍNDICE**

[LISTA DE ACRÓNIMOS 1](#_Toc433889623)

[CAPÍTULO 1: INTRODUCCIÓN 1](#_Toc433889624)

[1.1 OBJETIVOS 1](#_Toc433889625)

[1.1.1 Objetivo general 1](#_Toc433889626)

[1.1.2 Objetivos específicos 1](#_Toc433889627)

[1.2 RESUMEN 1](#_Toc433889628)

[1.3 METODOLOGÍA 1](#_Toc433889629)

[CAPITÚLO 2: APRENDIZAJE Y DOMINIO DE LAS VARIABLES INVOLUCRADAS. 1](#_Toc433889630)

[2.1 DEFINICIONES IMPORTANTES 1](#_Toc433889631)

[2.1.1 LENGUAJE DE PROGRAMACIÓN: 1](#_Toc433889632)

[2.1.2 LENGUAJE INFORMÁTICO: 1](#_Toc433889633)

[2.1.3 LENGUAJE DE PROGRAMACIÓN C 1](#_Toc433889634)

[2.1.4 LENGUAJE DE PROGRAMACIÓN C++ 1](#_Toc433889635)

[2.1.5 PRINCIPALES DIFERENCIAS ENTRE C Y C++ 1](#_Toc433889636)

[2.1.6 LENGUAJE DE PROGRAMACIÓN HTML 1](#_Toc433889637)

[2.2 CURSO DE PROGRAMACIÓN C: 1](#_Toc433889638)

[2.2.1 VIDEOS EXPLICATIVOS PROGRAMACIÓN C: 1](#_Toc433889639)

[2.2.2 VIDEOS EXPLICATIVOS PROGRAMACIÓN C++: 1](#_Toc433889640)

[2.2.3 VIDEOS EXPLICATIVOS HTML: 1](#_Toc433889641)

[2.3 DEFINICIONES IMPORTANTES 1](#_Toc433889642)

[2.3.1 LENGUAJE DE PROGRAMACIÓN XHTML: 1](#_Toc433889643)

[2.3.2 DIFERENCIAS ENTRE HTML Y XHTML: 1](#_Toc433889644)

[2.3.3 LENGUAJE DE PROGRAMACIÓN JAVASCRIPT: 1](#_Toc433889645)

[2.3.4 USO DE JAVASCRIPT EN PÁGINAS WEB: 1](#_Toc433889646)

[2.4 CURSO TUTELLUS: 1](#_Toc433889647)

[2.4.1 VIDEOS EXPLICATIVOS XHTML: 1](#_Toc433889648)

[2.4.2 VIDEOS EXPLICATIVOS JAVASCRIPT: 1](#_Toc433889649)

[CAPÍTULO 3: CREACIÓN Y MANEJO WEB EN BASE A LO APRENDIDO. 1](#_Toc433889650)

[3.1 SOFTWARE NOTEPAD++. 1](#_Toc433889651)

[3.1.1 PROGRAMACIÓN BÁSICA EN NOTEPAD DE LA PLATAFORMA A DISEÑADA 1](#_Toc433889652)

[CAPITULO 4 : IMPLEMENTACIÓN DEL HOST PARA VISUALIZACIÓN EN INTERNET. 1](#_Toc433889653)

[4.1 SOFTWARE CPANEL: 1](#_Toc433889654)

[4.1.1 ENLAZAMIENTO PLATAFORMA COMPUTACIONAL DISEÑADA Y DOMINIO WEB. 1](#_Toc433889655)

[4.2 SOFTWARE FILEZILLA: 1](#_Toc433889656)

[4.3 ESTRUCTURA DE LA PÁGINA WEB DISEÑADA. 1](#_Toc433889657)

[4.3.1 HEADER O CABECERA: 1](#_Toc433889658)

[4.3.2 BODY O CUERPO: 1](#_Toc433889659)

[4.3.4 PESTAÑA SOBRE: 1](#_Toc433889660)

[4.3.5 PESTAÑA CONTACTO: 1](#_Toc433889661)

[CAPITULO 5: RESULTADOS ESPERADOS Y MEJORAS A CORTO PLAZO 1](#_Toc433889662)

[5.1 RESULTADOS ESPERADOS 1](#_Toc433889663)

[5.2 MEJORAS A CORTO PLAZO 1](#_Toc433889664)

[REFERENCIAS BIBLIOGRÁFICAS 1](#_Toc433889665)

LISTA DE ACRÓNIMOS

HTTP: Hypertext Transfer Protocol.

PC: Personal Computer.

LAN: Local Area Network.

WAN: Wide Area Network.

CCD: Charged Coupled Device.

RTC: Real Time Clock

RFID: Radio Frequency Identification

GPIO: General Purpose Input/Output

IMEI: International Mobile Equipment Identity.

IP: Internet Protocol.

TCP: Transmission Control Protocol.

HTTP: Hypertext Transfer Protocol.

UART: Universal Asynchronous Receiver Transmitter.

USB: Universal Serial Bus.

ASCII: American Standard Code for Information Interchange.

PHP: Hipertext Preprocessor

HTML: Hiper Text Markup Language.

WWW: World Wide Web.

DOCTYPE: Declaración de documento para creación de una página web.

# **CAPÍTULO 1: INTRODUCCIÓN**

El control de acceso se viene realizando tradicionalmente en grandes edificios y recintos de gran ocupación. Hoy en día estos sistemas representan un plus importante de seguridad para cualquier tipo de empresa o actividad. El control de acceso básicamente permitirá diferenciar:

* Qué personas tienen acceso a las dependencias de trabajo.
* A qué dependencias concretas tienen acceso.
* En que horario tienen acceso.

Una gran ventaja que aportan estos sistemas es tener un registro con todos los accesos que se producen a las instalaciones. De esta manera se puede controlar cualquier evento interno (robo, sabotaje, etc) ya que se pueden recuperar el registro de fichajes de un día determinado a una hora en cualquier punto de la instalación donde exista dicho control.

En este trabajo de título, es administrar el acceso a algún recinto privado, bajo el monitoreo y administración remoto a través de una plataforma web. Esta última, brinda una mayor eficiencia y movilidad al administrador al permitir que desde cualquier lugar con acceso a red internet, se pueda controlar y confirmar el acceso de cualquier persona. La forma de identificación de los usuarios, es mediante el sistema biométrico de la huella digital y también por RFID (identificación por radiofrecuencia). La plataforma web debe además permitir ingresar, validar, editar y eliminar usuarios, además de facilitar la visualización y exportación del historial de ingresos y estadísticas del usuario.

## OBJETIVOS

### 1.1.1 Objetivo general

Diseñar y construir un sistema electrónico para el control de acceso biométrico de huella dactilar gestionado por una plataforma WEB.

### 1.1.2 Objetivos específicos

* Investigación detallada del funcionamiento del sensor biométrico de huella dactilar y sistemas RFID (identificación por radiofrecuencia).
* Diseñar un módulo de control de acceso que sea capaz de identificar huellas dactilares y tarjetas RFID, lograr diferenciarlas entre sí, almacenarlas y modificarlas en la memoria de un microcontrolador, debe contar con un lector de memoria SD para almacenar el registro de los usuarios y el historial de ingreso de estos, de modo de respaldo.

El módulo debe tener conexión estable a una Red de Área Local (LAN), enviar y recibir información a un servidor de base de datos **Mysql**.

* Desarrollar una plataforma web en donde el administrador del módulo, tenga acceso a los usuarios registrados, con la opción de ingresar e editar datos personales, permitir y denegar acceso, ver historial y estadísticas de ingreso durante un periodo determinado.
* Realizar un estudio económico del sistema de control de acceso.
* Implementar el módulo control de acceso en un lugar determinado y verificar su correcto funcionamiento.

## 1.2 Planteamiento de la investigación

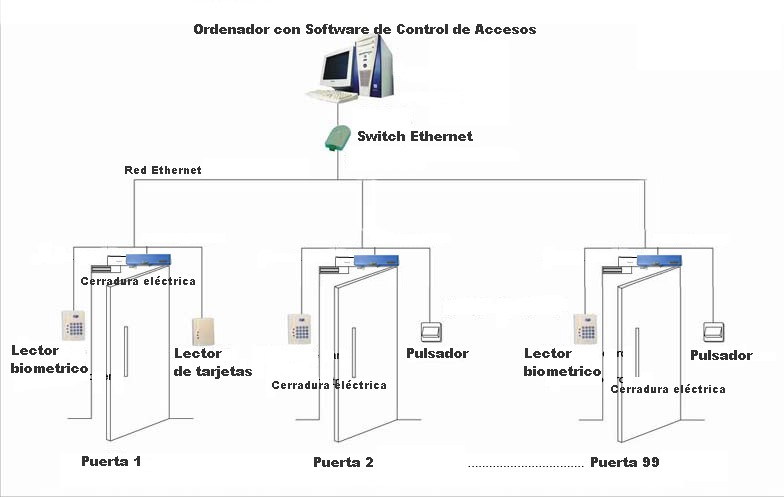
Un sistema de control de acceso hace referencia al mecanismo que en función de la identificación ya autentificada permite acceder a datos o recursos. Básicamente encontramos sistemas de controles de acceso en múltiples formas y para diversas aplicaciones. Por ejemplo, encontramos sistemas de controles de acceso por software cuando digitamos nuestra contraseña para abrir el correo, otro ejemplo es cuando debemos colocar nuestra huella en un lector para desbloquear un teléfono inteligente. Estos casos, son ejemplos que permiten el acceso a datos. Sin embargo, este trabajo está enfocado en la seguridad electrónica, relacionada al acceso de recursos para la apertura de una puerta.

Un sistema de control de acceso es un sistema electrónico que restringe o permite el acceso de un usuario a un área específica, validando la identificación por medio de diferentes tipos de lectura (clave por teclado, tarjetas de proximidad o biometría) y a su vez controlando el recurso (puerta) por medio de un dispositivo eléctrico como un electroimán, solenoide, pestillo o motor.

Básicamente los controles de acceso se clasifican en dos tipos:

* Sistemas de Control de Acceso Autónomos
* Sistemas de Control de Acceso en Red

Los Sistemas de Control de Acceso Autónomos son sistemas que permiten controlar una o más puertas, sin estar conectados a un PC o un sistema central, por lo tanto, no guardan registro de eventos. Aunque esta es la principal limitante, algunos controles de acceso autónomos tampoco pueden limitar el acceso por horarios o por grupos de puertas, esto depende de la robustez de la marca. Es decir, los más sencillos solo usan el método de identificación (ya sea clave, proximidad o biometría) como una "llave" electrónica.

Los Sistemas de Control de Acceso en Red son sistemas que se integran a través de un PC local o remoto, donde se hace uso de un software de control que permite llevar un registro de todas las operaciones realizadas sobre el sistema con fecha, horario, autorización, etc. Van desde aplicaciones sencillas hasta sistemas muy complejos y sofisticados según se requiera. En la siguiente figura se representa un ejemplo de un Sistema de Control de Acceso en Red

**Figura** **1-1. Ejemplo de un Sistema de Control de Acceso en Red.**

En este trabajo se desarrollará un sistema de control de acceso en red, para el control físico de una puerta, se debe implementar un mecanismo electrónico que autentifique al usuario y en base a ello otorgar acceso a alguna dependencia, este sistema involucra el desarrollo de un software y hardware.

El software está asociado a una plataforma web para la gestión de la información, por medio de una interfaz gráfica, el administrador del sistema puede controlar los recursos que están disponibles para los usuarios; crear, editar y eliminar el perfil de los usuarios, configurar permisos, establecer horarios de acceso, exportar e imprimir informes y relatorios, entre otras funciones.

El hardware corresponde a un módulo electrónico de control de acceso, su función principal es realizar la autentificación de usuarios a través de diferentes métodos como; contraseñas, huella dactilar (biometría) y tarjetas RFID.

El módulo electrónico cuenta principalmente con los siguientes componentes:

* **Controlador**, este puede ser un microcontrolador o computador, es el encargado de gerenciar los diferentes sensores y principal comunicador con la plataforma web.
* **Pantalla**, el usuario puede visualizar gráficamente opciones de configuración e instrucciones.
* **Teclado**, necesario para realizar las configuraciones del módulo e ingreso de contraseñas de acceso.
* **Lector RFID**, este puede leer etiquetas RFID a una distancia de varios centímetros, utilizado para el registros de etiquetas al sistema y identificación de las mismas.
* **Sensor biométrico**, Lector de acceso biométrico de huella dactilar. Identifica al usuario a través de su huella dactilar (característica única e irrepetible para cada persona), lo que da un nivel de seguridad superior a cualquier otro sistema que utilice otro tipo de identificadores.
* **Reloj de tiempo real (RTC)**, utilizado para almacenar los horarios de las actividades en los registro.
* **Conector Ethernet o Wifi**, es necesario tener conectividad a una red de área local (LAN[[1]](#footnote-1)) o una red de área extensa (WAN[[2]](#footnote-2)), para realizar la transferencia de datos entre la plataforma y el módulo electrónico.
* **Relé**, dispositivo electromagnético, funciona como un interruptor controlado por un pulso electrónico, es el encargado de activar el circuito eléctrico del control de la puerta.

Conociendo los componentes necesarios para la implementación del módulo de control de acceso, a continuación se plantean diferentes modelos para el hardware.

## 1.2.1 Modelos planteados para el hardware

Se proponen 3 modelos para la arquitectura del hardware del módulo de control de acceso:

1. El controlador del hardware es un microcontrolador[[3]](#footnote-3) este es el encargado de gerenciar cada uno de los periféricos (ver figura 1-2), este ejecuta instrucciones almacenadas como números binarios organizados secuencialmente en la memoria principal, puede ser programado en diferentes lenguajes de programación como; visual basic, C, C++, assembler, Python, etc.

Este módulo electrónico interactúa con los usuarios, realiza la matriculación, verificación e identificación, controla la activación del relé para el actuador instalado en la puerta, funciona como servidor, este gestiona y administra la plataforma web, incluye un reloj de tiempo real y una memoria externa para el almacenamiento y registro de la información.



**Figura** **1-2. Arquitectura módulo control de acceso, modelo 1.**

1. El controlador del hardware es un computador de placa reducida, se propone utilizar una “Raspberry Pi”, su sistema operativo oficial es RaspBian, una versión adaptada de Debian, aunque es compatible con diversos sistemas operativos, la Raspberry Pi cuenta con pines GPIO (General Purpose Input/Output), un sistema de Entrada/Salida de propósito general, las cuales son utilizadas para el gerenciamiento de los periféricos (ver figura 1-3). Cuenta con 4 puertos USB, adaptador para una memoria µSD, conector Ethernet y Wifi. Es compatibles con diferentes lenguajes de programación como; C++, Visual Basic, Phyton, Java, Javascript, HTML, PHP, Ruby, Node.js, etc.



**Figura 1-3. Arquitectura módulo control de acceso, modelo 2.**

Este módulo tiene las mismas funciones que el anterior, con la diferencia que este puede ejecutar diferentes procesos y tareas de forma simultáneas, e instalar servidores dedicados para el manejo de la plataforma web, es compatible con diferentes bases de datos (Mysql, MongoDB, Firebase) para el registro de la información.

1. Este modelo está compuesto por 2 hardware, el primero es el módulo de control de acceso, cuyo controlador principal es un microcontrolador, este comanda los sensores que interactúan con los usuarios, encargado de realizar los procesos de matriculación, verificación e identificación de los usuarios. Este hardware es gerenciado por un controlador maestro (Raspberry Pi) a través de un puerto serial (ver figura 1-4).



**Figura 1-4. Arquitectura módulo control de acceso, modelo 3.**

El segundo hardware corresponde a la computadora (Raspberry Pi), cumple la función de servidor y administra toda la información de la plataforma web, almacena los registros de los usuarios, permite el acceso a estos, contiene los historiales de acceso, etc. Este hardware integra un circuito con un reloj de tiempo real, para llevar el control del tiempo del servidor y un relé para el control del actuador de la puerta.

Este último funciona como un controlador maestro, está siempre recibiendo información del módulo de control de acceso, y cuando recibe una petición de un usuario este gestiona la información y envía una respuesta de vuelta.

## 1.2.2 Justificación del modelo seleccionado

Los modelos planteados del hardware para el módulo de control de acceso utilizan como controlador principal un microcontrolador o un computador de placa reducida (Raspberry Pi), en la tabla 1-1 se pueden observar las principales características de cada controlador.

**Tabla 1-1. Características microcontrolador v/s Raspberry Pi.**

|  |  |  |
| --- | --- | --- |
|  | Microcontrolador (ATmega2560) | Raspberry Pi |
| Memoria RAM | 0.008 [MB] | 1 [GB] |
| Pines de Entrada/Salida | 54 | 40 |
| Velocidad de reloj | 16 [MHz] | 1.2 [GHz] |
| Multitarea | No | Si |
| Network | Ninguno | Ethernet y Wifi |
| Voltaje de entrada | 7 a 12 [V] | 5 [V] |
| Memoria Flash | 256 [KB] | Tarjeta SD, 2 a 16 [GB] |
| Puertos USB | 1 | 4 |
| Sistema operativo | Ninguno | Distribuciones de Linux |

La Raspberry Pi es 75 veces más rápido que el microcontrolador y tiene 125.000 veces más de memoria RAM. La Raspberry Pi es una computadora independiente que puede ejecutar un sistema operativo real en Linux. Puede realizar varias tareas, soportar cuatro puertos USB y puede conectarse de forma inalámbrica a Internet. En pocas palabras, es lo suficientemente potente como para funcionar como una computadora personal.

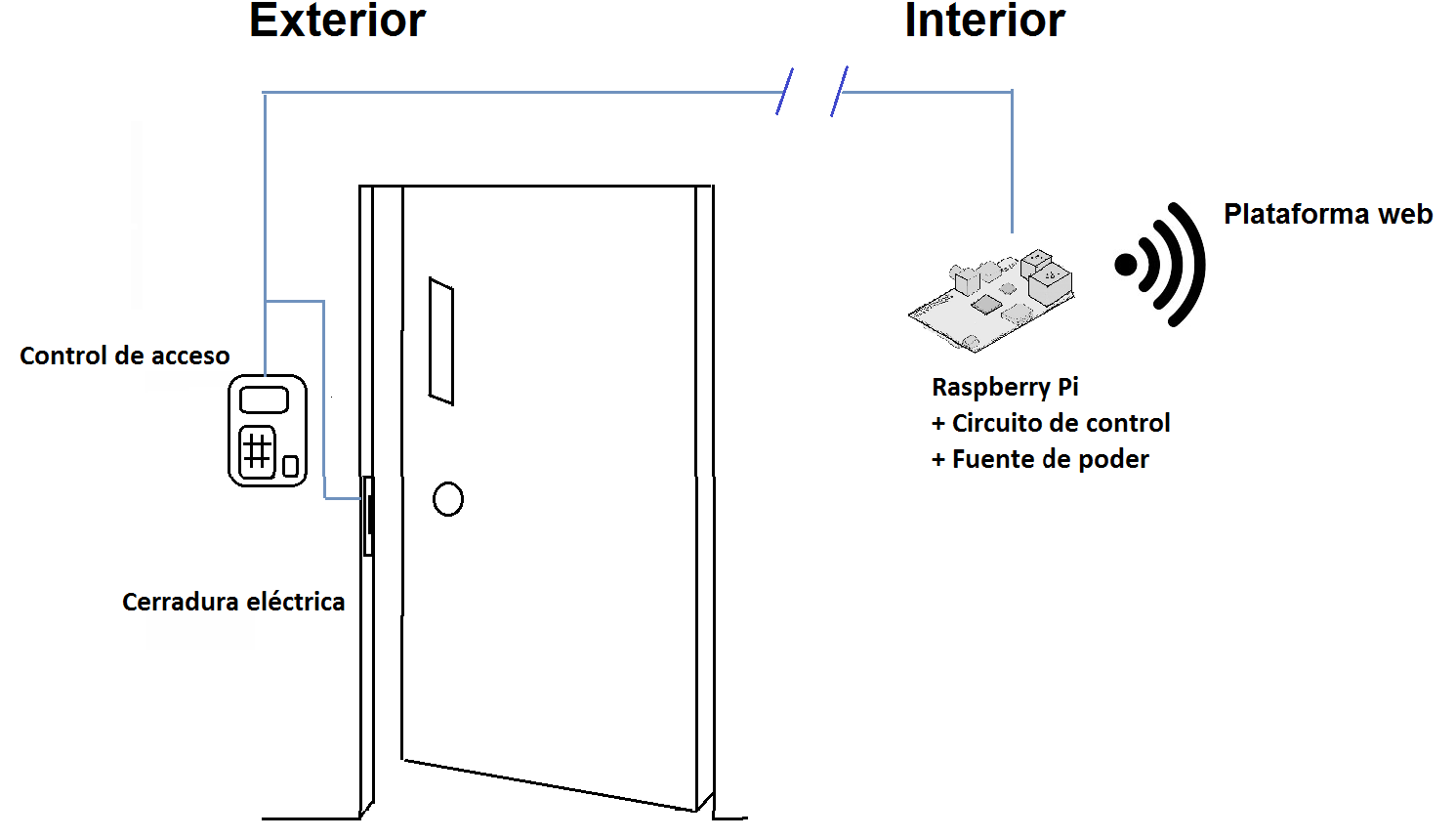
Puede sonar que Raspberry Pi es superior al microcontrolador, pero eso es sólo cuando se trata de aplicaciones de software. Por otra parte el microcontrolador tiene la capacidad de trabajar con pines analógicos y en tiempo real, esta flexibilidad le permite trabajar con casi cualquier tipo de sensor o chip.

Raspberry Pi puede procesar varias tareas, éste puede ejecutar múltiples programas en segundo plano mientras está activado. Por ejemplo, puede estar funcionando como un servidor y transmitiendo información por un puerto USB.

El modelo seleccionado es el número 3, en este modelo se aprovecha las ventajas de usar el microcontrolador y la Raspberry Pi, el módulo de control de acceso es controlado por el microcontrolador, es el encargado de gestionar la parte sensorial, el usuario interactúa con este módulo por medio de una pantalla LCD y un teclado matricial, en este dispositivo se realizan los procesos de matriculación, verificación e identificación de los usuarios. Cada vez que el usuario interactúa con este módulo se envía una petición a la Raspberry Pi, este procesa la información, y envía una respuesta.

La Raspberry Pi es el controlador principal del *sistema de control de acceso*, este aloja la plataforma web y almacena la información del sistema, tiene la capacidad de ejecutar procesos diferentes para la gestión del módulo de control de acceso, por lo que este sistema funcionara en tiempo real, además este integra los adaptadores de red ethernet y wifi de fácil configuración.

El módulo de control de acceso es instalado al exterior del lugar a controlar, y la Raspberry Pi se encuentra en el interior (ver figura 1-5), dando más seguridad e independencia al sistema, los usuarios que interfieran con el módulo exterior no podrán tener acceso a la información del sistema, ni manipular el actuador para la apertura de la puerta.



**Figura 1-5. Arquitectura módulo control de acceso, modelo 3.**

## METODOLOGÍA

Para realizar el diseño y construcción de este proyecto, es necesario dividir el trabajo por etapas, el capítulo 1 corresponde a la introducción, en donde se definen los objetivos del proyecto, descripción del planteamiento de la investigación y el análisis de diferentes soluciones para el diseño del hardware del proyecto.

En el capítulo 2 se describe las técnicas de identifación y autenticación de usuarios que están involucradas en este proyecto, aquí se detalla el funcionamiento de est

Para realizar la creación y el diseño de la plataforma computacional es necesario dividir el trabajo por etapas; teniendo como primera instancia una etapa inicial de aprendizaje y dominio de las variables involucradas dentro del diseño, luego continua la etapa 2, la cual consiste en la creación y diseño de la plataforma computacional en base a todo lo aprendido en la etapa inicial de aprendizaje. Posteriormente con la plataforma web ya creada se da comienzo a la etapa 3, la que tiene como objetivo la implementación del HOST y el dominio web para la visualización en internet de la plataforma creada.

Finalmente con la plataforma computacional creada y con la circulación en internet ya establecida, se procede a realizar la etapa 4 y final del proyecto, la cual consiste en la comprobación total del funcionamiento de todas las variables presentes en la plataforma computacional, fijándose que no posea errores en sus funciones, además de darle estilo, es decir, detalles como colores, fuentes específicas, tamaños, etc. En resumen un completo perfeccionamiento para que la plataforma sea lo más práctica y cercana al usuario final.

A continuación se procede a realizar un análisis más detallado de las etapas nombradas anteriormente.

# **CAPITÚLO 2**: TÉCNICAS DE IDENTIFICACIÓN Y AUTENTIFICACIÓN DE USUARIOS.

La identificación digital forma parte indisoluble de la mayoría de servicios en Internet y en las tecnologías de la información. Por ejemplo, para subir un vídeo en un servidor web, se pide que el usuario esté registrado y se identifique para poder llevar a cabo la publicación del contenido. Por otra parte, para utilizar una tarjeta magnética de acceso, es preciso que el usuario esté registrado en la plataforma de control. Para realizar acciones tan variadas como hacer un pago mediante tarjeta de crédito, utilizamos esta misma tarjeta para identificarnos. O bien para hacer gestiones bancarias a través de Internet, lo primero que haremos es especificar quién somos.

Esta identificación digital puede ser relativamente sencilla. Basta con disponer de un nombre de usuario, usar como identificador la dirección de correo electrónico o, en el caso de un pago, usar el número de tarjeta de crédito. Ahora bien, para la mayoría de servicios, además de la identificación digital, es necesaria una autenticación de esta identidad.

Mediante la autenticación de la identidad, el servicio se asegura de que el usuario es quien dice ser.

El concepto de "quien dice ser" lo resuelve el identificador de usuario: mediante una cadena de caracteres se denota cuál es la identidad del usuario. Y para demostrar la autenticidad de la identidad del usuario se pueden usar cuatro aproximaciones distintas:

**1)** El usuario es quien dice ser si demuestra conocer algo que solamente este conoce. Por ejemplo, conoce una palabra secreta de acceso.

**2)** El usuario es quien dice ser si posee algún objeto, como por ejemplo una tarjeta magnética para el acceso a un edificio.

**3)** El usuario es quien dice ser si posee alguna característica física que sólo él tiene: por ejemplo, la huella dactilar.

**4)** El usuario es quien dice ser si es capaz de hacer algo de forma única: por ejemplo, el patrón de escritura o la forma de andar.

Pese a tratarse de cuatro formas de abordar la autenticación de la identidad, no existe una frontera clara entre algunas de ellas. Además, es perfectamente posible el uso de varias de estas técnicas de forma combinada, para conseguir mayores grados de seguridad.

Por ejemplo, disponer de una tarjeta con códigos de seguridad para permitir operaciones bancarias a través de Internet (tarjeta de coordenadas) podría verse como una mezcla entre los dos primeros casos: el usuario posee la tarjeta, pero podría decirse que es conocedor de una información, aunque en este caso la tenga escrita.

Por otra parte, todas estas formas de abordar la autenticación no están exentas de problemas de seguridad. Por ejemplo, la tarjeta con códigos podría ser robada y usada por otro usuario. Otro caso podría ser la obtención de una clave de acceso mediante un correo electrónico fraudulento.

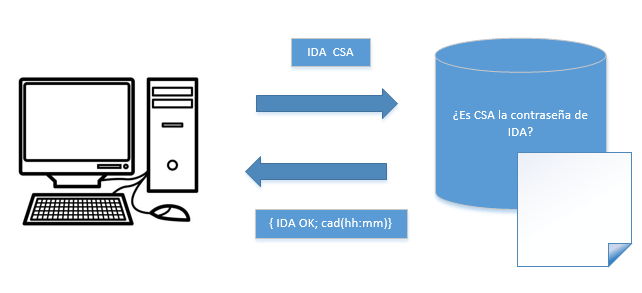
La usurpación de identidad consiste en que una entidad use con éxito el mecanismo de identificación que identifica a otra identidad.

En este apartado se exponen las principales técnicas y tecnologías para implantar la identificación y autenticación de usuarios en un sistema de control de acceso de personas.

**2.1. Contraseñas**

La autenticación por medio de contraseñas es relativamente sencilla: el usuario “*A”* envía su identificador “*IDA”* y acto seguido su contraseña “*CSA”*. La implementación del protocolo de identificación puede precisar que ambas informaciones se manden en un mismo mensaje, o bien que primero se pida el nombre de usuario y después la contraseña.

Esta contraseña es usada por el servicio para validar la identidad del usuario. Si *A* es el único conocedor de *CSA*, es altamente probable que el usuario sea realmente *A*. En la figura 2-1, el usuario utiliza su contraseña para validarse al servicio. Este dispone de una lista de pares con usuarios y contraseñas para comprobar las identidades. Ante una contraseña correcta, el servicio manda un mensaje de autenticación correcta al equipo del usuario. Algunos servicios incluye una hora de caducidad de esta autenticación, tras la cual el usuario deberá volver a autenticarse.



**Figura 2-1. Esquema de funcionamiento de autentificación mediante contraseña.**

La contraseña es una cadena de caracteres de longitud arbitraria. En algunos escenarios, se usa una versión reducida de la contraseña, formada tan sólo por unos pocos números. En este caso la contraseña se conoce con el nombre de PIN, en algunos entornos se utiliza el término PIN cuando en realidad la contraseña está formada tanto por letras como por números.

Así como un mismo servicio o sistema no permite que haya dos usuarios con el mismo identificador, es perfectamente posible que dos o más usuarios tengan la misma contraseña.

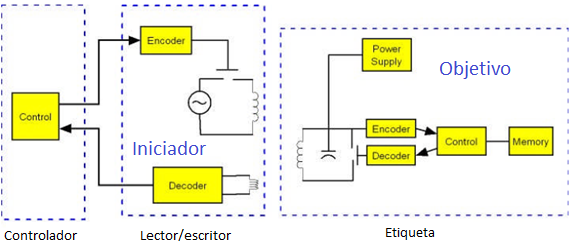
**2.2 Identificación por Radio Frecuencia (RFID)**

La identificación por radio frecuencia (RFID) se está convirtiendo en algo común en la vida cotidiana en estos días la podemos encontrar en tarjetas de pago, también es utilizado en el sistema de cobro de las autopistas o carreteras, por otra parte es utilizado por empresa para llevar el control del inventario de productos y equipos, la mayoría de nosotros encontramos etiquetas RFID al menos un par de veces a la semana y nunca pensamos en lo que se puede hacer con esta tecnología.

**2.2.1 Funcionamiento de la Identificación por Radio Frecuencia (RFID)**

El intercambio de un sistema RFID implica dos actores: un iniciador y un objetivo. El iniciador, es un lector o lector/escritor de dispositivo, este inicia el intercambio mediante la generación de un campo magnético y escucha las respuestas de cualquier objetivo que se encuentre dentro del radio. El objetivo, es una etiqueta, que responde cuando se recoge una transmisión desde un iniciador y responderá con un número identificador único (UID).

En la figura anterior se muestra, la estructura esquemática del proceso de lectura de una etiqueta RFID. En la misma se puede ver que en dicho lector contiene una unidad de alimentación, un codificador, un decodificador, una unidad de control y la unidad de memoria, la cual posee el número de identificación de esa etiqueta. Al ingresar la etiqueta al campo de la lectora, se energiza el sistema y a través de la unidad de control se envía la información contenida en la memoria de la tarjeta, la cual posteriormente es recibida, por el decodificador de lectora.



**Figura 2-2. Esquema y componentes de un sistema RFID**

Existen 3 componentes básicos en un sistema RFID.

1. La etiqueta o transponder de RFID consiste en un pequeño circuito, integrado con una pequeña antena, capaz de transmitir un número de serie único hacia un dispositivo de lectura, como respuesta a una petición. Algunas veces puede incluir una batería.

2. El lector, (el cual puede ser de lectura o lectura/escritura) está compuesto por una antena, un módulo electrónico de radiofrecuencia y un módulo electrónico de control, sirve para energizar la tarjeta de identificación y capturar su número, para posteriormente enviarlo a un sistema de control o un computador que se encargue de procesar los datos.

3. Un controlador o un equipo anfitrión, comúnmente una PC o Workstation, Este se encarga de recibir y procesar los datos pertenecientes al número de identificación de la etiqueta presentada en la lectora, la cual corre una base de datos y algún software de control

Las etiquetas RFID tienen una pequeña cantidad de memoria a bordo, por lo general menos de 1 kilobyte. Un dispositivo iniciador puede leer estos datos, y si se trata de un dispositivo de lectura/escritura, se puede escribir en el etiquetar también. Esto le permite almacenar pequeñas cantidades de información asociados con la tarjeta. Por ejemplo, a veces se utiliza en sistemas de transporte que utilizan RFID, para realizar un seguimiento del valor que queda en la tarjeta. Sin embargo, desde los sistemas de RFID en general, están conectados en red a una base de datos, es más común para almacenar un registro de datos indexada por el UID de la etiqueta en una base de datos remota, y almacenar toda la información acerca de la etiqueta en esa base de datos remota.

**2.2.3 Clasificación**

La tecnología de identificación por radiofrecuencia puede ser dividida principalmente en 3 categorías según el tipo de etiqueta:

1. Sistemas pasivos, en los cuales las etiquetas de RFID no cuentan con una fuente de poder. Su antena recibe la señal de radiofrecuencia enviada por el lector y almacena esta energía en un capacitor. La etiqueta utiliza esta energía para habilitar su circuito lógico y para regresar una señal al lector. Estas etiquetas pueden llegar a ser muy económicas y pequeñas, pero su rango de lectura es muy limitado.

2. Sistemas activos. Utilizan etiquetas con fuentes de poder integradas, como baterías. Este tipo de etiquetas integra una electrónica más sofisticada, lo que incrementa su capacidad de almacenamiento de datos, interfaces con sensores, funciones especializadas, además de que permiten que exista una mayor distancia entre lector y etiqueta (20m a 100m). Este tipo de etiquetas [7] son más costosas y tienen un mayor tamaño. Pueden permanecer dormidas hasta que se encuentran dentro del rango de algún lector, o pueden estar haciendo broadcast constantemente.

3. Sistemas Semi-Activos. Emplean etiquetas que tienen una fuente de poder integrada, la cual energiza la etiqueta para su operación [17], sin embargo, para transmitir datos, una etiqueta semi-activa utiliza la potencia emitida por el lector. En este tipo de sistemas, el lector siempre inicia la comunicación. La ventaja de estas etiquetas es que al no necesitar la señal del lector para energizarse (a diferencia de las etiquetas pasivas), pueden ser leídas a mayores distancias, y como no necesita tiempo para energizarse, estas etiquetas pueden estar en el rango de lectura del lector por un tiempo substancialmente menor para una apropiada lectura. Esto permite obtener lecturas positivas de objetos moviéndose a altas velocidades.

Tanto los tags activos como los pasivos pueden adicionalmente ser clasificados de la siguiente forma:

* Solo Lectura (RO)

En estos dispositivos, los datos son grabados en la etiqueta durante su fabricación, para esto, los fusibles en el microchip de la etiqueta son quemados permanentemente utilizando un haz láser muy fino. Después de esto, los datos no podrán ser reescritos. Este tipo de tecnología se utiliza en pequeñas aplicaciones, pero resulta poco práctico para la mayoría de aplicaciones más grandes, que intentan explotar todas las bondades de RFID.

* Una Escritura, Muchas Lecturas (WORM)

Una etiqueta WORM, puede ser programado sólo una vez, pero esta escritura generalmente no es realizada por el fabricante sino por el usuario justo en el momento que la etiqueta es creada. Este tipo de etiquetas puede utilizarse en conjunto con las impresoras de RFID, las cuales escriben la información requerida en él.

* Lectura y Escritura (RW)

Estas etiquetas, pueden ser reprogramadas muchas veces, típicamente este número varía entre 10,000 y 100,000 veces, incluso mayores. Esta opción de reescritura ofrece muchas ventajas, ya que la etiqueta puede ser escrito por el lector, e inclusive por sí mismo en el caso de los tags activos. Estas etiquetas regularmente contienen una memoria Flash o FRAM para almacenar los datos.

**2.2.4 Regulación y estandarización**

Las etiquetas RFID de baja frecuencia (LF: 125 - 134 kHz y 140 - 148.5 kHz) y de alta frecuencia (HF: 13.56 MHz) se pueden utilizar de forma global sin necesidad de licencia. La frecuencia ultra alta (UHF: 868 - 928 MHz) no puede ser utilizada de forma global, ya que no hay un único estándar global. En Norteamérica, la frecuencia ultra elevada se puede utilizar sin licencia para frecuencias entre 908 - 928 MHz, pero hay restricciones en la energía de transmisión. En Europa la frecuencia ultra elevada está bajo consideración para 865.6 - 867.6 MHz. Su uso es sin licencia sólo para el rango de 869.40 - 869.65 MHz, pero existen restricciones en la energía de transmisión. El estándar UHF norteamericano (908-928 MHz) no es aceptado en Francia ya que interfiere con sus bandas militares. En China y Japón no hay regulación para el uso de la frecuencia ultra elevada. Cada aplicación de frecuencia ultra elevada en estos países necesita de una licencia, que debe ser solicitada a las autoridades locales, y puede ser revocada. En Australia y Nueva Zelanda, el rango es de 918 - 926 MHz para uso sin licencia, pero hay restricciones en la energía de transmisión.

Contrariamente a la creencia popular, no existe un protocolo RFID universal para esta tecnología. Hay docenas. Los estándares de RFID desarrollado por la Organización Internacional de Estándares (ISO), en conjunción con los principales participantes en el mercado. ISO funciona como un órgano de mediación para ayudar a los competidores en muchas industrias diferentes a desarrollar estándares interoperables de modo que incluso cuando compiten, sus tecnologías pueden a veces trabajar juntas. Los diferentes estándares de RFID definen las frecuencias de radio utilizado, las velocidades de transferencia de datos, los formatos de datos, entre otros parámetros. Por ejemplo, la norma ISO-11784 fue originalmente desarrollada para el rastreo de animales. Se opera en las frecuencias comprendidas entre 129 y 139.4 kHz, y su formato de datos incluye campos adecuados para la descripción de los animales para ser rastreado. También se puede encontrar lectores con el protocolo EM4100 y etiquetas que operan en el 125kHz distancia. Estos se utilizan a menudo como tarjetas de proximidad, y cuentan con una información muy limitada en su protocolo de datos, por lo general sólo un UID. Las normas ISO-14443 se desarrollaron para utilizar con los sistemas de pago y tarjetas inteligentes. Estas operan a 13,56 MHz. Incluyen características en su formato de datos para aumentar o disminuir los valores y para la codificación de datos.

**2.3. Biometría**

La biometría es la aplicación de las matemáticas y la ciencia de la computación para identificar individuos de acuerdo con sus características o rasgos físicos. La palabra biometría deriva del griego bios (que quiere decir vida) y metria (que quiere decir medida). Los rasgos biométricos son medidas extraídas del cuerpo humano vivo. Y, además, todos los rasgos biométricos son una combinación de anatomía y de comportamiento.

La biometría se está convirtiendo en un factor esencial para la identificación eficaz de las personas. Esto se debe a que los rasgos biométricos no se pueden compartir o extraviar y representan intrínsecamente las formas corporales del individuo que identifica. Reconocer a una persona por su cuerpo y después enlazar este cuerpo con una identidad establecida externamente forma una herramienta muy poderosa para la gestión de la identidad con unas consecuencias potenciales enormes, tanto positivas como negativas. En consecuencia, la biometría no es solo un problema fascinante en el campo de la investigación dedicado al reconocimiento de patrones, sino una tecnología que, usada correctamente, puede permitir una sociedad más segura, reducir el fraude y proveer interfaces persona-máquina fáciles de usar.

El reconocimiento biométrico se refiere al uso de diferentes características anatómicas (como huellas dactilares, cara o iris) y de comportamiento (como habla, firma o teclear). Estas características se denominan identificadores biométricos o rasgos biométricos y sirven para reconocer automáticamente a los individuos.

A continuación enumeramos los rasgos biométricos más utilizados, comentando aspectos sobre su calidad y uso:

• **Huellas dactilares**. El uso de las huellas dactilares como medio de identificación es de alta fiabilidad. Tiene buena aceptación y popularidad.

• **Ojo**. La identificación de un individuo a través del análisis del iris tiene una fiabilidad muy alta. El problema es la facilidad de uso. El análisis de la retina para identificar al individuo es más complejo todavía.

• **Forma de la mano**. Este sistema está bastante extendido, pero quizás presenta un poco menos de fiabilidad que los anteriores. Este sistema es susceptible de padecer ataques, puesto que no resultaría difícil recrear la forma de una mano usando un molde.

• **Cara**. El análisis de la cara, ya sea en 2D o 3D, es un buen medio para identificar a un individuo. Aun así, siempre es preferible un estudio 3D, ya que se mejoran los resultados con respecto el mero estudio de una imagen 2D, y se dificulta el éxito de ataques. La desventaja de los equipos 2D es que el sistema no distingue si lo que está capturando es realmente un rostro o una fotografía de un rostro.

• **Venas del dedo o la mano**. El estudio vascular de los dedos o de la mano proporciona alta fiabilidad, además no es muy complejo su análisis.

• **Voz**. Este sistema presenta bastante fiabilidad, pero también es susceptible a los ataques. Además este sistema padece de poca estabilidad, con lo cual deberían ser varias la toma de voces analizadas para reducir la tasa de errores.

Claramente, son varias las características biométricas que pueden usarse para identificar o autenticar. Estas pueden usarse individualmente o bien en combinación entre ellas, o bien procediendo a la autenticación mediante información adicional, como podría ser una contraseña.

**2.3.1. Sistemas biométricos**

En función del contexto de la aplicación biométrica, podemos diferenciar dos tipos de sistemas, los sistemas de verificación y los sistemas de identificación.

Los sistemas de verificación (también llamados de autentificación) autentifican la identificación de la persona mediante la comparación del rasgo biométrico acabado de capturar con el rasgo biométrico que el sistema ha capturado antes en el proceso de inscripción al sistema.

El usuario tiene que presentar su identificación mediante un carné o clave secreta. El sistema realiza una única comparación entre el rasgo biométrico que el usuario acaba de presentar con el rasgo biométrico que hay en la base de datos con la misma identificación presentada. La salida de un sistema de verificación es normalmente binaria: es la misma persona si los rasgos biométricos coinciden (son muy similares) o son dos personas diferentes en otro caso. En algunos casos, los rasgos biométricos y la identificación de la base de datos están en la propia tarjeta del usuario en forma encriptada. En este caso, se dice que disponemos de una base de datos distribuida entre todas las tarjetas de los usuarios.

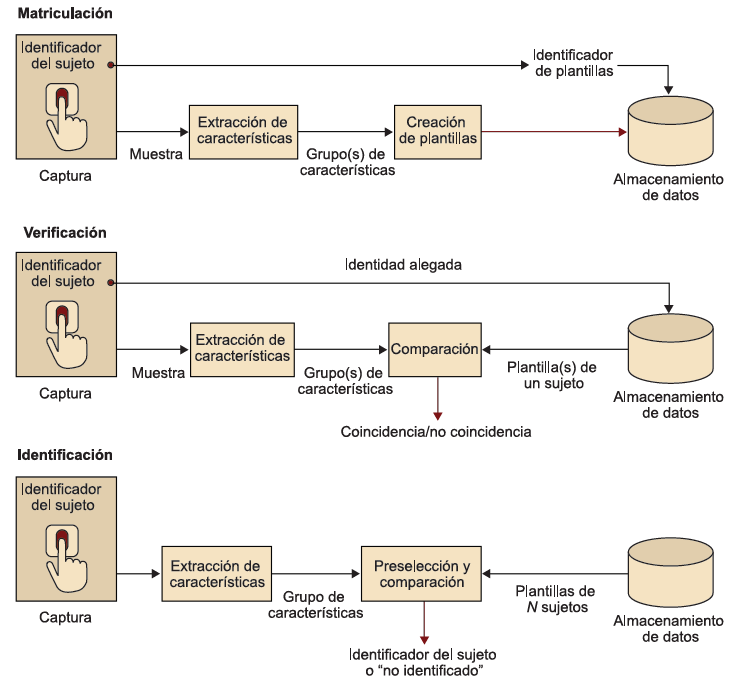
Los sistemas de identificación reconocen a la persona a través de la busca del rasgo biométrico que más se asemeja al usado para identificarlo en toda una base de datos.

El usuario no aporta ninguna información de su identificación, como era el caso del sistema de verificación. El sistema lleva a cabo una comparación uno a muchos. Esto quiere decir que el rasgo biométrico del usuario desconocido se compara con muchos rasgos biométricos de una base de datos. Hay varias salidas a este sistema. La más sencilla es devolver el nombre de la persona (identificador) cuyo rasgo biométrico se asemeja al introducido. Otra posibilidad es deducir que este rasgo biométrico no pertenece a ninguna persona de la base de datos (este caso se da cuando las distancias del rasgo biométrico con todos los rasgos biométricos de la base de datos es superior a un umbral). Finalmente, y es el caso más usual en las aplicaciones forenses, el sistema no devuelve una sola persona sino una lista de candidatos. Es decir, devuelve las personas cuya distancia de sus rasgos biométricos con el rasgo biométrico introducido es inferior a un umbral.

Tanto los sistemas de verificación como de identificación necesitan un proceso previo llamado sistema de matriculación[[4]](#footnote-4). Este proceso se encarga de recoger el rasgo biométrico (o los rasgos biométricos) junto con la identificación de la persona. Este proceso es muy importante puesto que se encarga de relacionar la identificación de la persona con el rasgo biométrico. Normalmente, este proceso tiene lugar ante una persona autorizada que vela por la veracidad de los datos que aporta el usuario (carné de identidad, pasaporte) y controla que realmente sea este usuario el que presenta el rasgo biométrico al sistema. Además, durante el proceso de matriculado, esta persona verifica la calidad de los datos biométricos obtenidos. Si considera que los datos no tienen suficiente calidad, pide al usuario que vuelva a presentar el rasgo biométrico (huella dactilar, cara, iris) al sistema.

Es fundamental que los datos que se almacenan en la base de datos tengan la máxima calidad puesto que en los procesos de identificación o verificación no siempre se puede garantizar esta calidad. Algunos sistemas piden al usuario capturar varias veces el mismo rasgo biométrico (normalmente tres veces). El sistema puede elegir la mejor imagen o fusionarlas y así reducir los errores de captura.

Los tres sistemas antes mencionados usan los procesos siguientes (ver la figura 2-3):



**Figura 2-3. Etapas o procesos que componen los sistemas: a. matriculación; b. verificación; c. identificación.**

• Captura: La representación digital del rasgo biométrico tiene que ser capturada. El sensor biométrico es usualmente un sistema para capturar una imagen (excepto la identificación del hablador, que es una máquina de grabar voz). Normalmente, la información capturada se denomina muestra[[5]](#footnote-5). A veces, el sistema de captura también incorpora otros periféricos para introducir información no biométrica o mostrar información.

• Extracción de las características: Con el objetivo de facilitar la comparación, aumentar la información y reducir el ruido, la representación original digital (imagen digital) se procesa normalmente con un extractor de características para generar una representación compacta y más identificadora llamada registro de identificación[[6]](#footnote-6) o conjunto de características[[7]](#footnote-7).

• Creación de la plantilla: La plantilla[[8]](#footnote-8) es una forma compacta de representar un conjunto de muestras de una sola característica biométrica (por ejemplo, se puede crear una plantilla de dieciséis muestras diferentes de la imagen de la cara de una misma persona). El proceso de creación de la plantilla recibe como entrada los registros de identificación y crea una información más compacta donde se intenta extraer la información que persiste en todas las muestras puesto que se consideran los rasgos característicos.

En algunos casos, esta plantilla está formada por una única muestra y por lo tanto se puede representar como un registro de identificación.

• Comparación: El proceso de comparación recibe como entrada un registro de identificación y una plantilla y calcula una distancia entre los dos. A veces, en lugar de una distancia obtiene una probabilidad de que representen un mismo individuo. En el proceso de verificación, hay un umbral interno al sistema que solo lo puede modificar el administrador del sistema. Si la distancia es inferior al umbral (o la probabilidad es superior al umbral), el sistema considera que los dos datos provienen de la misma persona, de lo contrario, se considera que provienen de dos personas diferentes.

• Selección o filtrado: En los sistemas de identificación con muchos datos (podemos hablar de 50 millones de huellas dactilares), el filtrado es un método para aumentar el tiempo de respuesta del sistema. Con técnicas típicas de las bases de datos, logran no tener que explorar la base de datos entera y así ganar tiempo.

• Almacenamiento de los datos: Es el proceso para almacenar la información del usuario. Esta información está compuesta por un identificador único (por ejemplo, el número del DNI o del pasaporte), la plantilla biométrica y otros datos (por ejemplo, la dirección o la profesión). En función de la aplicación, los datos se almacenan en sistemas de almacenamiento centralizados (para poder llevar a cabo la identificación) o en tarjetas inteligentes[[9]](#footnote-9) (para poder llevar a cabo la verificación). Además, se aplican técnicas de encriptación con todos los datos para que así el registro formado por el número del DNI más los rasgos biométricos sea indivisible.

Dependiendo del dominio de la aplicación, un sistema biométrico puede operar como un sistema en línea o como un sistema fuera de línea.

a) Los sistemas en línea requieren que la comparación se lleve a cabo rápidamente y se requiere una respuesta inmediata, por ejemplo es el caso del permiso para iniciar una aplicación o la entrada física de una persona a unas instalaciones. Normalmente, son sistemas de verificación.

b) Los sistemas fuera de línea no requieren que la respuesta sea inmediata y se tolera que haya un admisible retraso en la respuesta. Normalmente, son sistemas de identificación. Los sistemas en línea suelen ser completamente automáticos, requieren que el rasgo biométrico sea capturado con un sensor electrónico y no hay control humano de la calidad de los datos. Por otro lado, los sistemas fuera de línea son usualmente semiautomáticos. La captura del rasgo biométrico puede haber sido con un sistema no electrónico (por ejemplo, la captura de una huella dactilar dejada en la escena de un crimen) y hay un control de la calidad de los datos por parte de un especialista. Además, este especialista dispone de herramientas informáticas para arreglar los datos o ayudar al programa a que lleva a cabo la comparación biométrica.

Dependiendo de la aplicación, se llevan a cabo dos tipos de buscas en los sistemas de identificación, que son las buscas positivas y las buscas negativas:

a) Las buscas positivas son aquellas por las que queremos comprobar si aquel rasgo biométrico se encuentra en la base de datos, es decir, si el usuario ha sido matriculado. Deseamos saber la identificación de aquel rasgo biométrico. El caso más típico es introducir en el sistema una huella dactilar que hemos encontrado en una escena de un crimen o una cara que hemos podido fotografiar y queremos saber a quién pertenece.

b) Por otro lado, las buscas negativas son aquellas por las que queremos comprobar que aquel individuo no se ha matriculado. Deseamos saber que no hay ninguna persona matriculada con aquellos rasgos biométricos. La aplicación más usual es asegurarse de que una persona no quiere usar más de una vez un servicio que solo tiene derecho a usar una sola vez, como, por ejemplo, no cobrar varias veces una ayuda estatal o no votar varias veces en unas elecciones.

**2.3.2. Lector huella digital**

Los seres humanos tienen tarjetas de identificaron integradas, muy fácilmente accesibles, las cuales son diseños virtualmente únicos. Estas son diminutos “Valles y crestas” de piel en la punta de los dedos, se forman por una combinación de factores genéticos y ambientales aleatorios, por lo que es única para cada persona.

Un lector de huellas dactilares es un dispositivo de [hardware](http://www.alegsa.com.ar/Dic/hardware.php) que permite escanear o leer una huella digital de un ser humano con el fin de identificarlo.

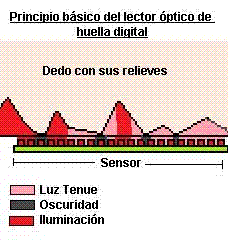
El funcionamiento de un lector de huella digital cumple dos tareas principales:

* Obtener una imagen de la huella digital.
* Comprar el patrón de valle y cresta de dicha imagen con los patrones de las huellas que tiene almacenadas.

El uso más típico de un lector de huellas dactilares es la [autentificación](http://www.alegsa.com.ar/Dic/autentificacion.php) de un [usuario](http://www.alegsa.com.ar/Dic/usuario.php) para permitirle o no el acceso a un dispositivo o servicio. También se incluyen en cerraduras para abrir puertas o para control de entrada-salida de personas de una empresa.

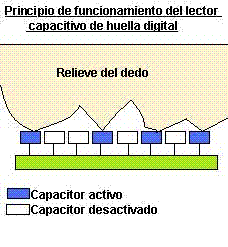
 Hay 2 tipos de tecnología empleadas para estos dispositivos:

Lector óptico: esta tecnología utiliza diminutos [diodos](http://www.informaticamoderna.com/Electricidad_y_computadoras.htm#elem) que emiten luz, integrados en un sensor denominado CCD (Charged Coupled Device), el cuál detecta los relieves por medio de sombras e iluminación de la superficie del dedo, posteriormente forma un mapa digital con la información y determina que tenga la iluminación correcta y una vez realizada esta verificación, esta puede ser enviada a la computadora para su almacenamiento, de lo contrario seguirá creando la imagen digital. En la figura 2-4 se observa el método de lectura de un sensor óptico.



**Figura 2-4. Método de lectura de huella digital óptico**

Lector capacitivo: esta tecnología utiliza diminutos [capacitores](http://www.informaticamoderna.com/Electricidad_y_computadoras.htm), los cuáles se cargan con diferentes cantidades de corriente de acuerdo a la posición de los relieves del dedo, de este modo crea un mapa digital con la información y no es necesario verificarla, sino que se envía de manera inmediata a la computadora, siendo una tecnología más fiable y veloz. En la figura 2-5 se observa el principio de funcionamiento del lector capacitivo de huella digital.



**Figura 2-5. Método de lectura de huella digital capacitivo**

En general existen distintos problemas a los que hacer frente. El primero es el de la recogida de características: los sistemas deben obviar ruidos, imperfecciones o distintos matices que los sistemas captadores y sensores recogen a cada acceso de un individuo. En segundo lugar, se deben minimizar las tasas de errores del sistema de identificación: no se debe permitir la autenticación de un individuo no registrado o no válido y, al contrario, no se debe denegar la correcta autenticación de un individuo registrado o válido.

## DEFINICIONES IMPORTANTES

### **2.1.1 LENGUAJE DE PROGRAMACIÓN:**

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que son llevados a cabos por computadores.

Pueden usarse para crear programas que controlen el comportamiento físico y lógico de un dispositivo, para expresar algoritmos con precisión, o como modo de comunicación dentro una tarea específica.

Está formado por un conjunto de símbolos que definen su estructura y el significado de sus expresiones.

Para entender mejor el significado de lenguaje de programación se debe tener claro que básicamente es el proceso de creación de un programa o aplicación de procedimientos lógicos y variables

En forma general, los lenguajes de programación tienen directa relación con los lenguajes informáticos y se interrelacionan en el gran círculo de la programación.

### 2.1.2 LENGUAJE INFORMÁTICO:

Este lenguaje es muy similar al lenguaje de programación y permite especificar de manera precisa sobre qué datos y características debe operar un computador, cómo deben ser almacenados o transmitidos los datos y qué acciones debe tomar bajo una variada gama de circunstancias pre-definidas.

El lenguaje informático y la mayoría de los lenguajes involucrados dentro de la programación son lenguajes que intentan estar relativamente próximo al lenguaje humano convencional.

Una característica relevante de los lenguajes de programación es precisamente que más de una persona o también llamado programador pueda usar un conjunto común de instrucciones que sean comprendidas entre sí , esto da paso a que una programación pueda ser generada para una tarea múltiple o una tarea compleja.

Luego de definir y explicar lo que significa lenguaje de programación, se define a continuación el más básico lenguaje de programación llamado programación C.



Figura 1.El universo de la programación.

### 2.1.3 LENGUAJE DE PROGRAMACIÓN C

Es un lenguaje de programación originalmente desarrollado por Dennis M. Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del los anteriores lenguajes de programación como el B.

Este es un lenguaje orientado a la implementación de Sistemas Operativos. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Entre sus características poseen extensiones al lenguaje de programación que posibilitan mezclar código ensamblador con código C o acceder directamente a la memoria o dispositivos periféricos.

Uno de los objetivos de diseño del lenguaje C es que sólo sean necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución. Este lenguaje se usa como intermediario entre diferentes lenguajes.

Aunque este lenguaje posee un nivel básico de programación, está disponible en un amplio abanico de plataformas (más que cualquier otro lenguaje). Además, a pesar de su naturaleza sencilla, el lenguaje se desarrolló para incentivar la programación independiente del computador. Un programa escrito cumpliendo los estándares e intentando que sea portátil puede compilarse en muchos computadores.

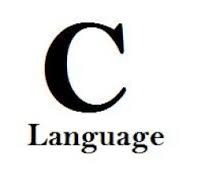
****

Figura 2. Logo típico del lenguaje de programación C.

### 2.1.4 LENGUAJE DE PROGRAMACIÓN C++

Este tipo de lenguaje de programación fue diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje compuesto.

Posteriormente se añadieron facilidades de programación genérica, que se sumaron a los paradigmas de programación estructurada y programación orientada a objetos. Por esto se suele decir que el C++ es un lenguaje de programación multipropósito.

Una particularidad del C++ es la posibilidad de redefinir los operadores, y de poder crear nuevos tipos que se comporten de una manera fundamental.

El nombre C++ fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases", significa "incremento de C" y se refiere a que C++ es una extensión de C.

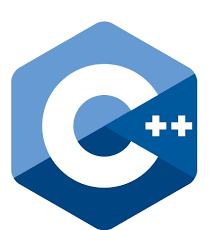


Figura 3. Logo típico del lenguaje de programación C++.

### 2.1.5 PRINCIPALES DIFERENCIAS ENTRE C Y C++

Entender la diferencia entre C y C++, a pesar de parecer para muchos un tema sencillo, es una pregunta muy común entre estudiantes y personas que apenas comienzan a aprender del mundo de la programación. El gran conflicto surge cuando no se les ha hecho notar la diferencia entre C y C++.

De esta forma se procederá a mostrar las principales diferencias entre la programación C y C++.

* **La extensión del archivo:**

Los archivos de C se guardan con extensión **C**, ejemplo práctica C mientras que los archivos de C++ se guardan con extensión **CPP** (C plus plus), ejemplo practica.cpp.

* **Operadores lógicos:**

Los conocidos operadores lógicos AND, OR y NOT pueden ser utilizados como palabras en el lenguaje C++ o como operadores, mientras que en C deben utilizarse únicamente los operadores &&, || y ! En el lenguaje C sería:

|  |  |
| --- | --- |
| 1  2  3  4 | if ( ( a == b ) || ( a == c) )  {     ...  } |

Mientras que en lenguaje C++ el anterior es válido, pero también es válido:

|  |  |
| --- | --- |
| 1  2  3 | if ( ( a == b ) or ( a == c) )  {  } |

* **Las bibliotecas:**

Las bibliotecas en C terminan con h (son HEADERS). Esas mismas bibliotecas sirven para C++ pero por convención, se elimina la terminación ‘.h’ y mejor se agrega ‘c’ al principio.

**En C                      En C++**

math.h                  cmath

time.h                   ctime

Como se puede observar la diferencia entre C y C++ es variada, y existen muchas más diferencias entre estos dos lenguajes de programación. Se considera que estos son los más importantes si se esta aprendiendo ambos lenguajes, o avanzando de C hacia C++.

### 2.1.6 LENGUAJE DE PROGRAMACIÓN HTML

Este lenguaje de programación obedece a las siglas HIPER TEXT MARKUP LANGUAJE, que en español significa lenguaje de marcas de hipertexto, hace referencia al lenguaje para la elaboración de plataformas y páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código denominado código HTML, el cual se utiliza para la definición de contenido de una página web, como texto, imágenes, videos, entre otros.

HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la WORLD WIDE WEB. Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

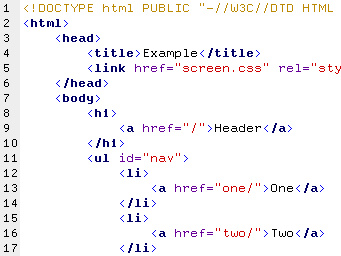


Figura 4. Ejemplo esquemático del lenguaje de programación HTML.

Luego de dar explicación a los lenguajes de programación más importantes, se dará inicio a los más comunes cursos de programación que servirán como base para creación de la plataforma computacional a diseñar.

## 2.2 CURSO DE PROGRAMACIÓN C:

Este curso es sumamente básico y puede ser desarrollado por cualquier persona que posea solo conocimientos básicos de informática, está separado por etapas.

La descripción de este curso se detalla a continuación:

* Nombre: “Aprende a programar”.
* Fuente: http://www.youtube.com
* Estado: Gratuito y abierto a todo usuario.
* Nivel de dificultad: Básico a intermedio.
* Tópicos del curso: C, C++ y HTML.



Figura 5. Página principal curso “aprende a programar”.

Este curso posee 44 videos claramente explicados. La estructura detallada de cada etapa se muestra a continuación:

### 2.2.1 VIDEOS EXPLICATIVOS PROGRAMACIÓN C:

Bloque 1.1: ¿Cómo empezar a programar?

Bloque 1.2: Primer hola al mundo.

Bloque 1.3: Variables.

Bloque 1.4: Asignación de valores y operaciones.

Bloque 1.5: Ampliando conocimientos sobre asignación y operaciones.

Bloque 1.6: Método de la burbuja.

Bloque 1.7: Errores comunes.

Bloque 2.1: Estructuras selectivas.

Bloque 2.2: Ver si un numero es par y hallar máximo 3 números.

Bloque 2.3: Estructuras IF, ELSE, anidadas y SWITCH.

Bloque 2.4: Condiciones múltiples.

### 2.2.2 VIDEOS EXPLICATIVOS PROGRAMACIÓN C++:

Bloque 3.1: Estructuras iterativas WHILE y FOR.

Bloque 3.2: Ejercicios de FOR Y WHILE.

Bloque 3.3: Bucle de WHILE.

Bloque 3.4: Bucles anidados y ejercicio cronómetro.

Bloque 3.5: Introducción y depuración en C++.

Bloque 3.6: Comprobar si un numero es primo.

Bloque 3.7: Rutas de escape y calculo de números primos.

Bloque 4.1: ¿Qué es una función?

Bloque 4.2: Ejemplos de funciones.

Bloque 4.3: Funciones tipo VOID y Factorial de un número.

Bloque 4.4: Punteros y paso por referencia.

### 2.2.3 VIDEOS EXPLICATIVOS HTML:

Bloque 5.1: ¿Qué son los vectores?

Bloque 5.2: Vectores con funciones y ejemplos.

Bloque 5.3: Vectores con punteros y ejemplos.

Bloque 5.4: Descripción de una matriz y ejemplos.

Bloque 5.5: ¿Qué son los STRINGS? Y como compararlos.

Bloque 5.6: Ejemplo de cómo acceder a un carácter de STRINGS.

Bloque 6.1: ¿Qué son las estructuras?

Bloque 6.2: Formulario con estructuras.

Bloque 6.3: Traductor inglés/español.

Bloque 6.4: Ejemplo de gestión de varias bibliotecas.

Bloque 6.5: Punteros a estructuras y funciones con estructuras.

Bloque 7.1: ¿Qué es la memoria dinámica?

Bloque 7.2: MALLOC Y REALLOC con vector dinámico.

Bloque 7.3: CALLOC Y FREE con vector dinámico.

Bloque 7.4: STRINGS dinámicos y función EXIT.

Bloque 7.5: Matrices dinámicas.

Bloque 7.6: Funciones y memoria dinámica.

Bloque 7.7: Estructuras dinámicas y ejemplo sistema de batallas.

Bloque 8.1: Lectura de ficheros de texto con FGETC y FGETS.

Bloque 8.2: Ejemplo lectura de ficheros e ingresar datos en estructura.

Bloque 8.3: Escribir en fichero de texto con FPRINTF.

Bloque 8.4: Bloque final.

Al completar el curso en su totalidad mostrado anteriormente, se está en condiciones de pasar al segundo curso de programación y dominio web, el cual ya adquiere un carácter más importante debido a la complejidad de este.

Antes de mostrar al segundo curso es sumamente importante mencionar y explicar cuáles son los lenguajes de programación involucrados dentro él.

De esta manera se procede a dar algunas definiciones de dichos lenguajes:

## 2.3 DEFINICIONES IMPORTANTES

### 2.3.1 LENGUAJE DE PROGRAMACIÓN XHTML:

Este lenguaje de programación debe su nombre a las siglas EXTENSIBLE HIPERTEXT MARKUP LANGUAJE (que en español significa Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas informáticas.

Ante la llegada al mercado de un gran número de dispositivos, XHTML surge como el lenguaje cuyo etiquetado, más estricto que HTML, va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. XHTML puede incluir lenguajes más variados que HTML.

XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados, etc.

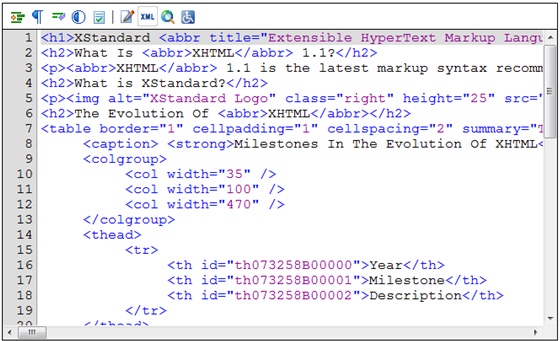


Figura 6. Ejemplo esquemático del lenguaje de programación XHTML.

### 2.3.2 DIFERENCIAS ENTRE HTML Y XHTML:

La siguiente lista muestra algunas reglas de XHTML que lo diferencian de HTML.

* Los elementos vacíos deben cerrarse siempre:

Incorrecto: <br>

Correcto: <br></br> o <br/> o <br />

Nota: Cualquiera de las tres formas es válida en XHTML. Para compatibilidad según el "Apéndice C" debe usarse<br />.

* Los elementos no vacíos también deben cerrarse siempre:

Incorrecto: <p>Primer párrafo<p>Segundo párrafo

Correcto: <p>Primer párrafo</p><p>Segundo párrafo</p>

* Los elementos anidados deben tener un correcto orden de apertura/cierre (el que se abre último, debe cerrarse primero).

Incorrecto: <em><strong>Texto</em></strong>

Correcto: <em><strong>Texto</strong></em>

* Los valores de los atributos deben siempre ir encerrados entre comillas (simples o dobles).

Incorrecto: <td rowspan=3>

Correcto: <td rowspan="3">

* Los nombres de elementos y atributos deben ir en minúsculas.

Incorrecto: <A HREF="http://www.domname.com">Domname</A>

Correcto: <a href="http://www.domname.com">Domname</a>

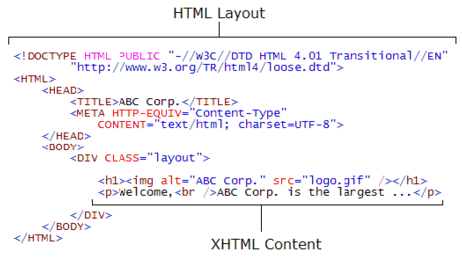


Figura 7. Ejemplo esquemático entre Programación HTML Y XHTML.

### 2.3.3 LENGUAJE DE PROGRAMACIÓN JAVASCRIPT:

Este lenguaje abreviado comúnmente "JS" es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, y dinámico. Se utiliza principalmente en su forma del lado del cliente (CLIENT SIDE), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JAVASCRIPT del lado del servidor (SERVER SIDE JAVASCRIPT o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, es también significativo.

Este lenguaje de programación se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación más semejante al C++.

Todos los navegadores modernos interpretan el código JAVASCRIPT integrado en las páginas web.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías.

Desde el lanzamiento en junio de 1997, han existido varias versiones de JAVASCRIPT, hasta la actualidad que se utiliza la versión ECMASCRIPT.

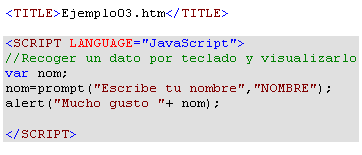


Figura 8. Ejemplo esquemático del lenguaje de programación JAVASCRIPT.

### 2.3.4 USO DE JAVASCRIPT EN PÁGINAS WEB:

El uso más común de JAVASCRIPT es escribir funciones incluidas en páginas HTML o XHTML y que interactúan con el DOM o Modelo de Objetos del Documento de la página. Algunos ejemplos sencillos de este uso son:

* Cargar nuevo contenido para la página o enviar datos al servidor a través de AJAX sin necesidad de recargar la página (por ejemplo, una red social puede permitir al usuario enviar actualizaciones de estado sin salir de la página).
* Animación de los elementos de página, hacerlos desaparecer, cambiar su tamaño, moverlos, etc.
* Contenido interactivo, por ejemplo, juegos y reproducción de audio y vídeo.



Figura 9. Formulario ejemplo de lenguaje de programación JAVASCRIPT.

Luego de dar explicación a los lenguajes de programación más importantes que se utilizaran en el segundo curso, se dará inicio a éste, que servirán como base para creación de la plataforma computacional a diseñar.

## 2.4 CURSO TUTELLUS:

TUTELLUS es la plataforma colaborativa para Aprender y Enseñar en español más grande del mundo. En la escuela de TUTELLUS se ofrecen Video-cursos para que aprender a realizar tus propios cursos de la mejor manera posible.

Entre sus temas y cursos más importantes se destacan los del tipo informático, en donde, se extrae el curso de programador y desarrollador web desde cero.

La descripción de este curso se detalla a continuación:

* Nombre: “Desarrollo web desde cero”
* Fuente: http://www.tutellus.com
* Estado: Curso pagado.
* Nivel de dificultad: Avanzado.
* Tópicos del curso: XHTML y JAVASCRIPT.



Figura 10. Logo página web de cursos TUTELLUS.

Este curso posee 57 videos separados por clases, claramente explicados mediante un profesor. La estructura detallada de cada etapa se muestra a continuación:

### 2.4.1 VIDEOS EXPLICATIVOS XHTML:

* Clase 1: Introducción a XHTML.
* Clase 2: Creando tu primer código XHTML.
* Clase 3: HEADERS, parágrafos y LINE BREAKS.
* Clase 4: Negrito, itálico y comentarios.
* Clase 5: LINKS.
* Clase 6: Imágenes.
* Clase 7: Tablas.
* Clase 8: CELLPADDING y CELLSPACING.
* Clase 9: DOCTYPE y Listas.
* Clase 10: Empezando con CSS.
* Clase 11: Más CSS
* Clase 12: Imágenes con CSS.
* Clase 13: BORDERS.
* Clase 14: WIDTH Y HEIGHT.
* Clase 15: Estados de LINK.
* Clase 16: Propiedades de tabla en CSS.
* Clase 17: Propiedades de lista en CSS.
* Clase 18: DIVS.
* Clase 19: ID y CLASS.
* Clase 20: CSS externo.
* Clase 21: Posición ABSOLUTE, RELATIVE y FIXED.
* Clase 22: Formularios.
* Clase 23: Radio BUTTON y CHECKBOX.
* Clase 24: OPTION MENU.
* Clase 25: WEBHOSTING y TEMPLATES.
* Clase 26: Proyecto final.

### 2.4.2 VIDEOS EXPLICATIVOS JAVASCRIPT:

* Clase 32: Introducción a JAVASCRIPT.
* Clase 33: Tu primer código JAVASCRIPT.
* Clase 34: Variables.
* Clase 35: PROMPTS.
* Clase 36: ARRAYS.
* Clase 37: IF STATEMENT.
* Clase 38: IF ELSE STATEMENT.
* Clase 39: ELSE IF STATEMENT.
* Clase 40: FOR LOOP.
* Clase 41: WHILE LOOP.
* Clase 42: BREAK y CONTINUE.
* Clase 43: .Funciones.
* Clase 44: Métodos y propiedades.
* Clase 45: Más Métodos.
* Clase 46: Objetos matemáticos.
* Clase 47: Eventos.
* Clase 48: Creando objetos.
* Clase 49: Iniciador de objetos.
* Clase 50: LOGIN básico.
* Clase 51: POO.
* Clase 52: DOM.

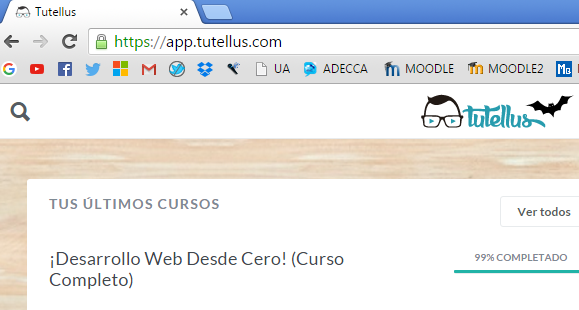


Figura 11. Ilustración Curso TUTELLUS en la web.

Una vez realizados los 2 cursos de programación ya mencionados, es momento de dar comienzo a la etapa de creación de la plataforma computacional.

En el siguiente capítulo se mostrará en detalle los pasos realizados para la creación de dicha plataforma.

# **CAPÍTULO 3**: CREACIÓN Y MANEJO WEB EN BASE A LO APRENDIDO.

**Luego de realizar los cursos de programación necesarios para el dominio de las variables involucradas, se procede a mostrar los software necesarios para la práctica y ejemplos de programación web.**

## 3.1 SOFTWARE NOTEPAD++.

Es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. De soporte nativo a Microsoft Windows. Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. Este software es fundamental para la estructura de programación.

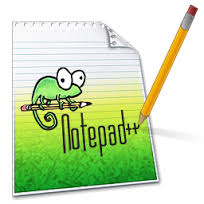


Figura 12. Logo del software NOOTEPAD++.

### 3.1.1 PROGRAMACIÓN BÁSICA EN NOTEPAD DE LA PLATAFORMA A DISEÑADA

La plataforma computacional diseñada básicamente cuenta con una estructura prototipo, la cual, es explicada por la mayoría de los diseñadores web.

La estructura de la plataforma diseñada cuenta principalmente por 4 pestañas: INDEX (portada), monitoreo, sobre y contacto. La programación de dichas pestañas se muestra a continuación.

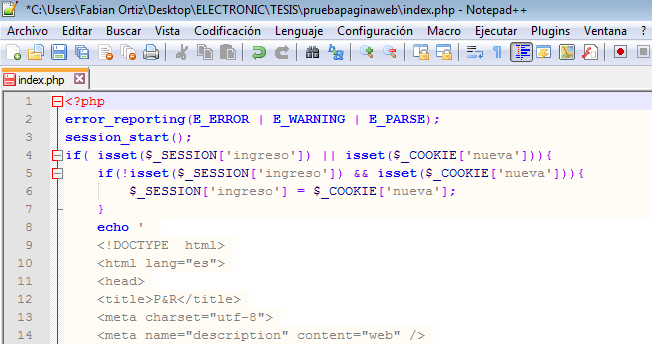


Figura 13. Extracto programación INDEX en NOTEPAD++.

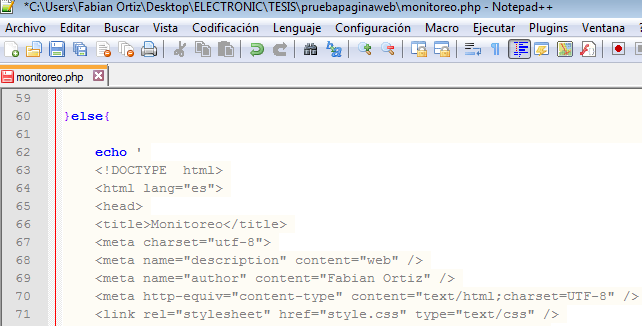


Figura 14. Extracto programación monitoreo en NOTEPAD++.

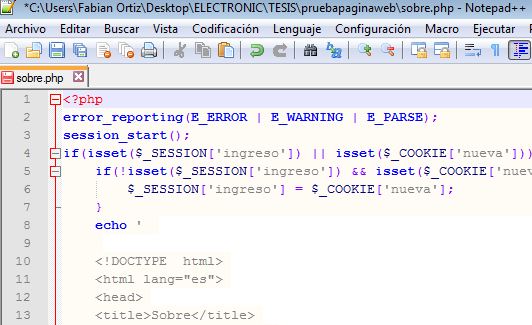


Figura 15. Extracto programación sobre en NOTEPAD++.

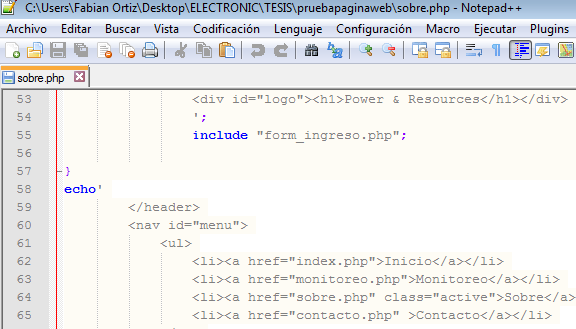


Figura 16. Extracto programación contacto en NOTEPAD++.

# CAPITULO 4 : IMPLEMENTACIÓN DEL HOST PARA VISUALIZACIÓN EN INTERNET.

Una vez realizada la programación básica de la página web diseñada, llega el momento de adquirir el dominio web necesario para poder visualizar la página en la web.

Para adquirir el dominio web es necesario adquirir un patrocinador quien vende sus dominios. El patrocinador elegido es “DOMINIOS CHILE”, la cual se muestra en la siguiente ilustración.



Figura 17. Página web patrocinador dominio.

Es muy importante señalar que el nombre del dominio web adquirido es:

http://pyrproyectos.com

La explicación de la elección del nombre para el dominio se detallará en la estructura de la plataforma computacional.

Existe una página web especializada en la configuración y entendimiento de los dominios web enlazados y adquiridos llamada “CPANEL”.

## 4.1 SOFTWARE CPANEL:

Es una herramienta de administración, su nombre viene del acrónimo panel de control y está basada en tecnologías web para administrar sitios Web. Aunque fue diseñado para el uso comercial de servicios de alojamiento web, el software no es gratuito y se distribuye solamente a los proveedores de alojamiento web.

Su interfaz es muy accesible y práctico, el cual se muestra a continuación:

****

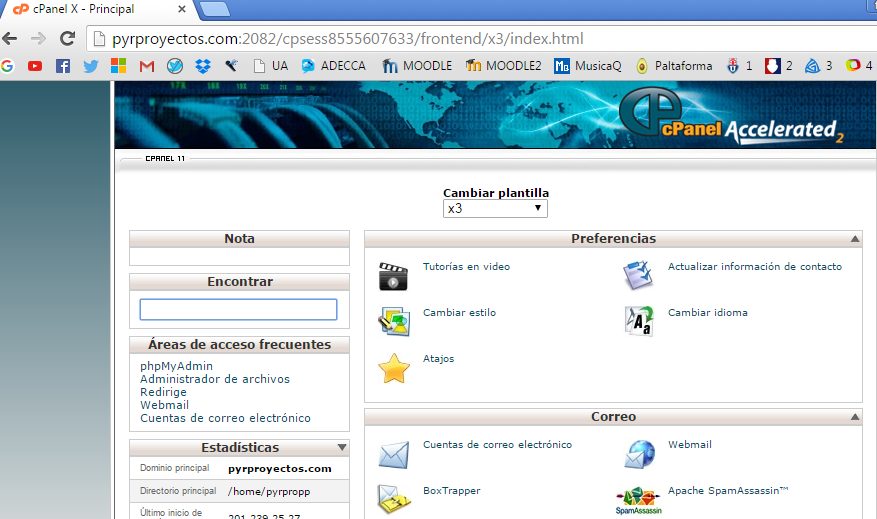
****

Figura 18. Página web directorio CPANEL.

### 4.1.1 ENLAZAMIENTO PLATAFORMA COMPUTACIONAL DISEÑADA Y DOMINIO WEB.

Luego de adquirir el dominio web con un patrocinador y administrarlo mediante la herramienta web CPANEL, es necesario enlazar la página web creada con el dominio mencionado. Este proceso se realiza con un software que se menciona a continuación.

## 4.2 SOFTWARE FILEZILLA:

Cuando se está aprendiendo a programar o trabajando en entornos web, se hace necesario un programa que permita subir y bajar archivos a un servidor y gestionar los archivos en el servidor. Hay diferentes alternativas para ello. Una de ellas, de gran interés, es FILEZILLA.

La transmisión de archivos entre un ordenador personal y un servidor se hace mediante un protocolo denominado FTP y que se corresponde con la terminología inglesa FILE TRANSFER PROTOCOL. El protocolo FTP se basa en la arquitectura cliente-servidor.



Figura 19. Logo Software FILEZILLA.

La estructura del software FILEZILLA es muy práctica y sencilla, en la cual, para comenzar a realizar un proyecto es necesaria la previa identificación del dominio web adquirido y posteriormente se enlazan los archivos de programación creados en el software NOTEPAD++ que gracias a FILEZILLA son subidos y actualizados a tiempo real en el dominio web adquirido.

De esta manera se muestra la ventana de funcionamiento del software FILEZILLA.

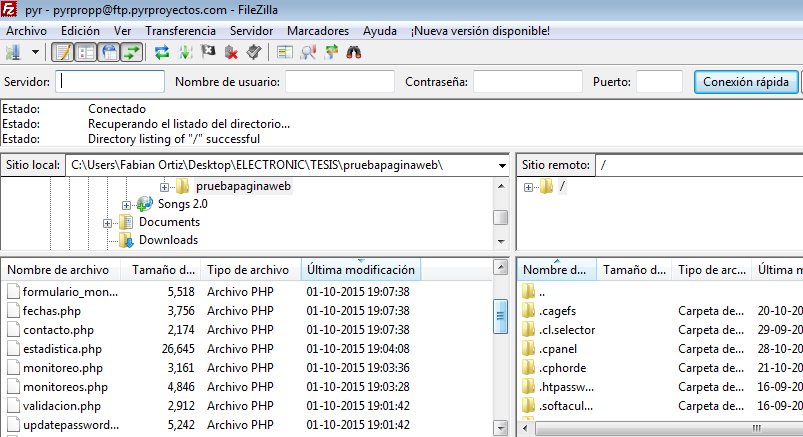


Figura 20. Ventana Principal software FILEZILLA.

Como se observa en la figura 20, en la parte izquierda se muestran todos los archivos de programación de la plataforma computacional diseñada, mientras que en la parte derecha de la imagen se muestra la ventana en la cual se deben arrastrar los archivos que serán vistos en la página web gracias al dominio y el HOST adquirido.

## 4.3 ESTRUCTURA DE LA PÁGINA WEB DISEÑADA.

Después de diseñar la plataforma computacional en bruto con los conocimientos de programación adquiridos y de adquirir el dominio web para la visualización de dicha plataforma en la web mediante los software mostrados anteriormente, es necesario ahora dar una breve explicación acerca de la estructura que presenta la plataforma.

Como se mencionó en la etapa de adquisición del dominio web, la plataforma computacional se llama “pyrproyectos”, cuyas siglas provienen del inglés POWER Y RESOURCES (poder y recursos en español).

La estructura de la plataforma web es muy similar a la mayoría de las páginas web que se basan en identificación usuario y contraseñas, que a pesar de tener cuatro pestañas todas son muy similares solo variando en su contenido.

A continuación se detalla la estructura de la plataforma computacional.

### 4.3.1 HEADER O CABECERA:

Aquí se muestra el título de la página web, además del lugar donde se deben identificar los usuarios que utilizarán la plataforma mediante una contraseña.

El HEADER tiene además la opción de registrar un nuevo usuario que desee ingresar con una nueva identificación y una opción de recuperación de contraseña si fuese necesario.



Figura 21. Ilustración HEADER plataforma computacional.

### 4.3.2 BODY O CUERPO:

En esta parte de la plataforma es donde se visualiza el contenido principal de información como en la mayoría de las páginas existentes.

En el caso de la plataforma computacional diseñada se muestra un ejemplo de esquema básico de un monitoreo remoto de variables físicas de un proceso industrial, en donde convergen las cuatro principales variables de un sistema HIDROPACK industrial, las cuales son, temperatura, presión, nivel y flujo.

Un sistema HIDROPACK del tipo industrial es un sistema presurizado que permite mantener presión y caudal de agua constante. Este sistema de presurización controla en forma automática la alimentación de agua y otro líquido que se desee presurizado, hacia y desde el HIDROPACK. Está equipado por una llave en mano, con una bomba de alimentación, válvulas, de acumuladores hidroneumáticos, de los diferentes transmisores e indicadores de variables y del sistema de monitoreo de las variables físicas el cual suministra las lecturas de dichas variables a la plataforma computacional como base de datos de información.



Figura 22. Ilustración de las pestañas de la plataforma.

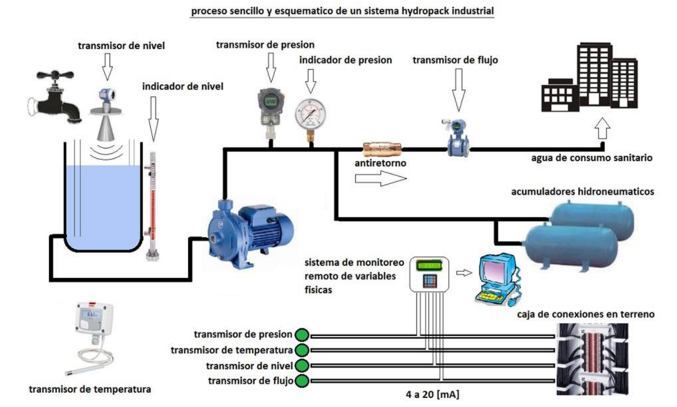


Figura 23. Esquema de HIDROPACK de la plataforma.

**4.3.3 PESTAÑA MONITOREOS:**

Esta pestaña muestra los diferentes monitoreos realizado por el módulo que puede contener las variables antes mencionadas.

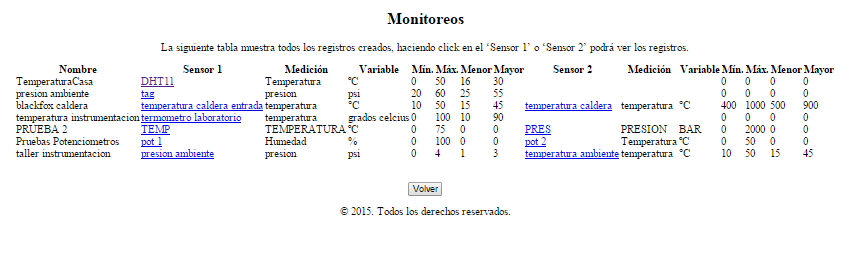


Figura 24. Pestaña de registros creados por monitoreos.

### 4.3.4 PESTAÑA SOBRE:

En esta pestaña se pretende la creación de una breve reseña del creador de la plataforma, además de anexos relacionados con los tópicos presentes en la página web.

Es muy importante señalar que esta pestaña aun se encuentra en construcción tal y como lo muestra la figura 25.



Figura 25. Pestaña sobre de la plataforma.

### 4.3.5 PESTAÑA CONTACTO:

En esta pestaña se describe la manera de como poder realizar alguna pregunta específica sobre lo mostrado en la plataforma ó para simplemente poder contactarse con los creadores.

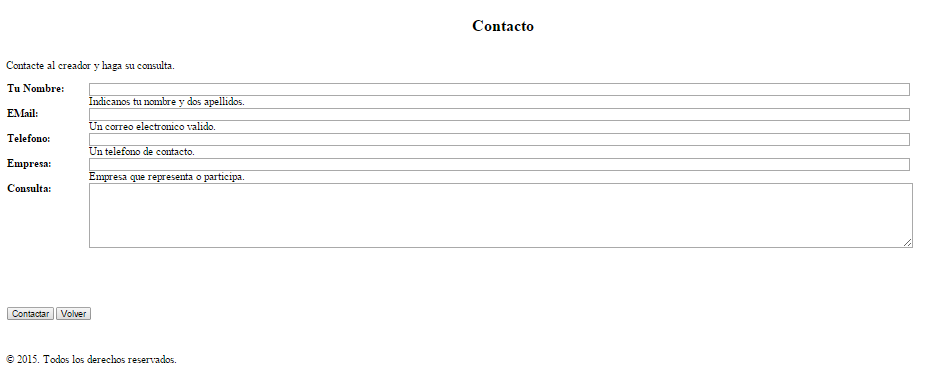
****

Figura 26. Pestaña contacto de la plataforma.

Después de darle una forma básica a la plataforma computacional diseñada y comprobar su programación lógica se da por finalizada la primera parte del proyecto.

# CAPITULO 5: RESULTADOS ESPERADOS Y MEJORAS A CORTO PLAZO

## 5.1 RESULTADOS ESPERADOS

Para esta fecha, se tiene realizado un 40% del proyecto final realizado, es decir, falta aún un 60% del trabajo total. La finalidad es poder obtener la aprobación de los usuarios en cuanto al tamaño y forma de utilización de la plataforma diseñada, en donde se pueda comprobar su programación y funcionamiento lógico de sus principales funciones, además de darle estilo como colores, tamaños, etc, además de dar a conocer las aplicaciones que puede llegar a otorgar en su completa totalidad una vez ya finalizado el proyecto.



Figura 27. Portada Plataforma computacional creada a la fecha.

## 5.2 MEJORAS A CORTO PLAZO

El 60% de lo restante del proyecto finalizado tiene que ver con una lista de acciones a realizar para finalizar el trabajo en un 100%.

Los objetivos y tareas faltantes se enlistan a continuación:

* Finalización etapa final del curso TUTELLUS “Estilos y mejoramiento de tu presciencia online”.
* Detalles de estilo, colores, lógica programable y de funcionamiento completo de la página web, todo esto en base al aprendizaje del último curso de TUTELLUS.
* Enlazar la página web con el circuito a diseñar, es decir, lograr recibir datos desde el dispositivo electrónico que se va a diseñar en el otro proyecto de trabajo de título de Erasmo Páez, el cual, como se nombró en la propuesta de este proyecto, van de la mano y influyen directamente uno en el otro.

# REFERENCIAS BIBLIOGRÁFICAS

* Curso “Aprende a programar”.

https://www.youtube.com/watch?v=9idgIGmQvAQ&list=FLLupsAOIpsk7KNSmFhqug9g

* Curso “Desarrollo web desde cero”.

https://app.tutellus.com/cuaderno/desarrollo-web-desde-cero-curso- completo-1759

* “Introducción a páginas web y navegadores con el editor NOTEPAD++”. (Documento en formato PDF).

http://mermaja.act.uji.es/docencia/ib14/data/introHTML.pdf

* “Curso HTML + CSS” (Documento en formato PDF).

http://www.laigu.net/wp-content/uploads/2009/08/Curso-HTML-%2B-CSS.pdf

* “Manual básico del CPANEL” (Documento en formato PDF).

http://panel.redcoruna.com/manuales/Manual%20cPanel.pdf

* “Mini manual de uso y configuración FILEZILLA” (Documento en formato PDF). http://ceaip-zac.org/old/datos/trasparencia/informes/Filezilla.pdf

Algunas características que debe tener esta plataforma son:

* Sistema de ingreso por usuario, cada módulo de control de acceso se le asignará un usuario como administrador para su gestión.
* Un administrador podrá gestionar la cantidad que desee de módulos.
* Registrar y editar a los usuarios de un módulo, con información personal y asignación de privilegios.
* Registrar y revisar eventos y actividades críticas llevadas a cabo por los usuarios del sistema.
* Configurar eventos que envíen una notificación digital al administrador.
* Impedir el acceso no autorizado a los sistemas de información, bases de datos y servicios de información.
* Garantizar la seguridad de la información cuando se utiliza computación móvil e instalaciones de trabajo remoto.

1. Es una red que conecta los ordenadores en un área relativamente pequeña. [↑](#footnote-ref-1)
2. Es una red de gran cobertura en la cual pueden transmitirse datos a larga distancia. [↑](#footnote-ref-2)
3. Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. [↑](#footnote-ref-3)
4. Enrollment, en la bibliografía inglesa. [↑](#footnote-ref-4)
5. En inglés, Sample. [↑](#footnote-ref-5)
6. En inglés, Identification register. [↑](#footnote-ref-6)
7. En inglés, Feature set. [↑](#footnote-ref-7)
8. En inglés, Template. [↑](#footnote-ref-8)
9. En inglés, smart cards. [↑](#footnote-ref-9)