**Curso Gratis de Programación Básica**

**RECAP CURSO GRATIS DE PROGRAMACIÓN BÁSICA**

**RECAP PROGRAMACIÓN BÁSICA**

Luego de terminar este curso puedes saber que el internet funciona con 3 herramientas las cuales: HTML, CSS, JavaScript, que son los tres lenguajes de programación para crear aplicaciones web.

HTML (Lenguaje de marcas de hipertexto), es el lenguaje donde se define la información o el contenido del documento. El formato de los archivos es .html.

CSS (Lenguaje de estilos en cascada), es el lenguaje donde se especifica el diseño del documento, maneja todo lo relacionado con la parte visual. El formato de los archivos es .css.

JavaScript, es el lenguaje que hace que todo sea interactivo y que nos permite crear sitios web. El formato de los archivos es .js. Además, es el lenguaje de programación que interpreta el navegador.

Nunca pero NUNCA se te olvide que JAVASCRIPT no es JAVA.

* Java es un lenguaje para servidores, aplicaciones de escritorio y aplicaciones Android.
* JavaScript es el lenguaje de la web, servidores, robots, etc.

Todo lenguaje de programación, no solo javascript: trabaja con variables, condicionales, eventos, funciones, clases y arreglos.

Una variable es un elemento que se emplea para almacenar y hacer referencia a otro valor. Gracias a las variables es posible crear “programas genéricos”, es decir, programas que funcionan siempre igual e independientemente de los valores concretos utilizados. Por otra parte, las condicionales se definen como: es el código que se ejecuta siempre y cuando una circunstancia sea cierta.

Los eventos por su paso son funciones que suceden cuando algo ocurre, sucesos cómo un clic, pulsar una tecla, colocar el mouse sobre un botón, etc. Las funciones son una herramientas que nos permite escribir código que vamos a reusar múltiples veces. La clase es lo que define las características del Objeto. Un ejemplo de una clase es: la clase Persona, donde las características de la clase serían: altura, peso, género, edad, nombre, etc.

Los arreglos son un conjunto de datos ordenados por posiciones y todas asociados en una sola variable. Los datos pueden ser de cualquier tipo, es decir, es posible crear un arreglo que tenga un string en la primero posición, un número en el segundo, un objeto en el tercero y así sucesivamente. Podremos acceder a estos distintos datos de manera independiente o agrupándolos. Cabe resaltar que un arreglo es un objeto. Los objetos son envolturas para código. El navegador tiene algunos objetos nativos como:

* ***Navigator***: Es el objeto que contiene las funciones del navegador. También te permite acceder al sistema operativo como el GPS, guardar datos en el disco duro, etc.
* ***Window***: Es el objeto que maneja cada una de las pestañas.
* ***Document:*** Es el objeto que contiene todo lo que podamos ver de nuestra página.

En aplicaciones web tenemos un concepto llamado DOM (Document Object Model) es la forma en que internamente el navegador organiza todo el HTML dentro de una estructura de árbol.

**La creación de aplicaciones web se basa en:**

* HTML (.html): donde se define el contenido.
* CSS (.css): donde se define el diseño y la parte visual.
* JAVASCRIPT (.js): crea el dinamismo.

**Todo lenguaje de programación trabaja con determinados elementos:**

* VARIABLE: Elemento que almacena valores.
* CONDICIONAL: Circunstancia requerida para correr código.
* FUNCIÓN: Conjunto de instrucciones que crean nuevas variables.
* CLASE: Define las características del objeto.
* ARREGLO: Conjunto variado de datos con orden definido y agrupados en una variable.

**FUNDAMENTOS DE PROGRAMACIÓN**

**¿QUÉ ES HTML/CSS/JS?**

HTML, CSS y JavaScript son los tres lenguajes que están en el centro de crear aplicaciones web, en este curso vamos a enseñarte principalmente JavaScript.

Vemos cada uno:

* **HMTL** **(lenguaje de marcas de hipertexto),** es el lenguaje donde se define la información o el contenido del documento, el formato de los archivos es .html.
* **CSS (cascading style sheets),** el lenguaje donde se especifica el diseño del documento, maneja todo lo relacionado con la parte visual, el formato de los archivos es .css.
* **JAVASCRIPT,** el lenguaje que hace que todo sea interactivo, relativo el lenguaje de programación que nos permite crear sitios web, el formato de los archivos es .js.

El que realmente interpreta estos lenguajes es el Navegador.

TIP: Se puede escribir CSS y JavaScript dentro de HTML, los profesionales normalmente escriben esto por separado.

**¿CUÁL ES LA DIFERENCIA ENTRE PROGRAMACIÓN Y INTERACTIVIDAD?**

Programación es código que tu escribes para que nuestros dispositivos hagan algo. Mientras que la interactividad es el resultado de dicha programación.

* HTML = no tiene interactividad
* CSS = no tiene interactividad

JavaScript = si tiene interactividad con la información de html

**JAVASCRIPT NO ES JAVA**

Java es un lenguaje para servidores, aplicaciones de escritorio y aplicaciones Android

JavaScript es el lenguaje de la web, servidores, robots, etc.

**PRIMEROS PASOS EN EL NAVEGADOR CON ALERT**

Ya tienes un navegador y un editor de texto, eso es lo único que requieres.

Ve a la consola, desde ahí puedes escribir JavaScript sin tener que crear un archivo.

Primero mostraremos un mensaje, escribe:

alert("Hola mamá, estoy programando");

Lo que has hecho es ejecutar una función, estas son colecciones de código que hacen algo, en JS siempre que quieres ejecutar una debes escribir paréntesis de apertura y de cierre, las comillas delimitan un texto (string) y punto y coma (;) se utiliza para terminar la instrucción.

También, podemos crear operaciones básicas.

var x = 1;

var y = 2;

var z = x + y;

Hemos declarado variables, las declaramos con las palabras reservada *var*, el nombre, operador de asignación (=) y el valor que va a tener.

Recuerda: todo en programación tiene que ver con seguir las reglas de un lenguaje y luego crear las instrucciones que quieres para lograr los objetivos que deseas.

TIP: En consola puedes hacer más grande el tamaño de la letra con CTRL +.

Todo en la programación tiene que ver con seguir las reglas de un lenguaje y luego crear las instrucciones que quieres para que cumpla el objetivo que deseas.

***Parámetros*** (se colocan entre paréntesis): es la información que una función necesita para lograr algo, en el caso de este ejemplo el mensaje está colocado entre “”, esto se debe a que es una cadena de texto (*string*).

En JavaScript es una buena práctica colocar “;” al final de cada línea de código, esto facilitara la lectura del mismo. Hay líneas de código que no terminan en “;”, son conocidos como bloques de código.

***Alert:*** es un método que dispara algo.

***Var***: declara la existencia de una variable en memoria.

**HTML, CSS, JAVASCRIPT DE VERDAD**

Los archivos HTML funcionan con etiquetas, por ejemplo:

*Hola mamá* ***ya casi aprendo***

Todos los contenidos en HTML deberían estar dentro de una etiqueta, y todos los archivos HTML tienen esta estructura, cómo <title>Título de la página</title> Contenido de la página

La etiqueta que te permite escribir CSS dentro de HTML es <style>, se coloca dentro de head. La etiqueta para escribir JS dentro de HTML es <script>, se coloca antes de terminar el body.

Recuerda:

* Los nombres de las variables tienen algunas reglas, cómo, no pueden tener espacio, debe empezar siempre con una letra, las minúsculas y las mayúsculas importan.
* Identación, tienes bloques de código, estos deben ir un poco a la derecha.
* Los programadores pasan 80% de su tiempo leyendo código y 20% escribiendo código, por esto es importante seguir buenas prácticas. Es importante ver las extensiones de los archivos.

Los bloques de código no necesitan “;”, solamente las líneas de código.

**Document.write** es una función de JavaScript con la cual podremos escribir en el HTML.

Muy importantes las reglas para nombrar variables en JavaScript:

* No pueden empezar con números, sólo con letras
* Se diferencian mayúsculas y minúsculas
* No se pueden utilizar caracteres “raros” como ñ o tildes

**PRIMER PROYECTO: PESO EN OTRO PLANETA**

**PESO EN OTRO PLANETA**

¿Cuánto pesas en la tierra? ¡Calculemos tu peso en otro planeta!

Creemos nuestro primer algoritmo que nos permita saber nuestro peso en otro planeta, luego escribamos esto en código usando JavaScript.

*Recuerda:*

* Los títulos se pueden agregar con las etiquetas h1, h2, h3, h4, h5, h6.

**OBTENIENDO DATOS DEL USUARIO**

¿Qué tal si obtenemos ahora datos del usuario?

*Recuerda:*

* La consola nos sirve para saber el estado de las variables
* Cuando tienen un valor en comillas (“”) es un texto
* Puedes usar la función prompt para recibir datos del usuario
* Concatenar es unir cadenas de texto a variables.

Podemos convertir un *string* en un numero con la función **ParseInt();**

**FLUJO Y CONDICIONALES**

¿Qué tal si ahora podemos elegir el planeta?

Creemos primero un algoritmo para resolver nuestro problema, para esto debemos aprender a usar condicionales.

Para escribir una condicional usas la palabra reservada if, puedes escribir una como:

if (planeta == 1) {

// Código si la condicón se cumple.

} else {

// Código si la condición no se cumple.

}

Si quisiéramos tener varias condiciones podemos escribir nuevas condiciones con la palabra reservada else if:

if (planeta == 1) {

// Código si la condición se cumple

} else if (planeta == 2) {

// Código se la segunda condición se cumple

} else {

// Código si las condiciones no se cumplen

}

*Recuerda:*

* El nombre de los archivos deberían seguir las mismas reglas que los nombres de variables,
* El código debería ser fácil de leer.

**EN RESUMEN**

**JAVASCRIPT:**

* Es un lenguaje de programación.
* Hace que las páginas webs sean **dinámicas.**
* Es el mismo tipo de archivo que un archivo de **texto plano.**
* Su extensión es **.js**.

**DIFERENCIAS ENTRE JAVASCRIPT Y JAVA**

* JavaScript fue creado para **dar interactividad** en la web.
* js es JavaScript, pero del **lado del servidor (Backend).**
* JavaScript sirve para **programar robots *(Nodebots).***
* Java es de Oracle.
* Java se creó inicialmente **para servidores.**
* Es particularmente **popular en bancos.**
* Java es el lenguaje en el que se **crean aplicaciones para Android.**

**REGLAS DE LOS NOMBRES DE VARIABLES EN JAVASCRIPT**

**Reglas de los Nombre de Variables:**

* Las variables **no permiten espacios.**
* El primer carácter de una variable **siempre tiene que ser una letra,** no un número.
* No se pueden usar **signos raros como la ñ** o los acentos.
* Las mayúsculas y minúsculas **no son indiferentes.**

**COMANDOS DE JAVASCRIPT**

* **Alert():** sirve para **disparar un mensaje** en la pantalla del navegador.
* **Prompt():** sirve para **pedir datos** al usuario (hacer un **input**).
* **Var:** sirve para **abrir un espacio en memoria** (en este caso, una variable) donde se pueden guardar un tipo de dato, ya sean numérico o de texto.
  + **Numérico:** para abrir un espacio en memoria (una variable) de tipo numérico es necesario añadirle **el valor numérico** que quieras en la variable.
  + **Texto (String):** para abrir un espacio en memoria (una variable) de tipo texto es necesario **abrir dos comillas “”** para que el navegador (o la computadora) lo detecte como texto.
* **write:** escribe texto en el **<body>.**
* **parseInt:** es una función de JavaScript que sirve para **convertir variables** de tipo int (entero) a tipo float (decimales).
* **parseFloat:** es una función de JavaScript que sirve para **convertir variables** de tipo int (entero) a tipo float (decimales).
* **.toFixed(2) elimina los decimales que sobran** y redondea el número. En el parámetro le indicas en número de decimales que quieres conservar.
* **;:** sirve para definir **donde termina un bloque de código** y dónde empieza el siguiente, no es obligatorio, pero es una buena práctica para que el código sea más fácil de leer.
* **=:** es para **asignarle** algo a un variable.
* **\n:** sirve para hacer un **salto de línea** en JavaScript.
* **If(){}:** sirve para **declarar una condición** con una comparación.
* **Else {}:** sirve para **declarar una condición si la comparación fue incierta.**
* **Else if:** sirve para declarar una condición con otra comparación además del de

**OPERADORES DE COMPARACIÓN**

* **Operador de desigualdad !=** se utiliza para **comparar si algo no es igual** en JavaScript.
* **Operador de igualdad ==** se utiliza para **comparar si algo es igual** en JavaScript.

**DEFINICIONES EN PROGRAMACIÓN**

* **Inspeccionar elemento:** es una herramienta que te permite **ver cómo están hechas las páginas webs por dentro** y se puede acceder a ella haciendo **Ctrl+Shift+I.**
  + **Console:** es un lugar donde puedes escribir JavaScript sin necesidad de programar en un archivo, además de saber el **estado de las variables** y si hay un **error en el código.**
* **Diagrama de flujo:** es la **representación gráfica** del algoritmo o proceso.
* **Función:** una función es **un procedimiento** en JavaScript.
* **Parámetros:** los parámetros **son la información que una función necesita** para lograr algo.
* **Concatenación:** la concatenación **es la unión de dos cadenas en una.**

**SEGUNDO PROYECTO: DIBUJANDO CON CANVAS**

**EL DOM: NUESTRO LUGAR DE TRABAJO EN LA WEB**

En programación existen objetos, estos son como envolturas para código, el navegador tiene algunos nativos como:

* **Navigator:** el objeto que contiene las funciones del navegador, también te permite acceder también al sistema operativo como el GPS, guardar datos en el disco duro, etc.
* **Window:** el objeto que maneja cada una de las pestañas.
* **Document:** el objeto que contiene todo lo que vemos dentro de nuestra página.

En aplicaciones web tenemos un concepto llamado DOM (Document Object Model) es la forma en que internamente el navegador organiza todo el HTML dentro de una estructura de árbol.

**CÓMO FUNCIONAN WINDOW Y DOCUMENT**

Veamos cómo utilizar el DOM.

Los objetos contienen

* **Métodos:** funciones dentro de un objeto.
* **Atributo o propiedad:** variables internas que almacenan valores.

*Recuerda*:

* Las únicas etiquetas que deben ir a la misma altura de la etiqueta html es head y body.
* location devuelve la ruta de la página.
* Para agregar soporte a tildes y letras especiales puedes usar dentro de head.
* Cuándo quieres depurar un programa puedes usar console.log para ver mensajes en la consola.

**OBJETOS**

Un objeto es una instancia de una clase, para que sea más entendible es un algo de algo con lo que lo relacionamos, por ejemplo, un vehículo. La palabra “Vehículo” es una clase, y el vehículo físico, es una instancia de esa palabra “Vehículo”.

El objeto está dividido en dos partes:

* **Características:** las características son todo lo que representa el objeto, por ejemplo, la cantidad de ruedas que tiene un vehículo, la marca del vehículo, el color, el tamaño, el número de placa, etc. A e esto se le llama **ATRIBUTOS** en el mundo de la programación.
* **Funcionalidades:** las funcionalidades son todas las funciones que tiene el objeto o los servicios que ofrece, por ejemplo, el vehículo nos ofrece el servicio de transportarnos, de escuchar música, de tener aire acondicionado, etc. A esto se le llama **MÉTODOS** en el mundo de la programación.

Ahora, construyamos el objeto Vehículo en código con todo lo que nos han enseñado hasta ahora:

// Atributos

var cantidad\_ruedas = 4; //int

var marca = "Mercedez"; //string

var color = "rojo"; //string

var tamano = "grande"; //string

var placa = "BDH865"; //string

var frenos = true; //boolean

// Métodos

function transporte() {

alert("Nos transportamos hasta la casa");

}

// Este es un método que recibe atributos como parámetros

// Cuando un método recibe un atributo como parámetro, signfica

// que cada vez que lo uses debes colocar un atributo entre los

// parentesis

function musica(cancion) {

alert("En este momento estamos escuchando la canción: " + cancion);

}

function aireAcondicionado(prendido) {

alert("El aire acondicionado esta: " + prendido);

}

Un boolean es un tipo de variable que solo puede tener 2 estados: True o False.

**DIBUJANDO EN EL DOM**

Vamos a dibujar en HTML, para esto usamos la etiqueta.

* Para dibujar usamos JS para darle las instrucción y los canvas funcionan con coordenadas.

Solamente llegando al canvas no podemos dibujar, debemos crear un contexto.

*Recuerda:*

* ID es el identificador con el que podemos encontrar etiquetas con JavaScript, los nombres siguen las mismas reglas que las variables.
* Puedes obtener un elemento con su id buscándolo con document.getElementById(‘nombre\_id’).

**¿QUÉ APRENDIMOS?**

* **getElementById**: obtener un elemento con su id del documento.
* **getContext(“2d”)**: método del objeto canvas.
* **beginPath:** arranca o activa el dibujo.
* **strokeStyle:** se comporta como variable, atributo o propiedad del objeto.
* **moveTo(100, 100):** función del canvas para definir el punto donde va a arrancar la línea.
* **lineTo(200, 200):** función del canvas para definir el punto final donde va a trazar la línea.
* **stroke():** dibuja la línea con el color que hemos elegido.
* **closePath():** cerramos el trazo, y lo damos por finalizado.

**FUNCIONES EN JAVASCRIPT**

La funciones son una herramienta que nos permite escribir código que vamos a reusar múltiples veces:

Puedes escribir una función en JavaScript así:

function nombreFuncion(parametros) {

// Código que ejecuta la función

}

Cómo crear una función con la que puedas crear las líneas que necesites.

*Recuerda*:

* Cuando escribes el mismo código muchas veces, es una buena idea mejor usar una función

Los parámetros de una función, son variables dentro de la función y solo funcionan dentro de ella.

**CICLOS WHILE Y FOR EN JAVASCRIPT**

Otro concepto fundamental en programación son los ciclos, los ciclos son piezas de código que se repiten hasta que se cumple una condición.

*Recuerda*:

* Ten cuidado de no crear ciclos infinitos que bloqueen el navegador.

**EVENTOS Y FORMULARIOS EN HTML Y JAVASCRIPT**

Los eventos son funciones que suceden cuando algo ocurre, sucesos cómo un clic, pulsar una tecla, colocar el mouse sobre un botón, etc.

Creemos un programa que reciba la cantidad de líneas con las que queremos crear nuestra imagen.

*Recuerda*:

* Cuando queremos enviar información a un servidor podemos usar un formularios
* En JavaScript puedes agregar un manejador de eventos con element.addEventListener('event', function)

**DETECTAR EVENTOS DEL TECLADO CON JAVASCRIPT**

Podemos hacer que nuestro dibujo reaccione al teclado, primero debemos aprender a detectar eventos del teclado.

*Recuerda*:

* Un programador divide los grandes problemas en problemas más pequeños.
* Todo manejador de eventos devuelve un objeto con los detalles del evento.

**PROBLEMAS A RESOLVER EN ESTA CLASE:**

1. ¿Cómo detectamos cuando el usuario presiona una flecha arriba, abajo, etc.?
2. ¿Qué evento es y cómo se llama? Keydown o keyup
3. **Detectar la tecla presionada con el evento keydown:**

document.addEventListener('keydown', nombreFuncion);

**CamelCase** es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en Camel Case se asemejan a las jorobas de un camello. El nombre Camel Case se podría traducir como *Mayúsculas/Minúsculas Camello.* El término tiene su origen en la disposición de los tipos móviles en casilleros o cajas.

**Existen dos tipos de Camel Case:**

* UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo:
* lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo: *ejemploDeLowerCamelCase:*

Atajo de teclado para abrir la consola en el navegador:

ctrl + shift + i

Recargar el navegador:

ctrl + r

Código de teclas para las flechas:

// Flecha Izquierda

keyCode = 37

// Flecha Arriba

keyCode = 38

// Flecha Derecha

keyCode = 39

// Flecha Abajo

keyCode = 40

Objeto para almacenar el valor numérico de las teclas: JSON (Javascript Object Notation)

var teclas = {

UP: 39,

DOWN: 40,

LEFT: 37,

RIGHT: 39

}

Estás líneas serían lo mismo que esto:

var telcas = {UP:38, DOWN:40, LEFT:37, RIGHT:39}

Estructura de control switch: (se utiliza en casos especiales porque no es muy legible el código).

switch(tecla.keyCode) {

case teclas.UP: console.log('arriba swith');

break;

case teclas.DOWN: console.log('abajo swith');

break;

case teclas.RIGHT: console.log('derecha swith');

break;

case teclas.LEFT: console.log('izquierda swith');

break;

default: console.log('Para dibujar utiiza las flechas del teclado');

}

**TERCER PROYECTO: VILLA PLATZI**

Vamos a empezar a crear las bases de lo que podría ser un videojuego

Para esto debemos resolver algunos problemas:

* ¿Cómo dibujar el mapa del juego?
* ¿Cómo dibujar los personajes?
* ¿Cómo definir las posiciones aleatorias de los personajes?

*Recuerda*:

* floor() devuelve el número entero por debajo de los decimales.
* ceil() devuelve el siguiente número entero arriba de los decimales
* random() devuelve un número aleatorio de 0 a 0.999..

**USO Y CARGA DE IMÁGENES EN CANVAS**

En canvas puedes insertar imágenes, hagámoslo con nuestra villa platzi.

*Recuerda*:

* Debemos agregar la imagen en el evento load del objeto.
* Debemos usar el método .drawImage del canvas para insertar la imagen.
* Canvas dibuja siempre encima del último objeto.

**CUARTO PROYECTO: PAKIMANES**

**DIVISIÓN, MÓDULO Y RESIDUO EN JAVASCRIPT**

Cuando te presentas a un trabajo de programador tal vez debas resolver un problema llamado fizzbuzz, este busca conocer tus conocimientos, veamos cómo resolverlos.

Primero debes conocer cómo calcular el residuo de una división, ahora puedes llevar esto a código,

*Recuerda*:

* Para calcular el módulo puedes usar %.
* Los programas dividen los programas complejos en problemas pequeños.
* La operación lógica and (y) se escribe en JavaScript con &&.

**SEXTO PROYECTO: CLIENTE/SERVIDOR**

**MODELO CLIENTE/SERVIDOR**

Las tecnologías utilizadas en aplicaciones web son:

* **Bases de datos**, MySQL es una base de datos relacionales y MongoDB es una base de datos no relacional
* **Backend**, existen muchos lenguajes que puedes usar cómo Python, Ruby, JavaScript
* Servidores, existen tecnologías como NGINX, Apache, Node
* **Frontend**, son las tecnologías que corren en el navegador, HTML, CSS y JavaScript

A un grupo de tecnologías se les conoce como Stack

*Recuerda*:

* Si tuvieras un código en el Frontend que se conectara a una base de datos, esta seria visible para todos.

**MATERIALES DE APOYO**

**LA WEB CON VISIÓN PROFESIONAL**

Crear para la web es un talento valorado estos días y nos encontramos en un mercado altamente demandante.

**LA WEB EN SU AUGE**

Poca oferta y mucha demanda en empresas y startups, incluso, aumenta diariamente.

**NO HAY SUFICIENTES PROFESIONALES WEB.**

Esos puestos deben cubrirse con **perfiles épicos, apasionados por la tecnología.**

Desgraciadamente, aparecen muchas personas que entran al sector sin disfrutarlo, más como obligación o porque está “caliente” el mercado. Y no debe ser totalmente así. Debes de gozarlo.

Las empresas quieren mejorar comunicación, costos, ventas, infraestructura, pero desconocen los “como” o peor aún, saben que existen, pero no cuentan con los profesionales aptos para esos puestos.

**BUENAS PRÁCTICAS EN LOS PROFESIONALES WEB**

La palabra “profesional”, de acuerdo al diccionario, significa:

* *Persona que realiza su trabajo con aplicación, seriedad, honradez y eficacia.*

**PRACTICAR, NO MEMORIZAR**

*“HTML5, CSS3, JavaScript, HAML, Slim, ERB, Stylus, SASS, LESS, Coffescript, Ruby, Rails, RVM, RubyGems, Bundler, Sinatra, GIT, MySQL, PostgreSQL, MongoDB, Python, Django, Responsive Design, Mobile First, API’s, NodeJS, BackboneJS, EmberJS, AngularJS.*

Hay miles de tecnologías y conceptos. **Y todas tienen un por qué.**

No te sientas mal si al principio te cuesta trabajo o aparecen millones de errores, **es natural**.

**SE CURIOSO Y BUSCA LOS ORÍGENES**

Todo tiene un por qué en la vida, en la Web también.

Si hay algo que no entiendes, búscalo y **trata de comprenderlo (no memorizarlo).**

Cuando encuentras el origen y la necesidad que cubre una tecnología, tendrás una ventaja y crearás estrategias mejor enfocadas.

**CONOCE DE TODO, PERO ESPECIALÍZATE**

Conforme vayas avanzando, descubrirás que hay muchas áreas.

**Por ello, prueba todas las diferentes áreas y cuando lo consideres, especialízate en la que más te guste.**

Cada área exige diferentes talentos y perfiles, por ejemplo:

* En Frontend es importante armonizar el sentido visual y usabilidad de un sitio.
* En Backend se pide solidez y lógica arquitectónica en servidores.
* En Marketing se determinan comportamientos de usuarios de acuerdo a analíticas, sintetizando procesos de venta.

Eso sí, todos son importantes y se complementan enormemente.

**PIENSA GLOBAL**

Cuando entras en este sector, empiezas a ver que salimos de nuestras fronteras en todos los sentidos.

No somos México, Colombia, Chile. **Somos desarrolladores y diseñadores internacionales.**

Debemos contar con la habilidad de contactarnos con clientes, empresas, comunidades sin importar la distancia o frontera. Todo se vale, no hay reglas.

Y es una lección muy difícil de aprender, pero todo se empieza con pasos pequeños.

Por otro lado, tus proyectos defínelos en función de que cualquier persona del mundo podría usarlos. **No sólo en ciertas regiones, en cualquier lado.**

**SÉ SOCIAL**

Si no te comunicas con personas que hacen lo mismo que tú, es muy probable que tengas dificultades para crecer profesionalmente.

Realmente necesitamos de todos para encontrar proyectos, empresas o ideas interesantes que se podrían ejecutar.

Ve a eventos, acércate a las comunidades, concursa, genera iniciativas.

No es sólo pasar miles de horas con código, es también influir positivamente en los demás.

**SÉ CONSTANTE**

La curva de aprendizaje en nuestro sector es alta, pero a cierto plazo, todo llega a tener sentido y agilizas. Sé organizado y aprende un poco cada día.

No te rindas si no llegar a entender nada al principio, **esto es un proceso de perseverancia.**

**¡ACTUALÍZATE!**

Es internet, existen blogs, redes sociales, personas con autoridad en ciertas especialidades web.

Dedícale un pequeño tiempo en fin de semana a conocer las novedades de la **industria, nuevos desarrollos y grandes tendencias** que se están suscitando.

**SÉ UNA PERSONA DE PROPUESTA, NO DE INSTRUCCIÓN**

La web necesita más personas que prueben, se equivoquen y jueguen con lo nuevo.

Por lo tanto, si trabajas todo el tiempo los mismos frameworks, herramientas, códigos durante un plazo exageradamente amplio, es probable que puedas estancarte.

Si en tu empresa te piden que trabajes con ciertas herramientas, sé estratega y muéstrales las bendiciones del desarrollo web moderno (con un proyecto, ejemplos y los resultados que se pueden generar).

* Todas las empresas quieren personas con iniciativa, más en nuestro sector.

**¡UN PORTAFOLIO, UN PORTAFOLIO, UN PORTAFOLIO! (CON URL)**

Un profesional web debe aprender a destacarse, a mostrar sus talentos de forma sintetizada y simple.

Ten un portafolio con todo tu trabajo online en una dirección de Internet (proyectos, networking, contenido etc.)

No copies y pegues el CV. Colabora, crea, promueve tus ideas.

**Muestra tu mejor trabajo y te aseguro un futuro brillantes en cualquier área.**