INVITACIÓN A PROGRAMAR LA RED

José del Carmen Rodríguez Santamaría
https://www.ejristos.com
Bogotá, Colombia

March 3, 2025

INTRODUCCIÓN

Así como existen demasiados lenguajes humanos para nuestra comunicación, también existen muchos lenguajes para nuestra comunicación con máquinas digitales y computadoras. Es imposible hablar de la mejor opción, pero hablar de las soluciones más populares en los últimos años es más fácil. El lenguaje dominante para programar la web es **HTML** (Lenguaje de marcas de hipertexto). Esto es lo que aprenderemos. Además, es obligatorio crear programas que produzcan páginas bonitas. Para ello utilizaremos **CSS** (Hojas de estilo en cascada). Y para hacer ajustes dinámicos y ejecutar nuestros propios cálculos, usaremos **JS** (JavaScript).

Programar es difícil pero aquí prometemos hacer las cosas accesibles.

Iniciamos la máquina en el capítulo 1, *Mi primer programa HTML*. Allí vemos la estructura básica de un programa en HTML y la ubicación apropiada para los **scripts**, fragmentos de código o líneas de programa, de CSS o JS.

En el segundo capítulo, *una aplicación web para una necesidad real*, construimos una escalera de programas de creciente complejidad para completar una tarea específica: servir como tutor web para aprender artículos y plurales en el idioma alemán. El Autor utiliza actualmente este programa y sirve como ilustrador

de todas las aplicaciones interactivas que son autocontenidas y que permiten una conversación entre el Usuario y el navegador:

El programa desafía al usuario con un sustantivo alemán. Su tarea es elegir el artículo adecuado entre tres opciones: masculino, femenino y neutro. Si el Usuario elige el artículo correcto, el programa responde con un mensaje de felicitación y se le plantea un nuevo desafío. Pero si el Usuario responde de forma incorrecta recibe un mensaje rojo y se le dan más oportunidades.

En el tercer capítulo, *Mi sitio web*, ofrecemos instrucciones para crear su sitio web.

Bogotá, Colombia, *José Rodríguez*
February 2025

Chapter 1

MI PRIMER PROGRAMA EN HTML

Nuestro primer programa desplegará un mensaje muy sencillo

1 Ejemplo. *El siguiente programa publica un mensaje de bienvenida en la pantalla.*

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>MiPrimerProgramaParaLaRed.html</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
    <h1>MI PRIMER PROGRAMA PARA LA RED</h1>
    <p>BIENVENIDO A UNA AVENTURA DE APRENDIZAJE!</p>
    <p>Aprenderemos paso a paso
        cómo hacer un tutor que ayude a memorizar
        artículos y plurales del lenguage alemán.
    </p>

</body>

</html>
```

2 Ejercicio. *Verifique que en el código anterior:*

- Por cada <, hay un >. El texto dentro de <> se llama **etiqueta**. Una etiqueta contiene señales para el intérprete HTML que es usado por el navegador web. Un **intérprete** es un programa que recibe como entrada un programa en un lenguaje que los seres humanos pueden entender fácilmente y lo traduce línea por línea a un lenguaje que el circuito de la computadora puede ejecutar. Un **compilador** hace lo mismo pero no línea por línea sino como un todo.

- La etiqueta <!doctype html> es única y denota una directiva global: el documento está escrito en HTML. Esto debe especificarse porque HTML es el lenguaje dominante en la web, pero no el único.

- El <html lang="en"> marca **el comienzo del documento HTML**, y también dice que el conjunto de caracteres usado es el del inglés.

- Tras la declaración de HTML como idioma oficial, el documento consta de dos partes: el **head** o encabezado y el **body** o cuerpo. Las directivas globales para el navegador y las agencias de clasificación, como Google, vienen en el encabezado. Pueden contener información para localizar su documento por título, autor, fecha, tema, palabras clave además de instrucciones para estilizar el documento. El cuerpo contiene las instrucciones específicas del documento dado.

- La etiqueta <meta charset="utf-8" /> es única y especifica que publicamos un documento en la codificación normal de la web. El 8 no se refiere al tamaño de los símbolos en una pantalla sino al tamaño de la memoria en bits que se asigna para cada símbolo.

- Hay etiquetas emparejadas que marcan un punto de partida al que le sigue algún código y que necesariamente debe terminar en algún punto.

- Cuando se empareja una etiqueta, se abre con < > y se cierra con < />. Por ejemplo, <html> y </html>, <head> y </head>, <body> y </body>. La etiqueta <html> es el comienzo del documento general, mientras que </html> indica el final del documento html. Reglas similares para <head> y <body> que definen secciones separadas.

Los programas pueden resultar difíciles de entender. Para facilitar esta tarea, se pueden hacer dos cosas: escribir una documentación como un documento separado, como éste, e insertar aclaraciones directamente en la página como comentarios. Algunos expertos consideran que el código es más claro cuando los comentarios y la documentación externa se vuelven superfluos e innecesarios. Exhortan a todo el mundo a escribir código claro que no necesite explicaciones. Esto es difícil.

3 Ejemplo. *La siguiente es la versión comentada de nuestro primer programa:*

```
<!---Este es un comentario en HTML, una exlicación para el lector humano. -->
<!---La siguiente línea contiene una declaración para el navegador. Dice: Este
<!doctype html>

<!---La siguiente línea contiene una declaración para el navegador. Dice:
    Este es el comienzo del documento html.
    Y, el alfabeto usado es el del inglés. -->
<html lang="en">

<!---Después de la declaración inicial,
    un programa html incluye dos secciones:
    1. El encabezado o head que contiene
       las directivas globales para el navegador
       y para las agencias de clasificación.
    2. El cuerpo o body, con instrucciones específicas. -->
```

```
<!---Comienzo del encabezado -->

<head>
    <meta charset="utf-8"> <!---Tipo de codificación para el sistema de
    <!---Título que se despliega en la barra del navegador-->
    <title>MiPrimerProgramaParaLaRedComentarios.html</title>
    <!--Ancho usado de la pantalla-->
    <meta name="viewport" content="width=device-width, initial-scale=1">

</head>
<!---Fin de la cabeza o head -->



<!---Comienzo del cuerpo o body -->

<body>
    <!---Se imprime un título en la pantalla -->
    <!---El texto esta formateado como h1, negra, grande -->
    <h1>MI PRIMER PROGRAMA PARA LA RED</h1>
    <!---Un texto ordinario es formateado como párrafo.   -->
    <p>Bienvenido a una aventura de aprendizaje!</p>
    <p>Aprenderemos paso a paso
        cómo hacer un tutor que ayude a memorizar
        artículos y plurales del lenguage alemán.
    </p>
    <!---Fin del cuerpo o body -->
</body>




<!---Fin del texto html -->
<!--Casi toda etiqueta html empieza con < >  y termina con <  />
-->

</html>
```

Todos los documentos html tienen una estructura o esqueleto básico.

1.1 LA PLANTILLA HTML

La estructura básica de un documento html es un esqueleto inmutable pero, de todos modos, se pueden realizar algunos pequeños cambios como en el lenguaje declarado.

4 Ejemplo. *La siguiente es la plantilla o esqueleto básico que define la estructura común a todos los programas HTML. Esta es la plantilla HTML en la que se puede insertar el código:*

```
<!doctype html>
<html lang="en">

 <head>
  <meta charset="utf-8" />
  <title>HTMLPlantilla</title>
   <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
            ...
  </style>
 </head>

 <body>
  <script type="text/javascript">
            ...
  </script>
 </body>

</html>
```

Podemos ver la ubicación adecuada tanto para las instrucciones CSS de estilo como para nuestra programación JavaScript personal.

5 Ejercicio. *Inserte comentarios en la ubicación adecuada del esqueleto html básico.* Respuesta

1.2 CORRAMOS NUESTRO PRIMER PROGRAMA

Veremos dos métodos para ejecutar los programas de este volumen: el primero para personas que no tienen mucho tiempo y el segundo para personas que quieren ganar independencia y tener tiempo para estudiar y practicar.

Nuestro primer método es el siguiente:

Todos los programas de este volumen fueron comprimidos por el autor en el archivo `InvitationWebProgramming.zip` que acompaña a este PDF y que también se puede descargar directamente. Ese archivo debe extraerse y cada programa se puede ejecutar por separado en un instante. Este archivo también contiene la fuente Latex de este PDF. Esto se hace porque el Lector tiene derecho a hacer algo mejor.

6 Receta. *El procedimiento para ejecutar un programa HTML que existe en un directorio determinado es el siguiente:*

1. Para ejecutar un programa guardado en un computador de escritorio: abra su navegador preferido, abra una nueva pestaña, haga clic sobre la ventana abierta, presione al mismo tiempo las teclas CTRL y O, donde la O viene de OPEN (no es un cero). Aparecerá entonces un cuadro de diálogo que contiene sus directorios y archivos. Elija su archivo y haga clic en ELEGIR o ABRIR o botón equivalente. El programa se ejecutará inmediatamente.

2. Para ejecutar el código en dispositivos móviles: copie y pegue el código en un documento nuevo y guárdelo con el nombre `MiPrimerPrograma.html`. Si ya tiene el programa como un archivo en su computador de escritorio, adjunte el archivo guardado a un correo, envíelo a Usted mismo, y guárdelo en su dispositivo móvil. Para ejecutar el programa desde el celular, simplemente toque su nombre.

El segundo método es el siguiente:

Un programa no es una intención. Es un hecho. Esto significa que está diseñado para ejecutarse y debe ejecutarse sin problemas. Usaremos un navegador, cualquiera, para ejecutar nuestro programa. Esto es posible si el programa ya existe como archivo en la memoria del ordenador, tableta o teléfono móvil utilizado. Por tanto, nuestra primera tarea es guardar un archivo apropiado.

Todos los programas en HTML, JS y CSS están codificados como texto sin formato, por lo que se pueden escribir y leer con los procesadores de texto más rudimentarios. Todo lo que se necesita es guardar programas con el sufijo correcto: `.html` para HTML, `.js` para JS y `.css` para CSS, pero usaremos solo un tipo de archivo, con sufijo `.html`, en el que se incluye todo, directrices html, métodos o procedimientos o funciones JS y estilo CSS. Así, para crear un documento html nuestro procedimiento es: utilice cualquier procesador de texto y guarde su trabajo como un archivo con sufijo `.html`.

7 Receta. *Para salvar un archivo, cuyo contenido sea nuestro primer programa, haga lo siguiente:*

1. *Resalte la porción de código sobre este documento pdf,*
2. *Péguelo sobre un documento nuevo en cualquier procesador,*
3. *Sálvelo con sufijo* `.html`*. Nombre completo:* `MiPrimerPrograma.html`

Un archivo correctamente escrito debe poderse correr en cualquier navegador, solo hay que abrirlo y listo. Para ello, el Desarrollador debe tener en cuenta todas las posibles incompatibilidades.

8 Ejercicio. *Corra el programa* `MiPrimerPrograma.html`*. Deberá ver un mensaje de bienvenida y una invitación.* *Respuesta*

La WEB ha sido construida sobre el criterio de que la privacidad y libertad del Usuario es el fundamento de la comunicación. Por eso tienes derecho a saber exactamente qué estás ejecutando. Para potenciar ese derecho, los navegadores tienen la posibilidad de mostrar el código. En Firefox, basta con hacer clic derecho en la ventana que contiene el programa y seleccionar `Mostrar el código fuente` o equivalente. En otros navegadores es posible que sea necesario buscar herramientas para Desarrolladores y elegir *ver el código fuente* entre las distintas ofertas.

9 Desafío. *La programación es difícil. Sería mejor decir que es extremadamente difícil. Por eso, muchos Desarrolladores argumentan que no quieren que Usted copie y reutilice en un instante el código que ellos desarrollaron durante largos meses de arduo trabajo. Investigue qué hacen para salvaguardar su trabajo pero cumpliendo aproximadamente con la exigencia de libertad total y el derecho a saber lo que se ejecuta.*

10 Ejercicio. *Guarde y ejecute un archivo con el siguiente código que representa un documento html con todas las complejidades posibles: esqueleto html, estilo css y código JS. Preste atención al hecho de que cada lenguaje tiene sus propios marcadores para los comentarios. El programa suma dos números: 1+2 = 3.*

```
<!doctype html>
<html lang="en">


<head>
    <meta charset="utf-8" />

    <!-- Título de este documento -->
    <title>MiPrimerProgramaConCSSyJS.html</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <style>
        /* Este es un comentario en css */
        /* Para un documento html, la estilización se hace en CSS */
        /* Si al background o transfondo no se le declara un color,
           aparece en blanco.*/
        body {
            background-color: lightblue;
        }
    </style>
</head>
```

```
<body>
    <!-- Imprime un mensaje importante con una subcadena, un submensaje, en itálica.  -->
    <h1>Mi primer <i>programa html-css-js </i> para el navegador</h1>
    <!-- Una línea en el documento recibe un nombre para uso futuro.
        Se imprime un mensaje en dicha línea -->
    <p id="demo">Haga Click sobr el botón y mire el efecto aquí.</p>


    <script>
        //Este programa en JS suma dos número y reporta el resultado.
        //Declaración de dos variables.
        var a = 1;
        var b = 2;

        //Esta función suma do snúmeros y retorna el resultado al hilo de ejecución del programa.
        function suma(a, b) {
            return a + b;
        }

        //Se llama a la función suma(a,b) y su resultado se imprime en el documento
        function principal() {
            //Las Variables a y b se imprimen, se llama el método o función suma
            // //y todo se une sobre una cadena que representa la respuesta total.
            let answer = a.toString() + " + " + b.toString() + " = " + suma(a, b);
            //La respuesta se imprime en la parte llamada demo que aquí es un párrafo.
            document.getElementById("demo").innerHTML = answer;
        }
    </script>

    <!-- Un botón debe ser cliqueado para correr el programa. -->
    <button type="button" onclick="principal()">Cliquée aquí</button>

</body>

</html>
```

¿Cómo funciona la interacción entre HTML y CSS? Cada elemento de HTML tine una lista de propiedades que el Usuario puede controlar. Un programa CSS es una especificación de algunas de dichas propiedades. En consecuencia, es una lista con asignación de valores. El Autor no se ha interesado en entender los detalles, por eso, su método es preguntarle a la web algo específico, ver qué código hay disponible, elegir el que mejor cuadre, copiar, pegar, modificar, y volver a modificar hasta que quede algo aceptable. Un experto es alguien que conoce los detalles, y sabe jugar con ellos.

1.3 EDITORES ESPECIALIZADOS

Si uno intenta comprender el código de un programa complejo usando un procesador de texto común, rápidamente se convierte en un héroe. La razón es que es extremadamente difícil distinguir qué es qué. Así, se han desarrollado editores especializados que

asignan diferentes colores a diferentes líneas de código en razón de sus funciones. Son maravillosos.

La forma más sencilla de familiarizarse con uno de estos editores es encontrar uno para el navegador utilizado. En Mozilla Firefox se busca una extensión con el tema `html editor`. Entre varias opciones, probé HTML5 Editor. Es suficiente para ver una visualización clara del código que se puede hacer más claro con la opción de la barra *embellecer código*. Se puede jugar con él para ver cómo abrir un archivo, cómo se ejecuta, cómo modificarlo, cómo guardarlo, es decir, cómo descargarlo pero con un nombre ad hoc dado por la extensión. Hay que cambiarle rápidamente el nombre por uno apropiado y fácil de recordar que indique su función. Para nuestro primer programa, el nombre correcto es `MiPrimerPrograma.html`.

Para correr un programa uno simplemente hace click en el botón run en la barra del navegador arriba a la derecha. El HTML5 Editor tiene una ventaja: gestiona las partes HTML, CSS, JS por separado en archivos distintos. Esta división se utiliza y se aprecia con frecuencia porque las diferentes partes tienen diversos grados de complejidad y las secciones fáciles se congelan y separan espontáneamente.

Con más tiempo e interés, se puede instalar un editor de código profesional gratuito. La mayoría de las personas señalan el *Visual Studio Code* de Microsoft como el mejor editor de código. Otras opciones reconocidas son Notepad++, Eclipse, Netbeans. El Autor utiliza todos estos para editar programas, a veces al mismo tiempo, pero Studio Code es su opción preferida. Para ejecutar programas ya guardados, utiliza Firefox o Chrome en Ubuntu. Windows tiene sus propias opciones, también muy buenas. Para editar un programa se siguen las mismas reglas que para un documento normal: se hacen cambios, se guardan, se prueban en el navegador, se hacen las correcciones... una y otra vez hasta que uno logre lo que busca.

Notepad++ es nativo en Windows pero se puede ejecutar en Ubuntu con Win. Es necesario seleccionar HTML como idioma en el menú Idiomas. Esto debe hacerse para cada programa. Esto activará el resaltado de sintaxis y será más fácil guardar el programa con el sufijo `.html`.

1.4 LOS NAVEGADORES PUEDEN SER DIFERENTES

Para tareas muy básicas, distintos navegadores se comportan de forma idéntica. Pero, a pesar de la gran uniformidad, existen diferencias. Por eso es normal que una aplicación un poco compleja que fue desarrollada para Firefox de escritorio no se ejecute en el celular que por defecto tiene Chrome. Así, cada navegador ofrece su propia documentación. El de Mozilla Firefox está en

https://developer.mozilla.org/en-US/

La documentación contiene explicaciones de cada detalle de su implementación de HTML, CSS, JavaScript y algunos otros elementos más avanzados. Hay que utilizar esta documentación para aprender y consultar cuestiones de la más variada índole. Y para repasar. Por ejemplo, uno puede ver una lista de las diversas áreas de HTML y que CSS puede embellecer.

1.5 CORRECCIÓN DE PROGRAMAS

La programación es difícil. Esta afirmación tiene dos significados. Primero: la programación es una de las empresas más apasionantes porque cada paso es un desafío, cuya solución produce una gran alegría. Segundo: una o dos mujeres en el mundo pueden desarrollar un programa no demasiado corto que se ejecute de inmediato. El resto de nosotros debemos sufrir mucho

para que nuestros programas funcionen porque siempre cometemos demasiados errores, fallas o bugs.

Una falla identificada en un programa de computadora se llama **bug**. Este nombre proviene de la historia: El ENIAC (Integrador Numérico Electrónico y Computadora), la primera computadora electrónica, estaba hecha de tubos y relés. Fue fabricado por los militares durante la Segunda Guerra Mundial para simular trayectorias balísticas en condiciones reales, pero comenzó a funcionar en 1946. Una mañana, la computadora no funcionó. Después de revisar todo, se encontró un bug, un bicho, un insecto, que interfería un relé. A partir de ese día, cada problema en una aplicación de software se denomina bug.

Los errores en los programas autónomos tienen dos fuentes principales: conceptos erróneos y mala codificación. Puede ser normal cometer muchos errores antes de poder ejecutar un programa, y mucho más para que produzca lo que debe. Sí: programar es difícil. Entonces, tratemos de entender de dónde viene la dificultad.

11 El origen de los bugs. *Aparte de la estricta sintaxis de todos los lenguajes de programación actuales, la dificultad de programar se debe a tres hechos muy simples:*

- Hay demasiadas opciones para hacer modificaciones.
- La mayoría de las modificaciones inutilizan los programas.
- Pequeños cambios que funcionan producen efectos muy diversos o de lo contrario el lenguaje carecería de suficiente expresividad.
- Decimos: todos los lenguajes, humanos o no, son caóticos o muy incipientes.

Por lo tanto, es extremadamente improbable realizar un cambio que sirva a un propósito predeterminado. Por eso hay que

estudiar mucho para triunfar, pensar muy bien, tener mucho cuidado y estar armado de una paciencia inmejorable.

El siguiente ejercicio es un desafío muy importante porque obliga a mirar más allá de la nariz.

12 Ejercicio. *Nuestra teoría está directamente relacionada con nuestro origen, el origen de la humanidad. Según la Ciencia, nuestro origen se explica por la Teoría Evolutiva del Origen de las Especies. Nuestra teoría y experiencia tienen algo que decir:*

1. *Demostrar que el ADN es software, es decir, que contiene instrucciones codificadas para fabricar cosas.*
2. *Demuestre que el lenguaje de programación ADN tiene todas las propiedades caóticas de nuestros lenguajes de programación.*
3. *Demostrar que la Teoría Evolutiva del origen de las especies es obviamente falsa.*

Respuesta

Ahora bien, ¿qué se puede hacer para aliviar nuestro sufrimiento ante la inmanencia de los bugs?

La comunidad ha desarrollado herramientas especiales para este propósito. Por ejemplo, Studio Code detecta inmediatamente errores sintácticos, por lo que pinta en rojo la línea donde se detecta el error. Un ejemplo de error sintáctico ocurre cuando un comando del idioma está mal escrito, por ejemplo, si en lugar de escribir

`<title>Éste es el Título</title>` uno escribe
`<title>Éste es el Título<title/>`

Studio Code y otros editores son extensibles, permiten agregar extensiones apropiadas para ayudar con un buen formato del código, lo cual facilita la lectura y la interpretación. Así, uno pasa el cursor sobre los íconos para descubrir el correspondiente a las extensiones, y escribe como mensaje de búsqueda: `html, css, js`. Se muestran muchas opciones y uno elige una con muchas estrellas y muchas funciones.

Los errores más terribles son aquellos en los que la sintaxis es correcta, pero el programa no se ejecuta, o se ejecuta pero infringe el propósito del Desarrollador. Estos errores se denominan genéricamente *errores de tiempo de ejecución*.

Existen herramientas especiales para **depuración**, los procesos de eliminación de errores, pero el Autor prefiere detectar errores mediante una regla simple: **un programa se conoce por sus frutos**. Entonces, cuando algo no funciona o funciona pero no como se desea, el Autor inserta un código especial para imprimir en la consola del navegador un comentario o el valor de una variable determinada. También se puede ordenar que se impriman esos mensajes en ventanas especiales que se llaman Áreas de texto. De esa manera, la depuración se puede convertir en una actividad pública en la que cada persona pueda ver con sus propios ojos que el programa se ejecuta según lo previsto.

La **consola** es una ventana en la que el navegador muestra mucha información, incluidos nuestros propios registros o informes. Abra la consola antes de iniciar el programa. Para abrirlo, vaya al menú del navegador, seleccione el elemento *Herramientas* o *Otras herramientas* o *Herramientas adicionales* o *Herramientas para Desarrolladores* y luego busque la consola. En Firefox, el menú consta de tres minibarras horizontales en la esquina superior derecha, lo cual en chino significa miércoles.

13 Ejercicio. *Descargue Visual Studio Code para su sistema, instálelo, agregue una extensión de 5 estrellas para* `html`*, vaya*

al menú `file`*, elija* `new text file` *y, en la ventana emergente, pegue el programa* `MyFirstCSS_JSprogram` *de este PDF. Cometa algunos errores menores de sintaxis para ver cómo reacciona Studio Code y corríjalos. Abra y ejecute el programa en un navegador. En el editor, busque en el programa la función que suma y cambie el + (suma) por un * (multiplicación). Guarde el nuevo programa. Abra y ejecute el nuevo programa en un navegador. Ese un error sumamente peligroso porque el navegador ejecuta el programa y todo parece estar bien, pero no hace lo que debe, sumar dos números.*

Chapter 2

UNA APLICACIÓN PARA UNA NECESIDAD REAL

El autor ha soñado con aprender alemán, pero no es muy bueno para ello. En particular, el alemán se basa en gran medida en las declinaciones, algo que no existe en inglés. Esta es una fuente de tremendas dificultades con varios pilares. Una de ellas es que los artículos dependen del género. Parece que el género de los sustantivos es caprichoso, por lo que es necesario memorizarlo o las declinaciones serán incorrectas y el sentido de lo que se dice puede perderse. Entonces, el Autor decidió hacer una aplicación para ayudarse a sí mismo y compartir su servicio con el mundo, es decir, la aplicación debe ser ejecutable por cualquier Usuario desde su propio escritorio o teléfono celular.

Si bien esta aplicación puede ser importante para unas pocas personas, ilustra la solución a una familia importante de problemas: crear una aplicación independiente que deba interactuar con las personas de una manera bastante compleja.

Dado que este capítulo pretende ser introductorio, analicemos ahora la tarea general que enfrentamos.

2.1 LA GUI (GRAPHIC USER INTERFACE) O INTERFAZ GRÁFICA

Pensemos en la sala de control de un avión o de un barco. Nos permite ver la información relevante para tomar decisiones y tiene los medios para emitir comandos. Nuestro propósito es aprender a hacer una simulación de esta situación en informática. El resultado se conoce como **GUI** (Graphic User Interface, Interfaz Gráfica de Usuario).

Dividiremos todo el diseño de una interfaz específica en muchos pasos, agregando más y más funciones hasta cumplir con la función completa. Nuestra primera tarea es mostrar cómo usar botones para publicar información.

2.2 BOTONES HTML

HTML tiene un botón básico, que no tiene otra función que existir tal como fue creado. Pero existen decenas de variantes que construyen diferentes funciones sobre el motivo básico.

14 Ejemplo. *Mostramos aquí dos tipos de botones. El más simple muestra un mensaje estático, invariante, y un* `onclick-button`*, que es una variante que muestra una alerta, un mensaje con información relevante, si uno hace clic en él. Esto es útil cuando uno tiene un mensaje largo para mostrar que se lee una vez y luego se oculta.*

```
<!doctype html>
<html lang="en">

<head>
 <meta charset="utf-8" />
 <title>HtmlButtons.html</title>
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <style> </style>
</head>

<body>
 <h1>The button Element</h1>
 <button type="button">I am a simple button. Look at Me!</button>
```

```
 <br>
 <button type="button" onclick="alert('Hello world!')">I am an onClick button. Click Me!</button>
</body>

</html>
```

15 Ejercicio. *Corra el programa* `HtmlButtons.html`

2.3 HERMOSOS BOTONES

La web ha sido desarrollada por personas que no se conforman con resolver tareas sino que además necesitan hacerlo con estilo, elegancia y distinción, convirtiendo cada producto en una bella obra de arte. Esto exige un esfuerzo terrible por parte del Desarrollador, que debe dominar muchos detalles finos de varias maquinarias complejas. Aruñaremos ese magnífico mundo mostrando algunos ejemplos sencillos pero significativos.

Nuestra primera tarea es ilustrar un ejemplo del uso de CSS para embellecer un botón de la forma más sencilla posible.

16 Ejemplo. *El programa* `HtmlButtons2.html` *muestra un botón adornado con la ayuda de CSS. Este programa muestra un botón con el nombre PROPÓSITO que, al hacer clic en él, presenta una alerta, una miniventana separada que se puede cerrar, con un mensaje sobre el propósito de esta aplicación web.*

17 Ejercicio. *Corra el programa* `HtmlButtons2.html`

18 Ejercicio. *Añada un segundo botón al programa anterior con el objetivo de reconocer los créditos de toda la comunidad de Desarrolladores que con sus blogs nos permitió aprender y progresar.* Respuesta

2.4 GRUPOS DE BOTONES

Cuando varios botones comparten las mismas propiedades, CCS permite agruparlos para facilitar la gestión global pero conser-

vando su individualidad, la capacidad de ser únicos en algún aspecto.

19 Ejemplo. *El Programa* `MasculineFeminine.html` *presenta dos opciones en las que se puede hacer clic. Denotan los géneros* `femenino` *y* `masculino`*, que conforman un grupo CSS, y al hacer clic presentan un mensaje. Estos botones permiten el cambio de color al pasar el mouse por encima. CSS permite cambiar cualquier propiedad de un botón al pasar el mouse.*

20 Ejercicio. *Ejecutar el programa* `MasculineFeminine.html`

21 Ejercicio. *Agregue* `Neutral` *como tercera opción al programa anterior.* Respuesta

2.5 MI PRIMER WEB-APP (APLICACIÓN PARA LA RED)

Nuestros botones sirven para habilitar la programación interactiva, aquella en la que el Usuario puede indicarle al programa el equivalente a una pregunta o petición y este responde con una acción adecuada.

22 Ejemplo. *El autor desarrolló una aplicación que le ayuda a aprender artículos en alemán. Para ello utilizó el material ofrecido con tanta delicadeza por* Anna Zwolinska-Simon *en su sitio web* `https://deutsch-mit-anna.de/lektion/nomen-liste/`*. La aplicación está en el programa* `ArticlesByAnna.htm`*.*

23 Ejercicio. *Ejecute el programa* `ArticlesByAnna.htm`*, pruébelo y evalúelo.* Respuesta

El problema con el programa anterior es que el número de sustantivos presentados es 30, pero el idioma alemán puede tener

80000. Para agregar más sustantivos utilizamos la lista presentada por Nery, que también tiene plurales:

http://frequencylists.blogspot.com/2015/12/the-2000-most-

Tomamos más de 500 sustantivos y plurales de la lista de Nery y agregamos una frase compuesta por Gemini, una IA de Google. Utilizando un traductor web (TWP = Translate Web Pages), una extensión de Mozilla Firefox, también proporcionamos una traducción de la frase al inglés.

Intentar aprender más de 500 artículos a la vez puede ser una tarea imposible. Pero trabajar con 10 artículos a la vez es realmente viable. Por lo tanto, nuestro plan es presentar el material en fragmentos de diez sustantivos. Para ello, planeamos utilizar un control deslizante HTML, que es una regla con un pulgar deslizante, cuya posición determina un valor que puede vincularse a un procedimiento, método o función en el código JS.

24 Ejemplo. *El código de un control deslizante se presenta en el programa* `HTML_Slider.html`. *Ejecútelo y échele un vistazo al código.*

25 Ejemplo. *Desarrollamos un programa que toma fragmentos de a 10 artículos,* `GermanArticlesPlurasSlider.html`.

26 Ejercicio. *Corra el programa* `GermanArticlesPluralsSlider.html` *y evalúelo.* *Respuesta*

27 Desafío. *Un problema con el programa anterior es que fragmentos de a 10 pueden parecer un poco aburridos para estudiantes ligeramente avanzados. Añada la opción de elegir fragmentos con 20 palabras o quizás con muchas opciones para que cada uno pueda regular a su gusto la complejidad de los ejercicios.*

Hemos optado porque se practiquen artículos en grupos de decenas. Pero perdimos la capacidad de poner a prueba nuestra memoria con muchos nombres tomados al azar. Nuestra solución es implementar dos modos de operación. El primero es para aprender, en trozos de a diez, y el segundo es para examen, en el que se toman en el ejercicio todas las palabras hasta la presente. La solución que ofrece HTML para implementar esta idea de dos modos de operación es un conjunto de radio-buttons, botones de opción, que son mutuamente excluyentes, es decir, solo se puede seleccionar uno de ellos en un momento dado.

28 Ejercicio. *Mire la web para ver un programa HTML que utiliza radio-buttons, botones de opción, para realizar una elección junto con una reacción al hacer clic en uno de ellos. Sugerencia: solicite a su máquina de búsqueda* html radio button onclick. *Respuesta*

29 Ejercicio. *Siguiendo los ejemplos de Internet estudiados en el ejercicio anterior, diseñe un programa HTML con dos opciones mutuamente excluyentes:* learning y examen *. Al hacer clic en cualquier botón, se debe realizar una acción. Respuesta*

Nuestro plan es que en el modo de aprendizaje, los sustantivos se den en grupos de decenas, por lo que el ejercicio consta de las últimas 10 palabras justo antes del pulgar deslizante. Pero en el modo Examen, el ejercicio toma todas las palabras antes del pulgar, comenzando con el primer sustantivo y terminando en el valor presente del pulgar.

30 Ejemplo. *Implementamos nuestras ideas en el programa.* `GermanArticlesPlurasTutor.html`. *Este programa incluye un modo de funcionamiento que puede adoptar dos valores: aprendizaje, por grupos de decenas, y examen, con muchas palabras presentadas al azar.*

31 Ejercicio. *Corra el programa* `GermanArticlesPlurasTutor.html` *y evalúelo.* *Respuesta*

32 Desafío. *Un problema grave con el programa anterior es que las palabras al principio de la lista se refuerzan más que las del final. Cree una solución para esto; de lo contrario, demuestre que no existe ninguna solución.*

33 Desafío. *Diseñar e implementar un sistema de calificación con promoción automática. Se concede un nuevo intento para practicar los casos defectuosos.*

34 Procedimiento para construir la lista de sustantivos. *La base de datos con los sustantivos y plurales utilizados por el Autor está contenida dentro del código. El autor sufrió mucho para crear esta base de datos, posiblemente debido a su uso de software y servicios web gratuitos. Para desafiar a todos a hacer algo mejor, contemos nuestra historia:*

1. La lista proporcionada por Nery fue pegada en una hoja de cálculo. Como la lista es compuesta, fueron necesarias varias columnas. Cada línea fue numerada para facilitar la localización en caso de que, de acuerdo al Autor, se necesitarse una corrección.

2. Se eligieron 100 sustantivos y luego se pegaron en una nueva página o ventana de un procesador de textos. Se guardó con un nombre apropiado, digamos `Nouns100To200.txt`.

3. Este archivo fue procesado por el programa `SeparateListForGemini.html` que agregó marcas de separación, comillas y comas.

4. El nuevo texto fue enviado a Gemini, una IA de Google, que nos ayudó gratamente con artículos y frases.

5. El resultado se completó con los datos de Nery y con traducciones realizadas por un traductor web (TWP = Translate Web Pages), una extensión de Firefox. Para ello, el material a traducir se pegaba en un borrador de correo electrónico para que pudiera ser recogido por los servicios de traducción de la web que ya teníamos instalados en Firefox.

6. Las traducciones se agregaron a una columna de la hoja de trabajo utilizada.

7. La salida fue separada por columnas. Para agregar marcas de separación, comas y comillas, el material se copió al portapapeles y se pegó en un procesador de textos en un documento nuevo. Esto se guardó con un nombre apropiado, por ejemplo, `NounsAll100To200.txt`.

8. El archivo guardado fue abierto y procesado por el programa `SeparateListForGemini.html`. Ese programa agregó comillas y comas e imprimió los resultados en un TextArea. El resultado es aceptado tanto por Gemini como por la estructura de matriz de JS en Visual Studio Code.

9. El resultado fue seleccionado y pegado en el inputArray del programa `GermanArticlesPlurasWebTutor.html`.

Este procedimiento fue repetido varias veces porque Gemini no acepta tareas largas.

Chapter 3

MI PROPIO WEBSITE (SITIO WEB)

Contamos con una aplicación desde el capítulo anterior. Es un programa que se guarda en la memoria de un computador de escritorio local y se ejecuta en un navegador web previamente instalado. ¿Cómo puedo hacer para que mi aplicación sea visible y ejecutable para todos los usuarios del mundo?

3.1 Servidores

Las computadoras pueden comunicarse entre sí y, si usted les da permiso, una computadora externa puede buscar en sus directorios, leer archivos específicos y escribir mensajes en el disco. Entonces, la web consta de un conjunto de computadoras que se permiten buscar libremente entre sí en un directorio cuyo nombre debe ser PÚBLICO. Una computadora con tales capacidades se llama **Servidor** y generalmente puede controlar otras computadoras y dispositivos.

Cualquier computadora puede convertirse en un servidor personal. Para que nuestra aplicación esté disponible para todo el mundo, basta con guardarla en el directorio público. Pero para ser percibido como un sitio web, hay que darle un nombre y

registrarlo en una oficina de control mundial de sitios web. El problema con esta solución es que hay que mantener encendida la computadora permanentemente. Esto es costoso en energía, operación y mantenimiento. En realidad, es extremadamente difícil tener un servidor funcionando sin interrupciones durante mucho tiempo. Una solución funcional es pagar *un trozo de memoria + operación + mantenimiento* en una computadora de una empresa que se especializa en ofrecer hosting. Esta es la solución más común para los laicos. Entonces, uno mira a su alrededor en dónde hay servicios de hosting, los compara en términos de diversas medidas y elige el mejor.

En el directorio público hay muchas aplicaciones. Por eso, un sitio web debe tener un encabezado que está definido por un programa en HTML que se llama `index.html` o `index.php` porque puede contener instrucciones en PHP, lenguaje que permite el análisis de respuestas de los clientes. Para ver cómo se ve un archivo `index.php`, se llama cualquier sitio web y se abre el código haciendo clic derecho o yendo a herramientas para Desarrolladores. Puede que se parezca mucho a nuestra aplicación y comparta el mismo esqueleto. Según el interés del Desarrollador, el índice puede contener enlaces a diversas imágenes, sonidos y programas.

Los sitios web compiten por las visitas. Un factor importante, quizás el decisivo, es la velocidad de reacción del sitio web. Por lo tanto, se recomienda evitar la belleza que exige demasiado tiempo para ser cargada.

Esto es básicamente todo.

3.2 CONCLUSIÓN

Con la llegada del desarrollo de programas mediante IA, puede parecer superfluo aprender a programar la web. En respuesta, nuestra política es simple: todo el mundo tiene derecho a convertirse en un experto, una persona que sabe los cómo y los por qué. Y, como hemos visto, conocer los fundamentos de la programación web no es tan difícil, y además contamos con mucha ayuda de excelentes y gratuitas herramientas desarrolladas por la Industria y por blogs escritos por la comunidad.

3.3 RESPUESTAS A LOS EJERCICIOS

5, página 9. El esqueleto podría ser comentado como sigue:

```
<!doctype html>
<html lang="en">

<!-- Este es un comentario html. No aparecerá en la página web. -->
<!-- Declaración de HTML como lenguaje de la página -->
<html>

<head>
    <!-- Comienzo del encabezado o head -->
    <meta charset="utf-8" />
    <title>PLANTILLA HTML COMENTADA</title>
    <!-- Inserte aquí los parámetros de la codificacfión y servicios  usados, tales como la visibilidad al teléfono móbil,
     y para cuales sistemas operativos.    -->
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <style>
        /*Este es un comentario en CSS */
        /* Ponga a quí los directivas de estilo CSS como
           el tamaño de la fuente- */
    </style>

    <!-- Todas las instrucciones html están entre una marca de comienzo y otra de final.-->
    <!-- Las marcas html son palabras reservadas delimitadas por    < > para el comienzo y por </ > para el final. -->
    <!-- Ejemplo:     <title>Mi Título</title>          -->

</head><!-- Final del encabezado-->

<!-- Comienzo del cuerpo o body, y de la programación  HTML y
de scripts en otros lenguajes, como JS.-->

<body>


    <!-- Ponga a quí las instrucciones en html-->

    <!-- Dentro de un documento HTML,
     puedes insertar código en varios lenguajes de programación.
     Por lo tanto, es necesaria una declaración del idioma,
     pero para JS se puede prescindir
     porque JS es en 2025 el idioma oficial de la web. -->

    <!-- Comienzo del código en JS -->
    <script type="text/javascript">
        //Este es un comentario en JS. Será ignorado por el intérprete,
        // es una especie de compilador que trabaja línea por línea.
        //Ponga a quí su código JS.
```

```
    </script>
    <!-- Final de la programación en JS -->

</body><!-- Final del cuerpo -->

<!-- Final del documento html -->

</html>
```

12, página 17. La relación de nuestra teoría con la Teoría Evolutiva del origen de las especies es la siguiente:

1. El ADN es software porque contiene instrucciones para sintetizar cadenas de aminoácidos y éstas instrucciones están codificadas según el código genético.

2. Consideremos la extremadamente grande variabilidad de perros, gatos y caballos. Esto significa que el lenguaje del ADN es muy expresivo y tiene demasiadas opciones para realizar cambios. Pero a pesar de todas las variaciones, la descendencia de un perro es un perro, la de un gato es un gato y la de un caballo también es un caballo. Esto implica que la mayoría de mutaciones, que son aleatorias, producen aborto, su fruto no es viable. En resumen: el lenguaje del ADN es caótico.

3. La Teoría Evolutiva dice que todas las especies están relacionadas por descendencia, por acumulación de caracteres divergentes que fueron filtrados por un proceso de selección: los que mejor se adaptan a su entorno dejan más descendencia y con el paso de las generaciones aparece la perfección y la complejidad. Entonces, el chimpancé, el gorila y el humano tienen un ancestro común que eventualmente podría ser un chimpancé, un gorila o un bonobo. Supongamos que efectivamente así fuera. Como el lenguaje del ADN es altamente caótico, tenemos una predicción obligatoria: junto con el surgimiento de cada especie, el Proceso Evolutivo debería haber dejado una cantidad innumerable de bugs, de rasgos imperfectos. Deben ser observables como

marcas hereditarias defectuosas que deberían estar profusamente representadas tanto en el registro fósil como en el cuerpo de cada bebé y, por tanto, en todos nosotros. ¿Es eso cierto? No. Esto no es cierto. Por tanto, la Teoría Evolutiva es obviamente falsa.

4. Para tener autoridad sobre este tema y ser una fuente de nuevas ideas, es necesario pasar varios cientos (¿o miles?) de horas programando y corrigiendo errores. En un caso así, simplemente se pregunta: *¿Aparecimos por Evolución? Si es así, ¿dónde están los bugs, los errores de programación?* Con estas preguntas todos entenderán que Usted es una persona más que ha entendido como experiencia personal que la Teoría Evolutiva es obviamente falsa.

5. Más detalles sobre el punto de vista del Autor en

   `https://www.ejristos.com/GPfalsifiesEvo/ScientificMethod`

18, página 23. El programa `HtmlButtons3.html` muestra dos botones que responden a los clicks. Uno muestra el propósito y otro los créditos reconocidos a la comunidad.

21, página 24. Program `MasculineFeminineNeutral.html` ofrece tres opciones para el tipo de artículo: masculino, feminino y neutro.

23, página 24. El programa es extremadamente útil para facilitar el aprendizaje de artículos en el idioma alemán.

26, página 25. La idea de tomar segmentos de a 10 sustantivos es buena.

28, página 26. Encontramos el programa para controlar botones de radio en

https://www.w3schools.com/js/tryit.asp?filename=tryjs

Este programa se puede ejecutar directamente en el servidor del sitio. El código es el siguiente (la versión en nuestro archivo `.zip` tiene algunas modificaciones para permitir ejecutar el programa en todas partes y sin quejas):

```
 <!doctype html>
<html lang="en">

 <head>
  <meta charset="utf-8" />
  <title>HtmlRadioButtons</title>
   <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="https://www.w3schools.com/js/tryit.asp?filename=tryjs\_form\_radio">



    <script>
        function check(browser) {
            document.getElementById("answer").value=browser;
        }
    </script>

</head>

<body>

    <p>What's your favorite browser?</p>

    <form>
        <input type="radio" name="browser" onclick="check(this.value)" value="Internet Explorer">Internet Explorer<br>
        <input type="radio" name="browser" onclick="check(this.value)" value="Firefox">Firefox<br>
        <input type="radio" name="browser" onclick="check(this.value)" value="Opera">Opera<br>
        <input type="radio" name="browser" onclick="check(this.value)" value="Google Chrome">Google Chrome<br>
        <input type="radio" name="browser" onclick="check(this.value)" value="Safari">Safari<br>
        <br>
        Your favorite browser is: <input type="text" id="answer" size="20">
    </form>

</body>
</html>
```

29, página 26. En el programa `ChooseOperationMode.html` se presenta un conjunto de dos botones de opción mutuamente excluyentes que ofrecen dos opciones *aprender y examen*. Al hacer clic, le dan un valor a una variable de 10 o 100.

31, página 27. El programa se ve bien y parece tener un efecto positivo en el alumno.