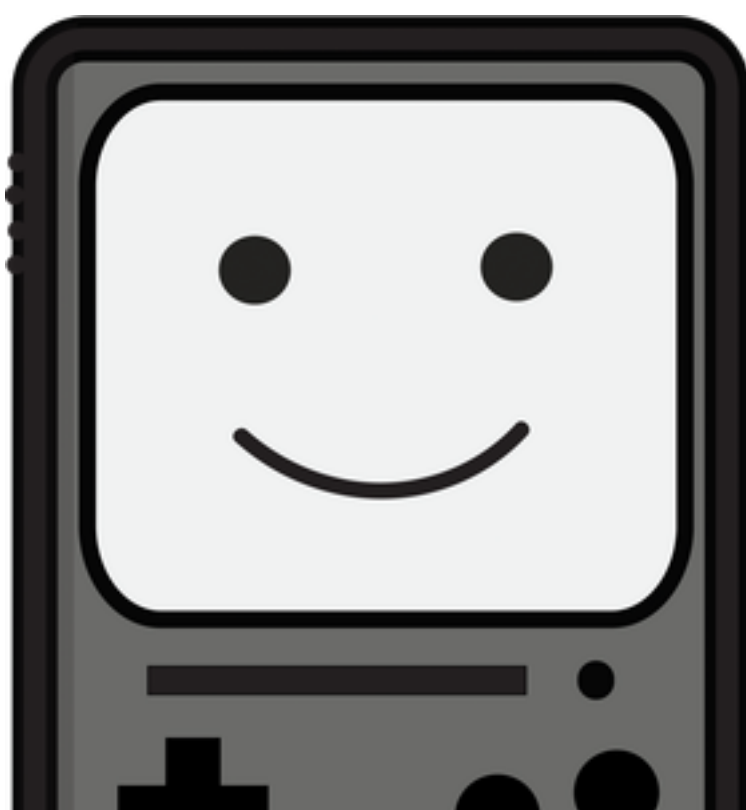


BMO EBOOK

Aprende como puedes iniciar tu carrera
como desarrollador de software

LEONARDO LUXBURG



BE MORE.

Capítulo 1

BMOPROJECT

bmoproject.com es un proyecto que nace a partir de la iniciativa de compartir conocimientos, pensamientos y habilidades. La idea se basa en generar y estructurar contenido educativo que guíe el aprendizaje y fomente el emprendimiento en el mundo digital.

El objetivo es servir como un canal de educación alternativa y formar una comunidad donde los participantes se impulsen unos a otros a desarrollarse, crecer y seguir aprendiendo.

Creo firmemente que la educación puede cambiar vidas. Me gusta pensar en la vida de forma binaria, compuesta únicamente por zeros y unos: prendido - apagado, pros - contras, vives - mueres, haces o no haces. Todas las decisiones en la vida se resumen a decir sí o no. De esta manera, se puede interpretar que todo es posible.

No existen metas irrealizables, todo está en nuestra decisión de hacer o no hacer lo que se requiere. Al final, depende de uno mismo. El propósito del Proyecto BMO es facilitar el aprendizaje y motivar a los participantes a que hagan todo lo que les apasiona, cumplan sus metas y no se rindan. Siempre se puede ser más, solo tienen que hacerlo.

Capítulo 2 Programación

La programación te permite crear un plan estructurado de instrucciones que cumplen una función, logran un objetivo o resuelven un problema.

Digamos que entras a trabajar a las 8:00 a.m. Para poder llegar a tu trabajo de manera puntual, antes de dormir organizas un plan basado en las tareas que debes cumplir para llegar a tu oficina. Tu día empieza temprano:

- 6:15 te levantas
- 6:30 te bañas
- 6:45 te vistes
- 6:55 preparas el desayuno
- 7:15 desayunas
- 7:30 manejas al trabajo
- 8:00 llegas a la oficina

El objetivo de llegar a tiempo se logró gracias a que seguiste puntualmente un proceso que permite el cumplimiento de este caso. El ejercicio de crear esa secuencia de pasos se conoce como *programación*.

La programación es un ejercicio lógico. No tendría sentido preparar el desayuno, salir, vestirse, desayunar y bañarse. Hay una secuencia razonable que te permite llegar de un punto A a un punto B de manera útil y eficiente.

Ese orden lógico es lo que se conoce como *lógica de programación*.

Por lo general, la programación hace referencia a la informática. Esto se debe a que casi todas las tecnologías tienen un flujo de trabajo que te lleva correctamente de un punto A a un punto B. Este flujo está compuesto de pasos que el sistema de la computadora debe seguir para cumplir una función.

Tomemos como ejemplo enviar un correo. Primero tienes que decirle al sistema quién eres, por eso inicias sesión. El sistema valida tus credenciales y te presenta la bandeja de entrada con todas las opciones esperando que le señales qué hacer. Haces click en “escribir nuevo correo” y el sistema pasa a solicitar la información del correo. Indicas el destinatario, escribes el asunto, redactas el contenido y al final oprimas “enviar” para mandar el mensaje. Todos estos son pasos que el sistema de la computadora debe seguir para enviar un correo.

Para poder informarle a la computadora qué hacer, qué opciones darte y como enviar un correo, se necesita un lenguaje de programación y el desarrollo de

software. Los *lenguajes de programación* están diseñados para ser interpretados y entendidos por máquinas. Esto significa que, escribiendo el código y los comandos del lenguaje, puedes comunicarle a una computadora los pasos e instrucciones que debe seguir para completar una tarea.

La construcción de la estructura del código de instrucciones se conoce como *desarrollo de software*.

El *código del lenguaje* está conformado por un set de herramientas basadas en datos que te permiten construir las instrucciones que el sistema tiene que seguir. Estos pueden ser datos como textos, números, booleanos (verdadero o falso), listas y muchos otros elementos que van a depender del lenguaje que utilices.

Todos estos datos tienen funciones y comandos únicos que les dan utilidad dentro de su lenguaje; como por ejemplo escribir mensajes, solicitar y almacenar datos, hacer operaciones matemáticas, contar múltiples elementos o estructurar el flujo para hacer un juego de cara o sello.

La utilidad y estructura de tu código van a depender del objetivo que tengas. Los lenguajes de programación te permiten crear tus propios objetos, asignarles tus propias funciones e integrar tu código a otras herramientas de desarrollo, con el fin de hacer sistemas de software más avanzados. Por ejemplo: páginas webs, aplicaciones móviles o video juegos.

Hay una infinidad de lenguajes de programación enfocados a diferentes áreas de desarrollo y no hay límites para lo que puedes construir. Puedes descargar e instalar un lenguaje de programación totalmente gratis buscando en internet. Sólo necesitas visitar su sitio web y encontrarás una guía detallada explicándote todo lo que necesitas saber para empezar a usarlos.

Te recomiendo aprender Python, ya que es uno de los lenguajes de programación más usados gracias a su alto nivel de desarrollo y sus múltiples utilidades.

¡No pierdas el tiempo!

Aprender a programar, te puede cambiar la vida.

Capítulo 3

Desarrollo de Software

El desarrollo de software es una práctica que se enfoca en la creación de *sistemas informáticos*. Estos sistemas te permiten almacenar, procesar y visualizar información a través de una computadora.

Por lo general, una persona sin experiencia en programación piensa que el desarrollo de software es una ciencia compleja y difícil de ejecutar, pero esto se aleja mucho de la realidad. Hay una serie de pasos y herramientas básicas que te permiten crear sistemas informáticos de manera sencilla.

El primer paso del desarrollo de software es pensar. Es importante definir el propósito y la utilidad del sistema que quieres desarrollar. Aquí es cuando comienzas a utilizar la *lógica de programación*, la cual te permite crear una estructura de pasos lógicos con el fin de cumplir el propósito de tu sistema.

Digamos que quieres construir un sistema que calcule el índice de masa corporal de tus usuarios. ¡Ya tienes un objetivo!, ahora necesitas estructurar una serie de eventos que permitan que ese objetivo se cumpla:

- tu sistema inicia,
- solicitas el peso de tu usuario,
- lo almacenas,
- solicitas la estatura de tu usuario,
- la almacenas,
- divides el peso entre la estatura,
- almacenas el resultado,
- le presentas el resultado al usuario,
- tu sistema termina.

Una vez que tienes definido tu objetivo y los pasos a seguir, solo queda decirle a la computadora que siga esos pasos y habrás creado un sistema informático funcional. El problema radica en que las máquinas solo entienden unos y ceros, pero a lo largo de la historia la tecnología ha evolucionado para facilitarnos esta comunicación por medio de los lenguajes de programación.

El segundo paso es programar/escribir código. Un *lenguaje de programación* es una herramienta diseñada para comunicarse con una máquina. Hay múltiples lenguajes de programación enfocados al desarrollo de diferentes tecnologías, pero todos tienen una base muy similar.

El lenguaje que utilices va a depender de lo que quieras desarrollar. Puedes descargar, instalar y manejar un lenguaje de programación a través de la *terminal*. La terminal, también conocida como *consola*, es una aplicación que te permite controlar el sistema de una computadora por medio de comandos. Al instalar un lenguaje de programación, vas a poder usar sus comandos a través de la terminal como un *compilador*. El compilador “traduce” los comandos y el código del lenguaje a instrucciones que la computadora entiende y ejecuta.

El *código del lenguaje* te permite organizarlo para poder controlar el funcionamiento del sistema de una computadora. Una vez que tienes una estructura lógica de código, vas a poder ejecutarlo usando los comandos a través de la terminal.

El código está compuesto de datos y herramientas que funcionan para crear la estructura de los pasos que tu sistema debe seguir. Estos datos y herramientas pueden ser: textos, números, booleanos (verdadero - falso) y listas de datos; variables para almacenar datos ($x = \text{dato}$); funciones que te permiten ejecutar acciones sobre los datos; condicionales lógicas que crean estructuras de flujo (if else) y muchos más elementos que dependerán del lenguaje que utilices.

Ejemplo (código escrito en Ruby):

```
• Tu sistema inicia,
puts 'Medidor de Índice de Masa Corporal'
• solicitas el peso de tu usuario,
puts 'Ingresa tu altura mts'
• lo almacenas,
altura = gets.chomp
• solicitas la estatura de tu usuario,
puts 'Ingresa tu peso kg'
• lo almacenas,
peso = gets.chomp
• divides peso entre estatura,
• almacenas el resultado,
imc = peso.to_i / altura.to_i
• le muestras el resultado al usuario,
puts "Índice de masa Corporal #{imc} imc"
• tu sistema termina.
```

El sistema inicia comunicándole a la computadora que estás corriendo el programa e imprime el mensaje 'Medidor de Índice de Masa Corporal' para indicarte que lo estás usando. Después, manda otro mensaje solicitando la altura del usuario, lo almacena en una variable y hace lo mismo con el peso. Seguidamente, el sistema corre una función que divide los valores y te presenta

una respuesta. Como ves, la lógica del código sigue todos los eventos de forma que permite el cumplimiento del objetivo de tu sistema.

El propósito y la funcionalidad del sistema puede crecer en base a la lógica que tenga el código. En este caso, podrías agregar código condicional que le presente al usuario recomendaciones dependiendo del resultado del IMC.

```
if imc < 50
```

```
    print "Toma agua"
```

```
else
```

```
    print "Haz ejercicio"
```

Los lenguajes de programación son herramientas poderosas que te permiten expandir tu código de muchas maneras. Generalmente, la lógica del código de lenguaje te permite crear tus propios tipos de datos, desarrollar funciones para esos datos e integrarlos a otras herramientas con el fin de expandir su funcionalidad. Para escribir código, puedes usar un *editor de texto*.

Un editor es una aplicación que te permite trabajar con código de manera sencilla. Puedes instalar librerías, estructurar archivos, ordenar tu código y utilizar muchas otras herramientas que van a depender del editor que uses. El tercer paso del desarrollo de software es expandir tu código. La belleza del desarrollo de software está en que se parece a armar un Lego. Siempre vas a poder agregar piezas para hacerlo más grande y funcional. La tecnología avanza con los años, facilitando el desarrollo de software. Si quisieras crear un sitio web basado en tu "Medidor de Índice de Masa Corporal", podrías usar un framework de desarrollo web que te presente una base para desarrollar tu idea. Así mismo, puedes expandirte al establecer una base datos para guardar registros, crear un API de los datos y vincularlo a una aplicación móvil. En fin, tú mismo eliges el límite.

El desarrollo de software te da la libertad y el poder de desarrollar tus ideas. Ten en cuenta que la programación es uno de los trabajos mejor pagados y aprender es gratis. Todos los lenguajes de programación tienen documentos y guías que te explican todo lo que necesitas para empezar a usarlos. Además, en internet hay muchos videos y tutoriales que enseñan a desarrollar múltiples ideas usando diferentes lenguajes. Solo tienes que querer hacerlo.

Cambia tu vida, aprende a programar.

Capítulo 4

Lenguajes de programación

Los lenguajes de programación son lenguajes diseñados para hablar con máquinas. Estos te permiten estructurar los pasos e instrucciones que debe seguir una computadora/máquina para cumplir una función. Por ejemplo: transmite un mensaje (puts “hello world”) o armar un sistema mas complejo como twitter.

Los lenguajes de programación y los sistemas funcionan procesando datos. Estos datos se pueden interpretar como información por ejemplo el nombre de un usuario, el correo electrónico, edad etc. Estos tipos de datos son los mismo que usamos para comunicarnos como humanos :

- textos (“Titulo: El principito”),
- números(420),
- booleanos: (verdad - falso).
- listas (agrupaciones de estos datos - [“El principito ”, 420, TRUE]).

Los lenguajes de programación incluyen una serie de funciones que permiten estructurar las instrucciones dependiendo de la utilidad que le quieras dar a estos datos.

- Puedes imprimir/transmitir mensajes = (puts “hello world”),
- comparar información = (420 < 69 (falso)),
- resolver ecuaciones de matemática = (420**2/69 (2556.52)),
- almacenar datos en variables = (nombre = “Leo”),
- contar elementos ([“leo”,420,true].size => #3)

y muchos otras funciones que van a depender del tipo de dato y lenguaje que uses.

Todos los lenguajes de programación comparten esa base, una vez la domines ya solo es estructurar los tipos de datos y sus funciones para cumplir el objetivo que tengas en mente. Esta base te permite crear flujo en el procesamiento de los datos en base a las condiciones que setees como por ejemplo si password es igual a “cualquiervaina” imprime “Bienvenido”.

Empieza a programar es fácil.

Capitulo 5

Terminal

El terminal es una aplicación que ya esta instalada en tu computadora. El terminal también conocido como la consola te permite controlar los datos y el funcionamiento de tu computadora de manera directa usando comandos. Estos comandos te permiten p moverte por todo tus archivos, editarlos, agregarlos y trabajar directamente con tu sistema.

A la hora de programar, el terminal es una de las herramientas mas importantes. Desde el momento que quieres instalar un lenguaje de programación, lo instalas por el terminal. A la hora de ejecutar tu código el terminal funciona como un compilador que le traduce tu código a la computadora. Hay múltiples aplicaciones que instalan en el terminal que te facilitan el proceso de programar.

Dependiendo del lenguaje que uses encontraras herramientas de consola que te permiten instalar librerias y funciones a tu codigo de forma directa. De la misma forma encotraras muchas mas herramientas de desarrollo y herramientas para el funcionamiento de tu terminal.

El terminal tiene multiples funciones y utilidades, puedes trabajar bases de datos, lenguajes de programación, herramientas de sistema, herramientas de internet y mucho mas.

El icono del terminal es mas o menos asi :



Buscalo, abrelo y juega un rato. escribe el comando `ifconf` y aprieta enter. Te va a regresar informacion sobre tu sistema corre un parde comandos y aprende. Aqui podras encontrar una lista de comandos basicos : [Lista de comandos](#) o escribenos y te explicamos mas cuando quieras.

Capítulo 6

Como empezar a programar

Para empezar a programar lo único que debes hacer es empezar. Todas las herramientas y información que necesitas ya están en tu computadora, en internet y no tienen costo. Lo más práctico para empezar a programar es buscar un tutorial o un curso donde una persona con experiencia te explique los procesos de instalación y los fundamentos de programar. Pero si tu eres un serio y quieres darle solo de tus 2 también puedes.

El primer paso es entender que es la programación y como funciona. Eso lo logras pensando, siéntate a analizar que significa la palabra programación y guíate por la lógica. Puedes buscar un par de artículos sobre la lógica de programación si quieres ver un lado más técnico del asunto pero es un tema 100% lógico.

Una vez entiendas eso, ya puedes pasar a tu máquina a escribir código. Para escribir código necesitas instalar y aprender a usar un par de herramientas. Estas son, un lenguaje de programación, el terminal y un editor de texto.

El lenguaje de programación que uses va a depender de ti y de lo que quieras hacer. Hay demasiadas opciones te recomiendo investigar porque al final es un tema de gustos y utilidad. Si no quieres investigar, python es una buena opción para empezar. Cuando ya sepas en que lenguaje quieres programar solo tienes que instalarlo. El proceso de instalación es super sencillo, todos los lenguajes de programación tienen una vaina llamada documentación.

La documentación es un “manual” que te explica el proceso de instalación y la base de como usar el lenguaje. La manera más sencilla y eficiente de instalar un lenguaje es a través del terminal con comandos de instalación. Todos esta información la encontrar en la documentación o la puedes buscar en google y encontraras información más detallada.

Una vez tengas el lenguaje instalado vas a poder correr comandos únicos del lenguaje, como un interpretador que es una herramienta que te permite escribir código directamente a tu máquina usando el terminal. Cuando ya tengas eso ready puedes empezar a escribir código y guardar tu código. Para eso necesitas un editor de texto.

Un editor de texto es una aplicación diseñada para programar, esta tiene múltiples funciones que facilitan el proceso de escribir código de manera

correcta. También te permiten guardar tus archivos de forma que una máquina lo pueda interpretar como un archivo ejecutable en ese lenguaje.

Ya con eso puedes empezar a programar, con el lenguaje de programación que escojas vas a poder construir múltiples instrucciones que tu computadora va a ejecutar para cumplir una función. Sé creativo, investiga y escribe código es la única forma de aprender.

Capítulo 7

POO

La programación orientada a objetos es una metodología de programación que te permite organizar de forma completa el funcionamiento de tus proyectos.

Para entender como funciona la programación orientada a objetos tienes que pensar de forma literal. Todo lo que te rodea es un objeto y todos esos objetos tienen atributos de data que los conforman. La programación orientada a objetos funciona creando moldes de datos que representan un objeto dentro tu programa.

A la hora de programar crear estos moldes te facilitan la estructura y el funcionamiento de tu app. Digamos que quieres hacer una aplicación como twitter. Tu idea es que las personas/usuarios puedan publicar mensajes para otros usuarios. Tu primer objeto es un usuario, ahí piensas que necesito de mis usuarios. Un correo para contactarlos un password para validarlos nombre, edad y país para identificarlos . Eso es un objeto.

```
user = {  
  id = 69  
  email = "sweetbubaloo@hotmail.es"  
  password = "mrkio123"  
  nombre = "Leo"  
  edad = 23  
  país = "PA"  
}
```

En base a eso empieza a meterle mente a tu back-end y front-end. Sabes que tienes que crear una tabla en tu base de datos donde guardaras usuarios con esos atributos, desarrollas las tu CRUD donde creas, vez, editas y borras tu objetos. Con el back-end ready desarrollas tus front-end tus visuales donde los usuarios ingresen sus datos y puedan interactuar con las funciones del CRUD. Tu formulario serio, el html pritty para ver la info del usuario, botones de acciones, etc.

Con eso tu tas volando. Ya solo es crear más objetos y relacionarlos entre si. Tu creas tu objeto tweet, que tiene un usuario, un mensaje y una hora de creación. Ahí es cuando le metes al código y le dices a tu base de datos que un tweet le pertenece a un usuario, creas tu relación, desarrollas el CRUD, le metes al front end y boom vas armando.

```
tweet = {
```

```
id = 420
user = 69
mensaje = "MRK IO"
date = 4/20/20
}
```

Así vas creando múltiples objetos y creando las diversas funciones en base al objetivo que tengas. Ya es un tema de creativa y solución de problemas. La programación orientada objetos es una de las metodologías más prácticas y eficientes. Muchos lenguajes de programación trabajan de esta forma, investiga y empieza a programar.

Capítulo 8

Frameworks

El desarrollo de software es un ejercicio que depende de múltiples componentes. Por lo general, la creación de un proyecto involucra el uso de varias herramientas, como bases de datos, librerías de código y herramientas de visualización. Los lenguajes de programación permiten la integración de todas estas herramientas, pero es fácil causar desorden en la estructura del código y eso puede generar errores en el desarrollo de tu proyecto. Para evitar este problema, se crearon los frameworks.

Los frameworks son estructuras genéricas que incluyen todas las herramientas necesarias para crear la base de un proyecto. Los frameworks dependen del tipo de proyecto que quieras desarrollar. Hay frameworks para desarrollo web, desarrollo de aplicaciones móviles, desarrollo de aplicaciones de escritorio, desarrollo de videojuegos y muchos otros proyectos.

Por lo general, los frameworks están creados sobre un lenguaje de programación específico y su uso depende de ese lenguaje. El uso de un framework te ahorra trabajo, te permite expandir el software de manera ordenada y facilita el desarrollo. Los frameworks se pueden instalar y ejecutar por la terminal. Busca en sus sitios oficiales, sigue la guía de instalación y aprende a usarlos leyendo la documentación.

Si conoces un lenguaje de programación, busca un framework relacionado con él. Encontrarás diferentes opciones basándose en la complejidad y el tamaño del proyecto que quieras desarrollar.

Capítulo 9

Debugging

A la hora de programar es común encontrarse con errores, el proceso de corregir estos errores se conoce como debugging. Si te sientas a escribir código es normal que en algo falles, lo primero que tienes que hacer para corregirlo es chillearte. Ponerte bravo o decepcionarte no sirve para verga tienes que frentear tus errores.

Hay diferentes tipos de bugs o errores, hay errores de sintaxis donde te comiste un “;”, hay errores de funcionamiento donde estas usando librerías que no tienes instaladas o errores lógicos donde tu código es una locura. Por lo general cuando ejecutas tu código y tiene un error tu consola o host te imprime un mensaje que dice “error en la línea 12” o alguna vaina así lee el mensaje y fíjate que estas haciendo mal.

Si aun así no puedes resolver o te detalla que es el error agarra el mensaje que te imprima y busca en google Todos los error tienen solución , es mentira que te inventaste un nuevo error alguien ya paso ahí y busco ayuda en internet. Confía en stack overflow.

Si nada de eso te funciona y no encuentras una solución. Echa para ataras y piensa. Fíjate que funcionaba y acuérdate que querías hacer. Ve línea por línea siguiendo el flujo de trabajo y piensa de forma lógica. Muchas veces no cerramos y la solución es más fácil de lo que pensamos.

Hay múltiples herramientas que te ayudan a debuguear y te facilitan el desarrollo. Investiga cuales funcionan con el editor y el lenguaje que uses y programa de manera eficiente. Una buena forma de desarrollar es no escribir mucho código, divide tu proyecto en objetivos y ármalo paso a paso. De esta forma siempre vas a estar claro de lo que estas haciendo, donde estas y adónde quieres ir.

Capitulo 10

FrontEnd vs BackEnd

El desarrollo de software se basa en el procesamiento de datos.

Cuando empiezas a programar por lo general escribes una variable, le asignas un dato, haces alguna operación con ese dato, lo imprimes y listo, código.

```
user="leo" //dato
```

```
userid = 420 // dato
```

```
username = user + userid //operación
```

```
puts username //código
```

```
#leo420
```

Esa es la base de la programación y se conoce como back-end. Cuando procesas datos, desarrollas el funcionamiento de tu programa. Estos datos los conectas a una base de datos, los solicitas cuando los necesitas, los relacionas a otros, estudias como se mueven, sumas restas multiplicas acciones y todo lo que se te ocurra. Haces algo con la data, eso es el back end.

Por lo general son acciones directas a tu base de datos y funciones ligadas a el funcionamiento de tu app. Involucra mucho tu lenguaje de programación y la lógica de tu algoritmo/programa.

El front-end es toda la parte visual de tu sistema, por lo general tu código crudo corre en consola. La máquina lo lee de forma directa, lo procesa y te da respuesta por ahí mismo. El front end se basa en como presentas ese sistema, puedes armar una aplicación web o una aplicación móvil, una aplicación de consola, un app de escritorio, lo que te de la gana. Pero el front-end es como presentas tu app en lo visual hacia el usuario.

Por lo general cuando uno empieza a aprender front-end ve mucho html, css, js, bootstrap.

Estas son herramientas de front-end que te permiten desarrollar visuales para un app web y algún otros tipos de aplicación dependiendo de la herramienta que uses.

Si en tu backend desarrollas el registro un nuevo usuario y le asigne un id. La lógica de tu front-end es crear un form donde el usuario ingrese su user y el botón que conecte la data con el código del back-end.

```
<form action="tu-backend">
  <label for="fname">User:</label><br>
  <input type="text" id="user" name="user" value="Data"><br>
  <input type="submit" value="Submit">
</form>
<h1>leo420</h1>
```

El front end de tu proyecto dependerá de lo que quieras desarrollar, puedes usar templates, librerías de componentes o desarrollar todo a pulmón. Hay muchas opciones y en lo personal depende 100% de tu creatividad.

Cuando dominas un lenguaje de programación, aprendes a conectarlo a una base de datos y aprendes a crear visuales para presentarlo y facilitar su utilidad, te conviertes en un full-stack developer. Para estos puedes dominar frameworks de diferentes aplicativos o dominar un lenguaje específico al 100%.