



Aptugo

Inicio en Programación Web

Guía de estudio

Index

Introducción al curso	3
¡Bienvenidos!	3
¿Qué es Aptugo?	4
¿No código? Entonces... ¿Qué es la programación?	5
¿Qué es un proyecto de software?	6
Etapas de un Proyecto de software	9
Discovery	9
Diseño	10
Coding	11
Testing	12
Deployment	13
Maintenance	13
Iniciemos en Aptugo	14
Pantalla de Bienvenida	14
¡Hello World!	18
Un recorrido por su interfaz	14
Ejecutando nuestro "HelloWorld"	23
Conclusión	27

Introducción al curso

¡Bienvenidos!

¡Hola! Te damos la bienvenida a este nuevo curso de Aptugo. Durante nuestro recorrido, te propondremos diferentes instancias donde aprenderás todo lo necesario para desarrollar un software funcional y eficiente, de principio a fin.

Comenzaremos con el proceso de identificación del problema o necesidad, bosquejo y diseño, seguidamente abordaremos la producción del mismo, cómo probarlo e ir mejorándolo, para finalizar en la publicación del proyecto a un servidor para su uso general. **Como habrás notado, en ningún momento mencionamos el código, ya que Aptugo se encargará de hacerlo por nosotros**, con diferentes elementos y funcionalidades.

Podrás diseñar y elaborar proyectos en poco tiempo, sin necesidad de escribir miles de líneas de código para conseguirlo.

Al finalizar el curso, **podrás pensar como un programador, tendrás todas las herramientas necesarias para desarrollar aplicaciones web**, y el código de las mismas se generará como si estuviera escrito por un programador tradicional. Con esto último, no solo podrás dar rienda suelta a tu creatividad, sino también contarás con una ventaja fundamental, sabrás diseñar proyectos por tí mismo, respondiendo a problemas puntuales, y con una ejecución definida.

Si decides aprender código o si ya sabes, Aptugo te dará muchísimas ventajas, por ejemplo: podrás acelerar tu curva de aprendizaje y construir desarrollos complejos, incorporando tu propio código mediante elementos integrados de la plataforma, observando cambios en tiempo real y sin necesidad de realizar tareas repetitivas.

Todo esto y mucho más lo irás descubriendo a lo largo de cada clase.



Al finalizar el curso, **podrás pensar como un programador, tendrás todas las herramientas necesarias para desarrollar software de todo tipo y para todo tipo de dispositivos**, y el código de tus aplicaciones tendrá el nivel y capacidad de un programador experto.

¿Qué es Aptugo?

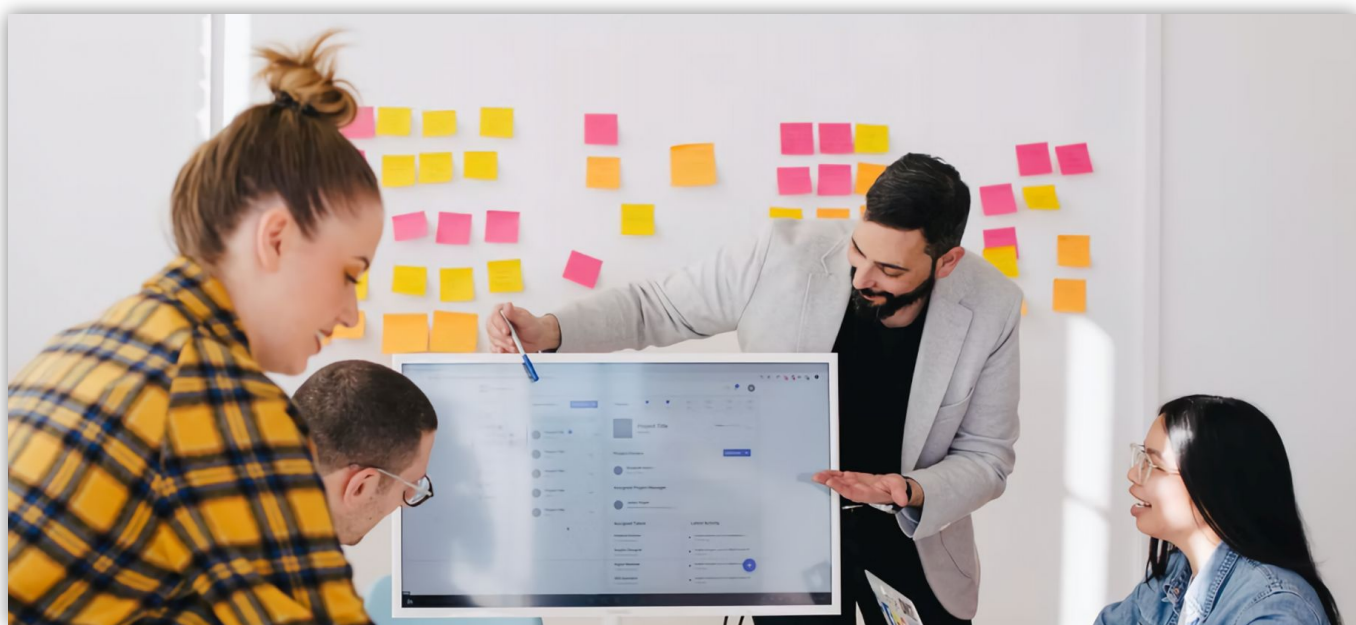
Se trata de **la primera herramienta de desarrollo que permite generar software (Programas) tanto de manera visual como tradicional**. Aptugo **nos permite desarrollar software de alta calidad con mayor rapidez, facilidad y eficacia que la programación tradicional**, sin necesidad de escribir miles de líneas de código.

A través de una interfaz visual, Aptugo nos permite "arrastrar y soltar" distintos elementos, los cuales consisten en "paquetes" de código aislado, los cuales pueden combinarse y generar estructuras escalables, que se convierten en aplicaciones completas. Aptugo se encarga de que todo funcione, generando de manera automática y en tiempo real, todo el código de nuestro desarrollo.

Aptugo sigue dos improntas principales:

- **Deep Coddling (Código en lo profundo):** A diferencia del Low Code (Poco Código), Aptugo genera todo el código sin necesidad de intervenirlo. Se pueden generar aplicaciones completas sin modificar el código fuente, o al contrario, complementarlo con diferentes elementos que habilitan el ingreso de líneas de código.
- **Atomic Programming (Programación Atómica):** este paradigma plantea que el código se compone de unidades indivisibles, estables e irreducibles. Así, cada una de esas partes se van combinando y complementando para generar estructuras más complejas.

La presencia de Aptugo dentro del entorno IT no sólo va a ayudar al programador convencional a **hacer su trabajo más sencillo, rápido y dinámico**, sino también contribuirá a **cualquier persona** que desee **insertarse dentro de la programación**, a desarrollar las nociones básicas y necesarias para comenzar a introducirse en este mundo.



¿No código? Entonces... ¿Qué es la programación?

Tal vez el mito más importante que Aptugo viene a derrumbar es el que establece que la parte más importante del desarrollo de software es escribir código. **La programación es mucho más que escribir instrucciones línea por línea durante semanas enteras**, necesita de ingenio y creatividad para generar soluciones a los diferentes problemas que enfrentamos día a día.

La programación es un proceso creativo.

"The art of computer programming" Donald Knuth, 1968 menciona que la **computación es un arte, porque se aplica el conocimiento acumulado de cada individuo al mundo**. Dicho conocimiento no consiste en la ejecución mecánica de diferentes acciones, como si de robots se tratara, sino que está intervenido por la imaginación e ingenio. Desde la funcionalidad hasta la estética, la programación piensa en cada detalle, generando un producto único e individual.

Escribir código es la parte específica del proceso de programación, donde podemos medir objetivamente aspectos como funcionalidad, optimización, eficiencia... y, a partir de las diferentes conclusiones, ir mejorando cada vez más el proyecto.



Si bien el código lleva más tiempo y es a lo que más se le suele dar protagonismo, está lejos de ser lo más importante de un proyecto de software.

Entonces, podemos decir que **la programación conjuga dos aspectos: es un proceso creativo** ya que se crean y planifican Apps, Tools, soluciones y features apoyado por el diseño, y un proceso de ingeniería, correspondiente a la ejecución específica del código.

Para realizar un proyecto eficiente y funcional, **se necesita que ambos aspectos se desarrollen de forma armónica**. Si bien el código lleva más tiempo y es a lo que más se le suele dar protagonismo, está lejos de ser lo más importante de un proyecto de software.

¿Qué es un proyecto de software?

Para definir qué es un proyecto de software, podemos comenzar con una definición sencilla de las palabras que lo componen:

Proyecto: Es la planificación de diferentes actividades orientadas a resolver un problema o satisfacer una necesidad.

Software: Es el conjunto de programas o instrucciones que permiten ejecutar diversas funciones dentro de un entorno tecnológico.

En conclusión, podemos decir que un **proyecto de software es la planificación de diferentes instrucciones o actividades que le indicaremos al medio tecnológico vigente (navegador, computadora, celular, etc) con el fin de resolver un problema o necesidad.**

Todo proyecto de software se puede dividir en etapas muy definidas:

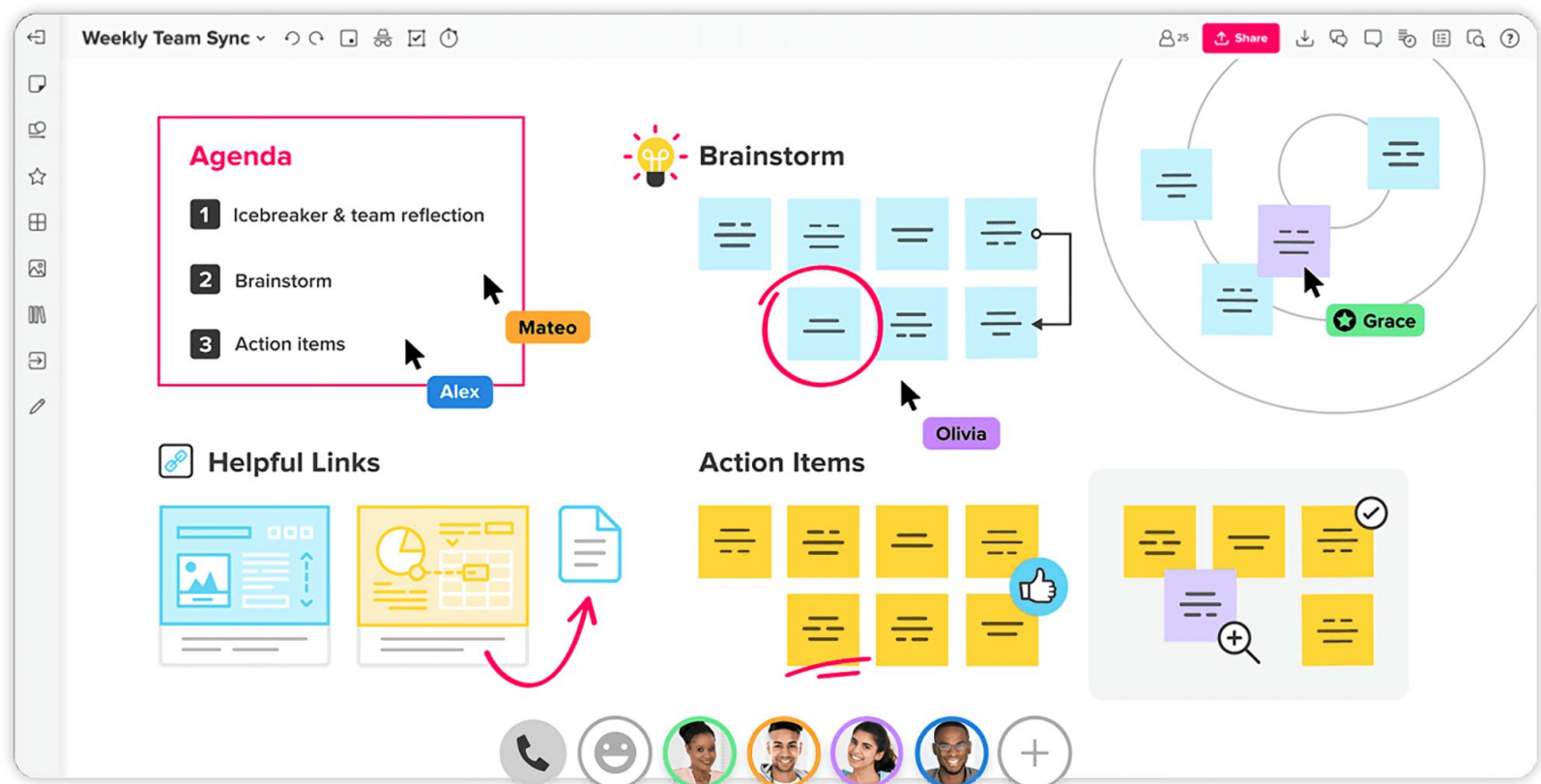
1. Discovery (Investigación)
2. Design (Diseño)
3. Coding (Codeo)
4. Testing - QA (Testeo)
5. Deployment (Lanzamiento)
6. Maintenance (Mantenimiento)

La mejor forma de visualizarlas es tomar un ejemplo práctico:

imaginemos un programador que, durante su vida profesional, ha desarrollado programas de diversa índole: ecología de peces para un laboratorio, un sistema de stock de un supermercado, y administración de un centro de salud ¿Será necesario ser un conocedor de todas estas áreas? Seguramente ayude, pero no es requisito.

Discovery (Investigación)

En esta etapa se definen los objetivos del software, las funciones que deberá cumplir, y todos aquellos detalles que podrían ser de utilidad para la elaboración del proyecto. Este paso puede hacer una gran diferencia en cuanto al tiempo de construcción, porque se vincula directamente a la funcionalidad del software.



Diseño del proyecto

El Discovery da paso a la segunda fase: ¿Cómo organizaremos nuestro proyecto? ¿Qué datos guardaremos? ¿Qué pantallas tendrá nuestro sistema? Estas preguntas se vinculan al **Diseño del proyecto**. Al finalizar este paso, ¡se obtiene un boceto del sistema completo! Y todo sin escribir una línea de código. Esto también es fundamental, ya que todo tipo de cambio involucra miles de líneas de código (lo cual, en programación tradicional, implicaría semanas de trabajo).

Este proceso nos permite visitar el anterior. ¿Nos faltó hacer discovery? ¿Cómo llenamos los huecos? Estos dos pasos iniciales nos permiten también pensar a futuro, anticipándonos a funcionalidades que se podrían necesitar más adelante.

Coding (Código)

El tercer paso es el coding. Sin Aptugo, este proceso normalmente lleva varios meses e involucra a varios programadores. **Con ayuda de Aptugo, podemos reducirlo considerablemente.**

Testing (Testeo)

La siguiente parada, una vez terminado el coding, es **probar el sistema**. Debemos ajustar las pequeñas diferencias entre las expectativas y lo que sucede. Si hicimos nuestro discovery y diseño bien, entonces los ajustes sólo serán en el código, algún que otro error en las más de 60 mil líneas de código (en programación tradicional).

Deployment (Lanzamiento)

Una vez probado, es momento de subirlo a su destino final. Existen numerosos proveedores (Servers) que nos permiten alojar nuestro proyecto. Algunos muy simples y generales, otros mucho más específicos para los requisitos de nuestro software.

Mantenimiento

Pero el proyecto no termina ahí. Ahora viene el mantenimiento: **Ir aplicando cambios a nuestro sistema a pedido del cliente**. Cambian las reglas, cambia el software. El mantenimiento es muy importante para garantizar el correcto funcionamiento de la aplicación y todo lo que dependa de ella (datos, logística, turnos...)



¡No cometas el error de pensar que una etapa importa más que otra!

En definitiva, estas son las 6 etapas básicas que podemos encontrar en un proyecto de software. Si quieres ser un desarrollador exitoso, o desarrollar exitosamente un proyecto, ¡No cometas el error de pensar que una etapa importa más que otra! En las próximas páginas veremos cada una de estas etapas en detalle.

Etapas de un Proyecto de software

En este apartado ampliaremos los conceptos introductorios de las etapas de un proyecto de software **¡y entraremos en detalles más apasionantes y útiles!**

Discovery (Investigación)

El objetivo del discovery es justamente: **Tener toda la información necesaria para pasar a diseñar nuestra aplicación.**

Con el tiempo y la experiencia, podremos ir asumiendo partes del diseño. Por ejemplo:

1 - ¿Nuestra aplicación tendrá una página principal y abierta a todo el mundo?

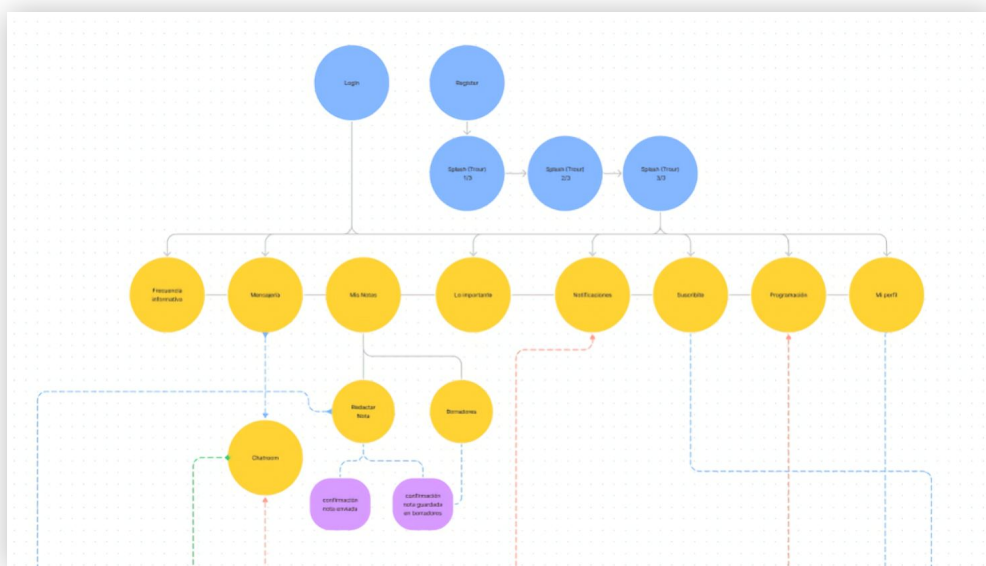
→ Esta pregunta nos da la pauta de si necesitamos involucrar un diseño atractivo o si con sólo un diseño funcional alcanzará.

2 - ¿Los usuarios pueden crear sus propias cuentas?

→ Hay una página de registro.

3 - ¿Los usuarios tendrán diferentes roles?

→ Habrá administradores con acceso irrestricto, mientras que otros usuarios podrán acceder a secciones particulares.



Haz el ejercicio, imagina que te dicen "Quiero hacer un sistema en donde yo pueda entrar en contacto con mis amigos, y que cuando suba una foto, todos puedan verla", **esto es, seguramente, la primera frase que el equipo de programación de Instagram recibió al comenzar a hacer su discovery.** ¡Imagina ahora las preguntas que harías!



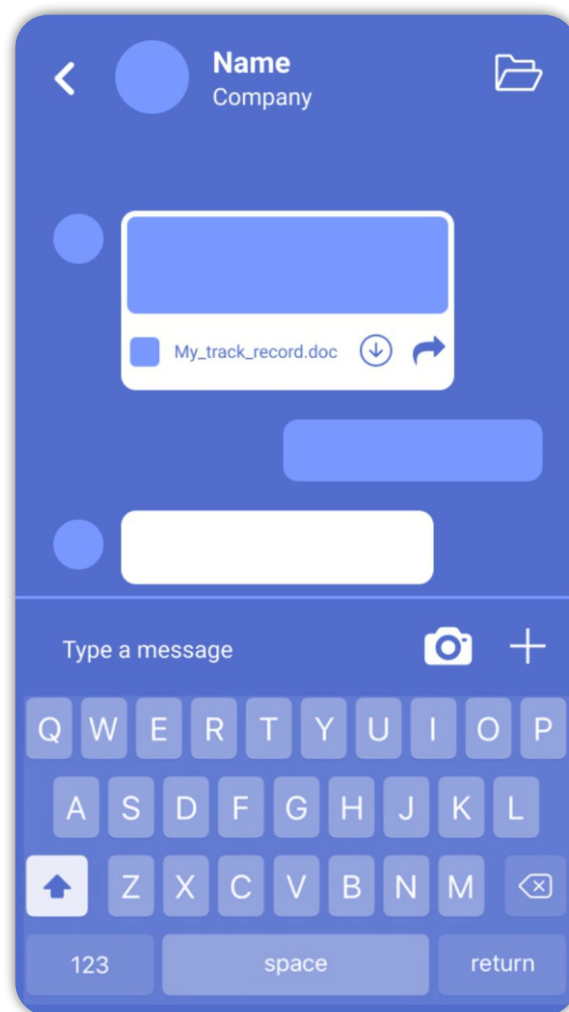
Recuerda tu objetivo: Poder hacer un diseño de tu sistema. En la Guía de Ejercicios de la Clase 1 te proponemos profundizar en esta etapa pensando en tu propio proyecto.

Diseño

Nuestro discovery nos dijo que "Desde la página principal tendremos que tener la posibilidad de subir una foto", y nuestro diseño lo refleja, entonces, dibujando un botón que nos lleva a la página de "subir foto".

Parece algo obvio, sin embargo, descubriremos la cantidad de trabajo involucrado en nuestro ejemplo: "instagram para fiambrerías" a través de las páginas que tiene, páginas que seguro no habías pensado cuando lo imaginaste: "Recuperar contraseña", "solicitud de amistad", "enviar mensaje".

Rápidamente nuestro sistema pasará de nuestras dos páginas ("página principal" y "subir foto") a un compendio de unas 50 páginas diferentes unidas entre sí.



Al tener nuestro diseño en papel, estamos posibilitados **de armar el esquema de nuestra base de datos**. Por ejemplo: Sabemos que tendremos un listado de fiambrerías, que tendremos muchas fotos por fiambrería, que una fiambrería tendrá un nombre seguramente, además de una dirección, tal vez la ciudad en la que se encuentra. ¿Nuestro sistema funciona en varios países? El país entonces.

Y así, luego de unas cuantas páginas, terminamos con el diseño de lo que va a ser nuestro sistema. Esto no es foráneo y sucede en todo proceso creativo. Una fábrica de automóviles necesita del diseño inicial para comenzar a armarlos, hasta el pintor menos estructurado sabe que tiene que hacer un planeamiento mínimo antes de comenzar a plasmar su idea en el lienzo. **Como desarrollador de aplicaciones, ¡cada**

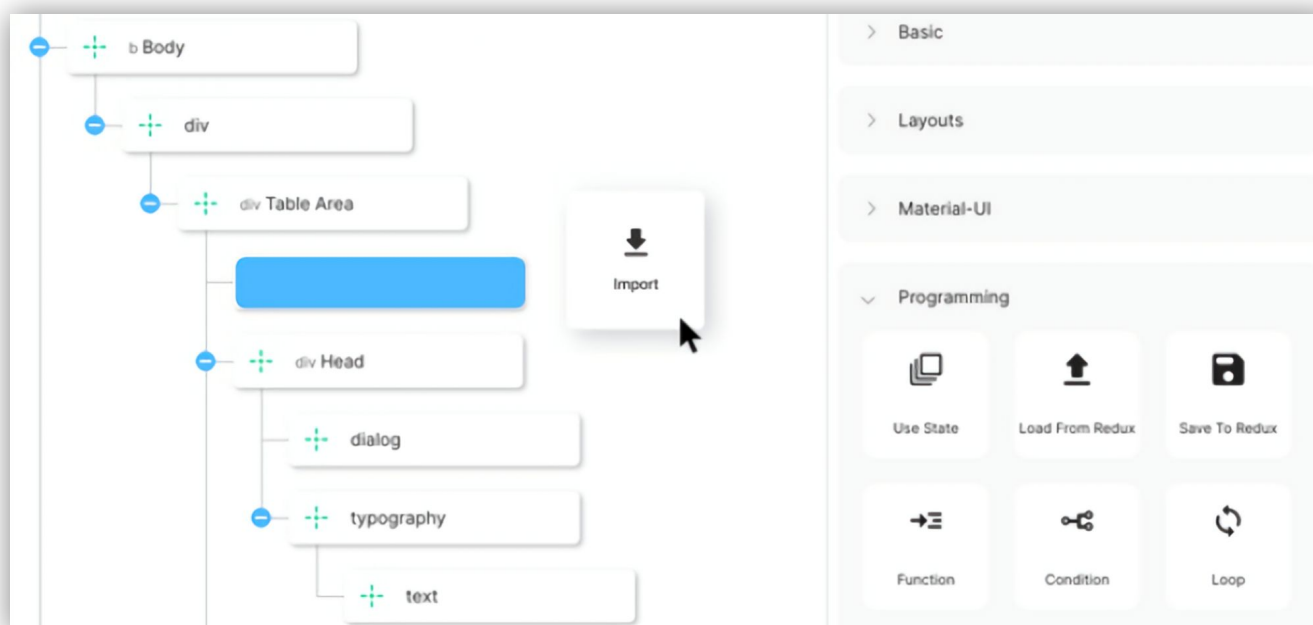
pantalla de tu sistema es un lienzo! Y cuando todas las pantallas tienen sentido, cuando no hay puntos que no llevan a ningún lugar, y cuando sabemos que nuestro sistema tiene la capacidad de realizar la tarea para la cual fue concebido podemos pasar al siguiente paso: El código.

Coding (Código)

El código es el punto en el que el 67% de los proyectos de software falla, y generalmente esto sucede porque no se ha hecho un buen trabajo en los primeros dos pasos. Nos apuramos, lo hicimos más complicado de lo que era necesario, organizamos mal los datos. Sea la razón que sea, **una vez que comenzamos la parte del código, cada error lo pagamos con muchas horas de nuestras vidas.** Así que el primer paso de "Coding" es: revisar nuevamente nuestro diseño y ver si nos faltó algo de discovery.

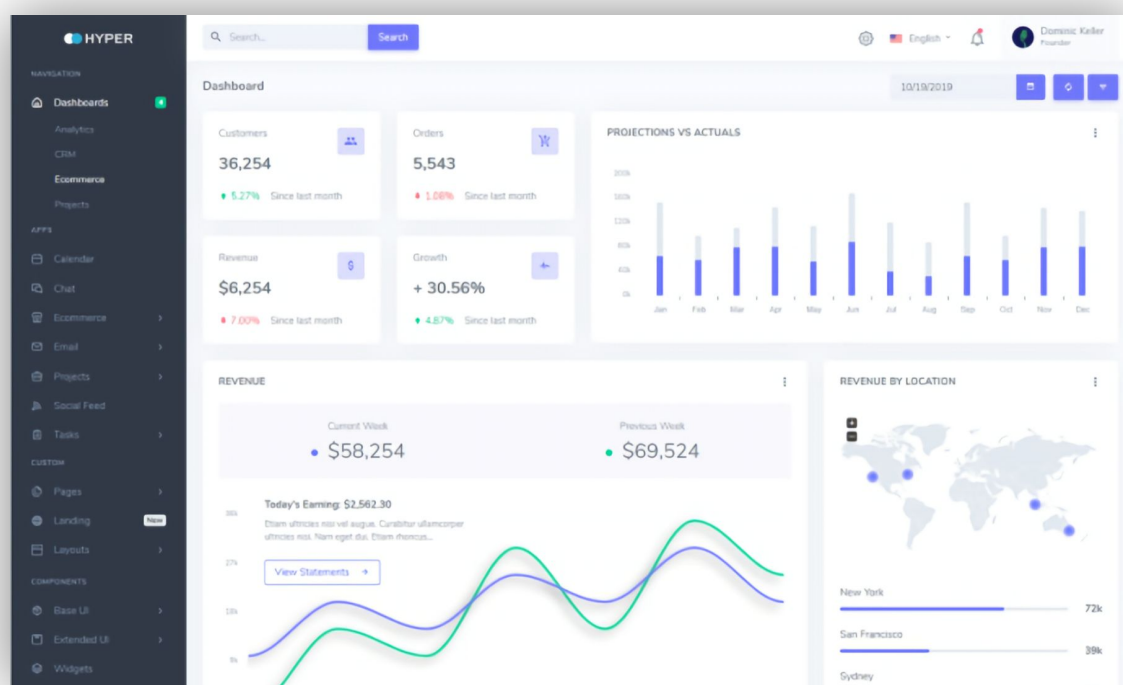
Si por el contrario, nuestro diseño en papel es sólido y sin fallas, con programación tradicional este paso puede llegar a ser el más aburrido, repetitivo y tedioso. **Con Aptugo, convertimos este proceso a uno visual, en donde el código es escrito por nosotros de la manera que nosotros pretendamos.**

Con solo entender la lógica de nuestro proyecto, **podremos ir combinando elementos de manera visual y Aptugo generará el código en tiempo real.** Código que podremos visualizar, editar, expandir y escalar, tanto dentro como fuera de Aptugo, ahorrando así, cientos de horas de trabajo, reduciendo la posibilidad de errores insalvables en nuestro sistema y permitiéndonos generar desarrollos mucho más complejos de los alcanzables con nuestro nivel de conocimiento actual.



Testing (Testeo)

La etapa de testing o testeo consiste en la verificación de que todo el software funciona de manera correcta y de la manera solicitada por el cliente, y se la conoce también como QA. Esta etapa es quizás la menos fascinante, pero es fundamental y se realiza antes de entregar al cliente su producto final. Permite identificar a tiempo errores o problemas en el software facilitando su resolución antes de la entrega, y garantizando calidad, seguridad, confiabilidad y rendimiento. Además, un buen QA nos ayudará también a identificar y evitar posibles errores a futuro.



Durante esta etapa **se realizan distintos tipos de pruebas que pueden ser funcionales** (verificar que cada función del software actúe de la manera esperada y en conjunto con el resto) **o no funcionales** (que evalúan otros aspectos como rendimiento, usabilidad y confiabilidad)

Llevado a la práctica, verificaremos por ejemplo que cuando un cliente incluye nueva información (como un pedido), este dato se cargue correctamente en la base de datos del software que hemos creado. Otros casos simples de testeo tienen que ver con las rutas que utilizamos para conectarnos a sistemas externos, como sistema de pagos, redes sociales, entre otros.

Deployment (Lanzamiento)

El deployment es la etapa de lanzamiento o "release" del producto. Se trata de ese momento tan ansiado del "pasaje a producción" para hacer disponible la versión de nuestro software a los consumidores. **Cada software tiene sus propias características y complejidades, por lo cual no existe una única definición o proceso paso a paso de esta etapa.** De este modo, el deployment abarca todo los requerimientos necesarios para que nuestro sistema de software o nuestra actualización pueda estar disponible para los usuarios. La mayoría de los sistemas de software hoy requieren de actualizaciones, correcciones y nuevas funcionalidades que se despliegan a través de una combinación de procesos manuales y automatizados.

La etapa de deployment puede tener distintas partes pero, **de modo sencillo implicaría colocar en posición, habilitar para uso o, simplemente, publicar nuestro software.**

Maintenance (Mantenimiento)

La etapa de Mantenimiento puede incluir desde actualizaciones programadas, o cambios adicionales hasta caídas de servicio o eventos inesperados entre otros.

Pensemos por ejemplo que nuestro software tiene un conector que va desde su página principal a una red social donde sus usuarios pueden dejarle "likes" y comentarios. Dicha red, un día, decide hacer una actualización que incorpora nuevas instrucciones para conectarse a la página. Esta es la etapa en la que deberemos actualizar y adaptar el software a cambios no anticipados para garantizar que la función siga ejecutándose con normalidad.

De la misma manera, podría ser que nuestro software está teniendo un éxito rotundo, completamente inesperado, y este alto tráfico hacia su sitio está haciendo que el sistema se trabe y ande tan lento que los potenciales clientes se van. ¿Se debe a un problema con la capacidad de nuestro servidor? **Todos estos aspectos son parte de la etapa final pero continua del desarrollo de un proyecto de software.**

Iniciemos en Aptugo

Pantalla de Bienvenida

Vamos abrir por primera vez **Aptugo**, nos pedirá **un correo electrónico y una contraseña** para validar nuestra licencia (Es recomendable activar el switch "Recordar contraseña" para guardar la contraseña), si es la primera vez que colocamos nuestro email, la contraseña puede ser la que nosotros queramos para crear nuestra cuenta:

The diagram illustrates the initial setup process in Aptugo. It starts with a 'Bienvenido a Aptugo!' screen where the user is prompted to 'Ingresa tu email para obtener una licencia'. A text input field labeled 'Dirección de email' is shown, followed by a blue button labeled 'DAME MI LICENCIA!'. A dashed orange arrow points from this button to a second screen titled 'Cambiar email'. This second screen shows the user is logged in as 'cristiannicolasgesualdo@gmail.com' and is prompted to 'Por favor, ingresa tu contraseña' with a masked password field. It also features a blue 'DAME MI LICENCIA!' button, a 'Recordar contraseña' checkbox, and a link for '¿Olvidó su contraseña?'.

Aptugo

Bienvenido a Aptugo!

Ingresa tu email para obtener una licencia

Dirección de email

DAME MI LICENCIA!

Cambiar email

Aptugo

Bienvenido a Aptugo cristiannicolasgesualdo@gmail.com!

Por favor, ingresa tu contraseña

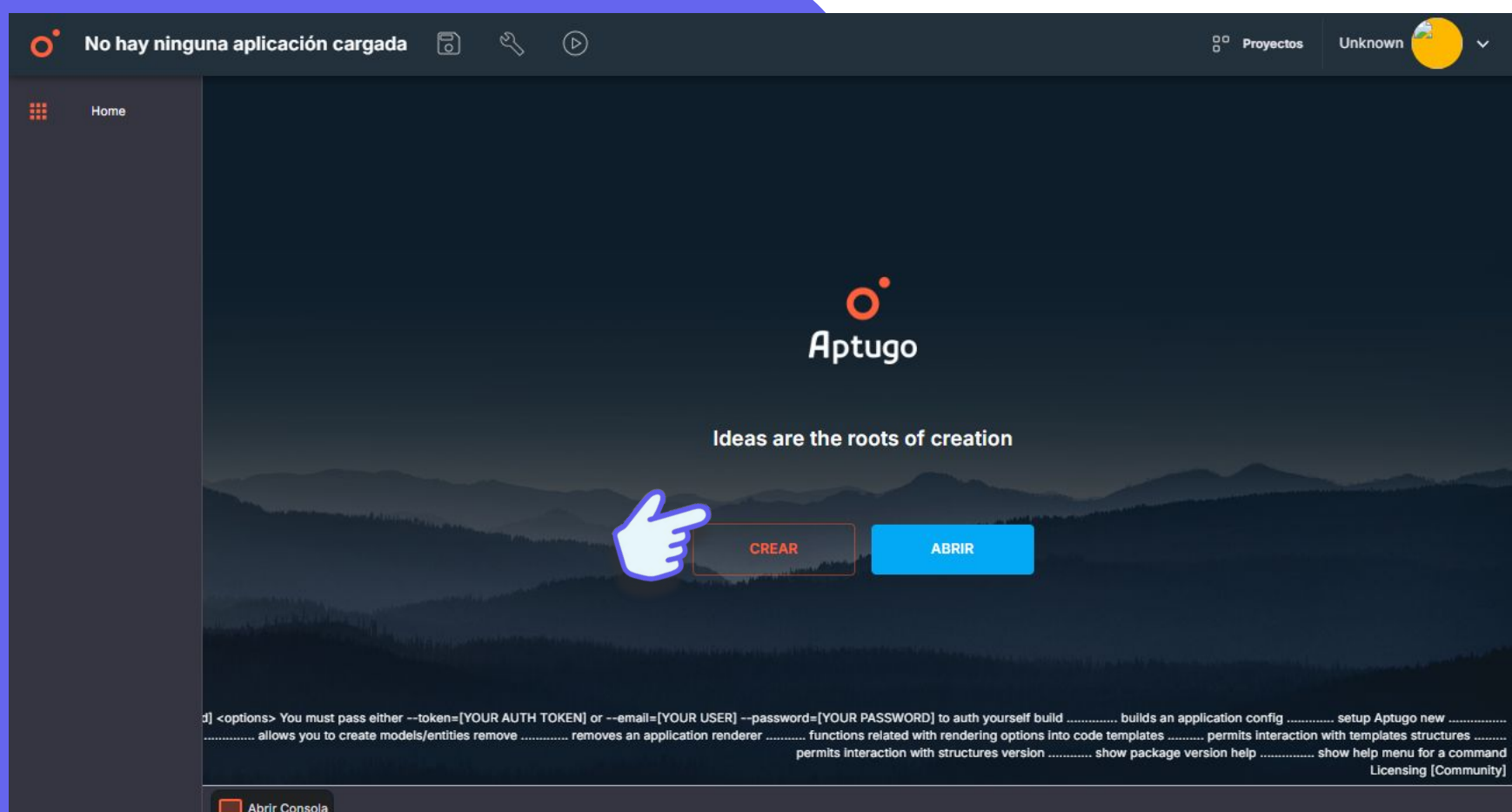
.....

DAME MI LICENCIA!

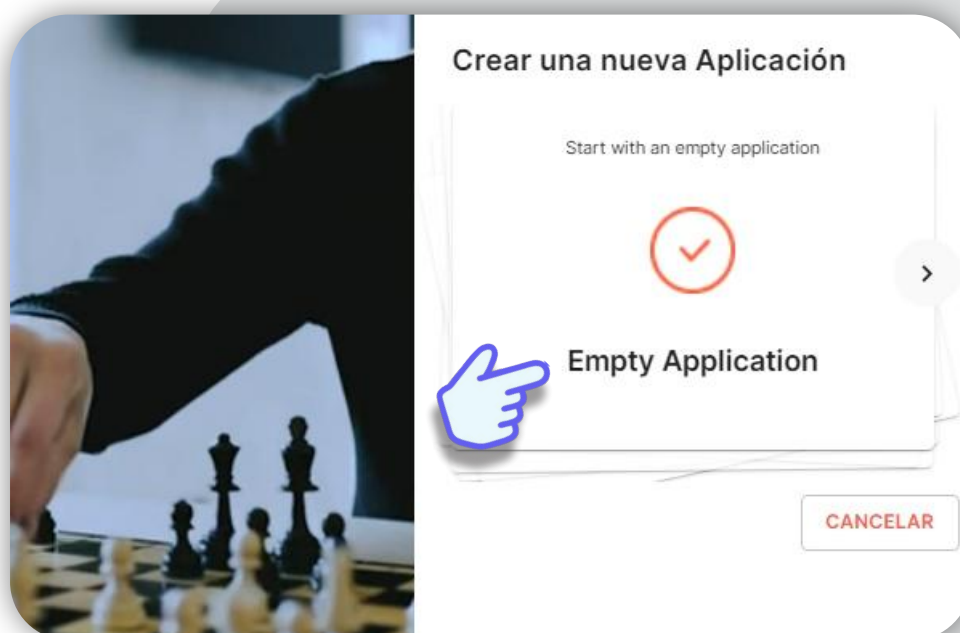
☐ Recordar contraseña [¿Olvidó su contraseña?](#)

¡Hello World!

Si es la primera vez que abrimos nuestro Aptugo, puede que no tengamos ninguna aplicación creada, si ya tenemos todo instalado como la guía de instalación indica, estamos en condiciones de generar nuestra primera aplicación web. Para comenzar, nos dirigimos a la pestaña **Home** y hacemos click en **"create"**



Posteriormente, nos aparecerá el siguiente recuadro y le daremos **click** donde dice **"Empty Application"** (Aplicación vacía).



En el siguiente recuadro, **se nos pedirá determinar un nombre a la aplicación.**



El mismo **no debe poseer espacios, caracteres especiales ni letras ñ.** Recordemos que todo software de programación está construido en idioma inglés, por lo cual no se reconocen caracteres pertenecientes exclusivamente al lenguaje hispano. **Por lo tanto, una alternativa es incorporar el nombre en inglés. Hemos elegido "HelloWorld" o "HolaMundo".**

Necesito algo de información:

Debes completar esta información antes avanzar..

New Application Name

HelloWorld

CANCELAR

EJECUTAR!

Una vez establecido,
hacemos click en
"EJECUTAR!"

Encontraremos en la parte superior izquierda de la pantalla, aledaño al logo de Aptugo, el **nombre de la aplicación que ingresamos.** A la derecha de dicho nombre, encontramos cuatro iconos.

Ahora, hacemos click en "Grabar o guardar los cambios" para guardar nuestro proyecto.

Grabar cambios



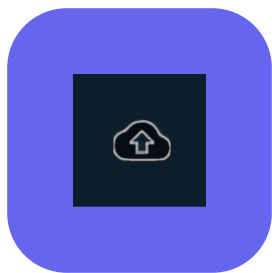
Los mismos responden respectivamente a:



Función Guardar (Save): Éste nos permite guardar los cambios realizados en la aplicación, sin embargo, no los compila.



Función Compilación Rápida (Quick Build): Este botón nos habilita a realizar una compilación con el seteo establecido en una anterior, o bien con los valores por defecto en caso de no haberse realizado ninguna anteriormente.



Función Compilación (Build): Con esta función podemos compilar y generar el código fuente de nuestra aplicación, aquello que se va a mostrar en el navegador. A diferencia del build anterior, **éste puede modificarse, habilitando tres compilaciones diferentes (Deployment/Local; Staging/pre-producción o testing; Production/versión web)**. Además, podemos establecer el nivel de detalle con el que se reflejarán en la consola los diferentes procesos que se llevan a cabo para compilar el código.

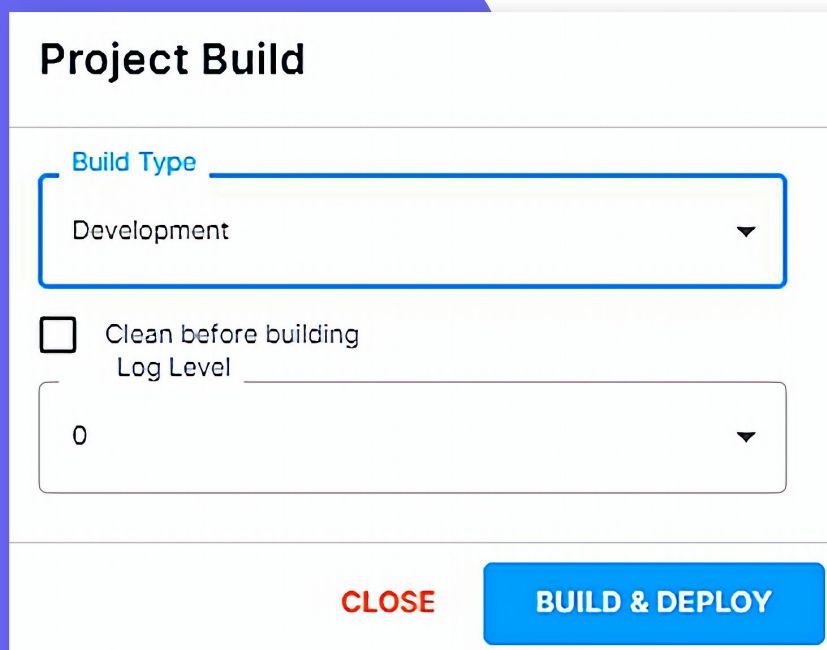


Función Ejecución (Control Program Execution): permite ejecutar nuestra aplicación y visualizarla en el navegador web predeterminado.

Realizada esta descripción de los botones, **el orden a seguir será:**

- 1- Hacer click en el primer botón (Save).
- 2- Luego en el tercer botón (Build).

A continuación, veremos la siguiente pantalla:



Project Build

Build Type Development

☐ Clean before building

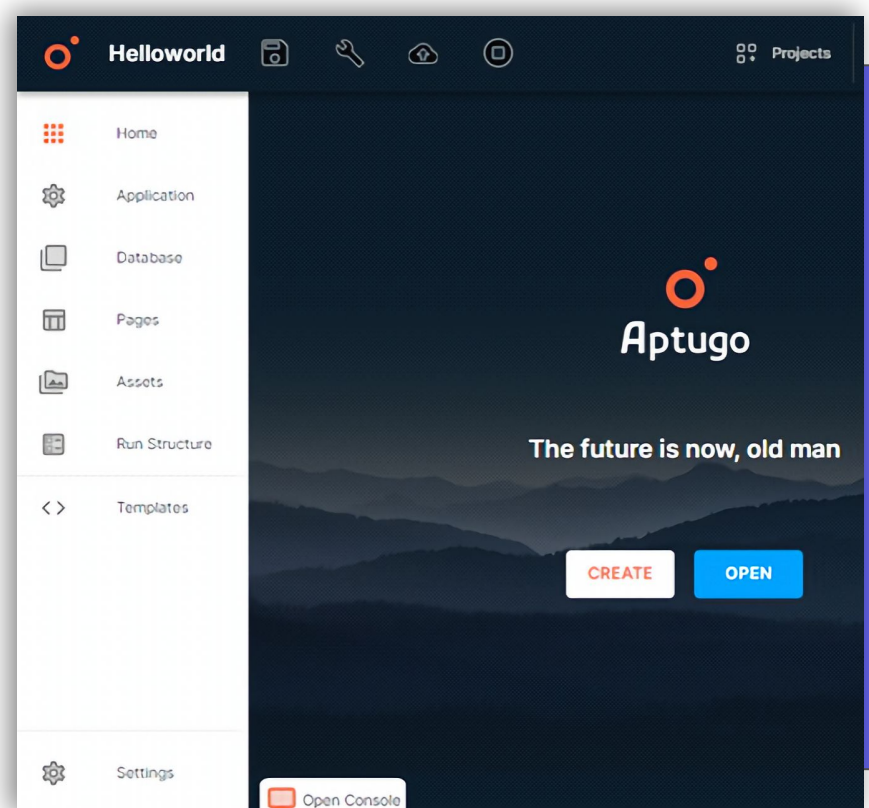
Log Level 0

CLOSE **BUILD & DEPLOY**

Por ahora, dejaremos el primer campo con la compilación por defecto. En el segundo, **recomendamos usar el Log Level en nivel 10, ya que éste es el máximo nivel de detalle**. Todo proceso, independientemente de su tipo, se verá reflejado en la consola, lo cual hará más sencilla la solución de errores en caso de haberlos.

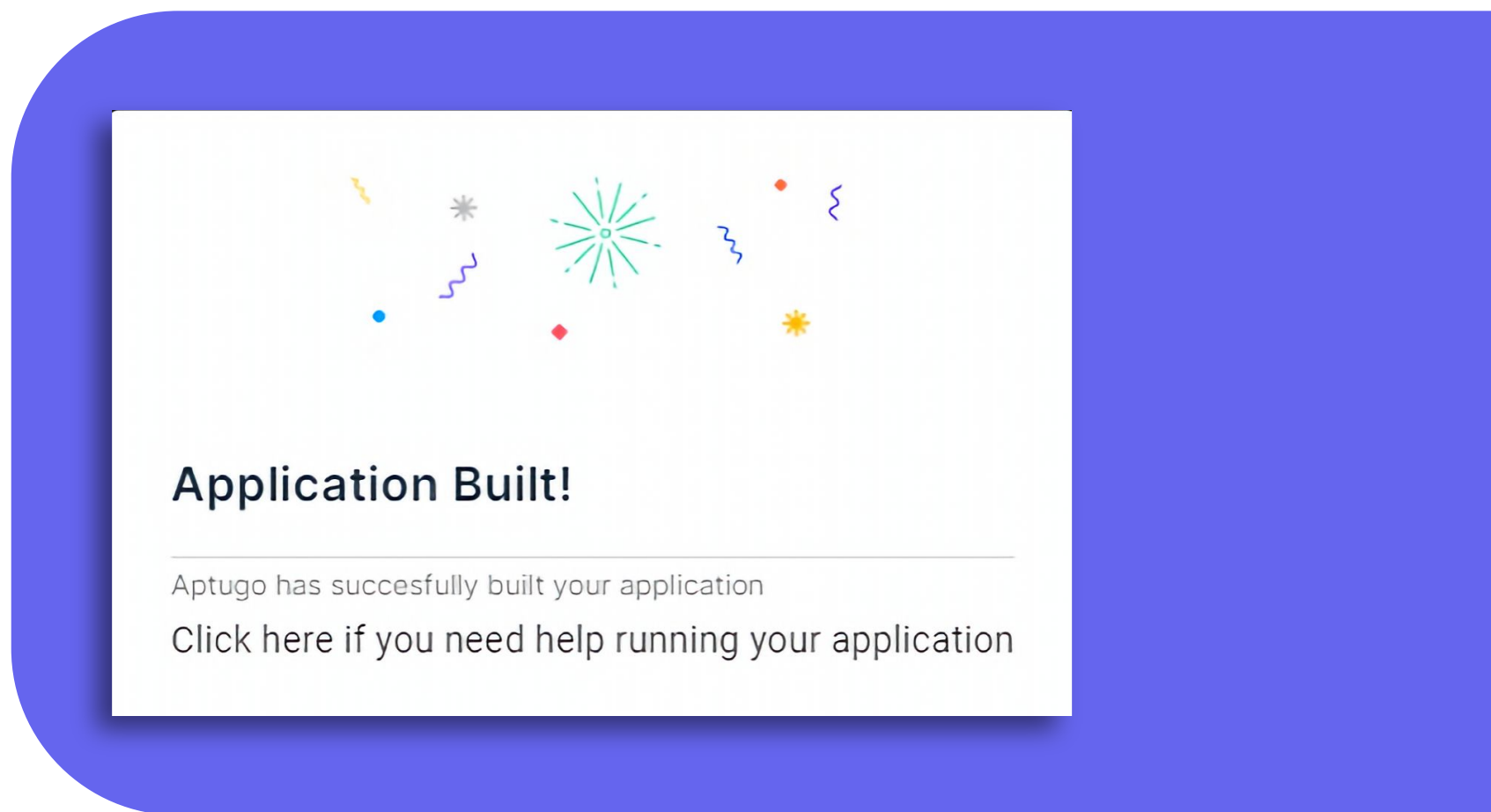
De tanto nombrarla, **podemos preguntarnos qué es esa famosa "consola"**. La misma **es una interfaz que refleja procesos en formato de órdenes escritas o "comandos"**. Tuvimos un primer acercamiento a ella al instalar PNPM, mediante el ingreso del comando adecuado, pero esta consola está especialmente diseñada para comunicar todo lo que sucede en Aptugo.

Para acceder a ella, debemos dirigirnos a la sección inferior izquierda, donde lo encontraremos con el nombre de **"Open Console"**.



Una vez finalizado el proceso de compilación, nuestra aplicación está lista para ser ejecutada.

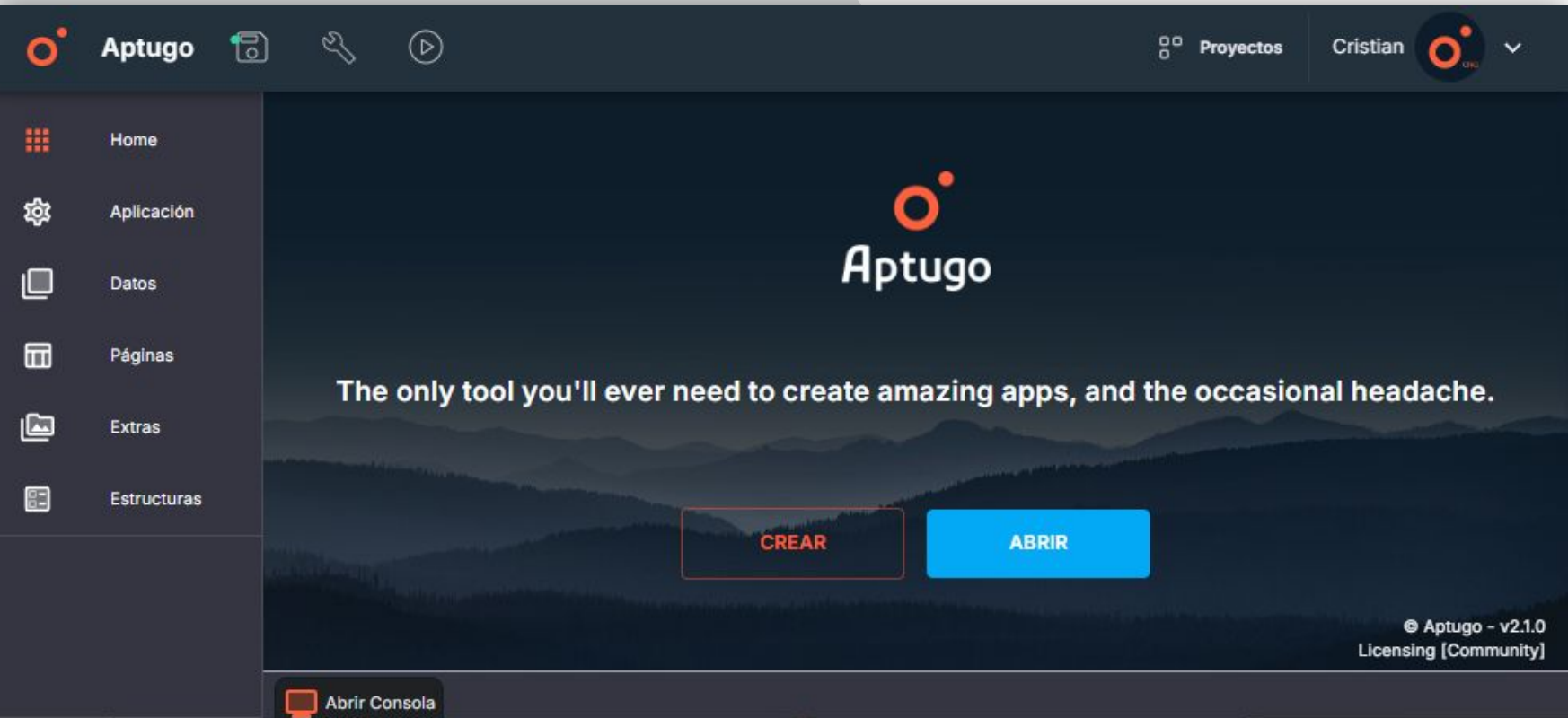
Sabremos que él mismo ha llegado a su fin cuando recibamos el siguiente mensaje:



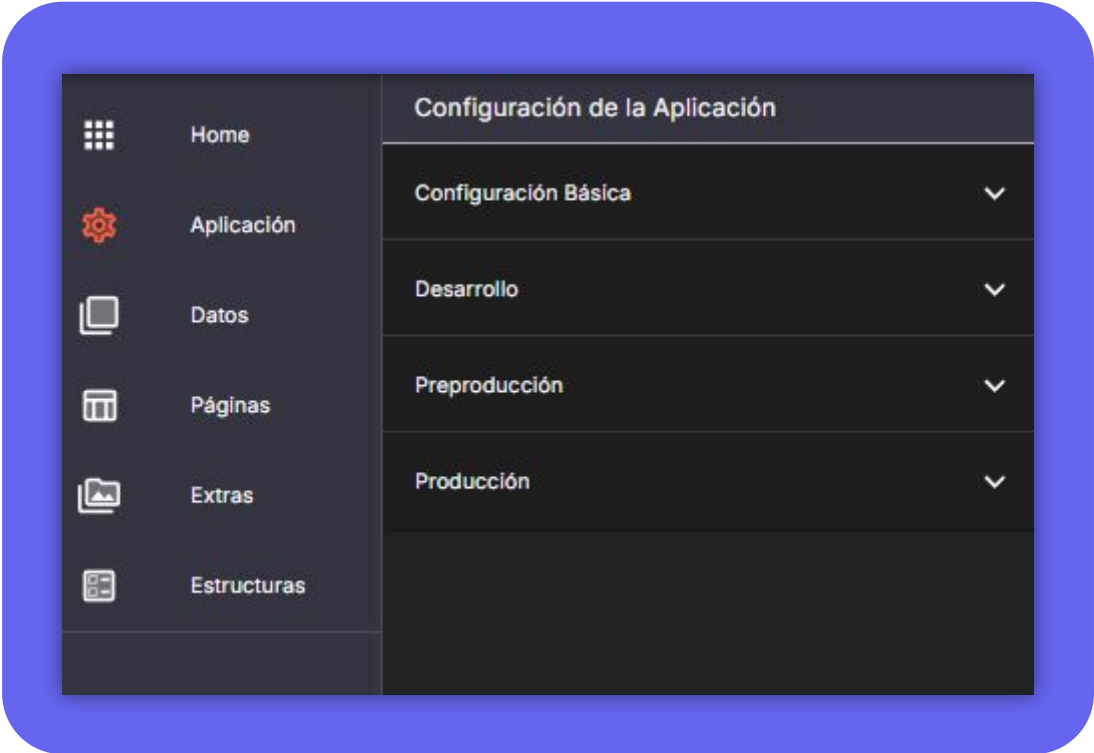
Un recorrido por su interfaz

Aptugo, una vez creado nuestra aplicacion, cuenta con diferentes ambientes de trabajo, correspondientes a las capas presentes en toda aplicaci3n web, y de la propia herramienta. Recorramos cada una de ellas:

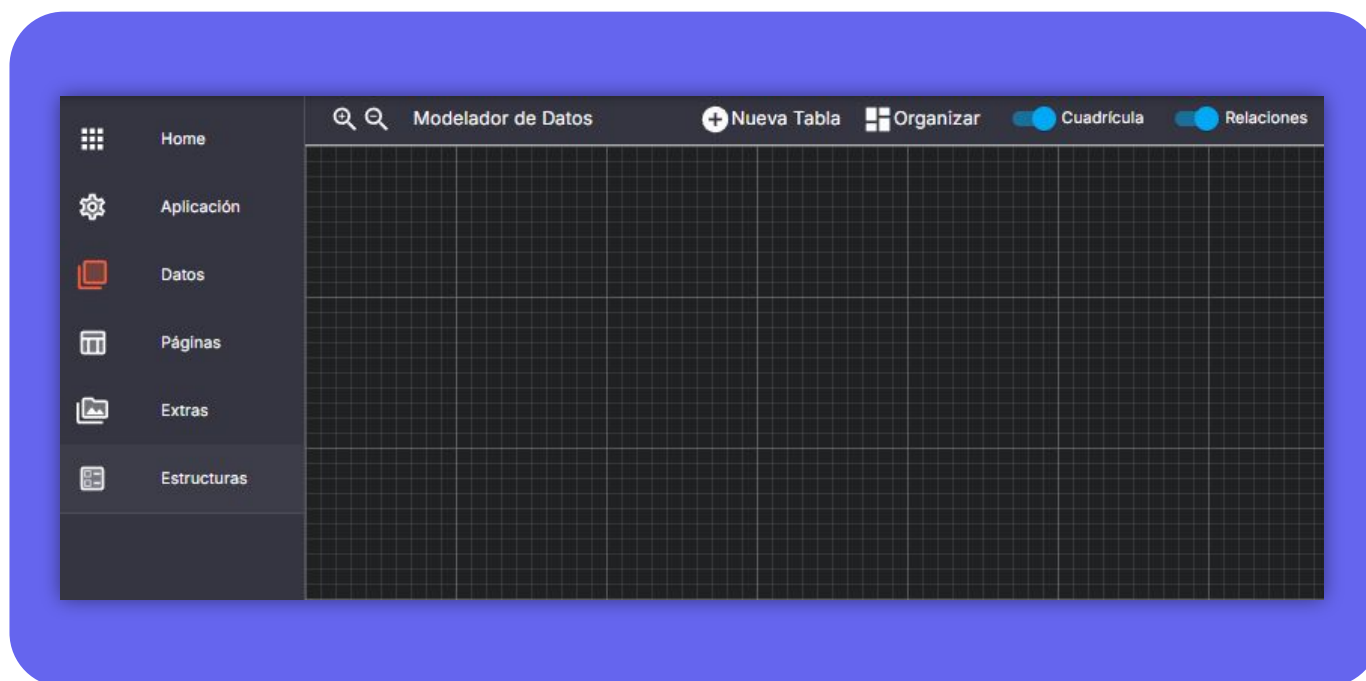
Home: Es la pantalla principal, que nos permite crear o abrir aplicaciones.



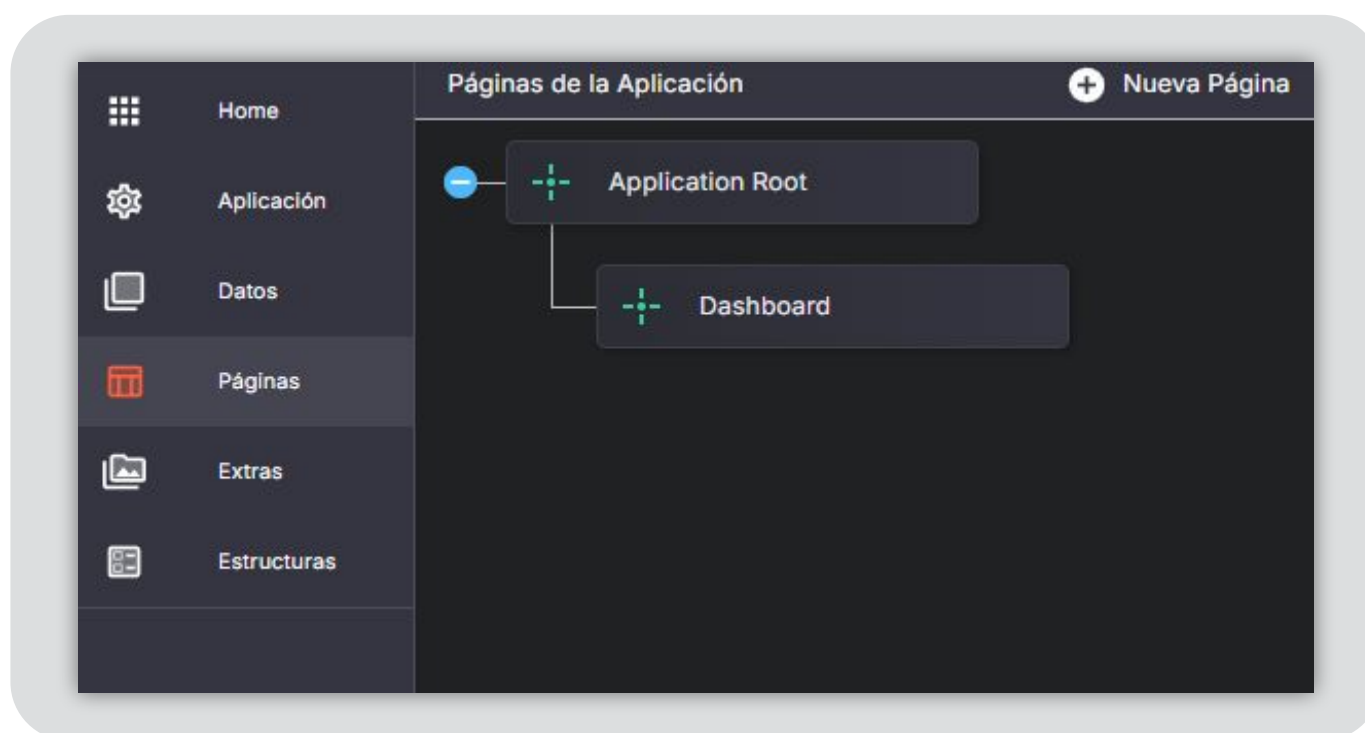
Aplicaci3n: Esta pantalla permite realizar modificaciones en la configuraci3n de la aplicaci3n, como el nombre de la misma. Adem3s, encontramos el seteo de los diferentes compilados disponibles como son Development (local), Stagging (pruebas) y Production (web).



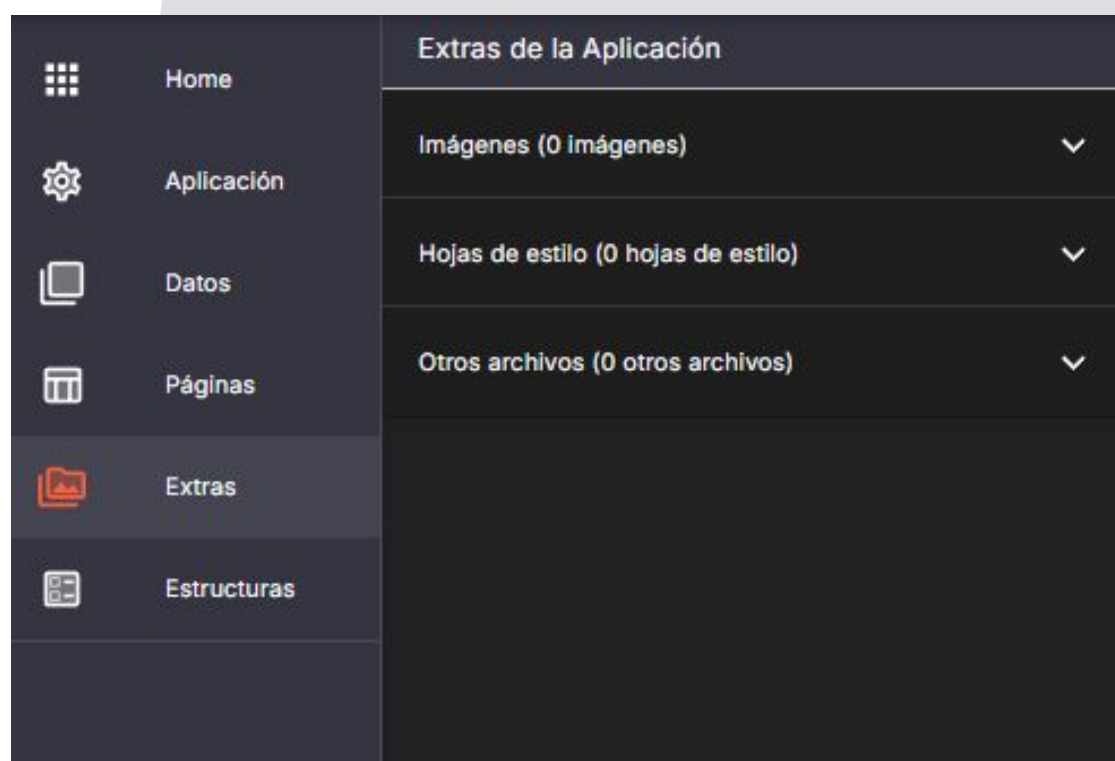
Datos: En esta sección se especializa en el desarrollo backend de la aplicación, todo aquello vinculado a la modelización de los datos. Aquí, podemos crear las diferentes tablas y datos que poseerá nuestro proyecto, encontrar las rutas que comunican el servidor con el navegador, y otros detalles importantes para el buen funcionamiento de la app web.



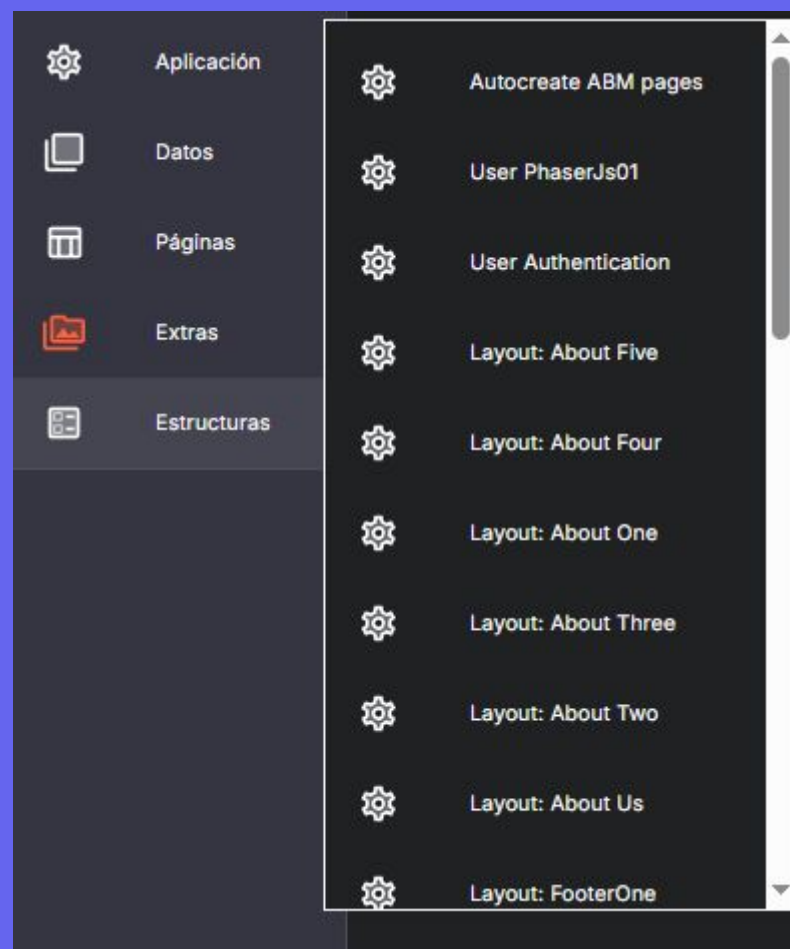
Páginas: El presente entorno se especializa en el desarrollo frontend del proyecto. Las páginas hacen referencia a las diferentes áreas visuales de la aplicación, las cuales interactúan con el usuario. En ellas trabajaremos aspectos tanto de estilo y estética, como de funcionalidad y animaciones. Por defecto, Aptugo crea una página raíz y otra principal, siendo la última, el Home de nuestro proyecto.



Extras: Aquí podremos incorporar tres elementos principales, vinculados a la estética de la aplicación. En la sección "Imágenes" encontraremos todas las imágenes que necesitamos cargar directamente en la aplicación. La segunda, "Hojas de estilos" permite crear y trabajar hojas de estilo escritas con lenguaje CSS. Y la tercera "Otros archivos" nos permite visualizar otro tipo de archivos no relacionado a los anteriores.



Estructuras: Si bien esta sección no entraría en la categoría de "entorno de trabajo", saber de ellas es realmente útil. Aptugo permite ingresar funcionalidades completas con un solo click, mediante la incorporación automática y preseteada de todos los elementos necesarios para dicha función. Podemos citar como ejemplo la opción final, "user Authentication".



Ejecutando nuestro "HelloWorld"

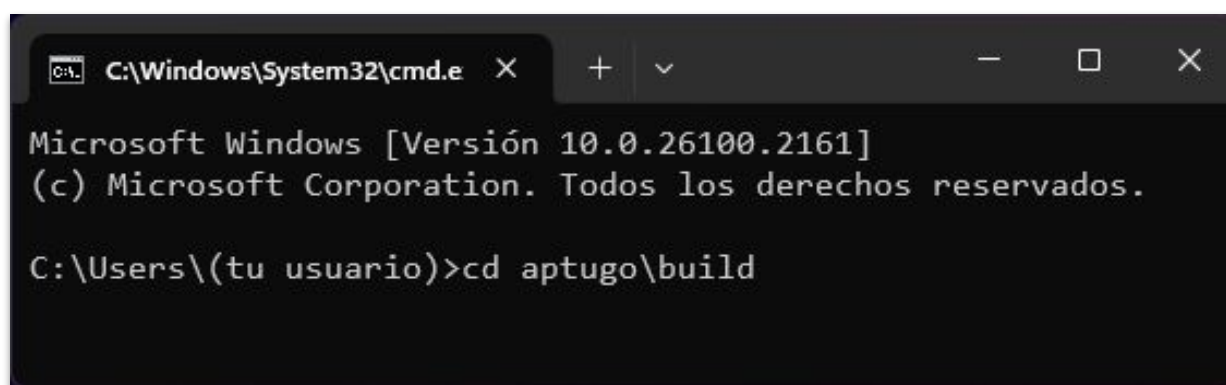
Para ejecutar nuestra primera app "**HelloWorld**", debemos dirigirnos desde el **Símbolo de sistema** o **CMD** a la carpeta o directorio donde está el código escrito por **Aptugo**.

Dicha carpeta se encuentra en esta ruta o path del sistema:

C:\Users\tu usuario\Aptugo\build\HelloWorld

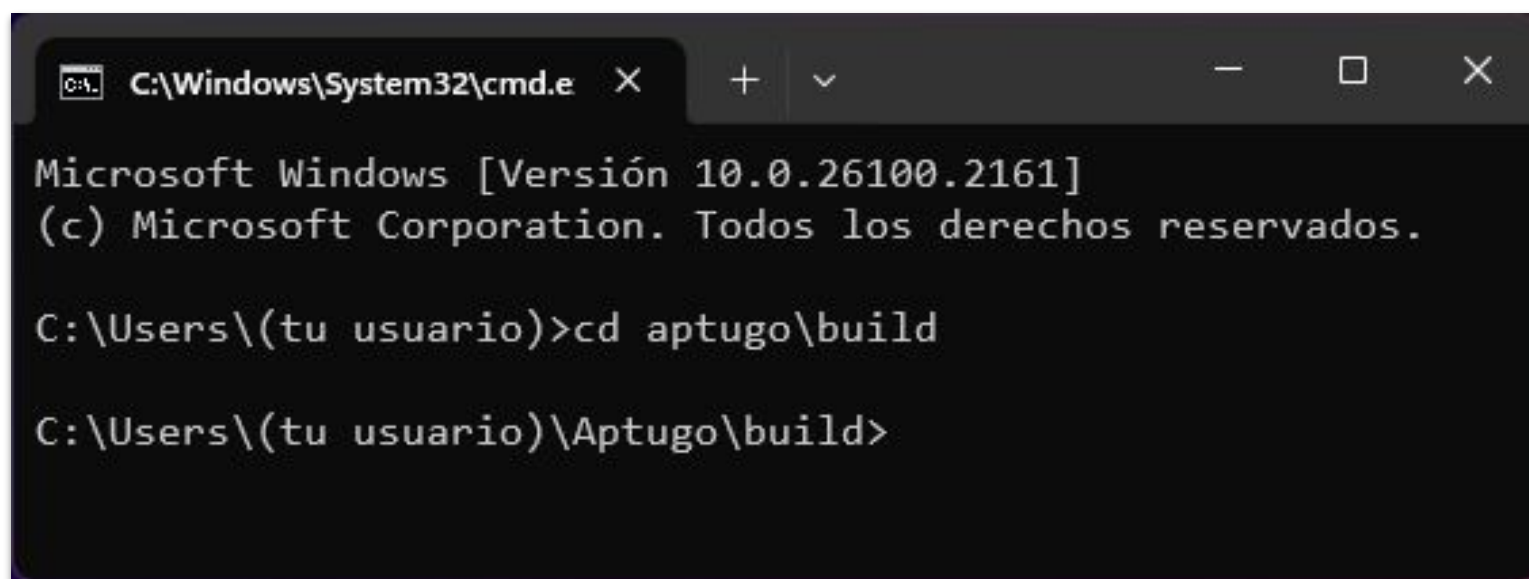
Y para llegar a ese directorio debemos realizar los siguientes pasos.
Abrimos el **Símbolo de sistema** o **CMD** y ejecutamos el siguiente comando:

cd aptugo\build



```
C:\Windows\System32\cmd.e x + v - □ x
Microsoft Windows [Versión 10.0.26100.2161]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\tu usuario>cd aptugo\build
```

Luego de presionar **enter**, logramos situarnos en la carpeta donde está y estarán los códigos escrito por Aptugo de cada una de las aplicaciones creadas.

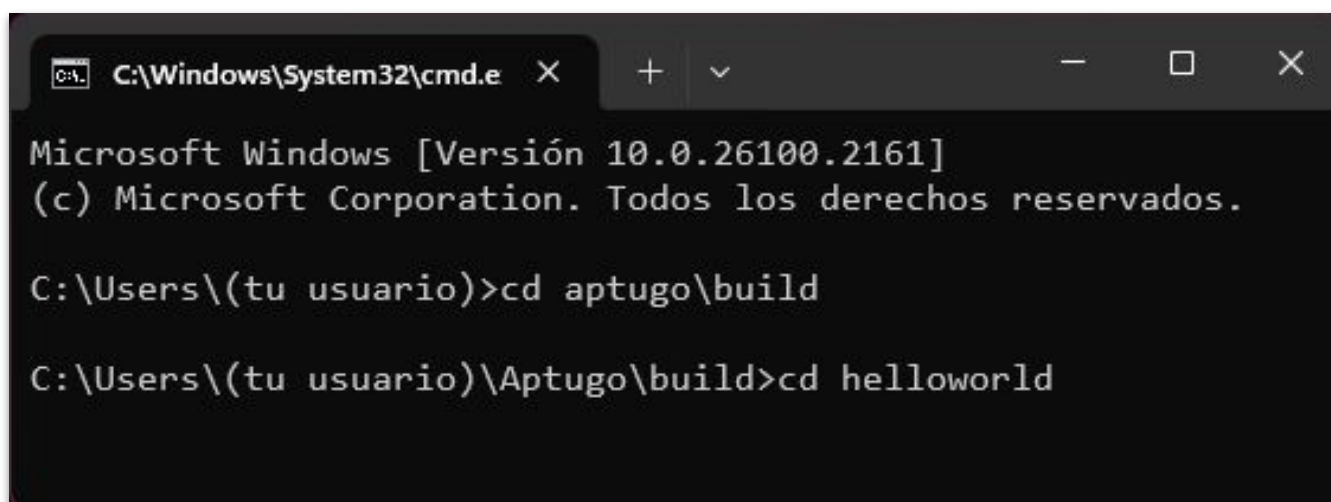


```
C:\Windows\System32\cmd.e x + v - □ x
Microsoft Windows [Versión 10.0.26100.2161]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\tu usuario>cd aptugo\build
C:\Users\tu usuario\Aptugo\build>
```

Solo nos queda ingresar a la carpeta de nuestra app.

Para esto debemos ejecutar el comando con el nombre de nuestra app, en este caso:

cd helloworld

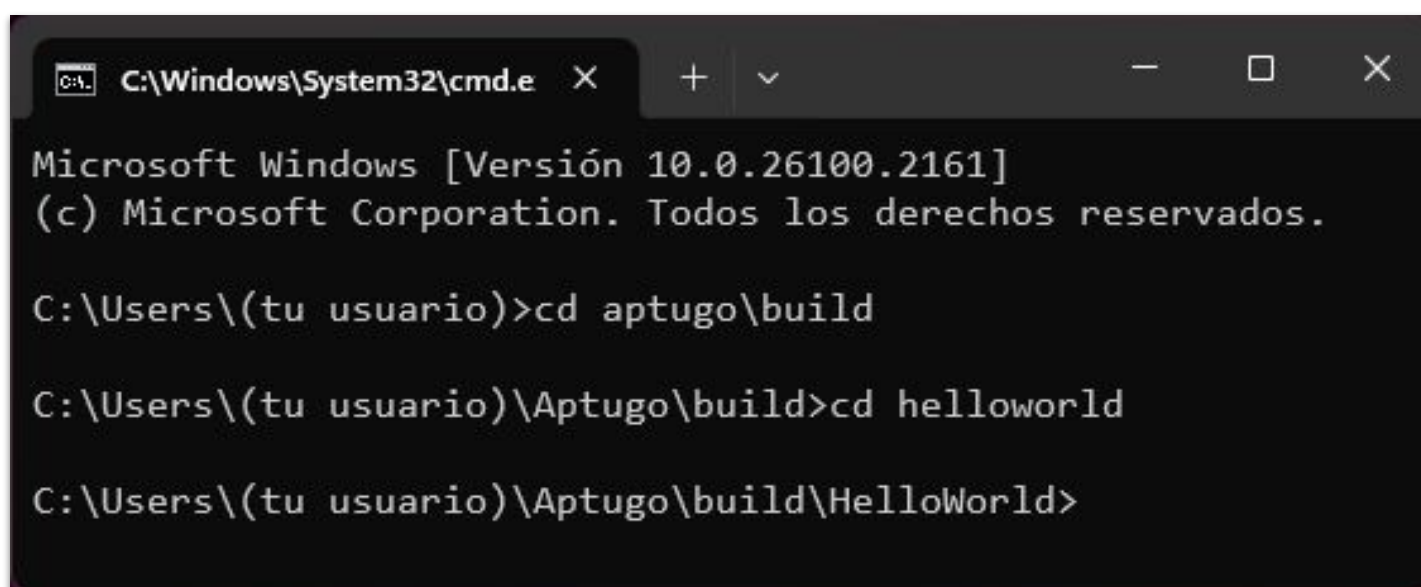


```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Versión 10.0.26100.2161]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\tu_usuario>cd aptugo\build

C:\Users\tu_usuario\Aptugo\build>cd helloworld
```

Presionamos **enter** y ya estamos situados en la carpeta donde podremos ejecutar la app.



```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Versión 10.0.26100.2161]
(c) Microsoft Corporation. Todos los derechos reservados.

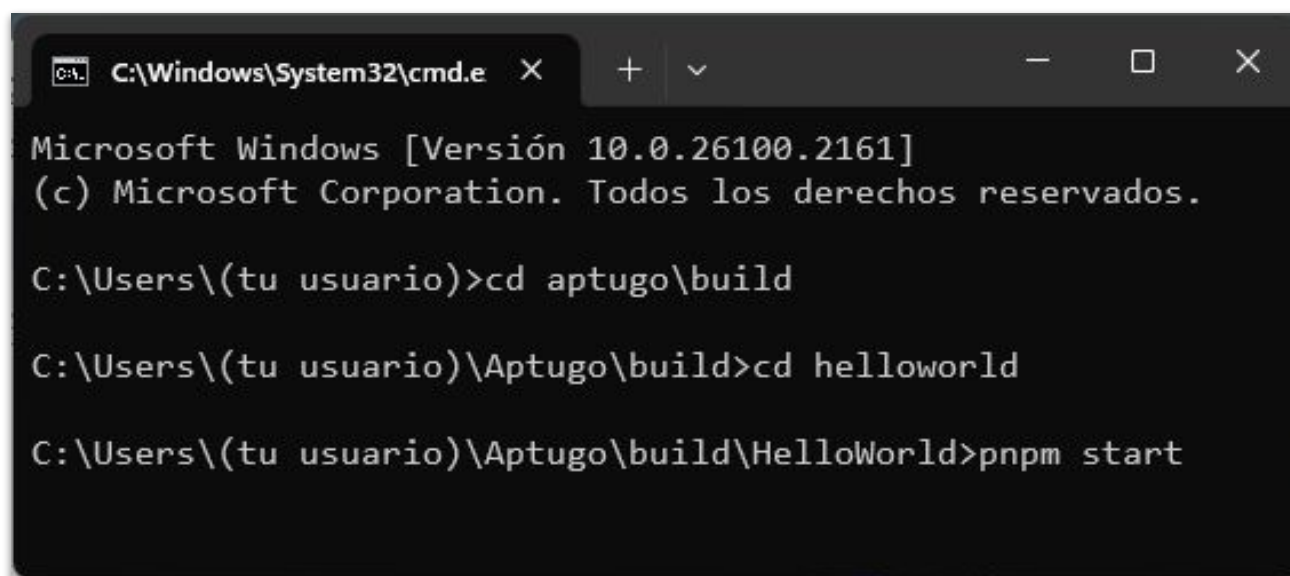
C:\Users\tu_usuario>cd aptugo\build

C:\Users\tu_usuario\Aptugo\build>cd helloworld

C:\Users\tu_usuario\Aptugo\build\HelloWorld>
```

Una vez allí ejecutaremos nuestro servidor local con el comando **"pnpm start"** de la siguiente manera:

pnpm start



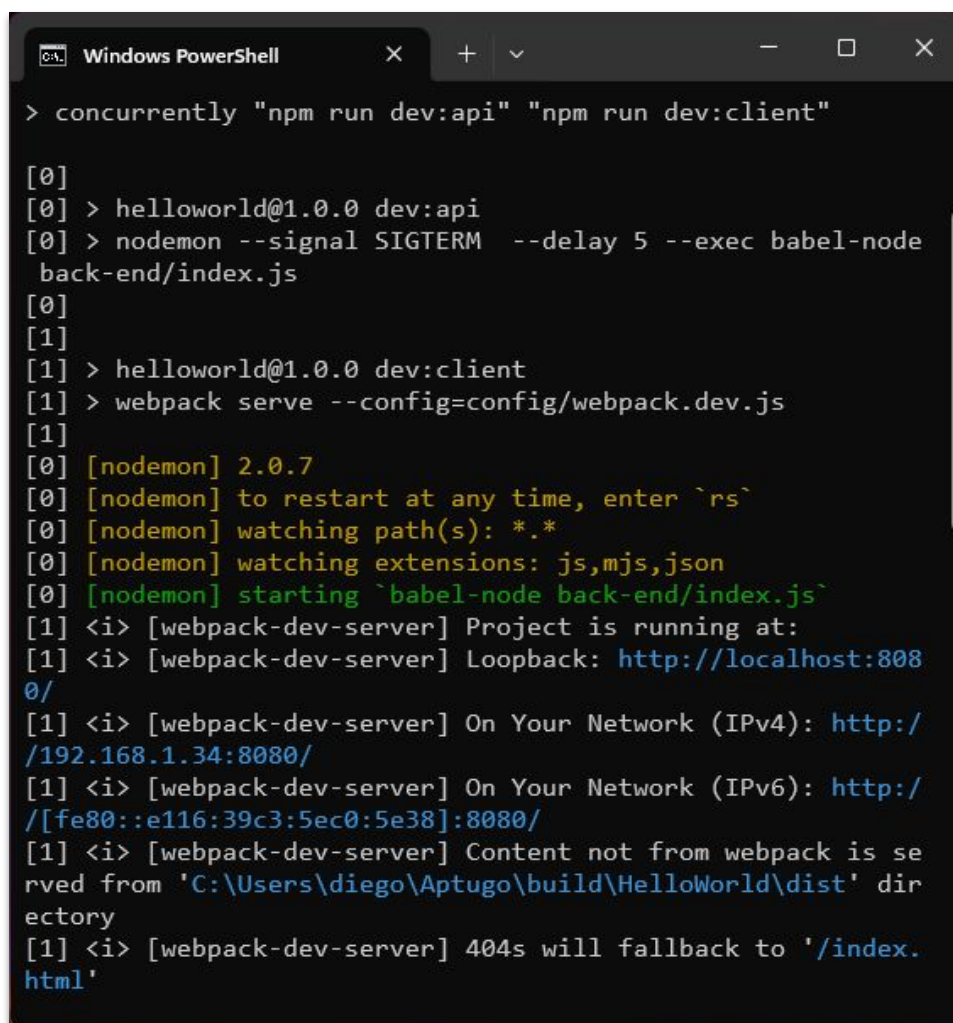
```
C:\Windows\System32\cmd.e X + - □ X
Microsoft Windows [Versión 10.0.26100.2161]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\tu_usuario>cd aptugo\build

C:\Users\tu_usuario\Aptugo\build>cd helloworld

C:\Users\tu_usuario\Aptugo\build\HelloWorld>pnpm start
```

Al presionar **enter** comenzará a ejecutar nuestra aplicación:

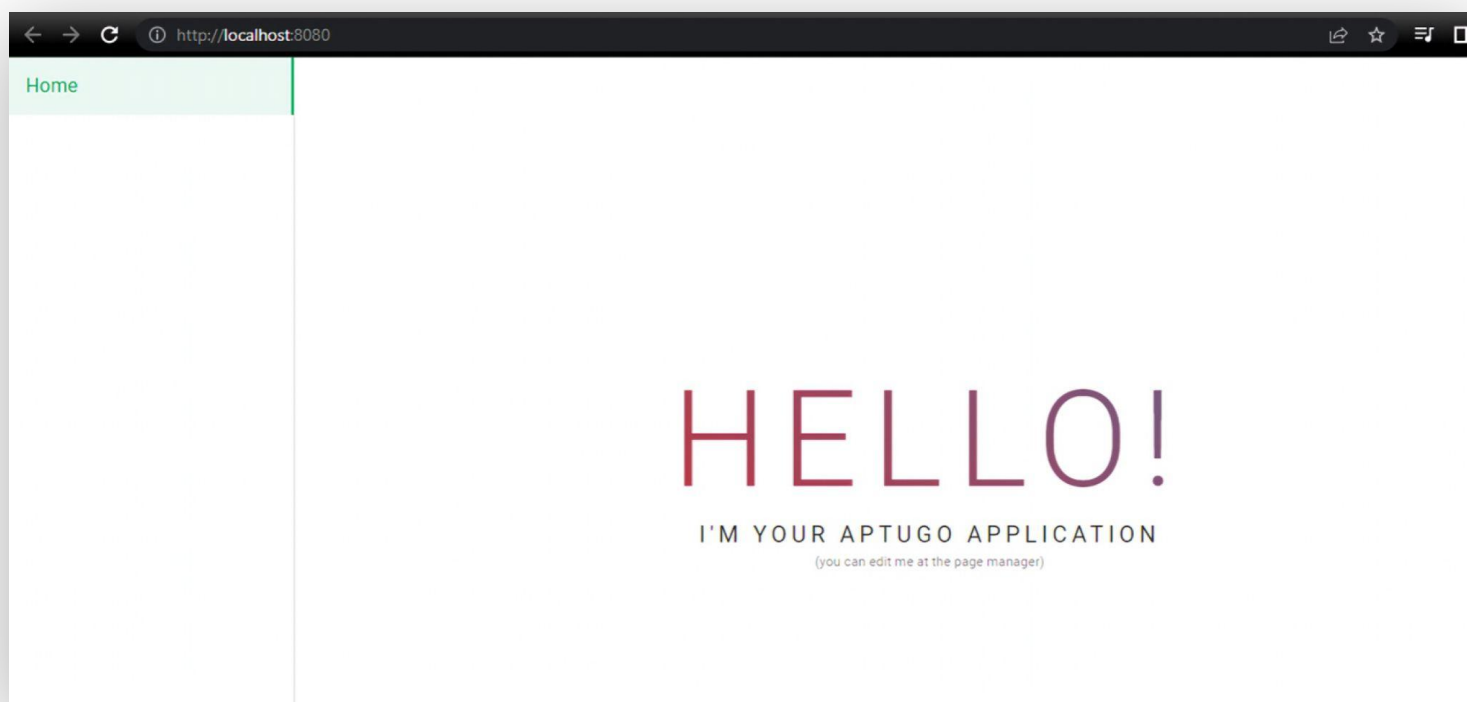


```
Windows PowerShell X + - □ X
> concurrently "npm run dev:api" "npm run dev:client"

[0]
[0] > helloworld@1.0.0 dev:api
[0] > nodemon --signal SIGTERM --delay 5 --exec babel-node
    back-end/index.js
[0]
[1]
[1] > helloworld@1.0.0 dev:client
[1] > webpack serve --config=config/webpack.dev.js
[1]
[0] [nodemon] 2.0.7
[0] [nodemon] to restart at any time, enter `rs`
[0] [nodemon] watching path(s): *.*
[0] [nodemon] watching extensions: js,mjs,json
[0] [nodemon] starting `babel-node back-end/index.js`
[1] <i> [webpack-dev-server] Project is running at:
[1] <i> [webpack-dev-server] Loopback: http://localhost:8080/
[1] <i> [webpack-dev-server] On Your Network (IPv4): http://192.168.1.34:8080/
[1] <i> [webpack-dev-server] On Your Network (IPv6): http://[fe80::e116:39c3:5ec0:5e38]:8080/
[1] <i> [webpack-dev-server] Content not from webpack is served from 'C:\Users\diego\Aptugo\build\HelloWorld\dist' directory
[1] <i> [webpack-dev-server] 404s will fallback to '/index.html'
```


Por último se abrirá el navegador con nuestro proyecto funcionando en el entorno **local**.

El proyecto estará disponible en **<https://localhost:8080>**



¡Listo!

Ya nos queda Aptugo funcionando
perfectamente.
¡Felicidades!



Video de Como Ejecutar el HelloWorld

Conclusión

Ahora ya tenemos en claro las distintas etapas que involucra el desarrollo de software y la importancia de cada una de ellas. Como vimos, el código es una parte más de este proceso, pero no necesariamente la más importante. Lo que distinguirá a un buen programador no será su conocimiento de más o menos lenguajes de programación, sino su capacidad de comprender y estructurar de manera lógica, simple y sustentable el proyecto a encarar.

Vamos a ir introduciéndonos poco a poco en el mundo de la programación, creando tus primeras aplicaciones web, conociendo algunos términos principales y animándonos a intervenir en la herramienta. Con el correr del tiempo, a medida que se acumula la lectura, experiencia y práctica, todos estos conocimientos se irán profundizando.

¡Esto recién empieza!