

Conceptos Básicos de la Programación No-Code

En este capítulo exploraremos los fundamentos de la programación sin código, o No-Code, que te permitirán crear automatizaciones de forma visual y sin necesidad de escribir código manualmente. Si bien el No-Code parece muy diferente a la programación tradicional, comparten una base lógica fundamental. Aprenderemos a comprender esta lógica y a utilizarla para sacarle el máximo provecho a las herramientas No-Code.

Diferencias entre programación sin código y programación tradicional

Si bien podríamos pensar que la programación sin código es muy diferente a la programación tradicional, el trasfondo en realidad es el mismo. La diferencia principal radica en la forma en que interactuamos con la lógica de programación.

- **Plataformas Visuales:** El No-Code utiliza plataformas visuales que nos permiten construir la lógica de una automatización arrastrando y soltando bloques predefinidos, sin escribir código línea por línea.
- **Lógica de Programación:** A pesar de la interfaz visual, las bases de la programación siguen siendo las mismas. Necesitamos comprender los conceptos de secuencialidad, flujo de control, variables y condicionales para aplicar correctamente el No-Code.

Conceptos Clave

Para poder aplicar el No-Code de forma efectiva, debemos entender los siguientes conceptos clave:

1. Secuencialidad

La secuencialidad se refiere al orden en que se ejecutan las acciones en un programa.

- **Orden de Ejecución:** Los ordenadores procesan las instrucciones paso a paso, en un orden predeterminado.
- **Make:** En la plataforma Make, por ejemplo, el flujo de trabajo se lee de izquierda a derecha y de arriba a abajo. Cada bloque representa una acción que se ejecutará en ese orden.
- **Importancia del Orden:** Es crucial definir el orden correcto de las acciones para que la automatización funcione según lo esperado. Si intentamos aplicar la lógica "de abajo a arriba", necesitaremos usar herramientas como interrupciones o definir órdenes específicas para que la computadora pueda entender la secuencia.

2. Flujo de Control

El flujo de control se refiere a las herramientas que nos permiten controlar qué acciones se ejecutan en una automatización y en qué orden.

- **Control de Eventos:** Los condicionales, bucles y otras herramientas de flujo de control nos permiten definir caminos alternativos dentro de una automatización, dependiendo de las condiciones que se cumplan o de los eventos que se produzcan.
- **Limitaciones:** Es importante recordar que el flujo de control está limitado por la lógica del ordenador. Las herramientas No-Code nos facilitan la implementación de la lógica, pero es importante comprender las bases para poder utilizarlas correctamente.

3. Iteraciones

Las iteraciones se refieren a la capacidad de repetir una acción o un conjunto de acciones varias veces.

- **Repetición de Procesos:** Los bucles nos permiten repetir una acción un número determinado de veces o hasta que se cumpla una condición.
- **Ejemplo:** Si tenemos una variable que contiene una lista de nombres, podemos usar una iteración para recorrer la lista y realizar una acción con cada nombre, como enviar un correo electrónico personalizado.

Elementos básicos en la programación No-Code:

1. Variables

Las variables son espacios en la memoria del ordenador que almacenan información para ser utilizada por la automatización.

- **Almacenamiento de Datos:** Las variables nos permiten guardar información de diferentes tipos, como texto, números, fechas, booleanos (verdadero/falso) y listas de datos.
- **Ejemplos:**
 - En Google Sheets o Microsoft Excel, las variables se almacenan en las celdas.
 - En Make, las variables se pueden almacenar en la base de datos integrada de la plataforma.
- **Uso en Automatizaciones:** Las variables son fundamentales para la personalización y el procesamiento de información en automatizaciones. Por ejemplo, podemos usar una variable para almacenar el nombre de un usuario y luego usar esa variable para enviar un correo electrónico personalizado.

2. Tipos de Datos

Los tipos de datos se refieren a las diferentes categorías de información que se pueden almacenar en una variable. Es importante definir el tipo de dato correcto para cada variable, ya que esto determina cómo se procesará la información.

- **Ejemplos:**

- **Números:** Se usan para almacenar cantidades numéricas, como la edad de una persona o el precio de un producto.
- **Cadenas de Texto:** Se usan para almacenar texto, como nombres, apellidos, direcciones o mensajes.
- **Booleanos:** Se usan para almacenar valores de verdadero o falso, que son útiles para la toma de decisiones dentro de la automatización.
- **Arrays o Listas:** Permiten almacenar una colección de datos del mismo tipo en una sola variable. Por ejemplo, podemos tener una lista de nombres de usuarios o una lista de precios de productos.

3. Condicionales

Los condicionales son herramientas que nos permiten controlar el flujo de trabajo de una automatización. Permiten definir caminos alternativos de ejecución, dependiendo de si se cumple o no una condición.

- **Ejemplo:** En una automatización de WhatsApp, podemos usar un condicional para identificar si el mensaje recibido es un audio o un texto, y luego realizar acciones diferentes en cada caso. Si es un audio, podríamos enviarlo a un servicio de transcripción, mientras que si es un texto, podríamos procesarlo directamente.
- **Componentes de un Condicional:**
 - **Condición:** Es una expresión que se evalúa como verdadera o falsa. Por ejemplo, "el mensaje es un audio" o "el usuario es mayor de edad".
 - **Caminos:** Son las acciones que se ejecutan dependiendo de si se cumple o no la condición.
- **Tipos de Condicionales:**
 - **Comparación:** Se usan para comparar valores numéricos o de texto. Por ejemplo, "si el precio es mayor a 100" o "si el nombre es igual a 'Juan'".
 - **Igualdad:** Se usan para determinar si dos valores son iguales o diferentes. Por ejemplo, "si el correo electrónico es igual al correo electrónico registrado".
 - **Booleanos:** Se usan para evaluar si una variable booleana es verdadera o falsa. Por ejemplo, "si el usuario está activo" o "si el pago se realizó correctamente".
- **Recomendaciones:**

Para trabajar con condicionales de forma efectiva, te recomiendo:

- Mantenerlo Simple: Si la condición es compleja, divídela en condiciones más simples y utiliza operadores lógicos como AND y OR para combinarlos.
- Documentar Todo: Anota las condiciones que has definido y las acciones que se ejecutan en cada camino. Esto te ayudará a entender mejor la lógica de la automatización y a solucionar problemas en el futuro.
- Realizar Pruebas: Prueba la automatización con diferentes inputs para asegurarte de que funciona correctamente en todas las situaciones posibles.

4. Bucles

Los bucles son herramientas que permiten repetir una acción o un conjunto de acciones un número determinado de veces o hasta que se cumpla una condición. Son muy útiles para automatizar tareas repetitivas.

- **Tipos de Bucles:**
 - For: Este bucle se utiliza para repetir una acción un número específico de veces. Por ejemplo, si queremos enviar un correo electrónico a 100 usuarios, podemos usar un bucle For para recorrer una lista de correos electrónicos y enviar un correo electrónico a cada uno de ellos.
 - While: Este bucle se utiliza para repetir una acción hasta que se cumpla una condición. Por ejemplo, podemos usar un bucle While para esperar la confirmación de un pago por parte de una plataforma externa. El bucle se ejecutará hasta que se reciba la confirmación de pago.

Mapas de Procesos

Los mapas de procesos son diagramas que representan visualmente el flujo de trabajo de un proceso. Son muy útiles para comprender el funcionamiento de un proceso, identificar inefficiencias y áreas de mejora, y planificar la automatización del proceso.

Beneficios de los Mapas de Procesos:

- Visualización Clara: Facilitan la comprensión del proceso, ya que muestran de forma clara los pasos, las decisiones y las responsabilidades.
- Eficiencia: Permiten identificar y eliminar pasos innecesarios o redundantes, lo que mejora la eficiencia del proceso.
- Automatización: Facilitan la planificación de la automatización del proceso, ya que proporcionan una visión detallada de los pasos que deben automatizarse.

Elementos Clave de un Mapa de Procesos:

- Entradas y Salidas: Definición de la información que se necesita para iniciar el proceso y los resultados esperados.

- Tareas y Actividades: Definición de los pasos específicos que se deben realizar para completar el proceso.
- Decisiones: Puntos de ramificación en el proceso, donde se toman decisiones basadas en una condición.
- Responsabilidades: Asignación de roles para cada paso del proceso, lo que define quién es responsable de cada tarea.
- Documentación: Registro de detalles relevantes del proceso, como normas, procedimientos, información de contacto, etc.

Herramientas para Hacer Mapas de Procesos:

Existen diversas herramientas que se pueden utilizar para hacer mapas de procesos, como:

- Lucidchart
- Miro
- Microsoft Visio
- Bizagi
- Lápiz y Papel

La herramienta que se utilice no es lo más importante, sino la claridad y la precisión con la que se represente el proceso.

Recomendaciones:

- Definir Claramente el Objetivo del Proceso: Es importante tener claro cuál es el objetivo del proceso antes de empezar a hacer el mapa.
- Involucrar a las Personas que Participan en el Proceso: Es fundamental involucrar a las personas que participan en el proceso en la elaboración del mapa. Esto te ayudará a obtener una visión más completa del proceso.
- Ser Detallista: Incluye todos los pasos, las decisiones y las responsabilidades en el mapa. Cuanto más detallado sea el mapa, más útil será para comprender y automatizar el proceso.

Ejemplo de un Mapa de Procesos para un Chatbot de Atención al Cliente por WhatsApp:

Entradas:

- Consulta del cliente por WhatsApp.

Tareas y Actividades:

- Recibir el mensaje del cliente.
- Identificar el tipo de cliente (nuevo cliente, cliente existente).
- En función del tipo de cliente, derivar la consulta al departamento correspondiente (ventas, postventa, logística, calidad).

- Responder al cliente con la información solicitada.

Decisiones:

- ¿Es un cliente nuevo o existente?
- ¿A qué departamento debe derivarse la consulta?

Responsabilidades:

- Chatbot: Recibir la consulta del cliente, identificar el tipo de cliente y derivar la consulta al departamento correspondiente.
- Departamento correspondiente: Responder al cliente con la información solicitada.

Documentación:

- Protocolo de atención al cliente por WhatsApp.
- Preguntas frecuentes.

Este es solo un ejemplo básico, y el mapa de procesos real sería mucho más complejo, dependiendo de la empresa y sus necesidades.

Conclusión:

Comprender los fundamentos de la programación No-Code es esencial para aprovechar al máximo las herramientas de automatización. Los mapas de procesos son una herramienta clave para la planificación de automatizaciones, ya que nos permiten visualizar y optimizar el flujo de trabajo antes de trasladarlo a una plataforma No-Code. A medida que te familiarices con estos conceptos y herramientas, podrás crear automatizaciones cada vez más complejas y eficientes.