

Contadores y Memorias

Laboratorio #2

Integrantes:	Ariel Núñez.
Profesor:	Ricardo Finger.
Auxiliar:	Sergio Saavedra.
Ayudantes de Laboratorio:	Alonso Caviedes.
	Daniel Vásquez.
	Diego Gonzalez.
	Néstor Henríquez..

Fecha de entrega: 24 de mayo de 2020
Santiago, Chile

1. Introducción

A partir de la siguiente experiencia de laboratorio se busca comprender el funcionamiento y diseño de circuitos cuyos componentes utilizan memoria, como los Flip Flop y una memoria ROM. Para esto, se diseñará y analizará un circuito capaz de contar en binario, el cual podemos modificar para que cuente en un intervalo deseado por el usuario. Mediante este procedimiento, se busca entender como estos elementos tienen memoria y sus aplicaciones en circuitos.

Posteriormente, se hace uso de una memoria ROM con la cual podemos almacenar diversos valores de mayor complejidad en cantidad de bits, y los podemos llamar con un menor número de bits, usando por ejemplo nuestro contador previamente diseñado. Además, combinando la capacidad del contador de reiniciar su conteo en cierto valor, podemos determinar un patrón de información que se presente por la memoria, definido a través del valor en que reinicia el conteo y los datos guardados en la memoria ROM.

Así, el objetivo fundamental de esta experiencia es poder diseñar e implementar un dispositivo que pueda contar y reiniciar este conteo en un número escogido por el usuario sin necesidad de alterar el circuito, y que a partir de esto el usuario pueda hacer uso del contador mediante una memoria, para presentar información definida por este.

2. Metodología

En el desarrollo de esta experiencia se utilizó el software SimulID, versión 0.3.12. Este software permite la simulación de componentes lógicos, entre los cuales utilizaremos los componentes fundamentales (AND, OR, NOT, XOR) al igual que componentes como el Flip Flop D, Multiplexor y entradas de diverso tipo. Al seguir la experiencia, se utilizó una memoria ROM, la cual se le cargó información mediante un archivo tipo “data”, configurado a partir de las instrucciones presentadas.

2.1. Contador

Para el contador desde 0 a 25, se utilizó la siguiente configuración, la cual utiliza una entrada de tipo CLOCK y Flip Flops D en serie, utilizando una configuración en la cual la entrada D se retro alimenta a partir de la salida Q negada. Los Flip Flop D usados presentan sus pines R y S no invertidos, y su entrada de CLOCK invertida.

Este contador fue modificado para indicar una señal de alerta una vez se alcanza el número 25 (en binario claramente). Esta señal se enciende cuando el número “11001” se presenta en el contador, haciendo uso de una compuerta AND que se activa con este valor en específico. Una vez se identifica que el contador llega al número 25, la salida de la compuerta AND se conecta a un Flip Flop D, para que en el siguiente tick del reloj, este se encienda y alerte el circuito que se llegó al número deseado, y podamos reiniciar el conteo.

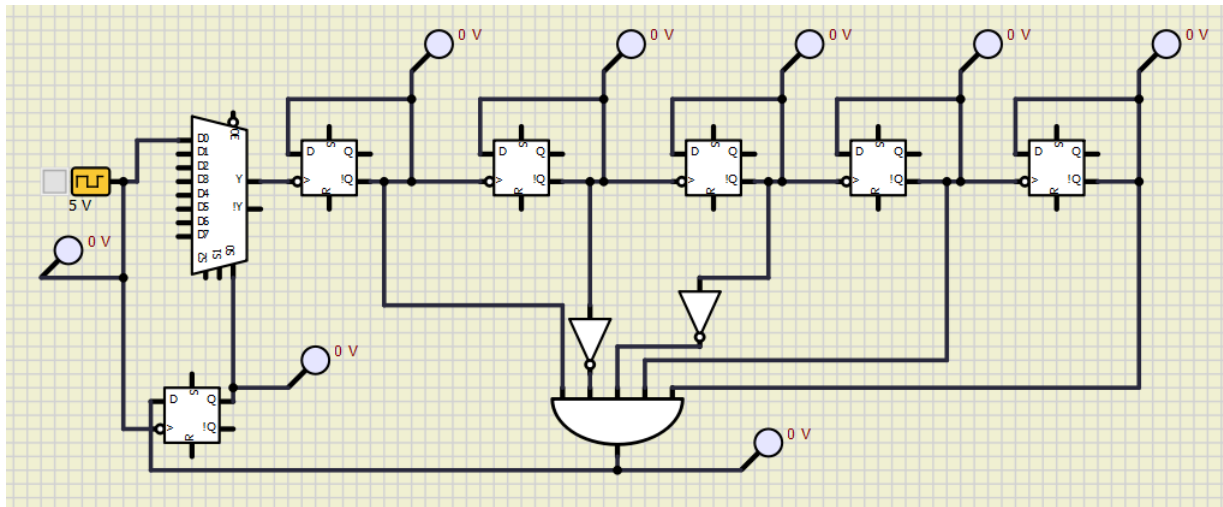


Figura 1: Contador de 5 bits, implementando una alerta al identificar el número “11001”, lo cual reiniciará el conteo desde “00000”.

La forma en que opera el Flip Flop que reiniciará el conteo, es mediante el uso de un Multiplexor. El contador original se modifica de forma que este se conecte al Multiplexor y la salida al primer Flip Flop en serie que realiza el conteo. Una vez el Flip Flop de alerta se enciende, este cambia el switch del Multiplexor, y desconecta el contador de la entrada del reloj. Esto hace que el contador

se detenga por un instante y prosiga su conteo en el siguiente tick del reloj. Dicho esto, queremos manipular los Flip Flop en el instante en que el conteo se detiene, y definir un número en el cual reinicie el conteo.

Para esto, la salida del Flip Flop de alerta se conecta a la entrada de un Multiplexor respectivo para cada Flip Flop de conteo, al igual que a un interruptor conectado al switch correspondiente de cada Multiplexor y a la entrada R del Flip Flop. La salida del Multiplexor correspondiente esta conectado a la entrada S del Flip Flop. La forma en que operan estas conexiones es que al abrir el interruptor, se enviará un valor alto a la entrada S del Flip Flop mientras que la entrada R se mantiene abierta, lo cual hace que en el siguiente tick de la entrada CLOCK la salida Q sea un valor alto, y por consecuencia, hacer un valor bajo la salida Q negada. Por el contrario, si se cierra el interruptor, esto activa el switch del Multiplexor y abre la entrada S, mientras que se conecta directamente un valor alto a la entrada R, lo cual hace que en el siguiente tick de la entrada CLOCK, la salida Q sea un valor bajo, y por consecuencia la salida Q negada un valor alto. Esto permite que controlemos el número deseado en el reinicio del conteo sin necesitar de alterar el circuito directamente y solamente manipulando las entradas de los Flip Flop a través de los interruptores.

Una vez ocurre el siguiente tick del reloj, el Flip Flop de alerta se apaga y el Multiplexor hace que los contadores reciban la entrada CLOCK nuevamente. Con los interruptores seleccionados por el usuario, se reinicia el conteo desde un número en específico. Además, se añadió un display de tipo hexadecimal para simplificar la representación del número resultante. A partir de esto, resulta el siguiente esquema de circuito.

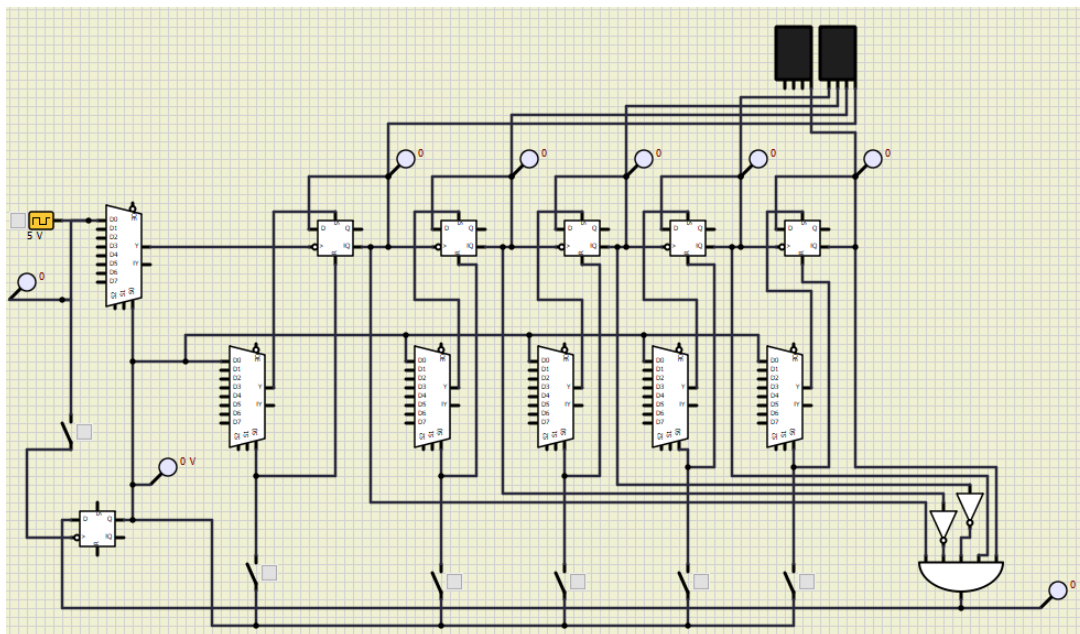


Figura 2: Contador de 5 bits, implementando el reinicio del conteo al identificar el número “11001”. Además, se implementan interruptores que definan un número en el cual el el conteo iniciará nuevamente.

2.2. Memoria ROM

La memoria ROM puede almacenar datos en valores decimales en sus múltiples bits de entrada, cargados directamente a la memoria mediante un archivo de tipo ".data", el cual define el usuario; al ingresar un número binario en entrada, se identifica el valor o palabra en la casilla de la memoria y se representa en los bits de salida, o datos. Para esta experiencia, la memoria requiere de 32 posibles direcciones, por lo tanto necesita 5 bits de dirección y 13 bits de datos.

Los pin de entrada se conectan directamente a las salidas de los Flip Flop de conteo respectivos, los cuales van a indicar la dirección de la memoria. Los pins de salida se conectan en un arreglo de LEDs de mismo tamaño (13 bits), los cuales van a representar la información almacenada en la dirección indicada por la entrada. En este caso, la información corresponde a números en base decimal capaces de ser representados en 13 bits. Además, la memoria ROM debe ser alimentada por una fuente de 5 Volts para funcionar apropiadamente, al igual que configurarse en las propiedades de la simulación.

La información en la memoria ROM se carga a través de un archivo de tipo ".data", el cual se crea a partir del programa Notepad, mediante las especificaciones dadas. Se creó este archivo, el cual posee 32 líneas (ya que se necesitan 2^n líneas, siendo n el número de bits, en este caso 5), siendo cada una de estas líneas una casilla de información dentro de la memoria ROM. En cada línea se ingresó un número en base decimal y una coma, asegurándonos de que los números ingresados puedan representarse en 13 bits.

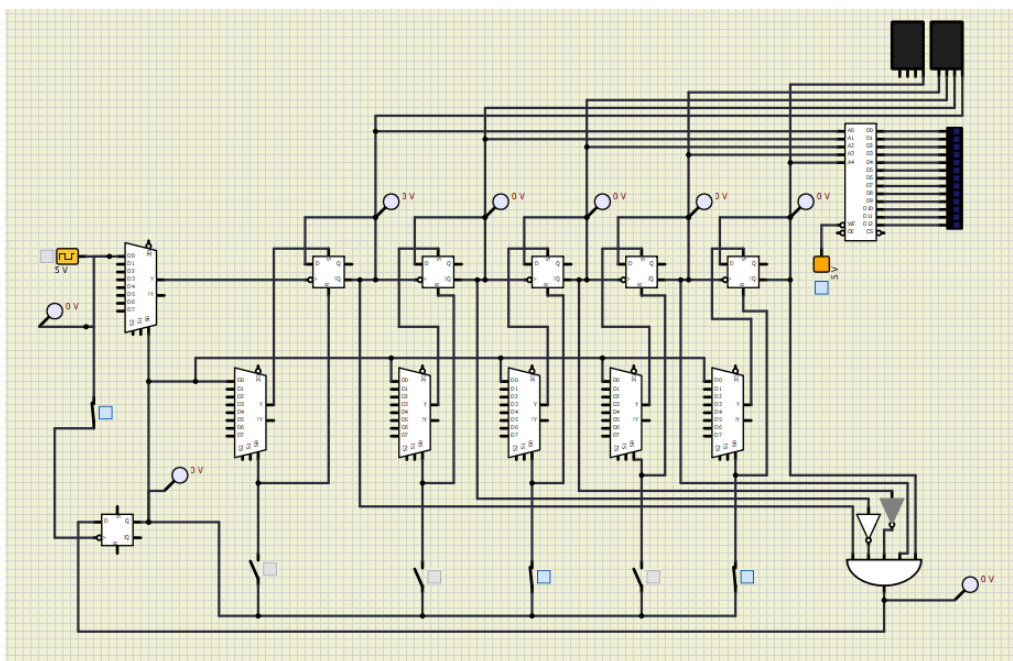


Figura 3: Circuito final: Contador de 5 bits con reinicio en número arbitrario, implementado junto a una memoria ROM que representa su información en un arreglo de LEDs.

3. Análisis

3.1. Contador

Al implementar el modelo de contador con Flip Flops D utilizado, se busca poder realizar este contador utilizando solamente una fuente de tipo CLOCK, con la cual el conjunto de Flip Flops estará basado. Vemos a continuación la forma de la onda de voltaje en la salida utilizada de cada Flip Flop, al igual que la entrada CLOCK.



Figura 4: Visualización lógica de las ondas en la entrada CLOCK, al igual que los primeros tres Flip Flop D del contador. Se puede apreciar la subida y bajada de voltaje, y como la frecuencia de un término equivale al doble del anterior

Vemos que a partir de la entrada CLOCK, la frecuencia del primer Flip Flop conectado resulta el doble de la onda de esta entrada. Al mismo tiempo, la frecuencia del segundo Flip Flop conectado en serie resulta el doble del primero; esta configuración nos permite realizar todas las combinaciones que necesitemos para un periodo de la onda del MSB (Most Significant Bit).

Notese que solo utilizamos una entrada CLOCK, por lo tanto, al utilizar esta configuración podemos seguir agregando el número de Flip Flop's que equivalga al número de bits que necesitemos, los cuales permitirán las combinaciones de bits necesarias para el conteo.

Al implementar la alerta de límite del conteo, en este caso el número 25 en base decimal, se implementó la compuerta AND que responda específicamente a la entrada "11001". Vemos que al llegar a este valor en el contador, efectivamente se emite una señal la cual detiene por un tick del reloj el conteo dada la configuración del Multiplexor.

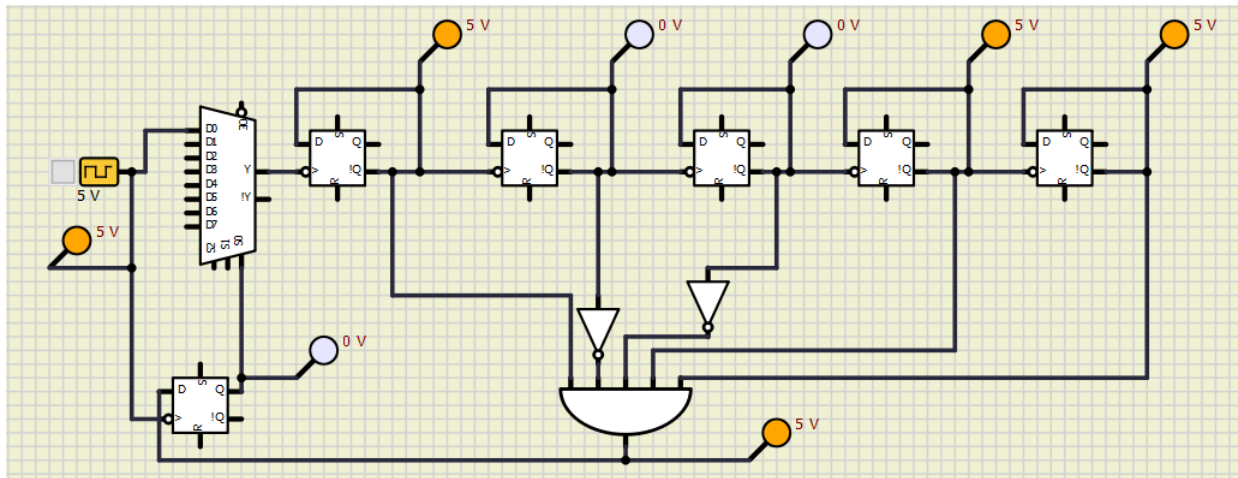


Figura 5: Contador con alerta al llegar a la entrada “11001”, esta emite una señal, la cual afectará al Flip Flop D en la que se conecta en el siguiente tick del reloj.

Enviada la señal, en el siguiente tick del reloj se prenderá el Flip Flop D conectado a la salida de la compuerta AND. La razón por la cual utilizamos este Flip Flop es debido a que queremos notificar al circuito al llegar a número 25, y realizar cambios en el siguiente tick, que originalmente sería el número 26, mas no queremos llegar a contar este número. Dicho esto, la configuración del circuito permite que reiniciemos todos los valores a 0 y empecemos el conteo desde el inicio, haciendo uso de las compuertas S y R de los Flip Flop D y el siguiente tick del reloj. Junto a esto, podemos hacer que ciertos Flip Flop inicien en un valor 1, para empezar el conteo en números mayores. Apreciamos esto en la siguiente figura.

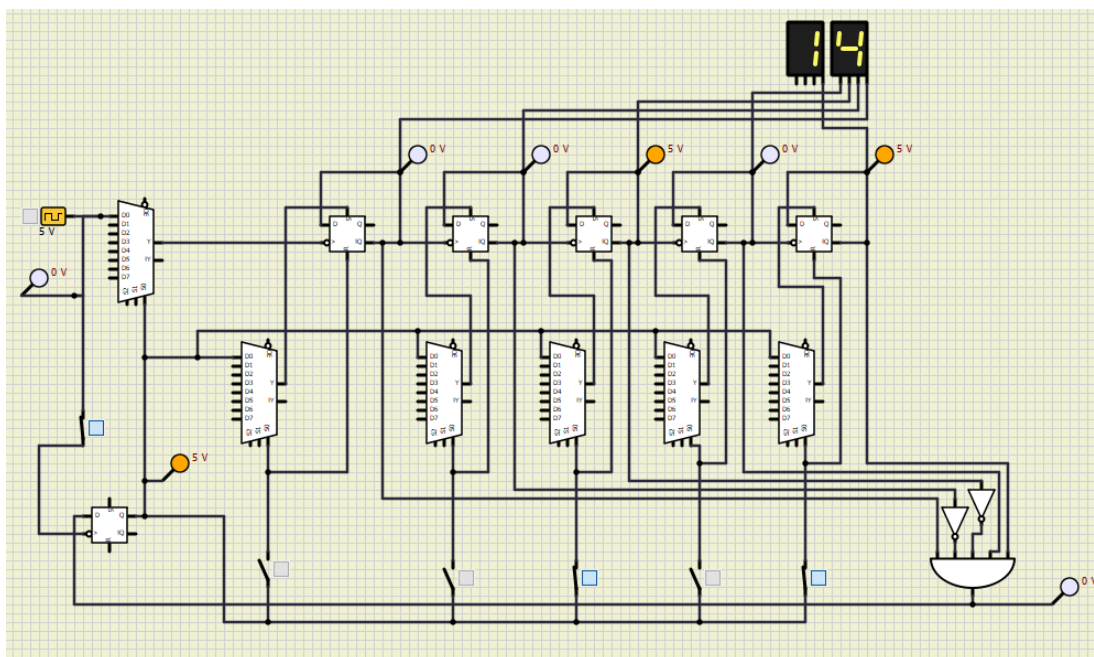


Figura 6: Contador reiniciando su conteo en un valor específico (en este caso el número 20 en base decimal, 14 en base hexadecimal o “10100” en binario.)

Vemos que el Flip Flop D que realiza los cambios a los bits interactúa con el circuito solo en este instante, y al avanzar al siguiente tick del reloj, el valor de este Flip Flop volverá a ser 0, dejando de afectar el circuito hasta que contemos nuevamente al número 25 en decimal, donde se encenderá nuevamente, haciendo de esto un ciclo. Además, notese que el usuario puede hacer diversos cambios a este circuito solamente medianete el uso de interruptores, como elegir el número en el cual se reiniciará el conteo, al igual que decidir si efectivamente hay reinicio, haciendo uso del interruptor con el cual el Flip Flop que modifica el contador se conecta al reloj principal del circuito.

3.2. Memoria ROM

Al implementar el contador junto a la memoria ROM, se vio que este efectivamente indicaba a la memoria ROM diversas casillas con información. A forma de prueba, se utilizaron dos archivos “.data”, en el cual el primero, “numerosconteo.data”, representa solo un conteo desde 0 a 31, mientras que el siguiente archivo, “numeroslimite.data”, utiliza valores decimales de mayor orden e ingresa el número en base decimal más grande que se puede representar en 13 bits, 8191.

A continuación, como ejemplo del funcionamiento de la ROM con el archivo “numerosconteo.data” cargado, vemos que al ingresar el número 25 decimal en la memoria ROM obtenido a partir del contador, la cual posee el valor 25 en decimal en la dirección número 25, se representa el número 25 decimal (“11001”) en forma binaria en el arreglo de LEDs.

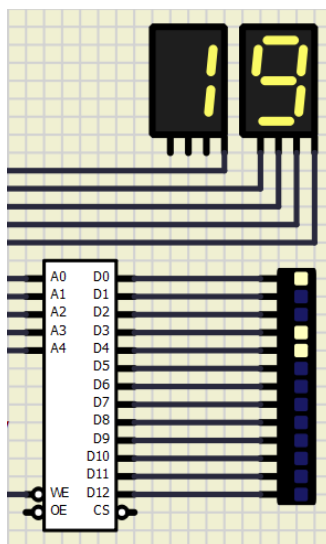


Figura 7: Ejemplo del funcionamiento de la memoria ROM utilizando el archivo “numerosconteo.data” con el contador, en el cual el valor “0000000011001” es representado en el arreglo de LEDs. Se representa el valor en hexadecimal para el contador, y en binario para la memoria ROM.

Utilizando el archivo “numeroslimite.data”, en el archivo la dirección 23 indica el valor 8191 en base decimal, el cual corresponde al número más grande que podemos representar en binario con 13 bits de información. Notamos que al ingresar este valor a partir de nuestro contador, obtenemos el número “111111111111” en el arreglo de LEDs.

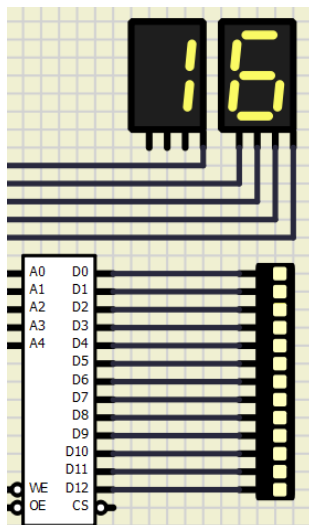


Figura 8: Ejemplo del funcionamiento de la memoria ROM utilizando el archivo “numeroslimite.data” con el contador, en el cual el valor “111111111111” es representado en el arreglo de LEDs, correspondiente al valor límite representable.

4. Conclusiones

A partir de los resultados obtenidos, la experiencia de laboratorio resultó exitosa en el diseño y análisis de un contador, al igual que su integración a una memoria ROM para representar información. Se logró diseñar un contador de 5 bits, el cual posee un límite de conteo en el cual reinicie, al igual que un diseño el cual permite al usuario definir un número arbitrario en el cual se reinicie el contador, si tener que alterar el circuito y solo sea necesario manipular interruptores. Junto a esto, se aplicó la funcionalidad del contador con una memoria ROM para poder representar información con un mayor número de bits, al igual que poder elegir un ciclo en específico de valores que se repitan a partir del número escogido para reiniciar el conteo en la primera parte de la experiencia.

Se destaca como al hacer uso de un circuito que posee solo 5 bits de información, se puede llegar a obtener hasta 32 palabras con una mayor cantidad de bits (en este caso 13), al hacer uso de un dispositivo como la memoria ROM, la cual es barata y fácil de utilizar, y de inmensa utilidad en dispositivos eléctricos que no requieran reemplazar o almacenar nueva información, sino que estos simplemente deben entregarla y representarla de un modo específico. Por lo tanto, el uso del contador de menor número de bits como una forma sencilla de indicar a la memoria que utilice cierto número de sus casillas y que repita cierto patrón, en vez de indicar cada número manualmente por el usuario, nos permite evidenciar como el uso en conjunto de múltiples funciones sencillas combinadas con diversos componentes nos permiten elaborar proyectos de mayor complejidad.