

RÚBRICA

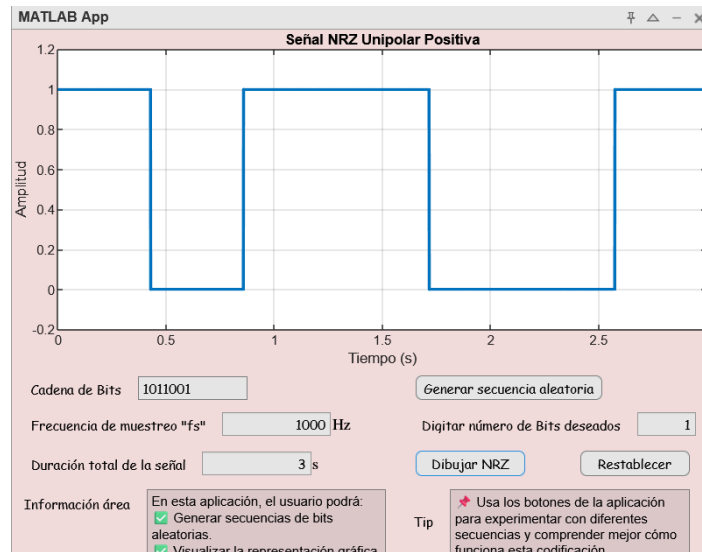
0	5	10
No entregó No se evidencia	Hace falta algo Tiene errores	Cumple correctamente No tiene errores

[50 point] AppDesigner

INTERFAZ	SCREENSHOT	EXPLICACIÓN
Unipolar Positiva NRZ [10p]	<p>Figura 1 <i>Señal NRZ Unipolar Positiva</i></p> <p>Nota. App de la señal Unipolar Positiva NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.</p>	<p>Descripción de la aplicación</p> <p>Esta aplicación permite visualizar la señal NRZ Unipolar Positiva a partir de una secuencia de bits ingresada por el usuario. En esta, se han añadido campos de ingreso de datos (<i>Numeric & Text</i>) para modificar parámetros como la cadena de bits, la frecuencia de muestreo y la duración total de la señal. Asimismo, se añadieron botones para la generación del gráfico, la generación de una cadena de bits totalmente aleatoria y un botón para restablecer todos los campos a su estado base (<i>default</i>). Además, como detalle adicional, se agregaron dos campos de texto con información relevante pero concisa acerca del programa y su funcionalidad; véase la Figura 1.</p> <p>Componentes de la interfaz</p> <ol style="list-style-type: none"> Cadena de Bits

Figura 2

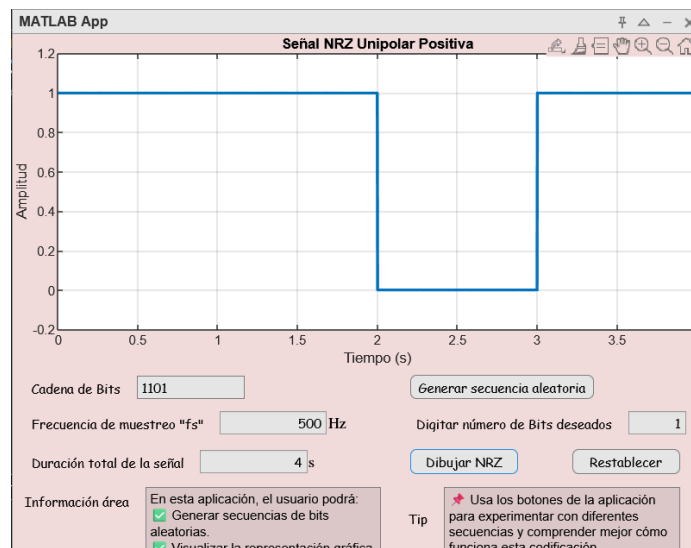
Señal NRZ Unipolar Positiva



Nota. App de la señal Unipolar Positiva NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

Figura 3

Señal NRZ Unipolar Positiva



Nota. App de la señal Unipolar Positiva NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

- Permite ingresar manualmente una secuencia binaria (0s y 1s).
- *Ejemplo de entrada:* 1011001.

2. Frecuencia de muestreo ("fs")

- Controla la cantidad de muestras por segundo en la representación de la señal.
- *Valores recomendados:* 100 - 1000 Hz.
- Afecta la resolución de la gráfica (más fs, más suavidad).

3. Duración total de la señal

- Define cuánto tiempo durará la señal generada en segundos.
- Se calcula dinámicamente con la cantidad de bits.

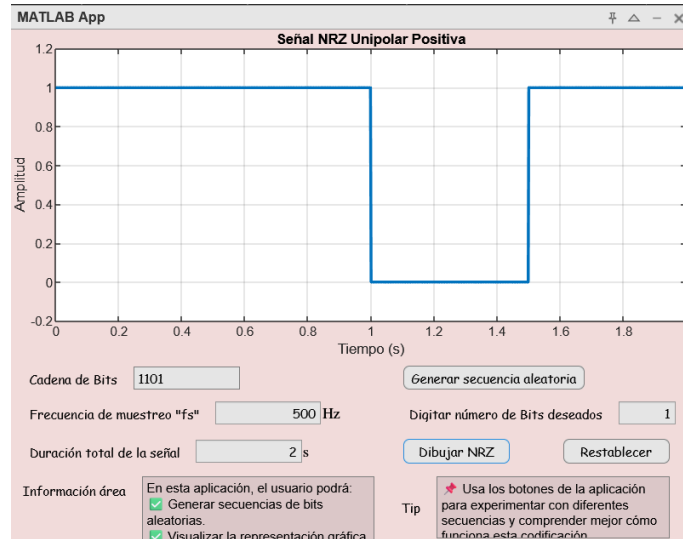
4. Generar secuencia aleatoria

- Permite generar automáticamente una secuencia aleatoria de bits.
- Requiere definir cuántos bits se desean generar.

5. Dibujar NRZ

Figura 4

Señal NRZ Unipolar Positiva



Nota. App de la señal Unipolar Positiva NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

- Grafica la señal en base a la secuencia ingresada o generada.

6. Restablecer

- Borra la secuencia y devuelve los valores a su estado original.

7. Área de Información

- Explica cómo se representan los valores 1 y 0 en la señal.
- Consejos para experimentar con diferentes parámetros.

Comencemos a probar y analizar la señal con la variación de los diferentes parámetros establecidos por la aplicación.

Ejemplo 1: Aumentando la frecuencia de muestreo

Parámetros:

- **Cadena de Bits:** 1011001
- **fs:** 100 Hz (baja resolución)
- **Duración:** 3 s

Observación: La señal puede mostrarse con bordes menos definidos; esto podemos notarlo en la **Figura 2**.

Cambio:

		<div>➤ Se aumenta f_s a 1000 Hz para mejorar la resolución.</div> <div>➤ La gráfica será más suave y precisa.</div> <div>Efecto: Al aumentar la frecuencia de muestreo (f_s) de 100 Hz a 1000 Hz, la señal se representa con mayor exactitud. En la gráfica, esto se refleja en transiciones más suaves y menos escalonadas en los cambios de bit, mejorando la resolución y la interpretación visual de la señal.</div> <div>Ejemplo 2: Reducción del tiempo de duración</div> <div>Parámetros:<ul style="list-style-type: none">• Cadena de Bits: 1101• f_s: 500 Hz• Duración: 4 s (originalmente); esto es visible en la Figura 3.</div> <div>Cambio:<div>➤ Se reduce la duración a 2 s.</div><div>➤ La misma secuencia ocupará menos tiempo total en la gráfica.</div></div> <div>Efecto: Al reducir la duración de la señal de 4 s a 2 s, los mismos bits deben representarse en menos tiempo, lo que se traduce en que cada bit ocupe</div>
--	--	---

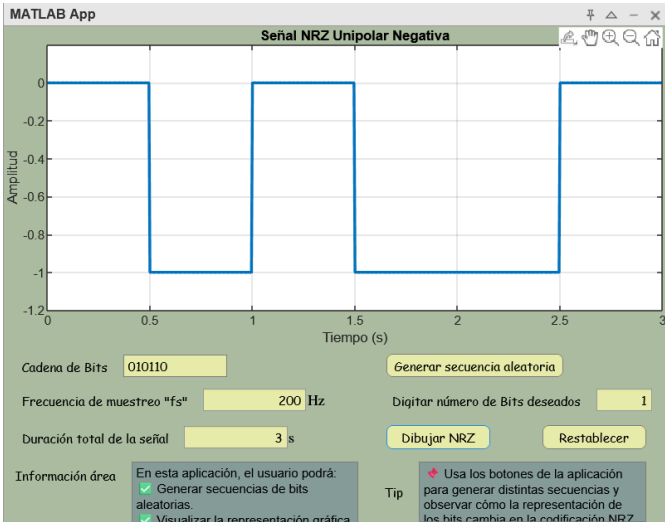
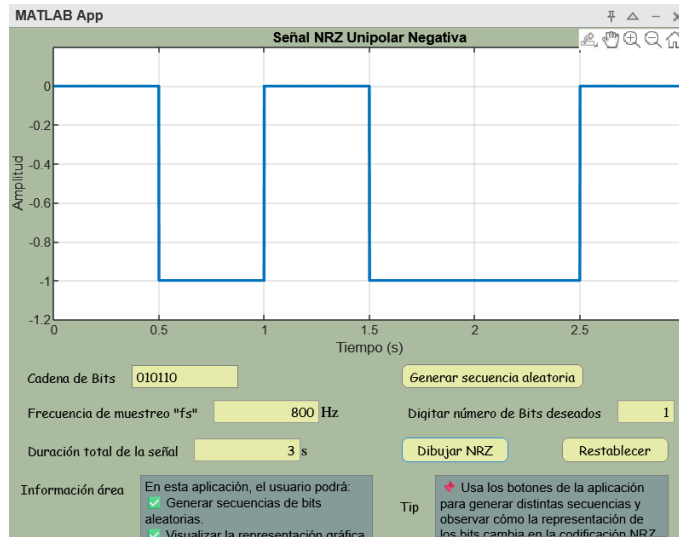
		<p>un intervalo más corto. Como resultado, la señal se verá más estrecha en el eje del tiempo, incrementando considerablemente la velocidad de cambio entre niveles; véase la Figura 4.</p> <h3>Apreciaciones finales</h3> <p>El esquema NRZ Unipolar Positiva representa los '1' con un nivel alto de voltaje y los '0' con un nivel bajo. Es un método sencillo de codificación, pero puede implicar problemas con sincronización si hay muchas ocurrencias de un mismo bit. Ajustando f_s y la duración, podemos mejorar significativamente la representación y la comprensión de la señal.</p>
Unipolar Negativa NRZ [10p]	<p>Figura 5 <i>Señal NRZ Unipolar negativa</i></p>  <p>Nota. App de la señal Unipolar Negativa NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.</p>	<h3>Descripción de la aplicación</h3> <p>Esta aplicación permite visualizar la señal NRZ Unipolar Negativa a partir de una secuencia de bits ingresada por el usuario. Se han incorporado campos de entrada de datos (<i>Numeric & Text</i>) para modificar parámetros clave como la cadena de bits, la frecuencia de muestreo y la duración total de la señal; véase la Figura 5.</p> <p>También se añadieron botones para:</p>

Figura 6

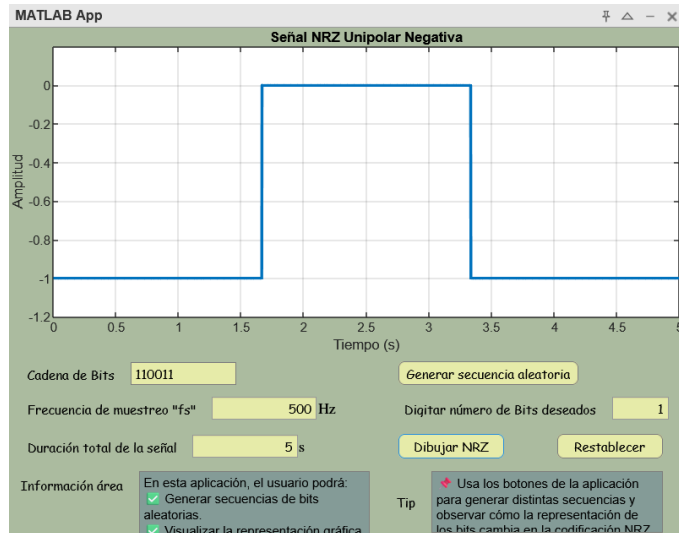
Señal NRZ Unipolar Negativa



Nota. App de la señal Unipolar Negativa NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

Figura 7

Señal NRZ Unipolar Negativa



Nota. App de la señal Unipolar Negativa NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

- ✓ Generar el gráfico.
- ✓ Crear una cadena de bits aleatoria.
- ✓ Restablecer todos los valores a su estado inicial (*default*).

Además, como complemento para brindar un acercamiento, se incluyen dos áreas de texto con información breve sobre la codificación **NRZ Unipolar Negativa** y recomendaciones para su visualización en la aplicación.

Componentes de la interfaz

1. Cadena de Bits

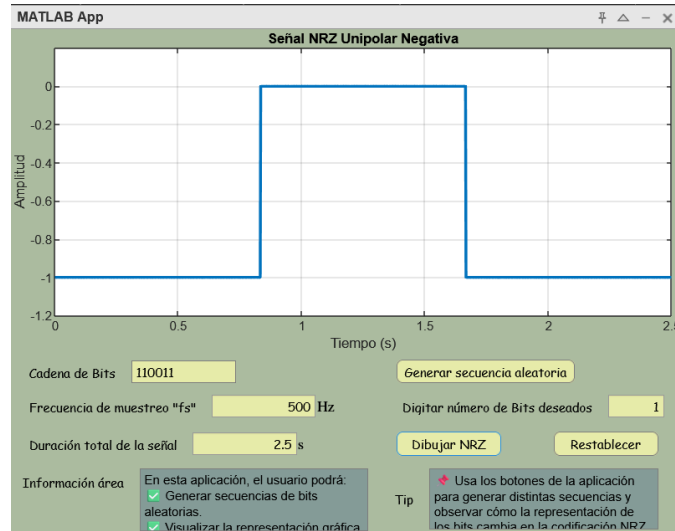
- Permite ingresar manualmente una secuencia binaria (0s y 1s).
- *Ejemplo de entrada:*
010110

2. Frecuencia de muestreo

- Controla la cantidad de muestras por segundo en la representación de la señal.
- *Valores recomendados:* 100 - 1000 Hz.
- Afecta la resolución de la gráfica (mayor $f_s \rightarrow$ mayor suavidad).

Figura 8

Señal NRZ Unipolar Negativa



Nota. App de la señal Unipolar Negativa NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

3. Duración total de la señal

- Define cuánto tiempo durará la señal generada en segundos.
- Se calcula dinámicamente con base en la cantidad de bits.

4. Generar secuencia aleatoria

- Genera automáticamente una secuencia aleatoria de bits.
- Requiere definir cuántos bits se desean generar.

5. Dibujar NRZ

- Grafica la señal en función de la secuencia ingresada o generada.

6. Restablecer

- Borra la secuencia y devuelve los valores a su estado original.

7. Área de información

- Explica cómo se representan los valores 1 y 0 en la señal.
- Consejos para experimentar con diferentes parámetros.

Ejemplo 1: Comparación entre frecuencias de muestreo

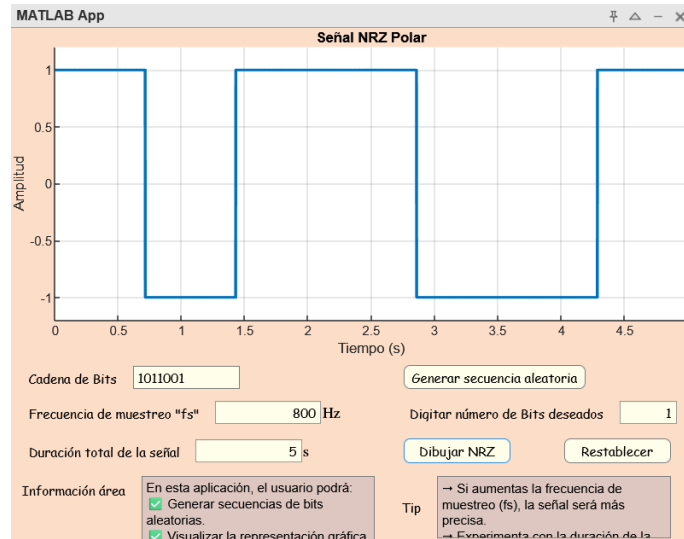
		<p>Parámetros:</p> <ul style="list-style-type: none">• Cadena de Bits: 010110• fs: 200 Hz (baja resolución)• Duración: 3 s <p>Observación:</p> <p>A baja frecuencia de muestreo, la señal puede mostrarse con bordes menos definidos y pequeños escalones en las transiciones; esto lo podemos observar en la Figura 5.</p> <p>Cambio:</p> <p>➤ Se aumenta f_s a 800 Hz para mejorar la resolución.</p> <p>Efecto:</p> <p>Con una mayor frecuencia de muestreo (f_s de 200 Hz a 800 Hz), la señal se representa con mayor precisión, mostrando transiciones más claras y reduciendo posibles obstrucciones de muestreo; véase la Figura 6.</p> <p><i>Ejemplo 2: Cambio en la duración de la señal</i></p> <p>Parámetros:</p> <ul style="list-style-type: none">• Cadena de Bits: 110011• fs: 500 Hz• Duración: 5 s (originalmente); véase en la Figura 7.
--	--	--

		<p>Cambio:</p> <p>➤ Se reduce a 2.5 s.</p> <p>Efecto:</p> <p>Disminuir la duración total de la señal hace que cada bit ocupe menos tiempo en la gráfica. Esto genera una representación más consistente y rápida en términos de cambios de escala o nivel; esto lo podemos observar en la Figura 8, lo que puede afectar cómo percibimos la estabilidad de la señal.</p> <p>Apreciaciones finales y comparación con NRZ Unipolar Positiva</p> <p>La codificación NRZ Unipolar Negativa representa:</p> <ul style="list-style-type: none">• Bit "1" → Nivel de voltaje bajo (-V).• Bit "0" → Nivel de voltaje neutro o en 0V. <p>En comparación con NRZ Unipolar Positiva, ambas codificaciones comparten características fundamentales:</p> <ul style="list-style-type: none">✓ Son codificaciones sencillas de implementar.✓ Mantienen el mismo nivel de voltaje durante la duración del bit.
--	--	---

		<div>✓ Presentan problemas de sincronización si hay muchas ocurrencias consecutivas de un mismo bit.</div> <div>Sin embargo, hay una diferencia importante:</div> <div><ul style="list-style-type: none">• NRZ Unipolar Positiva representa el bit 1 con voltaje positivo y el 0 con 0V.• NRZ Unipolar Negativa invierte esta lógica, asignando el 1 a un voltaje negativo y el 0 a 0V.</div> <div>Esta característica influye en el diseño de circuitos receptores y en la interpretación de la señal.</div>
<div>Polar NRZ</div> <div>[10p]</div>	<div><div>Figura 9</div><div>Señal Polar NRZ</div><div></div><div>Nota. App de la señal Polar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.</div></div>	<div>Descripción de la aplicación</div> <div>Esta aplicación permite visualizar la señal NRZ Polar a partir de una secuencia de bits ingresada por el usuario. Se han añadido campos de ingreso de datos (<i>Numeric & Text</i>) para modificar parámetros como la cadena de bits, la frecuencia de muestreo y la duración total de la señal.</div> <div>Asimismo, se incluyen botones para la generación del gráfico, la creación de una secuencia de bits aleatoria y un botón para restablecer todos los campos a su estado base (<i>default</i>).</div>

Figura 10

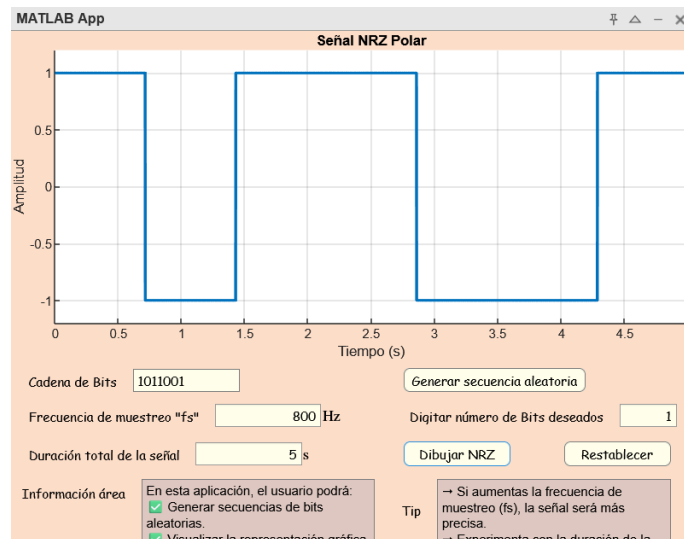
Señal Polar NRZ



Nota. App de la señal Polar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

Figura 11

Señal Polar NRZ



Nota. App de la señal Polar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

Además, se agregaron dos áreas de texto con información relevante y consejos sobre el funcionamiento del programa y su aplicación en la codificación digital; véase la **Figura 9**. ¡A probar!

Componentes de la interfaz

1. Cadena de Bits

- Permite ingresar manualmente una secuencia binaria (0s y 1s).
- *Ejemplo de entrada:* 1011001.

2. Frecuencia de muestreo ("fs")

- Define la cantidad de muestras por segundo en la representación de la señal.
- *Valores recomendados:* 100 - 1000 Hz.
- Afecta la resolución del gráfico (mayor fs = mayor precisión).

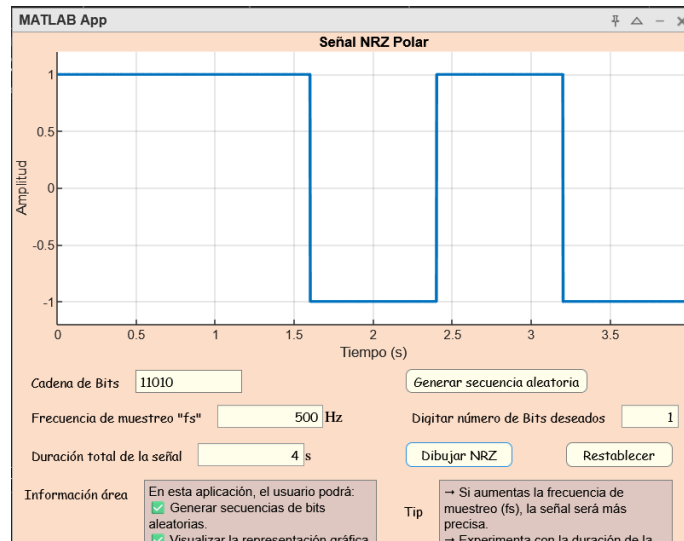
3. Duración total de la señal

- Determina el tiempo total en que se representará la señal.
- Se calcula dinámicamente en función de la cantidad de bits y el *bit rate*.

4. Generar secuencia aleatoria

Figura 12

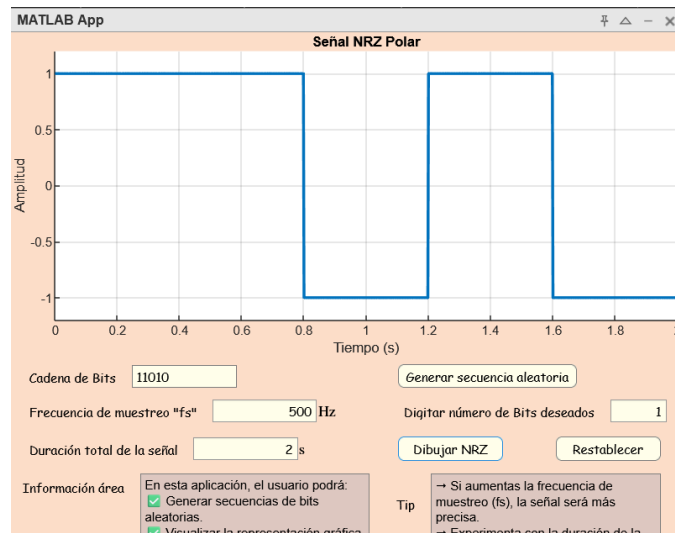
Señal Polar NRZ



Nota. App de la señal Polar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

Figura 13

Señal Polar NRZ



Nota. App de la señal Polar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

- Genera automáticamente una secuencia binaria aleatoria.
- Se puede definir cuántos bits se desean generar.

5. Dibujar NRZ Polar

- Grafica la señal de acuerdo con la secuencia ingresada o generada.

6. Restablecer

- Borra la secuencia ingresada y restablece los valores predeterminados.

7. Área de información

- Explica cómo se representan los valores 1 y 0 en NRZ Polar.
- Incluye consejos sobre cómo ajustar los parámetros para obtener una mejor visualización.

Ejemplo 1: Variación de la frecuencia de muestreo

Parámetros iniciales:

- **Cadena de Bits:** 1011001
- **fs:** 200 Hz
- **Duración:** 5 s

		<p>Observación:</p> <p>A baja frecuencia de muestreo, la señal tiene bordes menos definidos debido a la baja resolución; esto lo podemos apreciar en la Figura 10.</p> <p>Cambio:</p> <ul style="list-style-type: none">➤ Se aumenta f_s a 800 Hz para mejorar la resolución. <p>Efecto:</p> <p>Con una mayor frecuencia de muestreo, la transición entre niveles se representa con más detalle, permitiendo una mejor visualización de la señal; véase la Figura 11.</p> <p><i>Ejemplo 2: Cambio en la duración de la señal</i></p> <p>Parámetros iniciales:</p> <ul style="list-style-type: none">• Cadena de Bits: 11010• f_s: 500 Hz• Duración: 4 s <p>Estos parámetros están presentes en la Figura 12.</p> <p>Cambio:</p> <ul style="list-style-type: none">➤ Se reduce la duración a 2 s. <p>Efecto:</p> <p>Al reducir la duración, la señal se compacta en el eje del tiempo y los cambios entre niveles ocurren más rápido, mostrando una representación más comprimida de la señal. Lo</p>
--	--	---

		<p>anterior se puede apreciar mejor en la Figura 13.</p> <p>Apreciaciones finales y comparación con NRZ Unipolar</p> <p>El esquema NRZ Polar asigna:</p> <ul style="list-style-type: none">• +V para el bit "1".• -V para el bit "0". <p>Esto tiene la maravillosa cualidad de mantener transiciones de voltaje más exactas en comparación con NRZ Unipolar, lo que mejora la sincronización y reduce la componente de corriente continua (<i>DC</i>) en la señal.</p> <p>Comparación con NRZ Unipolar:</p> <p>NRZ Unipolar: Usa 0V y +V o 0V y -V, lo que genera un desbalance de voltaje que puede causar problemas en transmisiones largas.</p> <p>NRZ Polar: Utiliza +V y -V, lo que regula la señal y facilita su detección en el receptor.</p> <p>Eficiencia: NRZ Polar reduce la componente de corriente continua (<i>DC</i>), lo que mejora la calidad de transmisión y evita el uso de circuitos adicionales.</p>
--	--	--

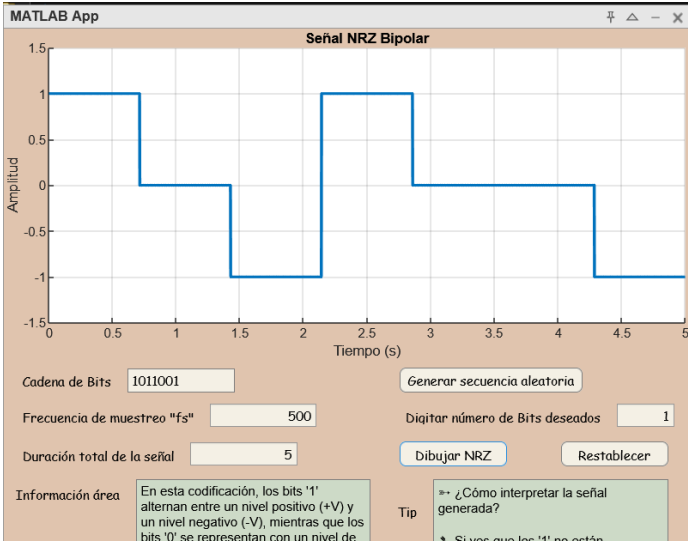
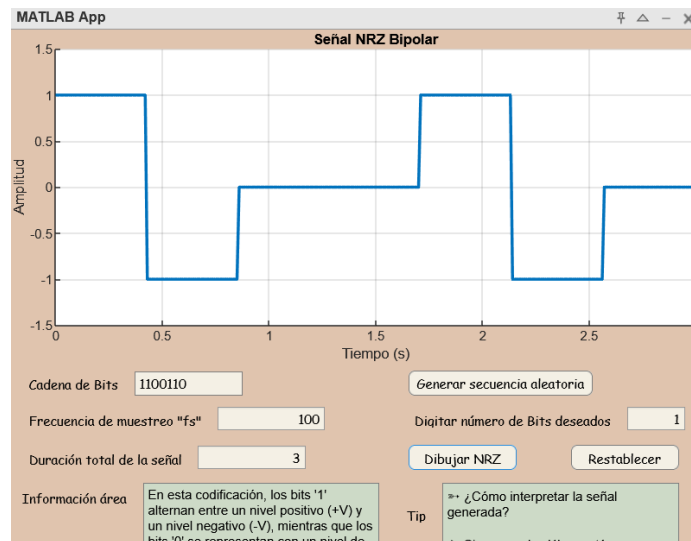
		<p>Sincronización: En NRZ Unipolar, secuencias largas de ceros pueden hacer que la señal pierda sincronización, mientras que NRZ Polar posee mayor variación de voltaje, lo que facilita su detección.</p> <p>Finalmente, ajustando <i>fs</i> y la duración de la señal, podemos mejorar su representación y análisis.</p>
<p>Bipolar NRZ</p> <p>[10p]</p>	<p>Figura 14</p> <p><i>Señal Bipolar NRZ</i></p>  <p>Nota. App de la señal Bipolar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.</p>	<p>Descripción de la Aplicación</p> <p>Esta aplicación permite visualizar la señal NRZ Bipolar a partir de una secuencia de bits ingresada por el usuario. Tiene campos para ingresar datos como la cadena de bits, la frecuencia de muestreo y la duración total de la señal. También se incluyen botones para generar el gráfico, crear una cadena aleatoria de bits y restablecer los valores a sus configuraciones predeterminadas.</p> <p>Adicionalmente, la aplicación proporciona un área de información donde se explica brevemente la codificación NRZ Bipolar y sus características, lo cual se puede observar con mayor detalle en la Figura 14.</p> <p>Componentes de la Interfaz</p> <p>1. Cadena de Bits</p>

Figura 15

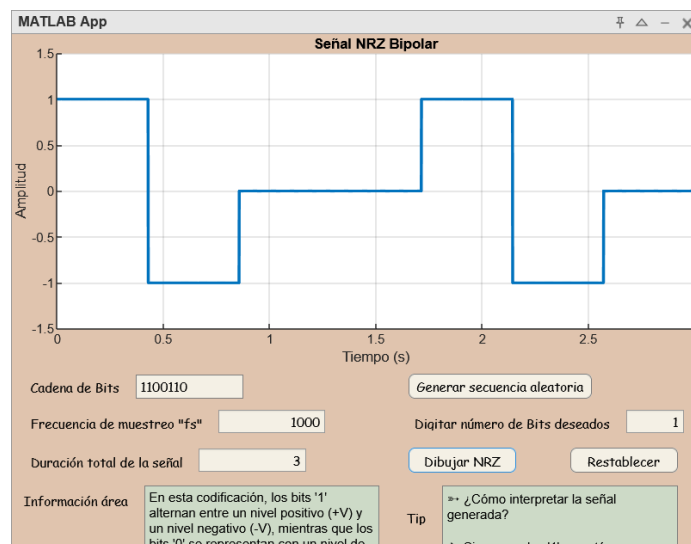
Señal Bipolar NRZ



Nota. App de la señal Bipolar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

Figura 16

Señal Bipolar NRZ



Nota. App de la señal Bipolar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

- Permite ingresar una secuencia binaria (0s y 1s).
- *Ejemplo de entrada:* 1011001.

2. Frecuencia de muestreo ("fs")

- Controla la cantidad de muestras por segundo en la representación de la señal.
- *Valores recomendados:* 100 - 1000 Hz.
- Afecta la resolución del gráfico (mayor f_s = mejor definición).

3. Duración total de la señal

- Define la duración total de la señal en segundos.
- Se ajusta dinámicamente según la cantidad de bits.

4. Generar secuencia aleatoria

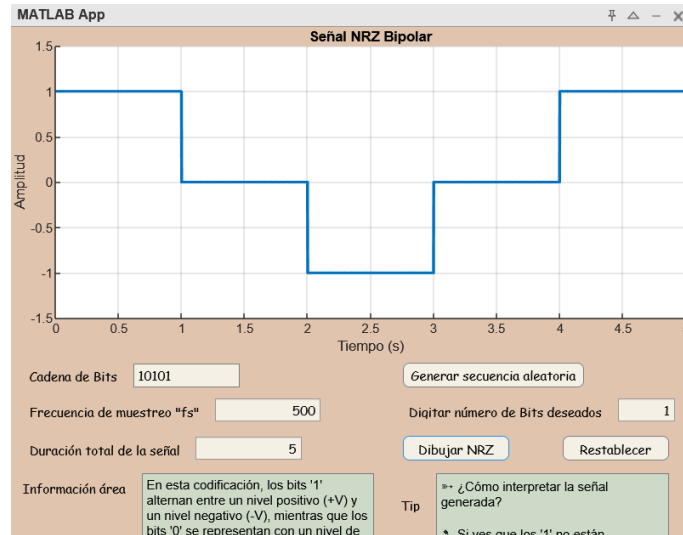
- Genera automáticamente una secuencia aleatoria de bits de longitud definida por el usuario.

5. Dibujar NRZ Bipolar

- Grafica la señal codificada a partir de la secuencia ingresada o generada.

Figura 17

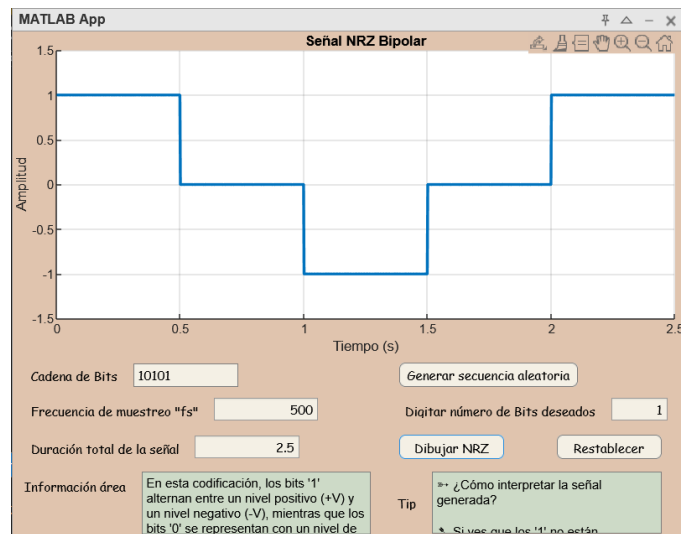
Señal Bipolar NRZ



Nota. App de la señal Bipolar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

Figura 18

Señal Bipolar NRZ



Nota. App de la señal Bipolar NRZ elaborado en MATLAB con la ayuda de la herramienta App Designer. Elaboración propia.

6. Restablecer

- Borra la secuencia de bits y devuelve los valores a sus configuraciones iniciales.

7. Área de Información

- Explica cómo se representan los bits en la señal **NRZ Bipolar**.
- Muestra consejos para experimentar con los parámetros y mejorar la visualización.

Ejemplo 1: Efecto del aumento de la frecuencia de muestreo

Parámetros iniciales:

- **Cadena de Bits:** 1100110
- **Frecuencia de Muestreo (fs):** 100 Hz (*baja resolución*)
- **Duración Total:** 3 s

Observación:

A bajas frecuencias de muestreo, la señal puede mostrar transiciones con menor precisión, observándose bordes escalonados; véase la **Figura 15**.

Cambio:

- Se aumenta f_s a 1000 Hz para mejorar la resolución.

Efecto:

La señal se representa con mayor precisión y transiciones más suaves, lo que mejora la visualización de la alternancia de los 1s y la estabilidad de los 0s. Esto se puede ver reflejado en la **Figura 16**.

Ejemplo 2: Reducción del tiempo de duración

Parámetros iniciales:

- **Cadena de Bits:** 10101
- **Frecuencia de Muestreo (f_s):** 500 Hz
- **Duración Total:** 5 s

Nótese en la **Figura 17**.

Cambio:

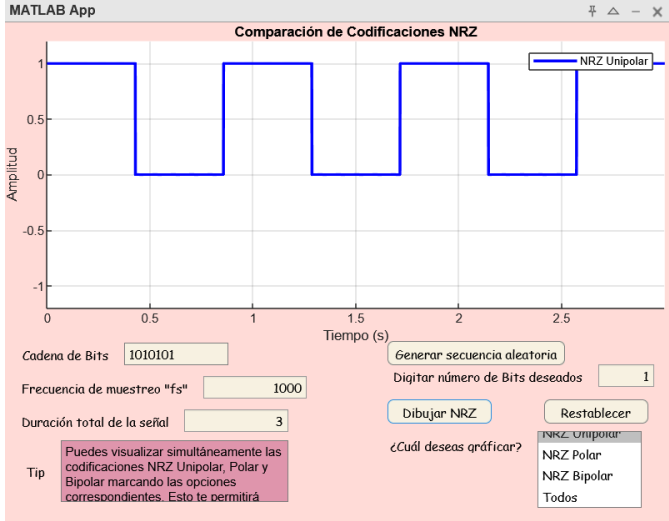
- Se reduce la duración total a 2.5 s.

Efecto:

Cada bit ocupa menos tiempo en la gráfica, lo que provoca que la señal se vea más compacta y las transiciones sean más rápidas; los cambios pueden apreciarse en la **Figura 18**.

Apreciaciones finales

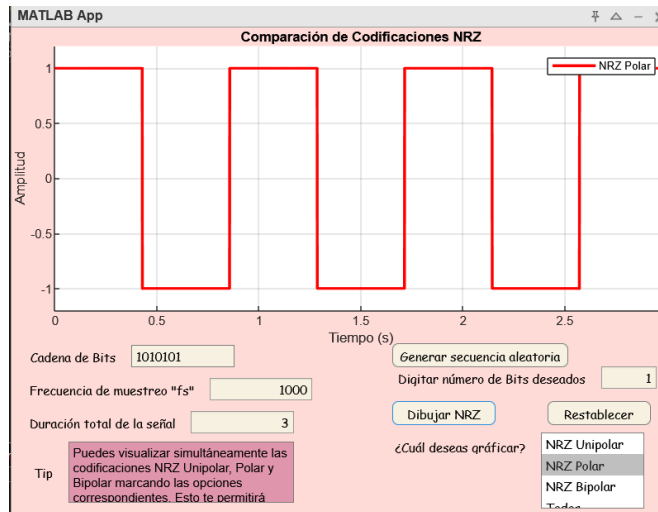
El esquema **NRZ Bipolar** alterna la polaridad de los 1s entre +V y -V, mientras que los 0s se mantienen en

		<p>0V. Esto disminuye notoriamente el obstáculo DC presente en las codificaciones Unipolar y Polar, mejorando la sincronización y transmisión de la señal.</p> <p><i>Comparación con NRZ Unipolar y NRZ Polar</i></p> <ul style="list-style-type: none">🚦 NRZ Unipolar: Usa 0V para 0s y +V para 1s, lo que genera un sesgo DC.🚦 NRZ Polar: Asigna +V a 1s y -V a 0s, eliminando parte del sesgo DC.🚦 NRZ Bipolar: Alterna los 1s entre +V y -V, minimizando aún más la componente DC y mejorando la eficiencia en la transmisión. <p>Esta codificación es la preferida en sistemas donde es fundamental una transmisión estable y con mejor sincronización.</p>
ADICIONAL NRZ	<p>Figura 19</p> <p><i>Comparaciones de codificaciones NRZ</i></p>  <p>Nota. App elaborada para la visualización de las 3 señales</p>	<p>Descripción de la Aplicación</p> <p>Esta aplicación permite visualizar y comparar distintas codificaciones de línea NRZ (Non-Return to Zero) a partir de una secuencia binaria ingresada por el usuario. Se han implementado tres variantes de NRZ:</p> <ul style="list-style-type: none">🚦 NRZ Unipolar: Representa el bit 1 con un nivel alto (+V)

de acuerdo a lo ingresado por el usuario en MATLAB con la ayuda de la herramienta App Designer; aquí estamos viendo NRZ Unipolar. Elaboración propia.

Figura 20

Comparaciones de codificaciones NRZ



Nota. App elaborada para la visualización de las 3 señales de acuerdo a lo ingresado por el usuario en MATLAB con la ayuda de la herramienta App Designer; aquí estamos viendo NRZ Polar. Elaboración propia.

y el bit 0 con un nivel bajo (0V).

NRZ Polar: Representa el bit 1 con un nivel positivo (+V) y el bit 0 con un nivel negativo (-V).

NRZ Bipolar: Representa los bits 1 alternando entre +V y -V, mientras que los bits 0 se mantienen en 0V.

El usuario puede seleccionar una o todas las codificaciones desde un **ListBox**, para visualizar todas en un solo gráfico y compararlas; véase la **Figura 19**.

Interfaz de usuario

1. Entrada de Bits

- Campo de texto donde el usuario ingresa una secuencia binaria.
- *Ejemplo de entrada:* 1011001.

2. Frecuencia de muestreo (Fs)

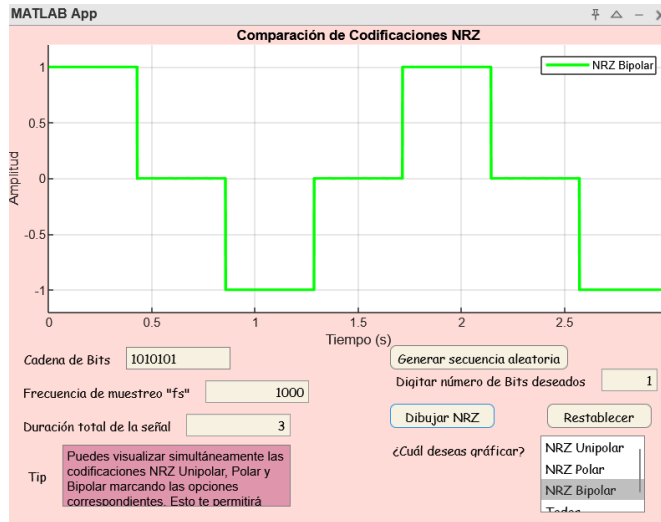
- Control deslizante o campo numérico que permite modificar la cantidad de muestras por segundo.

3. Duración de la señal

- Campo para definir cuánto tiempo ocupará la secuencia en segundos.

Figura 21

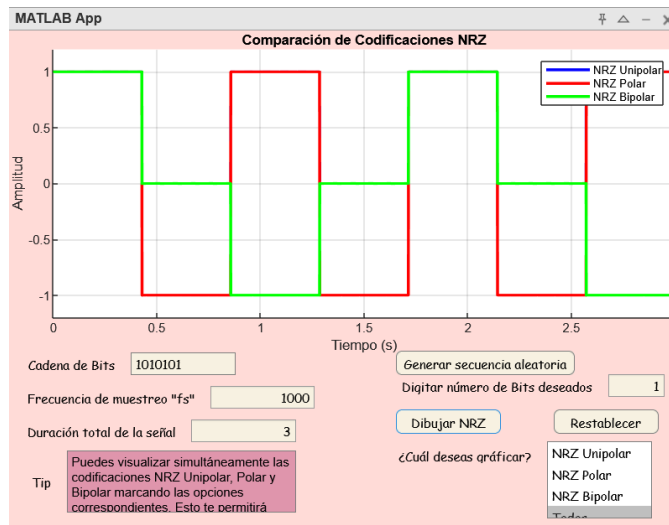
Comparaciones de codificaciones NRZ



Nota. App elaborada para la visualización de las 3 señales de acuerdo a lo ingresado por el usuario en MATLAB con la ayuda de la herramienta App Designer; aquí estamos viendo NRZ Bipolar. Elaboración propia.

Figura 22

Comparaciones de codificaciones NRZ



Nota. App elaborada para la visualización de las 3 señales de acuerdo a lo ingresado por el usuario en MATLAB con la ayuda de la herramienta App Designer; aquí estamos viendo todas. Elaboración propia.

4. Selección de codificación

- Un **ListBox** con las opciones **NRZ Unipolar, NRZ Polar, NRZ Bipolar y Todos**.
- Permite múltiples selecciones.

5. Botón "graficar"

- Genera la representación de la señal en función de la codificación seleccionada.

6. Área de gráficos

- Muestra la representación de la señal según la codificación elegida, con colores diferenciados para cada tipo de **NRZ**.

Apreciaciones finales

Comparación visual

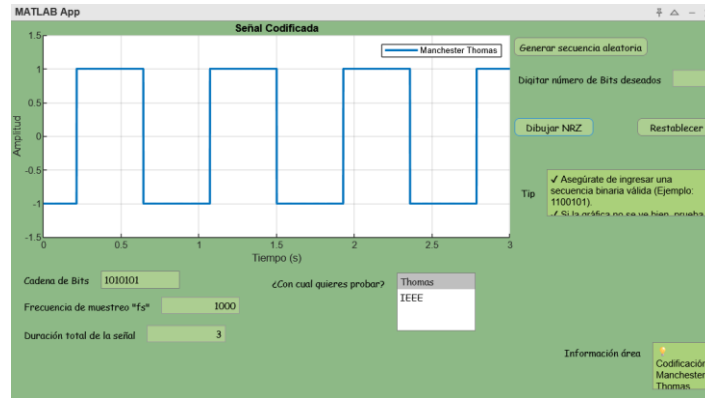
La aplicación permite visualizar y analizar cómo varía la representación de la señal según el tipo de codificación **NRZ**.

Diferencias entre las codificaciones

- 🚦 **NRZ Unipolar:** Solo usa valores positivos y **0V**, lo que introduce una componente de

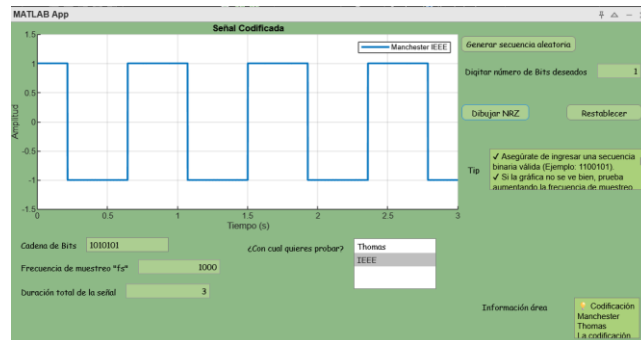
		<p>corriente continua; véase la Figura 19.</p> <p>✚ NRZ Polar: Usa valores positivos y negativos, eliminando la componente de corriente continua; véase la Figura 20.</p> <p>✚ NRZ Bipolar: Alterna la polaridad de los bits 1, mejorando la detección de errores y reduciendo la DC; véase la Figura 21.</p> <p>✚ Finalmente, en la Figura 22, obtenemos una unificación de las tres gráficas juntas.</p> <p><i>¡Increíble, no crees!</i></p> <p>Personalización</p> <p>El usuario puede modificar la frecuencia de muestreo, la duración de la señal y elegir qué codificaciones comparar, lo que es útil para ofrecer una gran variabilidad para el análisis de señales digitales.</p>
MANCHESTER [10p]		<p>Descripción de la Aplicación</p> <p>Esta aplicación permite visualizar las señales codificadas mediante los esquemas Manchester Thomas e IEEE a partir de una secuencia de bits ingresada por el usuario. Se han añadido campos para ingresar datos como:</p>

Figura 23
MANCHESTER



Nota. Aplicación elaborada para la visualización de las señales Manchester Thomas y Manchester IEEE, de acuerdo con lo ingresado por el usuario en MATLAB, con la ayuda de la herramienta App Designer. Elaboración propia.

Figura 24
MANCHESTER



Nota. Aplicación elaborada para la visualización de las señales Manchester Thomas y Manchester IEEE, de acuerdo con lo ingresado por el usuario en MATLAB, con la ayuda de la herramienta **App Designer**. Elaboración propia.

- ✓ Cadena de bits
- ✓ Frecuencia de muestreo
- ✓ Duración total de la señal

También se incluyen botones para generar el gráfico, crear una secuencia aleatoria de bits y restablecer los valores a sus configuraciones predeterminadas; véase la **Figura 23**.

Adicionalmente, la aplicación cuenta con un área de información donde se explica brevemente la **codificación Manchester** y sus diferencias entre los estándares **Thomas** e **IEEE**; véase la **Figura 24**, que corresponde al estándar **IEEE**.

Componentes de la Interfaz

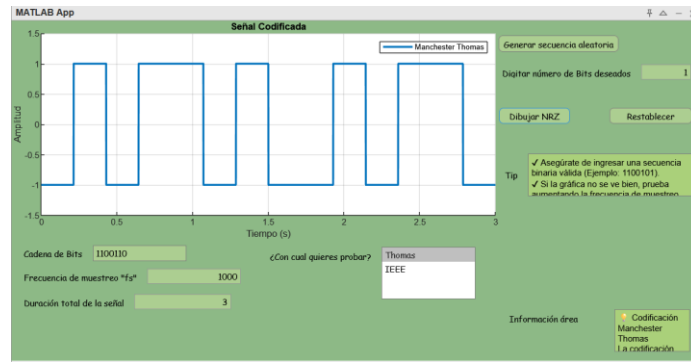
1. Cadena de Bits

- Permite ingresar una secuencia binaria (0s y 1s).
- *Ejemplo de entrada:* 1010101.

2. Frecuencia de muestreo (fs)

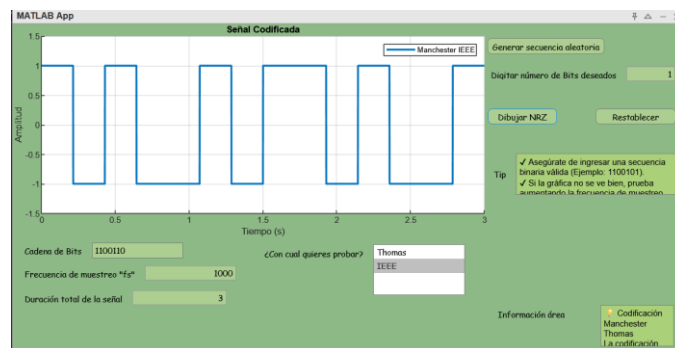
- Controla la cantidad de muestras por segundo en la representación de la señal.
- Valores recomendados: **100 - 5000 Hz**.
- Afecta la resolución del gráfico (a mayor fs, mejor definición).

Figura 25
MANCHESTER



Nota. Probando la señal Manchester Thomas con diferentes parámetros para la visualización de la señal en MATLAB, con la ayuda de la herramienta **App Designer**. Elaboración propia.

Figura 26
MANCHESTER



Nota. Probando la señal Manchester IEEE con diferentes parámetros para la visualización de la señal en MATLAB, con la ayuda de la herramienta **App Designer**. Elaboración propia.

3. Duración total de la señal

- Define la duración total de la señal en segundos.
- Se ajusta dinámicamente según la cantidad de bits ingresados.

4. Generar secuencia aleatoria

- Genera automáticamente una secuencia de bits de longitud definida por el usuario.

5. Dibujar Manchester Thomas / IEEE

- Grafica la señal codificada según el método seleccionado en la lista.

6. Restablecer

- Borra la secuencia de bits y devuelve los valores a sus configuraciones iniciales.

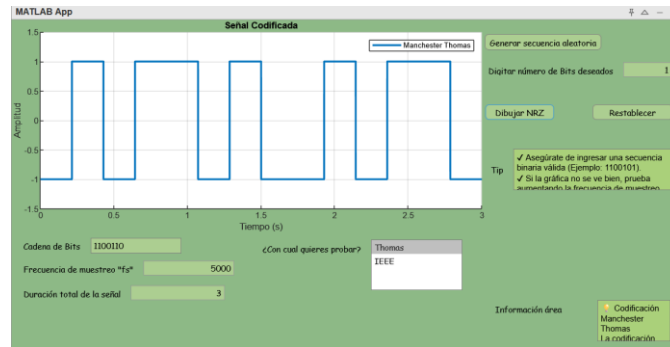
7. Área de información

- Explica cómo se representan los bits en la codificación **Manchester**.
- Describe las diferencias entre los estándares **Thomas** e **IEEE**.

Ejemplo: Efecto del aumento de la frecuencia de muestreo

Figura 27

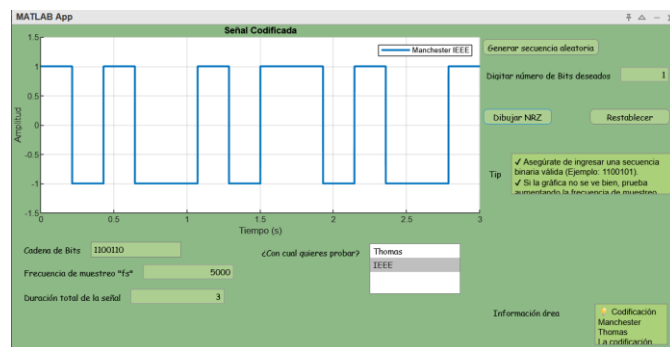
MANCHESTER



Nota. Probando la señal Manchester Thomas con diferentes parámetros para la visualización de la señal en MATLAB, con la ayuda de la herramienta **App Designer**. Elaboración propia.

Figura 28

MANCHESTER



Nota. Probando la señal Manchester IEEE con diferentes parámetros para la visualización de la señal en MATLAB, con la ayuda de la herramienta **App Designer**. Elaboración propia.

Parámetros iniciales:

- **Cadena de Bits:** 1100110
- **Frecuencia de Muestreo (fs):** 1000 Hz (*baja resolución*)
- **Duración Total:** 3 s
- **Codificación seleccionada:** Manchester Thomas y Manchester IEEE

Observaciones:

Al seleccionar ambas codificaciones, se observa en la gráfica que:



- ✚ **Manchester Thomas:** Representa el bit 0 con una transición de alto a bajo, y el bit 1 con una transición de bajo a alto; véase la **Figura 25**.
- ✚ **Manchester IEEE:** Hace lo contrario: el bit 0 tiene una transición de bajo a alto, y el bit 1 una transición de alto a bajo; véase la **Figura 26**.

Cambio:

- Se aumenta fs a **5000 Hz** para mejorar la resolución.

Efecto:

- ❖ La señal se representa con mayor precisión.

		<div>❖ Se mejora la visualización del cambio de estado en cada bit.</div> <div>❖ Se permite apreciar mejor las diferencias entre Manchester Thomas e IEEE.</div> <div>Este contraste es clave en la interpretación de la señal, ya que cada estándar es utilizado en distintos sistemas de comunicación; véanse las Figuras 27 y 28.</div> <div>Apreciaciones finales</div> <div><i>Comparación entre Thomas e IEEE</i></div> <div><div> Manchester Thomas:</div><div><ul style="list-style-type: none">• 0 → transición de alto a bajo.• 1 → transición de bajo a alto.</div><div><div> Manchester IEEE:</div><div><ul style="list-style-type: none">• Utiliza la transición inversa respecto a Thomas.</div></div><div>Ambos métodos mejoran la sincronización y reducen la componente DC presente en las codificaciones NRZ.</div><div>Finalmente, es importante recordar que la codificación Manchester Thomas se diferencia de la Manchester estándar (IEEE) porque no depende del valor absoluto del bit,</div></div>
--	--	--

		sino de la transición respecto al bit anterior.
--	--	---

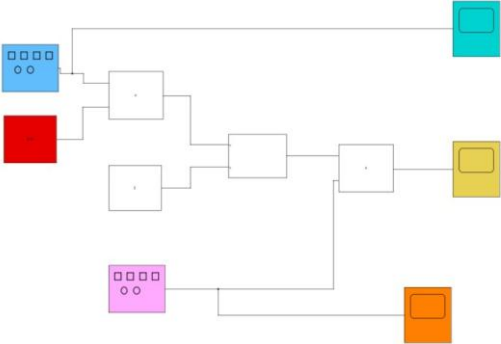
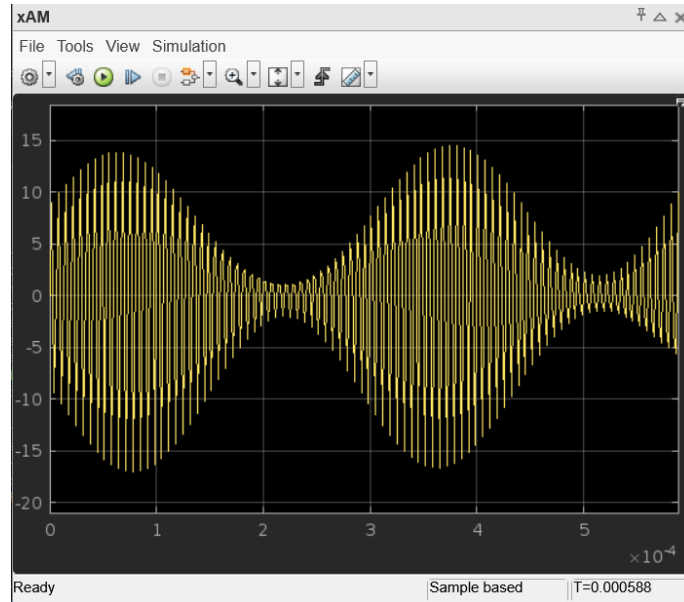
[50 point] Simulink		
INTERFAZ	SCREENSHOT	EXPLICACIÓN
<div>AM</div> <div>[10p]</div>	<div> <div>Figura 29</div> <div>AM</div>  <div> Nota. Señal modulada AM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta Simulink. Elaboración propia. </div> </div>	<div> <div>Descripción de la Aplicación</div> <p>Esta aplicación permite visualizar la modulación en amplitud (AM) utilizando Simulink. Se emplean señales senoidales para representar la señal portadora y la señal moduladora, generando una señal modulada en amplitud que se observa en el Scope.</p> <p>Se han añadido componentes para definir los siguientes parámetros:</p> <ul style="list-style-type: none"> • Señal moduladora (frecuencia y amplitud). • Señal portadora (frecuencia y amplitud). • Índice de modulación (profundidad de la modulación); en la Figura 29 es la de color rojo. • Visualización de señales en distintas etapas del proceso. <p>En la Figura 29, las diferentes salidas están contenidas en un Scope. Distribuidas de la siguiente manera: Cyan para $x(t)$, naranja para la salida del</p> </div>

Figura 30

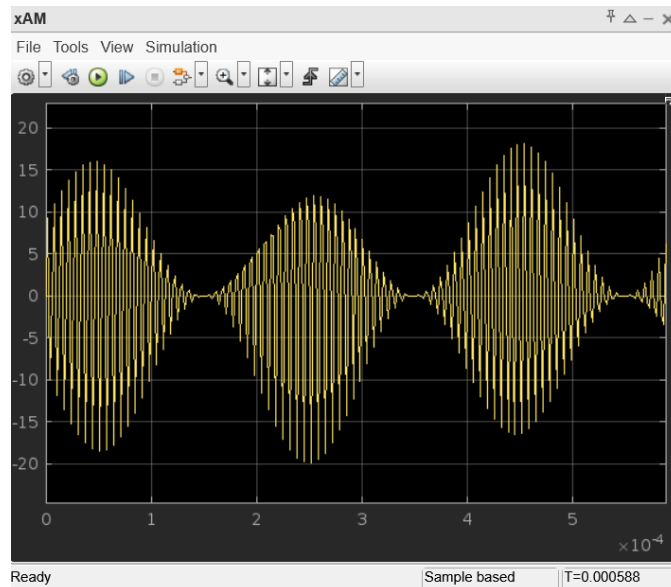
AM



Nota. Señal modulada AM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

Figura 31

AM



Nota. Señal modulada AM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

Carrier, y por último, el Scope de color amarillo muestra la señal AM. Además, se incluyen visualizaciones separadas de la señal portadora, la señal moduladora y la señal modulada, permitiendo ver su comportamiento en cada fase; véase la **Figura 29**.

Componentes del Modelo en Simulink

1. Señal portadora

- Es una onda senoidal de alta frecuencia utilizada para transportar la información.
- Definida en Simulink mediante un bloque *Sine Wave*.
- Su ecuación es:

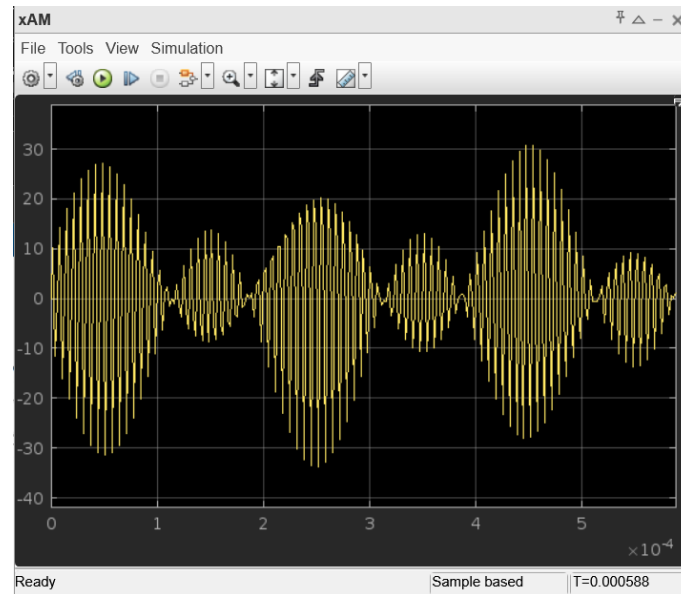
$$C(t) = A_c \cos(2\pi f_c t)$$

- Parámetros configurables:
 - **Frecuencia de la portadora** (f_c)
 - **Amplitud de la portadora** (A_c)

2. Señal moduladora

Figura 32

AM



Nota. Señal modulada AM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

- Es la señal que contiene la información a transmitir.
- También se genera con un bloque *Sine Wave*.
- Su ecuación es:

$$M(t) = A_m \cos(2\pi f_m t)$$

- Parámetros configurables:
 - **Frecuencia de la señal moduladora** (f_m)
 - **Amplitud de la señal moduladora** (A_m)

3. Índice de modulación

- Se controla con un bloque *Gain*, que ajusta la amplitud de la señal moduladora.
- Definido como:

$$\mu = \frac{A_m}{A_c}$$

- Un valor de $\mu > 1$ sobre modulación, mientras que $\mu < 1$ mantiene la modulación estándar.

4. Multiplicador (Producto de Señales)

- Se implementa con un bloque *Product* en Simulink.
- Realiza la operación:

$$x_{AM}(t) = [1 + \mu M(t)]C(t)$$

- Genera la señal modulada en amplitud.

5. Scope (Visualización de Señales)

- Se incluyen tres *Scope* para visualizar:
 - **Señal Portadora**
 - **Señal Moduladora**
 - **Señal Modulada en Amplitud**

Nótese la **Figura 30**.

Ejemplo de variación de parámetros

Configuración inicial

- **Frecuencia de la moduladora:**

$$f_m = 5000 \text{ Hz}$$

- **Frecuencia de la portadora:**

$$f_c = 150.000 \text{ Hz}$$

		<ul style="list-style-type: none">• Índice de modulación: $\mu = 0.5$ <p>Observaciones: Se observa una modulación clara donde la conteniente sigue la señal moduladora sin distorsión; esto lo podemos evidenciar en la Figura 31.</p> <p>Cambio: Aumento del Índice de Modulación (μ)</p> <ul style="list-style-type: none">➤ Se incrementa μ a 1.2 (sobre modulación). <p>Efecto:</p> <ul style="list-style-type: none">❖ La amplitud de la portadora varía en un 100% respecto a su valor original, logrando una modulación completa.❖ La envolvente ahora muestra picos más notorios y zonas donde la amplitud de la portadora casi se anula.❖ Si el índice de modulación supera 1.0, se produciría sobre-modulación, lo que generaría distorsión y pérdida de información en la transmisión.❖ Lo anterior lo podemos visualizar en la Figura 32. <p>Apreciaciones finales</p> <ul style="list-style-type: none">✓ Una modulación proporcional ($\mu \approx 1$) garantiza una buena transmisión sin
--	--	--

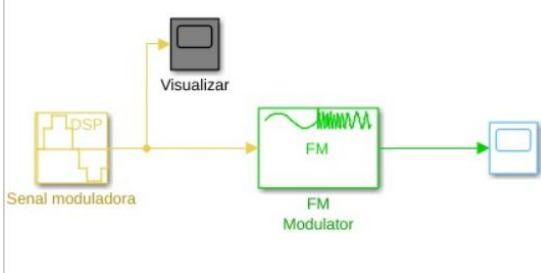
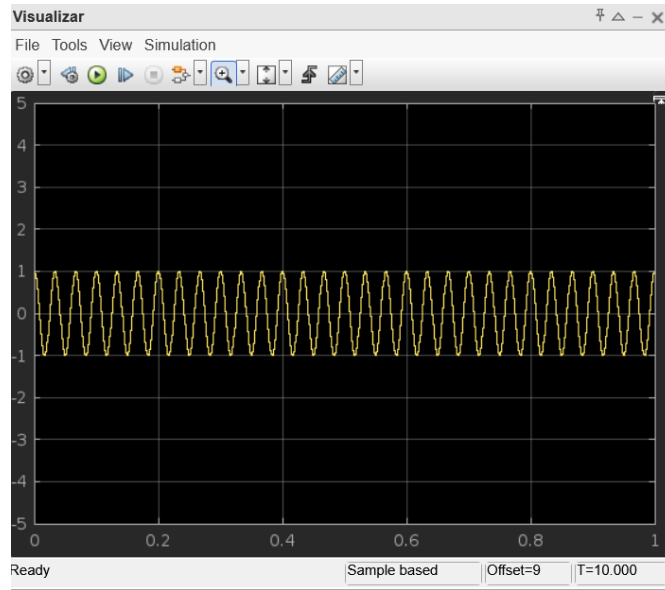
		<p>distorsión.</p> <ul style="list-style-type: none">✓ Una submodulación ($\mu < 1$) genera una señal más estable y sencilla de modular.✓ Una sobre modulación ($\mu > 1$) puede causar pérdida de información y distorsiones no esperadas.
<p>FM</p> <p>[10p]</p>	<p>Figura 33</p> <p><i>FM</i></p>  <p>Nota. Señal modulada FM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta Simulink. Elaboración propia.</p>	<p>Descripción de la Aplicación</p> <p>Esta aplicación permite visualizar la modulación en frecuencia (FM) utilizando Simulink. Se emplean señales senoidales para representar la señal portadora y la señal moduladora, generando una señal modulada en frecuencia que se observa en el Scope; véase la Figura 33.</p> <p>Se han añadido componentes para definir los siguientes parámetros:</p> <ul style="list-style-type: none">• Señal moduladora (frecuencia y amplitud); lo podemos identificar en la Figura 33 con el color amarillo.• Índice de modulación en frecuencia (determina la

Figura 34

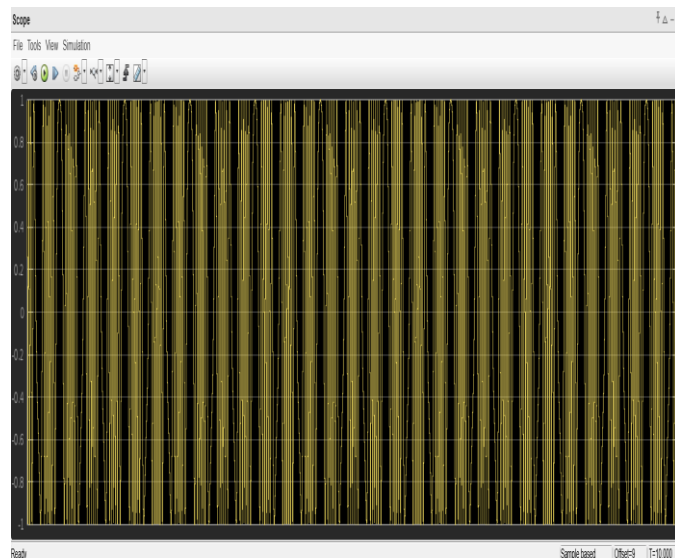
FM



Nota. Señal modulada FM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

Figura 35

FM



Nota. Señal modulada FM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

desviación de frecuencia de la portadora).

- **Visualización de señales en distintas etapas del proceso**, permitiendo analizar la modulación paso a paso; es el campo de color gris en la **Figura 33**.

Componentes del Modelo en Simulink

Señal Moduladora

- Es la señal que contiene la información a transmitir.
- Se genera con un bloque **DSP** en Simulink.
- Su ecuación es:

$$M(t) = A_m \cos(2\pi f_m t)$$

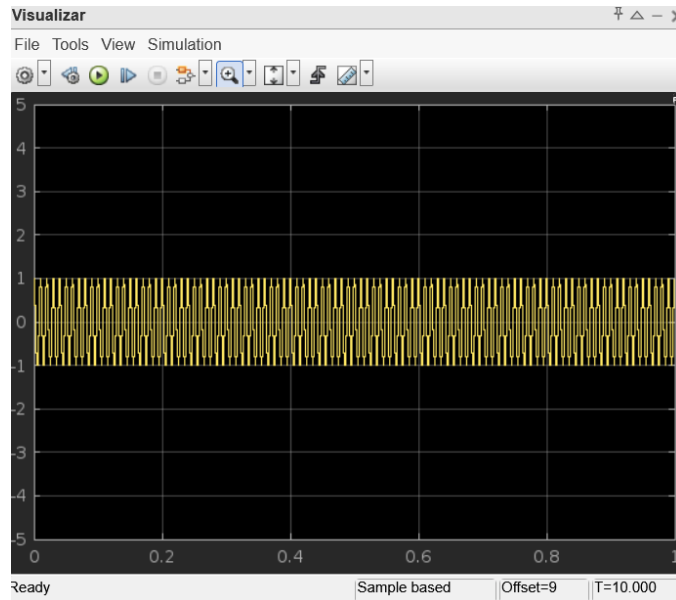
- **Parámetros configurables:**
 - Frecuencia de la señal moduladora (f_m).
 - Amplitud de la señal moduladora (A_m).
- Nótese la **Figura 34**.

Modulador FM

- Modula la señal portadora en frecuencia según la señal moduladora.
- Se implementa con un bloque **FM Modulator** en Simulink.

Figura 36

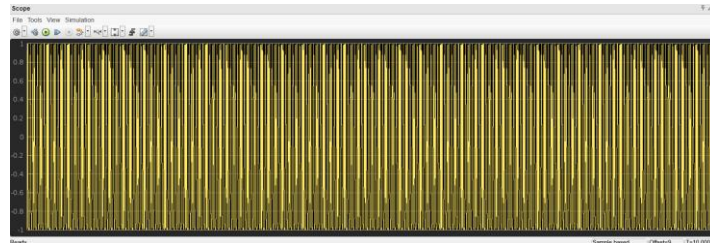
FM



Nota. Señal modulada FM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

Figura 37

FM



Nota. Señal modulada FM con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

- Su ecuación es:

$$S_{FM}(t) = A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t))$$

donde β es el índice de modulación en frecuencia.

- **Parámetros configurables:**
 - Frecuencia de la portadora (f_c).
 - Amplitud de la portadora (A_c).
 - Índice de modulación en frecuencia (β).

Visualización (Scope)

- Se incluyen Scopes para visualizar:
 - La **señal moduladora** antes de la modulación.
 - La **señal modulada en frecuencia (FM)**.

Ejemplo de Variación de Parámetros

Configuración inicial:

1. Señal moduladora
 - **Amplitud:** 1
 - **Frecuencia:** $f_m = 30 \text{ Hz}$
 - **Desfase:** $\frac{\pi}{2}$
 - **Modo de muestreo:** Discreto

- **Método de computación:**

Función
trigonométrica

- **Tiempo de**

muestreo: $\frac{1}{10000}$

Esto genera una señal senoidal con frecuencia de 30 Hz y un desfase de $\frac{\pi}{2}$, lo que significa que la onda inicia en su valor máximo (1) en $t=0$; véase la **Figura 35**.

2. Modulador FM

- **Frecuencia de la portadora:** $f_c = 200 \text{ kHz}$
- **Fase inicial:** 0 rad
- **Desviación de frecuencia:** 180 Hz

¿Cómo funciona?

- ✚ La señal moduladora (30 Hz) modifica la frecuencia instantánea de la señal portadora (200 Hz).
- ✚ La desviación de frecuencia de 180 Hz significa que la frecuencia instantánea oscilará entre $200 - 180 = 20 \text{ Hz}$ y $200 + 180 = 380 \text{ Hz}$ dependiendo del valor de la señal moduladora.
- ✚ En el dominio del tiempo, esto se traduce en un

		<p>cambio en la velocidad de oscilación de la portadora según la amplitud de la señal moduladora.</p> <p>Observaciones:</p> <ul style="list-style-type: none">• Se observa una variación en la frecuencia de la portadora de acuerdo con la señal moduladora.• Cuanto mayor es la amplitud de la señal moduladora, mayor es el desplazamiento en frecuencia de la portadora; véase la Figura 36. <p>Cambio: Modificación de parámetros</p> <ul style="list-style-type: none">➤ Frecuencia de la señal moduladora: 100 Hz (antes era 30 Hz)➤ Frecuencia portadora: Mantener en 200 Hz➤ Desviación de frecuencia: $6 * 100 = 600 \text{ Hz}$ (antes era 180 Hz) <p>Los cambios anteriores son visibles en la Figura 37.</p> <p>Efecto:</p> <p>Al aumentar la frecuencia moduladora, la variación de frecuencia ocurre más rápido y la señal modulada cambia bruscamente. Esto puede emplearse y ser útil en</p>
--	--	---

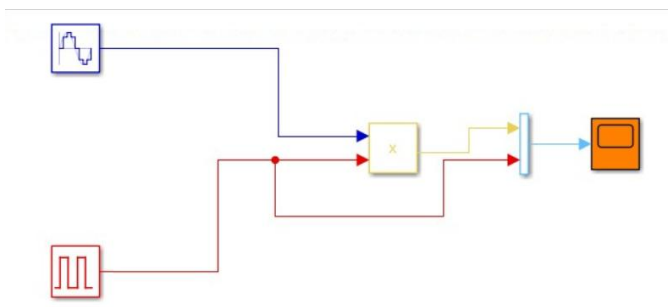
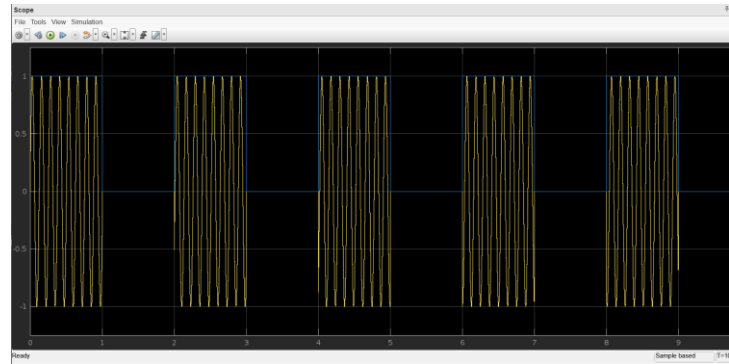
		<p>aplicaciones como la transmisión de datos, donde se necesita una respuesta más rápida en la modulación.</p> <p>Apreciaciones finales</p> <p>Este modelo Simulink permite una representación de la modulación FM, brindando un acercamiento a la exploración con distintos parámetros y su impacto en la señal modulada. Modificaciones en la frecuencia de la señal moduladora y la desviación de frecuencia pueden otorgarnos información sobre la respuesta del sistema, lo que hace que este experimento sea ideal para estudiar la variabilidad de señales en telecomunicaciones y procesamiento digital.</p>
ASK [10p]	<p>Figura 38 ASK</p>  <p>Nota. Señal modulada ASK con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta Simulink. Elaboración propia.</p>	<p>Descripción de la aplicación</p> <p>Esta aplicación permite visualizar la modulación por desplazamiento de amplitud (ASK) utilizando Simulink. Se emplean señales senoidales para representar la señal portadora y una señal de pulso para representar la señal moduladora. El resultado es una señal modulada en amplitud, cuya variación puede observarse en el Scope; véase la Figura 39 seguida de la Figura 38.</p>

Figura 39

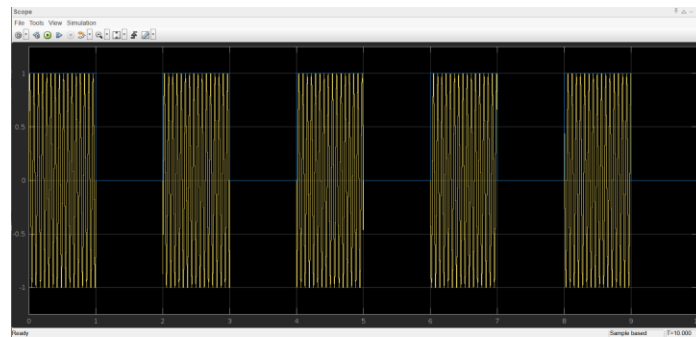
ASK



Nota. Señal modulada ASK con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia.

Figura 40

ASK



Nota. Señal modulada ASK con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

Se han añadido componentes para definir los siguientes parámetros:

- **Señal moduladora (frecuencia y amplitud):** Representada en color rojo en la **Figura 38**.
- **Portadora (frecuencia y amplitud):** Señal senoidal en color azul.
- **Multiplicador de señales:** Se encarga de realizar la modulación ASK.
- **Visualización de señales en distintas etapas del proceso,** lo que permite analizar la modulación paso a paso (área en color naranja **Figura 38**).

Componentes del modelo en Simulink

1. Señal moduladora

- Es la señal que contiene la información digital a transmitir.
- Se genera con un **bloque de Generador de Pulsos** en Simulink.
- Su ecuación es:

$$M(t) = A_m * Pulse(t)$$

Nota: Esta señal toma valores binarios (0 y 1), lo que permitirá activar o desactivar la portadora.

2. Señal portadora

- Es una señal senoidal de alta frecuencia que transportará la señal moduladora.
- Se genera con un **bloque de Onda Senoidal** en Simulink.
- Su ecuación es:

$$S_p(t) = A_c * \cos (2\pi f_c t)$$

3. Modulador ASK

- Modula la señal portadora en amplitud según la señal moduladora.
- Se implementa con un **bloque de Multiplicación** en Simulink.
- Su ecuación es:

$$S_{ASK}(t) = M(t) * S_p(t)$$

$$= A_m * A_c * \text{Pulse}(t) * \cos (2\pi f_c t)$$

Ejemplo de Variación de Parámetros

Configuración inicial:

- **Señal moduladora:**
 - Amplitud: 1

		<ul style="list-style-type: none">○ Frecuencia: $f_m = 0.5 \text{ Hz}$ (período = 2 s)○ Ancho de pulso: 50%○ Fase: 0• Portadora:<ul style="list-style-type: none">○ Amplitud: 1○ Frecuencia: $f_c = 0.5 \text{ Hz}$○ Tiempo de muestreo: 0.0001 s <p>Esto genera una señal ASK donde la portadora se transmite durante la mitad del tiempo y desaparece en la otra mitad.</p> <p>Nótese la Figura 39.</p> <p>Cambio: Modificación de parámetros</p> <p>➤ Frecuencia de la portadora: 100 Hz (antes era 50 Hz).</p> <p>Efecto de los cambios:</p> <ul style="list-style-type: none">✓ <i>Al usar un botón momentáneo, la señal ASK responderá en tiempo real a la interacción del usuario.</i>✓ <i>Aumento de la frecuencia de la portadora</i><ul style="list-style-type: none">○ La portadora se comprime en el dominio del tiempo.
--	--	---

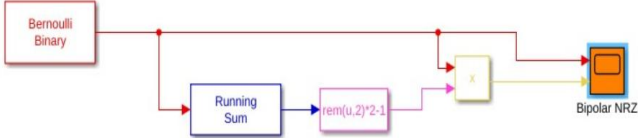
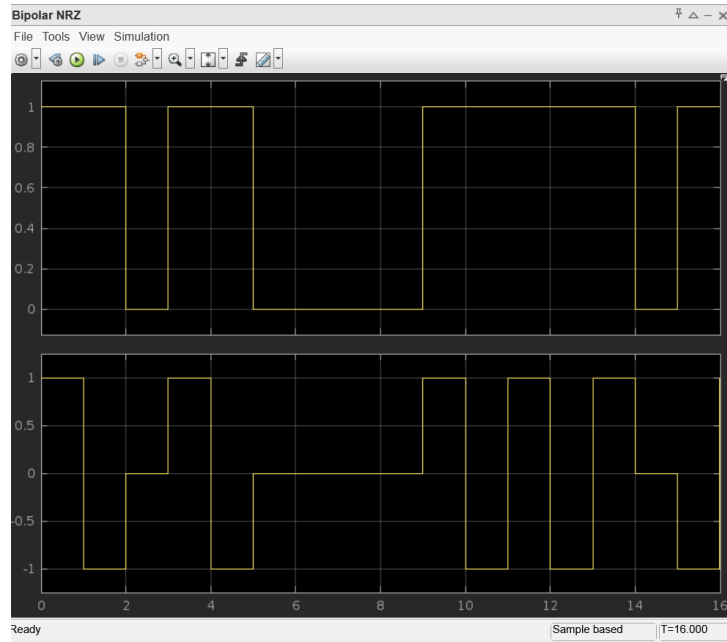
		<div><div>○ Se observa mayor densidad en la señal modulada.</div><div>Los efectos de esta variación pueden observarse en la Figura 40.</div><div>Apreciaciones finales</div><div>Este modelo en Simulink permite una representación interactiva de la modulación ASK, facilitando la experimentación con distintos parámetros y su importancia en la señal modulada, asimismo, como estudiantes de ingeniería de sistemas nos permite conocer más de cerca su influencia en telecomunicaciones y respuesta en tiempo real en sistemas digitales.</div></div>
<div>NRZ [10p]</div>	<div><div><div>Figura 41 <i>BIPOLAR NRZ</i></div><div></div></div><div>Nota. Señal modulada Bipolar NRZ con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta Simulink. Elaboración propia</div></div>	<div><div>Descripción de la Aplicación</div><div>Esta aplicación permite visualizar la señal codificada en formato Bipolar NRZ a partir de una secuencia binaria generada aleatoriamente, esto gracias a las modificaciones realizadas en el sistema de simulación. En la simulación de Simulink, se han utilizado los siguientes bloques clave:</div></div>

Figura 42

Bipolar NRZ



Nota. Señal modulada Bipolar NRZ con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

- **Bernoulli Binary:** Genera una secuencia binaria aleatoria.
- **Running Sum:** Calcula la suma acumulativa de la secuencia.
- **Operación módulo** ($\text{rem}(x,2)*2-1$): Convierte la secuencia binaria en una representación bipolar (-1 y 1).
- **Multiplicador:** Ajusta la señal antes de enviarla al bloque de salida.
- **Bipolar NRZ:** Muestra la señal codificada final.

Nótese el diagrama en la **Figura 41**.

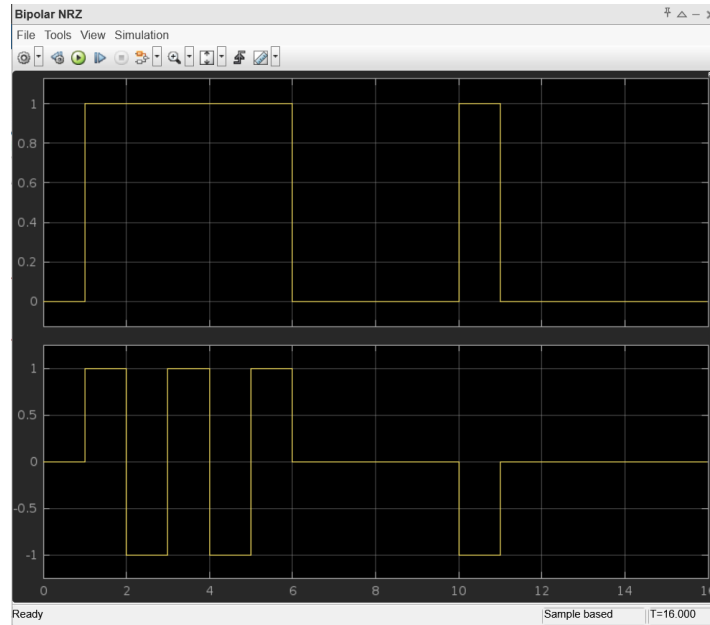
Componentes del Modelo

1. Bernoulli Binary

- Genera una secuencia aleatoria de bits.
- Parámetros ajustables:
 - **Probabilidad de 1:** Define la proporción de 1s en la secuencia.

Figura 43

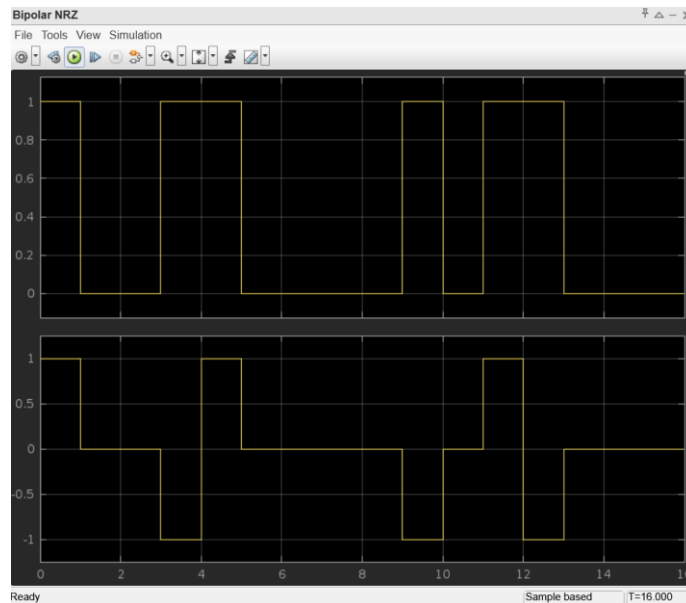
Bipolar NRZ



Nota. Señal modulada Bipolar NRZ con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

Figura 44

Bipolar NRZ



Nota. Señal modulada Bipolar NRZ con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

- **Periodo de muestreo:**
Determina la frecuencia con la que se generan los bits.

2. Running Sum

- Acumula la suma de la secuencia binaria para la conversión a la señal bipolar.

3. Operación Modulo ($\text{rem}(u,2)-1$)

- Convierte la secuencia acumulada en valores alternantes.

4. Multiplicador

- Ajusta la amplitud de la señal.

5. Bipolar NRZ

- Muestra la señal final en formato Bipolar NRZ.

Lo anterior lo podemos visualizar en la **Figura 42**.

		<p>Parámetros que pueden variar</p> <p>Al modificar los siguientes parámetros, podemos observar los siguientes cambios en la señal Bipolar NRZ:</p> <ul style="list-style-type: none">• Probabilidad de 1 en el generador Bernoulli: Cambia la densidad de unos en la señal generada.• Periodo de muestreo: Afecta la resolución temporal de la señal.• Factor de multiplicación en el bloque multiplicador: Ajusta la amplitud de la señal. <p>Estos parámetros nos permiten experimentar en la señal generada para analizar distintos escenarios en la codificación bipolar NRZ; véase la Figura 43.</p> <p>Apreciaciones finales</p> <p>La codificación Bipolar NRZ es empleado en diversos contextos debido a su habilidad para eliminar la componente DC y mejorar la sincronización en sistemas de transmisión de datos. En este esquema:</p>
--	--	--

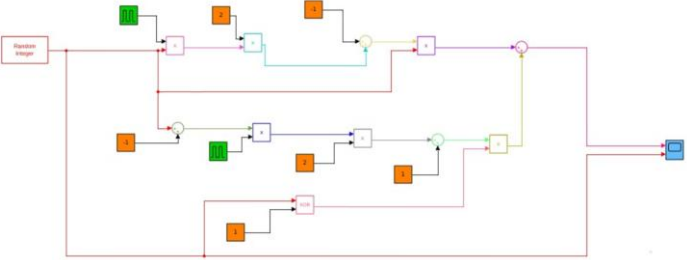
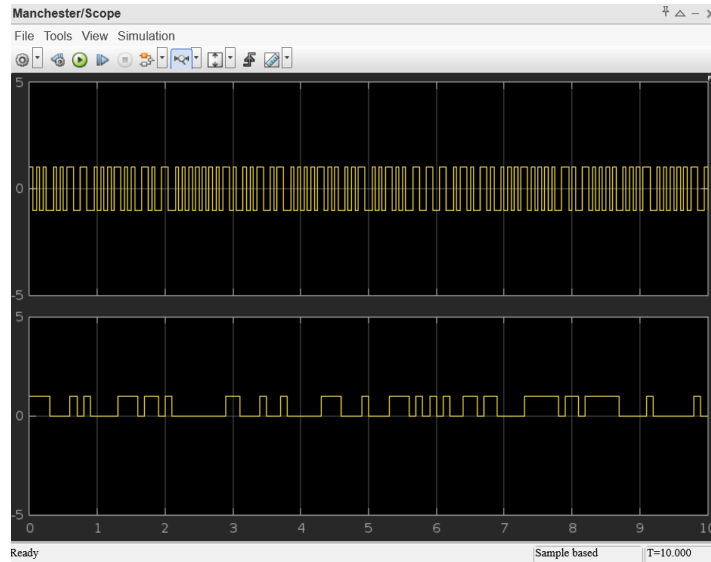
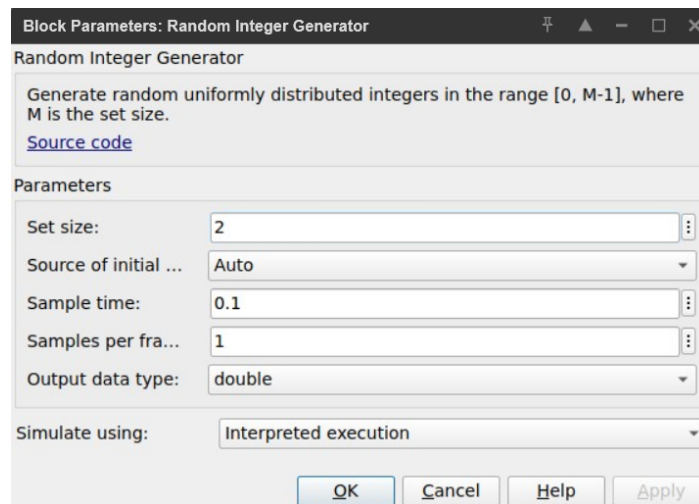
		<ul style="list-style-type: none">• Un 1 se representa como +1.• Un 0 se representa como -1. <p>Podemos visualizarlo de manera más amplia y rigurosa en la Figura 44.</p> <p>Este contraste mejora la profundización y entendimiento de la señal, además, es parte fundamental en sistemas de transmisión digital. Finalmente, se recomienda ajustar la frecuencia de muestreo según el contexto y la herramienta de visualización para optimizar la calidad de la representación.</p>
MANCHESTER R [10p]	<p>Figura 45 MANCHESTER</p>  <p>Nota. Señal MANCHESTER con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta Simulink. Elaboración propia</p>	<p>Descripción de la Aplicación</p> <p>Esta aplicación permite visualizar la señal codificada mediante el esquema de codificación Manchester a partir de una secuencia binaria generada aleatoriamente. La codificación Manchester representa los bits de una forma en la que cada bit tiene una transición en el medio del intervalo de tiempo asignado, proporcionando una mejora considerable en la sincronización de la señal y eliminando la componente DC.</p> <p>En este modelo de Simulink, se generan datos binarios aleatorios y se transforman en una señal Manchester, utilizando operaciones aritméticas y lógicas vistas en clase y extraídas del texto guía “<i>Computer Networks Data Communications, Internet and Security</i>” para modificar la</p>

Figura 46
MANCHESTER



Nota. Señal MANCHESTER con diferentes parámetros para la visualización en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

Figura 47
MANCHESTER



Nota. Valores iniciales de la codificación MANCHESTER en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

representación de los bits en el tiempo; véase la **Figura 45**.

Componentes de la Interfaz

1. Generación de datos aleatorios (Random Integer Block)

- Este bloque genera una secuencia binaria aleatoria (0s y 1s) que será codificada en formato Manchester.
- Los valores de salida son enteros (0 o 1), representando los bits de la señal digital.

2. Conversión de bits a señal Manchester

- La codificación Manchester se basa en representar cada bit con una transición en la mitad del período del bit.
- Para ello, el modelo implementa las siguientes operaciones:

a) Conversión de valores binarios a una representación bipolar

- El bloque multiplicador con un valor de 2 y una constante de -1 transforma la señal binaria X en la forma bipolar:

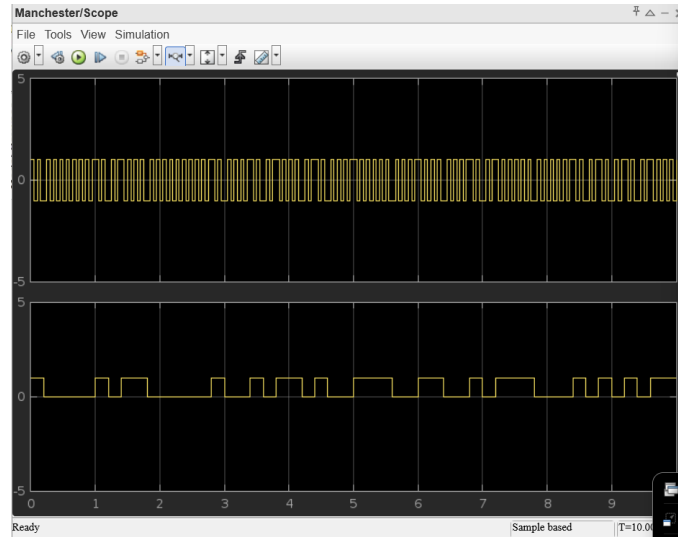
$$X' = 2X - 1$$

- Si $X = 0$, entonces $X' = -1$.
- Si $X = 1$, entonces $X' = 1$.
- Esto es necesario porque la codificación Manchester usa **valores** de amplitud **positiva y negativa**.

b) Multiplicación con una onda cuadrada de frecuencia doble

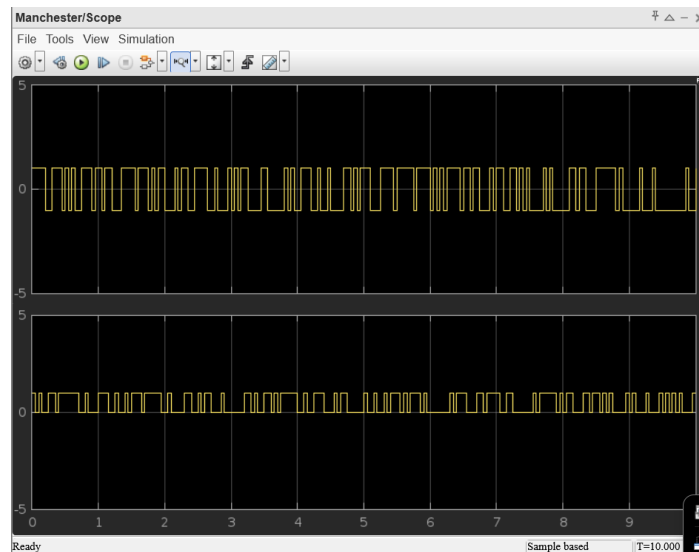
- Un generador de onda cuadrada se utiliza para alternar la señal, asegurando que cada bit tenga una

Figura 48
MANCHESTER



Nota. Señal MANCHESTER con variación de los parámetros de la codificación en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

Figura 49
MANCHESTER



Nota. Señal MANCHESTER con variación de los parámetros de la codificación en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

transición en la mitad del intervalo de bit.

- La señal original es modulada con esta onda cuadrada, resultando en la transición de la codificación Manchester.

c) Generación de la señal invertida

- En paralelo, la señal original es invertida sumando -1 y aplicando las mismas operaciones que la señal directa.
- Esta operación es clave para generar correctamente las transiciones de la codificación Manchester.

d) Operación XOR para garantizar las transiciones correctas

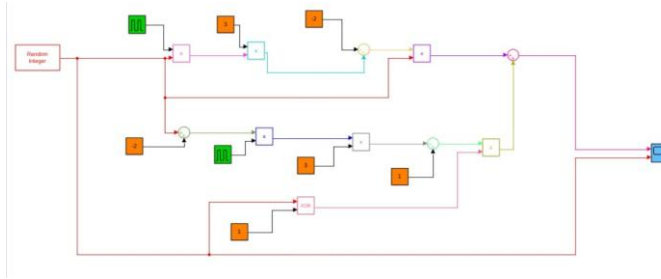
- Un bloque de operación lógica XOR se emplea en la parte inferior del diagrama.
- La entrada de este bloque proviene de la señal original y su versión desplazada en el tiempo.
- El resultado de esta operación garantiza que cada bit tenga una transición en la mitad del intervalo.
- Lo anterior visualizado en la **Figura 45**.

3. Suma de las señales y salida final

- En última instancia, las dos ramas del modelo se combinan mediante sumadores y multiplicadores para generar la señal Manchester completa.
- El resultado final es enviado a un bloque de visualización (SCOPE), donde se puede analizar la forma de onda de la señal codificada.

Figura 50

MANCHESTER

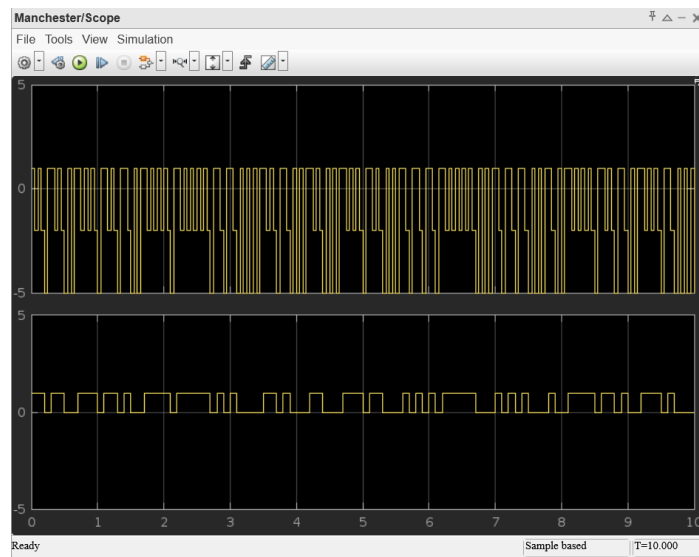


Nota. Valores iniciales de la codificación MANCHESTER en MATLAB, con la ayuda de la herramienta **Simulink**.

Elaboración propia

Figura 51

MANCHESTER



Nota. Señal MANCHESTER con variación de los parámetros de la codificación en MATLAB, con la ayuda de la herramienta **Simulink**. Elaboración propia

Parámetros ajustables

Visualicemos la salida en el Scope en la **Figura 46**; podemos ver claramente la señal MANCHESTER.

¿Pero qué pasaría si cambiáramos algunos parámetros? ¡Vamos a averiguarlo!

Cambio:

1. Bloque de generación de datos aleatorios (Random Integer):

- Variar el Sample Time (Ejemplo: 0.2 o 0.5) en la **Figura 47** ahora cambiemos a 0.2 y luego a 0.05.

Efecto:

- Para el cambio a 0.2, en la **Figura 48**, la generación de bits será más lenta, por lo que la señal Manchester resultante se verá más espaciada en el tiempo. Además, se visualizarán menos transiciones en el mismo período de simulación.
- Para el cambio a 0.05, en la **Figura 49**, la generación de bits será más rápida, aumentando la cantidad de transiciones visibles. Por otro lado, es importante recordar que, si lo reduces demasiado, la señal puede verse demasiado densa o difícil de analizar.

2. Bloque de conversión a bipolar (multiplicador y constante -1)

- Bit 1 → Valor positivo
- Bit 0 → Valor negativo

Cambio:

		<div><div><div><div></div><div></div><div></div></div><div>Cambiando la constante del multiplicador (Ejemplo: de 2 a 3)</div></div><div><div><div></div><div></div><div></div></div><div>Cambiar la constante -1 (Ejemplo: de -1 a -2)</div></div></div> <div>Efecto:</div> <div>Primero realizamos los cambios de los parámetros iniciales, en este caso modificaremos las constantes; véase la Figura 50.</div> <div><div><div></div><div></div><div></div></div><div>Si cambias de 2 a 3, la señal tendrá valores más altos y si cambias de -1 a -2, los valores negativos serán más intensos, lo anterior lo podemos presenciar y analizar con claridad en la Figura 51.</div></div> <div>Conclusión</div> <div>Este modelo de Simulink implementa la codificación Manchester mediante una combinación de operaciones matemáticas y lógicas. Se resalta la inversión de señal, la modulación con una onda cuadrada y la operación XOR para obtener transiciones correctas en cada bit. Experimentando con los valores teóricos y en un ejercicio de prueba y error, se puede observar con precisión la transformación de los datos binarios a la señal codificada Manchester.</div>
--	--	---

Nota: Quisiera referenciar en este espacio algunos libros y textos guía contenidos en el programa de Redes I y recomendados por la docente Zoraima Oñate, los cuales, durante la elaboración del trabajo, fueron de gran ayuda y me permitieron adentrarme más en los temas trabajados, así como adquirir los conocimientos necesarios para su desarrollo.

Referencias

Elahi, Ata (2024). Computer Networks Data Communications, Internet and Security. (1ª ed.). Springer International Publishing. En la base de datos Ebook Central Perpetual, DDA and Subscription Titles.

Moya, J. M. H. (2010). Telecomunicaciones: tecnologías, redes y servicios.

Olifer, Natalia., y Olifer, Victor (2009). Redes de computadoras: principios, tecnología y

protocolos para el diseño de redes (1ª ed.). McGraw-Hill

Stallings, William (2014). Data and computer communications (10ª ed.). Pearson.

Tanenbaum, Andrew (1997). Redes de computadoras (3ª ed.). Prentice-Hall.

Wikipedia contributors. (2025, 12 febrero). *ANSI/TIA-568*. Wikipedia. <https://en.wikipedia.org/wiki/ANSI/TIA-568>