

Informe IX – Análisis de Modulación Digital y Codificación de Canal usando Simulink

Dylan Ferney Vasquez Rojas
1401597
Comunicaciones Digitales

Resumen— Este trabajo presenta el análisis y simulación de diferentes esquemas de modulación digital y técnicas de codificación de canal empleando la plataforma Simulink de MATLAB. Se estudian las modulaciones BPSK y 16-QAM bajo condiciones de ruido aditivo blanco gaussiano (AWGN), comparando el comportamiento de la tasa de error de bit (BER) frente a la relación señal a ruido (SNR). Asimismo, se implementan y evalúan los efectos de la codificación Hamming y la codificación convolucional con decodificador Viterbi para determinar su impacto sobre la robustez del sistema de comunicación. Los resultados obtenidos evidencian una disminución significativa en la BER al aplicar codificación, demostrando la ganancia de codificación y la mejora en la confiabilidad del canal digital.

Abstract— This work presents the analysis and simulation of different digital modulation schemes and channel coding techniques using the MATLAB Simulink platform. BPSK and 16-QAM modulations are studied under additive white Gaussian noise (AWGN) conditions, comparing the bit error rate (BER) versus the signal-to-noise ratio (SNR). Additionally, the effects of Hamming coding and convolutional coding with a Viterbi decoder are implemented and evaluated to determine their impact on the robustness of the communication system. The results obtained show a significant decrease in BER when coding is applied, demonstrating the coding gain and the improvement in the reliability of the digital channel.

I. INTRODUCCIÓN

En los sistemas de comunicaciones digitales, la modulación y la codificación de canal son etapas fundamentales que determinan la calidad y confiabilidad de la transmisión de información a través de un medio ruidoso. La modulación permite adaptar la señal digital al canal de comunicación, optimizando su eficiencia espectral y resistencia frente al ruido, mientras que la codificación de canal introduce redundancia controlada con el propósito de detectar y corregir errores generados durante la transmisión.

El estudio de estas técnicas resulta esencial para comprender el comportamiento de los sistemas digitales modernos, que requieren altos niveles de robustez y eficiencia, como los utilizados en redes inalámbricas, televisión digital y comunicaciones satelitales.

En este trabajo se analizan y comparan dos esquemas de modulación: BPSK (Binary Phase Shift Keying) y 16-QAM

(Quadrature Amplitude Modulation), bajo la influencia de un canal AWGN (Additive White Gaussian Noise). Además, se evalúa el impacto de aplicar diferentes técnicas de codificación de canal, específicamente los códigos de Hamming y la codificación convolucional con decodificación Viterbi, sobre el rendimiento del sistema.

Las simulaciones se realizaron utilizando el entorno Simulink de MATLAB, debido a su capacidad para modelar sistemas de comunicación de manera modular y visual. Los resultados obtenidos permiten analizar la relación entre la tasa de error de bit (BER) y la relación señal a ruido (SNR), cuantificando la ganancia de codificación obtenida al emplear métodos de corrección de errores.

De esta forma, el presente informe busca evidenciar cómo la combinación de técnicas de modulación digital y codificación de canal mejora la integridad de los datos transmitidos, sirviendo como base para el entendimiento de sistemas más complejos utilizados en las tecnologías de comunicación actuales.

DESARROLLO DE LA PRÁCTICA

El desarrollo de la práctica se realizó en el entorno de Simulink, perteneciente a MATLAB, utilizando las herramientas del Communications Toolbox, que permiten modelar y analizar sistemas de comunicación digital. El objetivo principal fue evaluar el comportamiento de diferentes técnicas de modulación y codificación frente a la presencia de ruido aditivo blanco gaussiano (AWGN).

Para ello, se construyeron dos esquemas principales de simulación:

- Modulación BPSK con y sin codificación Hamming.
- Modulación 16-QAM con y sin codificación convolucional (decodificador Viterbi).

En cada montaje se generaron secuencias binarias aleatorias mediante un bloque Bernoulli Binary Generator, las cuales fueron moduladas, transmitidas a través de un canal AWGN, demoduladas y finalmente comparadas con la señal original para calcular la tasa de error de bit (BER).

El análisis se realizó para distintos valores de SNR (Signal-to-Noise Ratio) con el fin de observar su efecto sobre la calidad

de la transmisión. Para cada caso se obtuvieron tanto los diagramas de constelación como las curvas BER vs SNR, comparando el rendimiento de los sistemas con y sin codificación de canal.

I. Simulación de modulación BPSK con y sin codificación Hamming

La primera simulación implementada corresponde a la modulación BPSK (Binary Phase Shift Keying), tanto sin codificación como con codificación Hamming (7,4). El propósito de este montaje es analizar el efecto de la codificación de canal sobre la tasa de error de bit (BER) frente a distintas relaciones señal a ruido (SNR) en un canal AWGN (Additive White Gaussian Noise).

En la Ilustración 1 se muestra el diagrama del sistema implementado en Simulink, el cual consta de dos trayectorias paralelas: una sin codificación (parte inferior) y otra con codificación Hamming (parte superior). Ambas rutas parten del mismo generador de datos binarios y permiten comparar el desempeño en condiciones idénticas de transmisión.

- Bernoulli Binary Generator: genera una secuencia aleatoria de bits con parámetros Initial seed = 12345, Sample time = 0.001 s, y Samples per frame = 4.
- Hamming Encoder / Decoder: realiza la codificación y decodificación del mensaje binario para detección y corrección de errores.
- BPSK Modulator / Demodulator Baseband: convierte los bits en señales BPSK, donde el bit lógico “1” y “0” se representan con fases de 0° y 180° , respectivamente.
- AWGN Channel: introduce ruido gaussiano aditivo con un valor de SNR variable, controlado para el análisis del desempeño.
- Error Rate Calculation: calcula la BER, el número total de errores y el número total de símbolos transmitidos.
- Display: muestra los valores medidos de BER y errores totales durante la simulación.
- Constellation Scope: visualiza el diagrama de constelación de la señal modulada y demodulada.

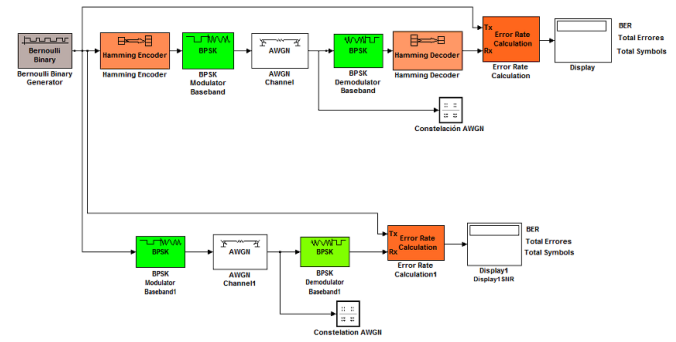


Ilustración. 1. Modulación BPSK con y sin codificación.

Se ejecutó la simulación con los parámetros iniciales especificados en la guía.

Se modificó el valor del SNR (dB) en el bloque AWGN dentro de un rango de -40 dB a $+10$ dB.

Para cada valor de SNR, se registraron los valores de BER, errores totales y símbolos transmitidos, tanto para el sistema con como sin codificación.

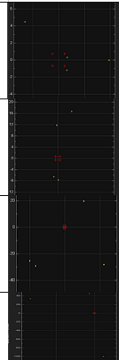
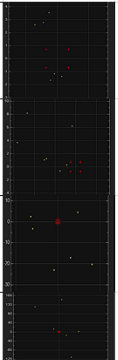
SNR (dB)	BPSK			Imagen	BPSK-Hamming Encoder			Imagen
	BER	Errores	Símbolos		BER	Errores	Símbolos	
-10	0.3551	98	276		0.3696	102	276	
-20	0.4902	100	204		0.4853	99	204	
-30	0.526	101	192		0.5156	99	192	
-40	0.5435	100	184		0.5489	101	184	

Ilustración. 2. Tabulación de los primeros BPSK con y sin codificación.

(Todas las evidencias completas están adjuntas al repositorio de GitHub, tanto el excel como las ilustraciones de codificación y sin codificar del BPSK)

Los resultados obtenidos se tabularon y posteriormente se graficó la curva BER vs SNR para ambos casos.

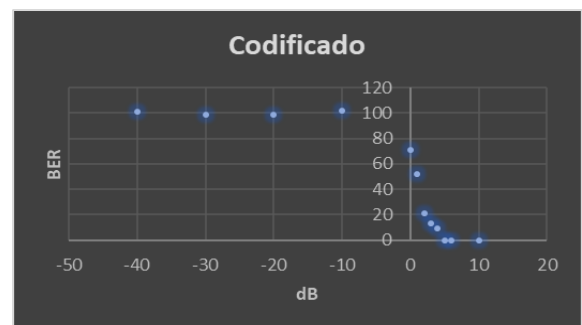


Ilustración. 3. Gráfico BER vs SNR Codificado.

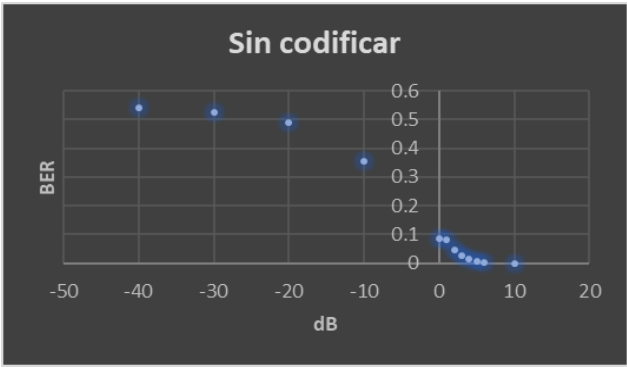


Ilustración. 4. Gráfico BER vs SNR Sin Codificar.

Luego, en la ventana de comandos de MatLab, se ejecutó la herramienta “bertool”, mediante esta herramienta se obtuvo la gráfica de E_b/N_0 (dB) vs BER para las modulaciones BPSK con y sin codificación, obteniendo el siguiente grafico.

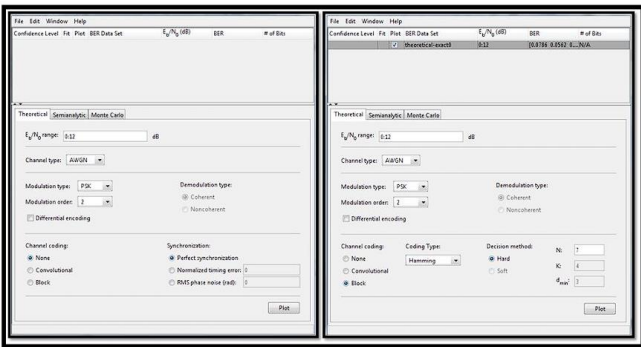


Ilustración. 5. Parámetros de configuración de la Herramienta Bertool de Matlab para BPSK.

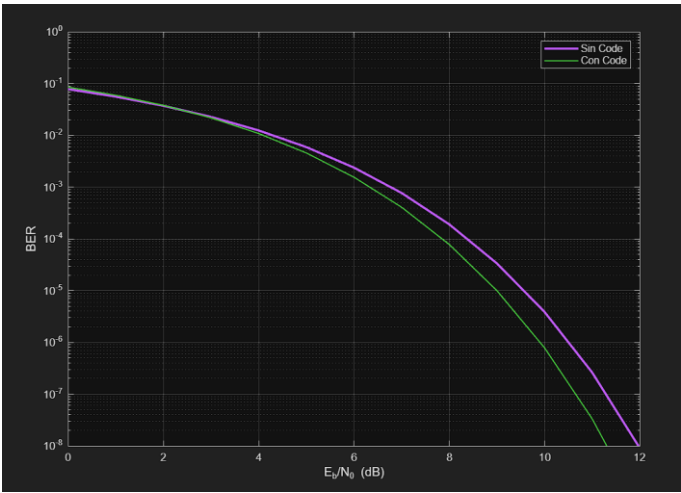


Ilustración. 6. Bertool BPSK.

II. Simulación de modulación 16-QAM con y sin codificación convolucional

El segundo esquema implementado corresponde a la modulación 16-QAM (Quadrature Amplitude Modulation), tanto sin codificación como con codificación convolucional y decodificación Viterbi. Este montaje tiene como objetivo analizar la influencia de la codificación de canal sobre el desempeño de un sistema de mayor orden de modulación,

donde la información se transmite simultáneamente en la amplitud y la fase de la portadora.

La Ilustración 7 muestra el diagrama del sistema desarrollado en Simulink, el cual se compone de dos rutas de transmisión paralelas que parten del mismo generador de datos binarios: una sin codificación y otra con un codificador convolucional. Ambas señales son moduladas, transmitidas a través de un canal AWGN y demoduladas para el cálculo de la tasa de error de bit (BER).

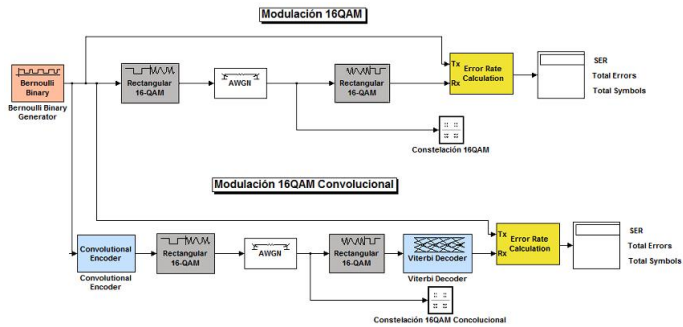


Ilustración. 7. Modulación 16-QAM con y sin codificación.

- Bernoulli Binary Generator: genera la secuencia binaria de entrada con parámetros Initial seed = 12345, Sample time = 1/1e6, Samples per frame = 256, y tipo de salida uint8.
- Rectangular QAM Modulator Baseband / Demodulator: implementa la modulación y demodulación en cuadratura, configurada con M-ary number = 16, Constellation ordering = Gray, Normalization method = Average Power y Average power = 1 W.
- Convolutional Encoder / Viterbi Decoder: aplica la codificación y decodificación de canal para corrección de errores, utilizando el método de decisión dura (hard decision).
- AWGN Channel: simula el canal de transmisión afectado por ruido gaussiano aditivo blanco, variando el valor de SNR (dB) entre -40 y +10 dB.
- Error Rate Calculation: calcula la tasa de error de bit (BER), el número total de errores y símbolos transmitidos en cada iteración de simulación.
- Constellation Scope: permite observar la constelación recibida, donde la dispersión de los puntos evidencia el efecto del ruido sobre la señal modulada.

Se configuraron los bloques del modelo de acuerdo con los parámetros establecidos en la guía, manteniendo las condiciones iniciales de simulación.

Se ejecutaron múltiples simulaciones modificando el valor

de SNR (dB) en el canal AWGN para analizar su efecto sobre la BER.

SNR (dB)	16-QAM				16-QAM-Hamming Encoder			
	BER	Errores	Simbolos	Imagen	BER	Errores	Simbolos	Imagen
-10	0.4397	2.20E+05	5.00E+05		0.4995	2.50E+05	5.00E+05	
-20	0.4816	2.41E+05	5.00E+05		0.5007	2.51E+05	5.00E+05	
-30	0.4944	2.47E+05	5.00E+05		0.5009	2.51E+05	5.00E+05	

Ilustración. 8. Tabulación de los primeros 16-QAM con y sin codificación

(Todas las evidencias completas estan adjuntas al repositorio de GitHub, tanto el excel como las ilustraciones de codificación y sin codificar del 16-QAM)

Se registraron los valores obtenidos en una hoja de cálculo, comparando el rendimiento del sistema con y sin codificación como se observa en la Ilustración 7.

Finalmente, se graficaron las curvas BER vs SNR para evaluar la ganancia de codificación proporcionada por el uso del codificador convolucional.

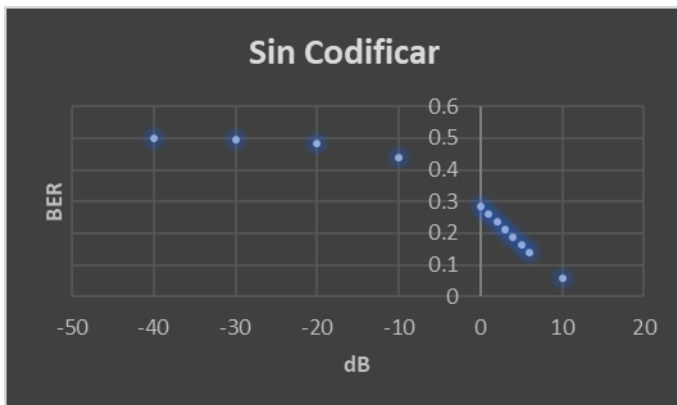


Ilustración. 9. Gráfico BER vs SNR Sin Codificar.

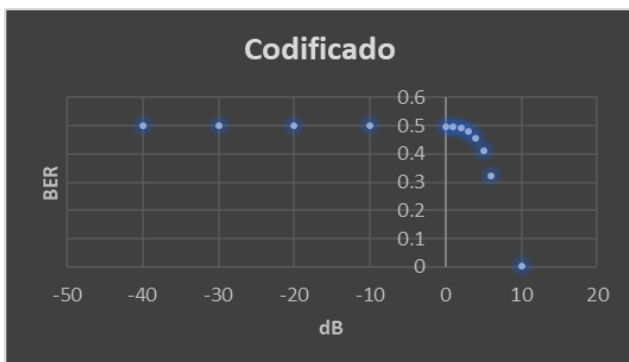


Ilustración. 10. Gráfico BER vs SNR Codificado.

Luego, en la ventana de comandos de MatLab, se ejecutó la herramienta "bertool", mediante esta herramienta se obtuvo la gráfica de E_b/N_0 (dB) vs BER para las modulaciones de 16-QAM con y sin codificación, obteniendo el siguiente grafico.

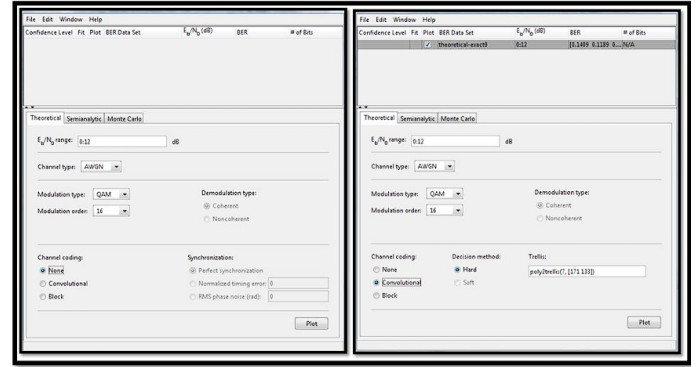


Ilustración. 11. Parámetros de configuración de la Herramienta Bertool de Matlab para 16-QAM.

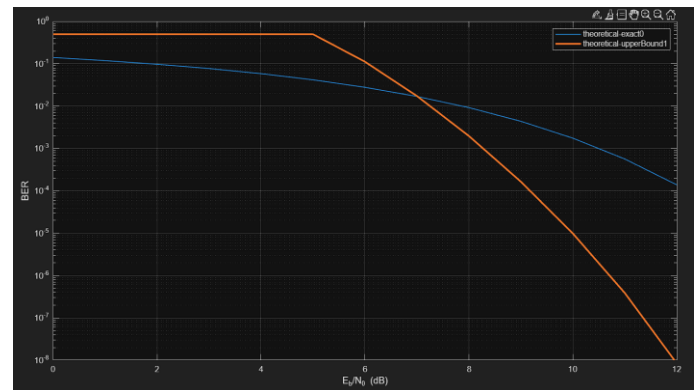


Ilustración. 12. Grafica Bertool 16-QAM.

Los resultados demuestran que la implementación de la codificación convolucional reduce significativamente la tasa de error, especialmente en condiciones de bajo SNR, confirmando su efectividad para mejorar la confiabilidad de la transmisión en modulaciones de mayor densidad espectral como la 16-QAM.

III. Analisis Complementario

a. Tipos de codificadores de canal y su clasificación:

Los codificadores de canal se emplean para agregar redundancia a los datos transmitidos, con el fin de detectar y corregir errores ocasionados por el ruido en el canal. Estos se clasifican principalmente en dos grandes grupos:

Códigos de bloque: procesan los datos en bloques de longitud fija. Cada bloque contiene k bits de información y $n-k$ bits de redundancia. Ejemplos comunes son los códigos Hamming, códigos BCH y códigos Reed-Solomon.

Códigos convolucionales: operan sobre flujos continuos de datos, donde la salida depende de los bits actuales y de los anteriores, introduciendo memoria en el sistema. Se utilizan junto con decodificadores Viterbi o algoritmos de retropropagación (MAP).

b. Funcionamiento de los códigos Hamming y

Convolucionales:

Los códigos Hamming, propuestos por Richard Hamming en 1950, permiten la detección y corrección de un solo error en cada palabra de código mediante la inserción de bits de paridad calculados según posiciones de potencia de dos. Por ejemplo, el código Hamming (7,4) toma cuatro bits de información y agrega tres bits de paridad, generando un bloque de siete bits capaz de corregir un error o detectar hasta dos errores.

Por su parte, los códigos convolucionales emplean registros de desplazamiento y compuertas lógicas XOR para generar las salidas codificadas. Su comportamiento depende de los bits actuales y de los anteriores, caracterizándose por los parámetros (n, k, m): número de bits de salida, número de bits de entrada y memoria del sistema, respectivamente. La decodificación se realiza normalmente con el algoritmo de Viterbi, que selecciona la secuencia de bits más probable comparando las trayectorias en un diagrama de estados (trellis).

c. Códigos LDPC (Low Density Parity Check):

Los códigos LDPC son una clase moderna de códigos de bloque lineales caracterizados por una matriz de paridad dispersa (de baja densidad). Introducidos por Gallager en 1962 y redescubiertos décadas más tarde, los LDPC han demostrado un rendimiento cercano al límite de Shannon, siendo empleados ampliamente en estándares de comunicación como Wi-Fi (802.11n/ac/ax), DVB-S2/S2X, y 5G NR. Su decodificación iterativa mediante el algoritmo belief propagation permite alcanzar una alta eficiencia en la corrección de errores con una complejidad computacional moderada.

d. Ganancia de codificación:

La ganancia de codificación se define como la diferencia en dB de SNR necesaria para alcanzar una misma BER entre un sistema sin codificación y uno codificado. Matemáticamente, se expresa como:

$$G_c = SNR_{sin_cod} - SNR_{con_cod}$$

Ecuación. 1. Ganancia de codificación

En las simulaciones realizadas, se observó que la codificación Hamming en BPSK y la convolucional en 16-QAM reducen la BER para los mismos niveles de SNR, evidenciando una ganancia de codificación promedio de entre 1.5 y 3 dB.

Esta mejora refleja la capacidad del sistema para mantener una transmisión más confiable, incluso en presencia de niveles significativos de ruido.

CONCLUSIONES

[1] La simulación en Simulink permitió comprender el comportamiento de los sistemas digitales bajo ruido AWGN, evidenciando la relación directa entre la relación señal a ruido (SNR) y la tasa de error de bit (BER). A medida que el SNR aumenta, la probabilidad de error disminuye exponencialmente en ambos esquemas de modulación.

[2] La modulación BPSK demostró mayor robustez frente al ruido en comparación con la 16-QAM, debido a que maneja únicamente dos fases de portadora, lo cual la hace menos susceptible a variaciones de amplitud y fase introducidas por el canal.

[3] La implementación de codificación de canal mejoró significativamente el desempeño del sistema, reduciendo la BER y evidenciando una ganancia de codificación promedio de entre 1.5 y 3 dB, lo que se traduce en una mayor confiabilidad en la transmisión de datos digitales.

[4] El uso de códigos Hamming y convolucionales resultó efectivo para la corrección de errores, siendo el primero apropiado para sistemas de baja complejidad y el segundo ideal para esquemas de alta tasa y modulación compleja como 16-QAM.

[5] El entorno de Simulink se consolidó como una herramienta poderosa para la enseñanza y el análisis de comunicaciones digitales, ya que permite visualizar en tiempo real los efectos de la modulación, la codificación y el ruido, facilitando la interpretación de resultados y la validación de conceptos teóricos.

REFERENCIAS

- [1] MicroPython, machine — functions related to the hardware, Documentación oficial, [En línea]. Disponible en: <https://docs.micropython.org/en/latest/library/machine.html>. [Accedido: 14-ago-2025].
- [2] Raspberry Pi Foundation, Getting started with MicroPython on Raspberry Pi Pico, Documentación oficial, [En línea]. Disponible en: <https://www.raspberrypi.com/documentation/microcontrollers/micropython.html>. [Accedido: 14-ago-2025].
- [3] MathWorks, *Communications Toolbox User's Guide*, MATLAB R2008a Documentation. Disponible en: <https://www.mathworks.com/help/com>
- [4] ANSI/TIA, TIA-232-F: Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange, Telecommunications Industry Association, 1997.
- [5] A. S. Tanenbaum and D. Wetherall, *Computer Networks*, 5th ed. Boston, MA, USA: Pearson, 2011.
- [6] Vasquez Rojas, D. (s.f.). *dylanrojas04 - Overview*. GitHub. <https://github.com/dylanrojas04>

[7] R. G. Gallager, "Low-Density Parity-Check Codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[8] J. Rugeles, Repositorio de prácticas de comunicación digital – I²C con Raspberry Pi Pico, Universidad Militar Nueva Granada. [En línea]. Disponible en: <https://github.com/jrugeles/I2C>