



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.edu.ec  
AMBATO-ECUADOR



**PROYECTO FINAL**

**UNIVERSIDAD TÉCNICA DE AMBATO**

**Facultad de Ingeniería en Sistemas, Electrónica e**

**Industrial “Proyecto Académico de Fin de Semestre**

Título: Simulación de los parámetros técnicos de propagación que ocurre en la Fibra óptica.

Carrera: Ingeniería en  
Telecomunicaciones

Unidad de Organización Curricular: Profesional

Línea de Investigación: Nanotecnología

Nivel y Paralelo: Octavo “A”

Alumnos participantes: Aldaz Saca Fabricio Javier

Balseca Castro Josué  
Guillermo

Chimba Amaya Cristian  
Orlando

Ibarra Rojano Gilber Andres

León Armijo Jean Carlos

Sivinta Almachi Jhon

Telenchana Tenelema Alex  
Roger

Toapanta Gualpa Paul Edwin

Módulo y Docente: Comunicaciones ópticas

Ing. Juan Pablo Pallo



## **I. INFORME DEL PROYECTO**

### **II. OBJETIVOS**

#### **1. OBJETIVO GENERAL**

- Realizar un programa en Python que simule los parámetros que ocurren en la fibra óptica cuando esta se transmite.

#### **2. OBJETIVOS ESPECIFICOS**

- Generar un programa indique la modulación digital que se usa en la propagación de la fibra óptica, así como trabaja en una red de fibra óptica.
- Investigar el funcionamiento de la propagación en la fibra y sus parámetros técnicos. el programa en Python
- Simular en Python un sistema de comunicación óptico y verificar como se propaga en el sistema.

### **III. INTRODUCCIÓN**

La fibra óptica, un componente fundamental en las redes de telecomunicaciones modernas, se caracteriza por su capacidad para transmitir datos a través de pulsos de luz. La simulación de los parámetros técnicos de propagación en la fibra óptica es esencial para comprender y optimizar su rendimiento en diversas aplicaciones. En este contexto, la programación en Python ofrece una herramienta versátil y potente para modelar estos fenómenos de manera eficiente. Hay algunos parámetros técnicos clave que afectan la propagación de la luz en la fibra óptica y cómo podemos simularlos utilizando Python.

### **IV. RESUMEN**

Este informe presenta el desarrollo de un programa en Python centrado en temas relevantes de propagación en las redes ópticas. El objetivo principal es lograr la comprensión de los conceptos que implican para que se puedan realizar en las redes GPON. Para lograrlo, se establecieron objetivos específicos para la generación de un programa para identificar los parámetros importantes para el envío del laser optico y para su implementación en Python, y la simulación de un



sistema de transmisión de datos en un canal ruidoso para analizar y minimizar problemas.

## **I. ABSTRACT**

This report details the creation of a Python program focusing on relevant topics in digital communications to achieve a better understanding of their operation. The primary objective is to develop a comprehensive understanding of digital communication concepts. To attain this goal, specific objectives have been outlined, including the creation of a program to identify types of digital modulation in the transmission of images and videos, the investigation of error detection and control systems for implementation in Python, and the simulation of a data transmission system in a noisy channel to analyze and minimize potential issues.

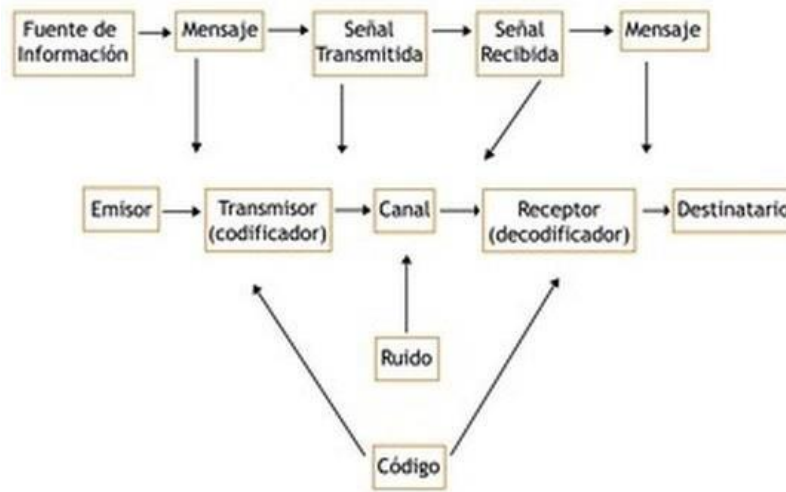
## **II. MARCO TEÓRICO**

### **Teoría de la Información**

La teoría matemática de la información fue propuesta en 1949 por el matemático e ingeniero Claude Shannon y el biólogo Warren Weaver. No obstante, es el resultado de investigaciones iniciadas casi treinta años antes por científicos como Andrei Markovi y Ralph Hartley, este último conocido por ser uno de los primeros representantes del lenguaje binario.

También conocida como teoría matemática de la comunicación, es un planteamiento que estudia el procesamiento y medición de datos en la transmisión de una información. El proceso de comunicación planteado por sus creadores establece el flujo de un mensaje entre un emisor y un receptor a través de un canal determinado.

La teoría de la información también se encarga de medir y representar la información, así como la capacidad de procesamiento de los sistemas de comunicación para transmitir dicha información. Es, además, una rama de la teoría de la probabilidad matemática. [1]



*Ilustración 1 Modelo de Shannon*

$$I(x_i) = -3.32 \log_{10} P(x_i) \text{ [bits]}$$

### **Fuentes Discretas sin Memoria**

#### **Librería tkinter**

Tkinter es una librería del lenguaje de programación Python y funciona para la creación y el desarrollo de aplicaciones de escritorio. Esta librería facilita el posicionamiento y desarrollo de una interfaz gráfica de escritorio con Python. Tkinter es el paquete estándar de Python para interactuar con Tk. [1]

#### **Librería PIL**

Pillow, que anteriormente se conocía como Python Imaging Library (PIL), es una biblioteca Python de código abierto ampliamente utilizada para manipular imágenes de manera eficiente. En este tutorial, aprenderás cómo usar Pillow para realizar diversas operaciones de procesamiento de imágenes en Python. [2]

#### **Librería OS**

En Python, el módulo os es una biblioteca estándar que proporciona una interfaz para interactuar con el sistema operativo en el que se ejecuta el programa. Permite realizar operaciones relacionadas con la gestión de archivos, directorios, rutas, variables de entorno y otras funcionalidades del sistema operativo. [3]



### **Librería Socket**

Socket es un módulo estándar del lenguaje de programación Python (y de otros muchos) que proporciona una interfaz de bajo nivel que permite conexiones TCP/IP y UDP. Esto ofrece soluciones sencillas a problemas sencillos. Si necesitamos trabajar con servidor, podremos usar el módulo socketserver. [4]

### **Librería Struct**

Este módulo realiza conversiones entre valores de Python y estructuras C representadas como objetos bytes de Python. Se puede utilizar para el tratamiento de datos binarios almacenados en archivos o desde conexiones de red, entre otras fuentes.

### **Librería Numpy**

Es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas. [5]



*Ilustración 2 Librería Numpy*

### **Librería Matplotlib**

Matplotlib es una librería de Python especializada en la creación de gráficos en dos dimensiones. Permite crear y personalizar los tipos de gráficos más comunes, entre ellos: Diagramas de barras. [6]



*Ilustración 3 Librería Matplotlib*

### **Librería Base 64**

El módulo base64 contiene funciones para traducir datos binarios en un subconjunto de ASCII adecuado para la transmisión utilizando protocolos de texto plano. [7]

### **Librería Scipy**

SciPy es una librería de Python para matemáticas, ciencias e ingeniería que proporciona módulos para optimización, álgebra lineal, integración, interpolación y otras tareas científicas y de ingeniería. [8]



*Ilustración 4 Librería Scipy*

### **Librería de MoviePy**

MoviePy es una biblioteca de Python que permite editar y crear videos de manera sencilla. Es una herramienta de código abierto que está disponible para Windows, macOS y Linux. [9]

### **Librería Tkinter as tk**

El paquete tkinter («interfaz Tk») es la interfaz por defecto de Python para el kit de herramientas de GUI Tk. Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, así como en sistemas Windows (Tk en sí no es parte de Python, es mantenido por ActiveState). [10]

### **Librería Tkinter as ttk**

El nuevo tema Tk (Ttk) es una nueva familia de widgets de Tk que proveen una mejor apariencia en diferentes plataformas mas que varios de los widgets clásicos de Tk. Ttk es distribuido como parte de Tk estando disponible a partir de la versión 8.5. Enlaces para Python son entregados en un módulo aparte, tkinter. Ttk. [11]



---

## **Librería TextwRap**

Wrap(text) Envuelve el párrafo individual en text (una cadena) para que cada línea tenga como máximo width caracteres de largo. Todas las opciones de envoltura se toman de los atributos de la instancia TextWrapper . Retorna una lista de líneas de salida, sin las nuevas líneas finales. [12]

## **Librería sounddevice**

A lo largo de entradas anteriores hemos visto el modo de efectuar tanto reproducciones de archivos de audio, así como la grabación de los mismos. No obstante, puede suceder en ocasiones, que queramos realizar dichas acciones, al mismo tiempo que se ejecuta otro código de forma simultánea (reproduciéndose o grabándose audio en segundo plano). Un procedimiento fácil y cómodo de realizar esta acción nos lo proporciona la librería «sounddevice» (que instalaremos previamente con «pip install sounddevice») que usaremos conjuntamente con «soundfile» («pip install soundfile») tanto para reproducir audio en segundo plano, así como para grabarlo. [13]



## **Multiplexación**

### **III. LISTADO DE EQUIPOS Y MATERIALES Y**

#### **METODOLOGIA MATERIALES**

- Laptop
- Apuntes de la materia
- Software de Simulación (Python)

#### **METODOLOGIA**

Primeramente, identificamos e importamos las librerías que necesitaremos en el transcurso de nuestro proyecto, cualquier librería que posteriormente se necesite se la agregará





**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cda. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.edu.ec  
AMBATO-ECUADOR



```
1 from tkinter import *
2 from PIL import Image, ImageTk
3 from tkinter import filedialog
4 from tkinter import scrolledtext
5 import os
6 import socket
7 import struct
8 import numpy as np
9 import matplotlib.pyplot as plt
10 import base64
11 import scipy.interpolate
12 import moviepy.editor as mp
13 import tkinter as tk
14 from tkinter import ttk
15 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
16 import tkinter as tk
17 from tkinter.font import Font
18 from tkinter import messagebox
19 from PIL import ImageTk, Image
20 import textwrap
21 from tkinter import font
22 import matplotlib
23 matplotlib.use("TkAgg")
24 from matplotlib.figure import Figure
25 from scipy.fft import fft
26 from tkinter import Tk, Label, Entry, Button
27 from moviepy.editor import VideoFileClip
28 import random
29 import sounddevice as sd
```

*Ilustración 11 librerías*

Creación de la pantalla de inicio de datos personales de los integrantes y la distribución de los botones

```
1384 tema=Frame(raiz, bg="white", height=50, width=1000)
1385 tema.place(x=1,y=1)
1386 mtema=Label(tema, bg="white", text="ENVIO DE IMÁGENES", foreground="black", font=("STENCIL", 20)).place(x=600,y=1)
1387 rojo=Image.open(r"C:\Users\ASUS\Desktop\Proyecto - Python Tx Rx - Modulaciones\rojo.png")
1388 rojo2=rojo.resize((500,100))
1389 rojo3=ImageTk.PhotoImage(rojo2)
1390 fondo=Label(raiz, image=rojo3, height="10", width="500")
1391 fondo.place(x=500,y=60)
1392 roj=Label(tema, image=rojo3, height="10", width="1000")
1393 opciones=Frame(raiz, bg="white", height="150", width="500")
1394 opciones.place(x=500,y=100)
1395 mselc=Label(opciones, bg="white", text="FUNCIÓN", foreground="black", font=("calibri", 12)).place(x=50,y=1)
1396 mselc=Label(opciones, bg="white", text="TIPO DE ENVÍO", foreground="black", font=("calibri", 12)).place(x=320,y=1)
1397 tipo=Frame(opciones, bg="white", width="200", height="100")
1398 tipo.place(x=50,y=25)
1399 cliente=Radiobutton(tipo, text="Transmisor", value=1, variable=seleccion, command=mostrar, width="12")
1400 cliente.place(x=10,y=10)
1401 servidor=Radiobutton(tipo, text="Receptor", value=2, variable=seleccion, command=mostrar, width="12")
1402 servidor.place(x=10,y=30)
1403 medio=Frame(opciones, bg="white", height="100", width="200")
1404 medio.place(x=320,y=25)
1405 wifi=Radiobutton(medio, text="Inalámbrico", value=3, variable=seleccion2, command=mostrar, width="12")
1406 wifi.place(x=30,y=10)
1407 cable=Radiobutton(medio, text="Alámbrico", value=4, variable=seleccion2, command=mostrar, width="12")
1408 cable.place(x=30,y=30)
1409 sec=Button(opciones, bg="white", font=("calibri", 11), text="Imagen", fg="black", command=selimagen)
1410 secv=Button(opciones, bg="white", font=("calibri", 11), text="Video", fg="black", command=selarchivo)
1411 iniciarr=Button(opciones, bg="white", font=("calibri", 11), text="Enviar", fg="black", command=enviar)
1412 mim=Label(raiz, bg="white", foreground="black", font=("calibri", 22))
1413 mim.place(x=494,y=295)
1414 simagen=Frame(raiz, bg="gray81", height="300", width="500")
1415 simagen.place(x=493, y=340)
1416 mselc=Label(simagen, bg="gray81", foreground="black", font=("calibri", 20))
1417 mselc.place(x=300,y=1)
1418 mcontrol=Label(simagen, bg="gray81", foreground="black", font=("calibri", 20))
1419 mcontrol.place(x=300,y=150)
```

*Ilustración 12 programación carátula*



### Programación de los botones para el uso de la interfaz grafica

```
1484 png=Radiobutton(opciones, text="png", value=1, variable=seleccion3, command=formato)
1485 jpg=Radiobutton(opciones, text="jpg", value=2, variable=seleccion3, command=formato)
1486 jpeg=Radiobutton(opciones, text="jpeg", value=3, variable=seleccion3, command=formato)
1487 gif=Radiobutton(opciones, text="gif", value=4, variable=seleccion3, command=formato)
1488 mp4=Radiobutton(opciones, text="mp4", value=5, variable=seleccion3, command=formato)
1489
1490
1491 ask=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1492 fsk=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1493 psk=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1494 qam=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1495 ofdm=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1496 ask2=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1497 fsk2=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1498 psk2=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1499 qam2=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1500 ofdm2=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1501 errores=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1502 ruido=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1503 Simulacion=Button(simagen,bg="firebrick4", font=("Calibri", 12, "bold"), relief="flat", width
1504
1505 raiz.after(200, recibir)
1506 raiz.mainloop()
```

*Ilustración 13 configuración de los botones*

### Configuración de las ip de la Pc transmisora y receptora



```
ip_localwifi="192.168.100.9" #ip de la red wifi cambia dependiendo la red
#ip_localcable="192.168.100.11" #ip de la pc
ip_pc2wifi="192.168.100.19"
#ip_pc2cable="192.168.100.10"
hj=0
kl=0
ty=''
ry=""
di1=200
di2=220
por2=None
k=0
def receive_file_size(sck: socket.socket):
    fmt = "<Q"
    expected_bytes = struct.calcsize(fmt)
    received_bytes = 0
    stream = bytes()
    while received_bytes < expected_bytes:
        chunk = sck.recv(expected_bytes - received_bytes)
        stream += chunk
        received_bytes += len(chunk)
    filesize = struct.unpack(fmt, stream)[0]
    return filesize
```

*Ilustración 14 programación de las ip*

Se define la apariencia y el comportamiento de la interfaz gráfica que se usará para la transmisión y recepción de datos.



```
133     print("Transmisor")
134     iniciarr.place(x=100,y=90)
135     sec.place(x=300,y=90)
136     ####
137     secv.place(x=400,y=90)
138     ####
139     jpg.place_forget()
140     png.place_forget()
141     jpeg.place_forget()
142     gif.place_forget()
143     mp4.place_forget()
144     mselec.configure(text="Modulaciones:")
145     mcontrol.configure(text="Controles:")
146     ask.place(x=300, y=40)
147     fsk.place(x=403, y=40)
148     psk.place(x=300, y=75)
149     qam.place(x=403, y=75)
150     ofdm.place(x=300, y=110)
151     errores.place(x=300, y=190)
152     ruido.place(x=300, y=225)
153     Simulacion.place(x=300, y=260)
154     modul=1
155     if (seleccion.get()==2):
156         print("Receptor")
157         iniciarr.place_forget()
158         sec.place_forget()
159         secv.place_forget()
160         png.place(x=30,y=110)
161         jpg.place(x=130,y=110)
162         jpeg.place(x=230,y=110)
163         gif.place(x=330,y=110)
164         mp4.place(x=430,y=110)
165         mselec.configure(text="Demodulaciones:")
166         mcontrol.configure(text="Controles:")
167         hj=2
```

*Ilustración 15 programación de Tx y Rx*

Esta parte del código carga y muestra una vista previa de la imagen o el primer fotograma del video en la interfaz gráfica, dependiendo de la extensión del archivo proporcionado y de igual manera verifica que haya subido en los formatos aceptados



```
214 # Verificar la extensión del archivo para determinar si es una imagen o video
215 _, extension = os.path.splitext(g)
216 extension = extension.lower()
217
218 ✓ if extension in ['.png', '.jpg', '.jpeg', '.gif', '.ico']:
219     # Es una imagen
220     por = Image.open(g).resize((250, 250))
221     por2 = ImageTk.PhotoImage(por)
222 ✓ elif extension in ['.mp4', '.avi', '.mov', '.mkv']:
223     # Es un video
224     por_clip = mp.VideoFileClip(g)
225     por_frame = por_clip.get_frame(0)
226     por_image = Image.fromarray(por_frame)
227     por2 = ImageTk.PhotoImage(por_image)
228     por_clip.close()
229 ✓ else:
230     # Otro tipo de archivo no compatible
231     print(f"Error: Archivo no compatible con extensión {extension}")
232     return
233
234 imagen1 = Label(simagen, bg='white', image=por2, height="250", width="250")
235 imagen1.place(x=1, y=1)
236
```

*Ilustración 16 programación de los formatos de video e imágenes*

Ahora convertimos a cadena de bits la imagen que se haya subido, para posteriormente ser procesados para un tipo de transmisión o comunicación





```
237 # Convertir a base64
238 with open(g, "rb") as file:
239     converted_string = base64.b64encode(file.read())
240 fg = str(converted_string.decode("utf-8"))
241 longitud = len(fg)
242 longitud1 = len(fg)
243 print(longitud)
244 if (longitud>100000):
245     longitud=100000
246 for j in range (0, longitud):
247     res=(str(format(ord(fg[j]),"b")))
248     long=len(res)
249     while long < 8 :
250         res='0'+res
251         long=len(res)
252     ty=ty+res
253     bits=longitud1*8
254 ----
255 raiz.after(200, recibir)
256 def selimagen():
257     global ficheros, por, por2,ty
258     longitud=0
259     res=0
260     ty=""
261     ficheros = filedialog.askopenfilename(title='Abrir', filetypes=(("PNG", "*.png"), ("JPG", "*.jpg")
262     por=Image.open(ficheros).resize((250,250))
263     por2=ImageTk.PhotoImage(por)
264     imagen1=Label(simagen, bg='white', image=por2, height="250", width="250")
265     imagen1.place(x=1,y=1)
266     mim.configure(text="Imagen a Enviar")
267     with open(ficheros, "rb") as image2string:
268         converted_string = base64.b64encode(image2string.read())
269         fg=str(converted_string.decode("utf-8"))
270         longitud=len(fg)
271         longitud1=len(fg)
```

*Ilustración 17 conversión de imágenes a cadena de bits*

Ahora definimos señales moduladas a partir de una secuencia de bits. La modulación ASK implica cambiar la amplitud de una señal portadora, mientras que la modulación FSK implica cambiar la frecuencia de la señal portadora.



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.e  
AMBATO-ECUADOR



```
330     print("Modulacion ask")
331     v=list(map(int, ty))
332     dim=100
333     vx=[]
334     plt.subplot(3,1,1)
335     plt.plot(vx)
336     for i in range (int(di1),int(di2)):
337         f=np.ones(dim)
338         x=f*v[i]
339         vx=np.concatenate((vx,x))
340     plt.subplot(3,1,1)
341     plt.plot(vx)
342     dim2=len(vx)
343     print("La longitud es:", dim2)
344     t=np.linspace(int(di1),int(di2),dim2)
345     print(t)
346     f1=5
347     plt.subplot(3,1,2)
348     w1=2*np.pi*f1*t
349     y1=np.cos(w1)
350     plt.plot(t,y1)
351     plt.subplot(3,1,3)
352     mult=(vx*y1)
353     plt.plot(t,mult)
354     plt.show()
355
356     else:
357         print ("Demodulacion ask")
358         v=list(map(int, ty))
359         dim=100
360         vx=[]
361         plt.subplot(3,1,1)
362         plt.plot(vx)
363         for i in range (int(di1),int(di2)):
364             f=np.ones(dim)
```

*Ilustración 18 modulaciones ASK y FSK*



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.e  
AMBATO-ECUADOR



```
486     print("Demodulacion PSK")
487     print("Modulacion psk")
488     v=list(map(int, ty))
489     dim=100
490     vx=[]
491     plt.subplot(3,1,1)
492     plt.plot(vx)
493     for i in range (int(di1),int(di2)):
494         f=np.ones(dim)
495         x=f*v[i]
496         vx=np.concatenate((vx,x))
497     plt.subplot(4,1,4)
498     plt.plot(vx)
499     dim2=len(vx)
500     t=np.linspace(int(di1),int(di2),dim2)
501     f1=2
502     plt.subplot(4,1,3)
503     w1=2*np.pi*f1*t
504     y1=np.cos(w1)
505     plt.plot(t,y1)
506     plt.subplot(4,1,2)
507     f2=2
508     w2=2*np.pi*f2*t
509     y2=np.sin(w2)
510     plt.plot(t,y2)
511     plt.subplot(4,1,1)
512     res=((y2*vx)-(y1*vx)+(y1))
513     plt.plot(t,res)
514     plt.show()
515 def qam_mod():
516     global ty
517     if (modul==1):
518         print("Modulacion qam")
519         print (ty)
520         v=list(map(int, ty))
```

*Ilustración 19 modulaciones PSK y QAM*

En este apartado se hace el control y detección de errores





```
934 QAM = Mapping(bits_SP)
935 print ("First 5 QAM symbols and bits:")
936 print (bits_SP[:5,:])
937 print (QAM[:5])
938 def OFDM_symbol(QAM_payload):
939     symbol = np.zeros(K, dtype=complex) # the overall K subcarriers
940     symbol[pilotCarriers] = pilotValue # allocate the pilot subcarriers
941     symbol[dataCarriers] = QAM_payload # allocate the pilot subcarriers
942     return symbol
943 OFDM_data = OFDM_symbol(QAM)
944 print ("Number of OFDM carriers in frequency domain: ", len(OFDM_data))
945 def IDFT(OFDM_data):
946     return np.fft.ifft(OFDM_data)
947 OFDM_time = IDFT(OFDM_data)
948 print ("Number of OFDM samples in time-domain before CP: ", len(OFDM_time))
949 def addCP(OFDM_time):
950     cp = OFDM_time[-CP:] # take the last CP samples ...
951     return np.hstack([cp, OFDM_time]) # ... and add them to the beginning
952 OFDM_withCP = addCP(OFDM_time)
953 print ("Number of OFDM samples in time domain with CP: ", len(OFDM_withCP))
954 def channel(signal):
955     convolved = np.convolve(signal, channelResponse)
956     signal_power = np.mean(abs(convolved**2))
957     sigma2 = signal_power * 10**(-SNRdb/10) # calculate noise power based on signal power and SNR
958     print ("RX Signal power: %.4f. Noise power: %.4f" % (signal_power, sigma2))
959     # Generate complex noise with given variance
960     noise = np.sqrt(sigma2/2) * (np.random.randn(*convolved.shape)+1j*np.random.randn(*convolved.shape))
961     return convolved + noise
962 OFDM_TX = OFDM_withCP
963 OFDM_RX = channel(OFDM_TX)
964 def grafic():
965     y1=list(map(int, ty))
966     t = np.arange(0,len(y1[int(di1):int(di2)]))/2, 0.5)
967     plt.subplot(3, 1, 1)
968     plt.plot(t, y1[int(di1):int(di2)], drawstyle='steps-post')
969     plt.subplot(3,1,2)
```

### *Ilustración 20 control y detección de errores*

Por otro lado aquí se hace la simulación de transmisión de información desde un canal ruidoso, donde se podrá añadir un audio y generarlo como ruido para posteriormente recuperar la señal sinruido



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdl. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.ec  
AMBATO-ECUADOR



```
1106 def plot_signal(codigo_binario, label, ax, color):
1107     ax.clear() # Limpiar el eje antes de graficar
1108     ax.step(range(1, len(codigo_binario) + 1), codigo_binario, where='post', color=color, label=label)
1109     ax.axhline(0, color='red', linestyle='--', linewidth=2) # Línea roja en el punto cero
1110     ax.set_xlabel('Tiempo')
1111     ax.set_ylabel(f'N. Señal')
1112     ax.set_xticks(range(1, len(codigo_binario) + 2)) # Mostrar números del 1 al 10 en el eje x
1113     ax.set_ylim(-1.5, 1.5) # Ajustar el rango del eje y de 1 a -1
1114     ax.legend() # Mostrar leyenda
1115     ax.grid(True) # Habilitar la cuadrícula
1116
1117 def on_plot_button_click(codigo_binario):
1118     original_signal = original(codigo_binario)
1119
1120     # Graficar la secuencia original (azul)
1121     plot_signal(original_signal, 'Error', ax1, color='red')
1122
1123     canvas1.draw() # Redibujar el lienzo para la secuencia original
1124
1125 def correccion_error(codigo_binario):
1126     original_signal = original(codigo_binario)
1127     nueva_secuencia = modificar_secuencia(original_signal)
1128
1129     # Graficar la nueva secuencia modificada (roja)
1130     plot_signal(nueva_secuencia, 'Correccion', ax2, color='green')
1131
1132     canvas2.draw() # Redibujar el lienzo para la secuencia modificada
1133
1134 def espectro(codigo_binario):
1135
1136     original_signal = original(codigo_binario)
1137     nueva_secuencia = modificar_secuencia(original_signal)
```

*Ilustración 21 transmisión desde un canal ruidoso*



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.edu.ec  
AMBATO-ECUADOR



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def generar_bits(num_bits):
5     return np.random.randint(0, 2, num_bits)
6
7 def codificar_grey(bits, num_bits_grupo):
8     codificacion_grey = []
9     for i in range(0, len(bits), num_bits_grupo):
10         grupo = bits[i:i + num_bits_grupo]
11         codificacion_grey.extend(codificar_grupo_grey(grupo))
12     return np.array(codificacion_grey)
13
14 def codificar_grupo_grey(grupo):
15     codificacion_grupo = []
16     codificacion_grupo.append(grupo[0])
17     for i in range(1, len(grupo)):
18         codificacion_grupo.append(grupo[i] ^ grupo[i - 1])
19     return codificacion_grupo
20
21 def graficar_secuencia(bits, bits_codificados):
22     plt.subplot(2, 1, 1)
23     plt.stem(range(len(bits)), bits, linefmt='b-', markerfmt='bo', basefmt='b-', use_line_collection=True)
24     plt.title('Secuencia Binaria')
25     plt.xlabel('Tiempo (bits)')
26     plt.ylabel('Valor')
27
28     plt.subplot(2, 1, 2)
29     plt.stem(range(len(bits_codificados)), bits_codificados, linefmt='r-', markerfmt='ro', basefmt='r-', use_line_collection=True)
```

*Ilustración 1 código*



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.ec  
AMBATO-ECUADOR



```
1 import math
2 import numpy as np
3 import matplotlib.pyplot as plt
4 #-----Variables de la reflexion-----
5 angulo_critico =0
6 indice1 =0
7 indice2 =0
8 indice3 =0
9 angulo_entrada =0
10 angulo_reflexion =0
11 #-----Menu-----
12 print("Propagacion de la luz en la fibra\n")
13 print("Seleccione los valores para las siguientes variable \n")
14 indice1 = float(input("Indice de reflexion del medio inicial (n): "))
15 indice2 = float(input("Indice de reflexion del nucleo de la fibra (n1): "))
16 indice3 = float(input("Indice de reflexion del nucleo de la fibra (n2): "))
17 reflexiones = int(input("Ingrese el numero de reflexiones: "))
18
19 print(indice1)
20 print(indice2)
21 print(indice3)
22 #-----Calculos realizados-----
23 print("Calculos\n")
24 print("-----1.-Angulo Critico-----\n")
25 angulo_critico=math.asin(indice1/indice2)
26 angulo_critico_grados = math.degrees(angulo_critico)
27 print(f"El angulo critico es: {angulo_critico_grados:.2f} °")
28
29 print("-----1.-Cono de aceptacion-----\n")
30 angulo_entrada=math.asin(math.cos(angulo_critico*indice2/indice3))
```



## RESULTADOS

```
1  import sys
2  from Pyqt5.Qtwidgets import QApplication, QWidget, QVBoxLayout, QLineEdit, QValidator
3
4  class BinaryValidator(QValidator):
5      def validate(self, input_str, pos):
6          if all(c in '01' for c in input_str):
7              return (QValidator.Acceptable, input_str, pos)
8          elif input_str == '' or input_str[-1] in '01':
9              return (QValidator.Intermediate, input_str, pos)
10         else:
11             return (QValidator.Invalid, input_str, pos)
12
13 class MainWindow(QWidget):
14     def __init__(self):
15         super().__init__()
16
17         # Crear un QLineEdit
18         self.line_edit = QLineEdit(self)
19
20         # Crear un validador para permitir solo '0' y '1'
21         binary_validator = BinaryValidator()
22         self.line_edit.setValidator(binary_validator)
23
24         # Crear un diseño vertical
25         layout = QVBoxLayout(self)
26         layout.addWidget(self.line_edit)
27
28 if __name__ == '__main__':
29     app = QApplication(sys.argv)
30     window = MainWindow()
```



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.ec  
AMBATO-ECUADOR



```
1  ✓ from PyQt5.Qtwidgets import QApplication, QDialog, QMainWindow, QlineEdit
2  from PyQt5.uic import loadUi
3  from PyQt5.QtGui import QDoubleValidator
4  import sys
5  import math
6  import numpy as np
7  import matplotlib.pyplot as plt
8  import tkinter as tk #Para la ventana de Tkinter
9  import tkinter.filedialog as tkf #Cuadro de dialogo de Tkinter
10 import time
11 from scipy.special import erfc
12
13 #-----Variables de la reflexion-----
14 angulo_critico =0
15 indice1 =0
16 indice2 =0
17 indice3 =0
18 angulo_entrad =0
19 angulo_reflexion =0
20 vector = []
21 cont = 0
22
23
24  ✓ class Menu1Dialog(QDialog):
25  ✓     def __init__(self):
26         super(Menu1Dialog, self).__init__()
27         loadUi('menu1.ui', self) # Cargar la interfaz desde el archivo .ui
28
29
```



## **CONCLUSIONES**

- El programa desarrollado se puede identificar la modulación digital utilizada en la propagación de la fibra óptica refleja una comprensión profunda de los principios de modulación y demuestra habilidades prácticas en la implementación de tecnologías específicas.
- La investigación realizada sobre los sistemas ópticos en el software Python proporciona una base teórica que ayuda al desarrollo práctico y la implementación del software.
- La verificación de la propagación en el sistema simulado permite evaluar la eficacia del diseño, realizar ajustes según sea necesario y obtener información valiosa sobre el comportamiento del sistema en condiciones controladas.

## **RECOMENDACIONES**

- Se recomienda comprender mejor cómo funcionan los diferentes tipos de modulación digital.
- Para el código de Python se debe tener en cuenta las librerías apropiadas en las cuales nos permitan una mejor ejecución y simulación de los parámetros de fibra óptica a los que se quiere simular, ya hay librerías que no constan con todas las especificaciones para el desarrollo del programa en la cual se quiere simular.
- Para la verificación de la propagación del sistema de fibra óptica se recomienda tener en claro el tipo de sistema el cual se va a querer simular, tanto sus parámetros técnicos como la parte de los cálculos ya que es importante que estos vayan de la mano para el funcionamiento.





## FE DE ERRATAS

Existieron inconvenientes para la identificación de los parámetros técnicos, y en el sistema de comunicación por óptica por satélite, ya que en la parte de los parámetros técnicos sobre la propagación de luz los cálculos no eran los correctos debido a las librerías en las cuales se estaban utilizando, y estas no eran las adecuadas.

## BIBLIOGRAFÍA

[1 KeepCoding, 4 12 2023. [En línea]. Available: <https://keepcoding.io/blog/que-es-tkinter/#:~:text=Tkinter%20es%20una%20librer%C3%ADa%20del,Python%20para%20interactuar%20con%20Tk..> [Último acceso: 15 01 2024].

[2 C. Python, 17 09 17. [En línea]. Available: <https://codigospython.com/procesamiento-de-imagenes-con-pillow-pil-en-python/#:~:text=Pillow%2C%20que%20anteriormente%20se%20conoc%C3%ADa,procesamiento%20de%20im%C3%A1genes%20en%20Python..> [Último acceso: 15 01 2024].

[3 Tutz, 15 03 2022. [En línea]. Available: <https://tutz.tv/python/os>. [Último acceso: 15 01 2024].

[4 Escribecodigo, 22 10 2019. [En línea]. Available: <https://www.escribecodigo.com/sockets-en-python/#:~:text=Socket%20es%20un%20m%C3%B3dulo%20est%C3%A1ndar,podremos%20usar%20el%20m%C3%B3dulo%20socketserver..> [Último acceso: 15 01 2024].

[5 A. c. Alf, 20 10 2020. [En línea]. Available: <https://aprendeconalf.es/docencia/python/manual/numpy/>. [Último acceso: 15 01 2024].

[6 A. c. Alf, 4 10 2020. [En línea]. Available: <https://aprendeconalf.es/docencia/python/manual/matplotlib/#:~:text=Matplotlib%20es%20una%20librer%C3%ADa%20de,Diagramas%20de%20barras>. [Último acceso: 15 01 2024].

[7 Python3, 17 08 2019. [En línea]. Available: <https://rico-schmidt.name/pymotw-3/base64/index.html#:~:text=El%20m%C3%B3dulo%20base64%20contiene%20funciones,utilizando%20protocolos%20de%20texto%20plano..> [Último acceso: 15 01 2024].

[8 inLinkedin, 3 05 2023. [En línea]. Available: <https://es.linkedin.com/pulse/scipy-el-aliado-de-un-matem%C3%A1tico-luis-alberto-oraa->





**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.edu.ec  
AMBATO-ECUADOR



garcia#:~:text=SciPy%20es%20una%20librer%C3%ADa%20de,tareas%20cient%C3%ADficas%20y%20de%20ingenier%C3%ADa.. [Último acceso: 15 01 2024].

[9 M. Python, 30 09 2023. [En línea]. Available: <https://codigospython.com/edicion-y-creacion-de-videos-con-moviepy-en-python/#:~:text=MoviePy%20es%20una%20biblioteca%20de,para%20Windows%2C%20macOS%20y%20Linux..> [Último acceso: 15 01 2024].

[1 I. d. Python, 02 12 2021. [En línea]. Available: [https://docs.python.org/es/3/library/tkinter.html#:~:text=El%20paquete%20tkinter%20\(%C2%ABinterfaz%20Tk,%2C%20es%20mantenido%20por%20ActiveState\)..](https://docs.python.org/es/3/library/tkinter.html#:~:text=El%20paquete%20tkinter%20(%C2%ABinterfaz%20Tk,%2C%20es%20mantenido%20por%20ActiveState)..) [Último acceso: 15 01 2024].

[1 I. d. Python, 14 03 2022. [En línea]. Available: [https://docs.python.org/es/3/library/tkinter.html#:~:text=El%20nuevo%20tema%20Tk%20\(Ttk,ttk%20..](https://docs.python.org/es/3/library/tkinter.html#:~:text=El%20nuevo%20tema%20Tk%20(Ttk,ttk%20..) [Último acceso: 15 01 2024].

[1 P. Docs, 06 08 2021. [En línea]. Available: [https://docs.python.org/es/3/library/textwrap.html#:~:text=wrap\(text\)&text=%C2%B6-Envuelve%20el%20p%C3%A1rrafo%20individual%20en%20text%20\(una%20cadena\)%20para%20que,sin%20las%20nuevas%20l%C3%ADneas%20finales..](https://docs.python.org/es/3/library/textwrap.html#:~:text=wrap(text)&text=%C2%B6-Envuelve%20el%20p%C3%A1rrafo%20individual%20en%20text%20(una%20cadena)%20para%20que,sin%20las%20nuevas%20l%C3%ADneas%20finales..) [Último acceso: 15 01 2024].

[1 E. p. Chapuzas, 02 12 2023. [En línea]. Available: <https://programacionpython80889555.wordpress.com/2020/10/22/reproduccion-y-grabacion-asincronica-de-audio-en-python-con-sounddevice-y-soundfile/>. [Último acceso: 15 01 2024].

[1 «Tipos de antenas, características y funcionamiento.» [En línea]. Available: <https://www.onulec.com/tipos-de-antenas-caracteristicas-y-funcionamiento/>.

[1 F. Ucha, Marzo 2009. [En línea]. Available: <https://www.definicionabc.com/comunicacion/antena.php>. [Último acceso: 2023].

[1 «Tipos de antenas y funcionamiento.» [En línea]. Available: <https://www.tesamerica.com/tipos-antenas-funcionamiento/>.

[1 Prometec, «Prometec,» 13 8 2009. [En línea]. [Último acceso: 14 01 2024].

[1 N. Electronics, 25 2 2014. [En línea]. Available: <https://naylampmechatronics.com/oled/83-display-oled-096-spi-12864-ssd1306.html>. [Último acceso: 14 01 2024].

[1 A. Tecnologia, 14 7 2008. [En línea]. Available: <https://www.areatecnologia.com/electricidad/pulsador.html>. [Último acceso: 14 1 2024].



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: carrera.industrial@uta.edu.ec  
AMBATO-ECUADOR



- [2 O. BBVA, 13 09 2019. [En línea]. Available: 0] <https://www.bbvaopenmind.com/tecnologia/innovacion/la-baquelita-el-primer-plastico-sintetico-que-transformo-el-mundo/#:~:text=La%20baquelita%20fue%20el%20primer,sus%20usos%20comenzaron%20a%20proliferar..> [Último acceso: 14 01 2024].
- [2 D. C. Ruiz, 4 09 2015. [En línea]. Available: 1] <https://prezi.com/1ydebr2pwq9e/la-importancia-de-la-soldadura-con-estano-en-la-electronica/#:~:text=El%20esta%C3%B1o%20que%20se%20utiliza,el%20fen%C3%B3meno%20denominado%20soldadura%20fr%C3%ADa..> [Último acceso: 14 01 2024].
- [2 Promat, 23 10 2010. [En línea]. Available: 2] <https://www.promart.pe/blog/diferencia-melamina-mdf-cual-escoger/#:~:text=El%20MDF%20o%20como%20sus,aporte%20de%20una%20mayor%20densidad..> [Último acceso: 14 01 2024].
- [2 Fluke, 7 10 2012. [En línea]. Available: 3] [https://www.fluke.com/es-es/informacion/blog/electrica/que-es-un-multimetro-digital/#:~:text=Un%20mult%C3%A1metro%20digital%20\(DMM\)%20es,las%20industrias%20el%C3%A9ctricas%20y%20electr%C3%B3nicas..](https://www.fluke.com/es-es/informacion/blog/electrica/que-es-un-multimetro-digital/#:~:text=Un%20mult%C3%A1metro%20digital%20(DMM)%20es,las%20industrias%20el%C3%A9ctricas%20y%20electr%C3%B3nicas..) [Último acceso: 14 01 2024].
- [2 O. Elesctronics, 25 11 2016. [En línea]. Available: 4] <https://osakaelectronicsltda.com/blog/recomendaciones/que-es-una-fuente-de-voltaje.> [Último acceso: 14 01 2024].
- [2 Yuasa, 19 3 2014. [En línea]. Available: 5] <https://www.yuasa.es/informacion/automocion-comercial-servicios-nautica/funcionamiento-de-las-baterias/#:~:text=Las%20bater%C3%ADas%20son%20dispositivos%20que,encuentra%20dentro%20de%20un%20recipiente..> [Último acceso: 14 01 2024].
- [2 Intrabanch, 10 12 2017. [En línea]. Available: 6] <http://intrabach.org/instalacion-y-configuracion-basica-de-ubuntu-server/#:~:text=La%20distribuci%C3%B3n%20Ubuntu%20Server%20es,servidores%20y%20funcionalidades%20de%20virtualizaci%C3%B3n..> [Último acceso: 14 01 2024].
- [2 Uned, 24 04 2013. [En línea]. Available: 7] <http://e-spacio.uned.es/fez/eserv/taee:congreso-2004-1034/S1F05.pdf>. [Último acceso: 14 01 2024].
- [2 M. W. Doc, 19 06 2018. [En línea]. Available: 8] [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics). [Último acceso: 14 01 2024].



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**  
**PERÍODO ACADÉMICO: OCTUBRE 2023 - MARZO 2024**

Cdla. Universitaria (Predios Huachi) / Casilla 334 / Telefax: 03-2851894 – 2411537, Correo Electrónico: [carrera.industrial@uta.ec](mailto:carrera.industrial@uta.ec)  
AMBATO-ECUADOR

---

