



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS

ELECTRÓNICA E INDUSTRIAL

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
COMUNICACIONES**

**SEMINARIO DE GRADUACIÓN “SISTEMA Y REDES DE
COMUNICACIÓN INALÁMBRICA”**

TEMA:

LABORATORIO VIRTUAL A TRAVÉS DE LA PLATAFORMA
LABVIEW PARA LA REALIZACIÓN DE PRÁCTICAS DE
MODULACIÓN Y DEMODULACIÓN DIGITAL EN LA FISEI DE LA
UNIVERSIDAD TÉCNICA DE AMBATO.

Proyecto de Trabajo de Graduación. Modalidad: SEMINARIO DE GRADUACIÓN,
presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

AUTOR: Wilson Efraín Llumán Yucailla

PROFESOR REVISOR: Ing. Luis Alfredo Pomaquero Moreno

Ambato - Ecuador

Octubre 2012

APROBACIÓN DEL TUTOR

En calidad de Tutor del trabajo de investigación, nombrado por el H. Consejo Superior de Pregrado de la Universidad Técnica de Ambato:

CERTIFICO:

Que el trabajo de investigación: "**Laboratorio virtual a través de la plataforma labview para la realización de prácticas de modulación y demodulación digital en la FISEI de la Universidad Técnica de Ambato**", presentado por el Sr. Wilson Efraín Llumán Yucailla, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

Por lo tanto, autorizo la presentación del mismo a fin de que sea sometido a evaluación por la Comisión Calificadora asignada por el H. Consejo Directivo de la Facultad.

Ambato, 16 de Octubre del 2012

EL TUTOR

Ing. Luis Alfredo Pomaquero Moreno

AUTORIA DEL TRABAJO DE GRADUACIÓN

El presente trabajo de investigación titulado: "**Laboratorio virtual a través de la plataforma labview para la realización de prácticas de modulación y demodulación digital en la FISEI de la Universidad Técnica de Ambato**". Es absolutamente original, auténtico y personal en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, 16 de Octubre del 2012.

Wilson Efraín Llumán Yucailla

C.I. 1600576696

APROBACION DEL TRIBUNAL DE GRADO

El Tribunal de Grado, aprueba el Trabajo de Graduación, sobre el tema: **"LABORATORIO VIRTUAL A TRAVÉS DE LA PLATAFORMA LABVIEW PARA LA REALIZACIÓN DE PRÁCTICAS DE MODULACIÓN Y DEMODULACIÓN DIGITAL EN LA FISEI DE LA UNIVERSIDAD TÉCNICA DE AMBATO"**, elaborado por la señor Wilson Efraín Llumán Yucailla, egresado de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, el mismo que guarda conformidad con las disposiciones reglamentarias emitidas por la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

Ambato, 16 de Octubre del 2012

M.Sc. Oswaldo Eduardo Paredes Ochoa

PRESIDENTE DEL TRIBUNAL

Ing. Marco Antonio Jurado Lozada

PROFESOR CALIFICADOR

Ing. Mario García

PROFESOR CALIFICADOR

DEDICATORIA

A Dios que me ha permitido alcanzar mis metas siempre con humildad y constante dedicación, a mis padres por ser mis mejores amigos y a los docentes que me supieron compartir al máximo, sus conocimientos para convertirme en lo que soy ahora.

Wilson E. Lumán C.

AGRADECIMIENTO

Un especial agradecimiento a la Universidad Técnica de Ambato, en especial a mi segundo hogar como es la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, por abrirme sus puertas al conocimiento de muchos docentes, que día a día, se capacitan para impartir una enseñanza de calidad.

Al Ing. Luis Pomaquero, que me ayudo a potenciar todos mis conocimientos, para así, finalizar con orgullo y sacrificio el presente proyecto.

A toda mi familia que me apoyó cuando ya no tenía, el aliento para alcanzar este sueño.

Wilson E. Llumán

Índice General

Contenido	Página
Carátula.....	i
APROBACION DEL TUTOR	ii
AUTORIA DEL TRABAJO DE GRADUACION.....	iii
APROBACION DEL TRIBUNAL DE GRADO.....	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
Índice General.....	vii
Índice de Figuras	xi
Índice de Tablas	xvii
Índice de Ecuaciones.....	xvii
RESUMEN EJECUTIVO	xviii
INTRODUCCION.....	1
CAPÍTULO I.....	3
1. EL PROBLEMA DE INVESTIGACIÓN	3
1.1 Tema	3
1.2 Planteamiento del problema	3
1.2.1 Contextualización.....	3
1.2.2 Árbol de Problema	5
1.2.3 Análisis crítico	5
1.2.4 Prognosis.....	6
1.2.5 Formulación del problema.....	6
1.2.6 Preguntas directrices	7
1.2.7 Delimitación del problema.....	7
1.3 Justificación.....	8
1.4 Objetivos	9
1.4.1 Objetivo General:	9
1.4.2 Objetivos Específicos:.....	9

CAPÍTULO II	10
2. MARCO TEÓRICO	10
2.1 Antecedentes investigativos.....	10
2.2 Fundamentación legal.....	10
2.3 Gráfica de inclusión de las categorías fundamentales.....	11
2.4 Categorías fundamentales.....	11
2.4.1 Software y Aplicación.....	11
2.4.2 Entornos virtuales de programación.....	15
2.4.3 Laboratorio virtual.....	15
2.4.4 Telecomunicaciones.....	16
2.4.5 Señales en telecomunicaciones.....	16
2.4.6 Ancho de banda y capacidad de información.....	17
2.4.7 Modos de transmisión.....	19
2.4.8 Ruido.....	20
2.4.9 Decibel (dB).....	20
2.4.10 Relación señal a ruido.....	20
2.4.11 Elementos de un sistema de comunicaciones.....	21
2.4.12 Espectro electromagnético.....	22
2.4.13 Sistema de comunicación digital.....	22
2.4.14 Señales Banda Base.....	23
2.4.15 Señales Pasa Banda.....	27
2.5 Hipótesis.....	48
2.6 Señalamiento de variables.....	48
CAPITULO III	49
3. METODOLOGÍA	49
3.1 Enfoque de la investigación.....	49
3.2 Modalidad básica de investigación.....	49
3.3 Tipos de investigación.....	50
3.4 Población y muestra.....	50
3.5 Operacionalización de variables.....	51
3.6 Técnicas e instrumentos de investigación.....	53
3.7 Recolección de la información.....	53
3.8 Procesamiento de la información.....	54

CAPITULO IV	55
4. ANALISIS E INTERPRETACION DE RESULTADOS	55
4.1 Análisis	55
CAPITULO V	70
5. CONCLUSIONES Y RECOMENDACIONES	70
5.1 Conclusiones	70
5.2 Recomendaciones	71
CAPITULO VI	72
6. PROPUESTA	72
6.1 Datos Informativos.....	72
6.2 Antecedentes de la propuesta	73
6.3 Justificación.....	73
6.4 Objetivos	74
6.4.1 Objetivo General	74
6.4.2 Objetivo Especifico.....	74
6.5 Análisis de factibilidad	75
5.5.1 Factibilidad técnica	75
5.5.2 Factibilidad operativa.....	75
5.5.3 Factibilidad económica	75
6.6 Fundamentación	75
6.6.1 Introducción a labview	75
6.6.2 Panel Frontal	76
6.6.3 Diagrama de Bloques	77
6.6.4 Paleta de herramientas.....	78
6.6.5 Paleta de controles	79
6.6.6 Paleta de funciones.....	79
6.6.7 Barras de Labview	81
6.6.8 Estructuras de labview	83
6.6.9 Tipos de conexiones.....	86
6.6.10 Tipos de datos numéricos	86
6.6.11 Programación en Labview	87
6.6.12 Ejecución de un VI.....	92
6.7 Metodología.....	93

6.8	Modelo operativo	94
6.8.1	MODELAMIENTO Y SIMULACIÓN EN LABVIEW 11.0.....	94
6.8.2	CÓDIGO DE LÍNEA NRZ_U (No Retorno a Cero Unipolar)	95
6.8.3	CÓDIGO DE LÍNEA NRZ_P (No Retorno a Cero Polar)	110
6.8.4	CÓDIGO DE LÍNEA RZ_U (Retorno a Cero Unipolar)	113
6.8.5	CODIGO DE LINEA RZ_B (Retorno a Cero Bipolar)	117
6.8.6	CODIGO DE LINEA NRZ_AMI (NRZ e inversion alternada de marca).....	121
6.8.7	CODIGO DE LINEA RZ_ AMI (RZ e inversión alternada de marca).....	126
6.8.8	CODIGO DE LINEA MANCHESTER.....	132
6.8.9	MODULACION Y DEMODULACION ASK.....	137
6.8.10	MODULACION Y DEMODULACION FSK	156
6.8.11	MODULACIÓN Y DEMODULACIÓN PSK.....	160
6.8.12	MODULACIÓN Y DEMODULACIÓN QAM	164
6.8.13	DESARROLLO DEL PROGRAMA PRINCIPAL.....	168
6.9	Presupuesto	177
CAPITULO VII	178
7. CONCLUSIONES Y RECOMENDACIONES	178

Índice de Figuras

Contenido	Página
Figura 2.1: Categoría Fundamental Variable Independiente	11
Figura 2.2: Categoría Fundamental Variable Dependiente.....	11
Figura 2.3: Señal Analógica.....	16
Figura 2.4: Señal Digital	17
Figura 2.5: Ancho de banda	17
Figura 2.6: Modos de transmisión	19
Figura 2.7: Señal contaminada por ruido.....	20
Figura 2.8: Diagrama simplificado de un sistema de comunicaciones	21
Figura 2.9: Espectro Electromagnético.....	22
Figura 2.10: Ejemplo de código unipolar.....	23
Figura 2.11: Ejemplo de código bipolar.....	24
Figura 2.12: Códigos de Línea.....	25
Figura 2.13: Bifásica digital.....	26
Figura 2.14: Técnicas de modulación digital	28
Figura 2.15: Modulación por desplazamiento en Amplitud (ASK).....	28
Figura 2.16: Diagrama de un modulador ASK	30
Figura 2.17: Espectro de una señal ASK	30
Figura 2.18: Representación ancho de banda ASK.....	31
Figura 2.19: Grafica constelación ASK	32
Figura 2.20: Demodulador coherente ASK.....	33
Figura 2.21: Demodulador no coherente ASK.....	33
Figura 2.22: Modulación por desplazamiento en Frecuencia (FSK)	34
Figura 2.23: Formas de onda de entrada y salida binaria FSK	35
Figura 2.24: Desviación de frecuencia.....	36
Figura 2.25: Espectro de una señal FSK	38
Figura 2.26: Constelación FSK.....	39
Figura 2.27: Demodulador FSK no coherente	39
Figura 2.28: Demodulador FSK coherente	40
Figura 2.29: Modulación Digital de Fase (PSK).....	40
Figura 2.30: Modulador PSK.....	41
Figura 2.31: Constelación BPSK	42
Figura 2.32: Demodulador PSK.....	44
Figura 2.33: Modulación de amplitud en cuadratura (QAM)	45
Figura 2.34: Modulador QAM.....	45
Figura 2.35: Constelación QAM: (a) 8-QAM; (b) 16-QAM;	46
Figura 2.36: Demodulación QAM	47
Figura 4.1: Grafico porcentual-Pregunta 1.	56
Figura 4.2: Grafico porcentual-Pregunta 2	57
Figura 4.3: Grafico porcentual-Pregunta 3	58
Figura 4.4: Grafico porcentual-Pregunta 4	59
Figura 4.5: Grafico porcentual-Pregunta 5	60
Figura 4.6: Grafico porcentual-Pregunta 6	61

Figura 4.7: Grafico porcentual-Pregunta 7	62
Figura 4 8: Grafico porcentual-Pregunta 1	63
Figura 4.9: Gráfico porcentual-Pregunta 2	64
Figura 4 10: Grafico porcentual-Pregunta 3	65
Figura 4.11: Gráfico porcentual-Pregunta 4	66
Figura 4 12: Grafico porcentual-Pregunta 5	67
Figura 4.13: Grafico porcentual-Pregunta 6	68
Figura 6.1: Interfaces de un	76
Figura 6.2: Panel frontal de VI	77
Figura 6.3: Diagrama de bloque deVI.....	78
Figura 6.4: Paleta de herramientas.....	78
Figura 6.5: Paleta de Controles.....	79
Figura 6.6: Paleta de funciones.....	79
Figura 6.7: Submenús de la paleta de funciones.....	80
Figura 6.8: Barras de Menú	81
Figura 6.9: Barra de herramientas del panel frontal.....	81
Figura 6.10: Barra de herramientas del diagrama de bloques.....	82
Figura 6.11: Estructura Case Structure	83
Figura 6.12: Sequence Structures.....	83
Figura 6.13: For Loop.....	84
Figura 6.14: Shift Register en For Loop	84
Figura 6.15: For Loop con Add Element	85
Figura 6.16: While Loop.....	85
Figura 6.17: Características de la conexión	86
Figura 6.18: Representaciones Numéricas.....	87
Figura 6.19: Inicialización de labview	87
Figura 6.20: Selección de Blank VI.....	88
Figura 6.21: Interfaces de un VI	88
Figura 6.22: Creación de los controles.....	89
Figura 6.23: Diagrama de bloques	90
Figura 6.24: Uso de las funciones.....	90
Figura 6.25: Etiquetando los controles	91
Figura 6.26: Creación del ciclo While-loop.....	91
Figura 6.27: Botón Run.....	92
Figura 6.28: Corriendo la aplicación.....	92
Figura 6.29: Botón Continuous Run	92
Figura 6.30: Botones Abort Execution y Stop	93
Figura 6.31: Ejecutando el VI.....	93
Figura 6.32: Instrumentos del Laboratorio Virtual	94
Figura 6. 33: Control numérico.....	96
Figura 6. 34: Control de Perilla.....	96
Figura 6. 35: Indicador numérico.....	96
Figura 6.36: Gráfico de Forma de Onda	97
Figura 6.37: Array.....	97
Figura 6.38: Ciclo For Loop	98
Figura 6.39: Case Structure.....	98

Figura 6.40: Array Size	99
Figura 6.41: Index Array.....	99
Figura 6.42: Insert Into Array	99
Figura 6.43: Initialize Array	100
Figura 6.44: Reshape Array	100
Figura 6.45: Get waveform components.....	100
Figura 6.46: Bundle	101
Figura 6.47: Number to Boolean array	101
Figura 6.48: Boolean to (0,1)	101
Figura 6.49: Square waveform.....	102
Figura 6.50: Expresiones matemáticas.....	102
Figura 6.51: Esquema del ingreso de bits	104
Figura 6.52: Control de los bits a codificar.....	104
Figura 6.53: Código NRZ_U	105
Figura 6.54: Señal de entrada en bits	106
Figura 6.55: Creación onda cuadrada	106
Figura 6.56: Señales y configuración.....	107
Figura 6.57: Panel Frontal del SubVI-NRZ_U	108
Figura 6.58: Diagrama de Bloques NRZ_U.....	109
Figura 6.59: Codificación NRZ_P	111
Figura 6.60: Panel Frontal del SubVI-NRZ_P.....	111
Figura 6.61: Diagrama de bloques Código NRZ_P	112
Figura 6.62: Codificación RZ_U	114
Figura 6.63: Panel Frontal del SubVI-RZ_U	115
Figura 6.64: Diagrama de bloques RZ_U	116
Figura 6.65: Condiciones del Case Structire.....	118
Figura 6.66: Código RZ_B	118
Figura 6.67: Panel Frontal SubVI-RZ_B	119
Figura 6.68: Diagrama de bloques RZ_B	120
Figura 6.69: Panel frontal de SubVI-NRZ_AMI	122
Figura 6.70: Diagrama de bloque del SubVI-NRZ_AMI	126
Figura 6.71: Etapa de obtención de la señal codificada en tiempo NRZ_AMI	127
Figura 6.72: Etapa de obtención de la señal codificada en tiempo RZ_AMI	128
Figura 6.73: Panel frontal de SubVI-RZ_AMI	128
Figura 6.74: Diagrama de bloque del SubVI-RZ_AMI	132
Figura 6.75: Codificación de línea Manchester	134
Figura 6.76: Panel frontal de SubVI- Manchester	135
Figura 6.77: Diagrama de bloque Manchester.....	137
Figura 6.78: Botón Booleano	138
Figura 6.79: Gráfico de Forma de Onda XY	138
Figura 6.80: Ciclo While.....	139
Figura 6.81: Transmisor de secuencia bits.....	139
Figura 6.82: Generador de parámetros del sistema.....	139
Figura 6.83: Generador de parámetros de sincronización.....	140
Figura 6.84: Generador de coeficientes de filtro.....	140
Figura 6.85: Modulador ASK	140

Figura 6.86: Generador AWGN.....	141
Figura 6.87: Demodulador ASK.....	141
Figura 6.88: Convertidor pasabanda.....	141
Figura 6.89: Diagrama de ojo.....	142
Figura 6.90: Diagrama Constelación.....	142
Figura 6.91: Calcula el Bit de Error.....	142
Figura 6.92: Ubundle by name.....	143
Figura 6.93: Bundle by name.....	143
Figura 6.94: Delete from array.....	143
Figura 6.95: Server reference.....	144
Figura 6.96: Clúster Message.....	145
Figura 6.97: Conexión del clúster.....	145
Figura 6.98: Conectando el Ubundle by name con Tx Sequence.....	145
Figura 6.99: Creación de un nuevo subarray.....	146
Figura 6.100: Generación de los bits transmitidos.....	146
Figura 6.101: Conectando los VIs.....	147
Figura 6.102: Seleccionando los VIs para la modulacion-demodulacion ASK.....	147
Figura 6.103: Conectando los VIs.....	148
Figura 6.104: Controles para la modulación ASK.....	149
Figura 6.105: Creación de los indicadores.....	150
Figura 6.106: Configuración Spectral Measurements.....	151
Figura 6.107: Parámetros de sincronización.....	152
Figura 6.108: Controles para la sincronización.....	152
Figura 6.109: Panel Frontal SubVI-ASK.....	154
Figura 6.110: Diagrama de bloques del SubVI-ASK.....	155
Figura 6.111: Barra deslizante horizontal con puntero.....	156
Figura 6.112: Expresiones Add y Divide.....	156
Figura 6.113: Cambio del selector polimórfico.....	157
Figura 6.114: Creación de los controles para el MT Generate System.vi.....	157
Figura 6.115: Panel Frontal SubVI-FSK.....	159
Figura 6.116: Diagrama de bloque del SubVI-FSK.....	159
Figura 6.117: Menu Ring.....	160
Figura 6.118: Enum.....	160
Figura 6.119: Modificación del selector polimórfico.....	161
Figura 6.120: Controles para seleccionar el tipo y el diferencial PSK.....	161
Figura 6.121: Panel Frontal del SubVI-PSK.....	163
Figura 6.122: Diagrama de bloques del SubVI-PSK.....	164
Figura 6.123: Panel Frontal SubVI-QAM.....	166
Figura 6.124: SubVI-QAM.....	167
Figura 6.125: Contenedor.....	168
Figura 6.126: Menú.....	168
Figura 6.127: Panel Frontal del VI Principal Banda Base.....	169
Figura 6.128: Control del ingreso de los bits a codificar.....	170
Figura 6.129: Parámetros de entrada y salida de cada SubVI.....	171
Figura 6.130: SubVIs dentro del Case Structure.....	172
Figura 6.131: Panel Frontal del VI Principal Pasa Banda.....	173

Figura 6.132: Diagrama en bloques del simulador de señales en Pasa Banda.....	174
Figura 6.133: SubVIs dentro del Case Structure.....	176

Índice de Tablas

Contenido	Página
Tabla 4.1: Tabulación-Pregunta 1.....	56
Tabla 4.2: Tabulación-Pregunta 2.....	57
Tabla 4.3: Tabulación-Pregunta 3.....	58
Tabla 4.4: Tabulación-Pregunta 4.....	59
Tabla 4.5: Tabulación-Pregunta 5.....	60
Tabla 4.6: Tabulación-Pregunta 6.....	61
Tabla 4.7: Tabulación-Pregunta 7.....	62
Tabla 4.8: Tabulación-Pregunta 1-Encuesta 2.....	63
Tabla 4.9: Tabulación-Pregunta 2-Encuesta 2.....	64
Tabla 4.10: Tabulación-Pregunta 3-Encuesta 2.....	65
Tabla 4.11: Tabulación-Pregunta 4-Encuesta 2.....	66
Tabla 4.12: Tabulación-Pregunta 5-Encuesta 2.....	67
Tabla 4.13: Tabulación-Pregunta 6-Encuesta 2.....	68
Tabla 4. 14: Tabulación-Ficha Observación.....	69
Tabla 6.1: Costo de la licencia de labview.....	177

Índice de Ecuaciones

Contenido	Página
Ecuación 2.1: Ancho de Banda	18
Ecuación 2.2: Ley de Hartley	18
Ecuación 2.3: Capacidad de canal.....	19
Ecuación 2.4: Decibel	20
Ecuación 2.5: Relación Señal a Ruido	21
Ecuación 2.6: Ecuación de la modulación ASK	29
Ecuación 2.7: Señal moduladora para 1 lógico.....	29
Ecuación 2.8: Señal moduladora para 0 lógico.....	29
Ecuación 2.9: Espectro de frecuencia ASK	30
Ecuación 2.10: Ancho de banda ASK.....	31
Ecuación 2.11: Constelación ASK.....	31
Ecuación 2.12: Modulación FSK	34
Ecuación 2.13: Desviación máxima de frecuencia	37
Ecuación 2.14: Índice de modulación	38
Ecuación 2.15: Modulación QAM	44
Ecuación 2.16: Ancho de banda QAM	47

RESUMEN EJECUTIVO

En la actualidad, el software está siendo utilizado como herramienta didáctica que ayuda a solucionar los problemas de equipamiento en los laboratorios de algunas universidades, obteniendo resultados satisfactorios tanto para los docentes como para los estudiantes.

Las aplicaciones que ofrecen el software y la programación gráfica, permiten crear entornos gráficos sencillos para que el usuario manipule como si fuese un equipo real. De esta manera, se pretende que no existan limitaciones en las prácticas de laboratorio.

Labview (Laboratory Virtual Instrumentation Engineering Workbench), es una herramienta gráfica, para pruebas, control y diseño mediante la programación. El lenguaje que usa se llama lenguaje G, donde la G simboliza lenguaje Gráfico. Los programas desarrollados con LabVIEW, se llaman Instrumentos Virtuales y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica sino también a su programación embebida. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW, consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante -tarjetas de adquisición de datos, Visión, instrumentos y otro Hardware. Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer (programas) relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. etc.

El presente trabajo se basa en la implementación de un laboratorio virtual a través de la plataforma labview, que solucione la carencia de prácticas de modulación y demodulación digital, en los laboratorios de electrónica en FISEI. Con el uso de instrumentos virtuales de Labview, se puede lograr: mediciones, procesar, visualizar y variar los parámetros de las señales tanto en banda base como en pasa banda.

INTRODUCCIÓN

Los primeros sistemas de comunicaciones fueron de naturaleza digital, tal es el caso del telégrafo, pero con la invención del teléfono las comunicaciones se analizan básicamente en forma analógica. Con el paso del tiempo, el uso masivo de computadoras hizo que los esfuerzos se centraran de nuevo en la comunicación digital, que ofrece una mayor disponibilidad de información de todo tipo.

Las comunicaciones digitales son términos ambiguos que podrían ser interpretados de una manera distinta por varias personas. En el contexto de este proyecto, se analizarán las modulaciones-demodulaciones digitales y a los códigos de línea, siendo estos dos temas de interés en las comunicaciones digitales.

El Laboratorio virtual a través de la plataforma labview para la realización de prácticas de modulación y demodulación digital en la FISEI de la Universidad Técnica de Ambato, es el resultado de esta investigación, a fin de tener nuevas herramientas de apoyo a los laboratorios de electrónica.

El presente proyecto consta de seis capítulos que se describen a continuación:

Capítulo I: Se realiza la investigación partiendo del problema: “Carencia de practicas de modulación y demodulación digital en la FISEI”, el mismo que se estudia y se contextualiza mediante un análisis critico. Además toda investigación se dirige a un objetivo el cual consta en un objetivo general y objetivos específicos, los mismos que sirven para el desarrollo del proyecto.

Capítulo II: La investigación se apoya en el marco teórico en donde se aborda las categorías fundamentales, que por su importancia intervienen en la investigación; las más relevantes son:

- Entornos virtual de programación
- Laboratorio virtual.
- Comunicaciones digitales.
- Modulación y demodulación digital.

Capítulo III: Contiene la metodología, el tipo de investigación y recolección de información, que es usada en el análisis de datos, verificación de hipótesis, con un análisis de las características de los laboratorios virtuales y las modulaciones digitales, además se detalla minuciosamente a la población que esta conformado por el coordinador de carrera, docentes y estudiantes de 7mo, 8vo y 9no semestre de la carrera de electrónica y comunicaciones respectivamente; y, el procesamiento de la información para un análisis posterior.

Capítulo IV: Se describe el análisis e interpretación de resultados obtenidos en la investigación realizada en la FISEI.

Capítulo V: Contiene las conclusiones y recomendaciones obtenidas luego de un análisis de la carencia de prácticas de modulación y demodulación digital, instrumentos que permitirán a las autoridades de la FISEI, observar las debilidades que tienen y mitigarlas a través de la implementación de laboratorio virtual para mejorarlas.

Capítulo VI: Se presenta la propuesta pertinente al tema de investigación, enfocada a la implementación del laboratorio virtual a través de la plataforma labview para la realización de prácticas de modulación y demodulación digital en la FISEI de la Universidad Técnica de Ambato.

CAPÍTULO I

1. EL PROBLEMA DE INVESTIGACIÓN

1.1 Tema

Laboratorio virtual a través de la plataforma labview para la realización de prácticas de modulación y demodulación digital en la FISEI de la Universidad Técnica de Ambato.

1.2 Planteamiento del problema

1.2.1 Contextualización

El avance constante de la tecnología asociado con la educación en todos los niveles, ha permitido desarrollar de una manera satisfactoria las oportunidades para transformar y mejorar los métodos de enseñanza e investigación a través de prácticas virtuales.

Por este motivo los países con grandes expectativas en educación, están realizando estudios para ampliar las posibilidades de cambiar la estructura rígida de las prácticas tradicionales, por una estructura flexible y que posibilite la emulación de distintos subsistemas. En todo el mundo, es indispensable la realización de prácticas como una manera de que el estudiante pueda aplicar sus conocimientos en posibles proyectos innovadores y contrastar la teoría expuesta en clase.

En el Ecuador como en el resto del mundo, existen varias maneras de aprovechar los recursos de software y hardware, para minimizar las carencias en equipos de laboratorio, de tal forma que el estudiante no tenga problemas en el desarrollo de sus prácticas o posibles proyectos.

En la ciudad de Ambato, se ha podido observar el desarrollo de las universidades, con mejores ofertas académicas, docentes calificados y buenas infraestructuras, siendo una de ellas la Universidad Técnica de Ambato, pero con algunas limitaciones en el equipamiento de sus laboratorios en la carrera de electrónica y comunicaciones.

La Facultad de Ingeniería en Sistemas, Electrónica e Industrial, por el momento no cuenta con equipos para ciertas prácticas, debido a la inversión económica que se requiere para su adquisición, ocasionando que la enseñanza teórica en la pizarra y la investigación en el internet, sean el medio de aprendizaje en lo que concierne a los temas de modulaciones digitales.

Las principales ventajas de realizar prácticas de modulaciones digitales es brindar a los estudiantes un entorno sencillo y práctico de experimentación, que permita entender la teoría de una mejor manera. Por otro lado, las desventajas pueden ser: dificultad en la asimilación de ideas, poca relación de la teoría con la práctica, etc.

1.2.2 Árbol de Problema

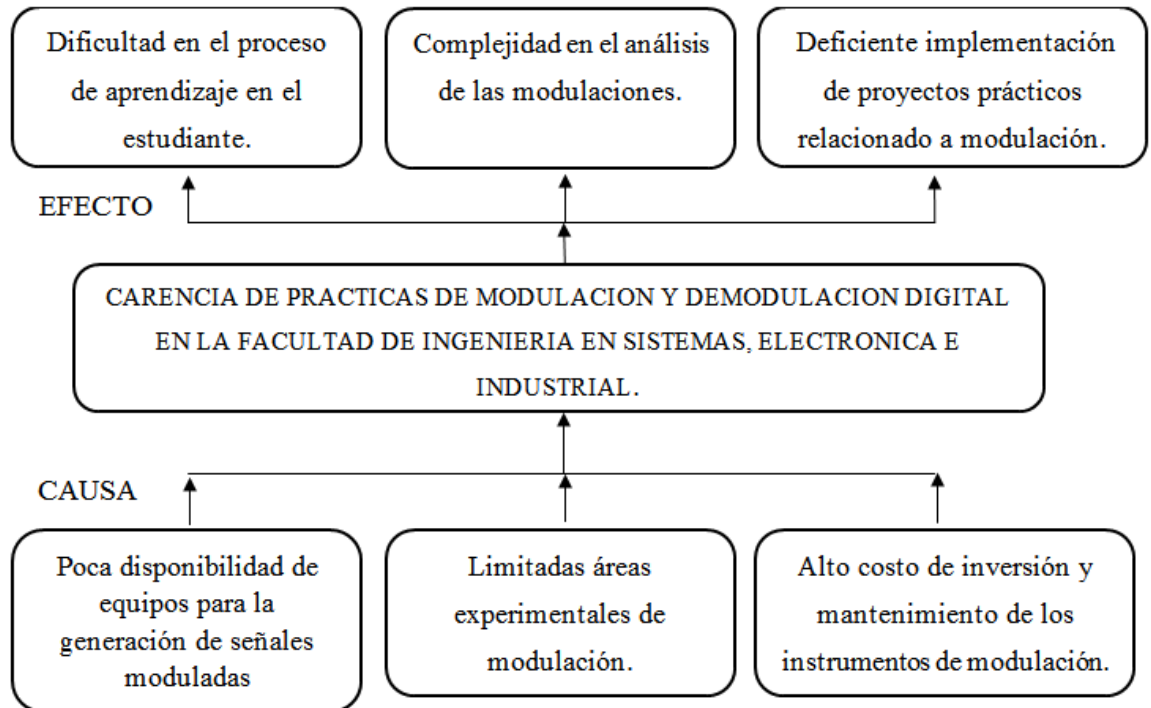


Figura 1.1: Árbol del problema
Elaborado por: El investigador

1.2.3 Análisis crítico

En los últimos años, la mayoría de los dispositivos de transmisión de datos son digitales que cuentan con tecnología moderna, por lo tanto, es de mucho interés tener conocimientos de las comunicaciones digitales. La Facultad de Ingeniería en Sistemas, Electrónica e Industrial no cuenta con un laboratorio virtual de modulación digital para las prácticas respectivas. La poca disponibilidad de equipos para la generación de señales moduladas, en los laboratorios de electrónica, origina dificultad en el proceso de aprendizaje en los estudiantes de la carrera de electrónica y comunicaciones.

Cuando se trata de entender un determinado tema de comunicaciones digitales, es importante: analizar, verificar y comprobar de una manera práctica, la teoría impartida por el docente en el aula de clases. Sin embargo, las limitadas áreas experimentales de modulaciones digitales en la FISEI, da como resultado la complejidad en el análisis y desarrollo de prácticas en ciertos temas referentes a comunicaciones digitales.

A pesar de los esfuerzos que se realizan en la FISEI, por contar con laboratorios muy bien equipados, sigue siendo un problema su alto costo de inversión y mantenimiento de los instrumentos del mismo, ocasionando la deficiente implementación de proyectos prácticos relacionado a modulaciones digitales.

1.2.4 Prognosis

La Facultad de Ingeniería Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, al tener limitaciones en la ejecución de prácticas acerca de la modulación-demodulación digital por falta de equipamiento en los laboratorios de electrónica, surgirá un desinterés por parte del estudiante en aprender y realizar proyectos de comunicaciones digitales. Se pretende entonces, aprovechar el software y hardware que se tiene en la FISEI, para crear un laboratorio virtual de modulaciones digitales.

1.2.5 Formulación del problema

¿La carencia de un laboratorio virtual a través de la plataforma labview no permite la realización de prácticas de modulación y demodulación digital en el laboratorio de electrónica de la FISEI?

1.2.6 Preguntas directrices

- ¿Qué herramientas de Software, se utilizan para la creación de laboratorios virtuales en ciertas áreas de la ingeniería?
- ¿Cómo el laboratorio virtual influirá en el proceso de aprendizaje de modulación y demodulación digital en la Facultad de Ingeniería Sistemas Electrónica e Industrial?
- ¿De qué manera el laboratorio virtual, aportará a la implementación de proyectos relacionados a modulaciones digitales?
- ¿Se puede plantear una propuesta innovadora a través del uso de un laboratorio virtual, para reducir la carencia de prácticas de modulación y demodulación digital, en el laboratorio de electrónica en la FISEI?

1.2.7 Delimitación del problema

CAMPO: Ingeniería Electrónica y Comunicaciones

ÁREA: Comunicaciones Digitales

ASPECTO: Laboratorio virtual

DELIMITACIÓN ESPACIAL: Esta investigación se la realizó en la Universidad Técnica de Ambato, en la que funciona la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

DELIMITACIÓN TEMPORAL: El presente proyecto de investigación se la efectuó en 6 meses, a partir de que este fue aprobado por el Honorable Consejo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.3 Justificación

En la actualidad, es un reto académico contar con laboratorios, que cumplan con las exigencias tecnológicas, para que los estudiantes estén relacionados al mundo actual en cuanto a los avances tecnológicos en el campo de la electrónica y puedan realizar futuras investigaciones. El uso de un laboratorio virtual, permite la simulación de procesos que eran imposibles de realizar en forma real por diferentes razones que son: la complejidad de análisis, costos y el tiempo que se tardaría en ejecutarse dicho proceso. La Facultad de Ingeniería en Sistemas, Electrónica e Industrial, por poseer carreras técnicas y al estar a margen del desarrollo tecnológico, es indispensable que haga uso de este tipo de laboratorio virtual que promueva la generación de proyectos y propuestas al país.

Los beneficiarios en la implementación de este proyecto, serán los docentes, estudiantes del área de electrónica y sus autoridades, que tendrán dentro de las instalaciones de los laboratorios, herramientas virtuales para complementar la enseñanza y el aprendizaje teórico. De esta manera, se estará dando soluciones a las carencias de prácticas e incentivando a la investigación en lo que concierne a modulaciones digitales.

El proyecto es factible de ejecutarse en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, porque se tiene los recursos necesarios para su implementación y control del mismo. Además, el costo de inversión es menor a los beneficios que aportará dicho proyecto, a la calidad de las prácticas en los laboratorios de electrónica.

1.4 Objetivos

1.4.1 Objetivo General:

- Analizar el uso de un laboratorio virtual y su influencia en la realización de prácticas de modulación y demodulación digital en el laboratorio de electrónica de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.4.2 Objetivos Específicos:

- Analizar los diversos programas de Software para la creación de un laboratorio virtual en la FISEI.
- Realizar un estudio sobre el nivel de uso de un laboratorio virtual que simule, prácticas de modulación y demodulación digital en los laboratorios de electrónica de la FISEI.
- Implementar un laboratorio virtual para la realización de prácticas modulación y demodulación digital para el laboratorio de electrónica en la FISEI.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Antecedentes investigativos

Dentro de los registros bibliográficos que reposan en la Biblioteca de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, no se logró encontrar trabajos investigativos que podrían guardar relación al tema propuesto. Continuando con la investigación previa se ha encontrado un trabajo relacionado a las modulaciones digitales pero con otro enfoque al proyecto propuesto, siendo la siguiente:

VALENCIA, Leopoldo (2010). “Integración del Downconverter NI 5660 a través de instrumentos virtuales bajo labview para generar diseños de referencia y procedimientos aplicables a sistemas de telecomunicaciones”. Ejecutado en la Escuela Superior Politécnica del Ejercito. Carrera de Ingeniería en Electrónica y Telecomunicaciones. Conclusión: El proyecto, trata de la realización de instrumentos virtuales de los distintos esquemas de modulación tanto analógicos como digitales, utilizando como interfaz, el software labview junto con algunos equipos de hardware como son: el equipo NI-5660 (analizador de RF) y generadores de señales de RF.

2.2 Fundamentación legal

La Universidad Técnica de Ambato, es una Institución de Educación Superior que ofrece carreras con título terminal de tercer nivel.

2.3 Gráfica de inclusión de las categorías fundamentales

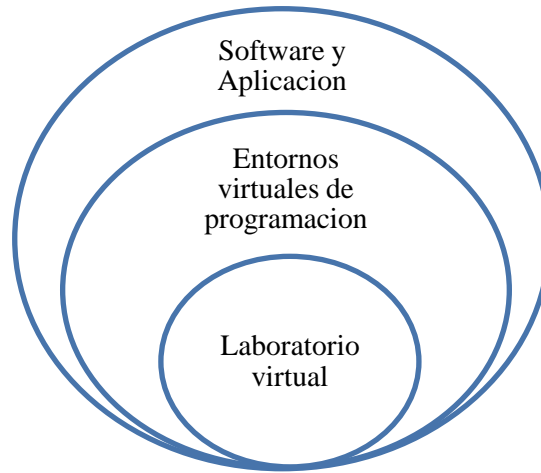


Figura 2.1: Categoría Fundamental Variable Independiente

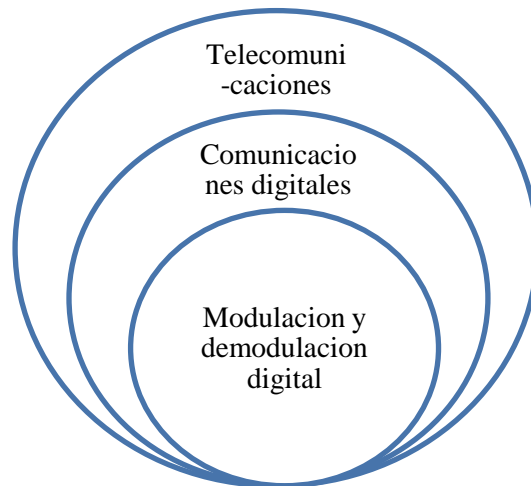


Figura 2.2: Categoría Fundamental Variable Dependiente

2.4 Categorías fundamentales

2.4.1 Software y Aplicación.

2.4.1.1 Software

El software, es la parte intangible del computador y junto con algunos componentes lógicos, se puede interactuar con el hardware, para la realización de una determinada aplicación.

En la industria se utilizan mucho los lenguajes de programación basados en texto: C, C++, C#, Visual Basic e incluso la plataforma de LabWindows/CVI que es una plataforma de programación basada en texto de National Instruments.

A continuación se cita algunas herramientas de software con sus aplicaciones de una manera resumida y que pueden ser:

Matlab

El Matlab, es un lenguaje de programación de muy alto nivel con un elevado grado de interactividad y flexibilidad en la implementación de algoritmos.

Con Matlab es posible combinar muchas disciplinas tales como: estadística, análisis matemático, procesamiento digital de señales, diseño de filtros analógicos, etc. Las siguientes características que presenta este lenguaje son:

- No requiere etapa de compilación y las etapas de escritura de código y ejecución pueden ser simultáneas
- Integra los requisitos clave de un sistema de cálculo técnico: cálculo numérico, gráficos, herramientas para aplicaciones específicas y capacidad de ejecución en múltiples plataformas.

La programación en Matlab, presenta las siguientes ventajas:

- Amplio soporte matemático
- Alta precisión
- Amplio soporte de funciones ya desarrolladas
- Integración con dispositivos hardware
- Una comunidad muy extendida

De Matlab también, se puede mencionar las siguientes desventajas:

- Gestión “oscura” de la memoria
- Problemas eventuales de velocidad
- Comercial
- Distribución de ejecutables

Lenguaje C

El Lenguaje C, es un lenguaje de programación estructurado orientado a la implementación de sistemas operativos concretamente Unix. Al decir que es un lenguaje estructurado se refiere a que el código o algoritmo esta ordenado.

El lenguaje C, sigue siendo uno de los lenguajes más populares, utilizados en empresas, organizaciones y fábricas de software de todo el mundo. Sus características se presentan a continuación:

- Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de archivos, proporcionadas por bibliotecas.
- Es un lenguaje muy flexible que permite programar con múltiples estilos
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Un conjunto reducido de palabras clave.
- Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.

Las ventajas que ofrece el lenguaje C, son las siguientes:

- Eficacia.
- Libertad para organizar el trabajo.
- Lenguaje de alto nivel, se asemeja al lenguaje humano.
- Es un lenguaje diseñado para la resolución de problemas, independientes de las características del computador.
- En ocasiones determinadas permite programar como lenguaje de bajo nivel, consiguiendo una mayor eficacia.
- Flexibilidad.
- Muy potente.

Como desventaja del lenguaje C, se tiene:

- No es un lenguaje muy estructurado como ocurre en lenguajes como PASCAL.
- Es un lenguaje complicado para el usuario novel

Lenguaje C#

El lenguaje C#, es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y muchas tareas más.

El lenguaje C#, posee las siguientes ventajas:

- Existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.

- Cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno-protegido y privado.
- Se puede declarar a los métodos para que acepten un número variable de parámetros.
- Métodos virtuales y redefiniciones
- C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres.

En cambio, las desventajas del lenguaje C#, son las siguientes:

- Se tiene que conseguir una versión reciente de Visual Studio .NET.
- Tener algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como contar con Windows NT 4 o superior.
- Poseer, 4 gigas de espacio libre para la instalación, etc.
- Dificultad para quien no está familiarizado con ningún lenguaje de programación.

LabView

Labview, es una herramienta gráfica para pruebas, control y diseño mediante la programación. El lenguaje que usa se llama lenguaje G, por ser un lenguaje Gráfico.

Las principales aplicaciones del software Labview son:

- Adquisición de datos y análisis matemático.
- Comunicación y control de instrumentos de cualquier fabricante.
- Diseño de controladores: simulación, prototipaje rápido y validación.
- Robótica.
- Domótica y redes de sensores inalámbricos.

Las ventajas presentadas de Labview, son las siguientes:

- La programación es tan sencilla o compleja como el programador lo desee.
- Permite trabajar con distintas plataformas de hardware utilizando la misma plataforma de software (LabVIEW), por ejemplo, tarjetas de adquisición de datos (entradas y salidas análogas, contadores, entradas y salidas digitales), multímetros digitales, osciloscopios, comunicación serial, tarjetas de control de motores, cámaras, microcontroladores, PLC's, etc.
- Se tienen múltiples toolkits y módulos para aplicaciones que requieran de procesamiento de señales, control, visión, control de motores, generación de reportes, comunicación con bases de datos, etc.
- Generación automática de código.

- Interface amigable para probar el hardware (Measurement & Automation Explorer).

En conclusión: Una vez analizado las aplicaciones más relevantes, así como también las ventajas y desventajas de cada herramienta de software que se ha mencionado anteriormente, se puede concluir que Labview es una de las mejores alternativas en programación, por las siguientes razones:

- No se tiene que digitar grandes cantidades de líneas de código para configurar la adquisición de datos y generar la interface de usuario.
- Tiene gran variedad de controles y funciones para el desarrollo de diferentes aplicaciones.
- La programación es de manera gráfica.

2.4.2 Entornos virtuales de programación.

El entorno virtual, es la interfaz gráfica que un programa tiene para realizar la programación de una manera didáctica y comprensible a diferencia de otros programas, que utilizan grandes extensiones de código para una determinada operación. A continuación citamos algunos entornos virtuales, haciendo énfasis en el entorno de LabView que pueden ser:

- WinFACT (Entorno de laboratorio virtual de simulación).
- DasyLab (Entorno de simulación con capacidades graficas).
- LabView (Entorno de programación grafica).

2.4.3 Laboratorio virtual.

Un laboratorio virtual, es un entorno de simulación que ayuda a disminuir, los problemas por falta de equipos en los laboratorios reales así como la complejidad de los temas a impartir. El laboratorio virtual brinda un aprendizaje didáctico y práctico con la ayuda de algunas herramientas orientadas al diseño y simulación de prototipos de laboratorio.

Los parámetros que podemos modificar por ejemplo en un laboratorio virtual de comunicaciones son:

- Generación de la señal: potencia, distribución, calidad
- Parámetros virtuales: Ratio de codificación, intervalo de guarda, constelación, muestreo, FEC, IFFT
- Simulación del canal RF: ruido impulsivo, ruido gaussiano o ruido térmico, ecos (Recepción multitrayecto), Doppler Shift
- Control de Recepción: PLL, Señales Piloto, Eb/No, BER, MER, ACPR.

2.4.4 Telecomunicaciones.

La **telecomunicación**, es una técnica que consiste en transmitir un mensaje desde un punto a otro a través de un medio de transmisión.

Santa Cruz (2008) afirma que: “*Telecomunicación es toda transmisión, emisión o recepción de signos, señales, escritos, imágenes, sonidos o informaciones de cualquier naturaleza por hilo, radioelectricidad, medios ópticos u otros sistemas electromagnéticos*”.

2.4.5 Señales en telecomunicaciones.

Las señales que contienen información pueden ser analógicas o digitales:

- **Señal analógica.**- La señal analógica, es la señal que puede asumir un número infinito de valores en cualquier instante de tiempo. La figura 2.3 muestra una señal analógica que al variar sus parámetros permite la representación de información.

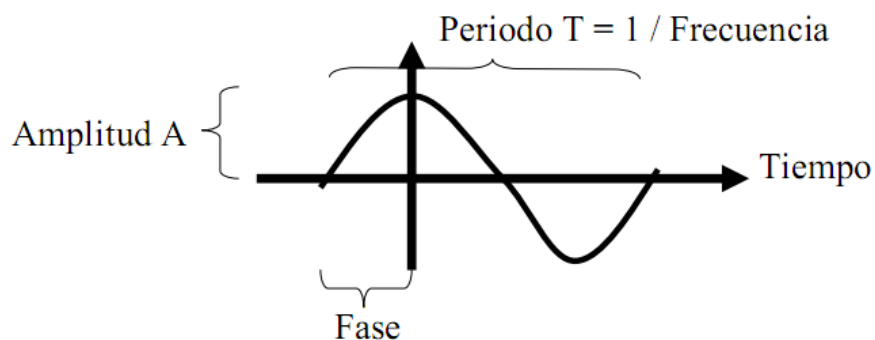


Figura 2.3: Señal Analógica
Fuente: El investigador

- **Señal digital.**- La señal digital, es una señal que puede asumir un número finito de valores, en cualquier instante de tiempo. La figura 2.4a muestra una señal digital que tiene una forma cuadrada con sus transiciones de un nivel alto a bajo y viceversa. Sin embargo, en la figura 2.4b se puede observar una señal digital de varios niveles (señal m-aria).

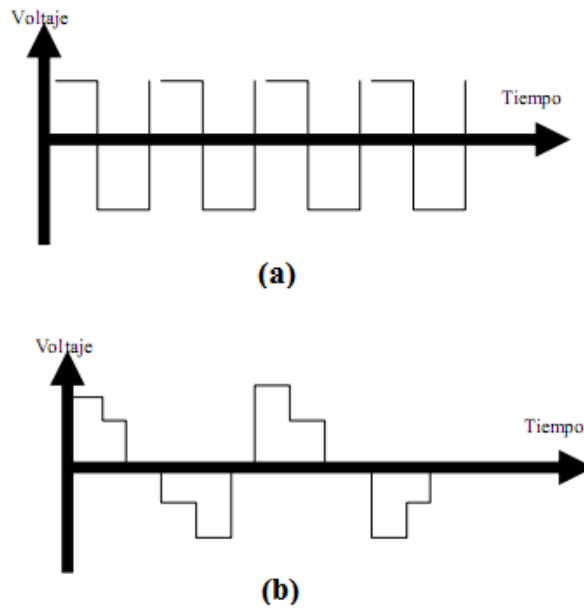


Figura 2.4: Señal Digital: (a) un estado; (b) m-aria
Fuente: El investigador

2.4.6 Ancho de banda y capacidad de información.

El **ancho de banda** de una señal de información, es la diferencia entre la frecuencia máxima y mínima, contenidas en la información. En la figura 2.5 se muestra la amplitud, frecuencia portadora (f_0) y el ancho de banda. También se indican, la frecuencia máxima (f_2) y la frecuencia mínima (f_1).

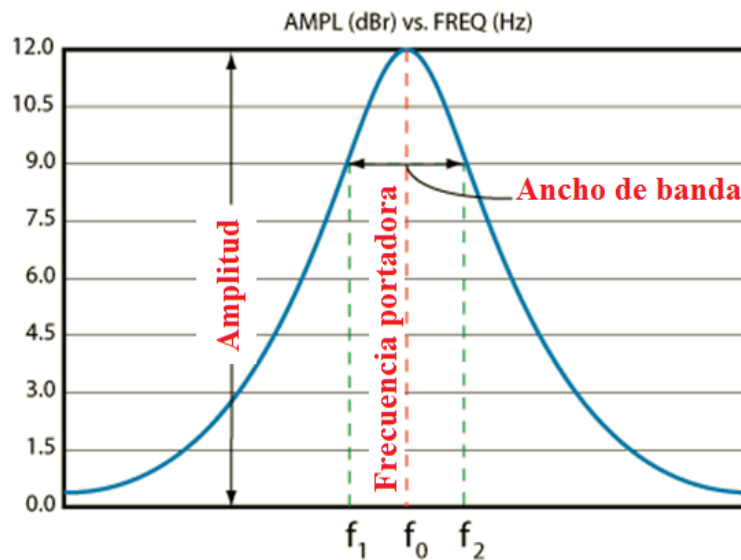


Figura 2.5: Ancho de banda
Fuente: El investigador

El ancho de banda de la figura 2.5, se expresa como en la ecuación 2.1.

$$B = f_2 - f_1$$

Ecuación 2.1: Ancho de Banda

Donde:

B: ancho de banda (hertz)

f₂: frecuencia máxima (hertz)

f₁: frecuencia mínima (hertz)

Por otro lado, la **capacidad de información** es una medida de cuánta información se puede transferir a través de un sistema de comunicaciones en determinado tiempo. La relación entre el ancho de banda, tiempo de transmisión y capacidad de información fue desarrollada en 1920 por R. Hartley de los Laboratorios Telefónicos Bell. De manera sencilla se muestra en la ecuación 2.2, la ley de Hartley:

$$I \in B \times t$$

Ecuación 2.2: Ley de Hartley

Donde:

I = capacidad de información (bits por segundo)

B = ancho de banda (hertz)

t = tiempo de transmisión (segundos)

En la anterior ecuación, muestra que la capacidad de información es una función lineal y directamente proporcional al ancho de banda del sistema y al tiempo de transmisión. Por lo tanto, si se modifica el ancho de banda o el tiempo, ocurrirá un cambio significativo a la capacidad de información.

Claude Shannon publicó un documento, en el año 1949, en donde relacionó, la capacidad de información de un canal de comunicaciones con el ancho de banda y la relación de señal a ruido. Esta relación se aprecia, en la ecuación 2.3.

$$I = B \log_2(1 + S/N) \quad (bps)$$

Ecuación 2.3: Capacidad de canal

Donde:

I: Capacidad de información (bits por segundo)

B: Ancho de Banda (hertz)

S/N: Relación de pote señal a ruido (sin unidades)

2.4.7 Modos de transmisión

Los sistemas electrónicos de comunicaciones se pueden diseñar para manejar la transmisión solo en una dirección, en ambas direcciones, solo en una a la vez, o en ambas direcciones al mismo tiempo. En la figura 2.6 se detalla los cuatro modos de transmisión posibles:

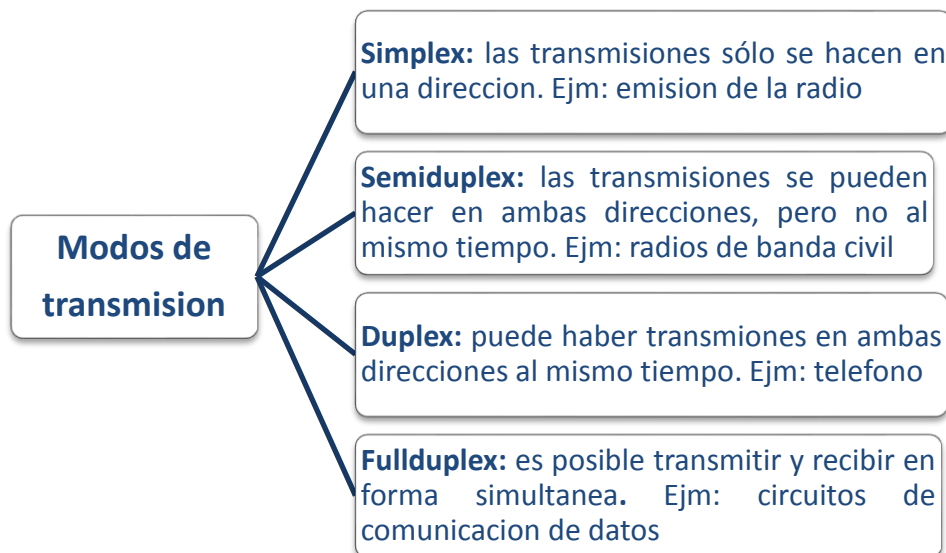


Figura 2.6: Modos de transmisión

Fuente: Por el investigador

2.4.8 Ruido

El **ruido**, es toda señal no deseada que se mezcla con la señal que se desea transmitir. En la figura 2.7 se ve a una señal senoide pura y otra con presencia de ruido.

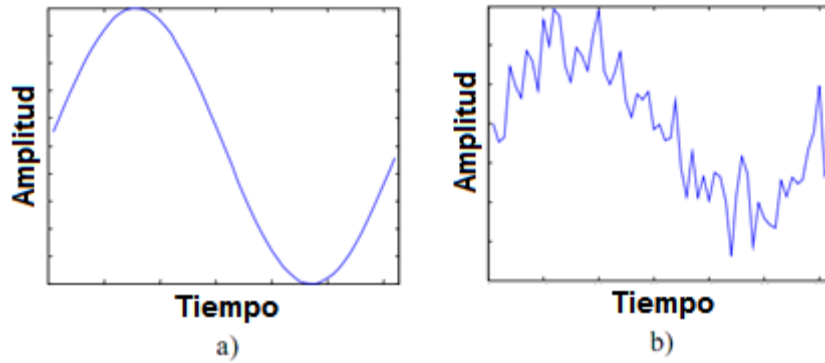


Figura 2.7: Señal contaminada por ruido: (a) senoidal pura, (b) senoidal con ruido
Fuente: Por el investigador

2.4.9 Decibel (dB)

Decibel, es una medida logarítmica de base 10, de las relaciones de potencia, es decir, es un valor relativo de la potencia de salida con respecto a la potencia de entrada, ver la ecuación 2.4:

$$dB = 10\log\left(\frac{P_{salida}}{P_{entrada}}\right)$$

Ecuación 2.4: Decibel

donde:

dB = decibel (sin unidades)

P_{salida} = potencia de salida (vatio)

$P_{entrada}$ = potencia de entrada (vatio)

2.4.10 Relación señal a ruido.

La relación señal a ruido, es una relación matemática que indica el nivel de señal respecto al ruido en un punto dado del algún dispositivo. La relación señal a ruido puede venir dada en forma de voltaje o potencia y normalmente es una función

logarítmica y su unidad es el decibel. La ecuación 2.5, muestra dicha relación en términos de potencia:

$$\frac{S}{N} = \left(\frac{P_S}{P_N}\right)^2$$

Ecuación 2.5: Relación Señal a Ruido

Donde:

S/N = Relación señal a Ruido (decibelio)

P_S/P_N = Potencia de la señal / Potencia del ruido

2.4.11 Elementos de un sistema de comunicaciones.

La figura 2.8 muestra un diagrama de bloques simplificado de un sistema electrónico de comunicaciones que comprende un transmisor, un medio de transmisión y un receptor.

- **Transmisor.-** Es un conjunto de uno o mas dispositivos electrónicos que convierte la información de la fuente original en una señal, para su posterior envío a través del medio de transmisión llamado canal.
- **Medio de transmisión.-** Es el medio por el cual se transporta la señal desde el transmisor hasta el receptor.
- **Receptor:** Es un conjunto de dispositivos electrónicos que acepta del medio de transmisión las señales transmitidas y las reconvierte a su forma original.

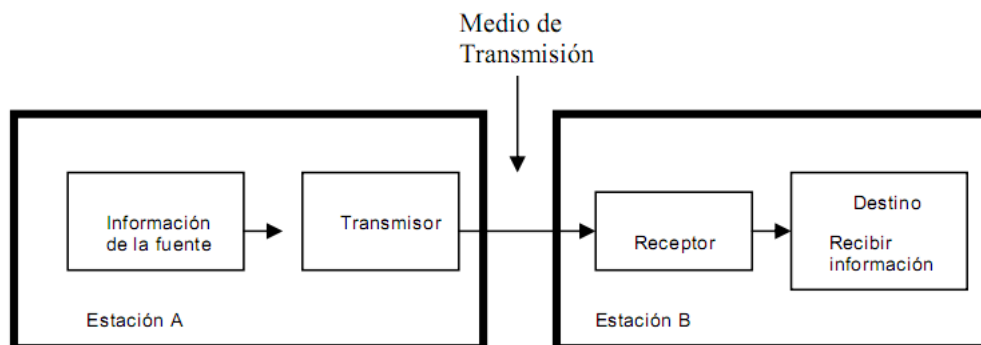


Figura 2.8: Diagrama simplificado de un sistema de comunicaciones

Fuente: Oscar Santa Cruz. Apuntes de Electrónica III

2.4.12 Espectro electromagnético.

El espectro electromagnético, nos permite mostrar localizaciones de varios servicios dentro de las bandas en forma aproximada. Los elementos de un espectro electromagnético y sus respectivos rangos de frecuencias o bandas, se indica en la figura 2.9.

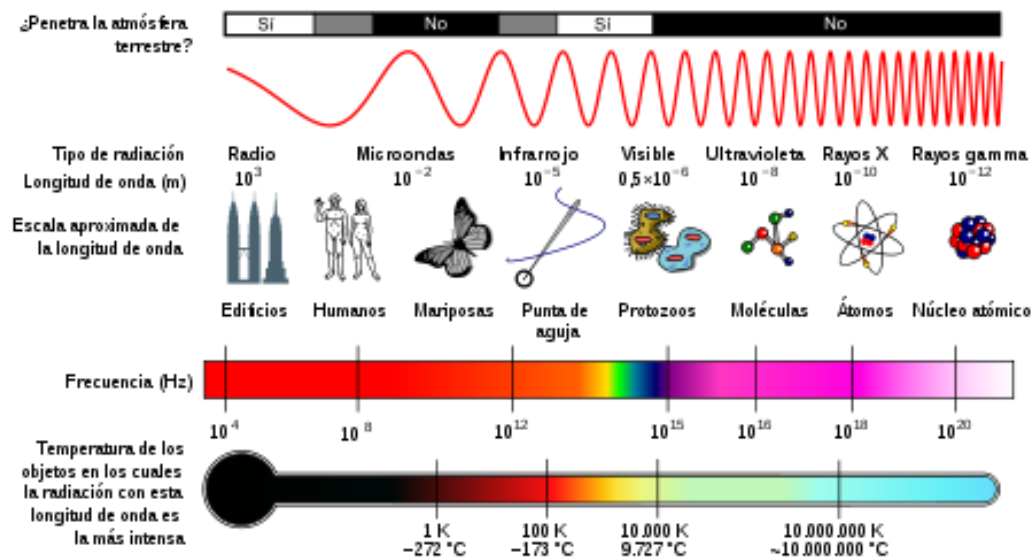


Figura 2.9: Espectro Electromagnético.

Fuente: "http://es.wikipedia.org/wiki/Espectro_electromagn%C3%A9tico"

2.4.13 Sistema de comunicación digital.

Sistema comunicación digital, es un sistema en la cual la energía electromagnética se transmite y recibe en forma digital.

La comunicación digital ofrece varias ventajas:

- Baratos: operaciones complejas se pueden implementar en forma económica en un circuito integrado.
- Permite mezclar y transmitir datos de voz, video y fuentes de datos con un sistema de transmisión digital común.
- Los receptores digitales se pueden hacer tolerantes al ruido.
- Se puede corregir los errores con el proceso de codificación.

La comunicación digital también tiene las desventajas:

- Necesidad de mayor ancho de banda de transmisión.
- Es necesario sincronización entre el transmisor y el receptor.

2.4.14 Señales Banda Base

La transmisión de datos en forma digital implica una cierta codificación. A la forma de transmisión donde no se usa una portadora se le conoce como transmisión en banda base.

2.4.14.1 Codificación de Línea

En libro titulado: “Sistema de Comunicaciones Electrónicas”, de Tomasi Wayne (2003), afirma: “*La codificación de línea consiste en convertir niveles lógicos normalizados (TTL, CMOS y semejantes) a una forma mas adecuada para su transmisión por línea telefónica*”.

Al momento de seleccionar un formato de codificación de línea, se deben tener en cuenta seis factores principales:

1. Voltajes de transmisión
2. Ciclo de trabajo.
3. Ancho de banda.
4. Recuperación de reloj.
5. Detección de errores.
6. Facilidad de detección y decodificación.

Voltajes de transmisión

Los voltajes o niveles de transmisión pueden ser unipolares (UP) o bipolares (BP). La **transmisión unipolar** de datos binarios, implica transmitir sólo en una polaridad ya sea: unipolar positivo (+V para 1 lógico y 0V para un 0 lógico) o unipolar negativo (-V para 1 lógico y 0V para un 0 lógico). La figura 2.10 muestra como ejemplo, una transmisión unipolar positiva.

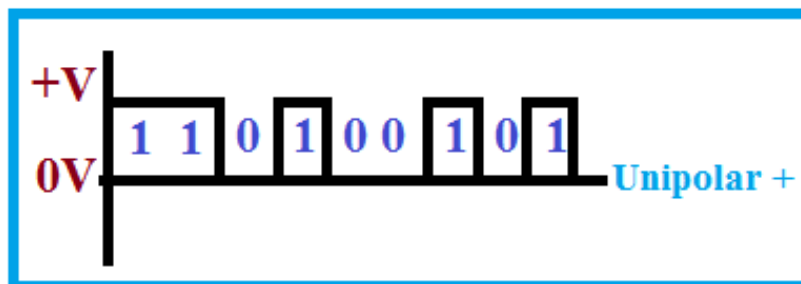


Figura 2.10: Ejemplo de código unipolar
Fuente: El investigador

En cambio, la **transmisión bipolar** posee dos polaridades distintos de cero (por ejemplo, $+V$ o $-V$ para 1 lógico y $0V$ para cero lógico). La figura 2.11 muestra este tipo de transmisión bipolar.

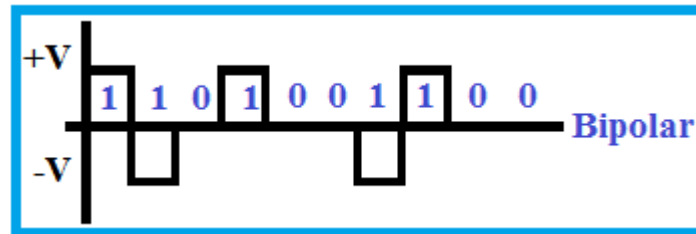


Figura 2.11: Ejemplo de código bipolar
Fuente: El investigador

En una línea de transmisión digital, es más eficiente, en lo que se refiere a potencia, codificar datos binarios con voltajes cuya magnitud sea igual pero de polaridad opuesta y balanceada simétricamente con respecto a $0V$. De esta manera, la potencia promedio se reduce en un 50%.

Ciclo de trabajo

El ciclo de trabajo de un pulso binario, también es un factor importante para clasificar el tipo de transmisión. A continuación, se explica de la manera más sencilla a los formatos de codificación de línea:

NRZ (nonreturn to zero).- Esta codificación se denomina sin regreso a cero o no retorno a cero y es cuando el pulso binario, se mantiene durante todo el tiempo de bit.

RZ (return to zero).- Se llama regreso o retorno a cero, si el tiempo activo del pulso binario ocupa menos que el 100% del tiempo de bit.

Ahora, los voltajes de transmisión unipolar y bipolar, y la codificación con regreso a cero y sin regreso a cero se pueden combinar de varias maneras, para así obtener cinco posibilidades de codificación de línea que son:

UPNRZ (unipolar no retorno a cero).- En la figura 2.12a, se muestra este código que se caracteriza por mantener el voltaje positivo durante todo el tiempo de bit (ciclo de trabajo de 100%), cuando se tiene 1 lógico y en 0 lógico, simplemente implica un 0 binario.

BNRZ (bipolar no retorno a cero).- Es cuando se tiene dos niveles de voltaje distintos de cero ($+V= 1$ lógico, y $-V= 0$ lógico), usando un ciclo de trabajo de 100% como se indica en figura 2.12b.

UPRZ (unipolar retorno a cero).- En este código, sólo se usa un voltaje distinto de cero, pero cada pulso sólo está activo durante el 50% del tiempo de bit. La figura 2.12c muestra el comportamiento del código UPRZ.

BPRZ (bipolar retorno a cero).- En la figura 2.12d se indica que hay dos voltajes distintos de cero (+V= 1 lógico y -V= 0 lógico), cada pulso sólo está activo el 50% del tiempo total de bit.

Cuando se tiene dos niveles de voltaje distintos de cero (-V y +V), ambas polaridades representan 1 lógico, y 0 V representa un 0 lógico. A este método de codificación, se lo denomina inversión de marca alternada (**AMI**). En las transmisiones AMI, cada 1 lógico sucesivo se invierte en polaridad respecto al 1 lógico anterior. Como se usa el retorno a cero, a esta técnica de codificación se le llama bipolar con retorno a cero e inversión alternada de marca (**BPRZ-AMI**), para una mejor comprensión vea la figura 2.12e.

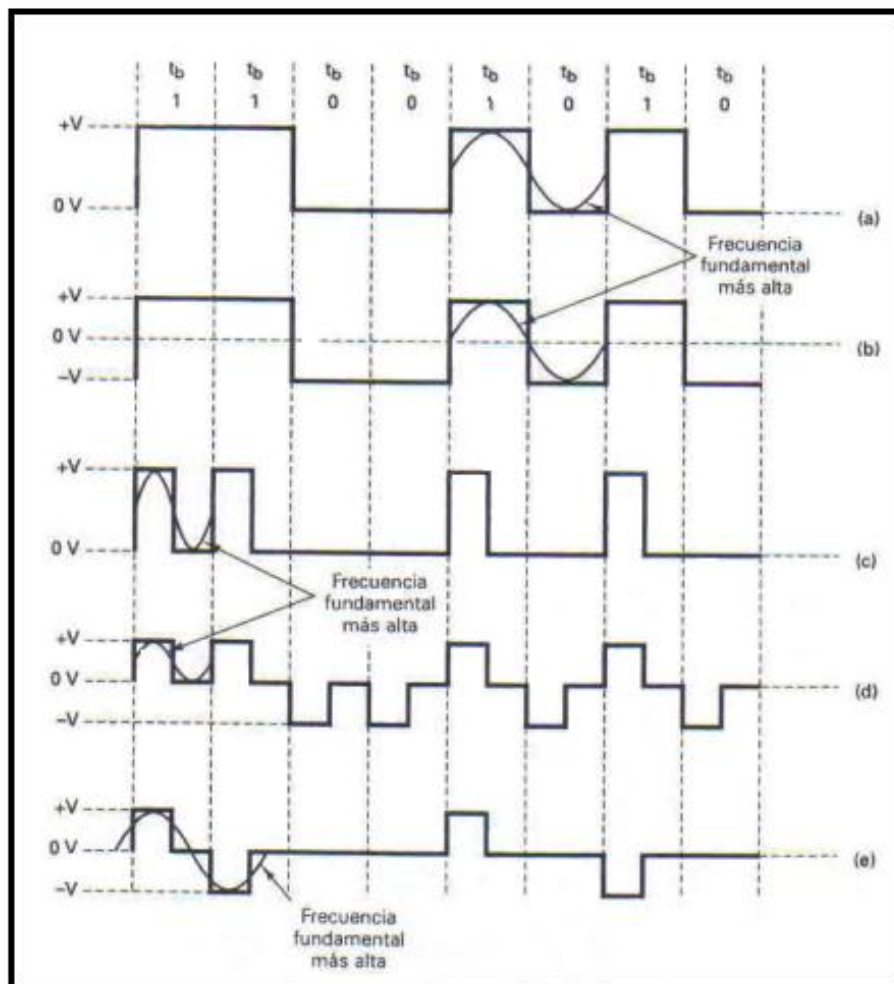


Figura 2.12: Códigos de Línea
Fuente: Wayne, Tomasi (2003). *Sistemas de Comunicaciones Electrónicas*

MANCHESTER.- La figura 2.13 muestra el código Manchester, conocido también como bifásica digital, en donde cada “1” lógico se representa con un pulso de período de medio bit positivo seguido por un pulso de período de medio bit negativo (es decir, un ciclo de onda cuadrada con fase de 0°). Del mismo modo, el “0” lógico se representa con un pulso de período de medio bit negativo seguido por un pulso de período de medio bit positivo (es decir, un ciclo de onda cuadrada con fase de 180°).

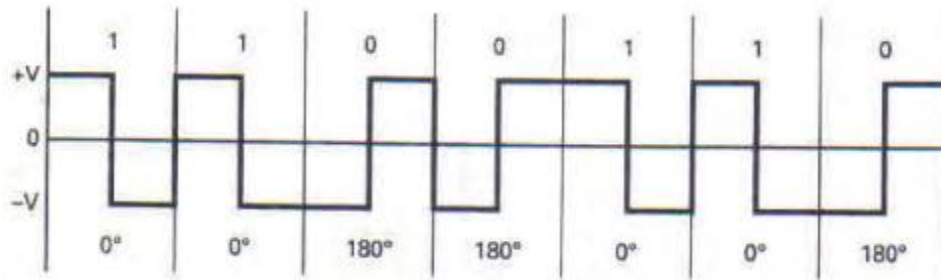


Figura 2.13: Bifásica digital

Fuente: Wayne, Tomasi (2003). *Sistemas de Comunicaciones Electrónicas*

Consideraciones de ancho de banda

Para determinar el ancho mínimo de banda necesario para propagar una señal codificada en línea, se debe determinar la mayor frecuencia fundamental que esté asociada a ella. La frecuencia fundamental máxima se determina con el peor de los casos (transición más rápida) de la secuencia de bits.

Recuperación de reloj

Para recuperar y mantener la información de sincronización a partir de los datos recibidos, debe haber una cantidad suficiente de transiciones en la señal de datos.

Detección de errores

La transmisión BPRZ-AMI, tiene incorporado el mecanismo de detección de errores debido que si se tiene un error en cualquier bit provocará una violación bipolar, permitiendo de esta manera determinar si los datos tienen errores.

Facilidad de detección y decodificación

La transmisión bipolar, es la que se adapta mejor a la detección de datos por tener una componente promedio de cd igual a 0V cuando se tiene las probabilidades iguales de ocurrencia de unos y ceros.

La tabla 2.1 muestra un resumen de los formatos de codificación de línea con sus características que presentan:

Tabla 2.1: Resumen de la codificación de línea
Fuente: Wayne, Tomasi (2003). *Sistemas de Comunicaciones Electrónicas*

Formato de codificación	Ancho mínimo de banda	CD Promedio	Recuperación de reloj	Detección de errores
UPNRZ	$f_b/2^*$	+V/2	Mala	No
BPNRZ	$f_b/2^*$	0V*	Mala	No
UPRZ	f_b	+V/4	Buena	No
BPRZ	f_b	0V*	Optima*	No
BPRZ-AMI	$f_b/2^*$	0V*	Buena	Si*

*Indica el mejor desempeño o calidad

2.4.15 Señales Pasa Banda

La modulación en pasa banda, es un proceso mediante el cual la señal digital modula una senoide llamada comúnmente onda portadora.

2.4.15.1 Modulación Digital.

La modulación digital, es el proceso mediante el cual se transforman los símbolos digitales en formas de onda adecuadas para la transmisión sobre el canal de comunicación.

2.4.15.2 Técnicas de Modulación Digital.

La figura 2.14 muestra las técnicas de modulaciones digitales que son necesarias para convertir los datos digitales en señales analógicas de tal manera que se pueda transmitir a través de líneas analógicas. Las técnicas básicas son: desplazamiento en amplitud (ASK, Amplitude-Shift Keying), desplazamiento de frecuencia (FSK,

Frecuency-Shift Keying), desplazamiento de fase (PSK, Phase Shift Keying) y modulación de amplitud en cuadratura (QAM, Quadrature Amplitude Modulation).

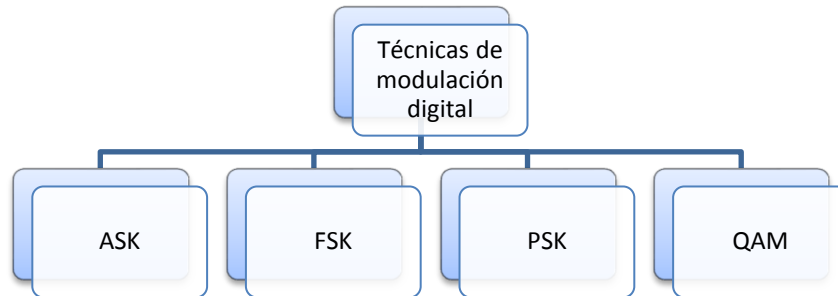


Figura 2.14: Técnicas de modulación digital

Fuente: Por el investigador

2.4.15.3 Modulación y demodulación ASK.

Modulación por desplazamiento en amplitud (ASK).- Consiste en variar la amplitud de la señal portadora en función de la señal moduladora. La frecuencia y la fase permanecen constantes. Si la señal de información es multinivel (M bits por símbolo), entonces habrán M-amplitudes posibles de la portadora.

En la figura 2.15 muestra la señal portadora, moduladora y la modulación ASK con portadora suprimida.

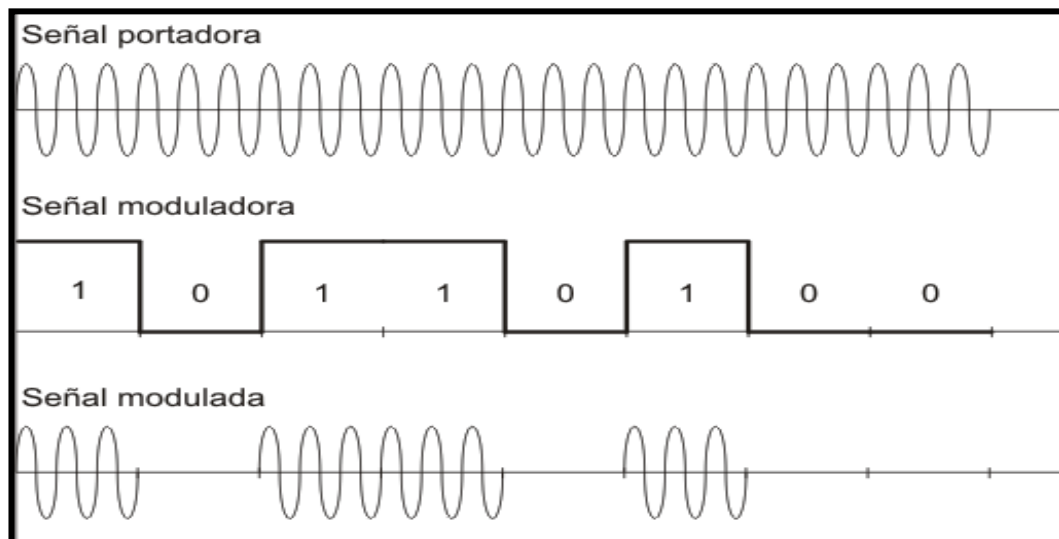


Figura 2.15: Modulación por desplazamiento en Amplitud (ASK)

Fuente: <http://www.textoscientificos.com/redes/modulacion/FSK>

La ecuación 2.6 describe la modulación en amplitud de una señal binaria.

$$V_{am}(t) = [1 + V_m(t)] \left[\frac{A}{2} \cos(w_c t) \right]$$

Ecuación 2.6: Ecuación de la modulación ASK

Donde:

$V_{am}(t)$ = voltaje de la onda de amplitud modulada

$\frac{A}{2}$ = amplitud de la portadora no modulada (volts)

$V_m(t)$ = señal binaria moduladora (volts)

w_c = frecuencia de la portadora en radianes (radianes por segundo)

En la ecuación 2.6, la señal moduladora $V_m(t)$ es una forma de onda normalizada, en lo que +1V= 1 lógico, y -1V= 0 lógico. Por consiguiente, para una entrada de 1 lógico, $V_m(t)$ = +1 y la ecuación 2.6 se reduce a

$$V_{am}(t) = [1 + 1] \left[\frac{A}{2} \cos(w_c t) \right]$$

$$V_{am}(t) = A \cos(w_c t)$$

Ecuación 2.7: Señal moduladora para 1 lógico

y para una entrada de 0 lógico, $V_m(t)$ = -1, y la ecuación 2.6 se reduce a

$$V_{am}(t) = [1 - 1] \left[\frac{A}{2} \cos(w_c t) \right]$$

$$V_{am}(t) = 0$$

Ecuación 2.8: Señal moduladora para 0 lógico

Este tipo de modulación es relativamente de baja calidad y bajo costo y, en consecuencia, se lo utiliza poco en sistemas de comunicaciones de gran capacidad y alta eficiencia.

En la figura 2.16 se muestra el diagrama de un modulador ASK que cuenta con un mezclador en donde se ingresa una señal moduladora (secuencia de unos y ceros) y una señal portadora ($\text{sen } \pi_{fp} t$), con el propósito de mezclar estas dos señales. Posteriormente se pasa esta señal mezclada, a un filtro encargada de reducir al

máximo el efecto de las frecuencias armónicas indeseadas, para así conformar la señal modulada.

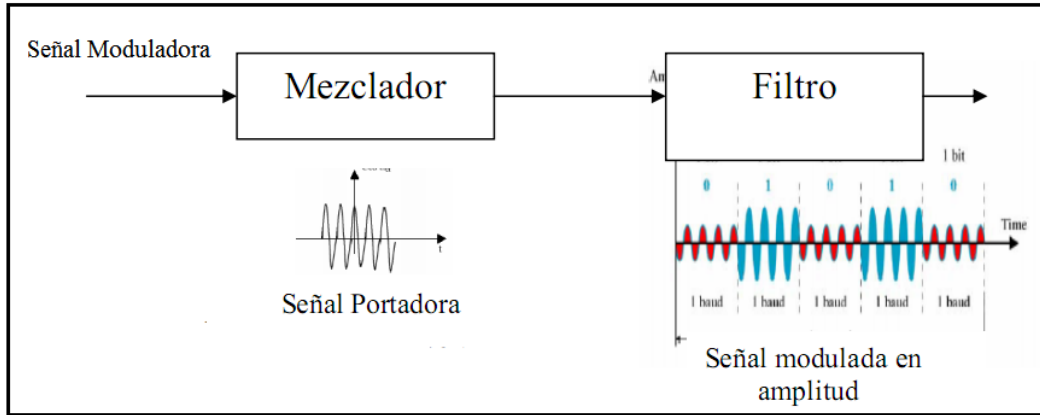


Figura 2.16: Diagrama de un modulador ASK

Fuente: Por el investigador

En consecuencia, la modulación de amplitud puede ser considerada, como el proceso de trasladar la gama de frecuencia de la señal moduladora, a una zona de frecuencias más altas, que están determinadas por la frecuencia portadora (ω_c). La figura 2.17 muestra la señal ASK en su espectro de frecuencia.

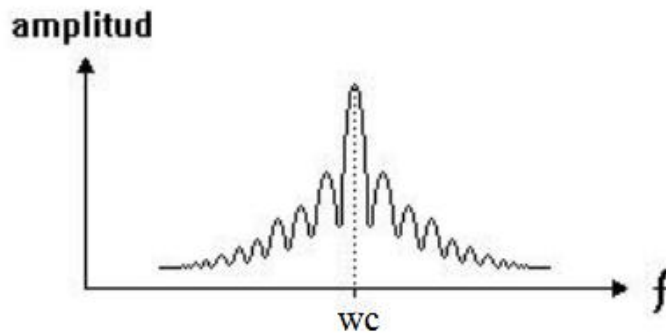


Figura 2.17: Espectro de una señal ASK

Fuente: Por el investigador

El espectro de frecuencia está expresado de la siguiente manera

$$v_{am}(t) = m(t) * p(t)$$

$$= \frac{1}{2} \cos \omega_c t + \frac{1}{\pi} \left\{ \cos(\omega_c - \omega_0) t + \cos(\omega_c + \omega_0) t - \frac{1}{3} \cos(\omega_c - \omega_0) t + \dots \right\}$$

Ecuación 2.9: Espectro de frecuencia ASK

y el ancho de banda es:

$$B = (w_c + w_0) - (w_c - w_0) = 2w_0$$

Ecuación 2.10: Ancho de banda ASK

Donde:

$v_{am}(t)$ = voltaje de la onda de amplitud modulada

$m(t)$ = señal binaria moduladora (volts)

$p(t)$ = señal binaria portadora (volts)

w_c = frecuencia de la portadora en radianes (radianes por segundo)

w_0 = frecuencia de la moduladora en radianes (radianes por segundo)

B: ancho de banda ASK (Hertz)

En la figura 2.18 muestra la representación del ancho de banda mínima ASK, que es igual a la tasa de baudio (N_{baud}) y a la tasa de símbolo por tratarse de señales binarias.

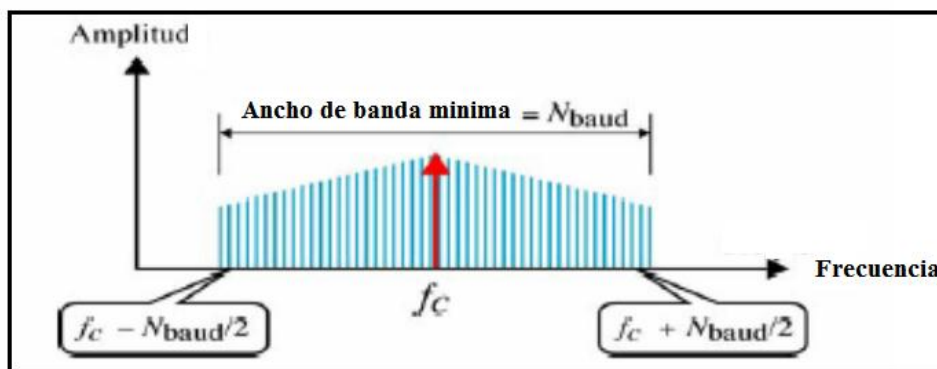


Figura 2.18: Representación ancho de banda ASK

Fuente: Por el investigador

La ecuación 2.11 indica la señal modulada ASK, para diferentes niveles de energía:

$$v(t) = \sqrt{E_i} \sqrt{\frac{2}{T}} \cos(w_c t + \varphi) \quad 0 \leq t \leq T$$

Ecuación 2.11: Constelación ASK

donde:

$v(t)$ = señal modulada (volts)

$E_i t$ = amplitud de los M-valores discretos (volts)

T = duración del símbolo (segundos)

ω_c = frecuencia de la portadora en radianes (radianes por segundo)

El **diagrama de constelación**, es muy importante para determinar la fortaleza que tiene la modulación frente al ruido. Mientras mas distanciados se encuentren los símbolos, será mas difícil que un símbolo se convierta en el otro por efectos del ruido añadido en el sistema.

La constelación consiste en representar la señal modulada en funciones unitarias ($\sqrt{E_i}$). En la figura 2.19 se muestra la constelación ASK binaria, en donde el espacio de la señal es unidimensional por lo tanto, todas las señales generadas estarán ubicadas sobre una línea recta. Además, se indica los dos símbolos 0 lógico y el 1 lógico por tratarse de una modulación binaria 2-ASK.

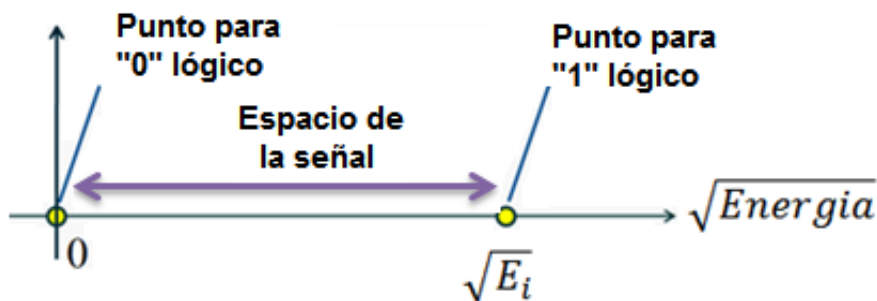


Figura 2.19: Grafica constelación ASK

Fuente: Por el investigador

Demodulación ASK.

En la demodulación se tiene dos tipos distintos que son: coherente e incoherente, la demodulación coherente posee problemas de sincronización y es más compleja su implementación en comparación con la demodulación no coherente.

A continuación se explica de manera resumida los tipos de demodulación ASK con sus respectivos diagramas de bloque:

- **Demodulación coherente ASK.-** En este tipo de modulación es necesario disponer de una replica de la portadora del transmisor, sincronizada en frecuencia y fase. El detector digital deberá comparar el nivel de tensión con los niveles prefijados, para extraer los bits enviados por el transmisor. La figura 2.20 muestra el diagrama de bloques del demodulador coherente ASK.

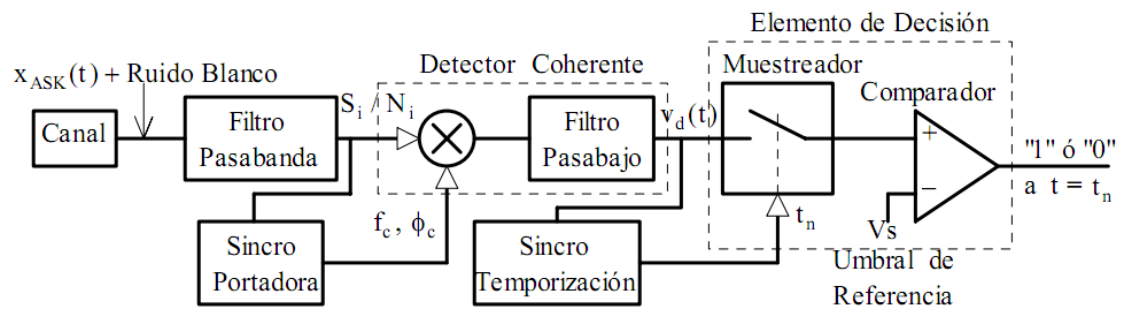


Figura 2.20: Demodulador coherente ASK

Fuente: Se encontró en la pagina web "<http://webdelprofesor.ula.ve/ingenieria/jbriceno/transmisiones/comdiP2.pdf>"

- **Demodulación no coherente ASK.-** La figura 2.21 muestra la demodulación no coherente ASK que permite extraer la información mediante un rectificador y un detector de pico, que detecta la envolvente de la señal modulada.

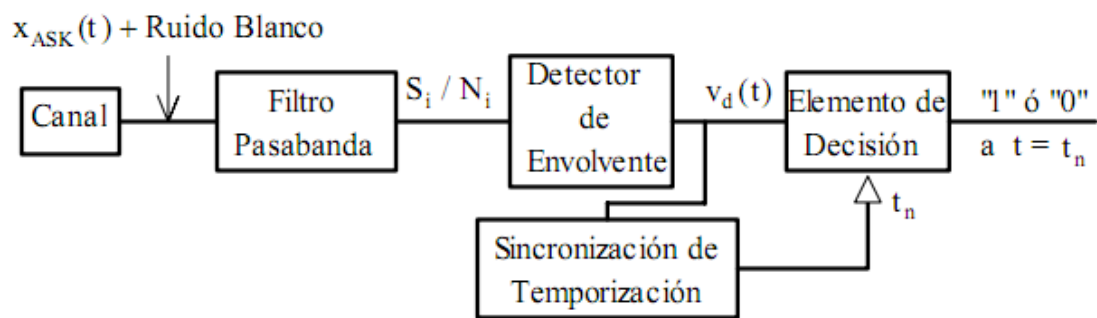


Figura 2.21: Demodulador no coherente ASK

Fuente: Se encontró en la pagina web "<http://webdelprofesor.ula.ve/ingenieria/jbriceno/transmisiones/comdiP2.pdf>"

2.4.15.4 Modulación y demodulación FSK

Modulación por desplazamiento en frecuencia (FSK).- La modulación por desplazamiento en frecuencia consiste en asignar una frecuencia diferente a cada símbolo de la señal moduladora. En la figura 2.22 se puede observar la señal modulada desplazada en frecuencia.

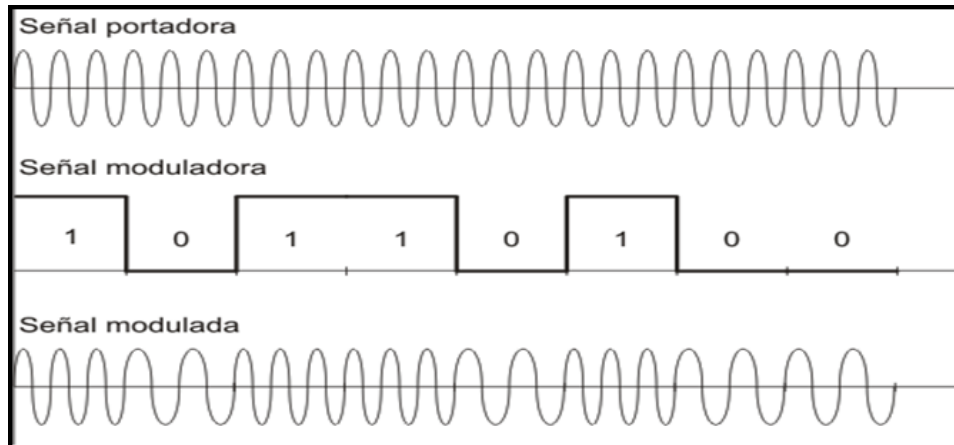


Figura 2.22: Modulación por desplazamiento en Frecuencia (FSK)

Fuente: <http://www.textoscientificos.com/redes/modulacion/FSK>

La ecuación general de la FSK binaria es

$$V_{fsk}(t) = V_c \cos\{2\pi [f_c + V_m(t)\Delta f]t\}$$

Ecuación 2.12: Modulación FSK

en donde

$V_{fsk}(t)$ = forma de onda binaria FSK

V_c = amplitud de la portadora (volts)

f_c = frecuencia central de la portadora (hertz)

Δf = desviación máxima de frecuencia (hertz)

$V_m(t)$ = señal moduladora de entrada binaria (± 1)

De acuerdo con la ecuación 2.12, la desviación máxima de frecuencia de la portadora, Δf , es proporcional a la amplitud y a la polaridad de la señal binaria de entrada. La señal moduladora $v_m(t)$ es una forma de onda binaria normalizada, en la que 1 lógico = 1, y el 0 lógico = -1.

Cuando la entrada es 1 lógico, $v_m(t) = +1$, y la ecuación 2.12 se deduce

$$V_{fsk}(t) = V_c \cos[2\pi(f_c + \Delta f)t]$$

Para una entrada de 0 lógico, $v_m(t) = -1$, y la ecuación 2.12 se transforma en

$$V_{fsk}(t) = V_c \cos[2\pi(f_c - \Delta f)t]$$

Con una FSK binaria, la señal binaria de entrada desvía a la frecuencia de la portadora. Cuando la señal binaria de entrada cambia de un 0 lógico a un 1 lógico y viceversa, la frecuencia de salida se desplaza entre dos frecuencias: una frecuencia de marca (f_m) o 1 lógico, y una frecuencia de espacio (f_s) o de 0 lógico. Las frecuencias de marca y de espacio están separadas de la frecuencia de portadora por la desviación máxima de frecuencia, es decir, por $f_c \pm \Delta f$.

En la figura 2.23, la frecuencia de marca es la frecuencia mayor, $f_c + \Delta f$, y la frecuencia de espacio es la menor, $f_c - \Delta f$.

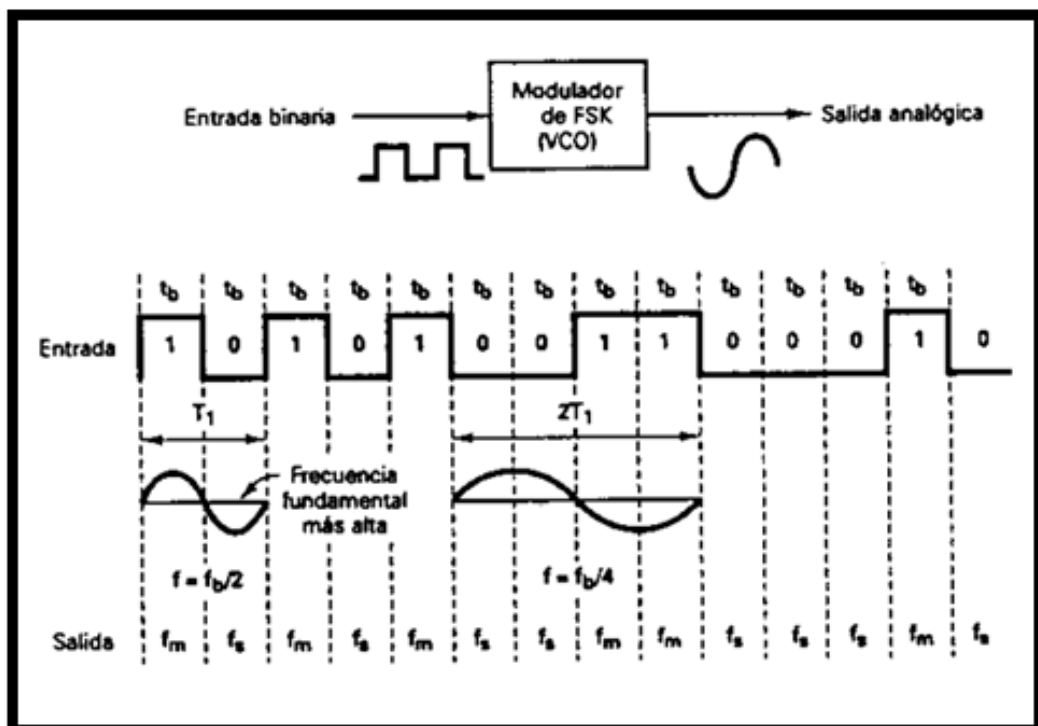


Figura 2.23: Formas de onda de entrada y salida binaria FSK

Fuente: Wayne, Tomasi (2003). *Sistemas de Comunicaciones Electrónicas*

Con la FSK binaria, la rapidez de cambio de salida es igual a la rapidez de cambio en la entrada. En la modulación digital, la rapidez de cambio en la entrada del modulador se llama frecuencia de bits (fb), y su unidad de medida es bis por segundo (bps). La rapidez de cambio de salida del modulador se llama baudio. El baudio es una rapidez de cambio, y es igual a la reciproca del tiempo de un elemento de señalización a la salida. En la manipulación por frecuencia (FSK), el tiempo de señalización en la salida es igual al tiempo de un solo bits, tb. Como se puede ver en la figura 2.23, la frecuencia de salida cambia de una frecuencia de marca a una de espacio y viceversa, con la misma rapidez con la que la condición de entrada cambia de 1 a 0 lógico y viceversa.

En la figura 2.24 muestra un transmisor FSK binario sencillo, que es similar a un modulador convencional FM, y que a menudo es un oscilador controlado por voltaje (VCO, voltaje-controlled oscillator). La frecuencia central de portadora se elige de tal modo que esté a la mitad entre las frecuencias de marca y de espacio.

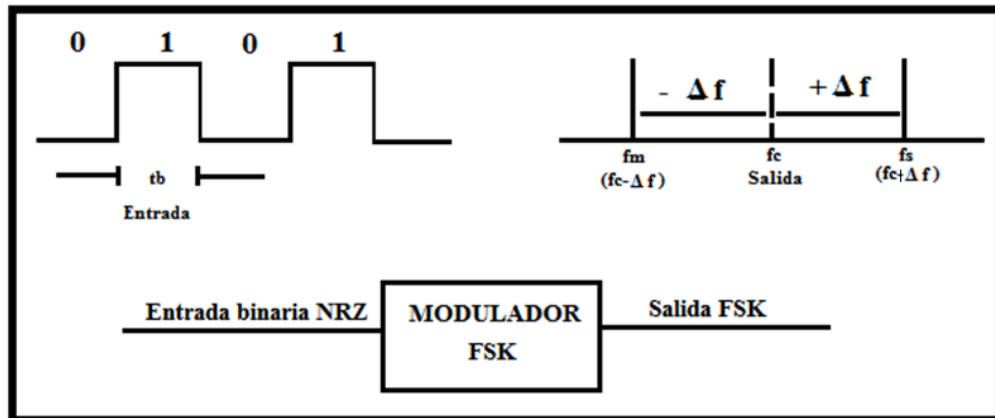


Figura 2.24: Desviación de frecuencia
Fuente: Por el investigador

Un modulador VCO-FSK puede funcionar en el modo de barrido, en el que la desviación máxima de frecuencia no es más que el producto del voltaje binario de entrada por la sensibilidad del VCO a la desviación. En modo de barrido la desviación máxima de frecuencia se expresa como sigue

$$\Delta f = v_m(t)k_f$$

en donde:

Δf = desviación máxima de frecuencia (hertz)

$v_m(t)$ = voltaje máximo de señal moduladora binaria (volts)

k_f = sensibilidad a la desviación (hertz por volt)

En FSK binaria, la amplitud de la señal de entrada sólo puede tener uno de dos valores: uno para una condición de 1 lógico y uno para una condición de 0 lógico. En consecuencia, la desviación de frecuencia es constante, y siempre tiene su valor máximo.

La salida de un modulador FSK se relaciona con la entrada binaria en la forma que se ve en la figura 2.24, donde un 0 lógico corresponde a la frecuencia de espacio f_s , un 1 lógico corresponde a la frecuencia de marca f_m , y f_c es frecuencia de portadora. La desviación máxima de frecuencia se determina con

$$\Delta f = \frac{[f_m - f_s]}{2}$$

Ecuación 2.13: Desviación máxima de frecuencia

en la que

Δf = desviación máxima de frecuencia (hertz)

f_m = frecuencia de marca (hertz)

f_s = frecuencia de espacio (hertz)

En la figura 2.24, se puede apreciar dos ondas senoidales pulsadas, de frecuencia f_m y f_s . Las ondas senoidales pulsadas tienen espectros de frecuencias que son funciones sen x/x, la figura 2.25 muestra el espectro de salida de una señal FSK. Suponiendo que los máximos del espectro de potencia contienen mayor parte de la energía, el ancho de banda mínimo para pasar una señal FSK se puede deducir

$$\begin{aligned} B &= [\langle f_s + fb \rangle - \langle f_m - fb \rangle] \\ &= \langle |f_s - f_m| \rangle + 2fb \end{aligned}$$

como $\langle |f_s - f_m| \rangle = 2\Delta f$, el ancho mínimo de banda es

$$\begin{aligned} B &= 2\Delta f + 2fb \\ &= 2\langle \Delta f + fb \rangle \end{aligned}$$

en donde

B= ancho mínimo de banda (hertz)

Δf = desviación mínima de frecuencias máximas (hertz)

f_m = frecuencia de marca (hertz)

f_s = frecuencia de espacio (hertz)

Cuando se tiene una secuencia alternada de unos y ceros, se genera una onda cuadrada. Que necesita de un estado alto y bajo para producir un ciclo, la máxima frecuencia fundamental contenida en una onda cuadrada es igual a la rapidez de repetición de esa onda cuadrada con la cual una señal binaria es igual a la mitad de la rapidez de bits. Es decir,

$$f_a = f_b/2$$

en donde

f_a = máxima frecuencia fundamental de la señal moduladora binaria (hertz)

$f_b/2$ = rapidez de bits (bits por segundo)

El índice de modulación, es la desviación máxima de fase. Su formula es similar al índice de modulación de FM, entonces

$$h = \frac{\Delta f}{f_a} \text{ (adimensional)}$$

Ecuación 2.14: Índice de modulación

en la que

h = índice de modulación, que en FSK se llama factor h

f_a = frecuencia fundamental de la señal moduladora binaria (hertz)

Δf = desviación máxima de frecuencia (hertz)

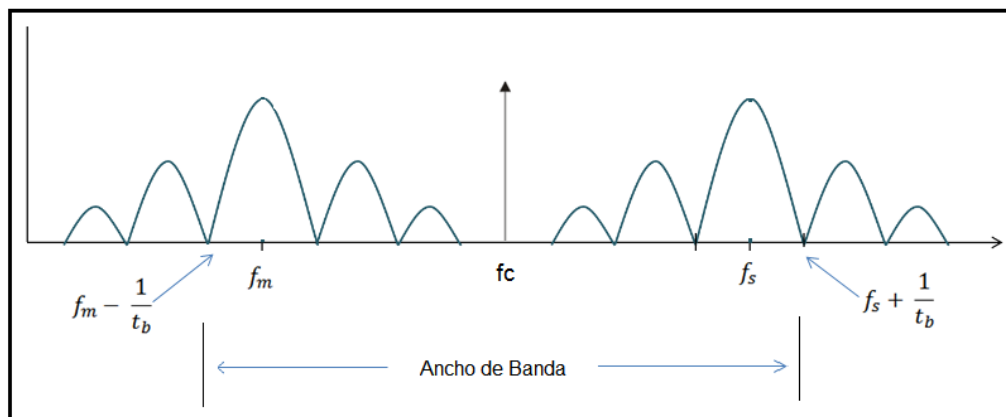


Figura 2.25: Espectro de una señal FSK

Fuente: Por el investigador

La figura 2.26 muestra la constelación FSK, cuando la frecuencia de marca y espacio son ortogonales en un intervalo t_b , Al aumentar el número de niveles la distancia entre los puntos en el espacio se mantiene constante $\sqrt{2E}$.

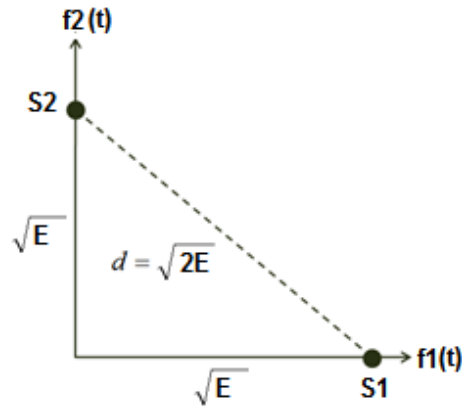


Figura 2.26: Constelación FSK
Fuente: Por el investigador

DEMODULACION FSK

La demodulación FSK es sencilla, con un circuito como de la figura 2.27. La señal de entrada FSK se aplica en forma simultanea a las entradas de los dos filtros pasabanda (BPF) a través de un divisor de potencia. El filtro respectivo sólo pasa la frecuencia de marca o sólo la de espacio a su respectivo detector de envolvente. A su vez, los detectores de envolvente indican la potencia total en cada banda pasante, y el comparador responde a la mayor de las dos potencias. A este tipo de detección FSK se llama **demodulador no coherente**; no hay frecuencia, implicada en el proceso de demodulación, que esté sincronizada en fase, o en frecuencias o en ambas, con la señal FSK, que entra.

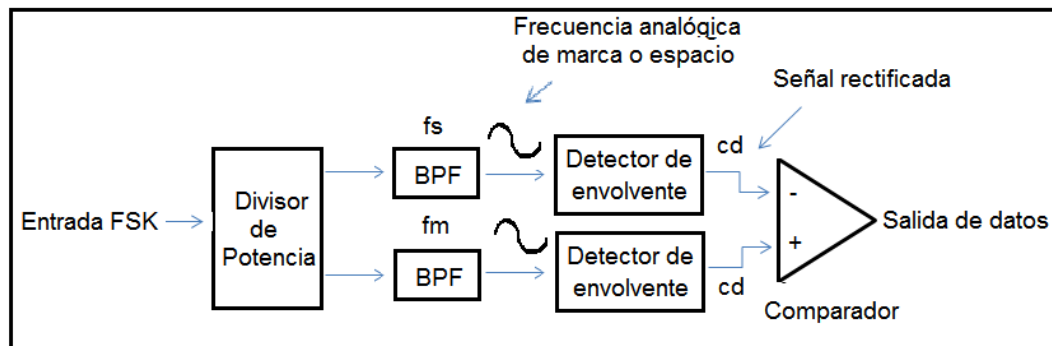


Figura 2.27: Demodulador FSK no coherente
Fuente: Por el investigador

En la figura 2.28 muestra el diagrama de bloques de un **demodulador FSK coherente**. La señal FSK de entrada se multiplica por una señal de portadora recuperada que tenga exactamente la misma frecuencia y fase que la referencia del transmisor. Sin embargo, las dos frecuencias transmitidas, de marca y de espacio, en general no son continuas. Por lo tanto, la detección FSK coherente, no se usa.

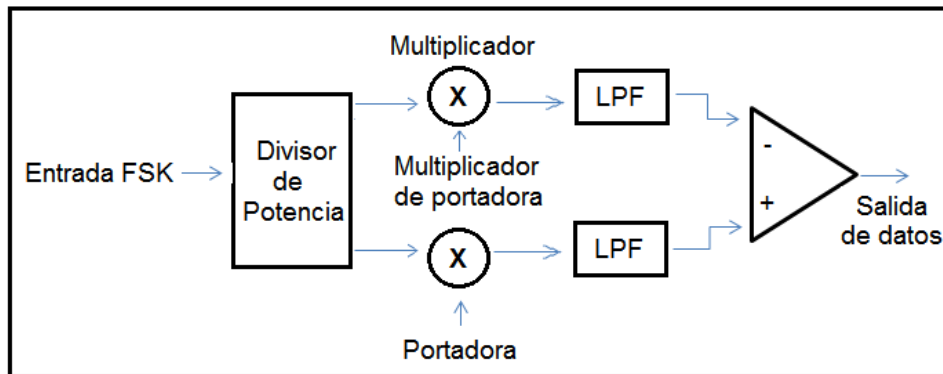


Figura 2.28: Demodulador FSK coherente
Fuente: Por el investigador

2.4.15.5 Modulación y demodulación PSK.

Modulación por desplazamiento de fase (PSK).- La modulación PSK, es una forma de modulación angular que consiste en hacer variar la fase de la portadora entre un número de valores discretos. En la figura 2.29 muestra la señal modulada con sus discontinuidades de fase que aparecen al comienzo y al final de cada intervalo T, cuando hay transición de 0 a 1 o viceversa.

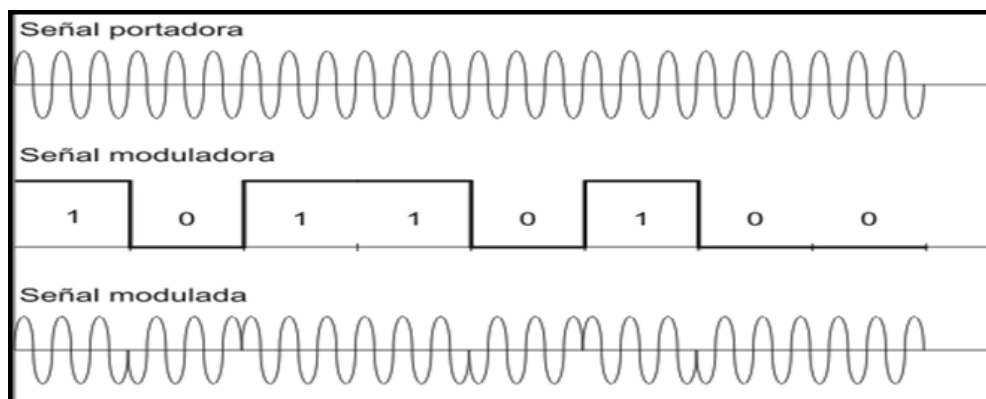


Figura 2.29: Modulación Digital de Fase (PSK)
Fuente: http://es.wikipedia.org/wiki/Modulaci%C3%B3n_de_fase

La señal modulada resultante, corresponde a la siguiente expresión:

$$V_{PSK} = V_m(t) \cdot A \cos(\omega_c t)$$

donde

V_{PSK} = voltaje de la onda de amplitud modulada

$V_m(t)$ = señal binaria moduladora (volts)

A = amplitud de la portadora no modulada (volts)

w_c = frecuencia de la portadora en radianes (radianes por segundo)

Existen 2 alternativas de modulación PSK que dependen de si la demodulación es o no coherente:

- **PSK convencional.**- En este tipo de modulación, es necesario tener una portadora en el receptor para la sincronización, o usar un código autosincronizante. Esta se refiere a cuando en el instante de asignar la frecuencia se mantiene la fase de la señal.
- **PSK diferencial (DPSK).**- En la modulación PSK diferencial, no se necesita recuperar la señal portadora para realizar la demodulación. Es diferencial puesto que la información esta contenida en las transiciones. La referencia de fase se toma del intervalo inmediato anterior, con lo que el detector decodifica la información digital basándose en diferencias relativas de fase.

En la figura 2.30 se muestra el diagrama de bloques de un modulador PSK, en la que el modulador balanceado actúa como un conmutador para invertir la fase, dependiendo de los datos binarios de entrada la señal portadora desplaza su fase 180 grados con el oscilador de la portadora de referencia.

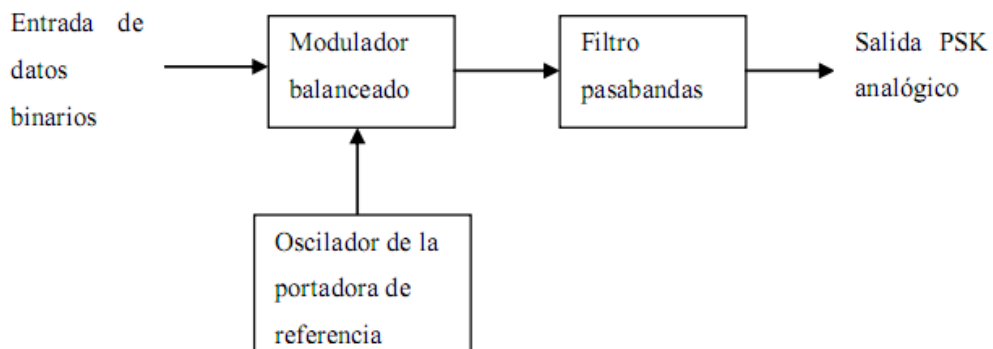


Figura 2.30: Modulador PSK

Fuente: LEOPOLDO, Javier. (2010). "Integración del Downconverter NI 5660 a través de instrumentos virtuales bajo labview para generar diseños de referencia y procedimientos aplicables a sistemas de telecomunicaciones".

El espectro de salida de un modulador PSK binario es el mismo de una señal de doble banda lateral con portadora suprimida, donde sus bandas laterales están separadas por la frecuencia de portadora con un valor igual a la mitad de la razón de bit.

La modulación PSK consiste en desplazar la fase de la señal portadora y es conocido que toda señal sinusoidal desfasada se puede descomponer en un seno y un coseno, de esta manera las funciones unitarias del espacio de señales de PSK (f_1 y f_2) se pueden expresar de la siguiente forma:

$$f_1(t) = \sqrt{\frac{2}{T}} * \cos w_c t$$

$$f_2(t) = \sqrt{\frac{2}{T}} * \text{sen}w_c t$$

Al tener dos funciones unitarias se dice que el espacio de las señales de PSK (S_1 y S_2) es bidimensional, siendo la señal de entrada binaria. Por lo tanto, tendremos:

$$S_1 = \sqrt{\frac{2E}{T}} * \text{sen}(w_c t + \frac{\pi}{2})$$

$$S_2 = \sqrt{\frac{2E}{T}} * \cos(w_c t - \frac{\pi}{2})$$

En la figura 2.31 muestra el diagrama de constelación de una señal PSK binaria, conocida como BPSK (modulación por desplazamiento binario de fase). En este tipo de modulación son posibles dos fases de salida para una sola frecuencia de portadora. Una fase de salida representa 1 lógico, y la otra un 0 lógico.

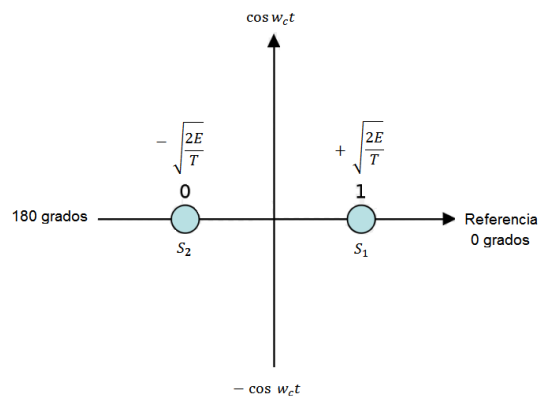


Figura 2.31: Constelación BPSK

Fuente: Por el investigador

Codificación M-aria

M-ario, es un termino que proviene de la palabra binario. M solo es un dígito que representa la cantidad de condiciones posibles para una determinada cantidad de variables binarias.

La cantidad de condiciones posibles de salida se calcula con la ecuación

$$N = \log_2 M$$

en donde

N = cantidad de bits codificados

M = cantidad de condiciones posibles de salida con N bits

Por ejemplo, con FSK binaria, cada bit de entrada actúa en forma independiente sobre la portadora y produce una de las dos frecuencias posibles de salida. De esta manera:

$$N = \log_2 2$$

$$2^N = 2$$

pasando a logaritmos se despeja N y se obtiene:

$$\log 2^N = \log 2$$

$$N \log 2 = \log 2$$

$$N = \frac{\log 2}{\log 2}$$

$$N = 1$$

Como resultado se obtuvo que con 2 combinaciones posibles de salida, se tenga 1 bit codificado.

Demodulación PSK

La demodulación PSK, es de tipo coherente. En la figura 2.32 muestra el diagrama de bloques de un receptor PSK que tiene similitud con el receptor ASK coherente pero con la diferencia que en el receptor PSK, el elemento de decisión es mucho mas sencilla que trata de determinar la polaridad (positiva o negativa) de la envolvente de las señal filtrada $v_d(t)$ en el instante de tiempo (t_n).

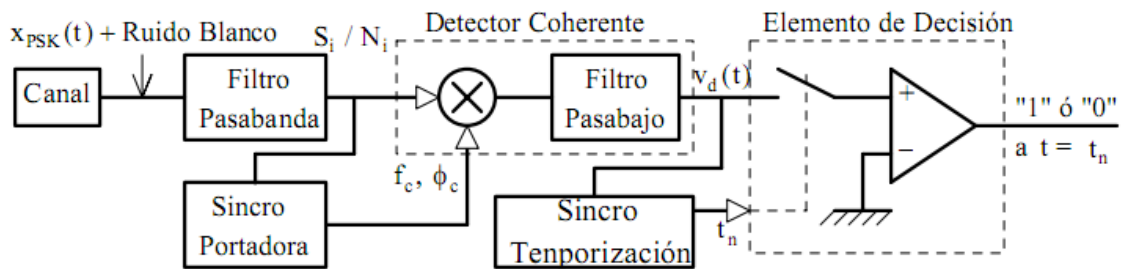


Figura 2.32: Demodulador PSK

Figura: http://blearning.itmina.edu.mx/dep/sada/carreras/Ingenieria%20en%20Sistemas%20Computacionales/5to%20Semestre/Teoria%20de%20las%20Telecomunicaciones/teoria_telecomunicaciones/CAPITULO%20II.pdf

2.4.15.6 Modulación y demodulación QAM.

Modulación de amplitud en cuadratura (QAM).- Es una técnica de modulación digital multinivel, donde la información está contenida tanto en la amplitud como en la fase de la portadora transmitida. Este método se basa en la transmisión de dos mensajes independientes transportados por un mismo camino, esto se logra modulando una misma portadora desfasada 90° entre un mensaje y otro.

La ecuación matemática de una señal modulada en QAM, es:

$$V_{QAM} = a_n \cos w_c t + b_n(t) \sin w_c t$$

Ecuación 2.15: Modulación QAM

Donde

V_{QAM} = voltaje de la onda modulada en cuadratura

a_n y b_n = amplitudes de las dos señales moduladas en ASK (volts)

w_c = frecuencia de la portadora en radianes (radianes por segundo)

En la figura muestra un ejemplo de modulación 8-QAM (modulated result) que es el resultado de las variaciones de la señal portadora (carrier) con respecto a la información digital. Esta información contiene 3 valores, representados por tres bits cada uno.

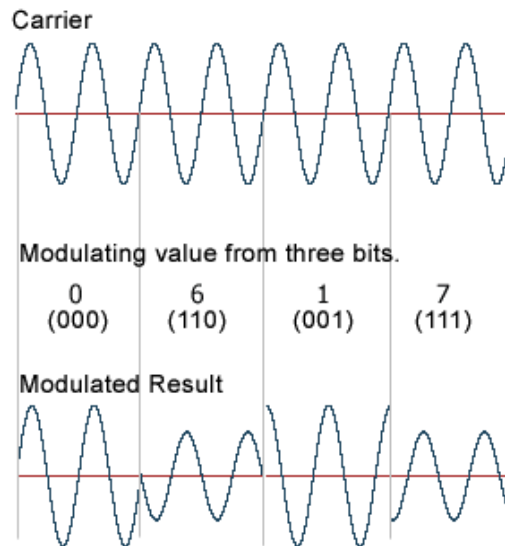


Figura 2.33: Modulación de amplitud en cuadratura (8-QAM)
Fuente: http://www.ecured.cu/index.php/Modulaci%C3%B3n_QAM

En la figura 2.34 se muestra el modulador QAM que esta conformado por portadoras moduladas P, Q que son dos canales de transmisión QAM diferentes. La separación de estos canales y la demodulación se efectúa en el demodulador. En este modulador se tiene en su entrada datos binarios que provienen de un transmisor, con sus frecuencias de muestreo (f_s), frecuencia de bit (f_b) y los demás elementos permiten modular en QAM.

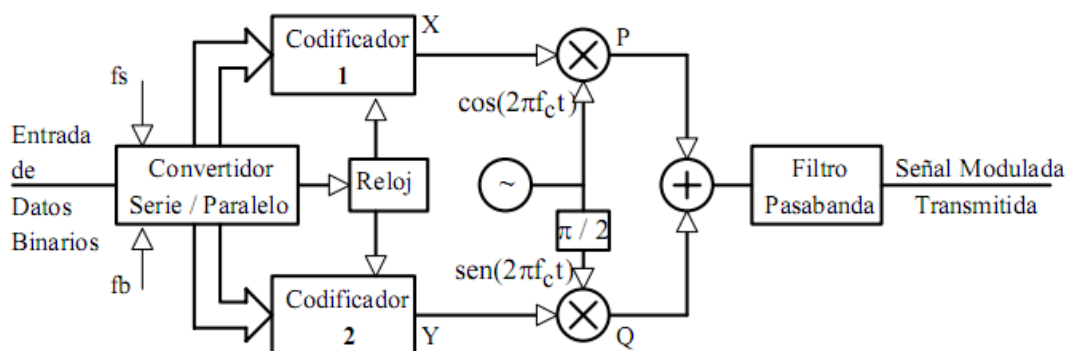


Figura 2.34: Modulador QAM
Fuente: http://blearning.itmina.edu.mx/dep/sada/carreras/Ingenieria%20en%20Sistemas%20Computacionales/5to%20Semestre/Teoria%20de%20las%20Telecomunicaciones/teoria_telecomunicaciones/CAPITULO%20II.pdf

Diagrama de constelación para QAM

Los diagramas de constelación muestran las diferentes posiciones de los estados dentro de las diferentes formas de QAM, Quadrature Amplitude Modulation. A medida que el aumento el nivel de modulación, también lo hace el número de puntos en el diagrama de la constelación QAM. En la figura 2.35, se muestra el diagrama de constelación para diferentes valores de nivel para la modulación QAM.

A manera de ejemplo tenemos el 8-QAM, que es una técnica de codificación M-aria en la que $M=8$, la salida que presenta un modulador 8-QAM no es una señal de amplitud constante. En el 8-QAM, los datos que llegan se dividen en grupos de tres bits (tribits): las corrientes I, Q, C de bits, cada una con rapidez de bits igual a la tercera parte de la rapidez de entrada de datos. Las fases posibles se determinan de acuerdo a la ecuación de M-aria:

$$M = 2^N$$

$$M = 2^3$$

$$M = 8 \text{ fases posibles}$$

Los puntos que presentan las graficas de la figura, son las amplitudes y las fases correspondientes a un transmisor 8-QAM, 16-QAM, 32-QAM y 64-QAM.

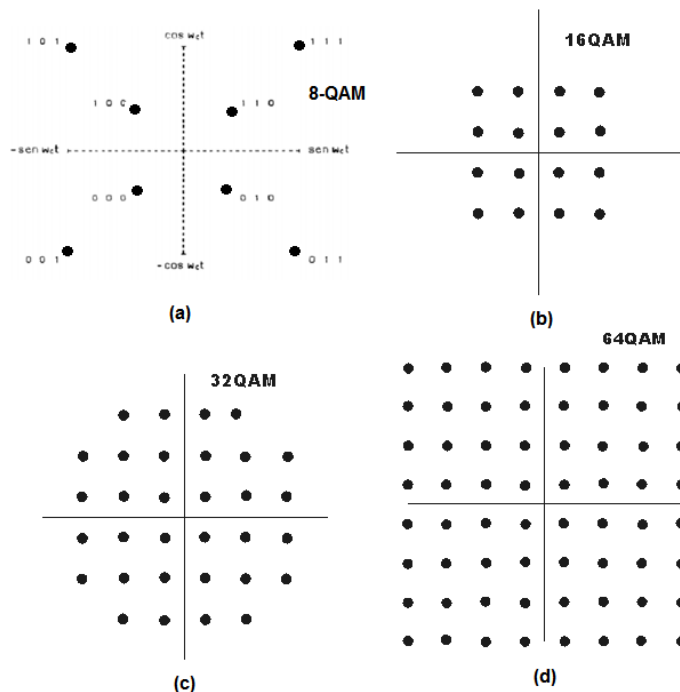


Figura 2.35: Constelación QAM: (a) 8-QAM; (b) 16-QAM; (c) 32-QAM; (d) 64-QAM

Fuente: Por el investigador

El ancho de banda para este tipo de modulación es el doble del ocupado por la señal digital original que lleva la información, ecuación 2.16.

$$\beta_{QAM} = 2\beta_F$$

Ecuación 2.16: Ancho de banda QAM

Donde:

β_F = ancho de banda de la señal digital de información (hertz)

Al relacionar la velocidad de transmisión en baudios (símbolo*segundo), los bits de transmisión y la velocidad de transmisión en bits por segundo (bps), tenemos lo siguiente:

$$V_s = 2\beta_f$$

$$V_b = kV_s ; M=2^k$$

$$B_{QAM} = 2\beta_f = 2 \frac{V_s}{2} = V_s = \frac{V_b}{k}$$

Con la ecuación concluimos que el ancho de banda requerido para transmitir una señal QAM, es el cociente entre la velocidad de transmisión y los bits por segundo.

Demodulación QAM.

La figura 2.36 muestra un diagrama de bloques de un demodulador QAM. En la demodulación se requiere la regeneración de las frecuencias de portadora y de señalización sincrónica con la correspondientes en el transmisor, al inicio de la transmisión se utilizan secuencias de sincronización para lograr la sincronización de los relojes.

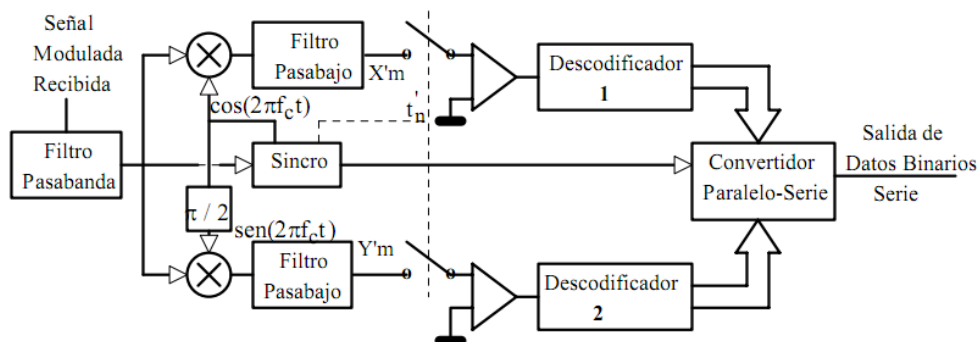


Figura 2.36: Demodulación QAM

Fuente: http://blearning.itmina.edu.mx/dep/sada/carreras/Ingenieria%20en%20Sistemas%20Computacionales/5to%20Semestre/Teoria%20de%20las%20Telecomunicaciones/teoria_telecomunicaciones/CAPITULO%20II.pdf

2.5 Hipótesis

La subutilización de un laboratorio virtual influye en el desarrollo de prácticas de modulación y demodulación digital en el laboratorio de electrónica de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

2.6 Señalamiento de variables

Variable independiente: Laboratorio virtual.

Variable dependiente: Modulación y demodulación digital.

CAPITULO III

3. METODOLOGÍA

3.1 Enfoque de la investigación

Esta investigación se basó en el enfoque cuali-cuantitativo, debido a que la recolección de datos se realizó directamente desde los laboratorios de electrónica de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, se llevó a cabo una exploración profunda, haciendo de esta investigación un estudio objetivo y controlado cuyas respuestas se pretende sean confiables, a más de esto se tomó datos de calidad buscando las causas y la explicación de los hechos a estudiarse, donde ésta se basó en una realidad estable.

3.2 Modalidad básica de investigación

La presente investigación se contextualizó en la modalidad de campo y documental – bibliográfica.

De campo; porque se realizó un estudio sistemático de los hechos en el laboratorio de electrónica, lugar donde se producen los acontecimientos; **documental bibliográfica;** porque se tiene como propósito detectar, profundizar, ampliar diferentes enfoques, teorías, conceptualizaciones y criterios en todo lo relacionado a prácticas de modulación-demodulación digital.

3.3 Tipos de investigación

La investigación estuvo sustentada con un nivel **exploratorio** pues se reconoce las variables que nos competen a las cuales se da una mayor amplitud y dispersión. Un nivel **descriptivo** permitiendo dar pronósticos básicos, para lo cual se requiere un conocimiento suficiente de la situación. El nivel **explicativo** detectando las causas de determinados comportamientos, explicando los factores precisos de ciertos procedimientos. Por último la **asociación de variables** también estará presente evaluando las variables de comportamiento, midiendo el grado de relación entre las mismas.

3.4 Población y muestra

Población

En la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, la población a tomar en cuenta en el presente trabajo de investigación, consta de:

Docentes del Área de Comunicaciones.....	6
Coordinador de la Carrera de Comunicaciones.....	1
Estudiantes de Electrónica y Comunicaciones de 7mo, 8vo y 9no semestre.....	69
TOTAL	76

Muestra:

Tomando en consideración el tamaño de la población se trabajó con todos sus componentes integrados por el coordinador de carrera, docentes y los estudiantes de electrónica y comunicaciones respectivamente, lo cual nos permitirán obtener resultados más confiables.

3.5 Operacionalización de variables

Tabla 3.1: Operacionalización de la variable independiente: Laboratorio virtual

CONCEPTO	DIMENSIONES	INDICADORES	ÍTEMS BÁSICOS	TÉCNICAS INSTRUMENTALES
Laboratorio virtual: es una herramienta flexible, basado en un modelo matemático de sistemas físicos, que permite la simulación de distintos subsistemas, en tiempo real.	Herramientas Modelo matemático Simulación Aplicación	Software Librerías y funciones Efectividad Robótica, comunicación inalámbrica, modulación etc.	¿Las herramientas, utilizadas en los laboratorios de electrónica de la FISEI son actuales? ¿En los laboratorios de electrónica se hace uso de modelos matemáticos para realizar proyectos? ¿Considera que la simulación de un laboratorio virtual, es efectiva para contrastar el aprendizaje teorico ? ¿Cree usted, que las aplicaciones de un laboratorio virtual permiten simular prácticas complejas de realizar?	Encuesta mediante un cuestionario dirigido a los estudiantes de 7mo, 8vo y 9no semestre de la carrera de electrónica y comunicaciones de la FISEI en la UTA.

Tabla 3.2: Operacionalización de la variable dependiente: Modulación y demodulación digital.

CONCEPTO	DIMENSIONES	INDICADORES	ÍTEMES BÁSICOS	TÉCNICAS INSTRUMENTALES
<p>Modulación y demodulación digital: son un conjunto de técnicas para que la señal digital sea convertida en ondas, de manera que se pueda transportar información sobre una portadora o extraerla.</p>	<p>Técnicas de modulación</p> <p>Señal digital</p> <p>Transmisión</p>	<p>Parámetros</p> <p>Datos</p> <p>Sincronización</p>	<p>¿Cree usted que modificar los parámetros, de las técnicas de modulación y demodulación digital es de gran interés?</p> <p>¿Sabe cómo se realiza los procesos de modulación y demodulación digital, con datos digitales en tiempo real?</p> <p>¿Conoce usted elementos para medir y visualizar los diferentes métodos de modulaciones digitales?</p>	<p>Observación a través de una ficha aplicada en los laboratorios de FISEI</p> <p>Encuesta mediante un cuestionario dirigido a los estudiantes de 7mo, 8vo y 9no semestre dela carrera de electrónica y comunicaciones de la FISEI en la UTA.</p>

3.6 Técnicas e instrumentos de investigación

Las técnicas que se ha empleado en la presente investigación es: la observación y la encuesta.

Observación.- Mediante una ficha de observación realizada a los laboratoristas de electrónica se logró confirmar la carencia de instrumentos para realizar prácticas de modulaciones digitales y los métodos de codificación de línea. Posteriormente se ha recogido los datos para su análisis e interpretación, que permita llegar a las conclusiones para la toma de decisiones.

Encuesta.- La encuesta a diferencia de la observación, la recolección de información se realizó por escrito a docentes en el área de comunicaciones y a los estudiantes de 7mo, 8vo y 9no semestre de la carrera de electrónica y comunicaciones en la FISEI. La encuesta se lo realizó mediante interrogantes técnicos, referentes a modulaciones digitales.

3.7 Recolección de la información

Tabla 3.3: Plan de recolección de información

Preguntas Básicas	
¿Para qué?	Para alcanzar los objetivos de la investigación
¿De qué personas u objetos?	Docentes en el área de comunicaciones y estudiantes de 7mo, 8vo y 9no semestre de la carrera de electrónica y comunicaciones en la FISEI.
¿Sobre qué aspectos?	Constelación de ideas de las variables
¿Quién?	Investigador: Wilson Llumán

¿Cuándo?	6 meses a partir de la aprobación
¿Dónde?	Facultad de Ingeniería en Sistemas, Electrónica e Industrial
¿Cuántas veces?	Las necesarias
¿Qué técnicas de recolección?	Observación y encuesta
¿Con qué?	Cuestionarios

3.8 Procesamiento de la información

- Revisión crítica de la información recurrida, de manera que se pueda tener un esquema ordenado del fundamento teórico para poder efectuar la implementación de un laboratorio virtual y su influencia en el desarrollo de prácticas de modulación y demodulación digital para los laboratorios de electrónica.
- Repetición de la recolección en casos para corregir fallas de contestación.

CAPITULO IV

4. ANALISIS E INTERPRETACION DE RESULTADOS

4.1 Análisis

A continuación se presenta la tabulación de los datos recogidos en la encuesta realizada a los docentes, estudiantes de electrónica y comunicaciones. Además, se utilizó una ficha de observación que fue contestada por la persona encargada de los laboratorios de electrónica.

Los datos obtenidos de este trabajo de investigación y su análisis respectivo de cada pregunta de estos instrumentos como son las encuestas y la ficha de observación, permitió obtener resultados confiables. Estos datos se los ha presentado en gráficos estadísticos; en los que se puede apreciar de manera ordenada las respuestas obtenidas de la encuesta y para la ficha de observación se utilizó una tabla que consta del total de respuestas contestadas si o no.

**ENCUESTA A ESTUDIANTES DE 7mo, 8vo y 9no SEMESTRE DE LA
CARRERA DE ELECTRONICA Y COMUNICACIONES**

1. Pregunta 1: Considera usted importantes los sistemas de comunicaciones digitales:

- a) Si
- b) No

Tabla 4.1: Tabulación-Pregunta 1

Ítems	Nro. de Personas	Porcentaje
Si	69	100
No	0	0
TOTAL:	69	100

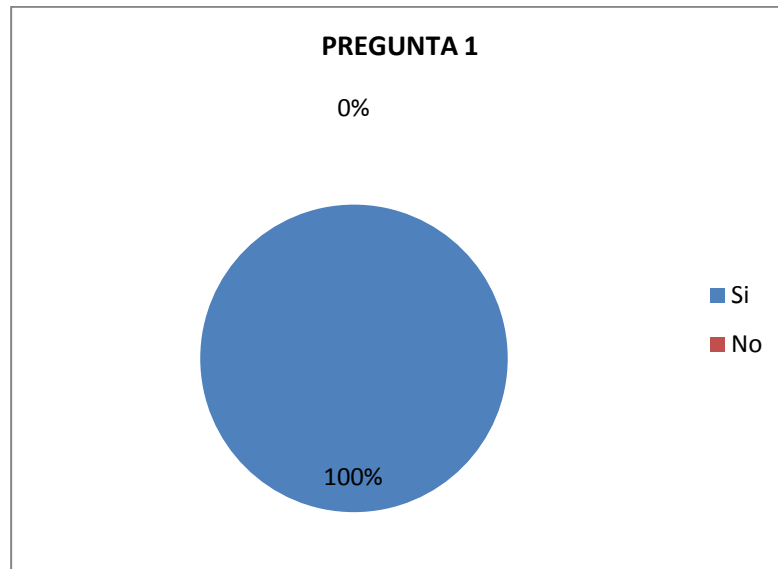


Figura 4.1: Grafico porcentual-Pregunta 1

El 100% de los encuestados, consideran importantes los sistemas de comunicaciones digitales.

2. Pregunta 2: Sabe como es el principio de funcionamiento de una comunicación digital:

- a) Si
- b) No

Tabla 4.2: Tabulación-Pregunta 2

Ítems	Nro. de Personas	Porcentaje
Si	67	97,10
No	2	2,90
TOTAL:	69	100

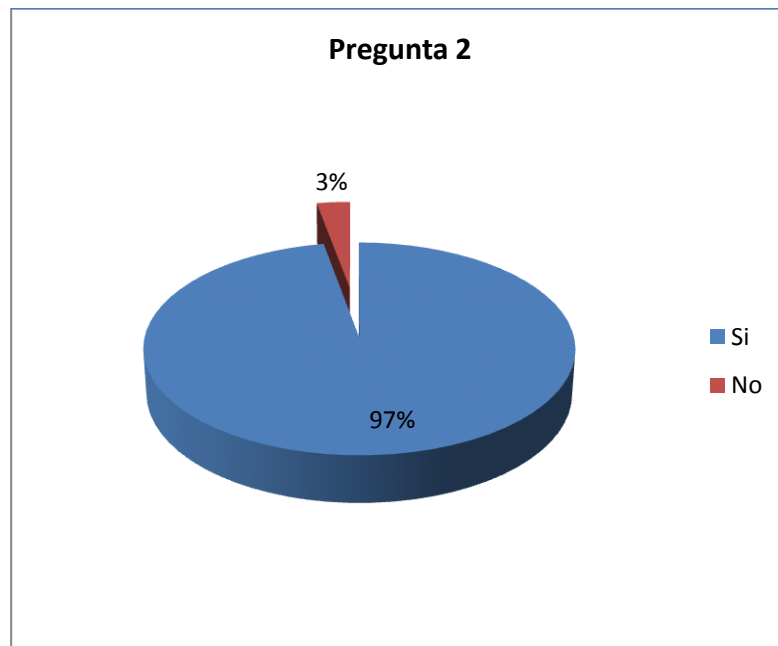


Figura 4.2: Grafico porcentual-Pregunta 2

En esta pregunta, el 97% de los estudiantes conocen el principio de funcionamiento de una comunicación digital, mientras que solo el 3% dice que no conocen.

3. Pregunta 3: Como le gustaría que fueran sus clases de comunicaciones digitales:

- a) Practicas
- b) Teóricas
- c) Ambos términos

Tabla 4.3: Tabulación-Pregunta 3

Ítems	Nro. de Personas	Porcentaje
Prácticas	12	17,39
Teóricas	0	0,0
Ambos términos	57	82,61
TOTAL:	69	100

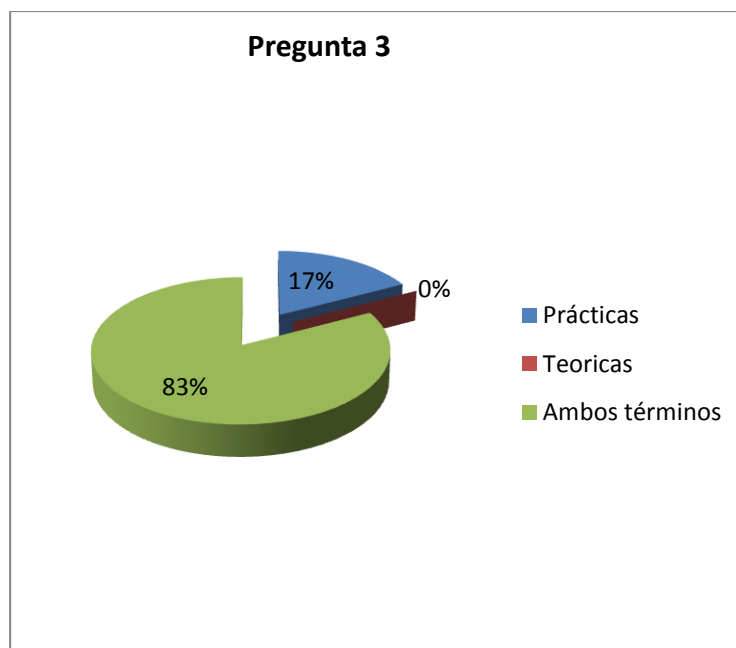


Figura 4.3: Grafico porcentual-Pregunta 3

El 83% de los encuestados, consideran que las clases de comunicaciones digitales deben ser teóricas-practicadas para un buen aprendizaje y el 17% prefieren que sus clases sean solo de manera práctica.

4. Pregunta 4: Conoce usted instrumentos para medir y visualizar los diferentes métodos de modulaciones digitales:

- a) Si
- b) No

Tabla 4.4: Tabulación-Pregunta 4

Ítems	Nro. de Personas	Porcentaje
Si	37	53,62
No	32	46,38
TOTAL:	69	100

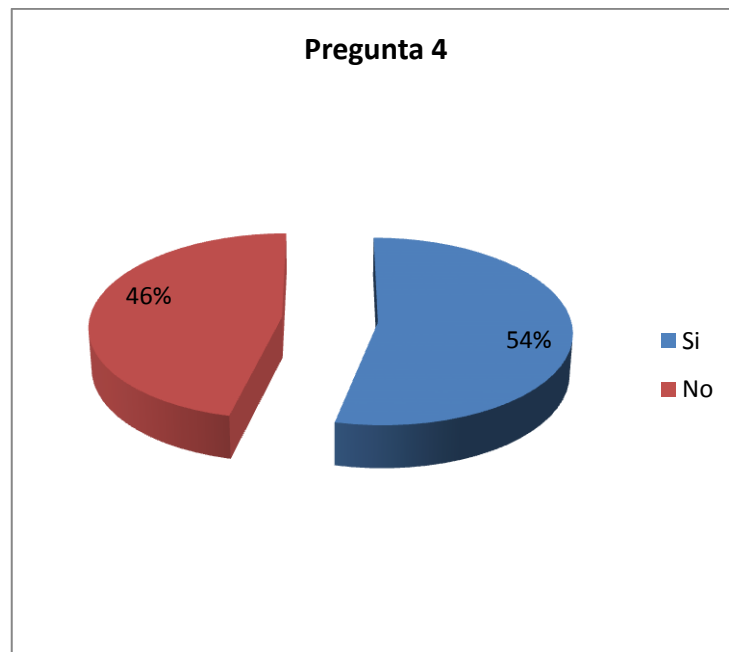


Figura 4 4: Grafico porcentual-Pregunta 4

El 54% de los encuestados, creen conocer instrumentos para medir y visualizar los diferentes métodos de modulaciones digitales, pero el 46% no conocen esos instrumentos.

5. Pregunta 5: Usted, tiene conocimientos de lo que es un laboratorio o instrumento virtual:

- a) Si
- b) No

Tabla 4.5: Tabulación-Pregunta 5

Ítems	Nro. de Personas	Porcentaje
Si	33	47,83
No	36	52,17
TOTAL:	69	100

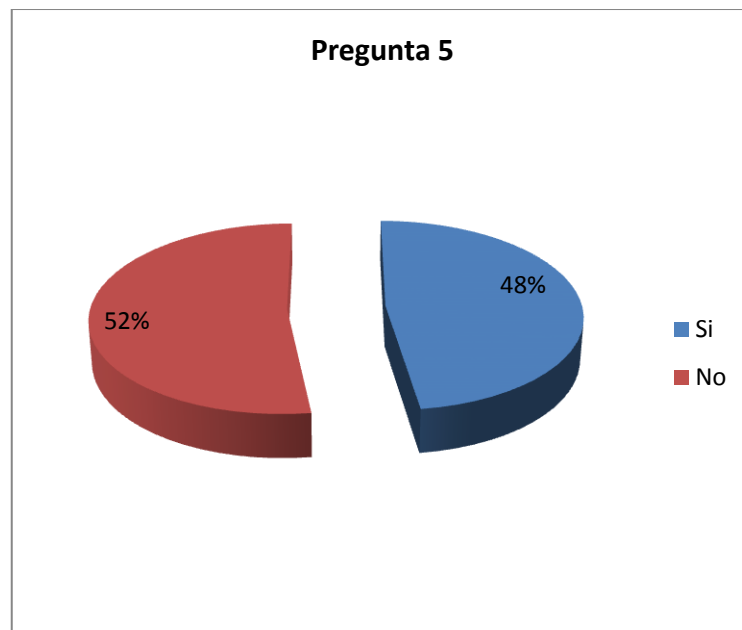


Figura 4.5: Grafico porcentual-Pregunta 5

El 52% de los encuestados, afirman no conocer lo que es un laboratorio o instrumento virtual, por otro lado el 48% si tienen conocimientos de lo que es un laboratorio virtual.

6. Pregunta 6: Esta de acuerdo, que las aplicaciones de un laboratorio virtual permiten mejorar la enseñanza-aprendizaje en el área de comunicaciones digitales:

- a) Si
- b) No

Tabla 4.6: Tabulación-Pregunta 6

Ítems	Nro. de Personas	Porcentaje
Si	67	97,10
No	2	2,90
TOTAL:	69	100

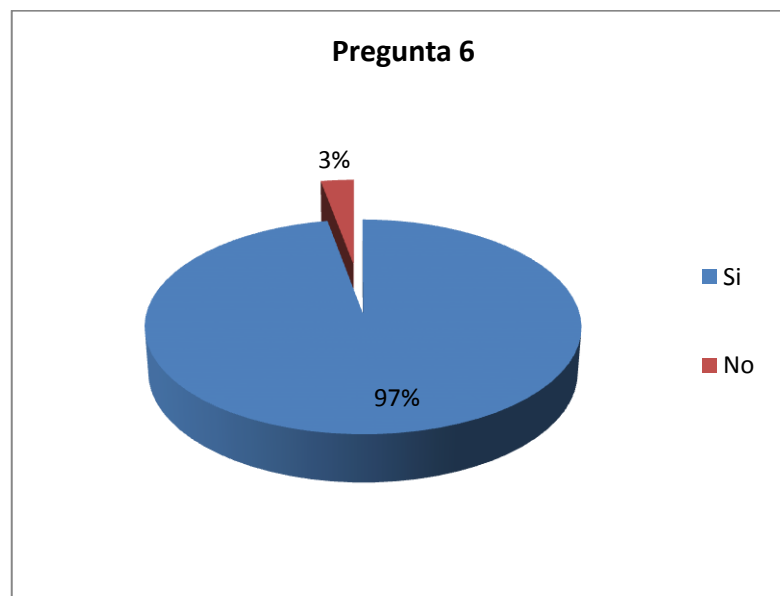


Figura 4.6: Grafico porcentual-Pregunta 6

El 97% de los encuestados, consideran que las aplicaciones de un laboratorio virtual permiten mejorar la enseñanza-aprendizaje en el área de comunicaciones digitales y el 3% dice que las aplicaciones de un laboratorio virtual no permiten mejorar la enseñanza-aprendizaje.

7. Pregunta 7: Considera importante que la FISEI, posea instrumentos con los que se pueda realizar prácticas de modulaciones y demodulaciones digitales:

- a) Si
- b) No

Tabla 4.7: Tabulación-Pregunta 7

Ítems	Nro. de Personas	Porcentaje
Si	66	95,65
No	3	4,35
TOTAL:	69	100

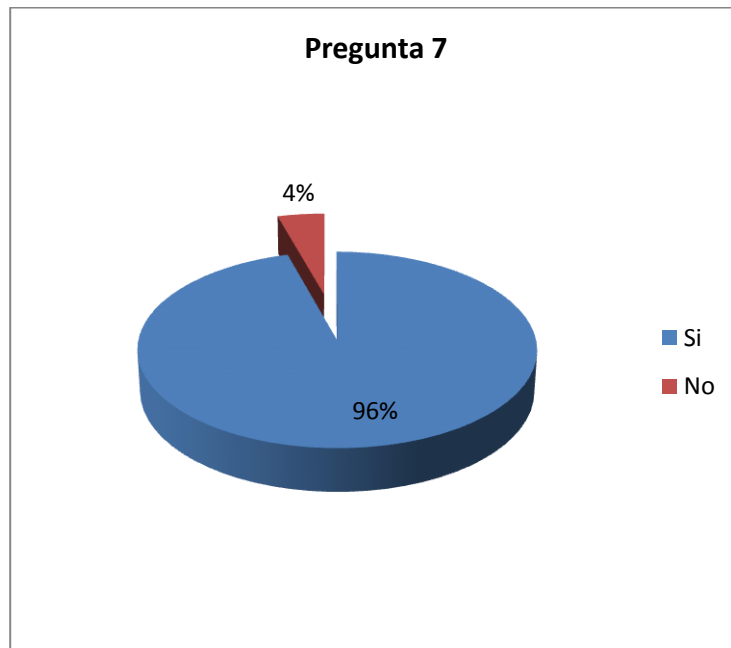


Figura 4.7: Grafico porcentual-Pregunta 7

En esta pregunta el 96% de los encuestados piensan que es importante que la FISEI, posea instrumentos con los que se pueda realizar prácticas de modulaciones y demodulaciones digitales, en cambio el 4% dicen que no es necesario tener instrumentos.

ENCUESTA A LOS DOCENTES DEL ÁREA DE ELECTRÓNICA Y COMUNICACIONES

1. Pregunta 1: Considera usted importante realizar las prácticas de modulaciones y demodulaciones digitales en los laboratorios de electrónica:

- a) Si
- b) No

Tabla 4.8: Tabulación-Pregunta 1-Encuesta 2

Ítems	Nro. de Personas	Porcentaje
Si	7	100
No	0	0
TOTAL:	7	100

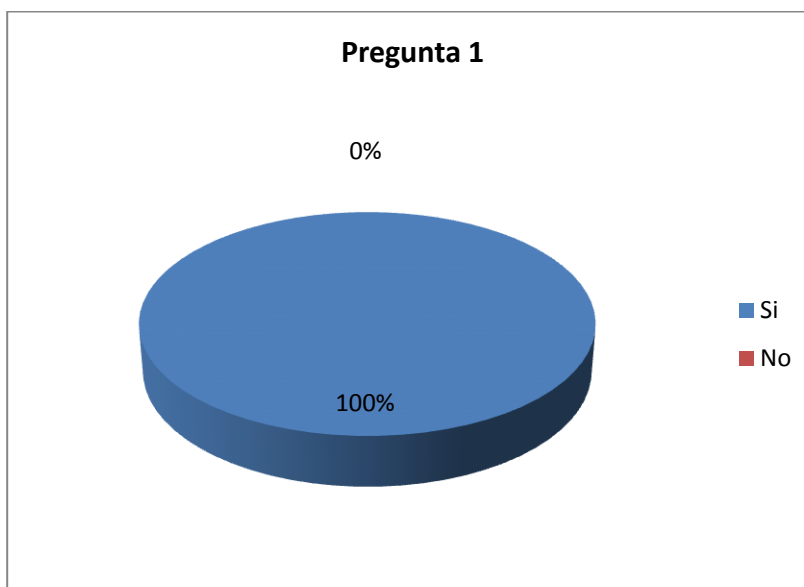


Figura 4 8: Grafico porcentual-Pregunta 1

Encuesta 2

El 100% de los docentes encuestados, consideran importante realizar, prácticas de modulaciones y demodulaciones digitales en los laboratorios de electrónica.

2. Pregunta 2: El Software, utilizado en los laboratorios de electrónica de la FISEI, cuenta con sus respectivas licencias y son óptimas para simular diferentes aplicaciones de comunicaciones digitales:

- a) Si
- b) No
- c) Ninguno

Tabla 4.9: Tabulación-Pregunta 2-Encuesta 2

Ítems	Nro. de Personas	Porcentaje
Si	4	57,14
No	3	42,86
Ninguno	0	0
TOTAL:	7	100,00

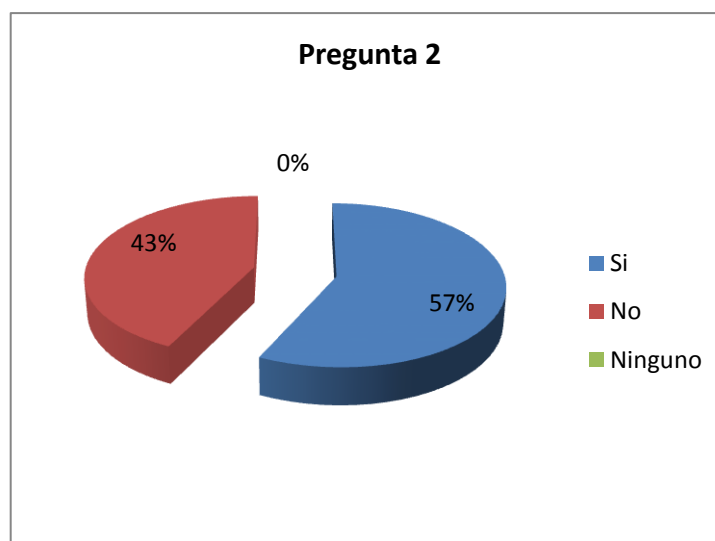


Figura 4.9: Gráfico porcentual-Pregunta 2

Encuesta 2

En esta pregunta el 57% de los encuestados, afirman que el Software, utilizado en los laboratorios de electrónica de la FISEI, si cuenta con sus respectivas licencias y son óptimas para simular diferentes aplicaciones de comunicaciones digitales, en cambio el 43% dice que no hay el software para este tipo de aplicaciones mencionadas anteriormente.

3. Pregunta 3: Cuenta los laboratorios de electrónica con equipos, capaces de generar, medir o visualizar las modulaciones y demodulaciones digitales para realizar practicas de calidad:

- a) Si
- b) No

Tabla 4.10: Tabulación-Pregunta 3-Encuesta 2

Ítems	Nro. de Personas	Porcentaje
Si	4	57,14
No	3	42,86
TOTAL:	7	100

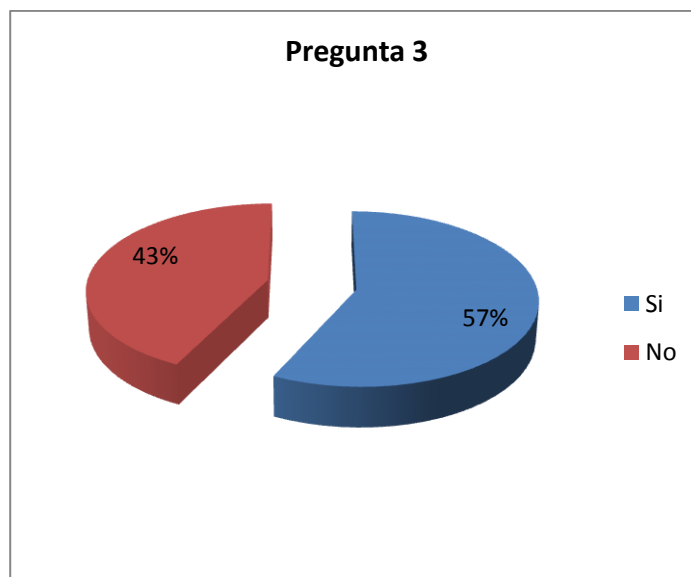


Figura 4 10: Grafico porcentual-Pregunta 3

Encuesta 2

El 57% de los encuestados, opinan que los laboratorios de electrónica si tienen equipos, capaces de generar, medir o visualizar las modulaciones y demodulaciones digitales para realizar prácticas, en cambio el 43% dice que no hay equipos para ciertos temas de interés.

4. Pregunta 4: Cree usted, que una manera de solucionar la carencia de prácticas por falta de equipos en los temas de modulaciones digitales, es haciendo uso de laboratorios virtuales:

- a) Si
- b) No
- c) Ninguno

Tabla 4.11: Tabulación-Pregunta 4-Encuesta 2

Ítems	Nro. de Personas	Porcentaje
Si	7	100,00
No	0	0,00
TOTAL:	7	100

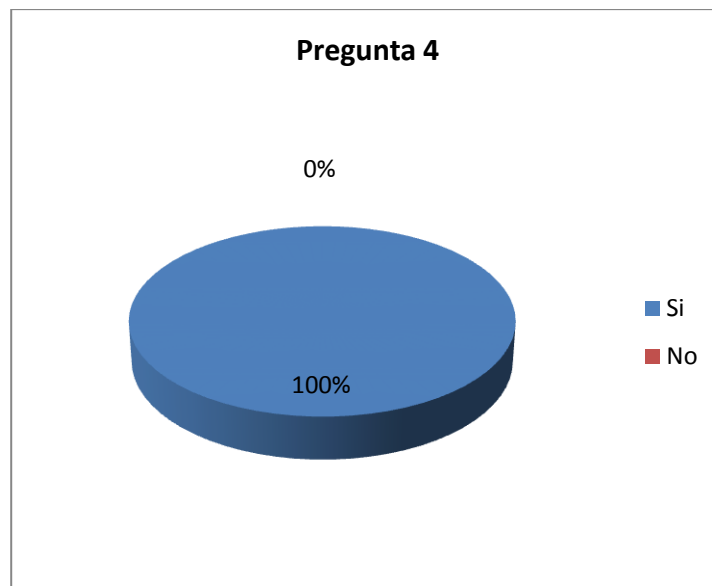


Figura 4.11: Gráfico porcentual-Pregunta 4
Encuesta 2

El 100% si cree, que una manera de solucionar la carencia de prácticas por falta de equipos en los temas de modulaciones digitales, es haciendo uso de laboratorios virtuales.

5. Pregunta 5: Cree usted, que variar de manera práctica los parámetros de una señal digital y verificando los resultados, son de gran interés para entender las modulaciones digitales:

- a) Si
- b) No

Tabla 4.12: Tabulación-Pregunta 5-Encuesta 2

Ítems	Nro. de Personas	Porcentaje
Si	7	100,00
No	0	0,00
TOTAL:	7	100

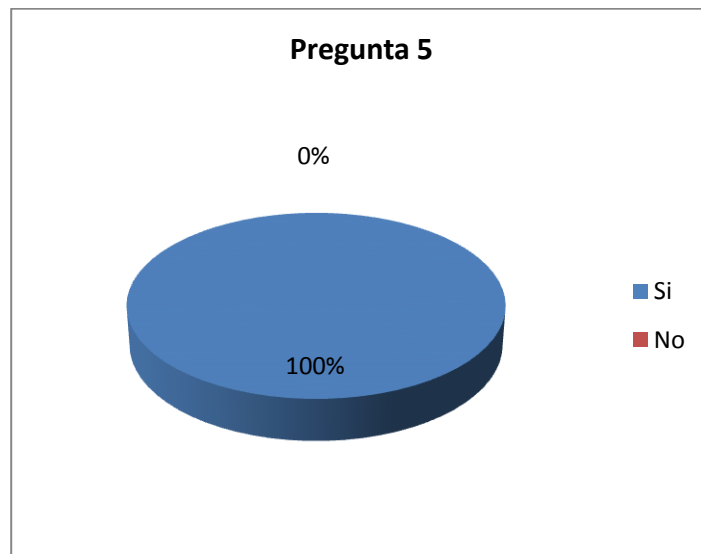


Figura 4 12: Grafico porcentual-Pregunta 5

Encuesta 2

En esta pregunta el 100% concuerda que variar de manera práctica los parámetros de una señal digital y verificando los resultados, son de gran interés para entender las modulaciones digitales.

6. Pregunta 6: Considera importante que la FISEI posea un laboratorio virtual, en la que se pueda realizar las modulaciones y demodulaciones digitales, con el fin de mejorar la enseñanza-aprendizaje:

- a) Si
- b) No

Tabla 4.13: Tabulación-Pregunta 6-Encuesta 2

Ítems	Nro. de Personas	Porcentaje
Si	6	85,71
No	1	14,29
TOTAL:	7	100

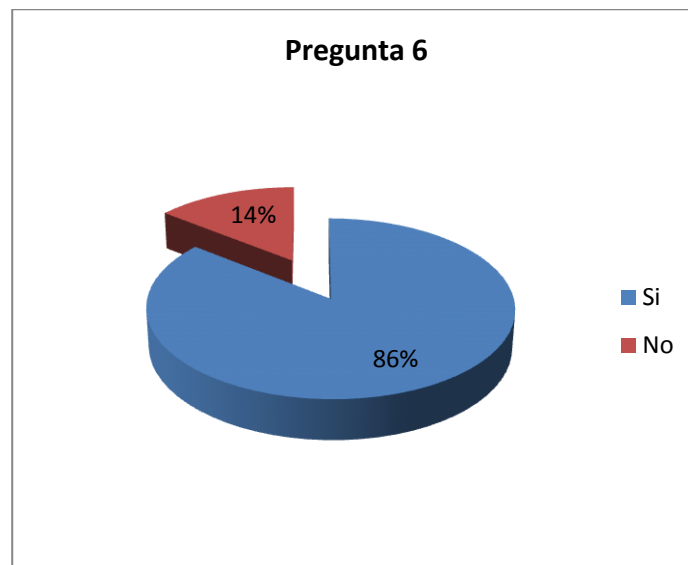


Figura 4.13: Grafico porcentual-Pregunta 6

Encuesta 2

El 86% de los docentes consideran beneficioso que la FISEI posea un laboratorio virtual, en la que se pueda realizar las modulaciones y demodulaciones digitales, con el fin de mejorar la enseñanza-aprendizaje, en cambio el 14% no esta de acuerdo en tener un laboratorio virtual.

FICHA DE OBSERVACIÓN AL LABORATORISTA DE ELECTRÓNICA Y COMUNICACIONES

Objetivo: Recolectar información sobre los equipos y los instrumentos que se tiene para las prácticas de laboratorio e investigación.

Tabla 4. 14: Tabulación-Ficha Observación

PARÁMETROS	SI	NO
Se cuenta con las suficientes herramientas de software y hardware para las prácticas de modulaciones digitales en los laboratorios de la FISEI.		X
Los programas que posee la FISEI, son actualizados para el desarrollo de la investigación.		X
Existen equipos con interfaces, que permitan interactuar con programas como labview, etc.		X
Es necesario tener laboratorios para diferentes aplicaciones en el área de comunicaciones digitales.	X	
Los instrumentos de medición en los laboratorios, satisfacen las necesidades de los docentes.		X
Los equipos de los laboratorios de electrónica, en la FISEI, cumplen con los requerimientos de los estudiantes, a la hora de realizar prácticas de comunicaciones digitales.	X	
En el laboratorio, las prácticas de modulación y demodulación digital son difíciles de ejecutar por la falta de equipos.	X	
Sería una gran contribución si se implementara un laboratorio virtual, para el análisis de las modulaciones digitales en los laboratorios de electrónica de la FISEI.	X	
TOTAL	4	4

De acuerdo, a la tabulación de la ficha de observación se puede concluir que los laboratorios de electrónica, si tiene carencias de equipos o instrumentos para la realización de practicas referentes a comunicaciones digitales.

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Al recoger los resultados de las encuestas y de la ficha de observación, se logró establecer las siguientes conclusiones:

- La FISEI, no posee laboratorios para ciertas prácticas de comunicaciones digitales como: modulaciones-demodulaciones digitales, codificación de línea, etc. Por consecuencia se tiene una enseñanza-aprendizaje de manera teórica.
- Los módulos que se tienen en los laboratorios de electrónica, no satisfacen los requerimientos de los estudiantes de la carrera de electrónica y comunicaciones, ocasionando prácticas deficientes de modulaciones digitales.
- Los docentes del área de electrónica y comunicaciones de la FISEI, concuerdan en que es una herramienta de gran apoyo la implementación de un laboratorio virtual de modulaciones digitales, debido al poco costo de inversión que se requiere y al ambiente didáctico que ofrece este tipo de laboratorio.

5.2 Recomendaciones

A continuación, se describen las recomendaciones más relevantes luego de los análisis de las encuestas y de la ficha de observación como son:

- Se recomienda que la FISEI, implemente un laboratorio virtual para la realización de prácticas de modulación y demodulación digital, junto con la simulación de los códigos de línea. El objetivo es de mejorar la enseñanza-aprendizaje de una manera práctica.
- La FISEI, al tener varios software de programación con sus respectivas licencias y actualizaciones, debería explotar al máximo sus aplicaciones para crear entornos sencillos de modulaciones digitales. De esta manera, los estudiantes ya no tendrán inconvenientes a la hora de realizar sus prácticas en los laboratorios.
- Implementar un simulador de modulaciones y demodulaciones digitales, que ayude a potenciar a la formación académica en la FISEI, de una manera didáctica.

CAPITULO VI

6. PROPUESTA

6.1 Datos Informativos

- **Titulo**

“Laboratorio virtual a través de la plataforma labview para la realización de prácticas de modulación y demodulación digital en la FISEI de la Universidad Técnica de Ambato”.

- **Institución ejecutora**

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

- **Beneficiarios**

- ✓ Docentes de la carrera de electrónica y comunicaciones
- ✓ Estudiantes de electrónica y comunicaciones

- **Ubicación**

Provincia: Tungurahua

Cantón: Ambato

Parroquia: Huachi Chico

Lugar: Universidad Técnica de Ambato-Campus Huachi- Av. Chasquis y Rio Payamino-Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

- **Equipo técnico**

Investigador: Wilson Llumán

Tutor: Ing. Luis Pomaquero

6.2 Antecedentes de la propuesta

Los nuevos retos académicos, obligan a las universidades ejecutar nuevos cambios en los métodos pedagógicos de los docentes, con la finalidad de que los estudiantes puedan asimilar los conocimientos impartidos en las aulas, de una mejor manera y proporcionar el desarrollo de la investigación científica.

La FISEI, no cuenta con los suficientes módulos en los laboratorios de electrónica para realizar ciertas prácticas de modulación-demodulación digital, así, como también los códigos de línea. Por ende el aprendizaje teórico-practico provocan: la investigación limitada en los estudiantes de la carrera de electrónica. Estos factores mencionados, traerán consecuencias negativas a los futuros profesionales; en el análisis de las modulaciones digitales.

Para estos inconvenientes la FISEI tendrá que optar por el uso del software Labview con su respectiva licencia, dicho software se tiene en la Facultad por lo que permite facilitar la implementación del laboratorio virtual de modulación-demodulación digital. También, se cuenta con los docentes familiarizados al software labview, permitiendo que el uso de un laboratorio virtual sea sencillo de manipular.

Con la ayuda de la interfaz amigable de labview, se tiene un ambiente didáctico en donde se puede realizar una determinada práctica, las veces que se quiera sin temor a equivocarse. De manera contraria, sucede con los equipos reales de laboratorio en la que un error de uso, puede dañar totalmente el equipo causando grandes pérdidas económicas y el miedo a ser utilizados por parte del estudiante.

6.3 Justificación

El presente proyecto, es de gran importancia para la FISEI, en especial para los docentes, estudiantes de la carrera de electrónica y comunicaciones, en disponer de un laboratorio virtual de modulación-demodulación digital junto con la simulación de la codificación digital. Este laboratorio, surge de la necesidad que tiene el estudiante dentro del aula de clases, que es tener prácticas luego de haber recibido

sus clases de manera teórica. Por consiguiente, los estudiantes podrán asimilar mejor los conocimientos impartidos por los docentes.

Además, los docentes que dicten sus clases relacionado a modulaciones digitales, tendrán en sus manos una herramienta de apoyo como es el laboratorio virtual, capaz de facilitar el aprendizaje y atraer, el interés del estudiante. El requisito indispensable para el uso de este laboratorio, no es más que tener conocimientos de la materia, en este caso de comunicaciones digitales y no hace falta la supervisión de alguna persona especializada o la capacitación del estudiante, sino únicamente una serie de recomendaciones antes de su uso.

En este documento se establecen los parámetros importantes a tener en cuenta para hacer la implementación del proyecto propuesto.

6.4 Objetivos

6.4.1 Objetivo General

- Implementar un laboratorio virtual a través de la plataforma labview para la realización de prácticas de modulación y demodulación digital en la FISEI de la Universidad Técnica de Ambato.

5.4.2 Objetivo Especifico

- Analizar la plataforma labview que se posee en la FISEI, para la implementación de un laboratorio virtual.
- Definir los controles e indicadores para que el usuario manipule los parámetros de las modulaciones-demodulaciones digitales.
- Implementar un laboratorio virtual que permita satisfacer las necesidades de los estudiantes de la carrera de electrónica y comunicaciones en la FISEI.

6.5 Análisis de factibilidad

La posibilidad de la implementación y puesta en marcha de la propuesta se fundamenta en los siguientes aspectos.

5.5.1 Factibilidad técnica

El proyecto es factible desde el punto de vista técnico, ya que para la implementación del laboratorio virtual de modulación-demodulación digital se cuenta con la herramienta *toolkit modulation* de labview.

5.5.2 Factibilidad operativa

La propuesta de implementar un laboratorio virtual de modulación-demodulación digital es factible desde el punto de vista operativo, debido a que se cuenta en la FISEI, con los docentes y estudiantes preparados para poner en marcha un laboratorio virtual.

5.5.3 Factibilidad económica

Los gastos especificados para la realización del presente proyecto se encontraban dentro del presupuesto proyectado por lo tanto fue factible la ejecución del mismo en cuanto al factor económico se refiere.

6.6 Fundamentación

6.6.1 Introducción a labview

El software Labview, es un lenguaje de programación de alto nivel, de tipo grafico, y enfocado al uso de instrumentación. Posee todas las estructuras, para elaborar un algoritmo en cualquier aplicación, como en análisis, telemática, manejo de textos y comunicaciones digitales, etc.

Cada uno de los programas desarrollados en LabView, es conocido como instrumento virtual (VI), ocupando un espacio de memoria del computador y utilizados para el control de instrumentos. De esta manera, se puede obtener soluciones fáciles y eficientes a problemas comunes en las áreas industriales,

investigativas y académicas para realizar virtualmente el trabajo de instrumentos reales.

En labview, se puede implementar varios instrumentos virtuales sin tener limitaciones y además se puede crear sub-instrumentos virtuales (SubVI) que dependan de un VI Principal. La figura 6.1 muestra las interfaces de labview que constan de dos ventanas: Panel Frontal y Diagrama de Bloques. Éstas cuentan con paletas que contienen los objetos necesarios para implementar y desarrollar tareas.

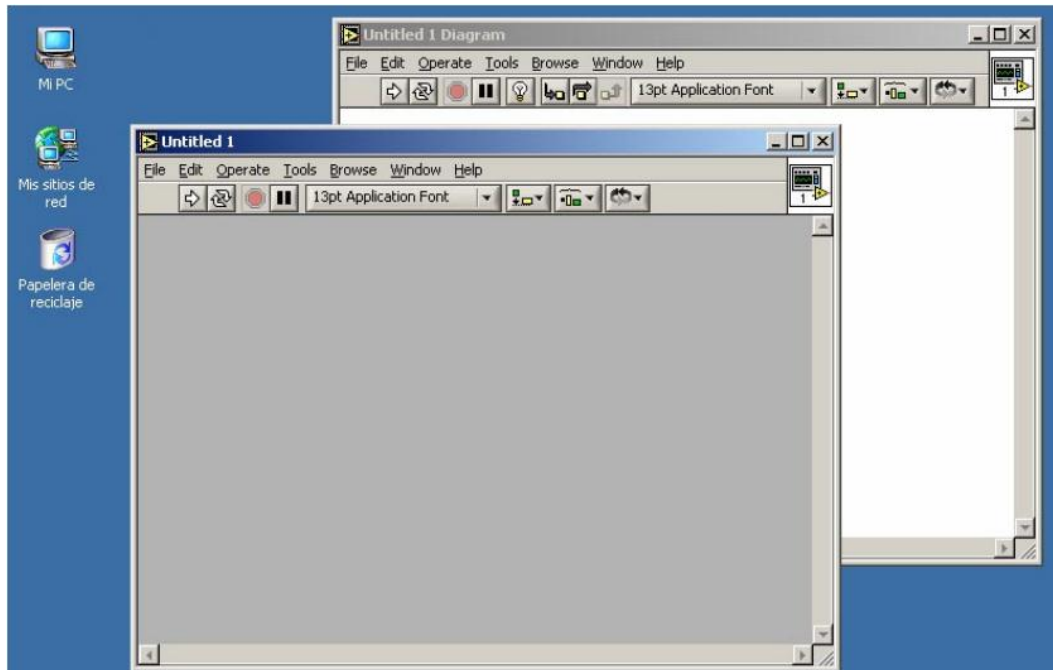


Figura 6.1: Interfaces de un VI

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

6.6.2 Panel Frontal

El panel frontal, es la interface gráfica que simula el panel de un instrumento real, permite la entrada y salida de datos, puede contener pulsadores, perillas, botones, gráficos y en general controles e indicadores.

Controles.- Los controles son objetos que sirven para ingresar datos al programa y pueden ser manipulados por el usuario. Los controles son variables de entrada.

Indicadores.- Los indicadores sirven para visualizar los resultados entregados por el programa y no pueden ser manipulados por el usuario. Los indicadores son variables de salida.

La figura 6.2 muestra las partes de un panel frontal, controles e indicadores mencionados anteriormente.

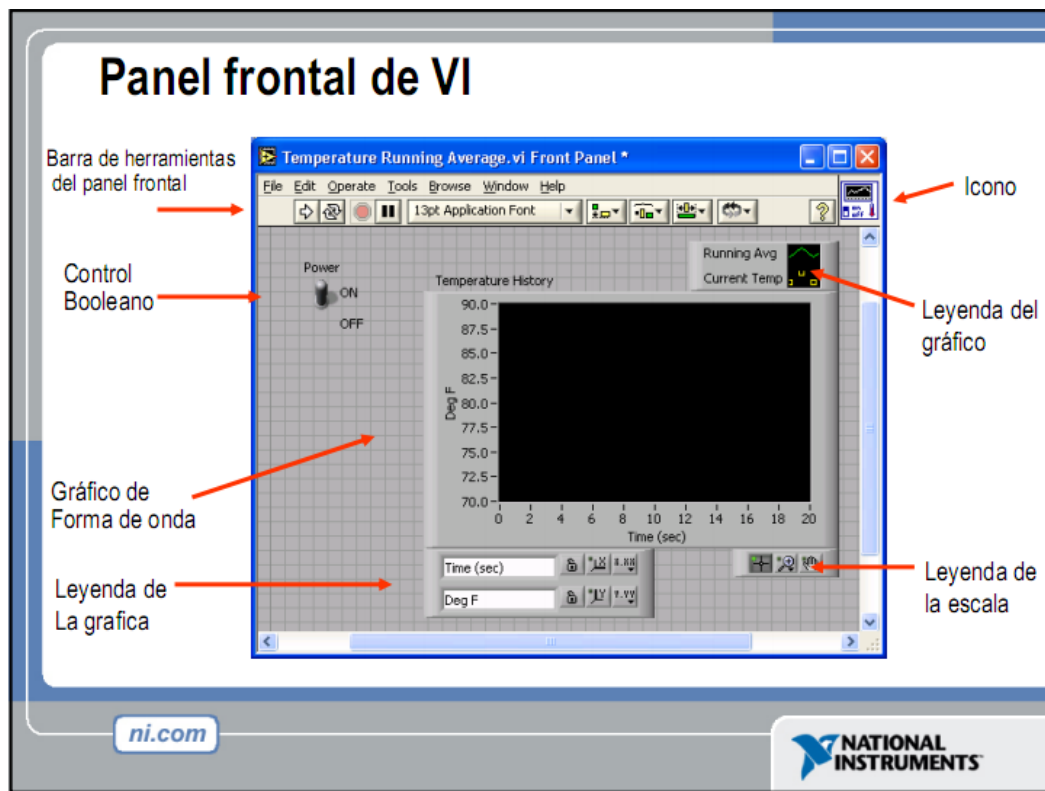


Figura 6.2: Panel frontal de VI

Fuente: [http://laimbio08.escet.urjc.es/docencia/IE/Introduccion%20 LabVIEW%20-%20Seis%20Horas.pdf](http://laimbio08.escet.urjc.es/docencia/IE/Introduccion%20LabVIEW%20-%20Seis%20Horas.pdf)

6.6.3 Diagrama de Bloques

El diagrama de bloques contiene el código fuente gráfico del VI, posee funciones y estructuras que relacionan las entradas con las salidas, creadas en el panel frontal.

En un diagrama de bloques se distinguen:

- **Terminales**, que representan los controles e indicadores del panel.
- **Funciones y subVIs**, que realizan tareas específicas.
- **Estructuras y Conexiones**, que determinan el flujo de datos en el programa. En general, cualquiera de estas partes del diagrama de un VI se denomina **NODO**.

El diagrama de bloques de la figura 6.3, muestran los controles e indicadores correspondientes al panel frontal de la figura 6.2.

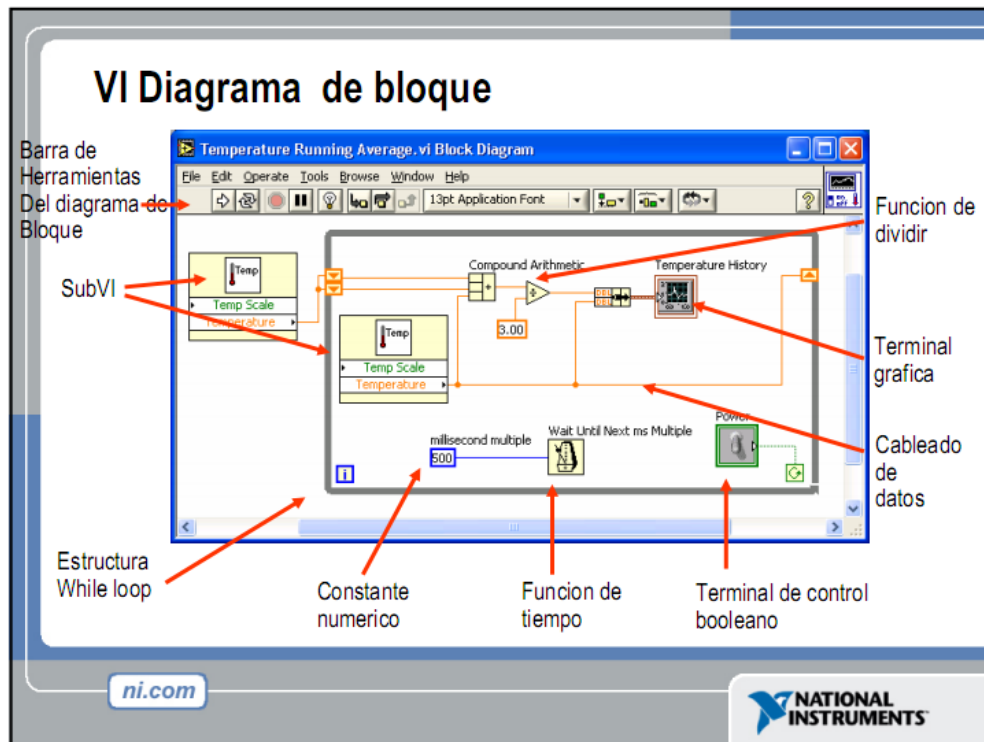


Figura 6.3: Diagrama de bloque de VI

Fuente: [http://laimbio08.escet.urjc.es/docencia/IE/Introduccion%20 LabVIEW%20-%20Seis%20Horas.pdf](http://laimbio08.escet.urjc.es/docencia/IE/Introduccion%20LabVIEW%20-%20Seis%20Horas.pdf)

6.6.4 Paleta de herramientas

La paleta de herramientas muestra una paleta que contiene las herramientas necesarias para editar y depurar los objetos tanto del panel frontal como del diagrama de bloques como se visualiza en la figura 6.4.



Figura 6.4: Paleta de herramientas

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

6.6.5 Paleta de controles

La paleta de control, se utiliza únicamente en el panel frontal y contiene los objetos necesarios para crear una interface de entrada y salida de datos (controles e indicadores). Esta barra se obtiene de la barra de menús con la opción de **Window>> Show Controls Palette** o dando clic derecho sobre el panel frontal. La apariencia de la paleta de controles se muestra en la figura 6.5.

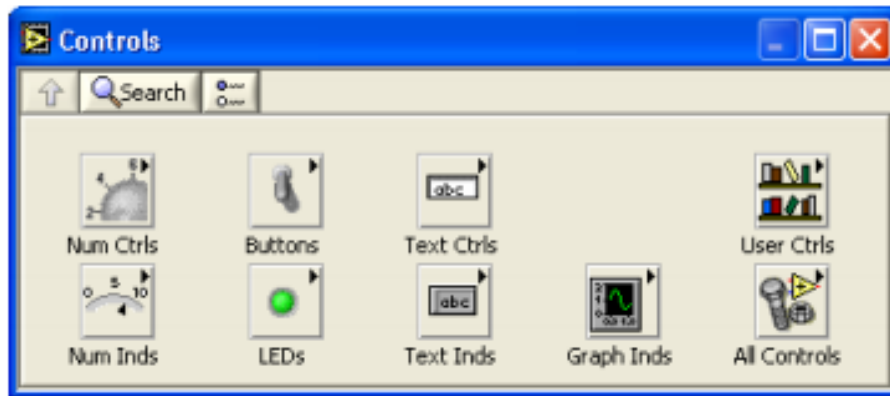


Figura 6.5: Paleta de Controles

Fuente: Por el investigador

6.6.6 Paleta de funciones

La paleta de funciones se usa únicamente en el diagrama de bloques y contiene todos los objetos para crear y editar el código fuente.

En la figura 6.6, se tiene la paleta de funciones que se obtiene de las barras de menús con la opción **Window>> Show Functions Palette**, o haciendo clic derecho en el diagrama de bloques.

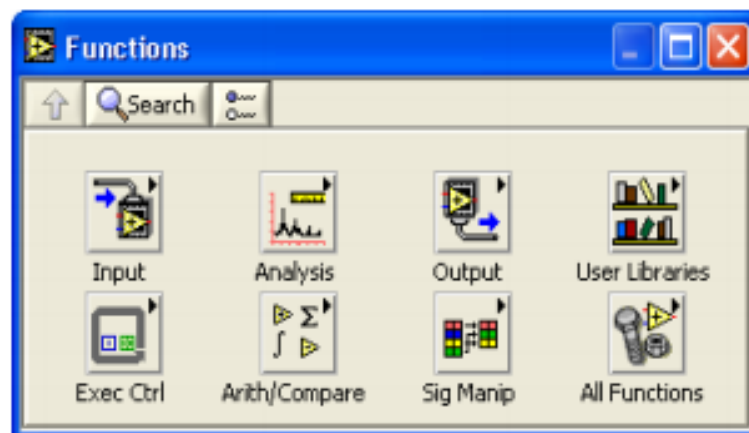


Figura 6.6: Paleta de funciones

Fuente: Por el investigador

La figura 6.7 muestra los submenús de la paleta de funciones que más se utiliza al momento de programar una aplicación.

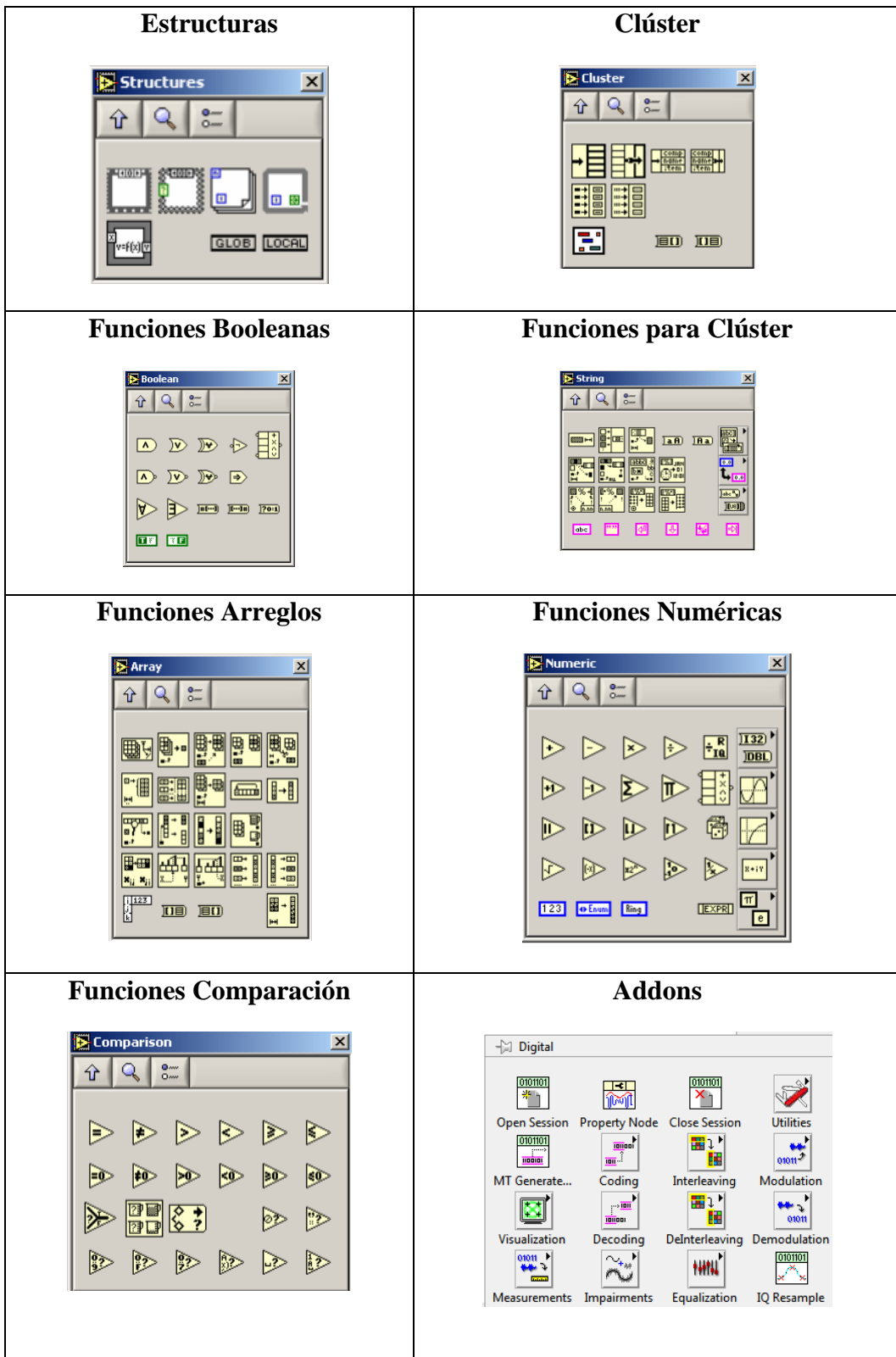


Figura 6.7: Submenús de la paleta de funciones
Fuente: Por el investigador

6.6.7 Barras de Labview

Barras de Menús de labview

La barra de menús de labview ubicado en la parte superior de un VI, contiene diversos menús desplegables que cumplen diferentes funciones como se muestra en la figura 6.8.



Figura 6.8: Barras de Menú

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

Descripción de los elementos de la barra de menú:

- **File:** Sus opciones se usan para abrir, cerrar, guardar imprimir Vis, etc.
- **Edit:** Se usa principalmente para organizar el panel frontal y el diagrama de bloques y establecer nuestras preferencias.
- **Operate:** Sus comandos sirven para ejecutar el Vi.
- **Tools:** Contiene herramientas para la configuración de labview, de los proyectos o SubVI.
- **Browse:** Contiene opciones que permiten observar aspectos del VI activo y toda su jerarquía.
- **Window:** Permite configurar la apariencia de las paletas y ventanas.
- **Help:** Presenta la ayuda en línea, los manuales de referencia, enlaces de internet y los archivos de ayuda de cada toolkit instalado.

Barra de herramientas del panel frontal de labview

La barra de herramientas permite la ejecución, depuración y organización de los VIs. Esta localizada debajo de la barra de menús y se muestra en la figura 6.9.

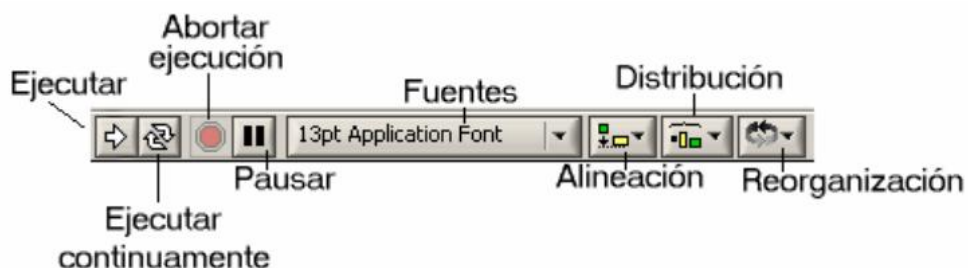


Figura 6.9: Barra de herramientas del panel frontal

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

A continuación se describe rápidamente los menús de la barra de herramientas:

- **Ejecutar:** Botón para correr la aplicación, su forma varía de acuerdo al nivel jerárquico donde se ejecute.
- **Ejecutar continuamente:** Botón para ejecutar la aplicación repetidamente hasta que se presione de nuevo o se presione abortar.
- **Abortar ejecución:** Aborta la ejecución de una aplicación.
- **Pausar:** Botón para pausar la aplicación.
- **Fuentes:** Menú para seleccionar tipos de letra en objetos y textos.
- **Alineación:** Permite alinear los objetos tanto en el panel frontal como en el diagrama.
- **Distribución:** Permite distribuir uniformemente objetos tanto en el panel frontal como en el diagrama de bloques.
- **Reorganización:** Permite reorganizar la posición de los objetos tanto en el panel frontal como en el diagrama.

Barra de herramientas del diagrama de bloques de labview

En la figura 6.10 muestra la barra de herramientas del diagrama de bloques que contiene además de los mismos botones de la barra del panel frontal, las herramientas de depuración.

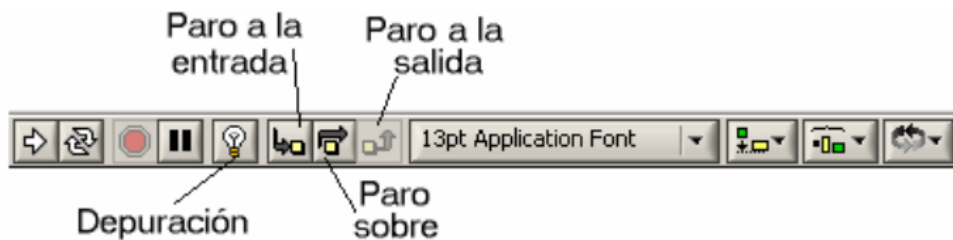


Figura 6.10: Barra de herramientas del diagrama de bloques
Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

Se presenta la descripción de los botones de la barra del panel frontal:

- **Depuración:** Botón para observar el flujo de datos en el diagrama de bloques.
- **Paro a la entrada:** En modo depuración sirve para generar un paro a la entrada de un nodo.
- **Paro sobre:** En modo depuración sirve para saltar un nodo.
- **Paro a la salida:** En modo depuración sirve para salir de un nodo.

6.6.8 Estructuras de labview

Las estructuras de labview, son funciones que permiten controlar ciertos algoritmos y su operación es similar a cualquier lenguaje de programación. A continuación se describen algunas estructuras básicas que son:

Case Structures

Case Structure es una estructura de comparación y ejecución condicionada, que de acuerdo al parámetro ingresado al selector, permite ejecutar un determinado subdiagrama. La figura 6.11 muestra la descripción de esta función, la cual, en la parte superior del subdiagrama aparece el identificador del que se está representando en pantalla y en ambos lados de este identificador aparecen unas flechas que permiten pasar de un subdiagrama a otro.

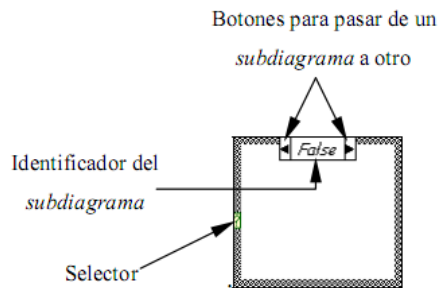


Figura 6.11: Estructura Case Structure

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

Sequence Structures

Sequence structures, es una función que permite ejecutar un conjunto de operaciones de manera simultanea y en el orden que el programador desee. La figura 6.12, describe las partes de esta función.

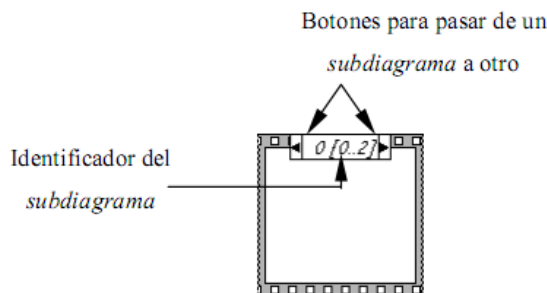


Figura 6.12: Sequence Structures

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprende/Curso%20LabVIEW6i.pdf>

For Loop

For Loop, es el equivalente al bucle For en los lenguajes de programación convencionales. Ejecuta el código dispuesto en su interior un número determinado de veces, su forma se muestra en la figura 6.13.

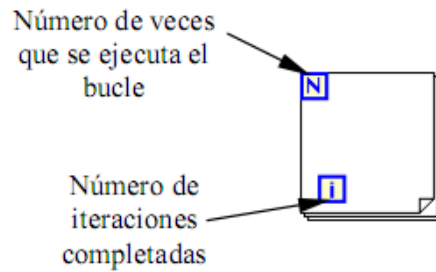


Figura 6.13: For Loop

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

Cuando se necesita pasar valores de una iteración a otra se emplean los llamados *shift registers*. Para crear uno, se pulsará el botón derecho del ratón mientras éste se halla situado sobre el borde del bucle, seleccionando la opción *Add Shift Register*. En la figura 6.14, se muestra los valores que se pueden obtener con el *shift register* en un For Loop.

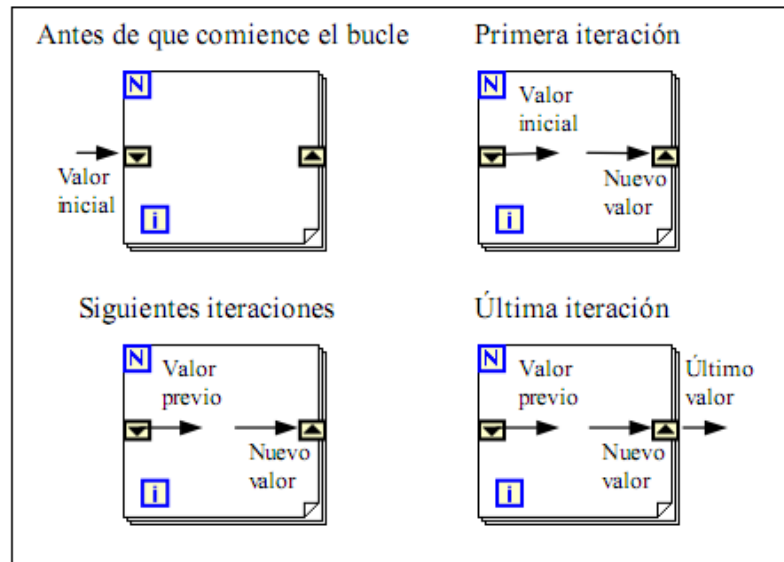


Figura 6.14: Shift Register en For Loop

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

El *shift register* consta de dos terminales, situados en los bordes laterales del bloque: el terminal izquierdo almacena el valor obtenido en la iteración anterior y el terminal derecho guardará el dato correspondiente a la iteración en ejecución.

Además, se puede configurar un *shift register*, para memorizar valores de varias iteraciones previas como se muestra en la figura 6.15. Para ello, con el ratón situado sobre el terminal izquierdo del *shift register* se pulsará el botón derecho, seleccionando a continuación la opción *Add Element*.

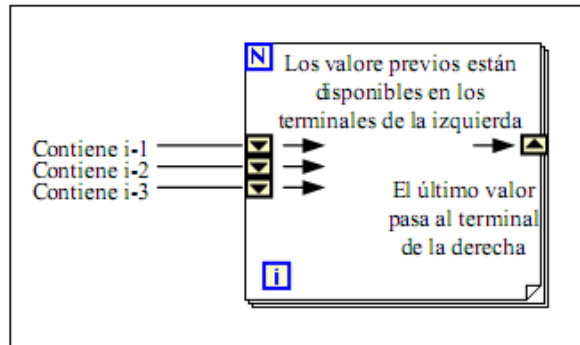


Figura 6.15: For Loop con Add Element

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

While Loop

While loop es empleado para hacer que una secuencia de instrucciones se repitan una cantidad de veces, siempre y cuando una afirmación sea verdadera. Su funcionamiento es similar al del bucle For, su estructura se muestra en la figura 6.16. Donde el número de iteraciones es un contador que se incrementa una unidad cada vez que se repite el ciclo y el terminal condicional puede recibir un valor de true para repetir el ciclo y un valor de false, que detiene el ciclo.

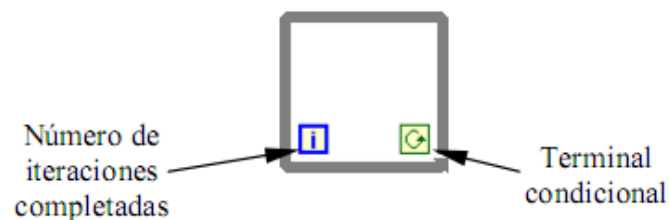


Figura 6.16: While Loop

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

Con esta estructura también se pueden emplear los *shift registers* para tener disponibles los datos obtenidos en iteraciones anteriores (es decir, para memorizar valores obtenidos). Su empleo es análogo al de los bucles for, por lo que omite su explicación.

6.6.9 Tipos de conexiones

En el lenguaje G, es posible identificar los tipos de datos por la forma, tamaño y color de sus terminales y conexiones. Entre controles e indicadores se pueden manejar datos escalares y arreglos de una o mas dimensiones.

Las características de las conexiones son diferentes para cada tipo de dato, esto se muestra en la figura 6.17.

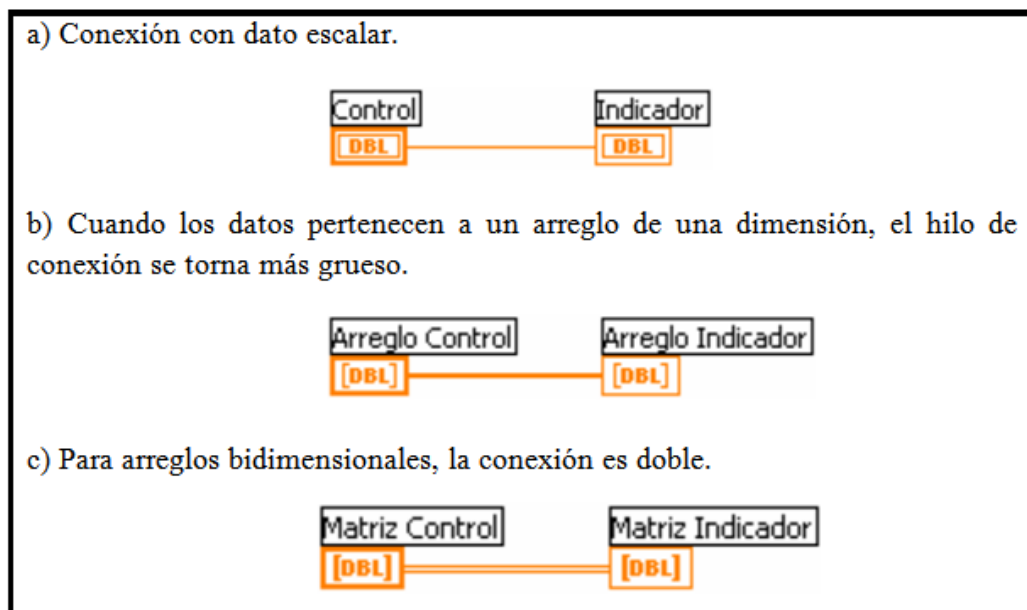



Figura 6.17: Características de la conexión

Fuente: Por el investigador

Cada control e indicador numérico encierra el tipo de dato que tiene asignado. Por ejemplo DBL significa que es de precisión doble. El color de la conexión es indicativo del tipo de dato. Por ejemplo Naranja es para números de punto flotante, Azul para enteros, Verde para booleanos, Fucsia para cadenas, etc.

6.6.10 Tipos de datos numéricos

La figura 6.18 muestra los datos numéricos con sus diferentes tipos, colores, numero de bits y el tamaño que ocupan en memoria. Estas características de los datos son muy importantes al realizar una determinada programación porque evita la generación de errores al conectar funciones que no son del mismo tipo de dato numérico.



	TIPO	COLOR	BITS	RANGO
EXP	REAL-Extendido	Naranja	128	$\pm 1.19e+4932$
DBL	REAL-Doble	Naranja	64	$\pm 1.79e+308$
SGL	REAL-Simple	Naranja	32	$\pm 3.4e+38$
I32	Entero-LONG	Azul	32	± 2147483647
I16	Entero-WORD	Azul	16	$-32768..+32.67$
I8	Entero-BYTE	Azul	8	$-128..+127$
U32	NATURAL	Azul	32	$0..4294'967.295$
U16	NATURAL	Azul	16	$0..65.535$
U8	NATURAL	Azul	8	$0..255$
CXT	COMPLEJO	Naranja	$2*128$	$2*\pm 1.19e+4932$
CBD	COMPLEJO	Naranja	$2*64$	$2*\pm 1.79e+308$
CSG	COMPLEJO	Naranja	$2*32$	$2*\pm 3.4e+38$

Figura 6.18: Representaciones Numéricas

Fuente: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>

6.6.11 Programación en Labview

Para empezar a programar, se debe ejecutar labview en la cual aparecerá la ventana de inicialización como en la figura 6.19.



Figura 6.19: Inicialización de labview

Fuente: Por el investigador

Seleccionamos la opción *Blank VI*, que se indica con la flecha en la figura 6.20 para crear un nuevo VI (instrumento virtual).



Figura 6.20: Selección de Blank VI

Fuente: Por el investigador

Luego aparecerán dos ventanas que son: el panel frontal (front panel) y el diagrama de bloques (block diagram), que se observa en la figura 6.21.

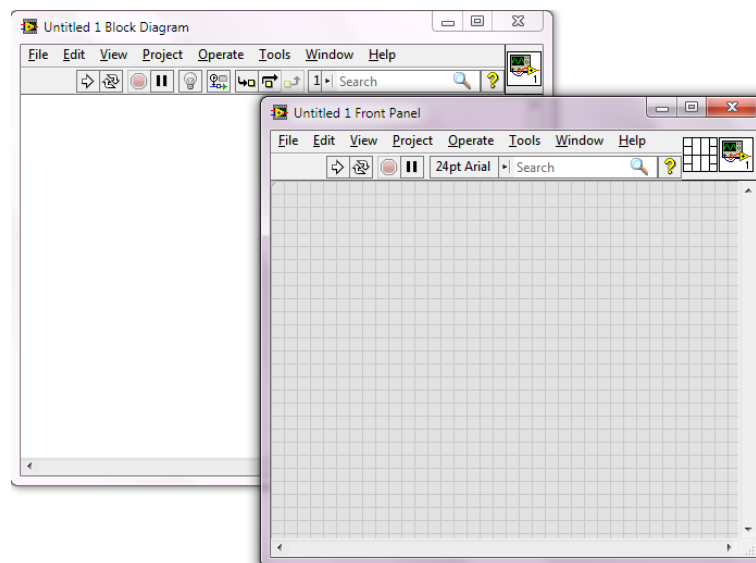


Figura 6.21: Interfaces de un VI

Fuente: Por el investigador

Como un ejemplo sencillo explicaremos la programación para realizar la suma de tres valores (A, B, C) y su resultado. Los pasos a seguir son:

1. En primer lugar se debe ubicar en el panel frontal para definir y seleccionar de la paleta de controles, todos los controles (entradas que dará el usuario) e indicadores (salidas que presentará en pantalla e VI) que se emplearán para introducir los datos por parte del usuario y presentar en pantalla los resultados. La figura 6.22 muestra el control *Numeric Constant*, utilizado para ingresar los valores a sumar y el *Numeric Indicator* para visualizar el resultado de esta operación.

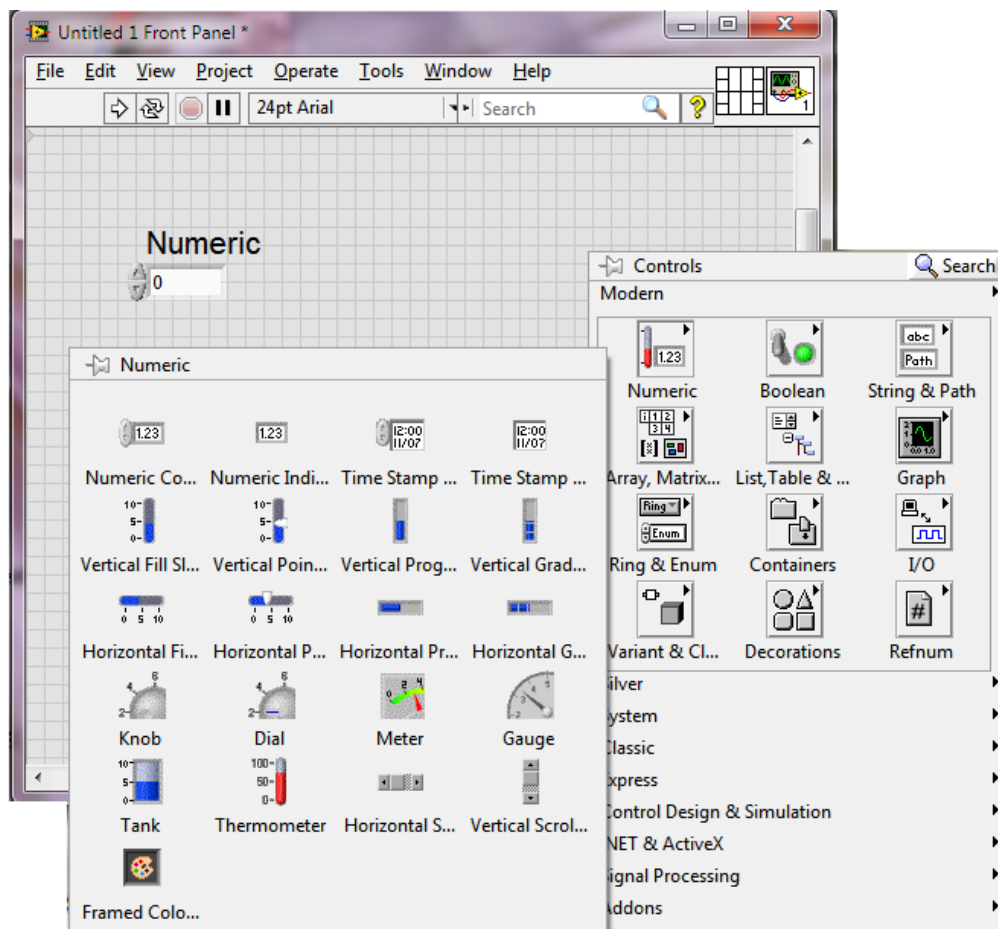


Figura 6.22: Creación de los controles

Fuente: Por el investigador

2. Una vez colocados en la ventana correspondiente al panel frontal todos los objetos necesarios, debe pasarse a la ventana Diagram (menú Windows > Show Diagram), que es donde se realiza la programación propiamente dicha (diagrama de bloques). Al abrir esta ventana, en ella se encuentran los terminales correspondientes a los objetos situados en el panel frontal,

dispuestos automáticamente por LabVIEW. La figura 6.23 muestra el diagrama con los objetos correspondiente al panel frontal.

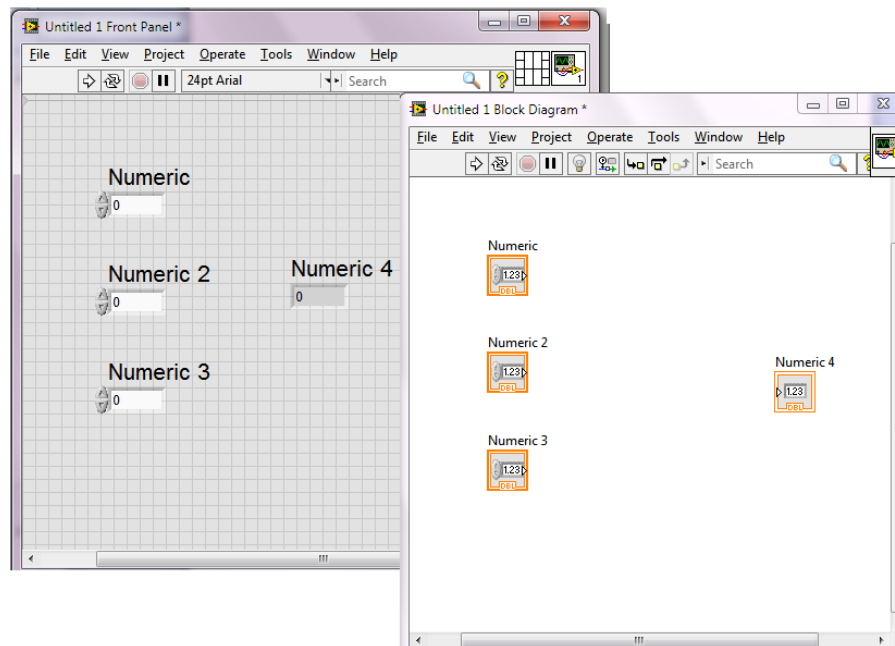


Figura 6.23: Diagrama de bloques
Fuente: Por el investigador

3. Se deben ir situando las funciones, estructuras, que se requieran para el desarrollo del programa, las cuales se unen a los terminales mediante cables como se muestra en la figura 6.24.

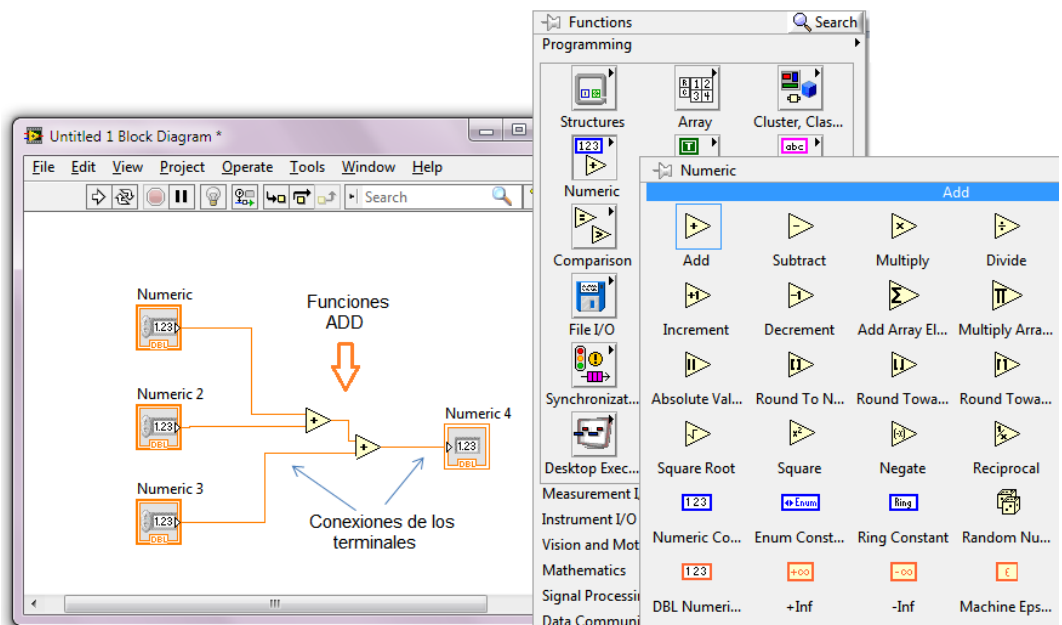


Figura 6.24: Uso de las funciones
Fuente: Por el investigador

- Para etiquetar cada control e indicador, se selecciona la pestaña *View* de la barra de menús y se da clic en *Edit Text*. Ahora se da clic en cualquier etiqueta y se escribe el nombre que deseamos. La figura 6.25 muestra este procedimiento descrito anteriormente.

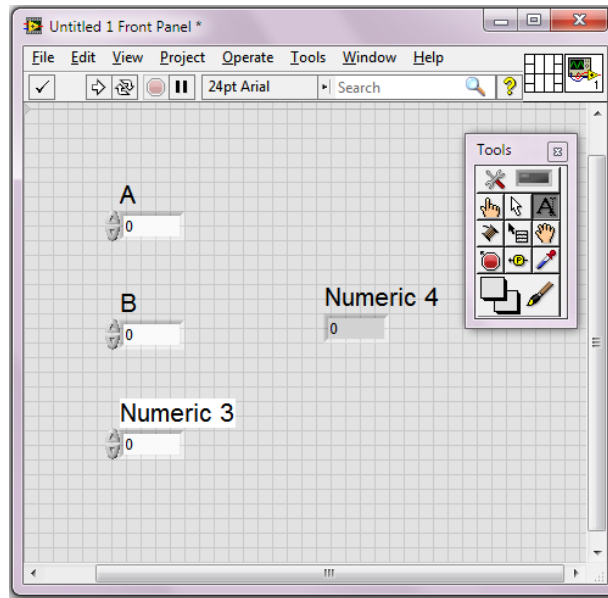


Figura 6.25: Etiquetando los controles
Fuente: Por el investigador

- Finalmente se debe utilizar una función *While Loop* para permitir que el programa creado se repita hasta presionar el botón Stop. La figura 6.26 muestra el programa finalizado.

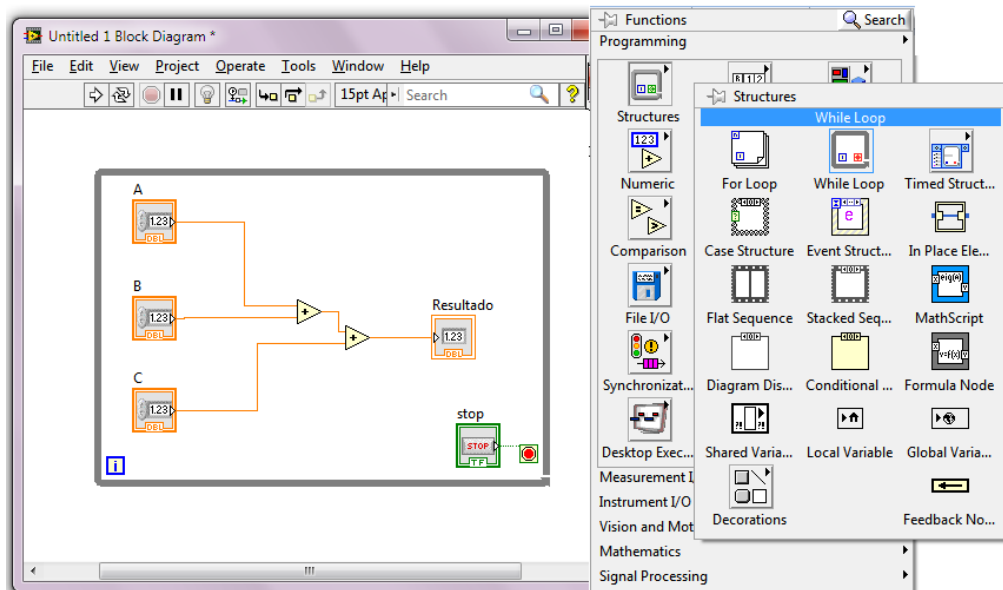


Figura 6.26: Creación del ciclo While-loop
Fuente: Por el investigador

6.6.12 Ejecución de un VI

Una vez se ha concluido la programación del VI anterior, se debe proceder a su ejecución. Para ello, la ventana activa debe ser el panel frontal donde se pulsará el botón de *Run* que esta dentro del círculo de la figura 6.27.

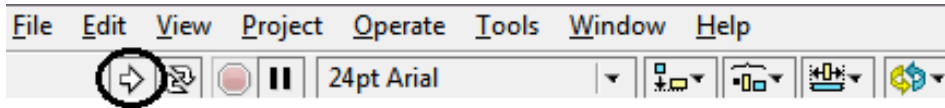


Figura 6.27: Botón Run
Fuente: Por el investigador

El programa comenzará a ejecutarse. Mientras dura la ejecución del mismo, cambia la apariencia del botón de *Run* tal como se muestra en la figura 6.28. De este modo el programa se ejecutará una sola vez.

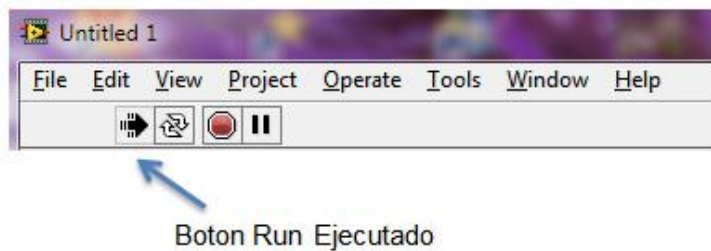


Figura 6.28: Corriendo la aplicación
Fuente: Por el investigador

La figura 6.29 muestra el botón *Continuous Run* que permite una ejecución continua, al ser pulsada. Si durante el funcionamiento continuo del programa se vuelve a pulsar el citado botón, se finalizará la última ejecución del mismo, tras lo cual el programa se parará.

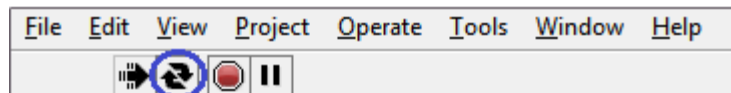


Figura 6.29: Botón Continuous Run
Fuente: Por el investigador

En la figura 6.30 se tiene dos botones para finalizar la ejecución de un programa La primera, y la más aconsejable, es el botón *Abort Execution*, cuya pulsación produzca la interrupción del bucle de ejecución de la aplicación. La segunda forma de detener la ejecución del VI es pulsando el botón de pausa o el de *stop*.

La diferencia entre ambos es que si se pulsa *Abort Execution*, la ejecución del programa finaliza inmediatamente, mientras que si se pulsa *Stop*, se produce una detención en el funcionamiento del programa, retomándose su ejecución una vez se vuelve a pulsar el mismo botón.

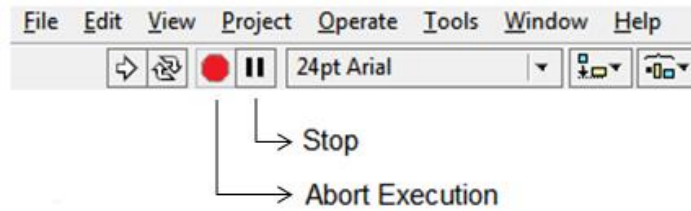


Figura 6.30: Botones Abort Execution y Stop

Fuente: Por el investigador

La figura 6.31 muestra la apariencia del panel frontal luego de haber presionado el botón RUN. La aplicación de la suma de tres valores, se encuentra ejecutándose correctamente.

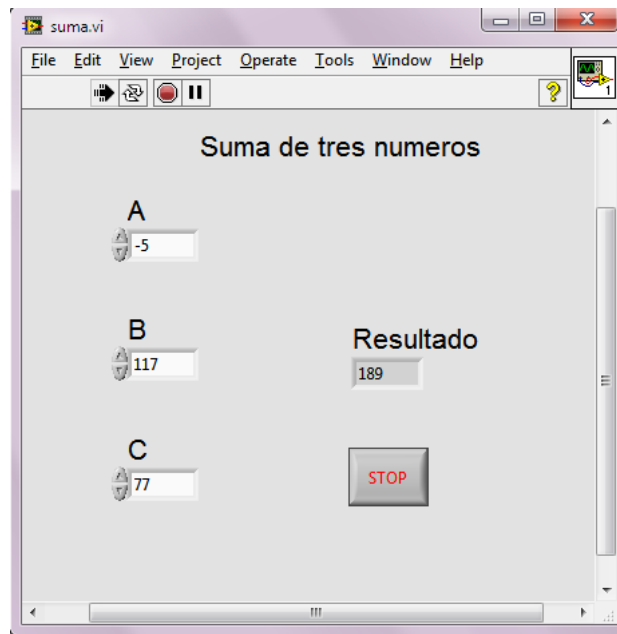


Figura 6.31: Ejecutando el VI

Fuente: Por el investigador

6.7 Metodología

Para el desarrollo de este proyecto en Labview 2011, fue importante analizar: los controles del panel frontal, la paleta de funciones del diagrama de bloques y su entorno. Gracias al *Toolkits Modulation* de labview, se logró encontrar los instrumentos virtuales necesarios para la implementación de un laboratorio virtual que simule las modulaciones digitales.

En primer lugar fue diseñar la interfaz de usuario en el panel frontal de labview; con los controles e indicadores apropiados para la manipulación del simulador de modulaciones digitales. Luego se realizó un estudio de cada instrumento virtual del *toolkits modulation* del diagrama de bloques, con el propósito de programar las técnicas de modulación digital. Además, surgió la idea de crear una aplicación adicional como son los códigos de línea. Finalmente se procedió a unir todos los objetos creados en el panel frontal con sus respectivos VIs y funciones, para cumplir con los requerimientos de un simulador de modulaciones digitales.

6.8 Modelo operativo

6.8.1 MODELAMIENTO Y SIMULACIÓN EN LABVIEW 11.0

Para el Laboratorio virtual de modulación-demodulación digital, se ha utilizado la plataforma Labview versión 11.0, el cual posee un entorno grafico y permite la creación de instrumentos virtuales. El laboratorio virtual propuesto, esta compuesto de un simulador de modulaciones digitales, con los elementos necesarios para tener practicas de calidad. En la figura 6.32, se especifica los instrumentos virtuales implementados:

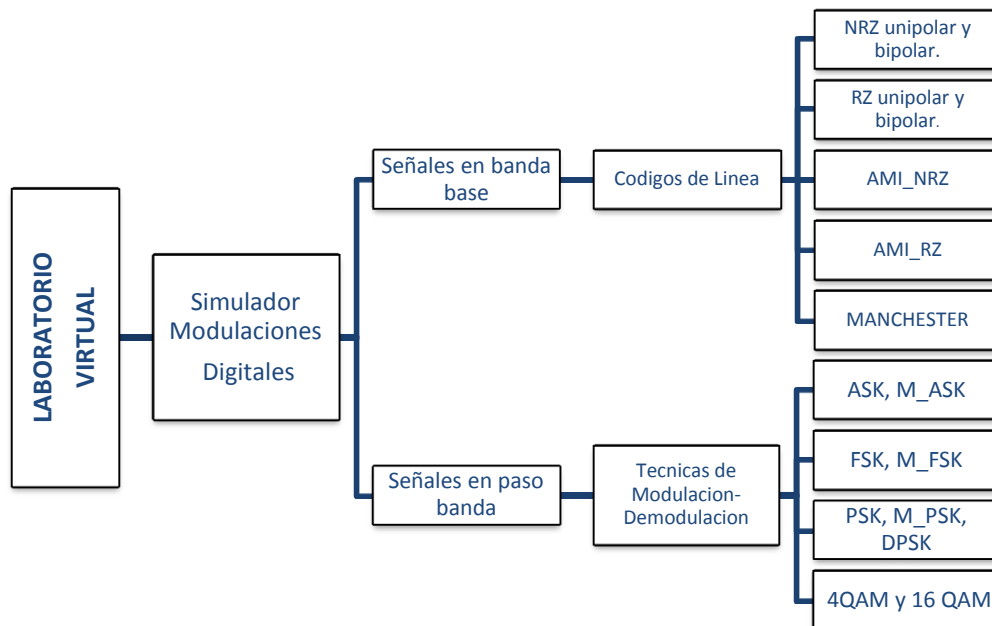


Figura 6.32: Instrumentos del Laboratorio Virtual

Fuente: Por el investigador

El laboratorio virtual propuesto tiene herramientas muy importantes para analizar las señales en banda base y pasabanda. A continuación se detallan, los parámetros necesarios para simular estos dos tipos de señales:

Señales en banda base

En el simulador de señales en banda base, se tiene los formatos básicos de la codificación de línea que se mencionaron en la anterior grafica. De acuerdo a esta aplicación, se ha creado los siguientes elementos para la interfaz grafica de señales banda base:

- Tipo de Código Línea
- Datos a codificar.
- Frecuencia: muestreo y de bits.
- Conjunto de controles.
- Representación señal de entrada, codificación en tiempo y espectro de potencia.

Señales en pasa banda.

En el simulador de señales en pasa banda, se tiene las modulaciones digitales básicas que se mostraron en la figura 6.32. Por consiguiente, se ha creado un instrumento virtual de modulaciones digitales que constan:

- Tipo de modulación.
- Parámetros del Mensaje.
- Parámetros: sincronización, filtro y diagrama de ojos.
- Niveles M-Aria.
- Tasa de símbolos.
- Frecuencia portadora.
- AWGN.
- Medidas de la señal demodulada.
- Potencia promedio del espectro de potencia.
- Representación de la señal de los símbolos, señales moduladas, espectro de potencia, constelación y diagrama de ojo.

6.8.2 CÓDIGO DE LÍNEA NRZ_U (No Retorno a Cero Unipolar)

6.8.2.1 Descripción de elementos a ser utilizados en el código NRZ_U.

PANEL FRONTAL DEL INSTRUMENTO VIRTUAL

Los elementos que se utilizaron en el panel frontal para el código de línea NRZ_U (No Retorno a Cero Unipolar), son los siguientes:

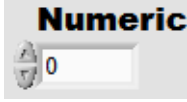
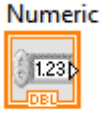
PESTAÑA DE CONTROLES>>NUMERIC>>NUMERIC CONTROL	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
Sirve para ingresar datos en diferentes formatos numéricos y definir los números de decimales, los límites o rangos.	

Figura 6. 33: Control numérico

Fuente: Por el investigador

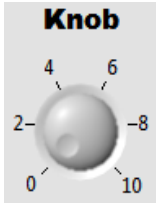

PESTAÑA DE CONTROLES>>NUMERIC>>KNOB	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
Es utilizado para variar los parámetros como por ejemplo: amplitud, frecuencia, fase, etc. Se puede establecer los límites de esta herramienta.	

Figura 6. 34: Control de Perilla

Fuente: Por el investigador

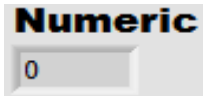
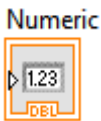
PESTAÑA DE CONTROLES>>NUMERIC>>NUMERIC INDICATOR	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
Es un indicador numérico, utilizado solo para mostrar un resultado en cualquier formato y en cualquier parte del programa.	

Figura 6. 35: Indicador numérico

Fuente: Por el investigador

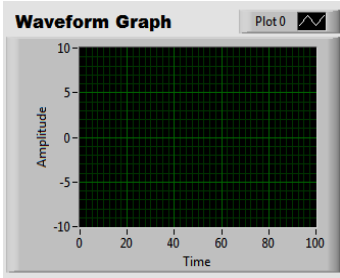

PESTAÑA DE CONTROLES>>GRAPH>>WAVEFORM GRAPH	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
<p>Esta herramienta permite visualizar cualquier forma de onda, como también permite fijar las escalas necesarias para el usuario.</p>	

Figura 6.36: Gráfico de Forma de Onda

Fuente: Por el investigador

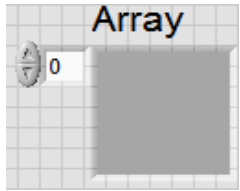

ARRAY	
PANEL FRONTAL	DIAGRAMA BLOQUES
	
<p>Este control representa un array vacío. Para crear un array hemos de llevar un objeto dentro de la ventana de objetos.</p>	

Figura 6.37: Array

Fuente: Por el investigador

DIAGRAMA DE BLOQUES DEL INSTRUMENTO VIRTUAL

A continuación se describen cada uno de los elementos utilizados en el diagrama de bloques para implementar el *código de línea NRZ_U* (No Retorno a Cero Unipolar), si necesita una descripción más detallada de los terminales de cada uno de los elementos del diagrama de bloques, se debe dirigir **Anexos 1**.

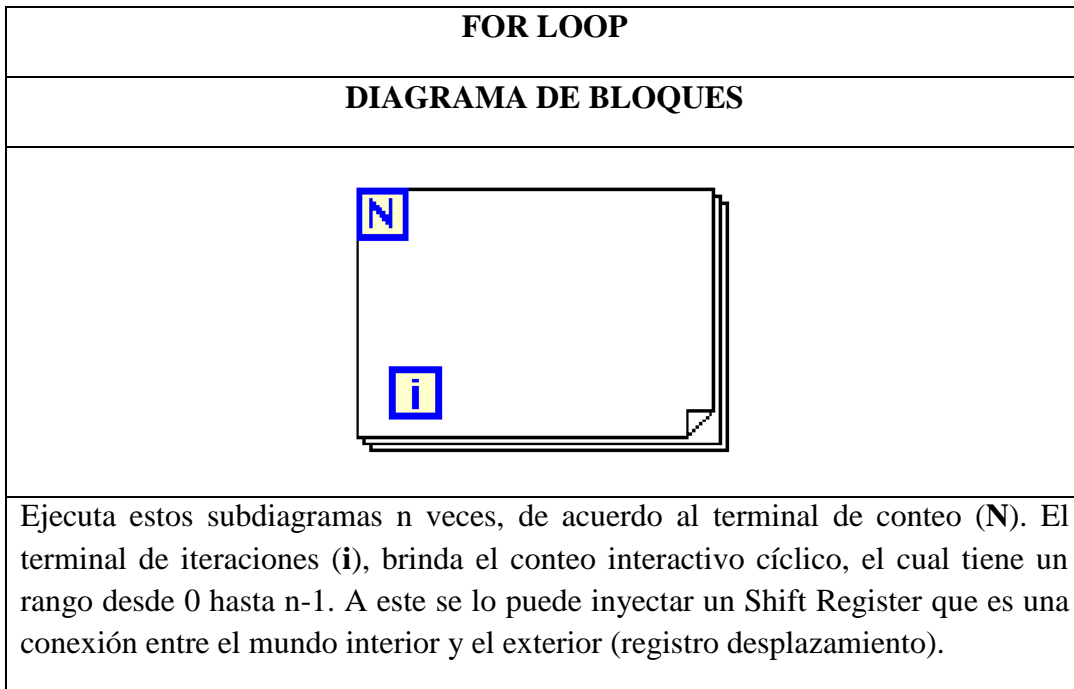


Figura 6.38: Ciclo For Loop
Fuente: Por el investigador

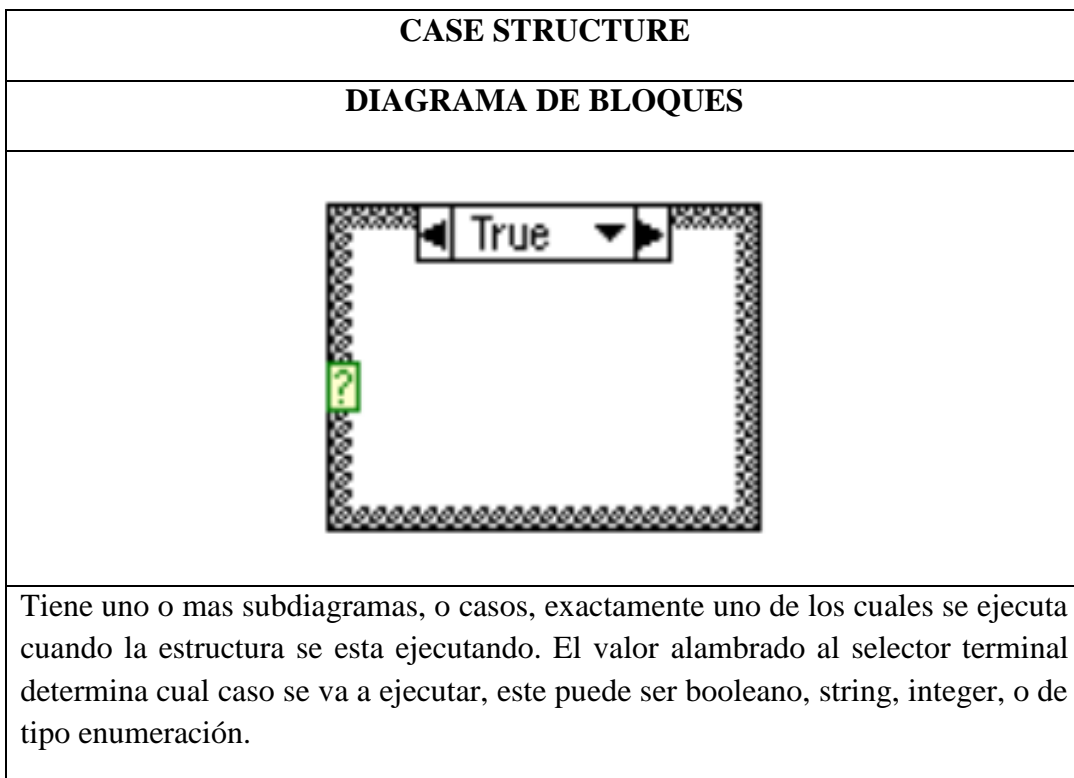


Figura 6.39: Case Structure
Fuente: Por el investigador

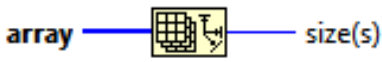
ARRAY SIZE
DIAGRAMA DE BLOQUES
<p>Array Size</p> 
<p>Esta función entrega el número de elementos del arreglo de entrada. Si el arreglo de entrada es de dos o mas dimensiones, la salida es un arreglo de dos o más elementos.</p>

Figura 6.40: Array Size
Fuente: Por el investigador

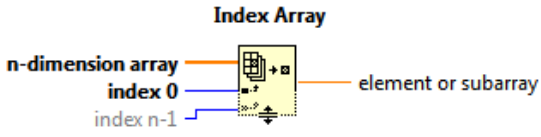
INDEX ARRAY
DIAGRAMA DE BLOQUES
<p>Index Array</p> 
<p>Esta función accede al elemento del arreglo de entrada especificado en la terminal Index. Si el arreglo de entrada es multidimensional se tienen que agregar terminales index adicionales, agrandando el icono de la función.</p>

Figura 6.41: Index Array
Fuente: Por el investigador

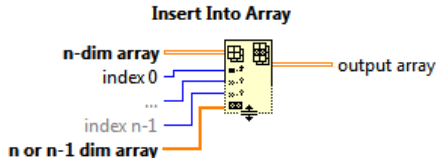
INSERT INTO ARRAY
DIAGRAMA DE BLOQUES
<p>Insert Into Array</p> 
<p>Inserta un elemento o subarray dentro n-dim array en el punto que se especifique en el índice. Al conectar una matriz para esta función, la función cambia de tamaño automáticamente para mostrar las entradas de índice para cada dimensión de la matriz.</p>

Figura 6.42: Insert Into Array
Fuente: Por el investigador

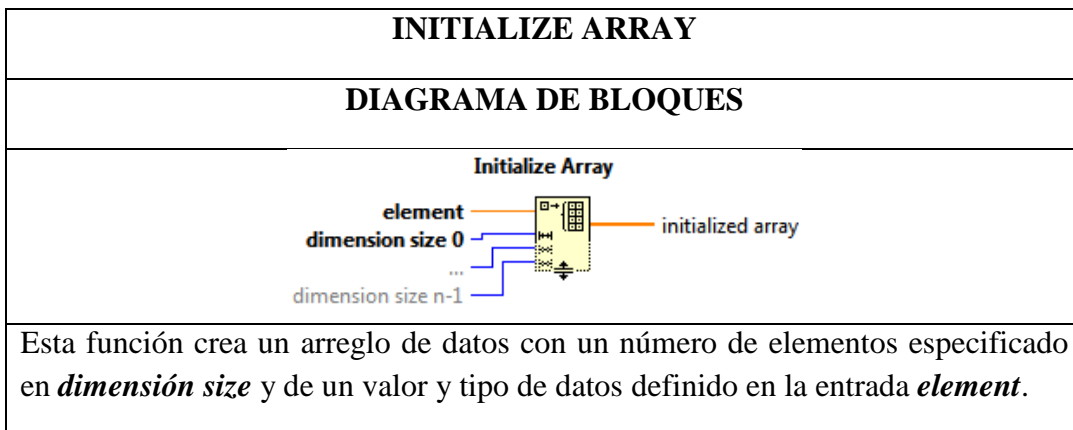


Figura 6.43: Initialize Array
Fuente: Por el investigador

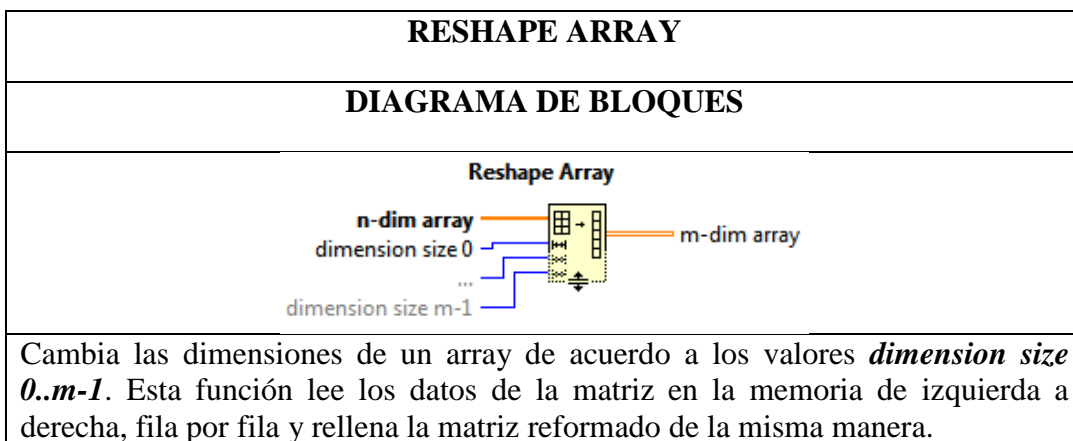


Figura 6.44: Reshape Array
Fuente: Por el investigador

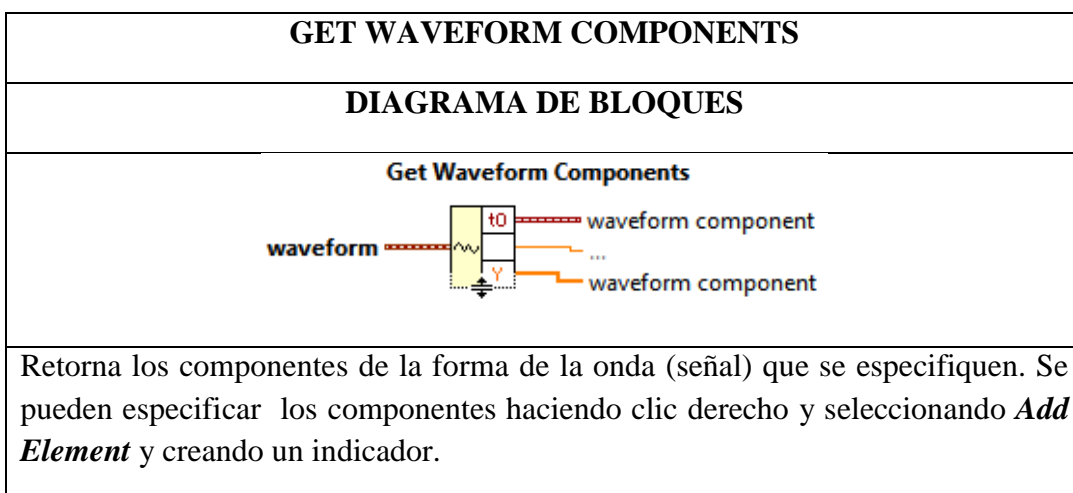


Figura 6.45: Get waveform components
Fuente: Por el investigador

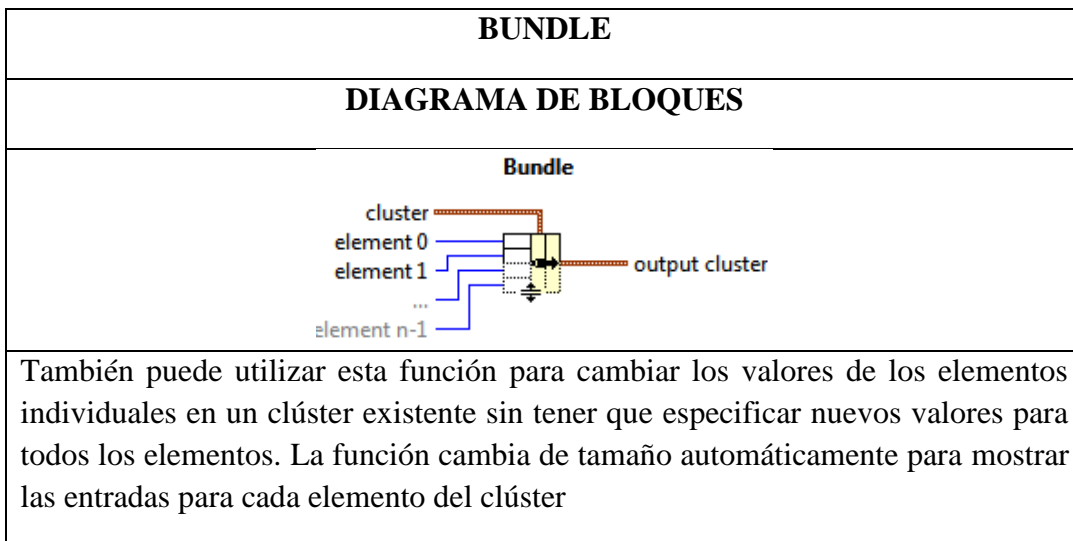


Figura 6.46: Bundle
Fuente: Por el investigador

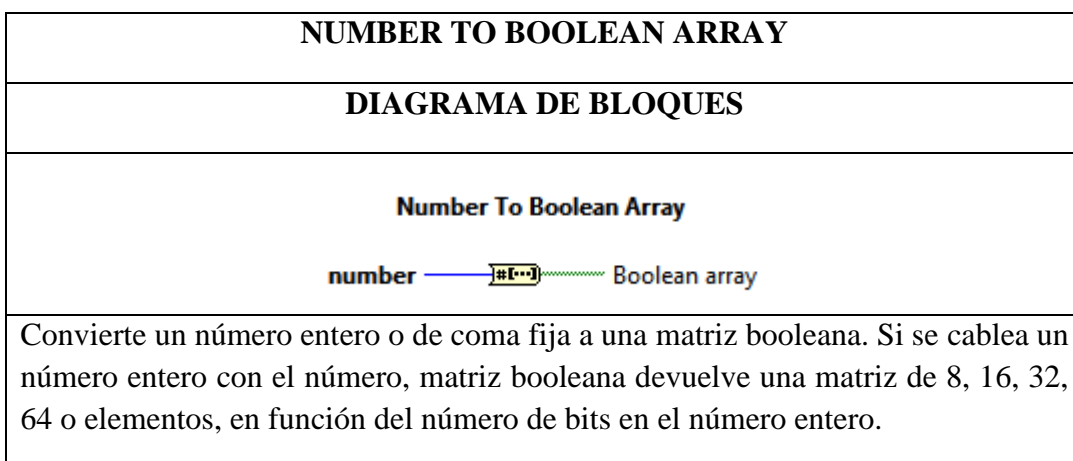


Figura 6.47: Number to Boolean array
Fuente: Por el investigador

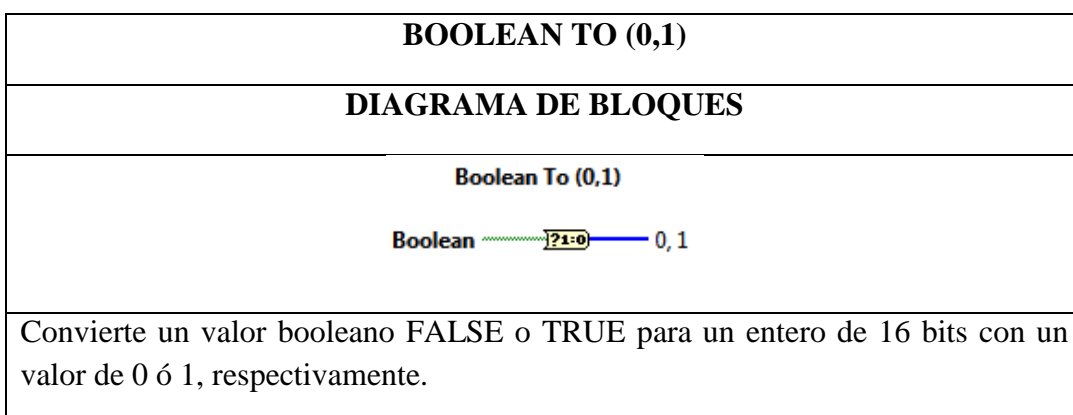


Figura 6.48: Boolean to (0,1)
Fuente: Por el investigador


SQUARE WAVEFORM
DIAGRAMA DE BLOQUES

Genera una forma de onda cuadrada, según los parámetros que tengan sus entradas como: fase, amplitud, frecuencia, etc.

Figura 6.49: Square waveform
Fuente: Por el investigador

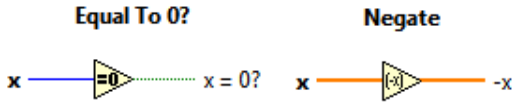
EXPRESIONES MATEMATICAS
DIAGRAMA DE BLOQUES

<i>Equal To 0?</i> , devuelve TRUE si x es igual a 0. De lo contrario, esta función devuelve FALSO. Mientras que la función <i>Negates</i> , niega el valor de entrada.

Figura 6.50: Expresiones matemáticas
Fuente: Por el investigador

DISEÑO DEL VI PARA EL ESQUEMA DE CODIGO NRZ_U

En el diseño del VI para el esquema de codificación de línea *No retorno a cero unipolar* (NRZ_U) se aplicó los siguientes pasos.

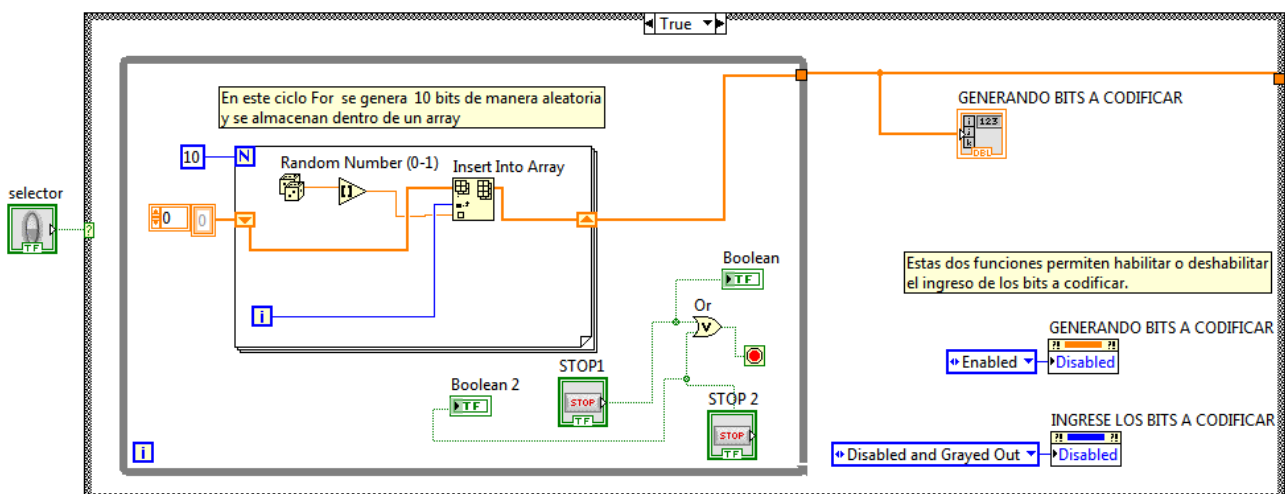
INICIALIZACION DEL INSTRUMENTO VIRTUAL (VI)

- 1. Inicio/** Todos los programa/ National Instruments/Labview 2011.
- 2. New/** Blank VI o digitar el comando Ctrl+N.
- 3. Aparecen** dos ventanas: **panel frontal**, donde se podrá visualizar todos los controles e indicadores que el usuario pueda utilizar y **diagrama de bloques**, en la que se encuentra toda la programación para diseñar un VI.

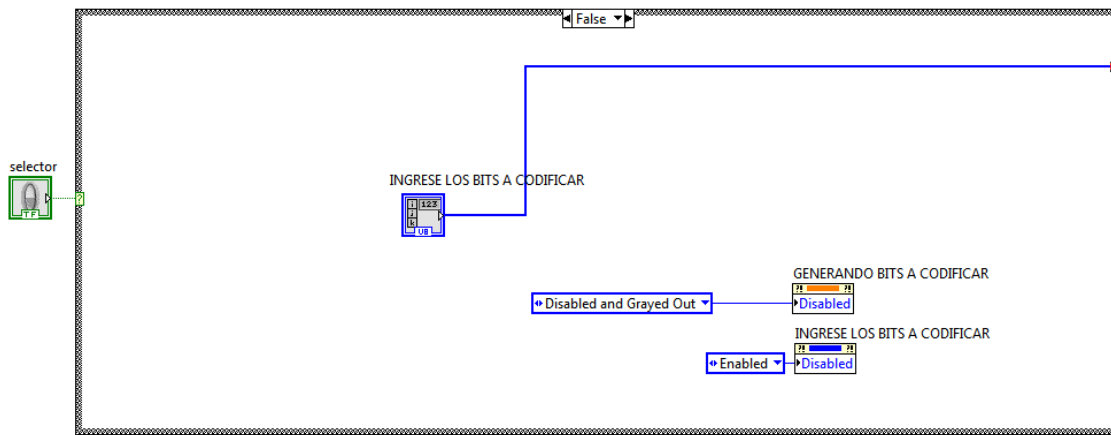
PROCEDIMIENTO:

1. Se inicializa un nuevo VI.
2. Como segundo paso, es crear una estructura condicional que permita elegir dos opciones: la primera para generar los bits a codificar de manera aleatoria y la segunda para ingresar los bits a codificar manualmente. Teniendo en cuenta que necesitamos de una secuencia de 10 bits en cualquiera de las dos opciones anteriores.
3. Para cumplir con el paso 2, se da clic derecho en el diagrama de bloques para visualizar la paleta de funciones. Ahora se procede a la utilización de un *case structures*, en el *case true* se programa como muestra la figura 6.51, que servirá para generar los bits aleatorios a codificar mediante el uso de un ciclo *For loop* (repite el proceso diez veces), un *Random* (genera diez valores aleatorios), un *Number (0-1)* que permite que los valores aleatorios se aproximen a valores entre 0,1 y *el Insert Into Array* (permite almacenar cada valor dentro de un array). Finalmente, se utiliza un ciclo *While Loop* para repetir el proceso de generar bits para la codificación, hasta pulsar el botón *Stop 1* y en cambio el *Stop 2* permitirá también detener la generación de bits aleatorios cuando se escojan la otra opción ingresar bits. La función que permite activar o desactivar un elemento del diagrama de bloques es la *funcion Disabled*, en este caso lo utilizamos en los dos array encargados del ingreso de bits a codificar.

En la figura 6.52b se indica el *case false* en donde se programa el ingreso de diez bits a codificar de manera manual y también se utilizan las *funciones Disabled* que ya lo explicamos en el paso anterior.



(a)



(b)

Figura 6.51: Esquema del ingreso de bits: (a) aleatorios; (b) manualmente

Fuente: Por el investigador

- En la figura 6.52, se muestra con una flecha el control de los bits a codificar. Para esto, se utilizan un *Array Size* que devuelva el tamaño del array denominado *generando bits a codificar* y del array *ingrese bits a codificar*, para luego compararla con una constante de 0. Este proceso de comparación permite tomar la decisión de si hay o no bits para su posterior codificación.

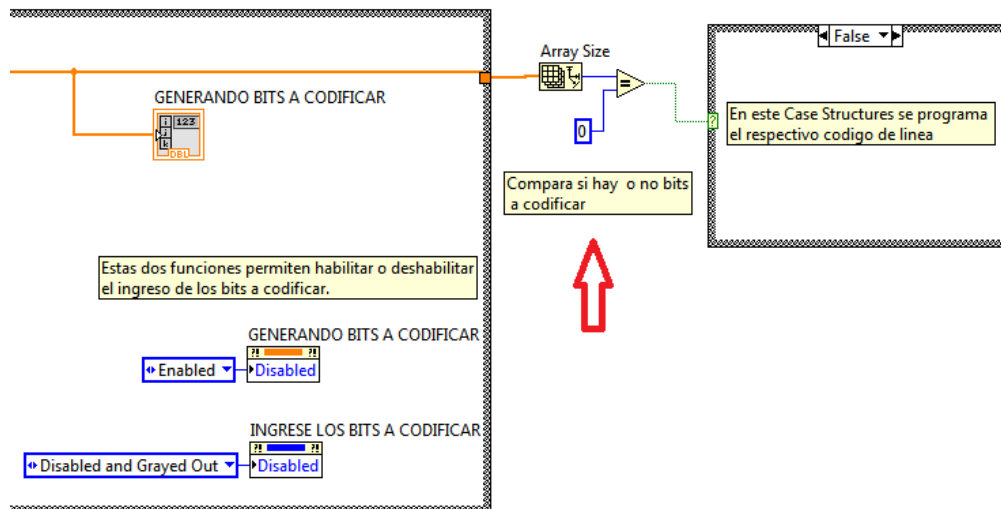


Figura 6.52: Control de los bits a codificar

Fuente: Por el investigador

- Continuando con la programación, la siguiente etapa es tener dos controles numéricos: *frecuencia de muestreo* y la *frecuencia de bits*, de modo que nos dirigimos al panel frontal: *paleta de controles/Numeric Control*.

6. Proseguimos con el uso de otro *ciclo For*: para comparar a cada elemento de la conversión anterior con el valor de cero y relacionar las frecuencias, mientras que dentro de un *Case Structures* se debe cumplir con las condiciones del código NRZ_U que es: asignar un +V, al símbolo “1” y 0V al símbolo “0”.
7. Para saber si los bits del array es “0” o “1”, se debe comparar los valores del *Index Array*, que se encuentra: *paleta de funciones/Array* con el *Equal to 0*, ubicado en la misma paleta pero en la función *Comparison*, para que de esta manera se pueda relacionar las frecuencias con los datos de entrada. Se empieza a multiplicar la *frecuencia de muestreo* por el *reciproco* de la *frecuencia de bits*, su resultado será las veces que se repetirá este ultimo *For* y servirá para definir la dimensión del *Initialize array*, que permite obtener un nuevo array. Para obtener el recíproco de un número, nos ubicamos: *paleta de funciones/numeric/reciprocal*. La figura 6.53 muestra la programación de este paso.

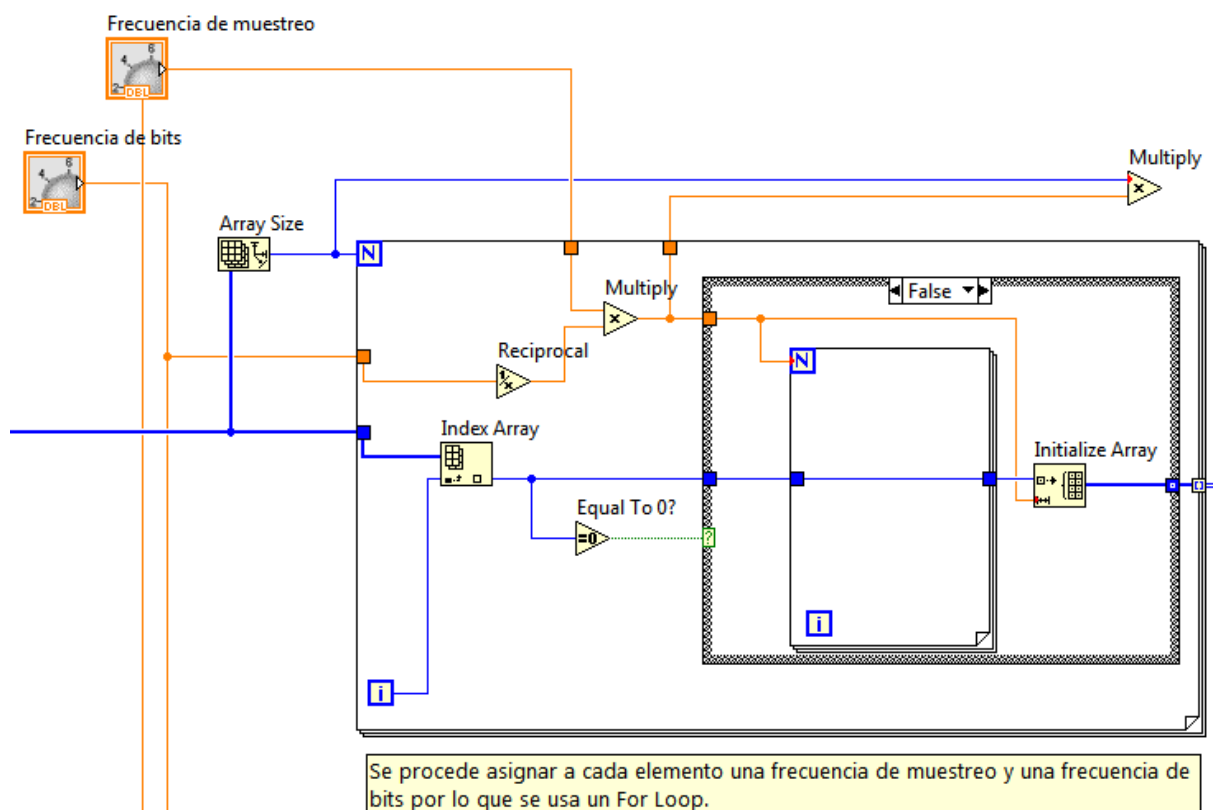


Figura 6.53: Código NRZ_U
Fuente: Por el investigador

8. La salida de la función *Initialize Array*, tendrá que conectarse a un *Reshape Array*, para leer la matriz en forma ordenada y construir la forma de onda con el *Build Waveform* que se encuentra: *paleta de funciones/Array*, su

salida estará conectado a un *Waveform Graph* para visualizar la señal de entrada de bits. En la figura 6.54 muestra la manera de conectar el terminal *dimension size* de la función *Reshape Array* con el producto de las dos frecuencias.

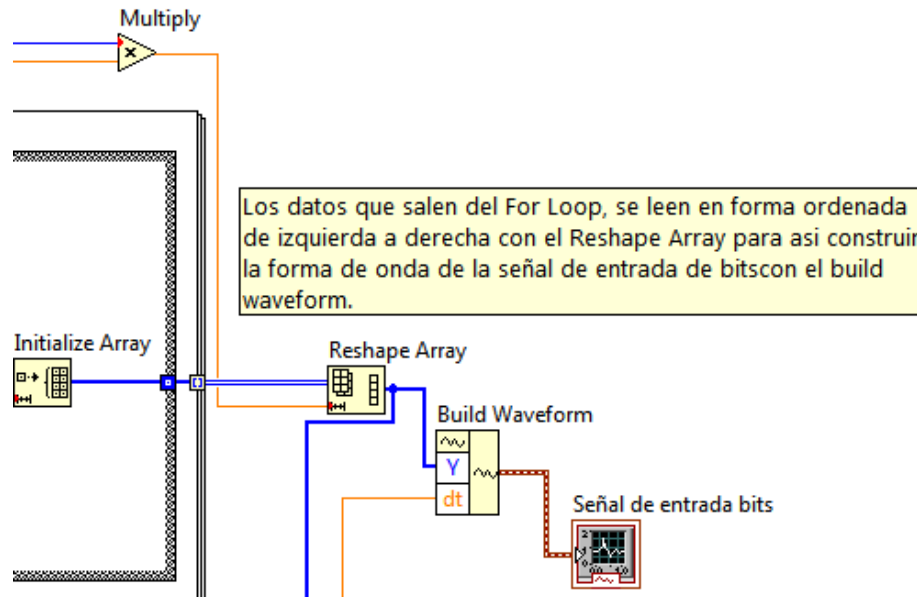


Figura 6.54: Señal de entrada en bits
Fuente: Por el investigador

9. Por otro lado, necesitamos una función *Square Waveform* que nos permita generar una señal cuadrada que se pueda obtener sus componentes *Y* y el *dt* con el *Get Waveform Components*. La *componente Y* deberá multiplicarse por los datos del segundo ciclo For para visualizar la señal codificada en el tiempo y el *dt* debe estar unido a todos los terminales *dt* de los *Build Waveform*, caso contrario no se puede tener sincronizada las señales, ver la figura 6.55.

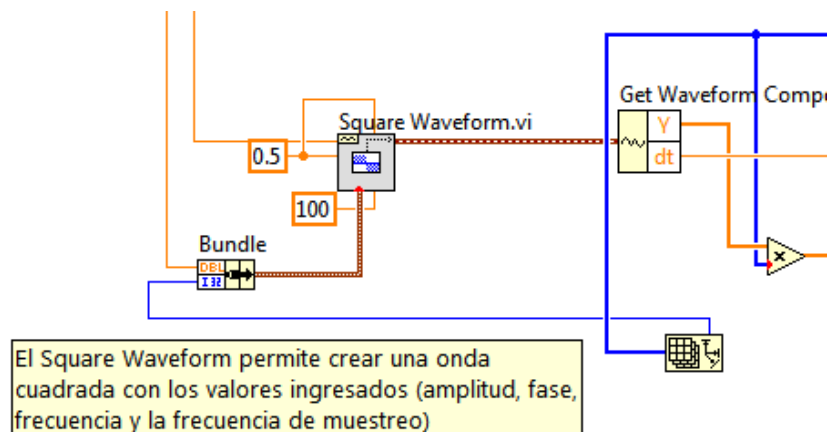
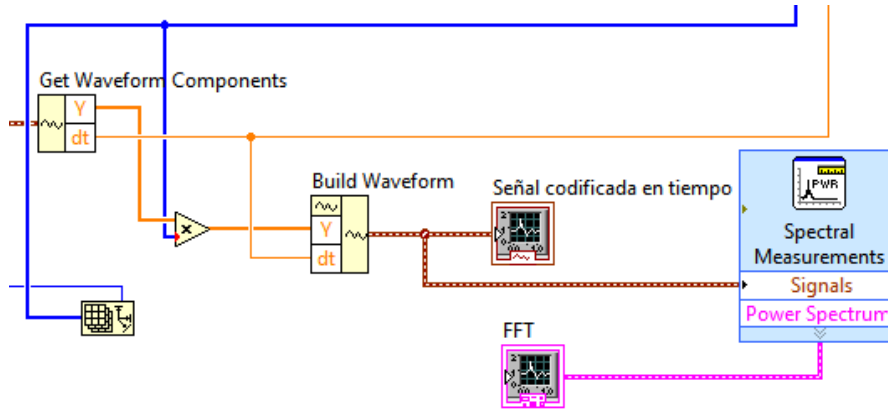


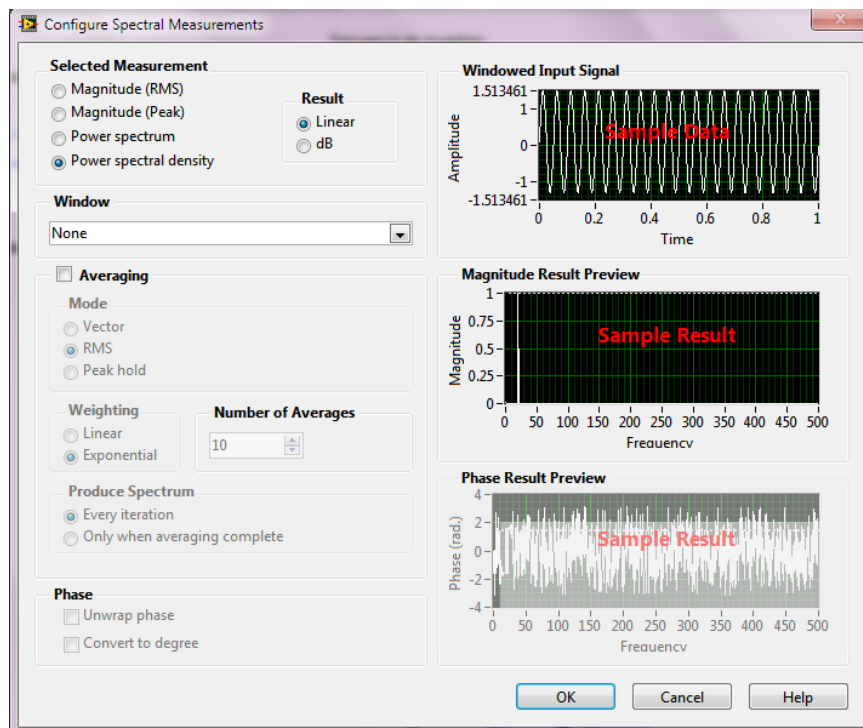
Figura 6.55: Creación onda cuadrada
Fuente: Por el investigador

10. Por ultimo, de la salida del *Build waveform* se obtiene la señal codificada en tiempo y de la misma salida se puede visualizar el espectro de potencia como se indica en la figura 6.56a, proveniente del *Spectral Measurements* que debe estar configurado correctamente, esto se muestra en la figura 6.56b.



Con la ayuda del Get waveform components, se obtiene las componentes de la onda cuadrada y su componente (y), se multiplica por los valores obtenidos del For Loop. Finalmente se vuelve a construir una forma de onda para mostrar la señal codificada y el espectro de potencia

(a)



(b)

Figura 6.56: Señales y configuración: (a) Señal codificada en tiempo y espectro de frecuencia (FFT); (b) Configuración Spectral measurements

Fuente: Por el investigador

Panel frontal del instrumento virtual

La figura 6.57 muestra el Panel Frontal del código de línea *No retorno a Cero Unipolar* (NRZ_U) en donde los bits a codificar se lo han ingresado de manera manual. Se puede observar que la señal codificada en tiempo cumple con la condición del presente código que es: asignar un $+V$, al símbolo “1” y $0V$ al símbolo “0”. Cada vez que se presione el botón *generar bits* tendremos diez bits aleatorios dentro de un array denominado *generando bits a codificar*, en cambio el botón *ingresar bits a codificar* permite el ingreso de bits en forma manual, para su posterior codificación de línea. Para escoger una de las dos opciones anteriores del ingreso de bits a codificar se utiliza el control booleano que tiene dos *posiciones* (1 o 2), en nuestro panel esta seleccionado en modo manual. La *frecuencia de muestreo* en cambio es un control encargado de variar la frecuencia a la que se desea muestrear la señal entrada (rango de frecuencia es 1 a 4KHz respectivamente). El control que varía la velocidad de los bits codificados es la denominada *frecuencia de bit* con un rango de dos veces menor a la *frecuencia de muestreo*, esta condición es propia de los códigos de línea. Por otro lado, el gráfico de la *señal de entrada de bits*, visualiza una señal binaria (con amplitudes de 1 o 0 voltios respectivamente) de acuerdo a los valores ingresados en el control *bits a codificar*. Ahora en la ventana *señal codificada en tiempo* se puede observar el resultado de la codificación de línea en el dominio del tiempo y en la otra ventana se muestra el *espectro de frecuencia* (FFT) del código implementado con sus respectivas amplitudes (eje y) e intervalos de frecuencia (eje x).

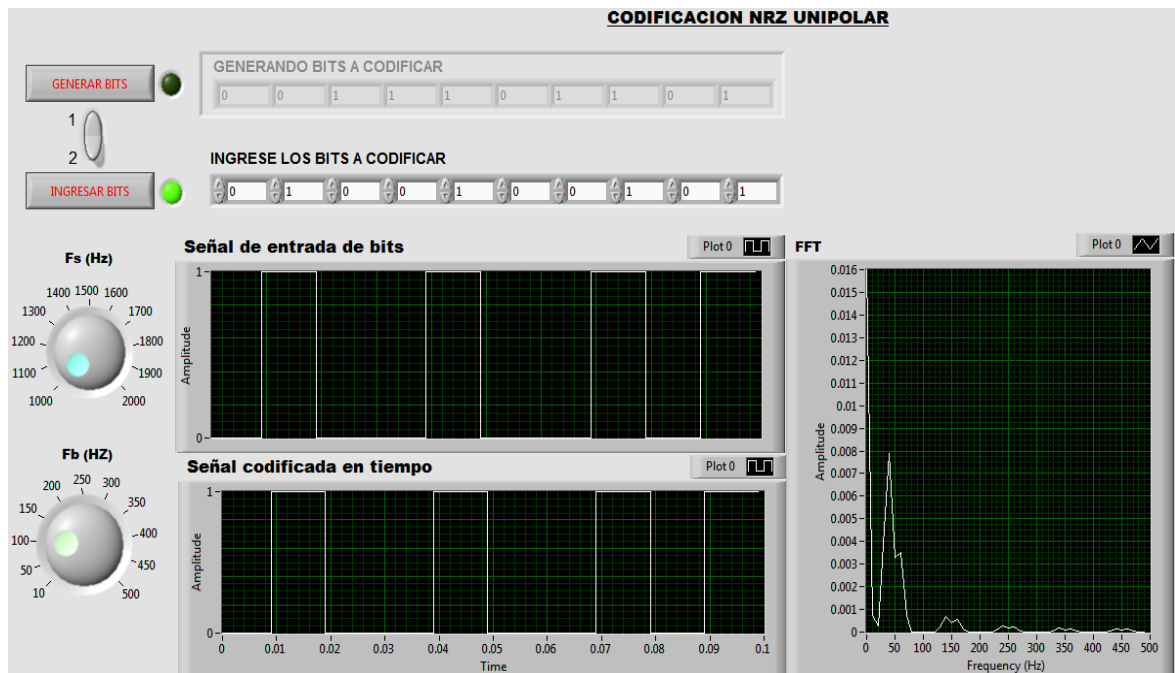


Figura 6.57: Panel Frontal del SubVI-NRZ_U

Fuente: Por el investigador

Diagrama de Bloques del instrumento virtual

La figura 6.58 muestra el diagrama de bloques de la programación del código NRZ-U (No retorno a cero unipolar) pero dividido en dos partes. En la figura 6.58a se donde se programa la manera de ingresar los bits y en la figura 6.58b se encuentra la programación encargada de cumplir con las condiciones del código NRZ_U que es: asignar un $+V$, al símbolo “1” y $0V$ al símbolo “0”.

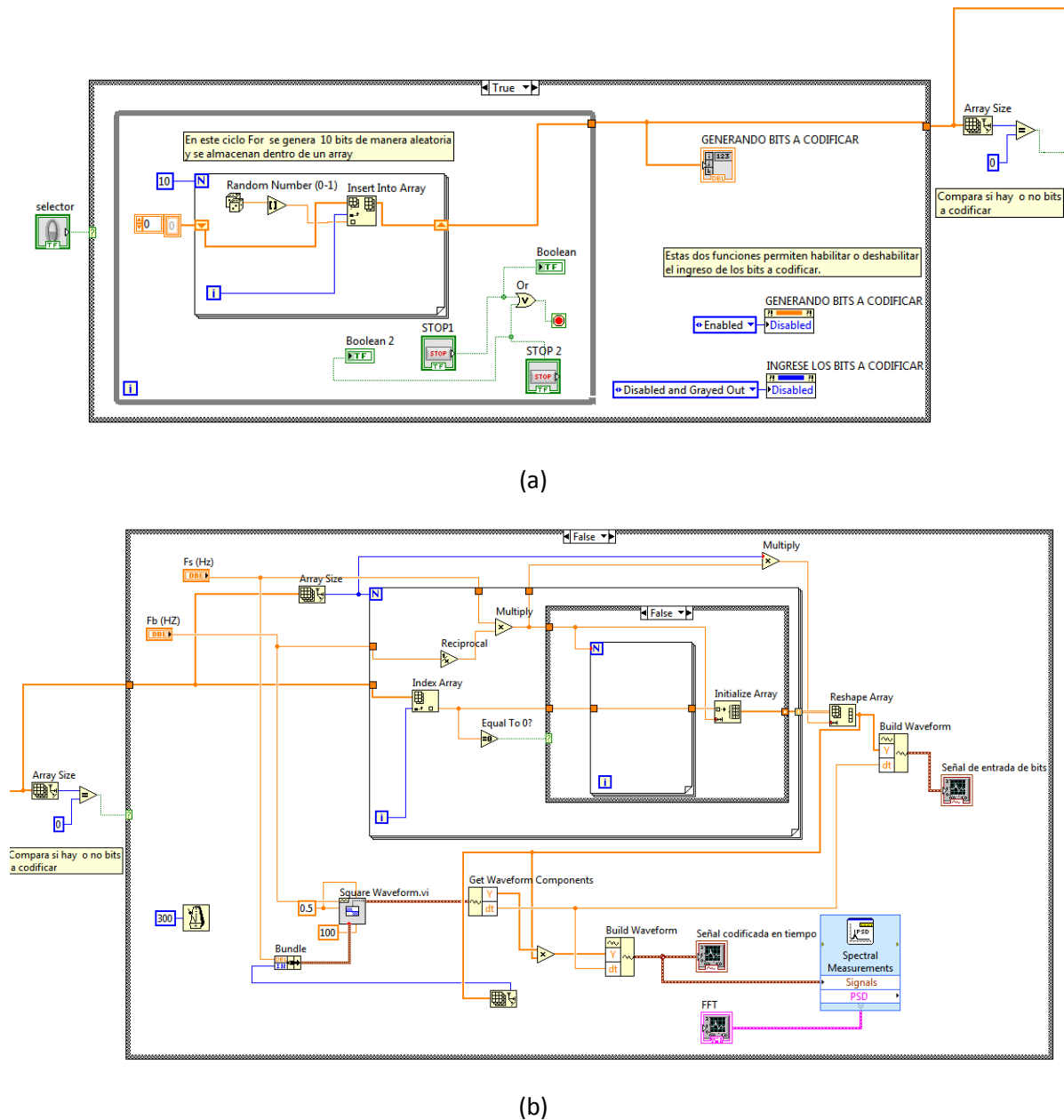


Figura 6.58: Diagrama de Bloques NRZ_U: (a) control del ingreso de bits; (b) codificación NRZ_U

Fuente: Por el investigador

6.8.3 CÓDIGO DE LÍNEA NRZ_P (No Retorno a Cero Polar)

6.8.3.1 Descripción de elementos a ser utilizados en el código NRZ_P

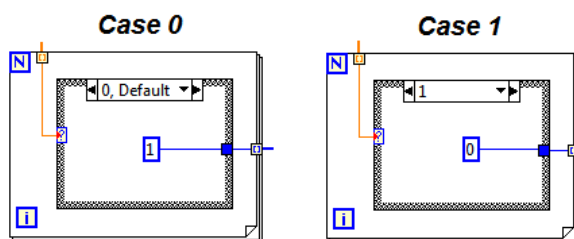
En la simulación del código *No retorno a cero polar*, se ha utilizado los mismos elementos del anterior código de línea (NRZ_U), tanto para el panel frontal como para el diagrama de bloques.

DISEÑO DEL VI PARA EL ESQUEMA DE CODIGO NRZ_P

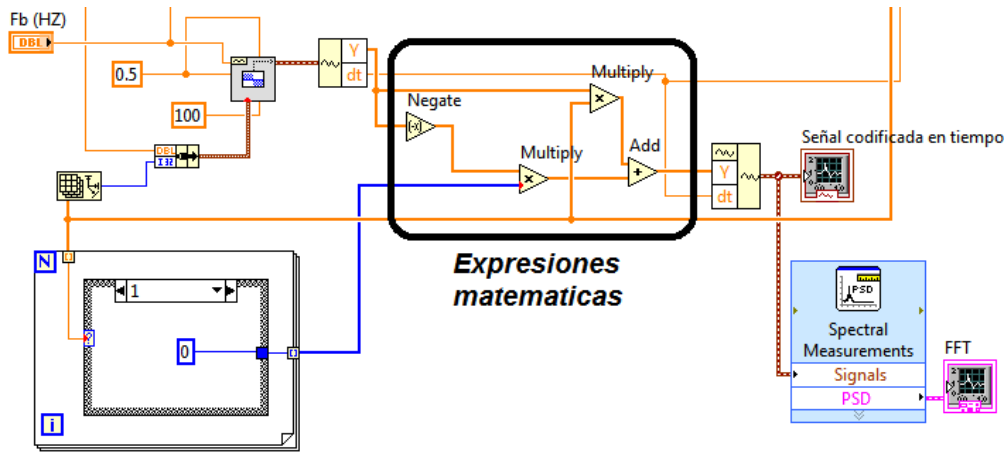
Para diseñar el esquema de codificación de línea *No retorno a cero polar*, se ha utilizado los pasos del código anterior *NRZ_U*, por su similitud. Sin embargo, el presente código cambia sus condiciones de codificación, por lo que se requiere seguir el siguiente procedimiento adicional:

PROCEDIMIENTO:

1. Se inicializa un nuevo VI.
2. La programación del ingreso de los bits a codificar se mantienen del código anterior NRZ_U, esto no cambia.
3. El siguiente procedimiento es cumplir con las condiciones de este código para eso, se realiza un control de los bits a codificar, en donde al **símbolo 1**, se le asigne un $+V$ y cuando el **símbolo es 0**, tendrá $-V$. Para esto, se debe utilizar la misma programación del código NRZ_U y adicionar, un cuarto ciclo *For* de la manera como se indica en la figura 6.59a en donde los bits a codificar serán comparados en un tercer *Case structures*, asignándoles un valor de 1 o 0 para que luego al pasar por algunas expresiones matemáticas tengamos como resultado las señal de entrada de bits, señal codificada y el espectro de frecuencia (FFT) como se muestra la figura 6.59b.



(a)



(b)

Figura 6.59: Codificación NRZ_P: (a) condiciones del *case structures*; (b) expresiones matemáticas

Fuente: Por el investigador

Panel frontal del instrumento virtual

La figura 6.60 muestra el panel frontal del código *No retorno a Cero Polar* (NRZ_P) con el ingreso de bits a codificar, en forma aleatoria o automática. Las descripciones de sus elementos son similares al primer código de línea NRZ_U con la diferencia que en la ventana de la señal codificada en tiempo muestra al **símbolo 1**, como un **+V** y al **símbolo 0**, tendrá **-V**.

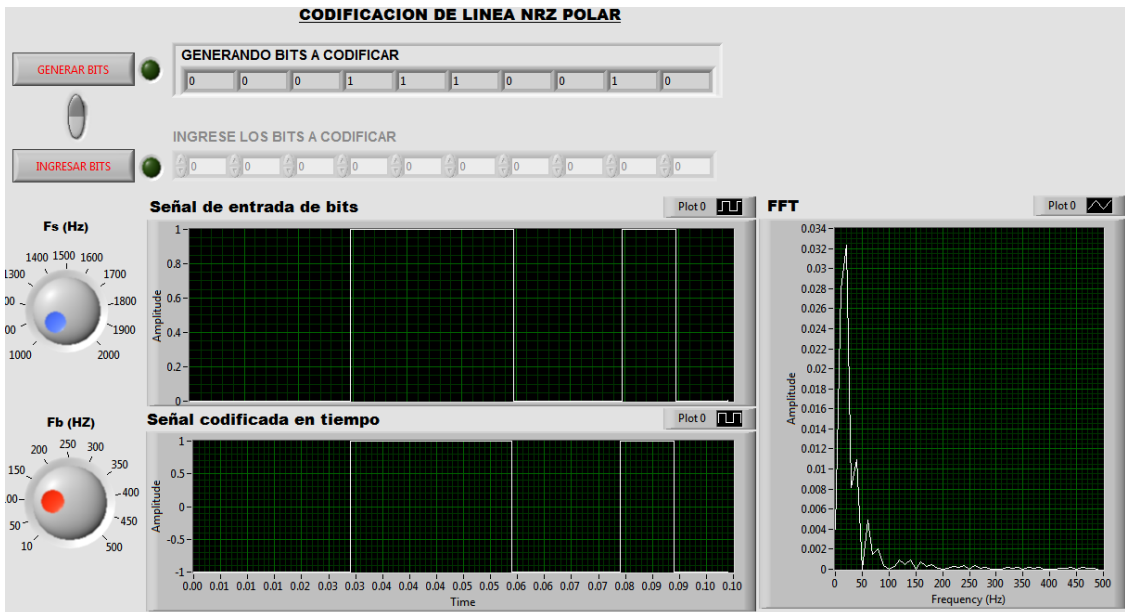
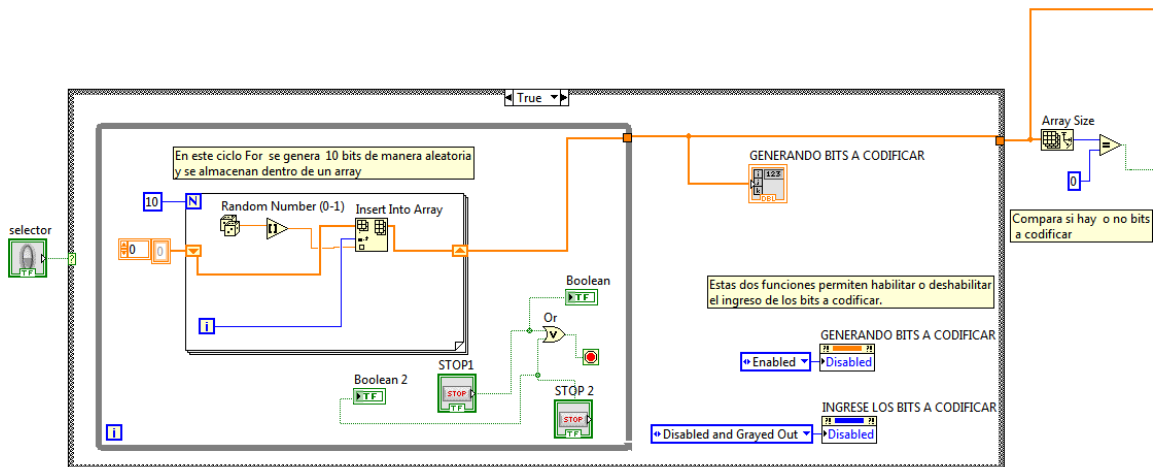


Figura 6.60: Panel Frontal del SubVI-NRZ_P

Fuente: Por el investigador

Diagrama de Bloques del instrumento virtual

En la figura 6.61 se muestran la programación por partes del código *No retorno a cero polar* (NRZ_P). Esto se lo ha realizado con el propósito de poder observar de una mejor manera cada parte de la programación realizada.



(a)

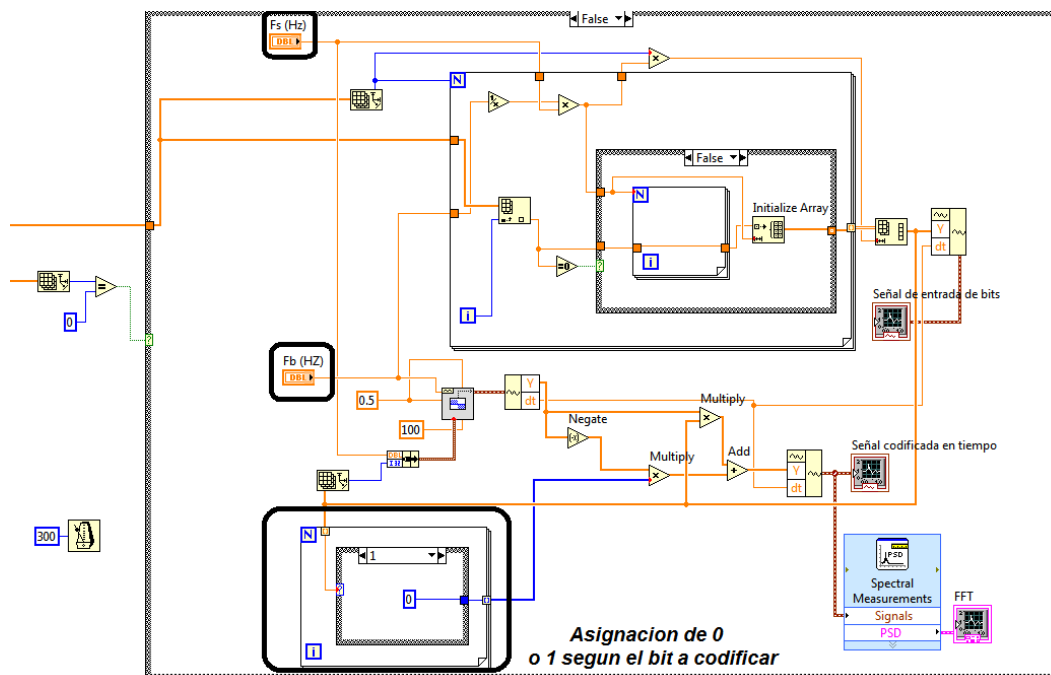


Figura 6.61: Diagrama de bloques Código NRZ_P: (a) control de ingreso de bits; (b) codificación NRZ_P

Fuente: Por el investigador

6.8.4 CÓDIGO DE LÍNEA RZ_U (Retorno a Cero Unipolar)

6.8.4.1 Descripción de elementos a ser utilizados en el código RZ_Unipolar

En la simulación del código *Retorno a cero unipolar*, se ha utilizado los mismos elementos del primer código (NRZ_U), tanto para el panel frontal como para el diagrama de bloques.

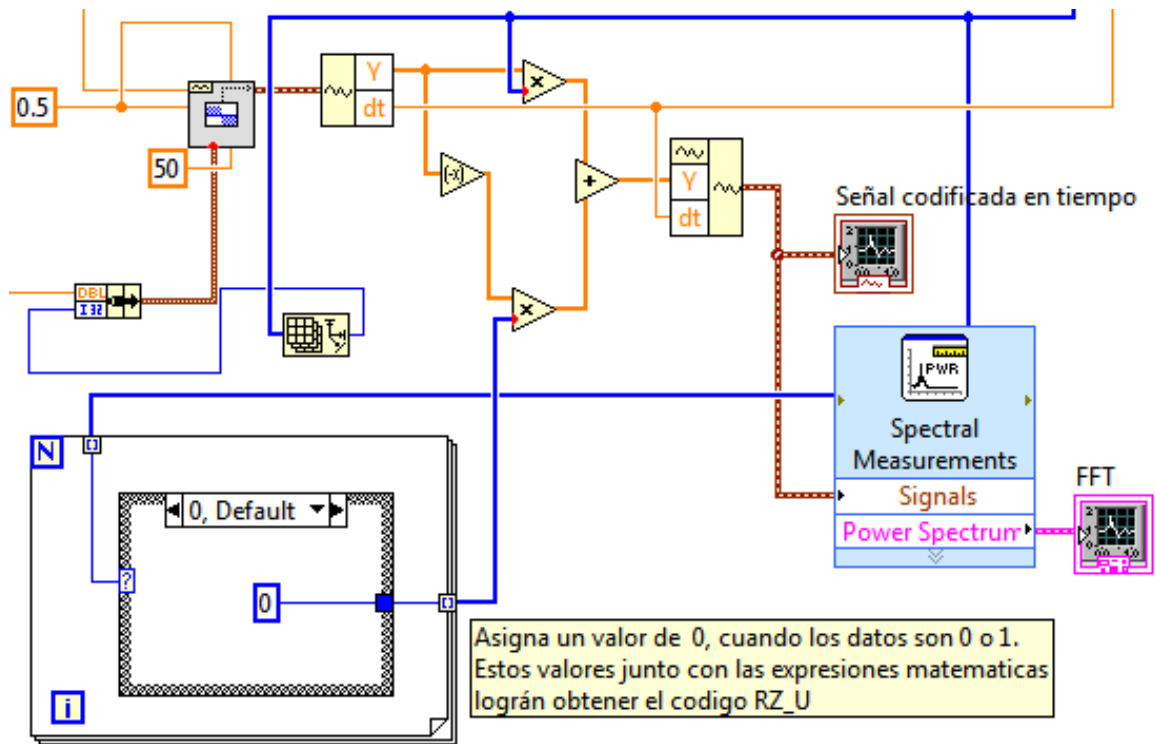
DISEÑO DEL VI PARA EL ESQUEMA DE CODIGO RZ_U

Para el diseño de la codificación de línea *Retorno a cero unipolar*, se ha mantenido la misma programación del código NRZ_U, tanto para el ingreso de los bits a codificar y la etapa de asignación de las frecuencias (f_s , f_b), a los bits a codificar. Sin embargo, el presente código tendrá una programación diferente para cumplir sus condiciones, por lo que se requiere seguir el siguiente procedimiento adicional.

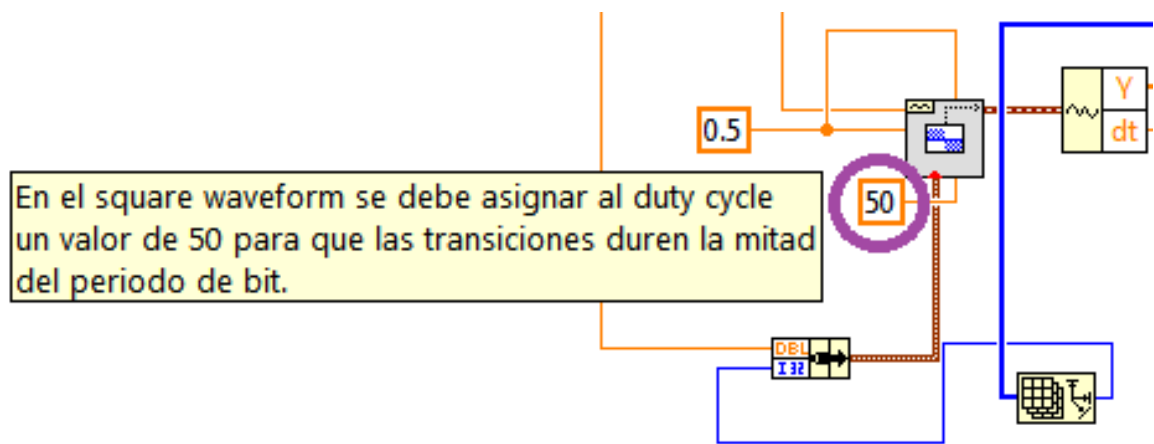
PROCEDIMIENTO:

A continuación, se describe los pasos para diseñar, la nueva etapa de la codificación de línea RZ_U.

1. Teniendo ya diseñado la codificación NRZ_P, se procede a cambiar los dos valores del *Case Structure* que se encuentra dentro del cuarto ciclo *For*. En ambos *case* (0, 1) se les asignara una *constante numérica* con un valor de 0 y con la ayuda de las expresiones matemáticas (*Add*, *Multiply*, *Negate*), se logró cumplir las condiciones para esta codificación RZ_U, vea la figura 6.62a. La condición para este código es, cuando se tiene un **símbolo 1**, tomara **+V durante la mitad del periodo de bit** y cuando el **símbolo es 0**, tendrá **0V durante todo el periodo de bit**. Los símbolos se refieren a los bits que se ingresen para la codificación que ya lo mencionamos anteriormente que puede ser de manera aleatoria o manual.
2. Como complemento del paso uno, para que el símbolo (1 o 0) duren un determinado periodo de bits se tiene que modificar el parámetro *duty cycle* del *Square Waveform* por un valor de **50** para que +V, dure la mitad del periodo de bits, vea la figura 6.62b.



(a)



(b)

Figura 6.62: Codificación RZ_U: (a) condiciones del case structure; (b) duración del periodo de bits

Fuente: Por el investigador

Panel frontal del instrumento virtual

La figura 6.63 muestra el panel frontal de la codificación de línea *Retorno a cero unipolar* (RZ_U) con el ingreso de los bits a codificar, de manera aleatoria y con la señal codificada en tiempo que cumple con la condición que al tener un **símbolo 1**, tomara **+V durante la mitad del periodo de bit** y cuando el **símbolo es 0**, tendrá **0V durante todo el periodo de bit**. Además se puede apreciar el espectro de frecuencia que esta relacionado con la señal codificada en tiempo. Estas señales presentan variaciones en sus características, dependiendo de la manipulación de los dos controles: frecuencia de muestreo (fs) y de la frecuencia de bit (fb). El rango de estos controles ya se encuentran definidas en el mismo control y que son las apropiadas para el correcto funcionamiento del presente VI.

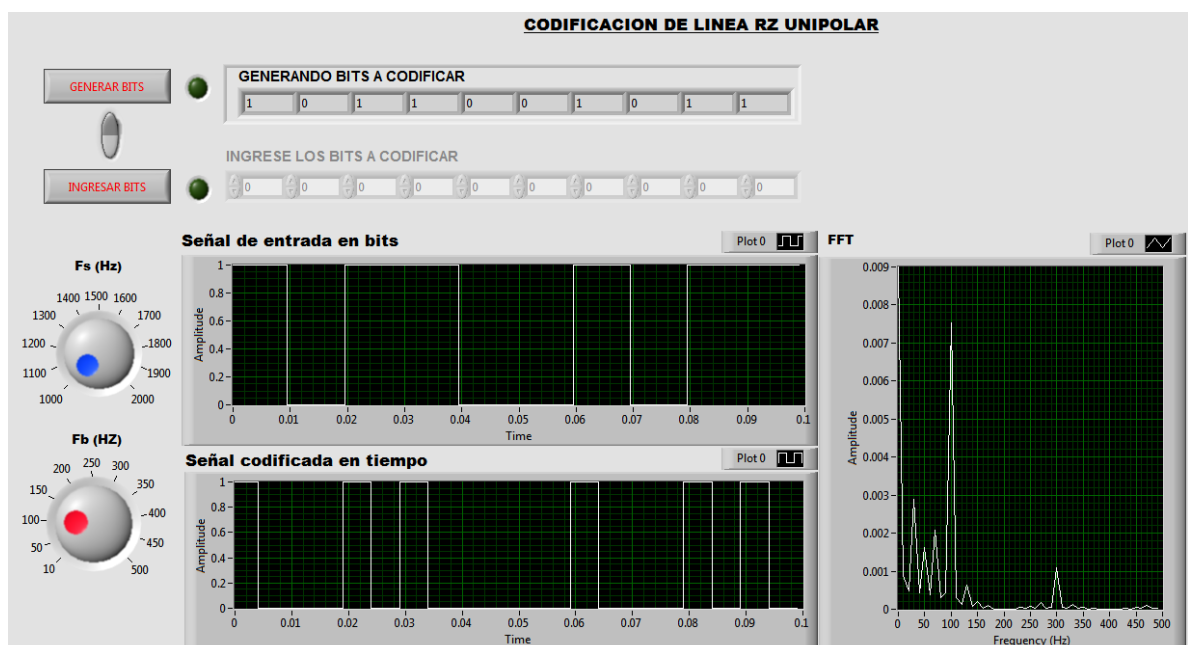


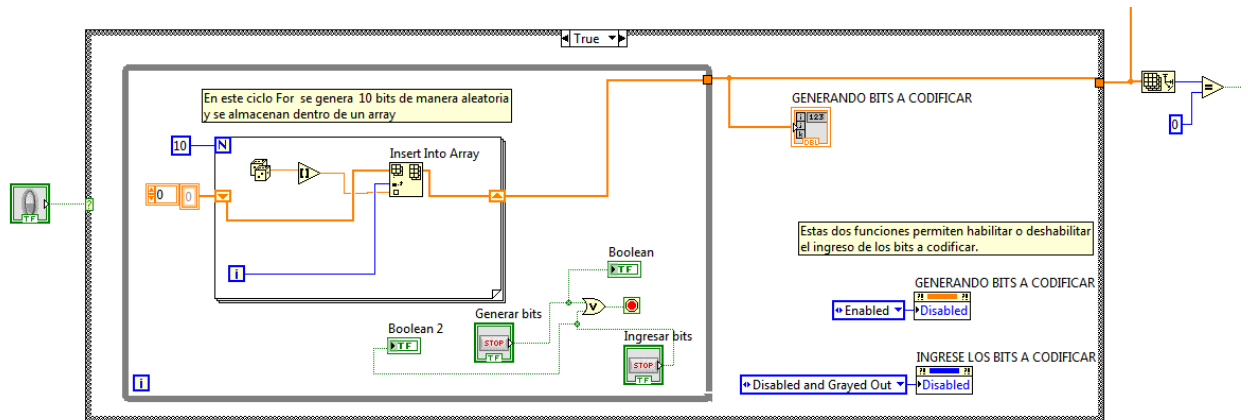
Figura 6.63: Panel Frontal del SubVI-RZ_U

Fuente: Por el investigador

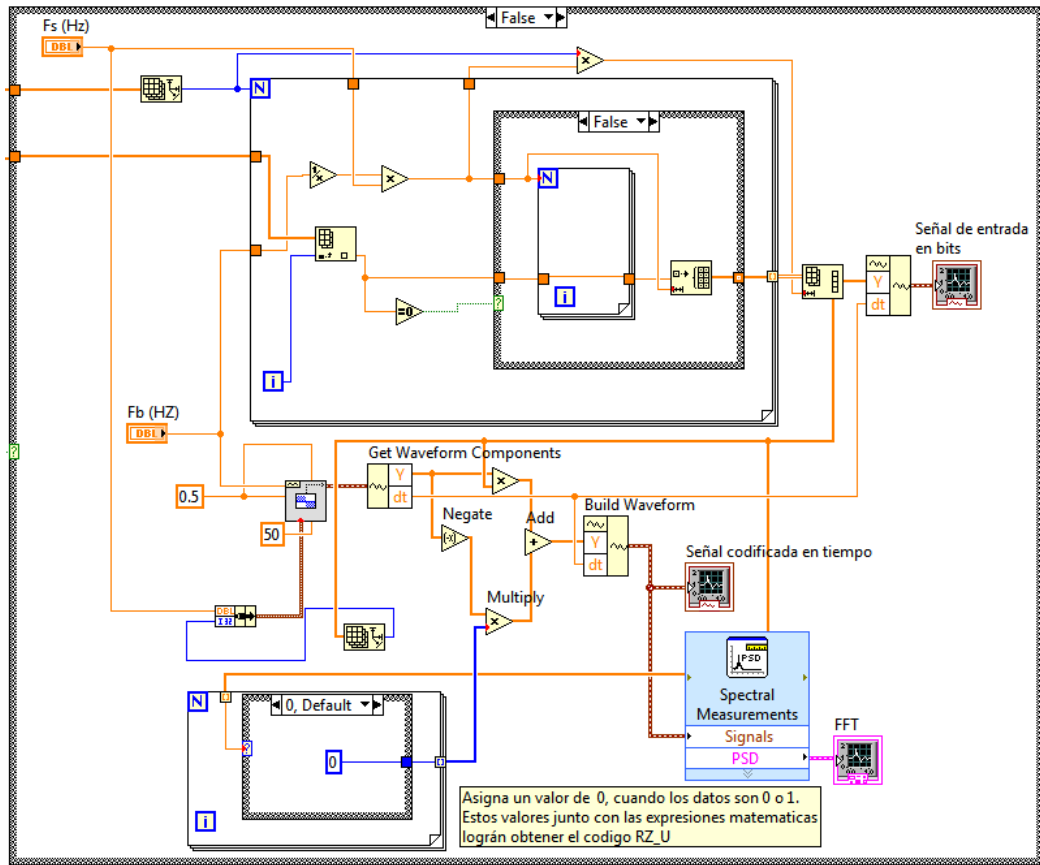
Diagrama de Bloques del instrumento virtual

En la figura 6.64 se muestra la programación por partes, de la codificación de línea *Retorno a cero unipolar* (RZ_U). Siendo la figura 6.64a, la programación enfocada al control del ingreso de los bits a codificar, ya sea el ingreso de estos bits de manera aleatoria o manual y la figura 6.64b muestra la programación de la codificación RZ_U. Ahora, el control encargado de seleccionar las dos maneras de ingresar los bits es el denominado *selector* que junto con dos controles adicionales

generar bits e ingresar bits, permite su activación de una de estas dos opciones de ingreso de bits.



(a)



(b) Código RZ_U

Figura 6.64: Diagrama de bloques RZ_U: (a) control del ingreso de bits a codificar; (b) codificación RZ_U

Fuente: Por el investigador

6.8.5 CODIGO DE LINEA RZ_B (Retorno a Cero Bipolar)

6.8.5.1 Descripción de elementos a ser utilizados en el código RZ BIPOLAR

En la simulación del Código RZ Bipolar, se han utilizado los mismos elementos para el diseño del panel frontal y el diagrama de bloques del código NRZ_U. Por las razones expuestas anteriormente, se obvia la explicación de los mismos en este presente diseño.

DISEÑO DEL VI PARA EL ESQUEMA DE CODIGO RZ_B

Para la codificación de línea *Retorno a cero bipolar*, se ha mantenido la misma programación del código NRZ_U. Sin embargo, el presente código tendrá una programación diferente para cumplir sus condiciones, por lo que se requiere seguir el siguiente procedimiento adicional.

PROCEDIMIENTO:

A continuación, se describe los pasos para diseñar el esquema de codificación de línea RZ_B teniendo en cuenta que algunos pasos ya no se los detallan por ser repetitivos a los anteriores códigos de línea programados.

1. Se inicializa un nuevo VI.
2. Creamos la misma programación del código NRZ_U es decir la etapa en donde se realiza el control de los bits de entrada a codificar y la etapa en la cual se asigna la frecuencia de muestreo (fs) y la frecuencia de bit (fb).
3. Las condiciones para este código *Retorno a cero bipolar* es: cuando en la señal de entrada se tiene un **símbolo 1**, tomara **+V durante la mitad de periodo de bit** y cuando el **símbolo es 0**, tendrá **-V durante la mitad de periodo de bit**.

Estas dos condiciones se lo realizan en la figura 6.65, en donde los valores de las *constantes numéricas* del *case structure* son diferentes al código *Retorno a cero unipolar (RZ_U)*.

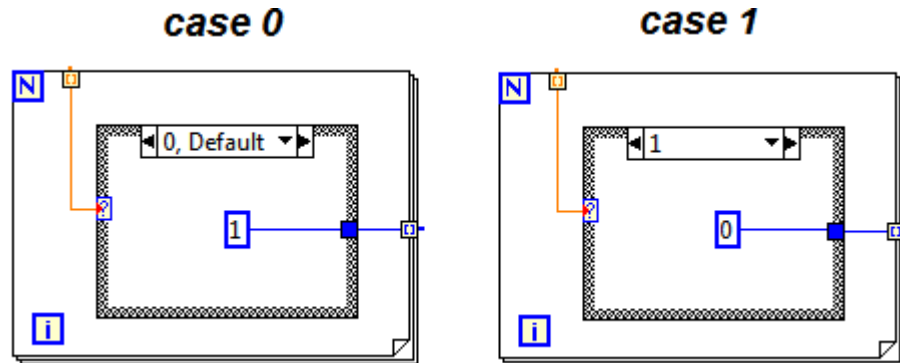


Figura 6.65: Condiciones del *Case Structure*
Fuente: Por el investigador

- Para cumplir con las condiciones del paso 2, creamos un *ciclo For*, en donde los datos de la señal de entrada bits serán comparados en un *Case structures*, para asignarles un valor de 1 o 0 para ser multiplicado por los datos negados del *Square Waveform* y sumados por el producto de la señal de entrada por la componente Y del *Get Waveform Components*, este paso se muestra en la figura 6.66.

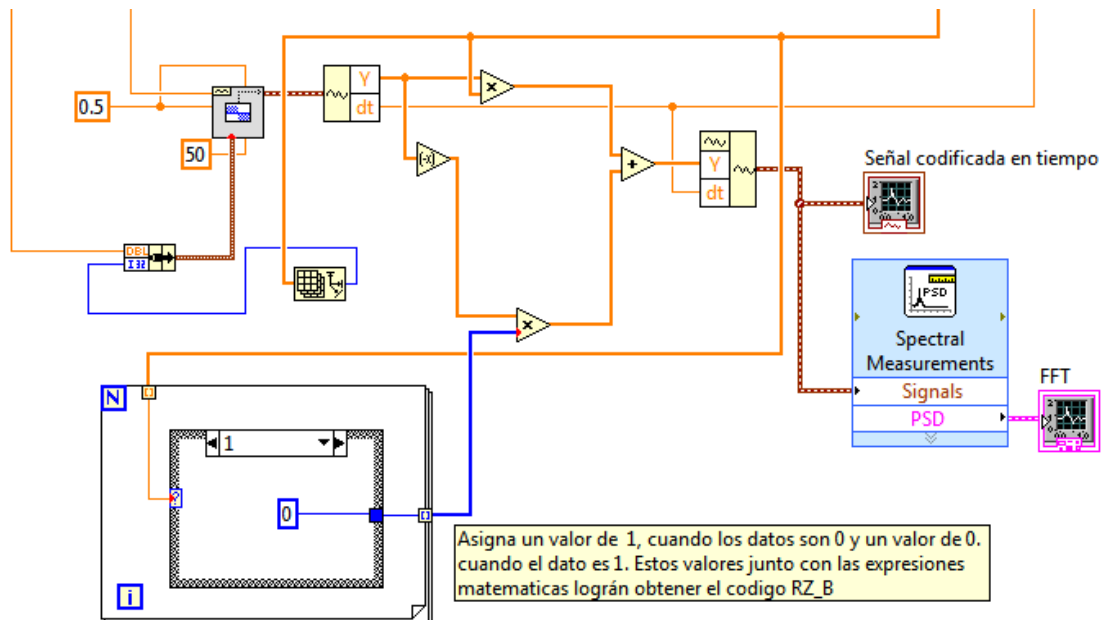


Figura 6.66: Código RZ_B
Fuente: Por el investigador

Panel frontal del instrumento virtual

La figura 6.67 muestra el panel frontal de la codificación de línea *Retorno a cero bipolar* (RZ_B) con el ingreso de los bits a codificar de manera manual. Su interfaz es el mismo descrito en el primer código de línea (NRZ_U), con la diferencia que el gráfico de la señal codificada en tiempo muestra una señal en donde el **símbolo 1**, tomara $+V$ durante la mitad de periodo de bit y cuando el **símbolo es 0**, tendrá $-V$ durante la mitad de periodo de bit. Estos símbolos se refieren a los valores que se tienen en la señal de entrada de bits.

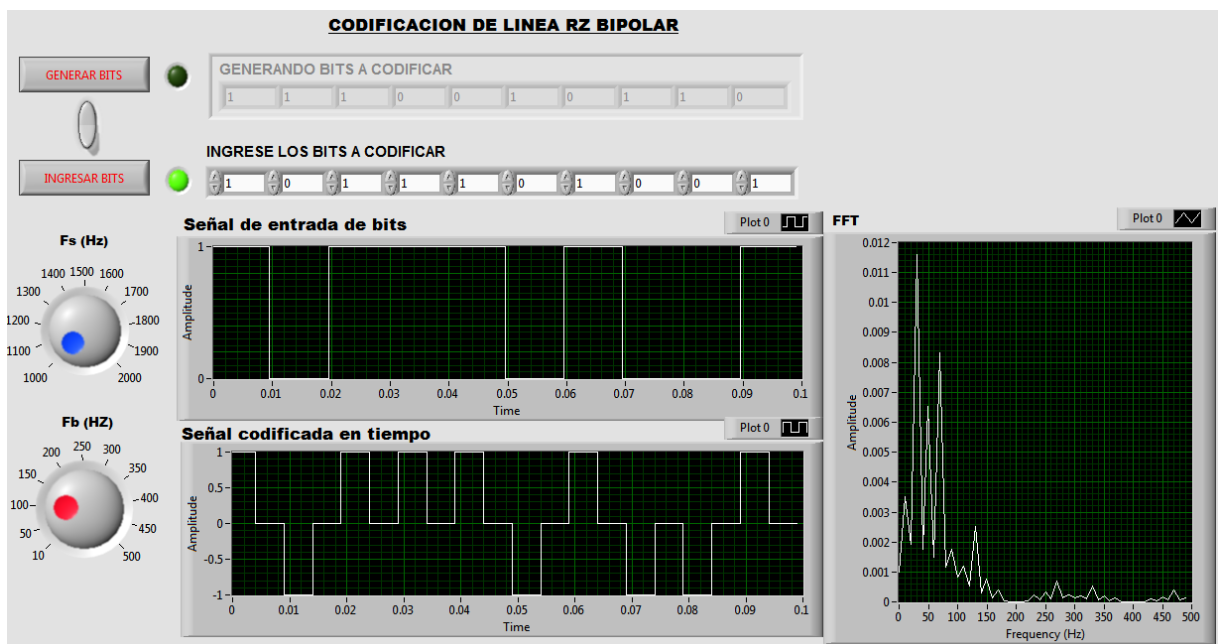
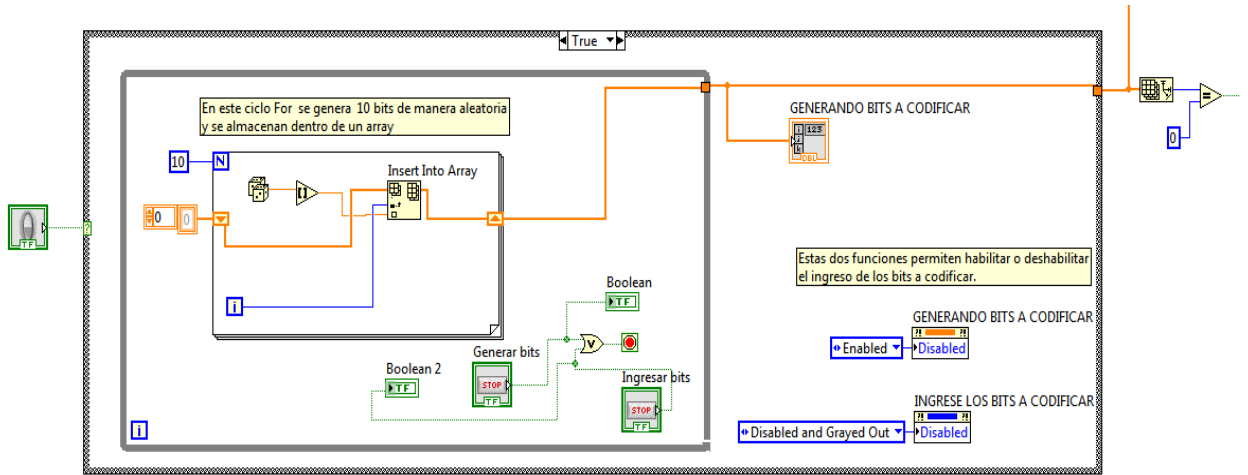


Figura 6.67: Panel Frontal SubVI-RZ_B

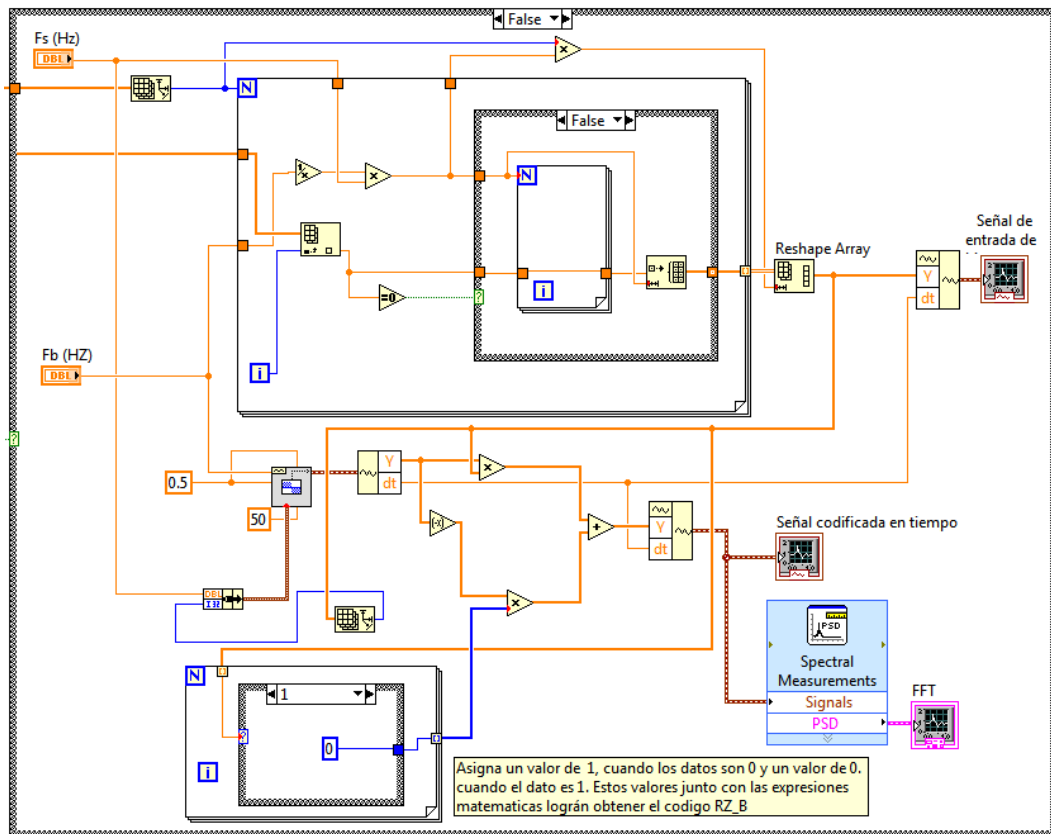
Fuente: Por el investigador

Diagrama de Bloques del instrumento virtual

A continuación, en la figura 6.68 muestra el diagrama de bloques de la codificación de línea *Retorno a cero bipolar* (RZ_B) por partes, con la finalidad de entender la programación realizada y mostrar las etapas que se han mantenido de las anteriores codificaciones de línea en especial del código NRZ_U.



(a)



(b)

Figura 6.68: Diagrama de bloques RZ_B: (a) control de ingreso de bits a codificar; (b) codificación RZ_B

Fuente: Por el investigador

6.8.6 CODIGO DE LINEA NRZ_AMI (No retorno a cero e inversión alternada de marca)

6.8.6.1 Descripción de elementos a ser utilizados en el código NRZ_AMI

En la simulación del código *No retorno a cero e inversión de marca* (NRZ_AMI), se han utilizado los mismos elementos para el diseño del panel frontal y el diagrama de bloques del código NRZ_U.

DISEÑO DEL VI PARA EL ESQUEMA DE CODIGO NRZ_AMI

Para el diseño de la codificación de línea *No Retorno a cero e inversión de marca*, se ha mantenido la misma programación del código NRZ_U, tanto para el ingreso de los bits a codificar y la etapa de asignación de las frecuencias (f_s , f_b), a los bits a codificar. Sin embargo, el presente código tendrá una programación diferente para cumplir sus condiciones, por lo que se requiere un procedimiento adicional.

PROCEDIMIENTO:

1. Se inicializa un nuevo VI.
2. Siguiendo la misma programación del código NRZ_U, se adiciona una nueva etapa para cumplir con la codificación de línea NRZ_AMI. Esta etapa o proceso se lo realiza con el uso de registros de desplazamientos (*shift register*) que se crea en los bordes del ciclo *For*, con solo dar un clic derecho en el mismo. El propósito de estos registros es almacenar los bits anteriores para luego compararlos con estructuras condicionales (*case structure*) y así cumplir las condiciones del código *No Retorno a cero e inversión de marca* (NRZ_AMI).
3. La condición que se necesita llevar a cabo cuando los bits a codificar son unos, es tomar valores alternados de +V o -V, mientras que a los 0 lógicos se los representa por 0V. El siguiente paso muestra la programación de estas condiciones.
4. Todas las comparaciones se lo realizaron con los *cases structures* ubicado en *panel de funciones/structures*. La función que elige que *case* a seguir de acuerdo a los bits a codificar, lo realiza la función *comparison* ubicado en la *paleta de funciones/ equal?*. Dentro de cada *case*, se encuentran constantes numéricas que tienen valores definidos para crear la codificación NRZ_AMI. Los registros de desplazamiento creados en los bordes del ciclo *For* y las

funciones de array, también ayudaron a manipular los bits a codificar en forma individual (*Index Array*) o creando elementos de array con valores de 0 (*Initialize Array*).

5. Los procedimientos descritos anteriormente se encuentran de manera grafica en la sección diagrama de bloques del instrumento virtual que se lo analiza después de presentar el panel frontal de este código NRZ_AMI.

Panel frontal del instrumento virtual

La figura 6.69 muestra el panel frontal de la codificación de línea *No Retorno a cero e inversión de marca* (NRZ_AMI) con el ingreso de los bits a codificar de manera manual. Su interfaz es el mismo descrito en el primer código de línea (NRZ_U). Sin embargo, la señal codificada en tiempo muestra valores alternados de $+V$ o $-V$ cuando los bits a codificar son **unos** y **0V** a los **0 lógicos**. Es importante también variar los dos controles de perilla que se encuentran a la izquierda de las ventanas señal de entrada de bits y señal codificada en tiempo. Estos controles: f_s es la que varía la frecuencia de muestreo de la señal de entrada de bits y f_b es la que modifica la frecuencia de los bits a codificar. De esta forma las señales se podrán visualizar con mejores resultados.

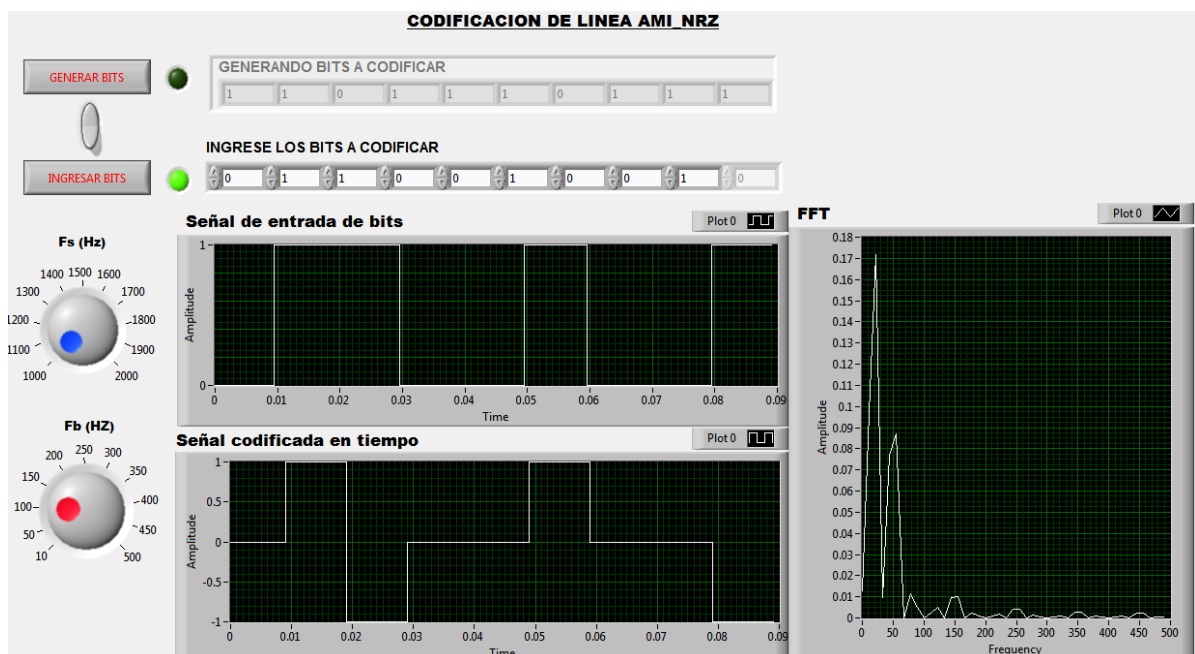
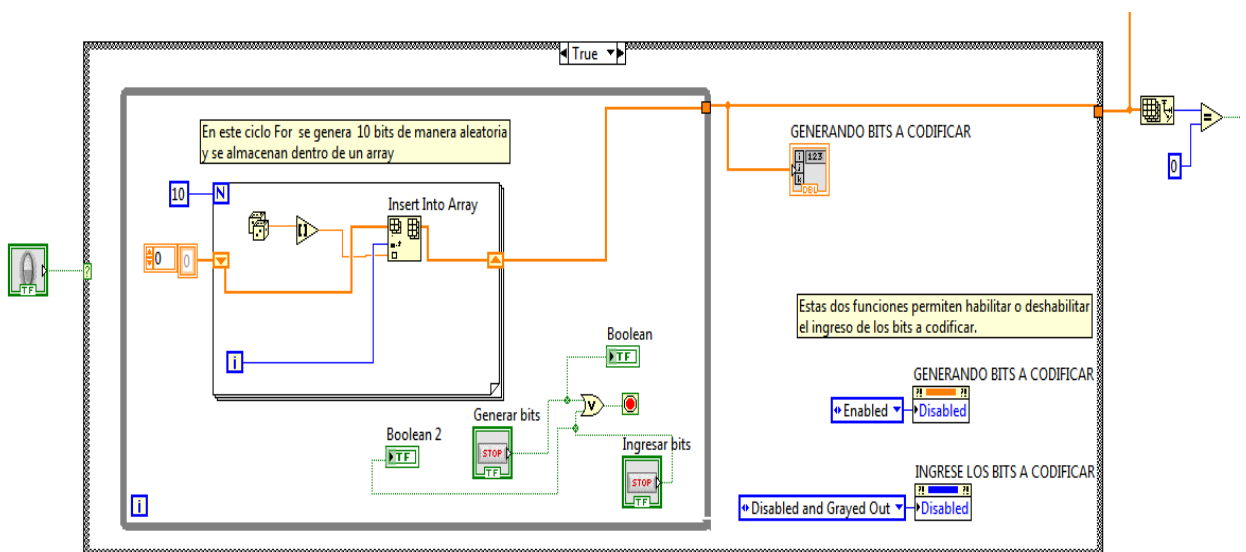


Figura 6.69: Panel frontal de SubVI-NRZ_AMI
Fuente: Por el investigador

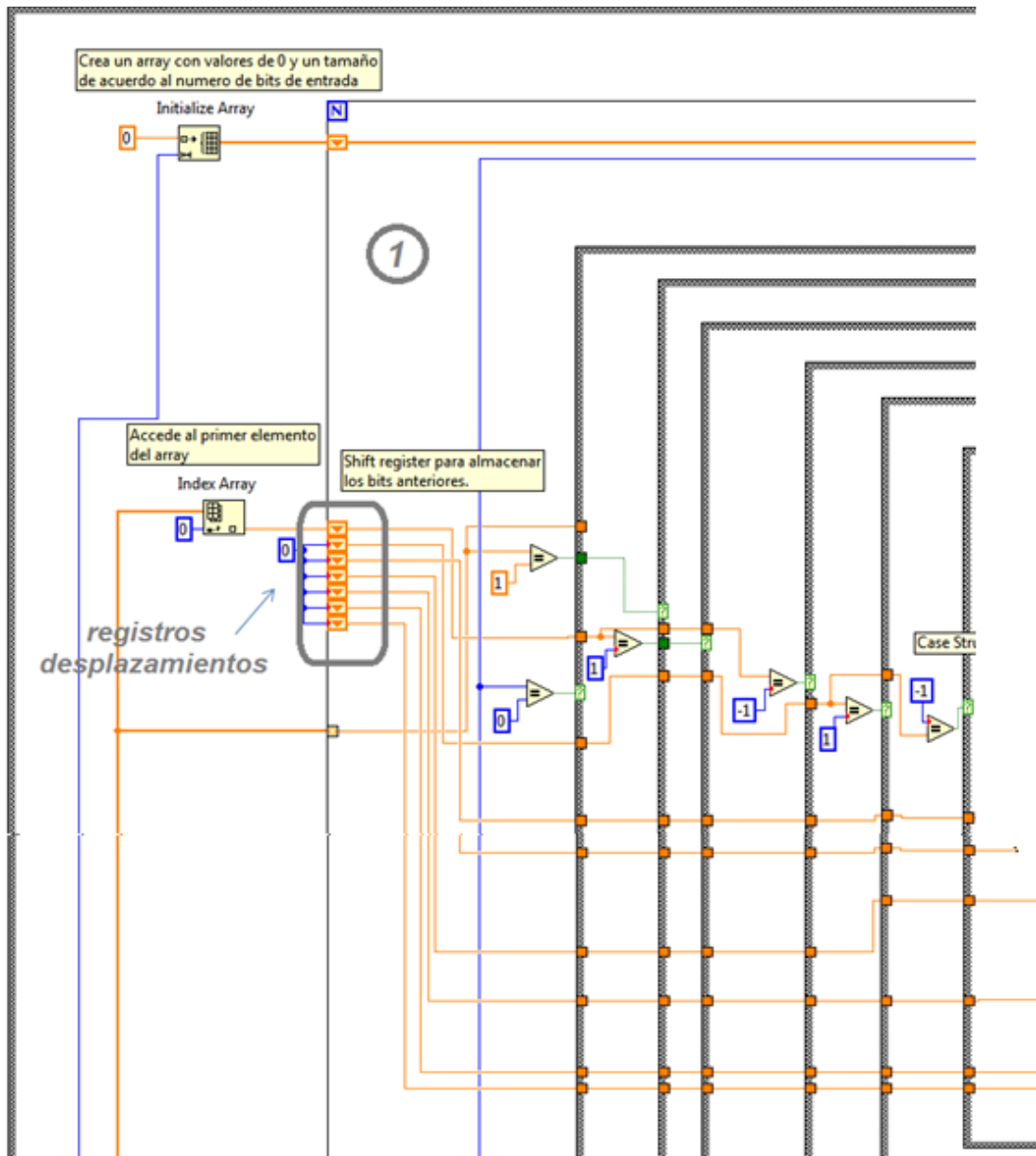
Diagrama de Bloques del instrumento virtual

A continuación, en la figura 6.70 muestra el diagrama de bloques de la codificación de línea *No Retorno a cero e inversión de marca* (NRZ_AMI) por partes, con la finalidad de entender la programación realizada. La figura 6.70a muestra la misma programación del control de ingreso de los bits a codificar, de los anteriores códigos de línea diseñados. Por otro lado, en la figura 6.70b se muestra una parte de la programación de la codificación NRZ_AMI, que esta conformada por varios *elementos* de la paleta de funciones y se usan los registros de desplazamiento para evaluar los bits de ingreso anteriores y compararlas con los bits actuales de tal manera de cumplir con la condición que presenta esta codificación de línea *No Retorno a cero e inversión de marca*.

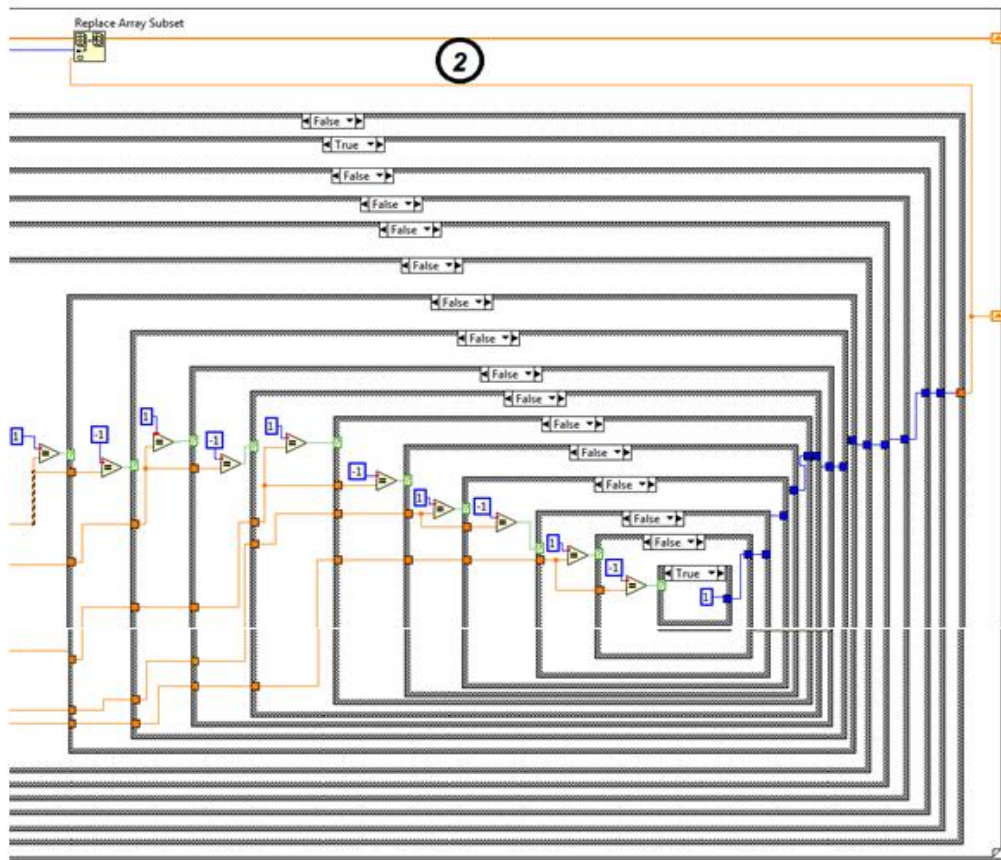
La figura 6.70c muestra el resto de *case structure* con sus respectivas funciones *comparison* encargadas de comparar cada bit a codificar con las funciones denominadas *constant numeric*. Cada una de estas funciones retorna un valor de *true* o *false* a cada estructura *Case*. Una vez que los bits ingresados lleguen al ultimo *case* se pueden ya obtener los bits codificados que tendrán relacionarlos con las frecuencias de muestreo y de bits, para finalmente crear la señal codificada en tiempo como se muestra en la figura 6.70d. En cambio la figura 6.70e muestra la programación relacionada a la obtención de la señal de entrada de bits y el espectro de frecuencia, que sin duda es la misma utilizada en los códigos que se han venido desarrollando anteriormente. En esta misma figura se puede observar que se encuentra señalada con una flecha una parte esencial que permite sincronizar el tiempo (*dt*) de las señales que se visualizan.



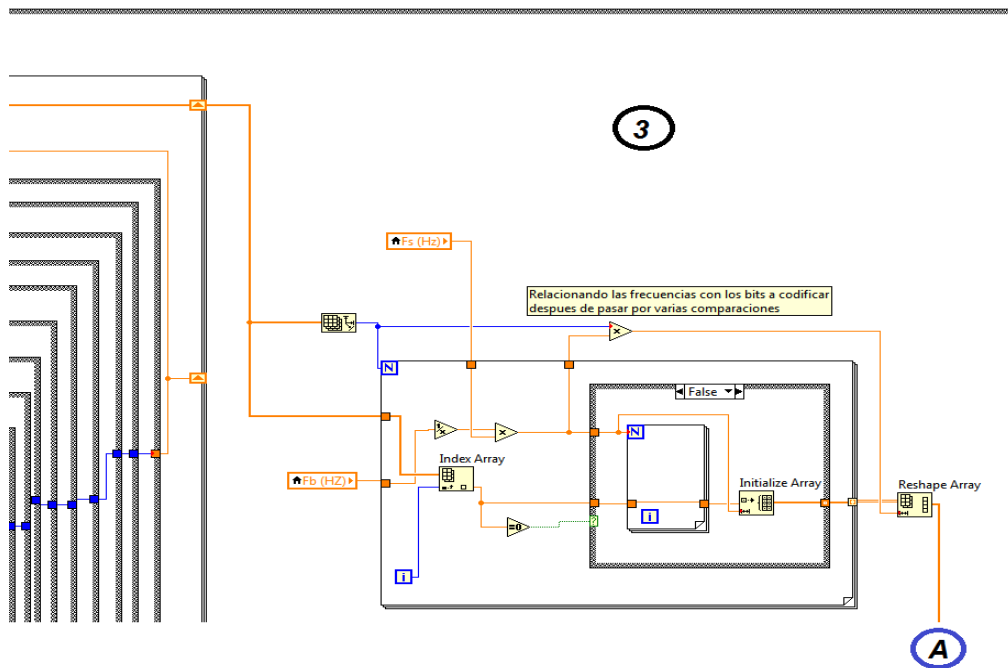
(a)



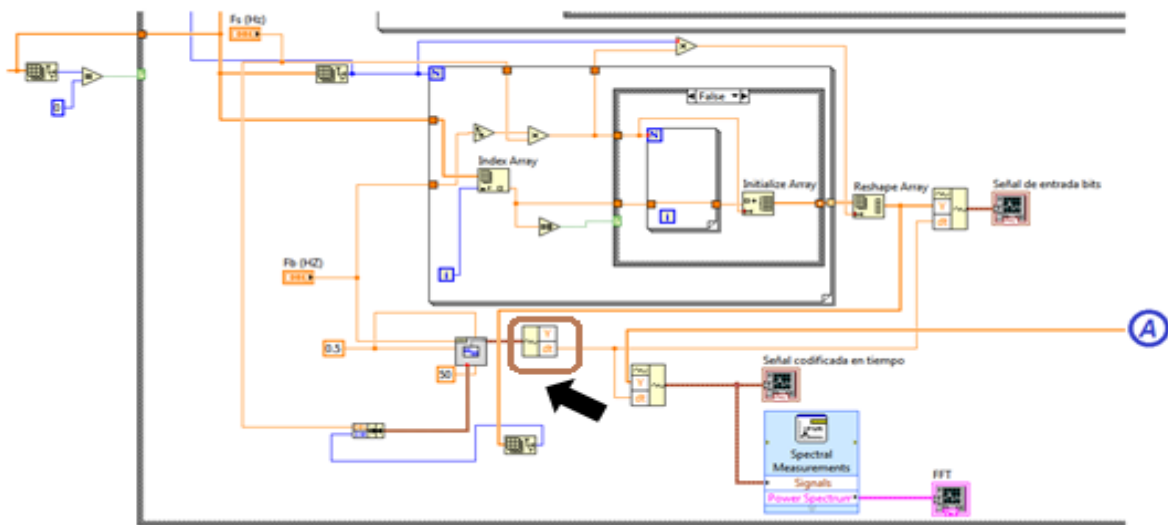
(b)



(c)



(d)



(e)

Figura 6.70: Diagrama de bloque del SubVI-NRZ_AMI: (a) control de entrada de bits a codificar; (b) registros de desplazamiento; (c) codificación NRZ_AMI; (d) relacionando las frecuencias con los bits codificados; (e) señal de entrada de bits y espectro de frecuencia.

Fuente: Por el investigador

6.8.7 CODIGO DE LINEA RZ_ AMI (Retorno a cero e inversión alternada de marca)

6.8.7.1 Descripción de elementos a ser utilizados en el código RZ_ AMI

En la simulación del código *Retorno a cero e inversión de marca* (RZ_ AMI), se han utilizado los mismos elementos para el diseño del panel frontal y el diagrama de bloques del código NRZ_ U

DISEÑO DEL VI PARA EL ESQUEMA DE CODIGO RZ_ AMI

Para el diseño de la codificación de línea *Retorno a cero e inversión de marca* (RZ_ AMI), se ha mantenido la misma programación del código NRZ_ AMI. Sin embargo, el presente código tiene que cumplir cierta condición que le hace diferente a las anteriores codificaciones de línea.

PROCEDIMIENTO:

1. Se inicializa un nuevo VI.

- Con la misma programación de la codificación de línea anterior NRZ_AMI, solo queda realizar una modificación para obtener la presente codificación. La condición para el código RZ_AMI, es representar a los **unos lógicos** con valores alternados de **+V** y **-V con retorno a cero en la mitad del periodo de bit**. Un **0 lógico** se representa con un **nivel de 0V**.
- Para llevar a cabo esta nueva codificación de línea (RZ_AMI), se ubica en la etapa de obtención de la señal de codificada en tiempo y del espectro de frecuencia del código NRZ_AMI, ver figura 6.71. Si analiza la programación notara que estas señales utilizan solo, la componente *dt* (variación de tiempo) del *Get waveform components* para mantener los mismos intervalos de tiempo, dejando sin conexión a la componente *Y*, debido a que no es necesario para la codificación anterior (NRZ_AMI).

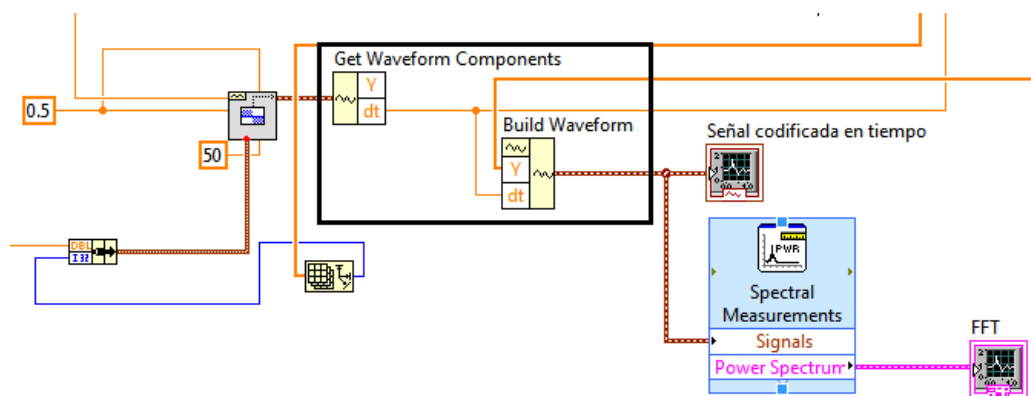


Figura 6.71: Etapa de obtención de la señal codificada en tiempo y el espectro de frecuencia del código NRZ_AMI

Fuente: Por el investigador

- En el paso 3, es donde hay que realizar un cambio para la obtener la codificación *Retorno a cero e inversión de marca* (RZ_AMI) y eso se muestra en la figura 6.72. Se ha utilizado una expresión matemática *Multiply* en la cual una de sus entradas proviene de la etapa de asignación de las frecuencias *fs*, *fb* luego de pasar por varios *Cases Structures* y la otra entrada viene del *Get Waveform Components*. El resultado de esta función *Multiply* es obtener la multiplicación de los valores que ingresan en sus entradas. Para el caso nuestro, necesitamos para multiplicar los bits codificados NRZ_AMI por los valores del *Square Waveform* (genera una señal cuadrada de acuerdo a sus parámetros de entrada) para así, obtener finalmente la codificación *Retorno a cero e inversión de marca* (RZ_AMI).

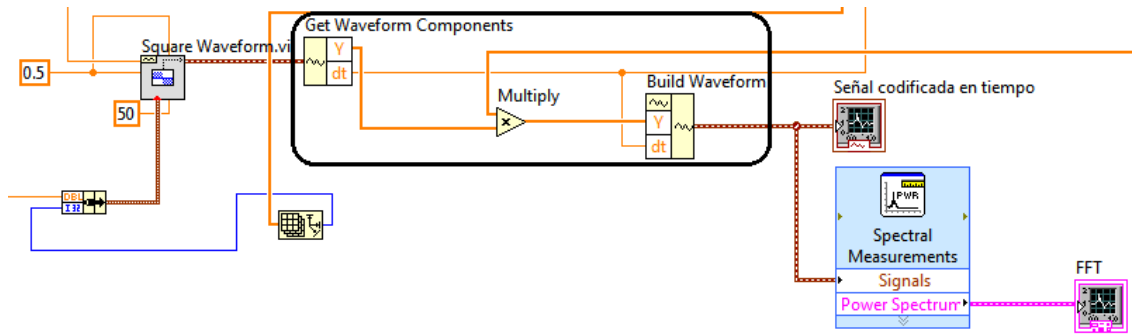


Figura 6.72: Etapa de obtención de la señal codificada en tiempo y el espectro de frecuencia del código RZ_AMI

Fuente: Por el investigador

Panel frontal del instrumento virtual

La figura 6.73 muestra el panel frontal de la codificación de línea *Retorno a cero e inversión de marca* (RZ_AMI) con el ingreso de los bits a codificar de manera manual, el número de bits que puede ingresarse es de diez bits. Su interfaz es el mismo descrito en el primer código de línea (NRZ_U), con la diferencia que el gráfico de la señal codificada en tiempo representa a los **unos lógicos** con valores alternados de $+V$ y $-V$ con **retorno a cero en la mitad del periodo de bit**. Un **0 lógico** se representa con un **nivel de $0V$** .

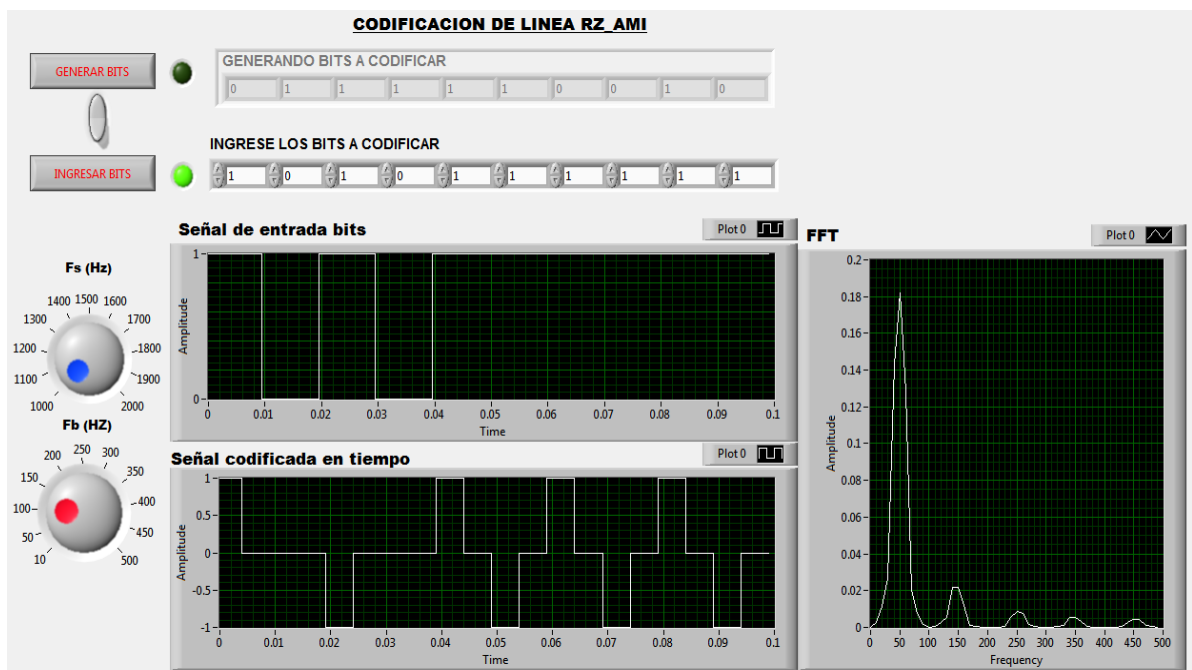


Figura 6.73: Panel frontal de SubVI-RZ_AMI

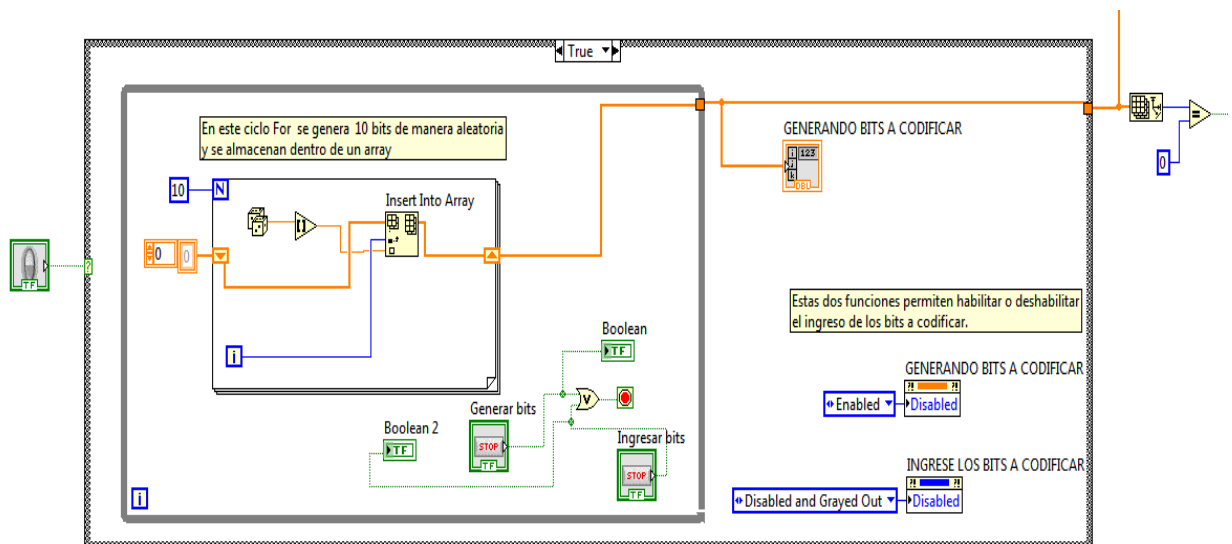
Fuente: Por el investigador

Diagrama de Bloques del instrumento virtual

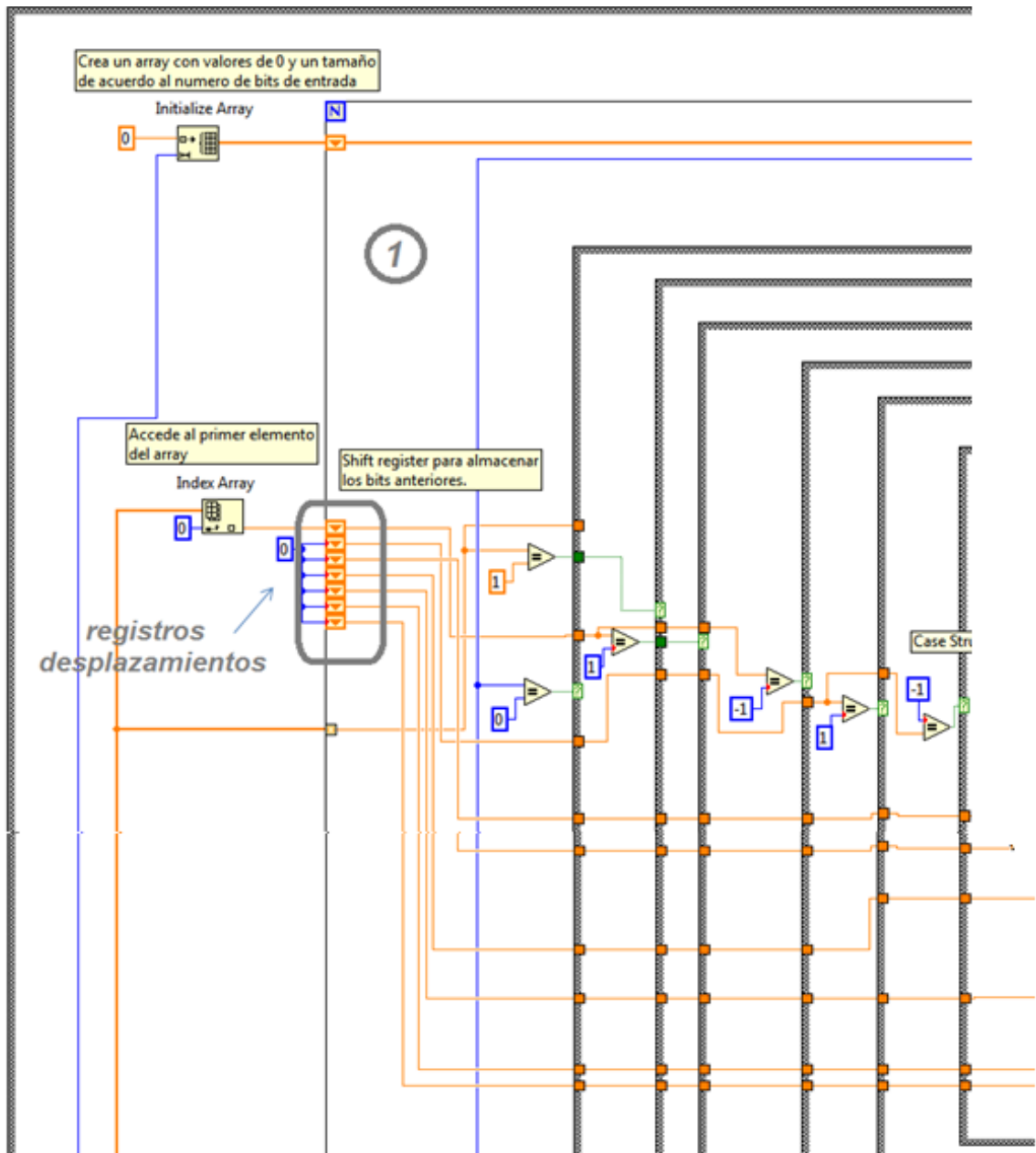
A continuación, en la figura 6.74 muestra el diagrama de bloques de la codificación de línea *Retorno a cero e inversión de marca* (RZ_AMI) por partes, con la finalidad de entender la programación realizada. Su explicación se lo realiza rápidamente por ser una codificación similar al código NRZ_AMI y que solo cambiamos en una sola etapa.

La figura 6.70a muestra la misma programación del control de ingreso de los bits a codificar, de los anteriores códigos de línea diseñados. Por otro lado, en la figura 6.70b muestra una parte de la programación de la codificación NRZ_AMI, que esta conformada por varios *elementos* de la paleta de funciones y se usan los *registros de desplazamiento* para evaluar los bits de ingreso anteriores y compararlas con los bits actuales de tal manera de cumplir con la condición que presenta esta codificación de línea *Retorno a cero e inversión de marca*.

La figura 6.70c muestra el resto de *case structure* con sus respectivas funciones *comparison* encargadas de comparar cada bit a codificar con las funciones denominadas *constant numeric*. Cada una de estas funciones retorna un valor de *true* o *false* a cada estructura *Case*. Una vez que los bits ingresados lleguen al ultimo *case* se pueden ya obtener los bits codificados que tendrán relacionarlos con las frecuencias de muestreo y de bits, para finalmente crear la señal codificada en tiempo como se muestra en la figura 6.70d. En cambio la figura 6.70e muestra la programación relacionada a la obtención de la señal codificada en tiempo y el espectro de frecuencia, que sin duda es la etapa en donde se modificó para obtener la codificación *Retorno a cero e inversión de marca*.

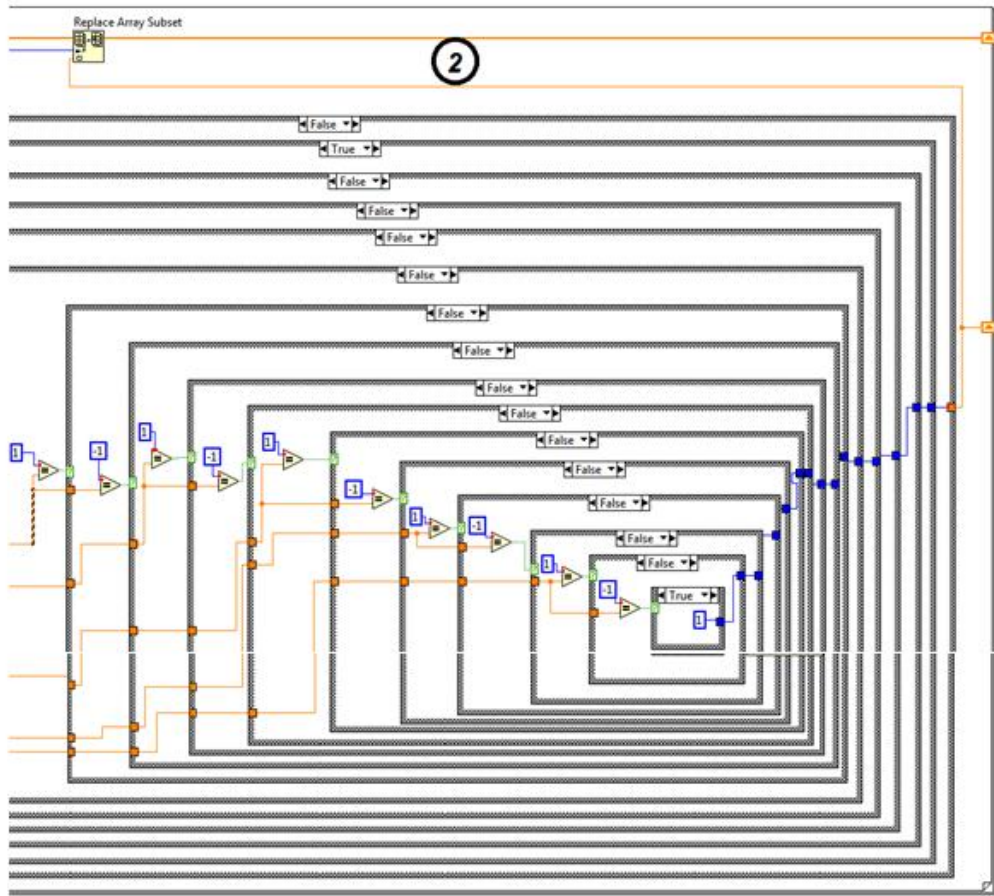


(a)

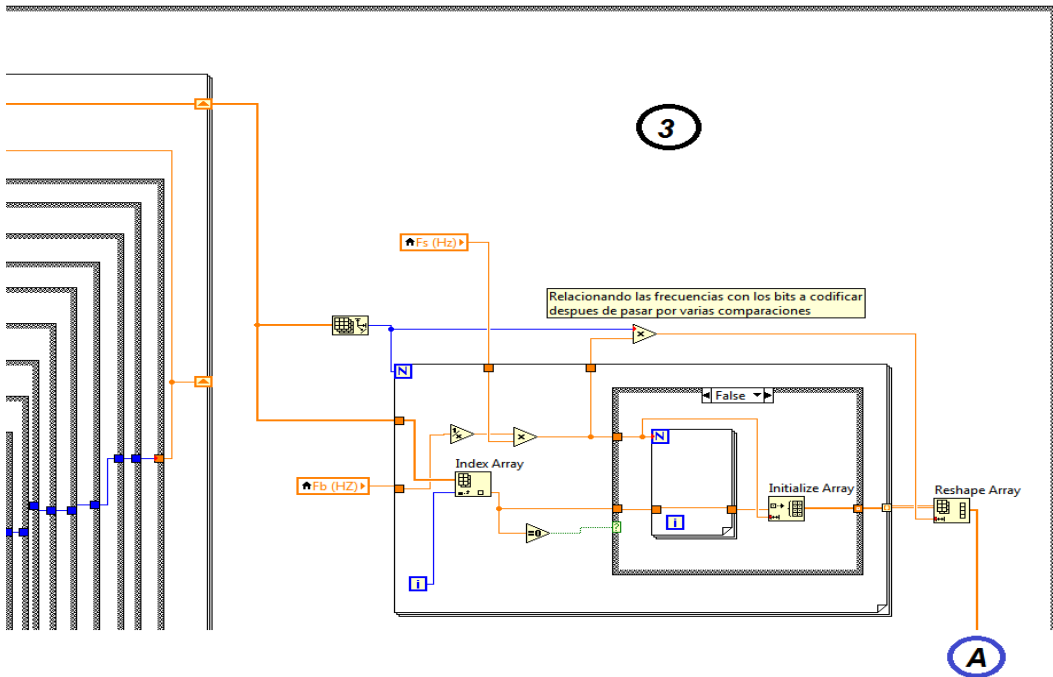


(b)

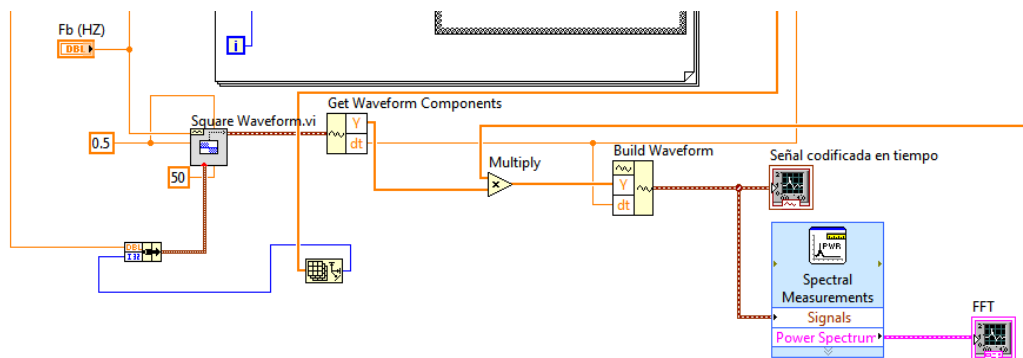
En el primer *case structure* (*case true*) los bits a codificar ingresan directamente, en el segundo *case structure* (*case false*) se crea una constante de 0, para el tercero *case structure* (*case true*) creamos una constante de -1 y para los siguientes *cases structures*, se tiene que ir alternando los valores de las constantes numéricas a crear con estos valores de: 1, -1, 1, -1, ..., 1.



(c)



(d)



(e)

Figura 6. 74: Diagrama de bloque del SubVI-RZ_AMI: (a) control de entrada de bits a codificar; (b) registros de desplazamiento; (c) codificación RZ_AMI; (d) relacionando las frecuencias con los bits codificados; (e) señal de entrada de bits y espectro de frecuencia.

Fuente: Por el investigador

6.8.8 CODIGO DE LINEA MANCHESTER

6.8.8.1 Descripción de elementos a ser utilizados en el código MANCHESTER

En la simulación del código *Manchester*, se han utilizado los mismos elementos para el diseño del panel frontal y el diagrama de bloques del código NRZ_U.

DISEÑO DEL VI PARA EL ESQUEMA DE CODIGO MANCHESTER

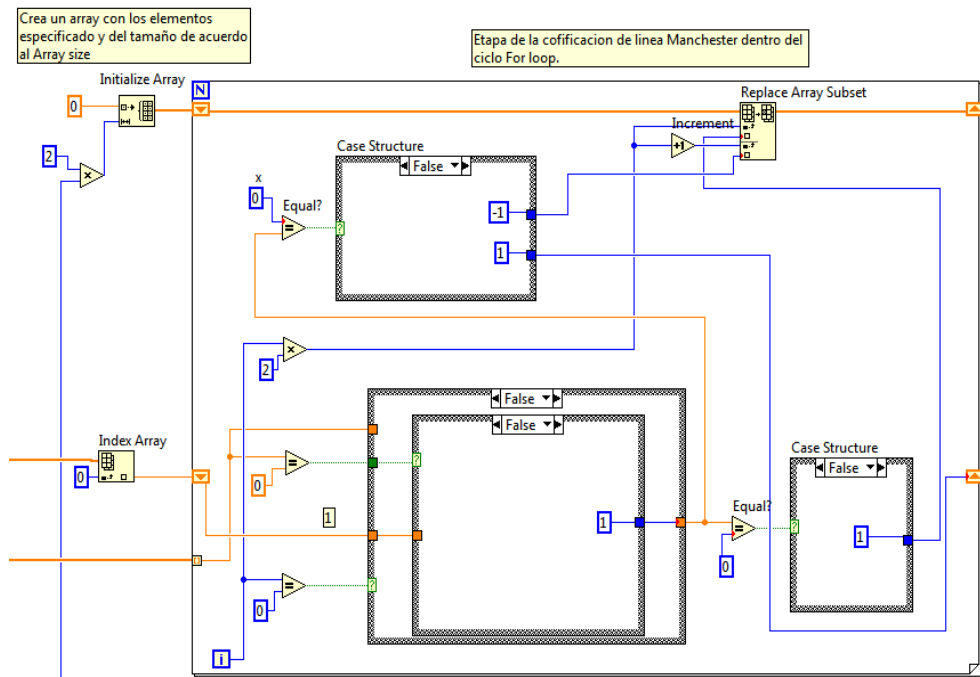
Para el diseño de la codificación de línea *Manchester*, se ha mantenido algunas etapas del código NRZ_AMI. Sin embargo, el presente código tendrá una nueva etapa de codificación que cumplirá los requisitos de un código *Manchester* con el siguiente procedimiento.

PROCEDIMIENTO:

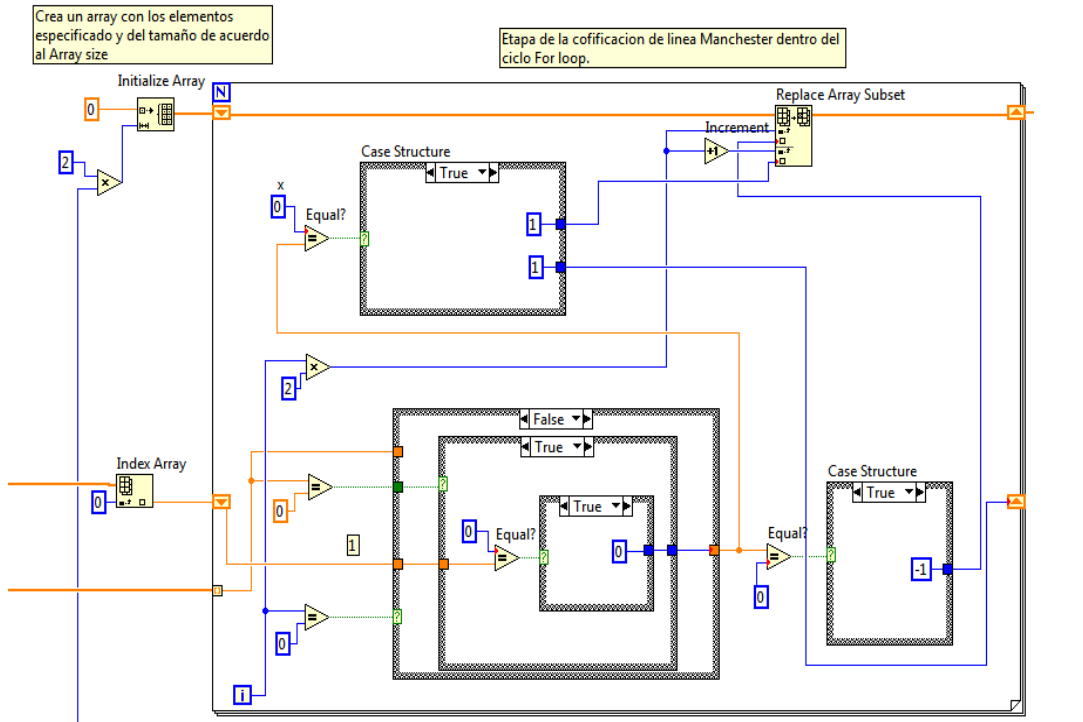
1. Se inicializa un nuevo VI.
2. Procedemos a mantener todas las etapas del código NRZ_AMI, a excepción de la etapa en donde se realiza la codificación. Por la razón que la etapa de codificación Manchester debe cumplir con la condición: de representar al **1 lógico** con un nivel de **+V durante la mitad de periodo de bit** y un nivel de **-V**

para la otra mitad, en cambio para **0 lógico** toma un **-V** durante la primera mitad de periodo de bit y con **+V** durante la segunda mitad.

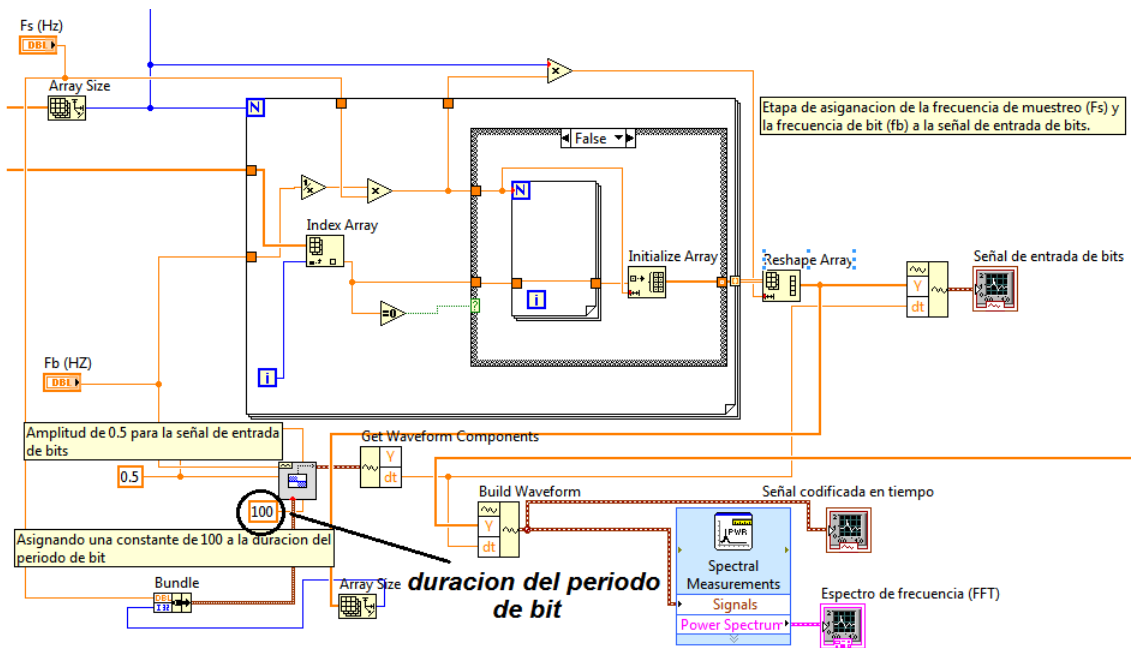
- La figura 6.75 muestra un Ciclo For que contiene en sus bordes dos *shift register*. En el *shift register* superior del ciclo for, se ingresan 20 elementos con valores de cero del Initialize Array, estos elementos provienen del producto (Multiply) del tamaño de los bits a codificar por una constante numérica (constant numeric) de 2. Mientras que en el *shift register* del borde inferior del ciclo for, se cuenta con la función Index Array que permite obtener un elemento de la posición especificado en su terminal index, en este caso necesitamos el primer elemento de los bits a codificar por lo que, se crea una constante numérica de 0. Luego mediante funciones de comparación ubicados en la *paleta de funciones/comparison/ Equal?* se los compara con constantes numéricas definidas que permitan seleccionar el case correcto de la estructura Case Structure. La función *Replace Array subset*, es una función que crea un nuevo array con los elementos que se ingresen a sus terminales *new element* con su respectivo index o posición. Al terminar la ejecución del ciclo For, se tendrá los bits codificados en Manchester. La figura 6.75a muestra la etapa de codificación Manchester con los cases structures con su condición false. En cambio, la figura 6.75b muestra los cases structures con su respectiva condición true. En la figura 6.75c se muestra un último proceso, que es asignar en el *Square Waveform* un valor de 100 en el terminal *duty cycle* para el tiempo de duración del periodo de bit, de la señal de entrada de bits y para la señal codificada en tiempo.



(a)



(b)



(c)

Figura 6.75: Codificación de línea Manchester: (a) casos structures en false; (b) casos structures en true; (c) duración del periodo de bits de las señales.

Fuente: Por el investigador

Panel frontal del instrumento virtual

La figura 6.76 muestra el panel frontal de la codificación de línea *Manchester* con el ingreso de los bits en forma aleatoria. Su interfaz gráfica, es el mismo del código de línea (NRZ_U) con la diferencia que la señal codificada en tiempo muestra al **1 lógico** con un nivel de **+V durante la mitad de periodo de bit** y un nivel de **-V para la otra mitad**, en cambio para **0 lógico**, toma un **-V durante la primera mitad de periodo de bit** y con **+V durante la segunda mitad**.

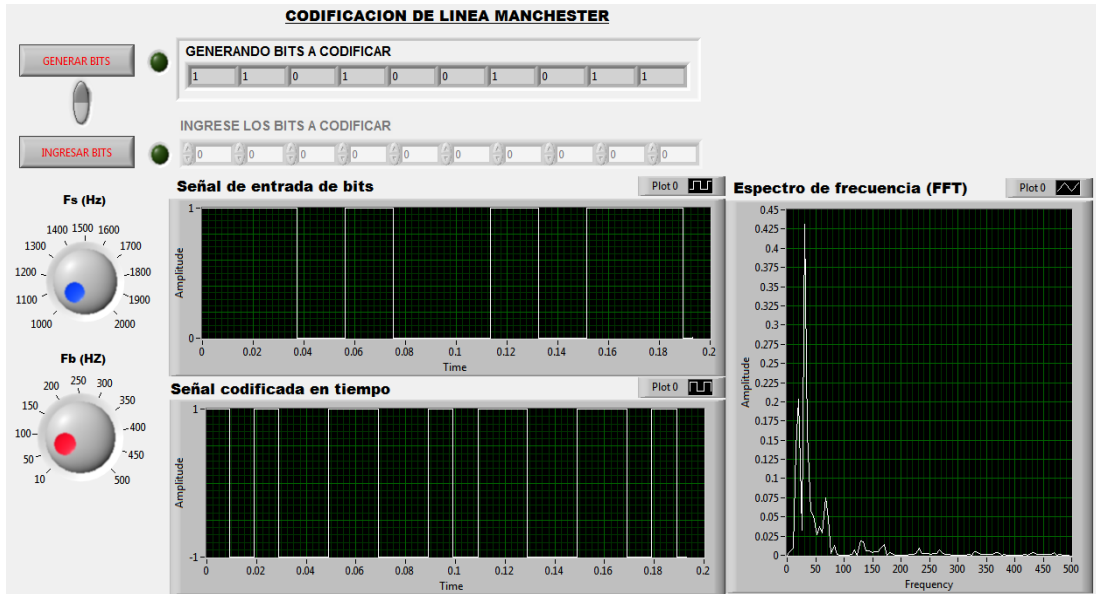
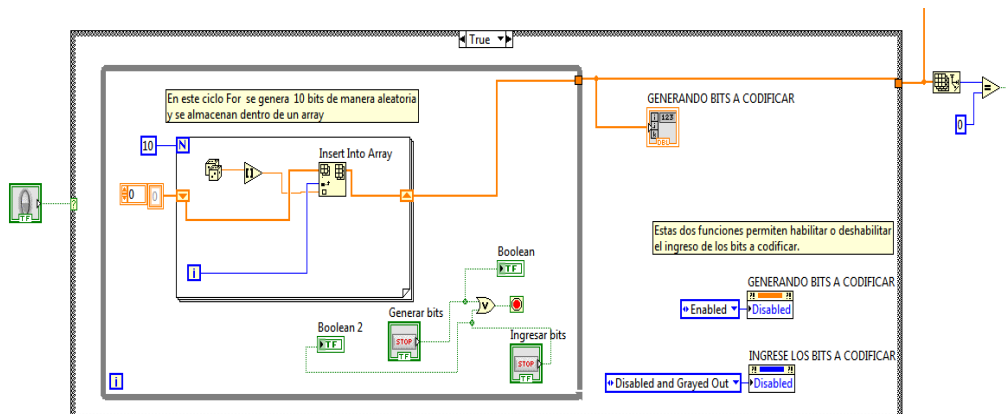


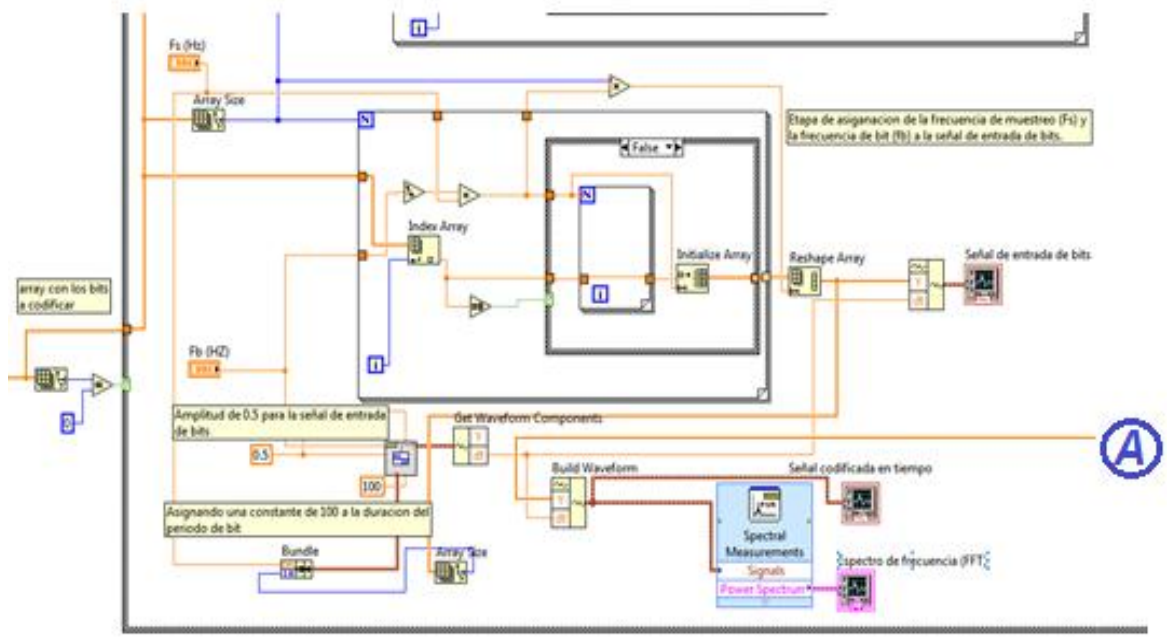
Figura 6. 76: Panel frontal de SubVI- Manchester
Fuente: Por el investigador

Diagrama de Bloques del instrumento virtual

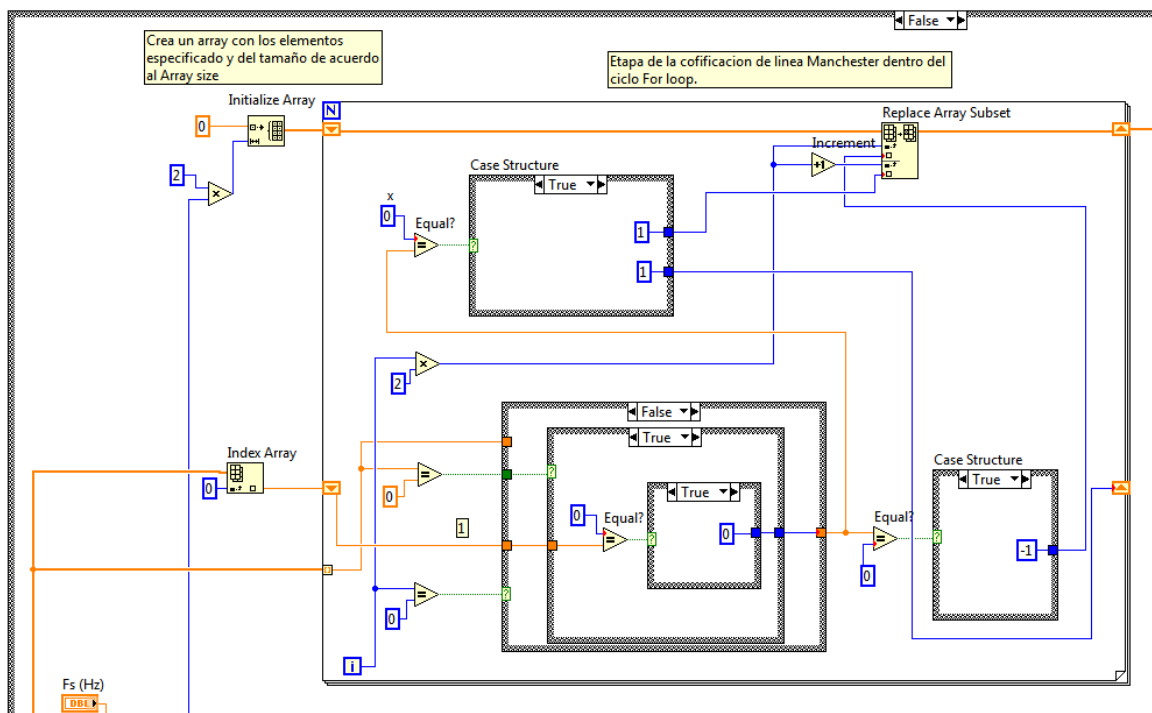
A continuación, en la figura 6.77 muestra el diagrama de bloques de la codificación de línea *Manchester* por partes, para su mejor entendimiento.



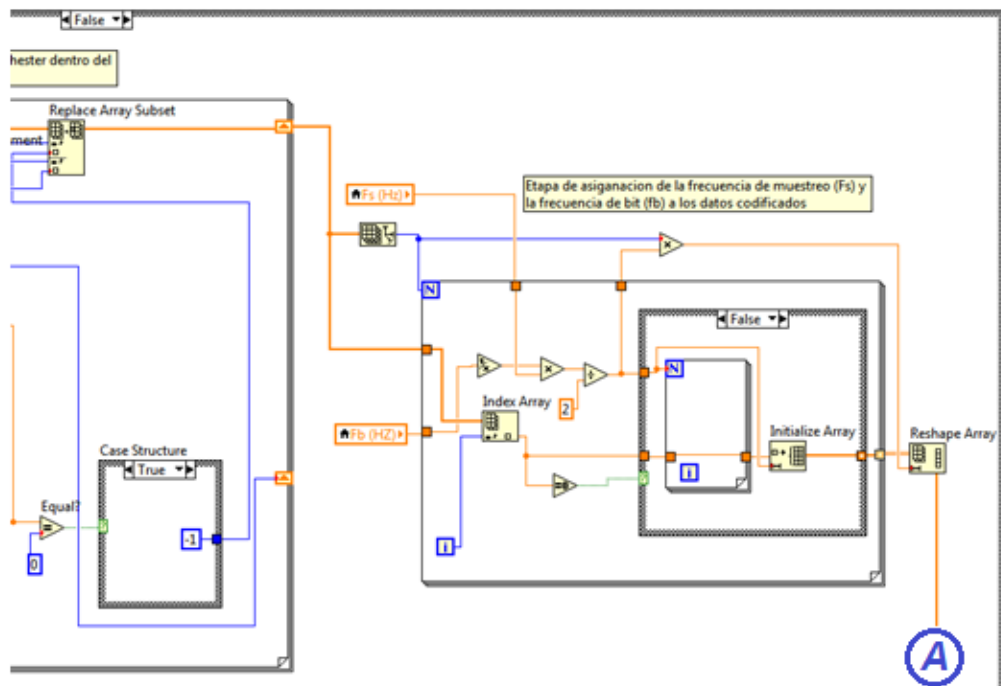
(a)



(b)



(c)



(d)

Figura 6. 77: Diagrama de bloque Manchester: (a) control del ingreso de los bits a codificar; (b) generación de la señal de entrada, codificada y el espectro de frecuencia; (c) creación de la codificación Manchester ; (d) relación de las frecuencias de muestreo y de bits con los bits codificados.

Fuente: Por el investigador

6.8.9 MODULACION Y DEMODULACION ASK

6.8.9.1 Descripción de los elementos a ser utilizados en la realización de la modulación y demodulación digital ASK.

Algunos de los elementos que se utilizaron para las codificaciones de línea son también utilizados en la simulación de la modulación y demodulación ASK.

PANEL FRONTAL DEL INSTRUMENTO VIRTUAL

Para visualizar el diagrama de constelación se ha utilizado, el indicador grafico *XY Graph*, que se describe en la figura 6.78.



PESTAÑA DE CONTROLES>>BOOLEAN>>PUSH BUTTON	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
<p>Botón que permite controlar acción de verdadero o falso de modo de activar o desactivar acciones de tipo booleano.</p>	

Figura 6.78: Botón Booleano
Fuente: Por el investigador

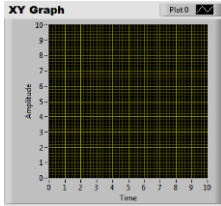
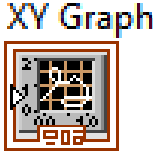
PESTAÑA DE CONTROLES>>GRAPH>>XY GRAPH	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
<p>Esta ventana se utiliza para visualizar cualquier grafico proveniente de un arreglo de dimensión 2, es decir de valores X y Y.</p>	

Figura 6.79: Gráfico de Forma de Onda XY
Fuente: Por el investigador

DIAGRAMA DE BLOQUES DEL INSTRUMENTO VIRTUAL

A continuación describimos de manera resumida las funciones utilizadas en la modulación-demodulación ASK. Para obtener más información de los terminales de cada función a utilizarse, se tiene que dirigir **Anexo 1**.



WHILE LOOP	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
<p>Repite el sub-diagrama en su interior hasta que el terminal de entrada, recibe un valor booleano en particular. Del valor booleano depende la continuación de la conducta del bucle while.</p>	

Figura 6.80: Ciclo While
Fuente: Por el investigador

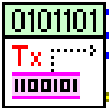
MOD CREATE TX SEQUENCE.VI
DIAGRAMA DE BLOQUES

<p>Permite crear una secuencia de bits ordenados para poder transmitirlos y luego sincronizarlos con los demás VIs.</p>

Figura 6.81: Transmisor de secuencia bits
Fuente: Por el investigador


MT GENERATE SYSTEM PARAMETERS (M) .VI
DIAGRAMA DE BLOQUES

<p>Calcula los parámetros para ser usado con el <i>MT Módulate ASK.vi</i> o <i>MT Demodulate ASK.vi</i>; dependiendo de la aplicación a ejecutarse. Este VI posee un selector polimórfica para seleccionar el tipo de modulación que se requiere junto con los parámetros M o Map.</p>

Figura 6.82: Generador de parámetros del sistema
Fuente: Por el investigador


MT GENERATE SYNCHRONIZATION PARAMETERS.VI (bits array)
DIAGRAMA DE BLOQUES

<p>Permite construir los parámetros de sincronización de un FSK, PSK, QAM, MSK, PAM, ASK. Los parámetros pasarían por un clúster a la demodulación.vi, utilizando un conjunto ya sea de bits de sincronización o números de sincronización.</p>

Figura 6.83: Generador de parámetros de sincronización
Fuente: Por el investigador


MT GENERATE FILTER COEFFICIENTS
DIAGRAMA DE BLOQUES

<p>Genera los coeficientes de filtros, aplicados a la modulación y demodulación digital.</p>

Figura 6.84: Generador de coeficientes de filtro
Fuente: Por el investigador


MT MODULATE ASK
DIAGRAMA DE BLOQUES

<p>Recibe una secuencia de bits de datos, realiza la modulación ASK, y devuelve la forma de onda compleja modulada en banda base.</p>

Figura 6.85: Modulador ASK
Fuente: Por el investigador


MT ADD AWGN.vi
DIAGRAMA DE BLOQUES

<p>Genera el Ruido Aditivo Blanco Gaussiano (AWGN), con densidad espectral de potencia uniforme y lo añade a la forma de onda compleja modulada en banda base.</p>

Figura 6.86: Generador AWGN

Fuente: Por el investigador


MT DEMODULATE ASK
DIAGRAMA DE BLOQUES

<p>Este instrumento virtual, demodula una forma de onda compleja ASK, modulada en banda base y devuelve el tiempo-alineado sobremuestreada a la forma de onda compleja, el flujo de bits demodulados, y los resultados de las mediciones a la deriva del offset.</p>

Figura 6.87: Demodulador ASK

Fuente: Por el investigador

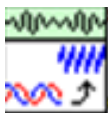
MT UPCONVERTER BASEBAND (Real).vi
DIAGRAMA DE BLOQUES

<p>Permitirá visualizar la señal modulada, transformando la señal compleja en banda base a una señal pasa banda equivalente</p>

Figura 6.88: Convertidor pasabanda

Fuente: Por el investigador


MT FORMAT EYE DIAGRAM
DIAGRAMA DE BLOQUES

<p>Este VI determina la longitud del segmento basado en la velocidad de símbolo y los parámetros de longitud de los ojos</p>

Figura 6.89: Diagrama de ojo
Fuente: Por el investigador


MT FORMAT CONSTELLATION.vi
DIAGRAMA DE BLOQUES

<p>Prepara una señal para la presentación de un gráfico que muestra las ubicaciones de símbolos detectados y las transiciones entre esos símbolos. El VI especifica una forma de onda con valores complejos y muestra un gráfico de la constelación de los contenidos de forma de onda.</p>

Figura 6.90: Diagrama Constelación
Fuente: Por el investigador


MT CALCULATE BER.vi
DIAGRAMA DE BLOQUES

<p>Calcula el promedio de la tasa de bit de error (BER).</p>

Figura 6.91: Calcula el Bit de Error
Fuente: Por el investigador

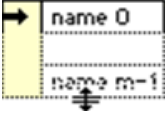
UNBUNDLE BY NAME
DIAGRAMA DE BLOQUES

<p>Permite devolver los elementos de un clúster, según el nombre del elemento especificado. Esta función permite, seleccionar un elemento individual, sin importar el orden en la que se encuentre dentro de la agrupación.</p>

Figura 6.92: Unbundle by name
Fuente: Por el investigador

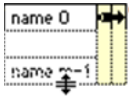
BUNDLE BY NAME
DIAGRAMA DE BLOQUES

<p>Reemplaza uno o más elementos de un clúster, esta función se refiere para los elementos que se encuentran dentro del clúster. Después de conectar el nodo a una entrada del clúster, clic derecho en el nombre de los terminales para seleccionar los elementos del menú contextual.</p>

Figura 6.93: Bundle by name
Fuente: Por el investigador


DELETE FROM ARRAY
DIAGRAMA DE BLOQUES

<p>Borrar un elemento del subconjunto para un conjunto de n-dimensiones y retorna el conjunto editado en array <i>w/subset deleted</i> y el elemento borrado o subconjunto en una porción Borrada</p>

Figura 6.94: Delete from array
Fuente: Por el investigador

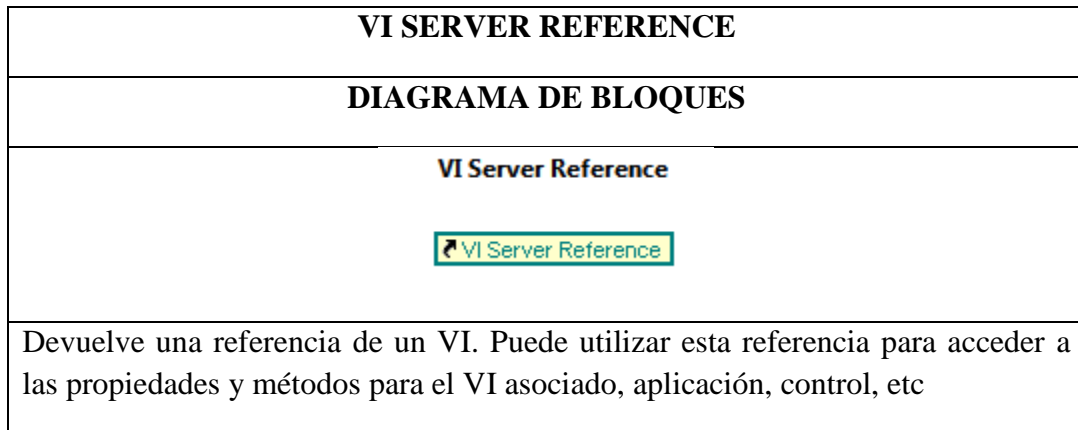


Figura 6.95: Server reference
Fuente: Por el investigador

DISEÑO DEL VI PARA EL ESQUEMA DE MODULACIÓN Y DEMODULACIÓN ASK.

Para el diseño del esquema de modulación-demodulación ASK se tiene que seguir el siguiente procedimiento y en el mismo orden establecido.

PROCEDIMIENTO:

1. Creamos un **VI**, siguiendo los pasos de inicialización de un VI.
2. Para generar los parámetros de los bits a ser transmitidos, se usa la función *Mod_Create Tx Sequence.vi* que se encuentra en la paleta de controles del diagrama de bloques. Los terminales que usamos de este VI, son: # Guard Bits (2), # Sync Bits (20), # Message Bits y PN sequence order.
3. Ubicados en el Panel frontal, elegimos cuatro controles numéricos para los terminales antes mencionados. Este tipo de control numérico, lo podemos encontrar: *Numeric/Numeric Control* y se lo etiqueta con su respectivo nombre con la ayuda del *Tool Palette* que esta ubicado en la pestaña *View*, de la parte superior de la ventana del panel frontal. Clic en **View**, aparecerá la paleta de herramientas, seleccionamos *Edit Text* y procedemos a cambiar los nombres de los controles.
4. Se procede a crear un *Clúster* para agrupar los cuatro controles creados, en un solo conjunto. Esta función esta ubicado en el Panel Frontal: *paleta de controles/Modern/Array,Matrix/Clúster*. Con tan solo arrastrar cada control

hacia el interior del clúster, tendremos todos los controles agrupados como muestra la figura 6.96.

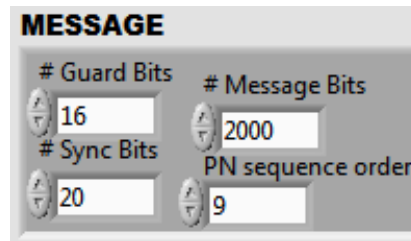


Figura 6.96: Clúster Message
Fuente: Por el investigador

- Ahora bien, para trabajar con cada elemento del clúster *Message* y conectarlos a los terminales del *Tx Sequence.vi* en el diagrama de bloques, se debe usar la función *Unbundle by Name* (ubicado: en la *paleta de funciones/Cluster, Class/Unbundle by Name*). Al conectar el clúster *Message* con el terminal *input clúster del Unbundle by Name*, aparecerá el primer elemento del *Message* dentro del clúster, para adicionar el resto de controles se da clic derecho en la misma función, clic en *Add Element* y ya tendremos un nuevo elemento en esta función como en la figura 6.97.

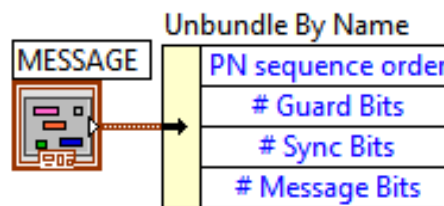


Figura 6.97: Conexión del clúster
Fuente: Por el investigador

- A continuación, la figura 6.98 muestra como se conectan las salidas del *unbundle by name* con los terminales respectivos del *Create Tx Sequence*.

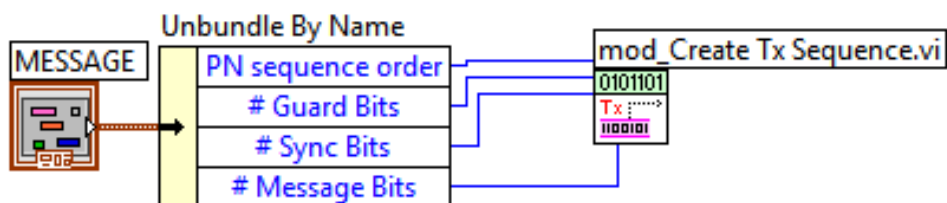


Figura 6. 98: Conectando el Unbundle by name con Tx Sequence
Fuente: Por el investigador

- Continuando con la programación, es necesario controlar los bits a ser transmitidos por lo que seleccionamos **Delete From Array**, ubicado en: *paleta de funciones/Array*, que permite retornar un nuevo sub array dependiendo de sus parámetros de entrada. Estos parámetros serán definidos como contantes numéricas y para ello, nos dirigimos: *paleta de funciones/Numeric/Numeric Constant*. Para el terminal **length**: le damos una longitud de 16 para trabajar con 16 bits y para **index** definimos 0, para que visualice todos los elementos del nuevo subarray como se muestra en la figura 6.99.

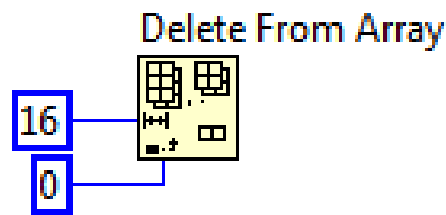


Figura 6.99: Creación de un nuevo subarray
Fuente: Por el investigador

- Una vez creadas las constantes, se conecta el terminal **output bitstream** del *Tx Sequence.vi* con la entrada **n-dim array** del **Delete From Array** y en su salida **array w/ subset deleted**, se crea un indicador para visualizar los bits transmitidos (*transmitted bitstream*) como en la figura 6.100.

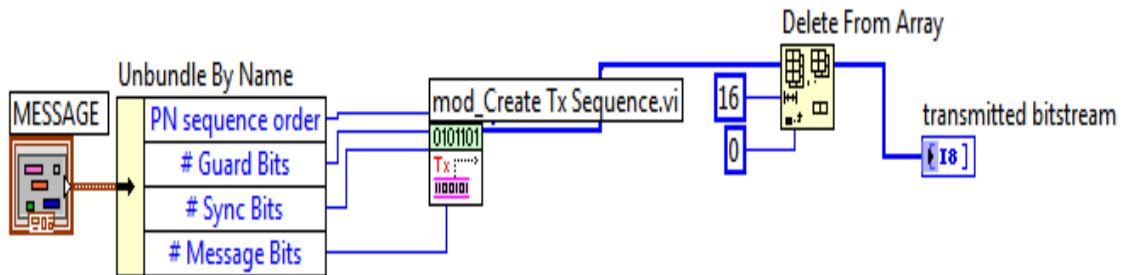


Figura 6.100: Generación de los bits transmitidos
Fuente: Por el investigador

- Como siguiente paso, seleccionamos los VI: **MT Modulate ASK** (ubicado en *panel de funciones/Addons/Modulation/Digital*) que sirve para modular la cadena de bits de entrada conocido como símbolos, y el **MT Generate Synchronization** (ubicado: *panel de funciones/Modulation/Digital/Utilities*) que permite sincronizar los bits de transmisión con los bits de recepción.

La figura 6.101 muestra las conexiones de los terminales de estos VI con los otros terminales respectivos de los otros VIs.

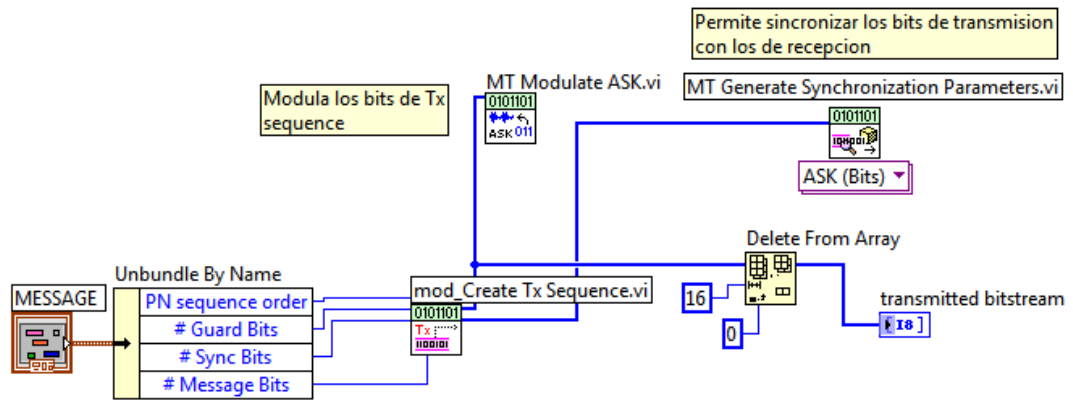


Figura 6.101: Conectando los VIs
Fuente: Por el investigador

10. El siguiente paso es seleccionar todos los demás VI necesarios para simular las modulaciones digitales, ubicados cada uno de estos módulos en la paleta de funciones: *MT Generate System Parameter.vi* y *MT Generate Filter Coefficients.vi* (se ubican en *Addons/Modulation/Digital/Utilities*), *MT Add AWGN.vi* (*Addons/Modulation/Digital/Impairments*), *MT Upconvert Baseband.vi* (*Addons/Modulation/Digital/Modulation*), *MT Demodulate ASK.vi* (*Addons/Modulation/Digital/Demodulate*), *MT Format Eye Diagram* y el *MT Format Constellation.vi* (*Addons/ Modulation/Digital/Visualization*). La figura 6.102 muestra todas estas funciones necesarias para el esquema de modulación-demodulación ASK.

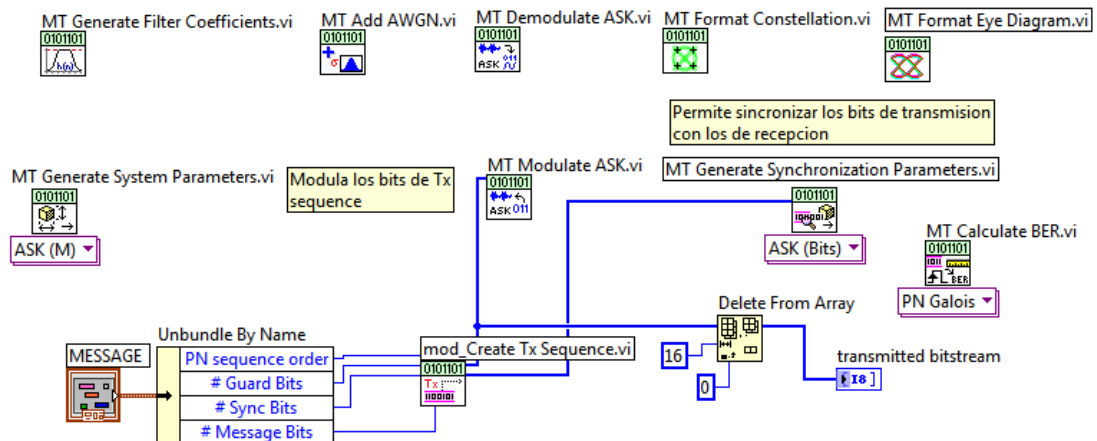
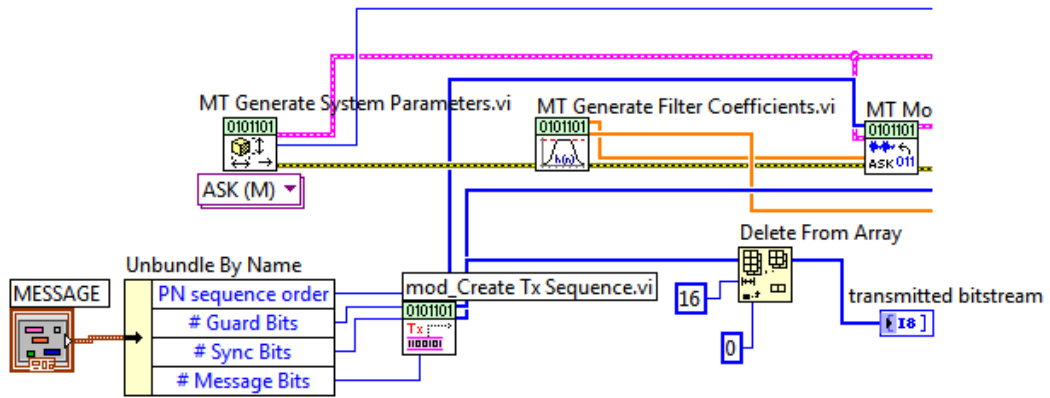
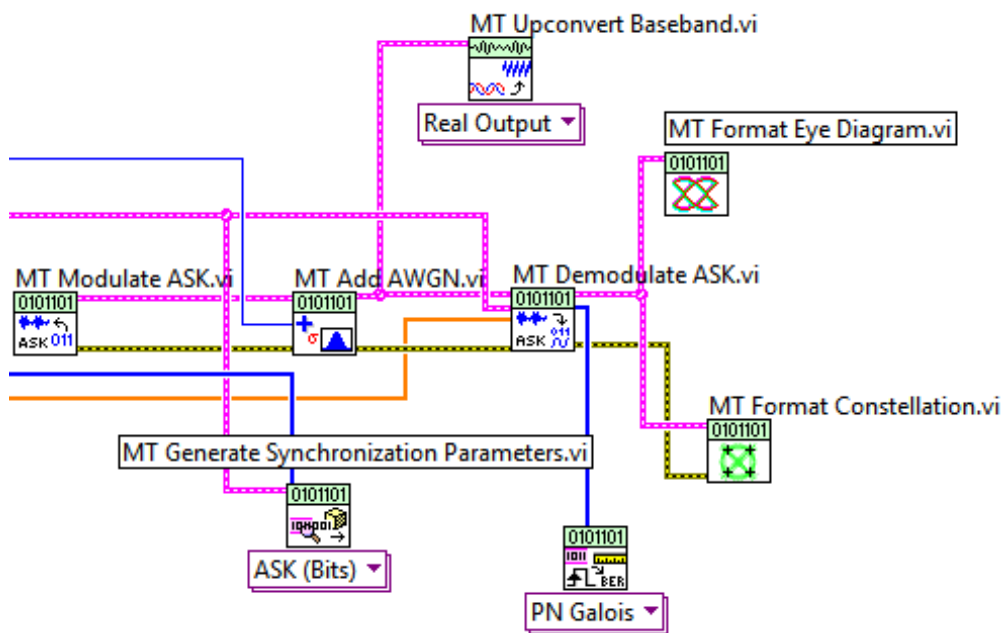


Figura 6.102: Seleccionando los VIs para la modulación-demodulación ASK
Fuente: Por el investigador

11. En la figura 6.103 se muestra la respectiva programación realizada y dividida en dos partes. La figura 6.103a muestra la conexión de los terminales del *MT Generate System Parameter.vi*. Por otro lado, se tiene el ruido AWGN que se utiliza para simular el ruido que afecta al canal de transmisión como se indica en la figura 6.103b.



(a)



(b)

Figura 6.103: Conectando los VIs; (a) Conexión de los parámetros del sistema; (b) Adicionando ruido AWGN

Fuente: Por el investigador

12. Una manera fácil y sencilla de crear: controles, constantes e indicadores de un determinado VI; es dando clic derecho en su terminal y escogiendo la opción *Create*, donde debe determinar que desea crear en el terminal un: control (*control*), constante (*constant*) o un indicador (*indicator*).
13. Después de la explicación anterior, es necesario crear una serie de controles como se muestra en la figura 6.104a. En cambio, la figura 6.104b muestra estos controles conectados a sus respectivos VIs.

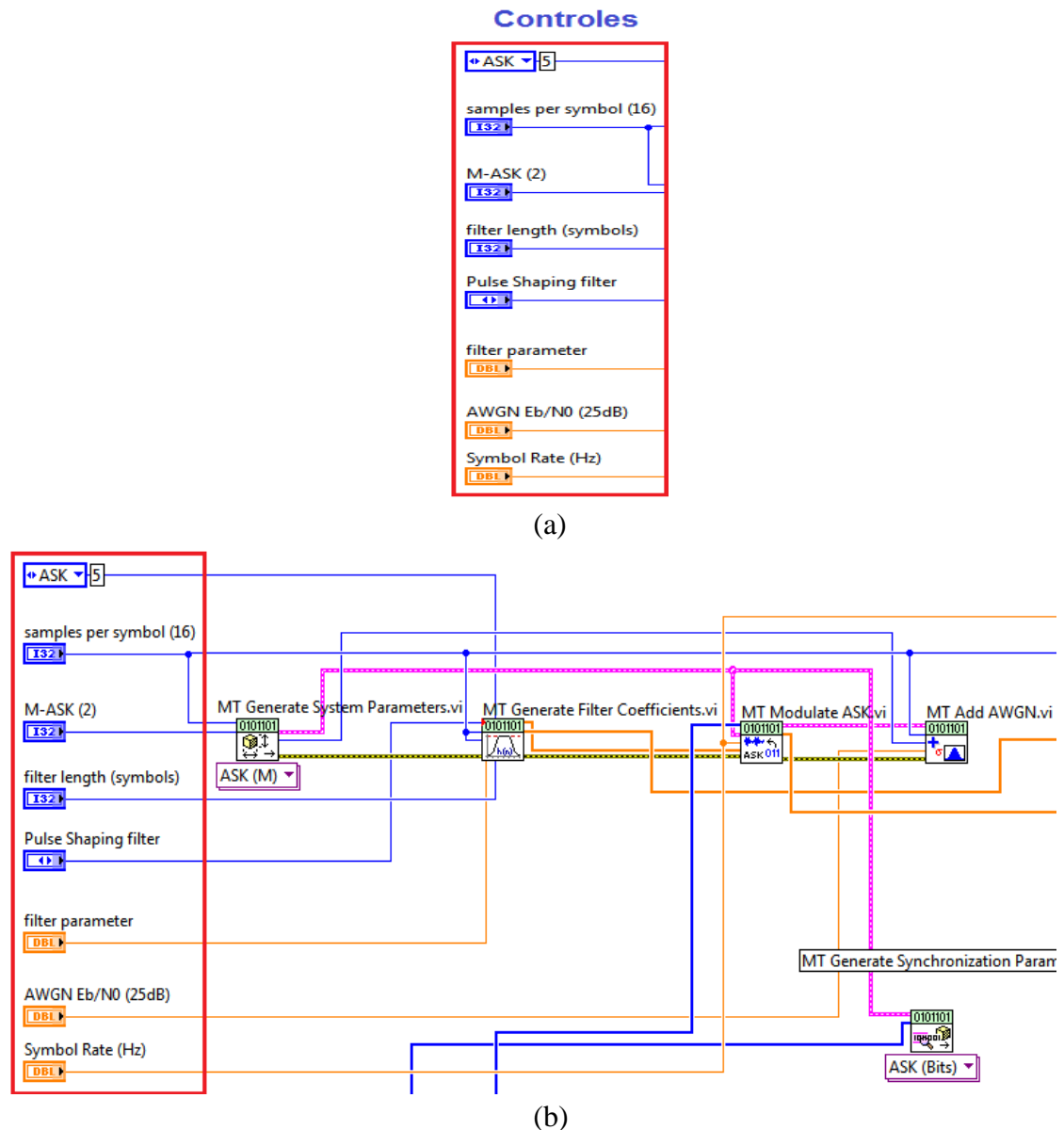


Figura 6.104: Controles para la modulación ASK; (a) controles; (b) conectando los controles a sus respectivos VIs.

Fuente: Por el investigador

14. Para variar los parámetros del diagrama de ojos, se creará dos controles: una para *eye length* (longitud del ojo) y otro para *I/Q* (Magnitud/Fase). La manera de observar los bits de recepción (*received bitstream*) del *MT Demodulate ASK*, es creando un indicador en aquel terminal. Los tipos de indicadores empleados en el Panel Frontal y que se puede ver en el diagrama de bloques de la figura 6.105 son:

- **Waveform Graphs:** para mostrar los símbolos, señal modulada, diagrama de ojo y para el espectro de frecuencia.
- **XY Graph:** para indicar el diagrama de constelación.
- **Clúster:** para agrupar los valores calculados de la demodulación en el indicador *measurements*.

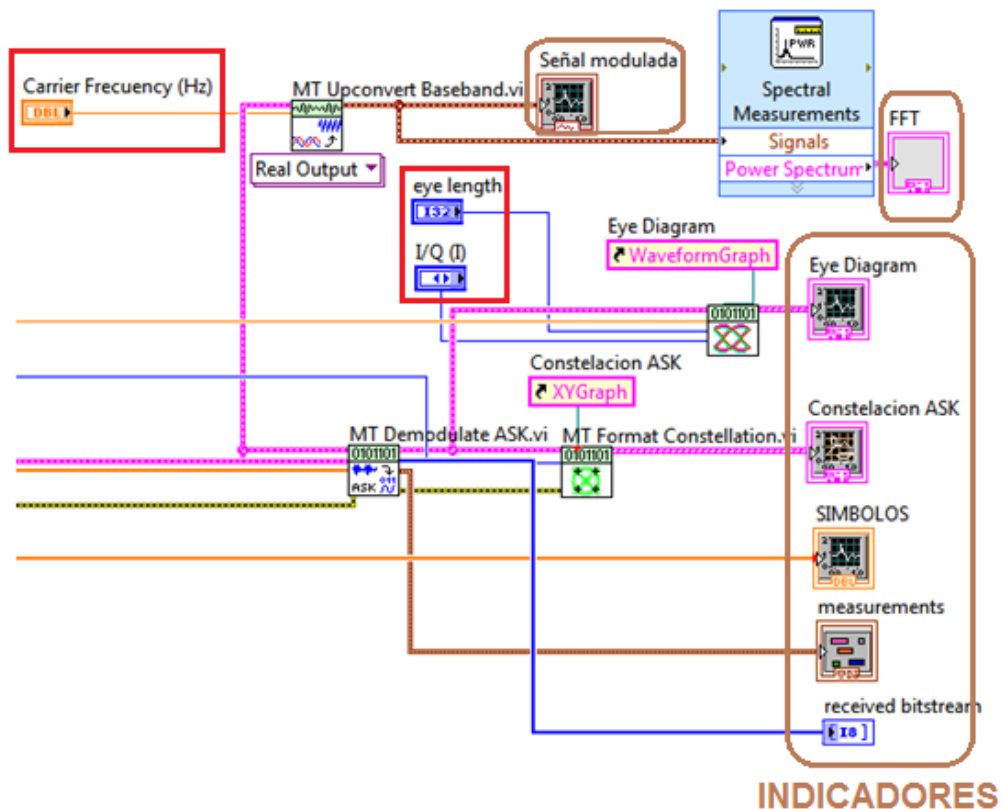


Figura 6.105: Creación de los indicadores
Fuente: Por el investigador

En la figura 6.106, se muestra la configuración del *Spectral measurements* que es utilizado para obtener el espectro de frecuencia. Para su uso se debe activar los

siguientes parámetros, que son esenciales para visualizar correctamente el espectro de frecuencia (FFT):

- **power spectrum:** Para visualizar varias frecuencias de una señal.
- **Result:** Debe ser Linear, para visualizar el resultado en las unidades originales,
- **Window:** Se elige **None**, para no aplicar ninguna ventana para señales).

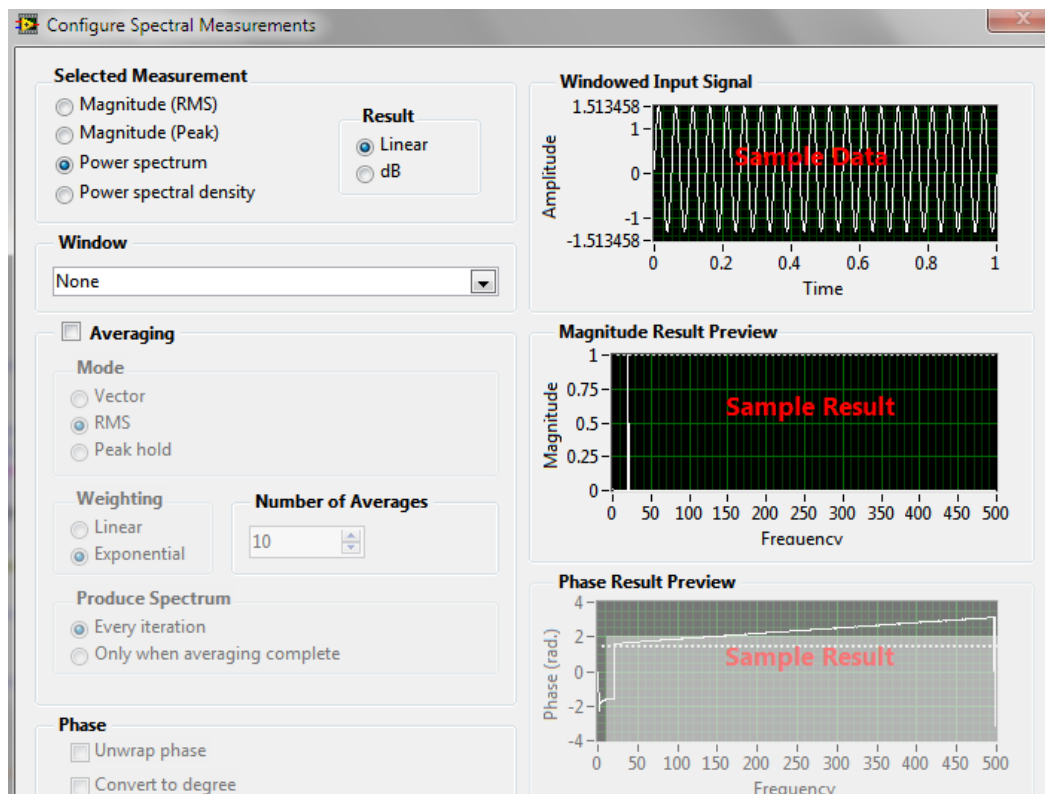


Figura 6.106: Configuración Spectral Measurements

Fuente: Por el investigador

15. Ahora para sincronizar los bits de transmisión con los de recepción, se debe realizar la siguiente programación: se necesita la función **Unbundle By Name**, para retornar los elementos de los parámetros de sincronización del **MT Generate Synchronization** al terminal **input element** de un **Bundle By Name**. De esta manera, la salida del mismo será utilizado por el terminal **synchronization parameters** del **MT Demodulate ASK**. La figura 6.107 muestra

la creación de los elementos de los cluster *unbundle by name* y del *bundle by name*.

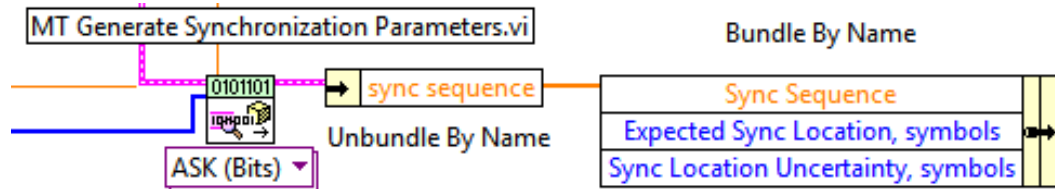


Figura 6.107: Parámetros de sincronización
Fuente: Por el investigador

- El procedimiento se describe así: en el *MT Generate Synchronization Parameter*, damos clic derecho en el terminal *ASK synchronization parameter* y seleccionamos *create/constant*, esta constante se une al *Bundle By Name*, en el terminal *input cluster*; donde se podrá ver los elementos creados y unir a los controles respectivos (*sync location uncertainty*, *expected sync location*). En este VI, se conecta el terminal *ASK synchronization parameter* al terminal *cluster of named* del *Unbundle By Name*. Por ultimo, se crea un indicador para el bit de error del *MT Calculate BER.vi*, ver la figura 6.108.

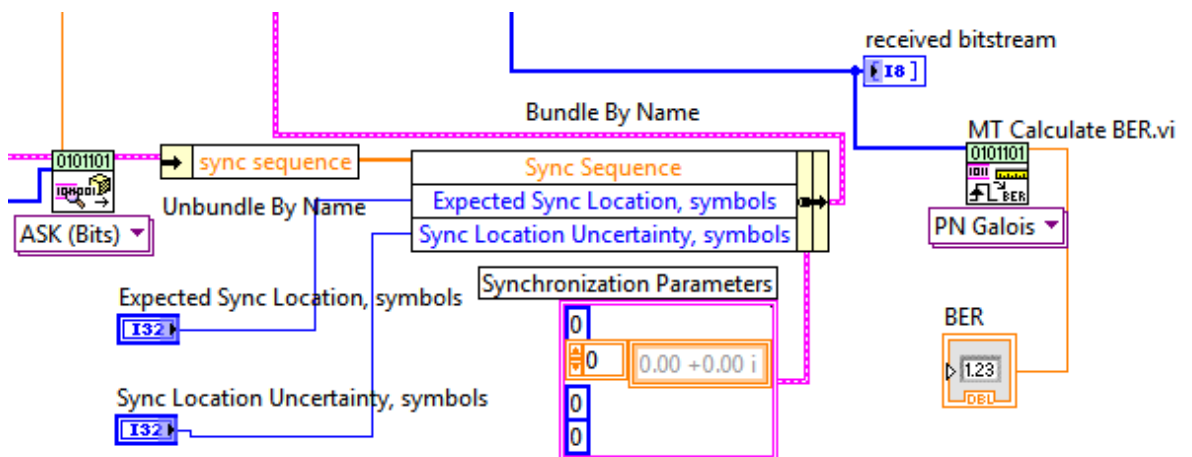


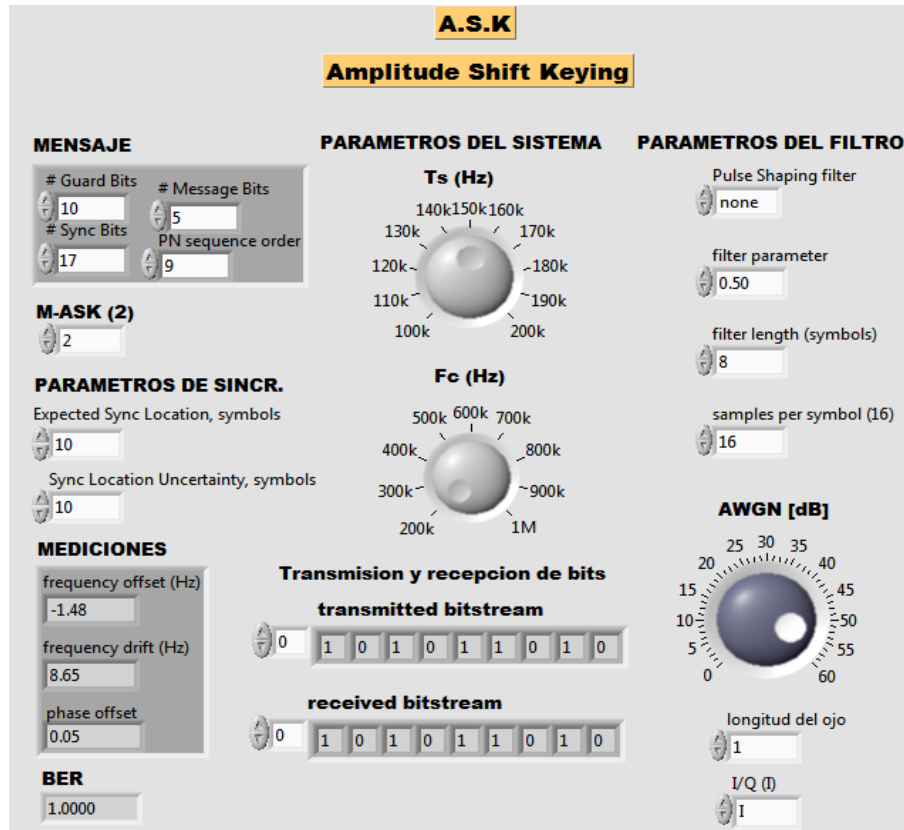
Figura 6.108: Controles para la sincronización
Fuente: Por el investigador

Nota: Si no entiende el funcionamiento de algún instrumento, puede dar clic derecho en el elemento y así seleccionar **Help**, para que se abra la ventana de ayuda.

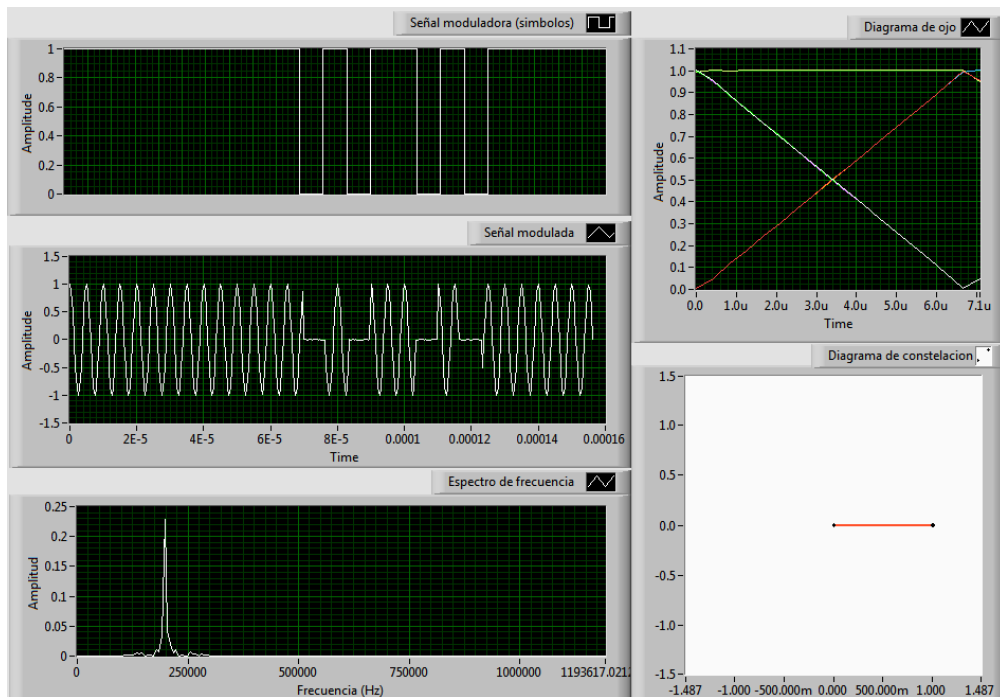
Panel frontal del SubVI modulación y demodulación ASK.

La figura 6.109 muestra el interfaz de la modulación-demodulación ASK dividida en dos partes. La primera parte, muestra la figura 6.109a que esta conformado por el **Mensaje**, la misma que contiene todos los controles necesarios para crear el mensaje a transmitirse. Los **Parámetros del sistema** en cambio son todos los controles que permiten modificar la señal modulada como son: el **Ts** (Tasa de símbolos), la **Fc** (Frecuencia de portadora) y el **M-ASK (2)**, control que permite variar la cantidad de condiciones posibles para una determinada cantidad de bits. Los **Parámetros del filtro** muestra los controles para simular un filtro que permite eliminar las señales no deseadas como el ruido. Los **Parámetros de sincronización** contienen dos controles encargados de sincronizar los bits transmitidos y recibidos. **Mediciones** presenta las mediciones de la demodulación ASK. El control **AWGN** es utilizado para simular como el Ruido Gaussiano Blanco Aditivo afecta al canal de transmisión. **Transmisión y recepción de la cadena de bits** son indicadores numéricos que muestran una cadena de bits transmitidos y recibidos. Se puede apreciar dos controles (**longitud de ojo, I/Q**) que se utilizan para modificar la grafica del diagrama de ojo. En cambio la segunda parte, mostrada en la figura 6.109b se tienen: la **Señal moduladora** que es la ventana donde se visualiza una señal digital correspondiente a los símbolos a transmitirse, la **Señal modulada** que permite observar la señal modulada ASK luego de aplicar la técnica de modulación digital correspondiente, el **Espectro de frecuencia** muestra el espectro de frecuencia (FFT) de la señal modulada ASK, el **Diagrama de ojo** en esta ventana se muestra la superposición de las distintas combinaciones posibles de unos y ceros para analizar el comportamiento del enlace de transmisión y finalmente se muestra el **Diagrama de constelación** para saber el espacio entre los símbolos recibidos en el demodulador ASK.

Como una manera de ejemplo se ha ingresado los valores que se muestran en la interfaz de la simulación de la modulación-demodulación ASK, de la figura 6.109. Los valores a ingresar en el mensaje, deben ser suficientemente grandes para visualizar claramente el comportamiento del diagrama de ojo. Sin embargo, para analizar si la señal de modulación es correcta, se ha ingresado valores bajos en el mensaje. La modulación ASK, se caracteriza por variar su amplitud en relación a los datos de la señal moduladora, condición que si se cumple en la grafica de la señal modulada. Es importante también ir variando el control del AWGN para mejorar la calidad de las señales, así como los controles de los parámetros de sincronización, con la finalidad de obtener el sincronismo entre los bits de transmisión y recepción. En la presente figura muestra la cadena de bits transmitidos que es igual a la cadena de bits recibidos, por consiguiente afirmamos que la transmisión de los datos fueron realizados correctamente.



(a)



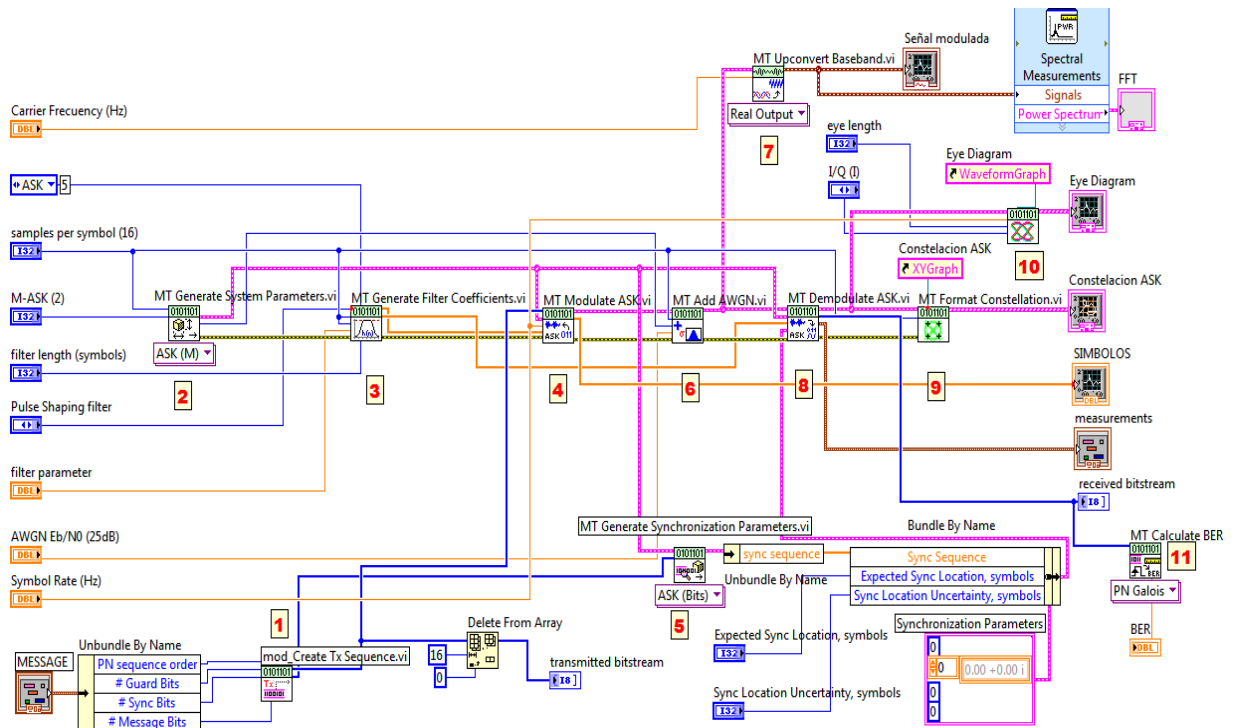
(b)

Figura 6.109: Panel Frontal SubVI-ASK: (a) controles; (b) visualización de las señales

Fuente: Por el investigador

Diagrama de bloques de la modulación y demodulación ASK

La figura 6.110 muestra el diagrama completo de la modulación-demodulación ASK, con sus respectivos controles e indicadores que permiten manipular sus parámetros de entrada y visualizar el comportamiento de sus señales frente a los valores que se vayan a ingresar.



1. Genera una cadena de bits
2. Genera los parámetros del sistema
3. Genera los coeficientes de filtro
4. Modulador ASK
5. Genera los parámetros de sincronización
6. Adiciona AWGN.
7. Convertidor de banda base
8. Demodulador ASK
9. Formato de la Constelación
10. Diagrama de ojo
11. Calculador de BER

Figura 6.110: Diagrama de bloques del SubVI-ASK

Fuente: Por el investigador

6.8.10 MODULACION Y DEMODULACION FSK

6.8.10.1 Descripción de los elementos a ser utilizados en la realización de la modulación y demodulación digital FSK.

PANEL FRONTAL DEL INSTRUMENTO VIRTUAL

Los elementos que se utilizaron para crear el panel frontal de la modulación FSK, son los mismos que la anterior modulación con la siguiente excepción:

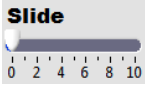

PESTAÑA DE CONTROLES>>NUMERIC>>HORIZONTAL POINTER SLIDE	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
<p>Es una barra deslizante horizontal con puntero, permitiendo ingresar valores a una escala fija o variable.</p>	

Figura 6.111: Barra deslizante horizontal con puntero

Fuente: Por el investigador

DIAGRAMA DE BLOQUES DEL INSTRUMENTO VIRTUAL

Para la programación de la modulación FSK, se utilizo las funciones de la anterior modulación, pero adicionando dos herramientas que son:

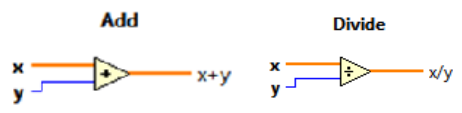
EXPRESIONES MATEMATICAS
DIAGRAMA DE BLOQUES

<p>Estas dos expresiones matemáticas permiten sumar (Add) o dividir (divide) los valores que ingresen en sus entradas x, y.</p>

Figura 6.112: Expresiones Add y Divide

Fuente: Por el investigador

DISEÑO DEL VI PARA EL ESQUEMA DE MODULACIÓN Y DEMODULACION FSK.

PROCEDIMIENTO:

1. Creamos un VI, siguiendo los pasos de inicialización.
2. Se empieza a realizar el mismo procedimiento que la modulación y demodulación ASK y se va cambiando el *selector polimórfico* de ASK a FSK, de todas las funciones que tengan este selector como en la figura 6.113.

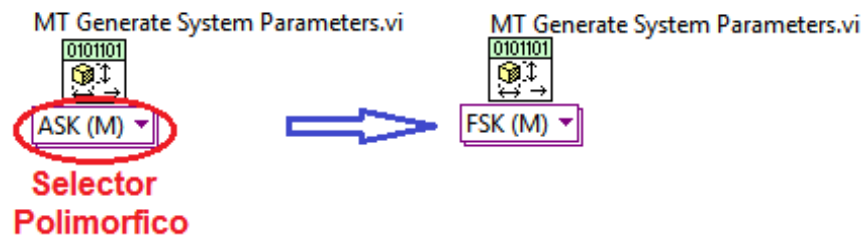


Figura 6.113: Cambio del selector polimórfico
Fuente: Por el investigador

3. En el *MT Generate System Parameter.vi*, es donde se debe agregar dos controles adicionales: una para la **frequency deviation** (*desviación de la frecuencia*) y la otra para **symbol phase continuity** (*símbolo de fase continua*). Para ello, se debe dirigir al panel frontal para crear el control para la desviación de la frecuencia que se encuentra: *paleta de control/Numeric/Horizontal Pointer Slide*, y para el control de símbolo de fase, se utiliza un *Numeric Control*. La figura 6.114 muestra los controles que se agregaron al *MT Generate System Parameter.vi*,

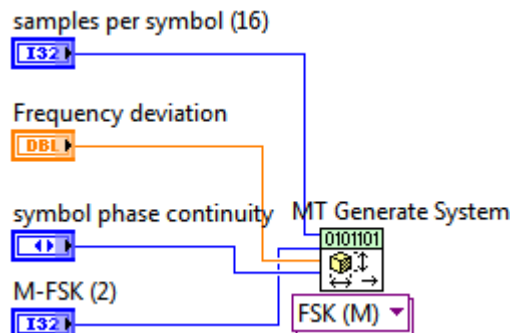
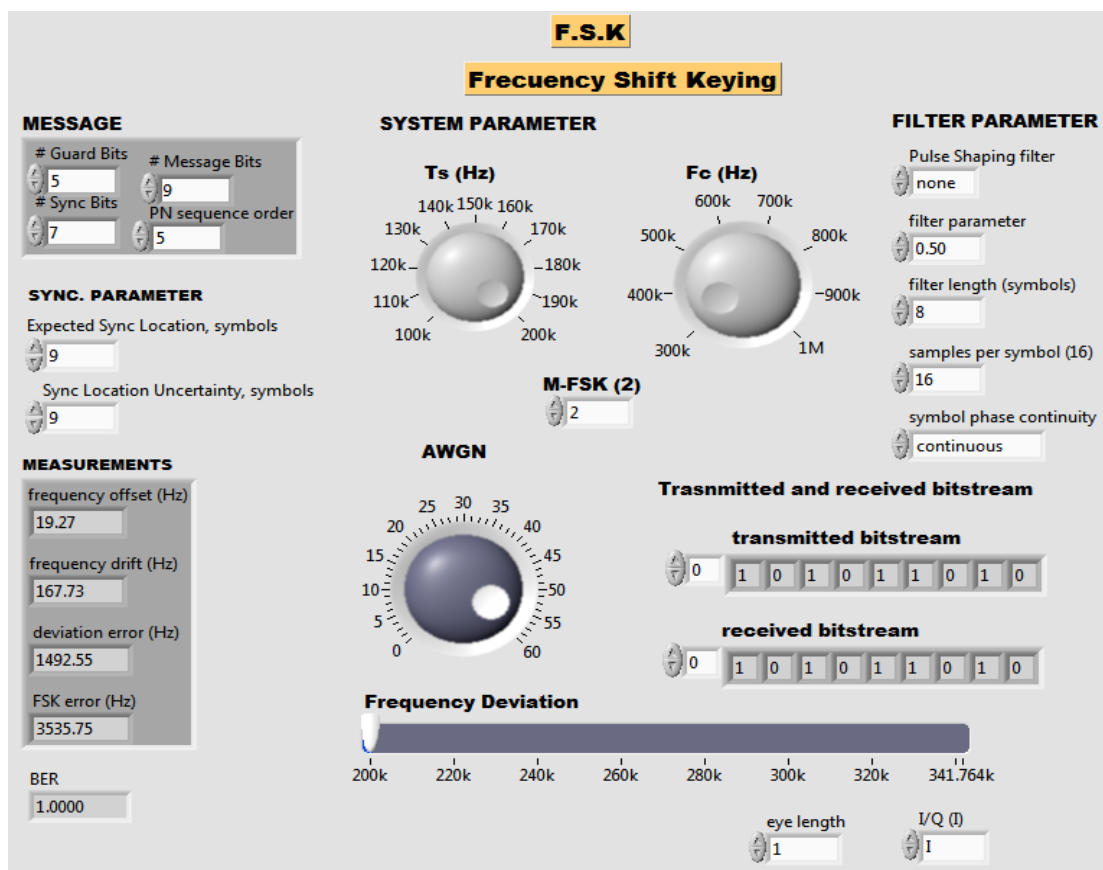


Figura 6.114: Creación de los controles para el MT Generate System.vi
Fuente: Por el investigador

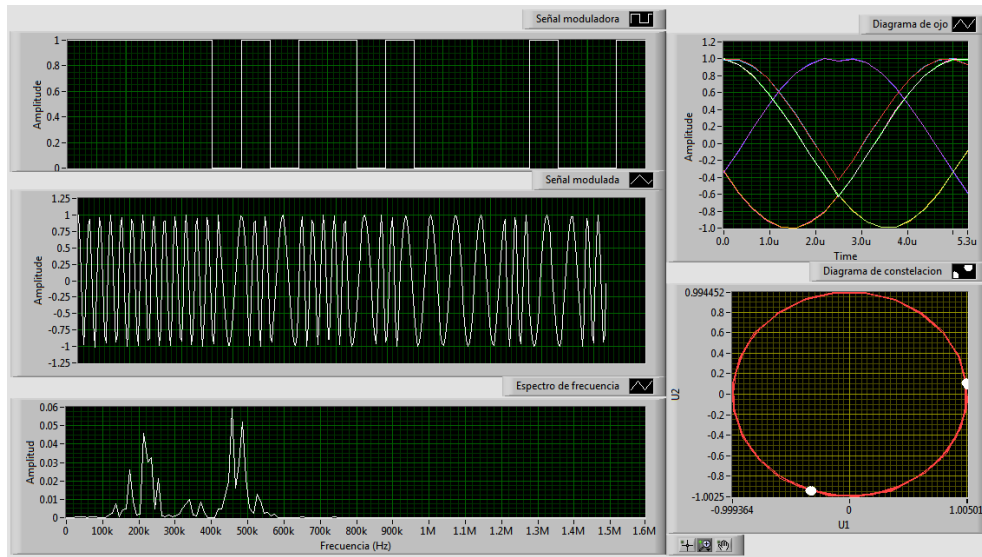
4. Luego de haber creado el control *frequency deviation* (desviación de frecuencia), se da clic derecho en el mismo y se ubica en *create/Property Node/Scale/Maximum* que permite crear un propiedad de nodo, relacionado a la escala máxima del control Fc (frequency carrier) y una escala mínima al control Ts (symbol rate), eligiendo *Minimum*.
5. En este tipo de modulación FSK, es indispensable el uso de funciones aritméticas como la suma (*Add*) y división (*Divide*), que se encuentran: *paleta de funciones/Numeric*. La desviación de frecuencia tiene que ser dividida por los *symbols out* del *MT Modulate FSK*, su resultado debe sumarse con 1 y finalmente se obtiene los símbolos a ser transmitidos. Este procedimiento es muy importante para tener la amplitud de los símbolos en 1, caso contrario tendremos variaciones de la misma.

Panel frontal del VI modulación y demodulación FSK.

La figura 6.115a muestra el panel frontal de la modulación-demodulación FSK con sus respectivos controles e indicadores. En cambio, la figura 6.115b muestra las graficas de las señales como son: la señal moduladora, señal modulada, espectro de frecuencia, diagrama de ojo y el diagrama de constelación.



(a)



(b)

Figura 6.115: Panel Frontal SubVI-FSK: (a) controles; (b) visualización de las señales
Fuente: Por el investigador

Diagrama de bloques del VI Modulación y demodulación FSK

La figura 6.116 muestra la programación completa del diagrama de bloques de la modulación-demodulación FSK, que es similar al esquema de modulación-demodulación ASK. Sin embargo, las funciones utilizadas hacen referencia a la modulación FSK.

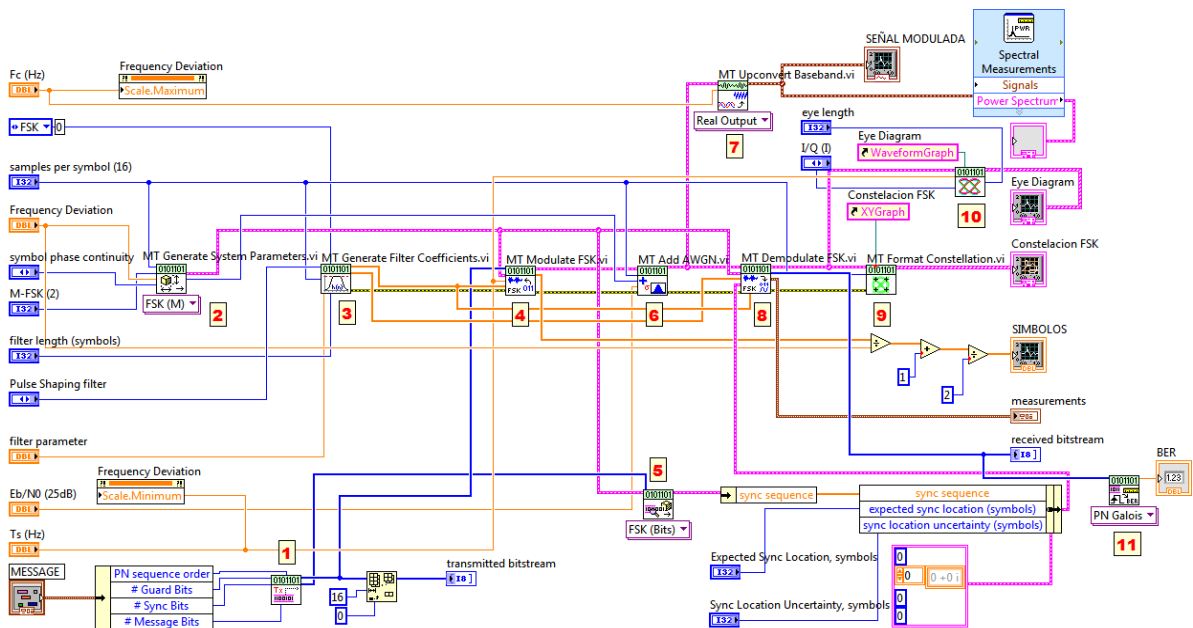


Figura 6.116: Diagrama de bloque del SubVI-FSK
Fuente: Por el investigador

6.8.11 MODULACIÓN Y DEMODULACIÓN PSK

6.8.11.1 Descripción de los elementos a ser utilizados en la técnica de modulación y demodulación PSK

PANEL FRONTAL DEL INSTRUMENTO VIRTUAL

Los elementos ya no hacen falta describirlos para el diseño del panel frontal de la modulación-demodulación PSK porque son similares a la modulación ASK. Sin embargo, es necesario conocer dos tipos de controles adicionales: una para seleccionar el tipo de PSK y el otro para el PSK diferencial.

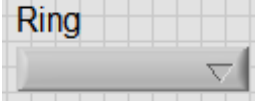
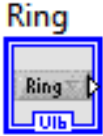
PESTAÑA DE CONTROLES>> RING&ENUM >>MENU RING	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
Es un control tipo menú que permite desplegar una o varias opciones que se haya editado en sus ítems.	

Figura 6. 117: Menu Ring
Fuente: Por el investigador

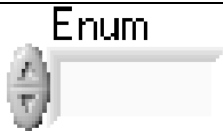

PESTAÑA DE CONTROLES>> RING&ENUM >>ENUM	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
Con esta herramienta se puede enumerar los valores que se desee obtener en cada incremento que se realice.	

Figura 6. 118: Enum
Fuente: Por el investigador

DIAGRAMA DE BLOQUES DEL INSTRUMENTO VIRTUAL

Las funciones para este tipo de modulación son las mismas que la modulación-demodulación ASK, pero con ciertas variantes que se explicaran a continuación.

DISEÑO DEL VI PARA EL ESQUEMA DE MODULACIÓN Y DEMODULACIÓN PSK

PROCEDIMIENTO:

1. Creamos un VI, siguiendo los pasos de inicialización de un VI.
2. Se empieza por cambiar el selector polimórfico de ASK a PSK de algunos VIs como se muestra en la figura 6.119 y agregar o eliminar algunos controles en del *MT Generate System Parameter.vi*

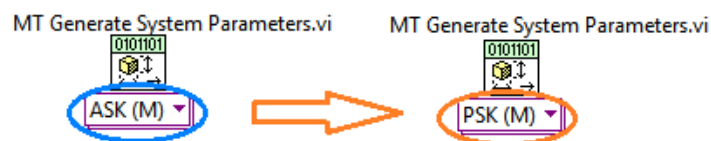


Figura 6.119: Modificación del selector polimórfico
Fuente: Por el investigador

3. El primer control a adicionar, es el **PSK type** que permitirá seleccionar el tipo de PSK, ubicado: *paleta de controles/Ring&Enum/Menu Ring*. En este mismo control, se debe dar clic derecho para acceder a sus propiedades en **Edit ítems** y editar (normal, shifted, offset). El segundo control, es para el **differential PSK**, que se encuentra en el panel frontal: *paleta de controles/Numeric/Enum*, en **Edit ítems** (editamos: enable, disable), ver la figura 6.120.

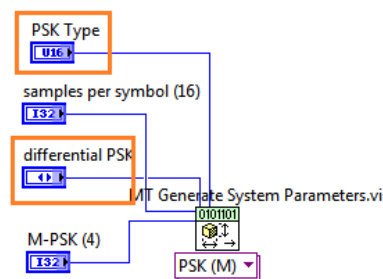
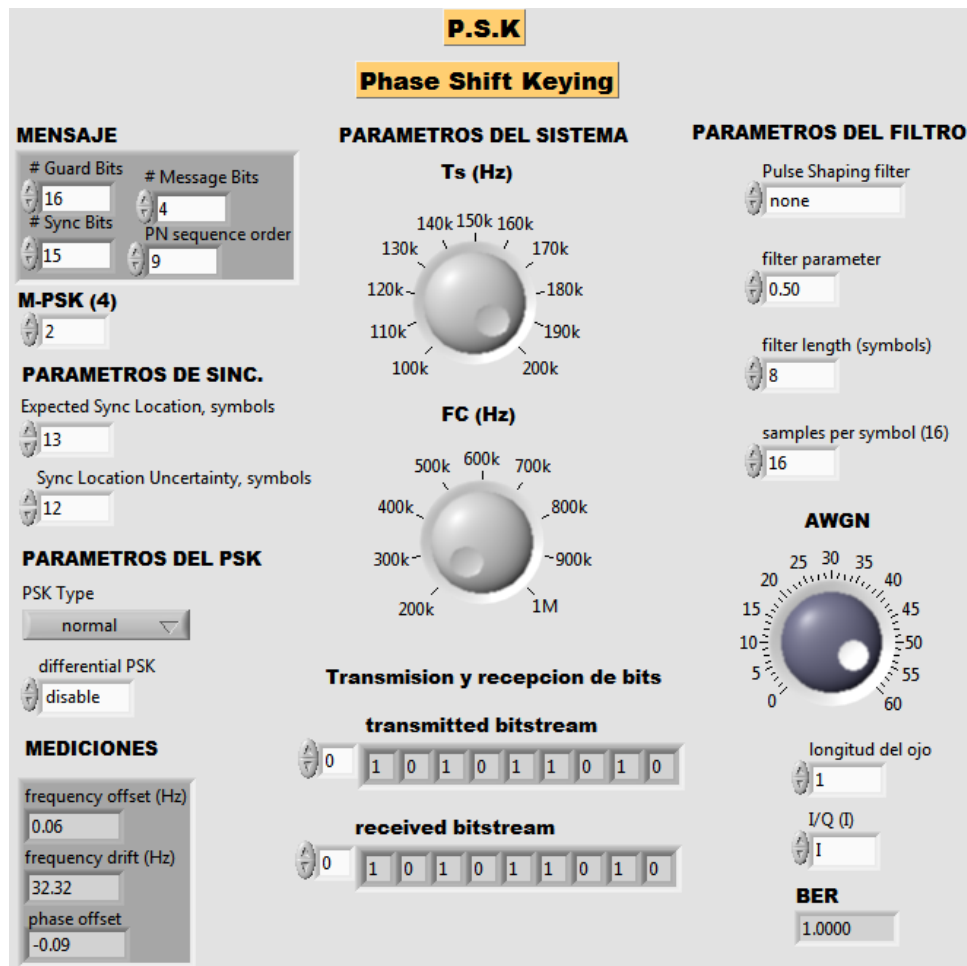


Figura 6.120: Controles para seleccionar el tipo y el diferencial PSK
Fuente: Por el investigador

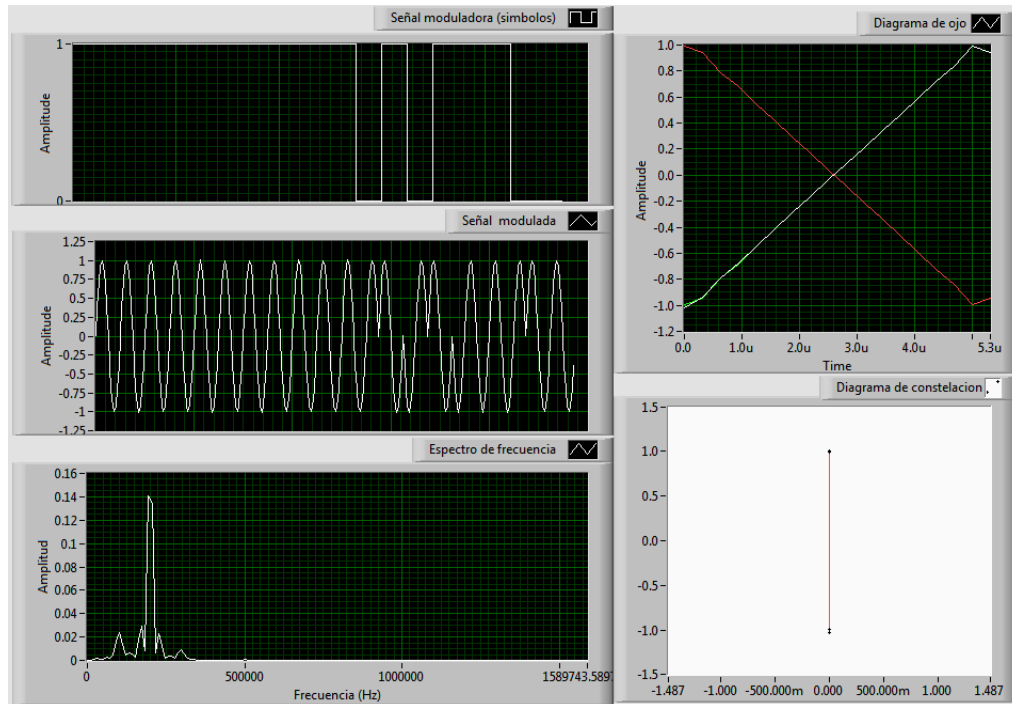
4. Para visualizar la señal moduladora o símbolos en el *Waveform Graph*, se debe utilizar un *Build Waveform*, que esta formado por una *componente Y* y un *dt*. Con la ayuda de la función *Reciprocal*, se dividirá a la unidad con la tasa de símbolos (Symbol Rate), para tener el *dt* y los *bitstream* transmitidos, será la *componente Y* del *Build Waveform* respectivamente.

Panel Frontal del VI Modulación y demodulación PSK

La figura 6.121a muestra el panel frontal de la modulación-demodulación PSK, con nivel $M=2$. Como una manera de comprobar el funcionamiento de este SubVI PSK, se asignado valores apropiados a sus controles como lo indica la presente figura. También se aprecia que la cadena de bits de transmisión con los bits de recepción son los mismos, por lo tanto la transmisión se lo ha realizado con éxito.



(a)



(b)

Figura 6.121: Panel Frontal del SubVI-PSK: (a) controles;
(b) visualización de las señales

Fuente: Por el investigador

En la figura 6.121 se muestra la grafica de la señal moduladora que dependiendo del tipo de modulación, en este caso PSK, permite determinar la señal modulada que varia su fase de acuerdo a la señal moduladora. El espectro de frecuencia muestra a la señal modulada en función de la amplitud y la frecuencia. El diagrama de ojo, no se lo puede visualizar claramente debido a que el mensaje tiene valores muy bajos en este ejemplo. En el diagrama de constelación se muestra los dos símbolos separados a una determinada distancia que significa que los símbolos son transmitidos sin ninguna interferencia de ruido.

Diagrama de bloques del VI Modulación y demodulación PSK

La figura 6.122 muestra el diagrama de bloques correspondiente a la modulación-demodulación PSK y como se puede observar se ha venido manteniendo el esquema de la modulación ASK. Pero con algunas diferencias como el uso de controles adicionales en los terminales del *MT Generate System Parameter*, uso de la función *Build Waveform* para generar la señal moduladora o símbolos y el cambio de los selectores polimórficos de los VIs a PSK. Finalmente, se conectó todos los terminales que tenían dependencia de otros VIs, utilizando las herramientas de ayuda que tiene Labview 2011.

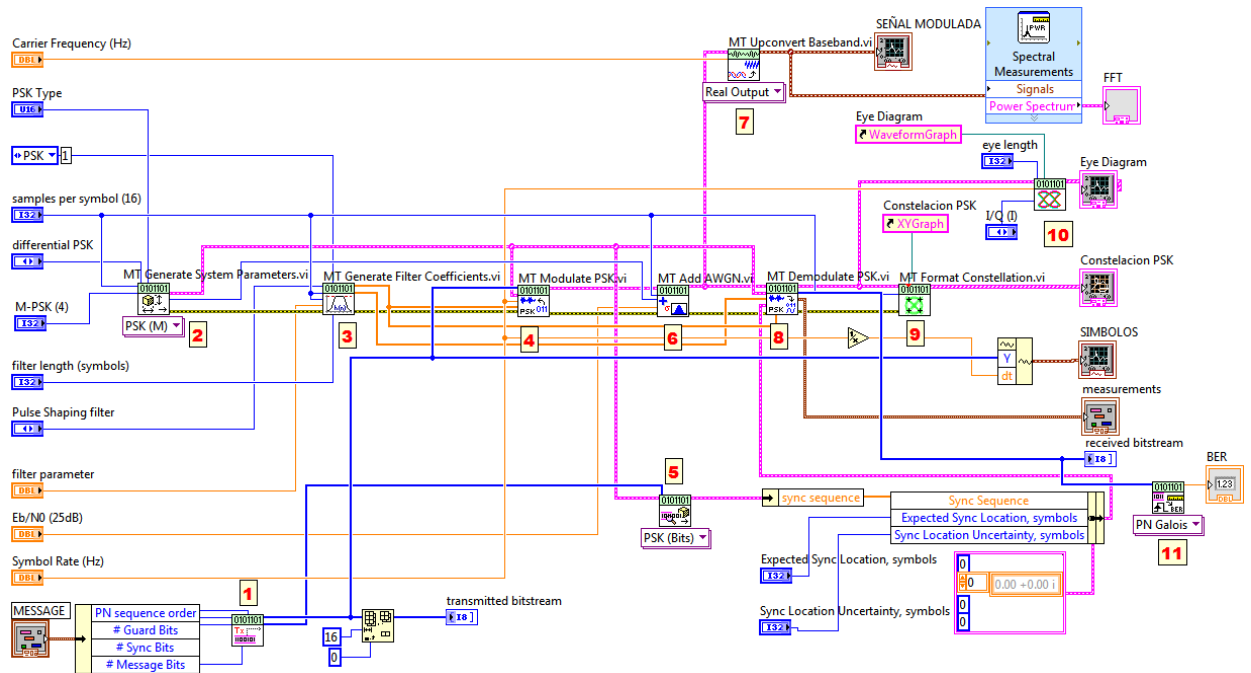


Figura 6.122: Diagrama de bloques del SubVI-PSK
Fuente: Por el investigador

6.8.12 MODULACIÓN Y DEMODULACIÓN QAM

6.8.12.1 Descripción de los elementos a ser utilizados en la realización de la modulación y demodulación QAM.

PANEL FRONTAL DEL INSTRUMENTO VIRTUAL

En el panel frontal de la modulación-demodulación QAM, se mantiene los mismos elementos que las anteriores modulaciones implementadas, por esta razón ya no se lo describe nuevamente.

DIAGRAMA DE BLOQUES DEL INSTRUMENTO VIRTUAL

Las funciones utilizadas para este tipo de modulación QAM, son las mismas que utilizamos para el diseño de la modulación ASK.

DISEÑO DEL VI PARA EL ESQUEMA DE MODULACIÓN Y DEMODULACIÓN QAM.

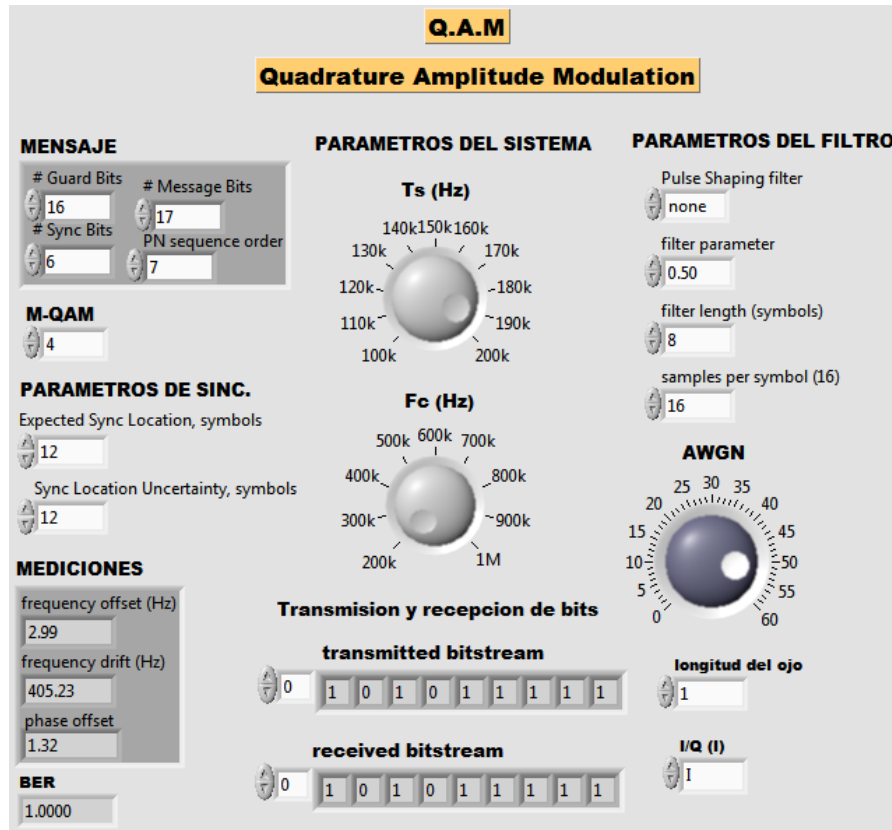
PROCEDIMIENTO:

1. Creamos un nuevo VI, siguiendo los pasos de inicialización.
2. El procedimiento, es el mismo que la modulación- demodulación ASK, solo que se debe cambiar el selector polimórfico a QAM de algunas funciones y crear los controles que sean necesarios en el *MT Generate System Parameter.vi*.
3. Para visualizar los símbolos en el panel frontal QAM, se tiene que utilizar un *Build Waveform*, que esta formado por dos terminales *componente y* y un *dt*. Siendo, la tasa de símbolos (**Ts**) la componente *dt* y los *bits stream* transmitidos del *Create Tx Sequence*, la *componente y*.

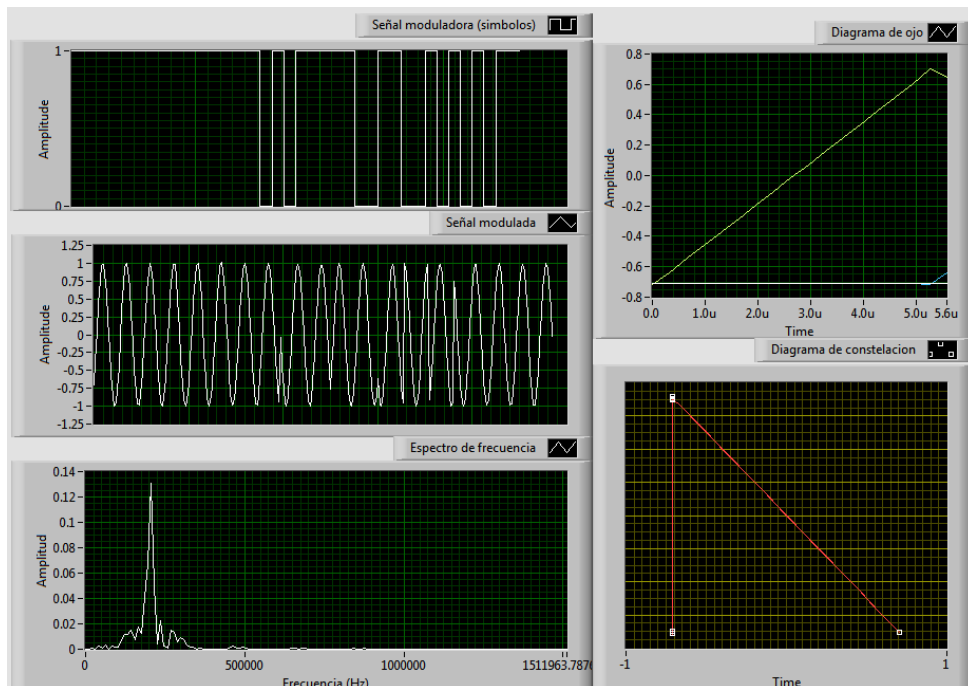
Panel frontal del instrumento virtual QAM

La figura 6.123 muestra el panel frontal con todos los elementos necesarios para su manipulación correcta, en la modulación–demodulación QAM. Es importante saber que todas las modulaciones anteriores tienen el mismo diseño del panel frontal QAM. Con la diferencia que sus resultados mostrados en la señal modulada, diagrama de ojo, diagrama de constelación y sus mediciones, son totalmente diferentes a las modulaciones antes diseñadas.

Los valores que se han ingresado en el panel frontal del SubVI QAM, es solo a manera de ejemplo para comprobar si los resultados del proceso de modulación–demodulación QAM, son los esperados. Una manera sencilla de conocer si los datos enviados llegaron correctamente al receptor, es analizando los transmitted bitstream (cadena de bits transmitidos) sean iguales a los received bitstream (cadena de bits recibidos). En nuestro ejemplo, se cumple esta condición por lo tanto el proceso de modulación y demodulación QAM, se ejecutaron correctamente.



(a)



(b)

Figura 6.123: Panel Frontal SubVI-QAM: (a) controles; (b) visualización de las señales
Fuente: Por el investigador

Diagrama de Bloques del instrumento virtual QAM

La figura 6.124 muestra todo el diseño de la modulación-demodulación QAM, desarrollado en el diagrama de bloques. Como se puede apreciar en esta figura, la estructura se sigue manteniendo de las modulaciones anteriores (ASK, FSK, PSK) pero con ciertos cambios en los selectores polimórficos de los VIs al tipo de modulación que se requiere implementar. Sus enumeraciones corresponden a las etapas realizadas, para obtener el SubVI QAM.

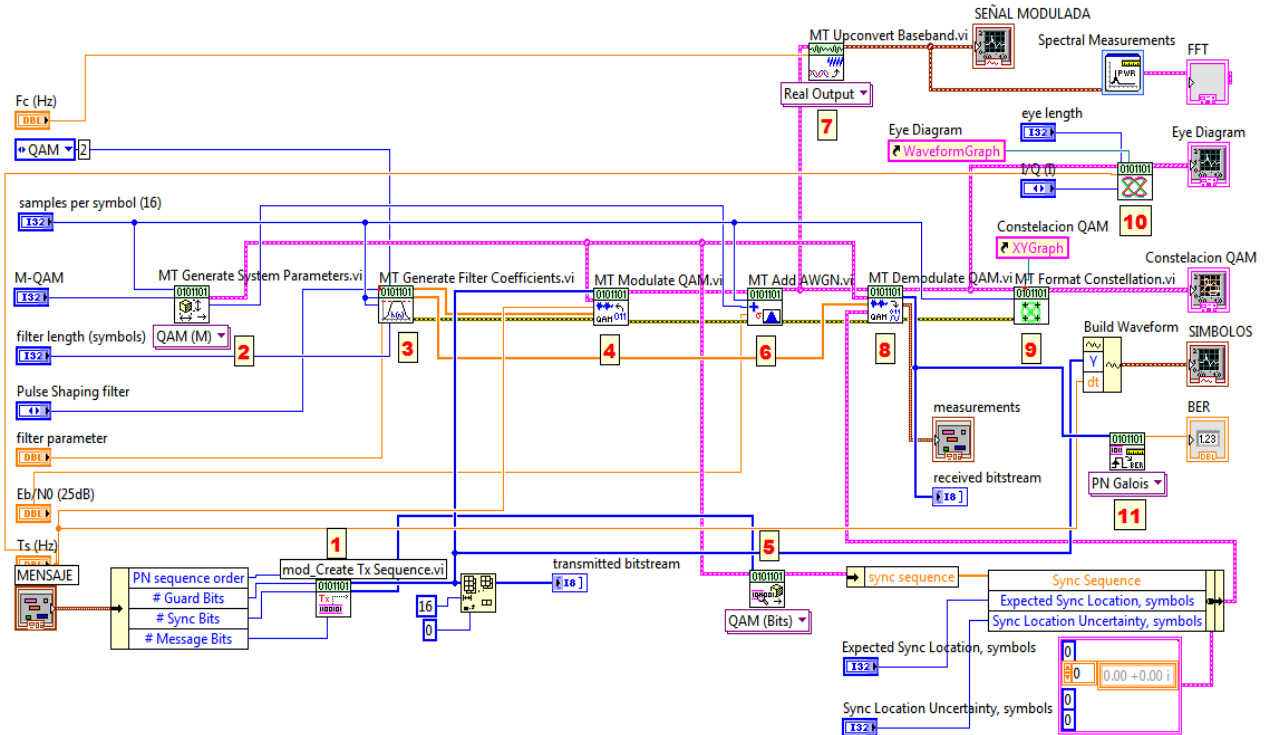


Figura 6.124: SubVI-QAM
Fuente: Por el investigador

6.8.13 DESARROLLO DEL PROGRAMA PRINCIPAL

Una vez finalizado los diferentes *SubVIs* correspondientes a cada codificación de línea se procede a unirlos en el programa principal denominado *Señales en banda base*. De la misma manera, se realiza un programa principal para las *señales pasa banda*. En conclusión, se debe tener dos programas principales para las dos aplicaciones mencionadas anteriormente.

Descripción de los elementos utilizados en el panel frontal para el desarrollo programa principal de las señales en banda base.

Los elementos utilizados en el panel frontal principal, son los mismos elementos utilizados en cada SubVI, desarrollados anteriormente a excepción:



PESTAÑA DE CONTROLES>>CONTAINERS	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
Esta herramienta, permite interactuar varias aplicaciones de manera simultanea para tener una mejor navegación dentro del diagrama de bloques, y se puede agregar las pestañas que se desee.	

Figura 6.125: Contenedor
Fuente: Por el investigador

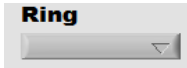

PESTAÑA DE CONTROLES>>RING & ENUM>>MENU RING	
PANEL FRONTAL	DIAGRAMA DE BLOQUES
	
Este control es utilizado, para seleccionar varios ítems de un menú definido por el usuario.	

Figura 6.126: Menú
Fuente: Por el investigador

PANEL FRONTAL DEL VI PRINCIPAL DE SEÑALES EN BANDA BASE.

La figura 6.127 muestra el panel frontal de VI Principal de las señales en banda base. Su diseño se ha mantenido de los SubVIs de cada codificación de línea desarrollado anteriormente, pero con algunas variantes como son: un control para seleccionar el **TIPO DE CÓDIGO** a codificar, un control para salir al **MENÚ PRINCIPAL** y por ultimo un control para abrir una ventana de **AYUDA**, con la información técnica de cada codificación de línea. La descripción detallada del VI principal, se encuentra en **Anexos 2**.

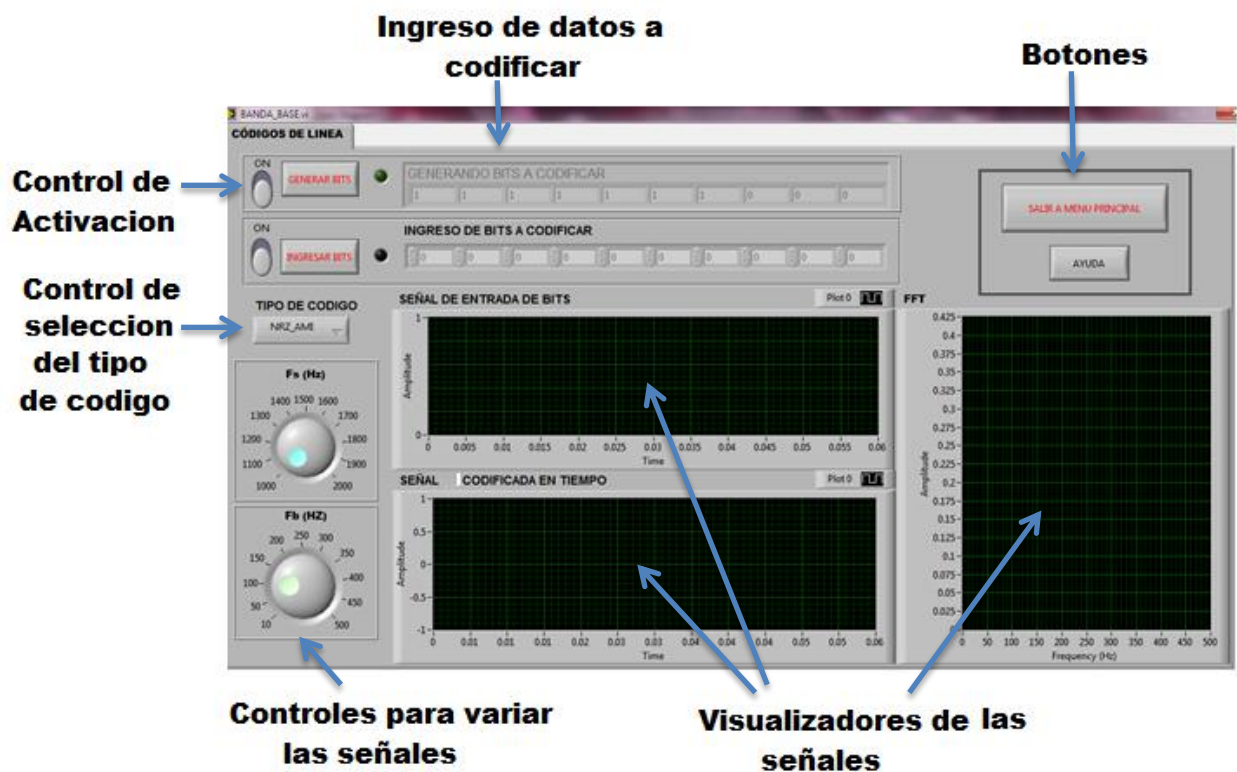


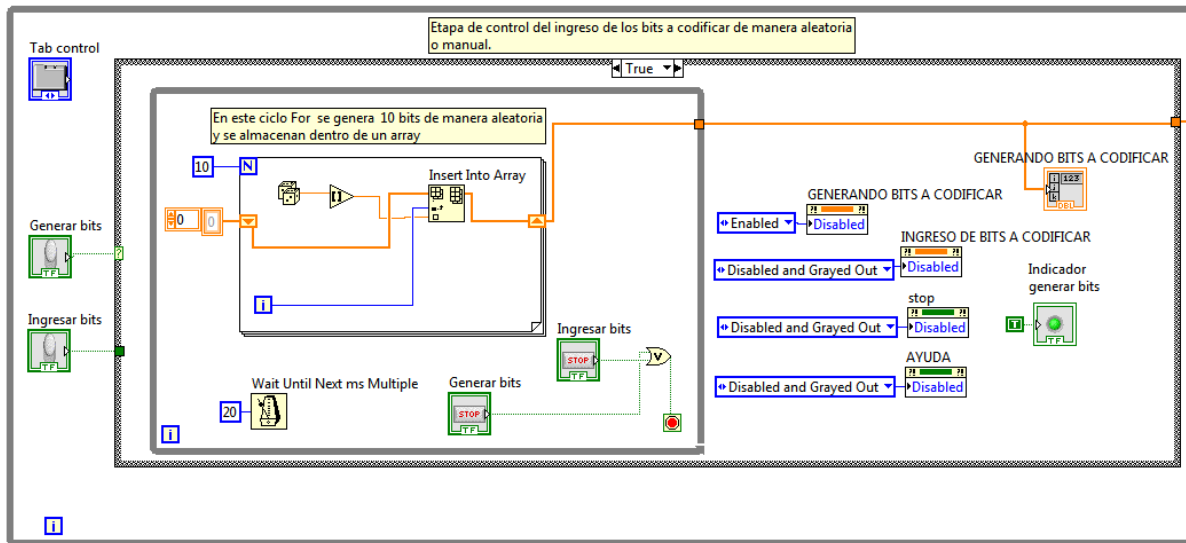
Figura 6.127: Panel Frontal del VI Principal Banda Base

Fuente: Por el investigador

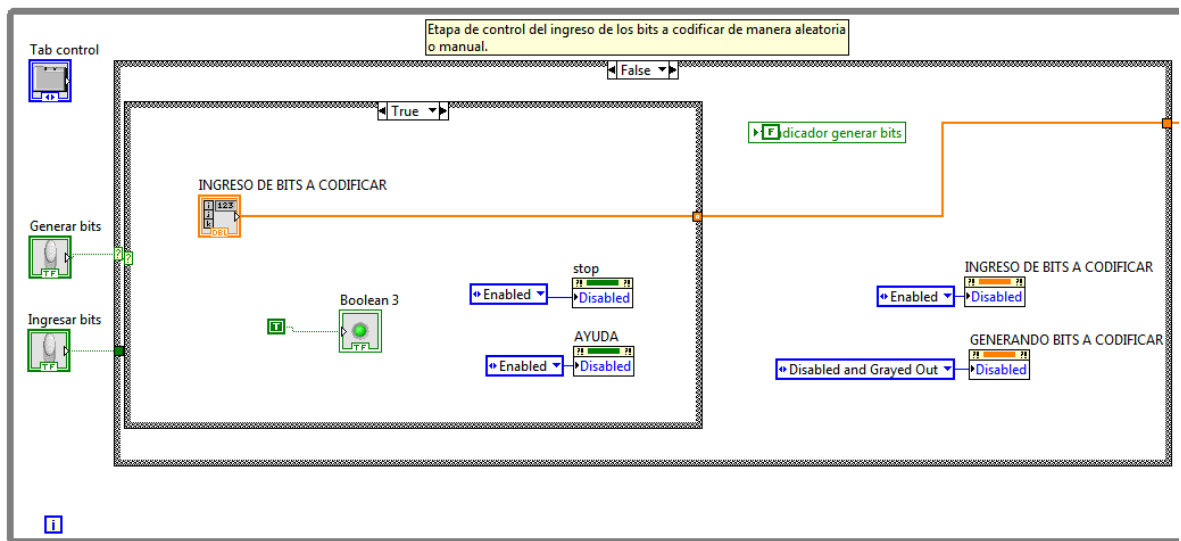
DIAGRAMA DE BLOQUES DEL VI PRINCIPAL PARA SEÑALES EN BANDA BASE.

La figura 128 muestra la primera etapa del simulador principal de señales banda base (códigos de línea), diseñado para el control del ingreso de 10 bits a codificar ya sea de manera aleatoria o en forma manual, según lo que el estudiante requiera para sus prácticas de laboratorio. En la figura 128a muestra la programación grafica del ingreso de los bits a codificar de manera aleatoria o randómica (*case true*). La figura 128b en cambio muestra la programación cuando se elige el ingreso manual de los 10 bits que se necesita codificar (*case false*). También se ha utilizado una

serie de controles (*Disable*) para activar o desactivar ciertos controles del panel frontal.



(a)



(b)

Figura 6.128: Control del ingreso de los bits a codificar: (a) aleatorio; (b) manual
Fuente: Por el investigador

La figura 6.129 muestra la continuación del simulador principal de señales banda base (códigos de línea) es decir la segunda parte, en donde para cada subVI se ingresan dos parámetros de entrada como son: la frecuencia de muestreo (f_s) y la frecuencia de bit (f_b). Se ha utilizado la función *case structure* para seleccionar el tipo de codificación de línea, mediante un control denominado *Menu Ring* (etiquetado como Tipo de Código). Además, se tiene la etapa de visualización de la

señal de entrada de bits, la señal codificada en tiempo y el espectro de frecuencia (FFT). Finalmente, se creo una etapa de ayuda en donde se encuentra información de cada tipo de código de línea.

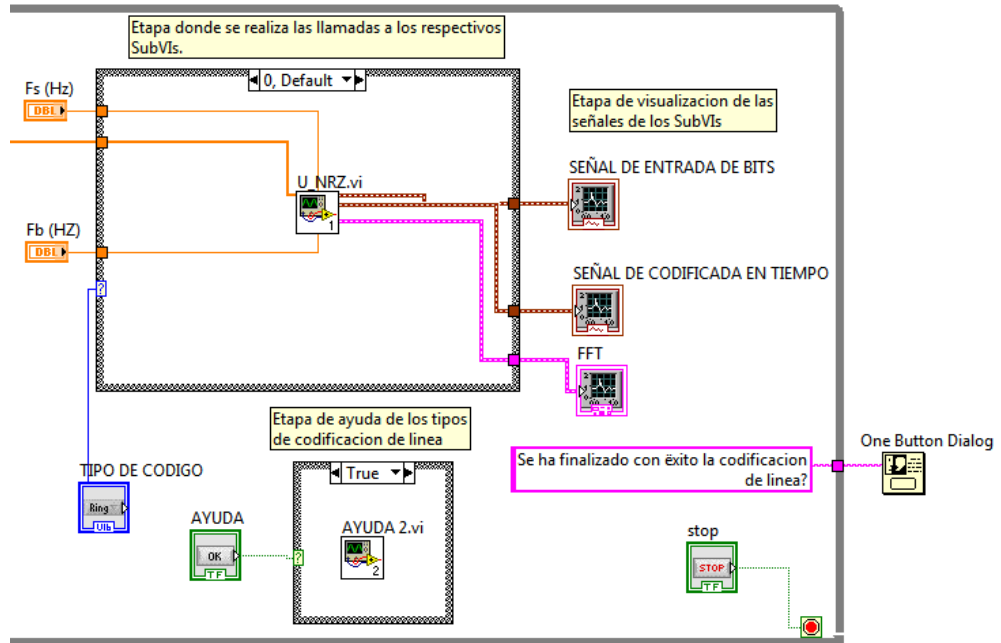
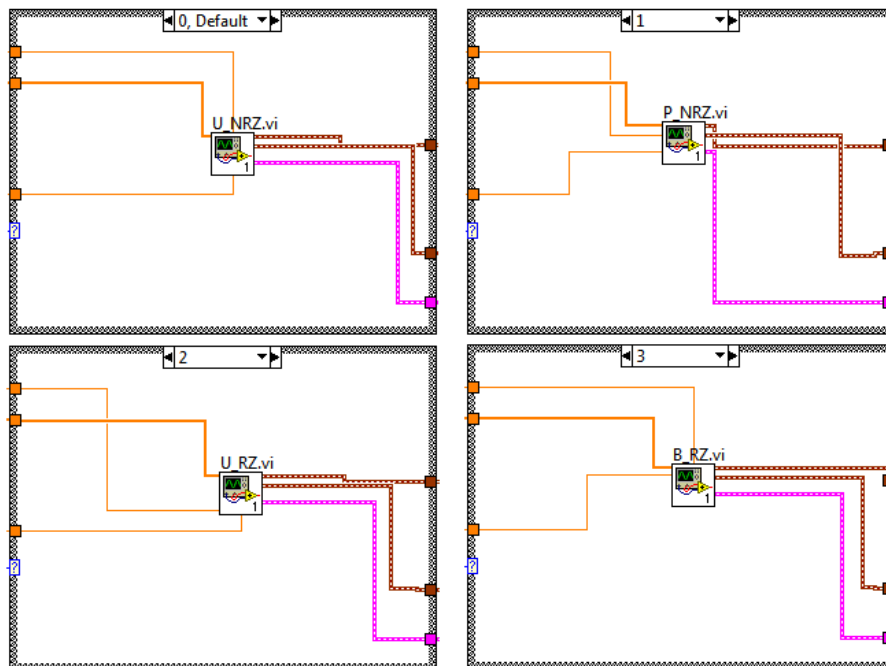


Figura 6. 129: Parámetros de entrada y salida de cada SubVI
Fuente: Por el investigador

En la figura 6.130 muestra la función *Case Structure* con los respectivos SubVIs dentro de cada *Case*.



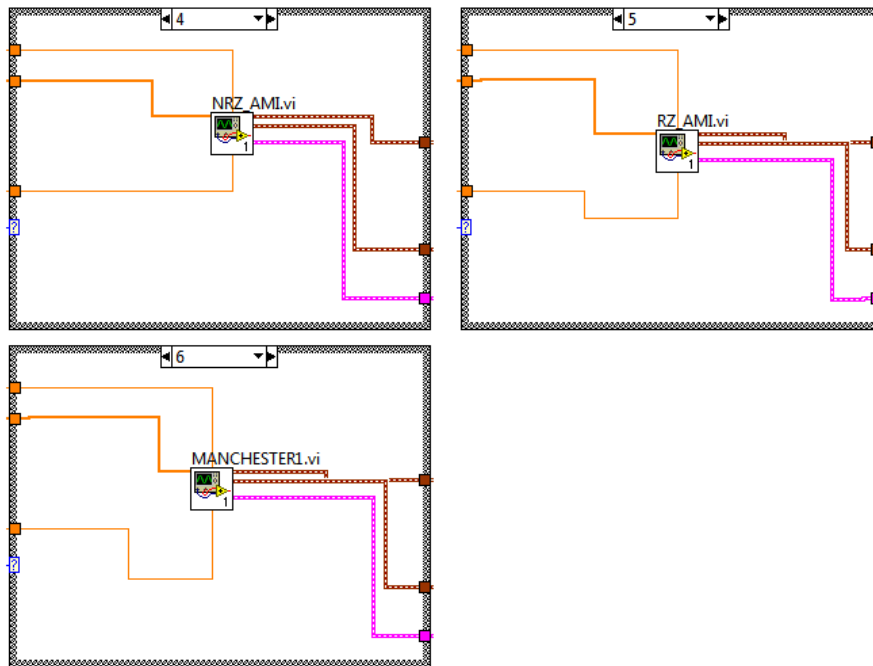


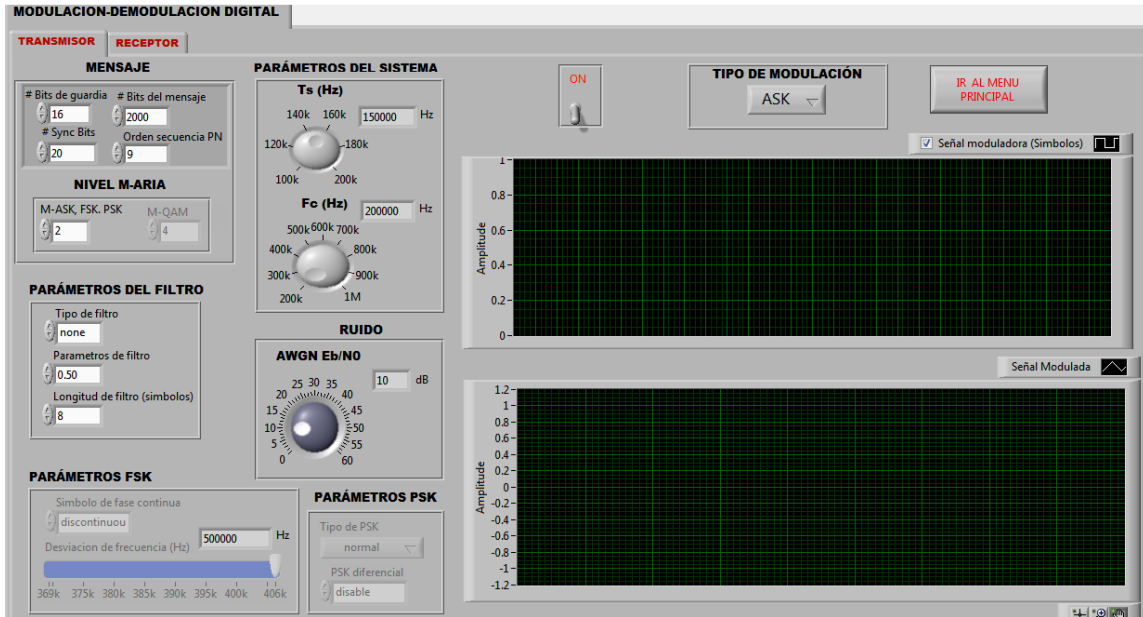
Figura 6.130: SubVIs dentro del Case Structure
Fuente: Por el investigador

PANEL FRONTAL DEL VI PRINCIPAL DE SEÑALES PASA BANDA.

Panel Frontal del instrumento virtual de señales Pasa Banda.

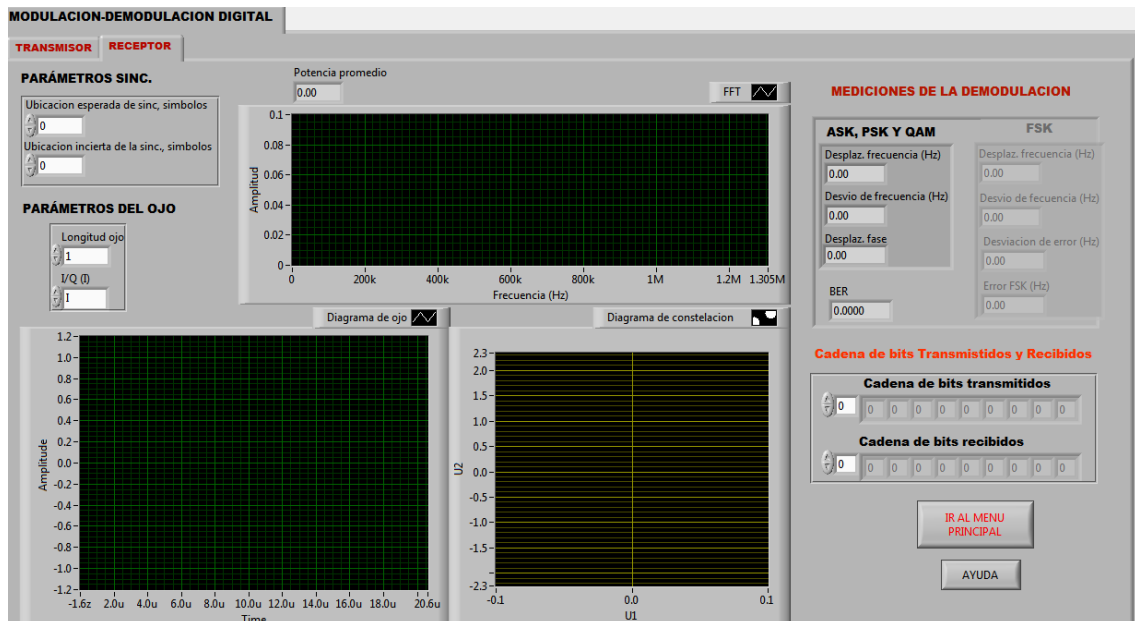
La figura 6.131 muestra el panel frontal del VI Principal de señales pasa banda en donde se puede simular las cuatro modulaciones digitales básicas como son: ASK, FSK, PSK y QAM. El VI Principal tiene los mismos elementos o parámetros de la modulación-demodulación ASK, por tal razón ya no se describen en esta parte. Sin embargo, se añadió cuatro controles: un control que permite seleccionar el **TIPO DE MODULACION**, un botón de encendido **ON**, un botón de **SALIR A MENU PRINCIPAL** y el botón **AYUDA** que muestra una ventana de ayuda con las características básicas de cada modulación diseñada. . La descripción detallada del VI principal, se encuentra en **Anexos 2**.

Transmisor



(a)

Receptor



(b)

Figura 6. 131: Panel Frontal del VI Principal Pasa Banda: (a) Transmisor; (b) Receptor

Fuente: Por el investigador

Diagrama de bloques del instrumento virtual de señales Pasa Banda.

El diagrama de bloques de este instrumento virtual de señales pasa banda fue diseñado con todos los parámetros para controlar las modulaciones digitales y obtener las respectivas mediciones de las demodulaciones. Los SubVIs son llamados mediante la función *Case Structure* y con un control llamado *Tipo de Código*. En la figura 6.132 muestra el presente diagrama de bloques del VI Principal de señales pasa banda, con los controles ubicados de manera ordenada a la izquierda del *Case Structure* y todos los indicadores a la derecha del mismo. También se tiene otro *case structure* controlado por un botón **AYUDA**, que visualice una ventana con la información de las modulaciones digitales. Por ultimo, se crea un ciclo **While Loop** para que el proceso se ejecute repetidamente hasta presionar el botón **Stop**.

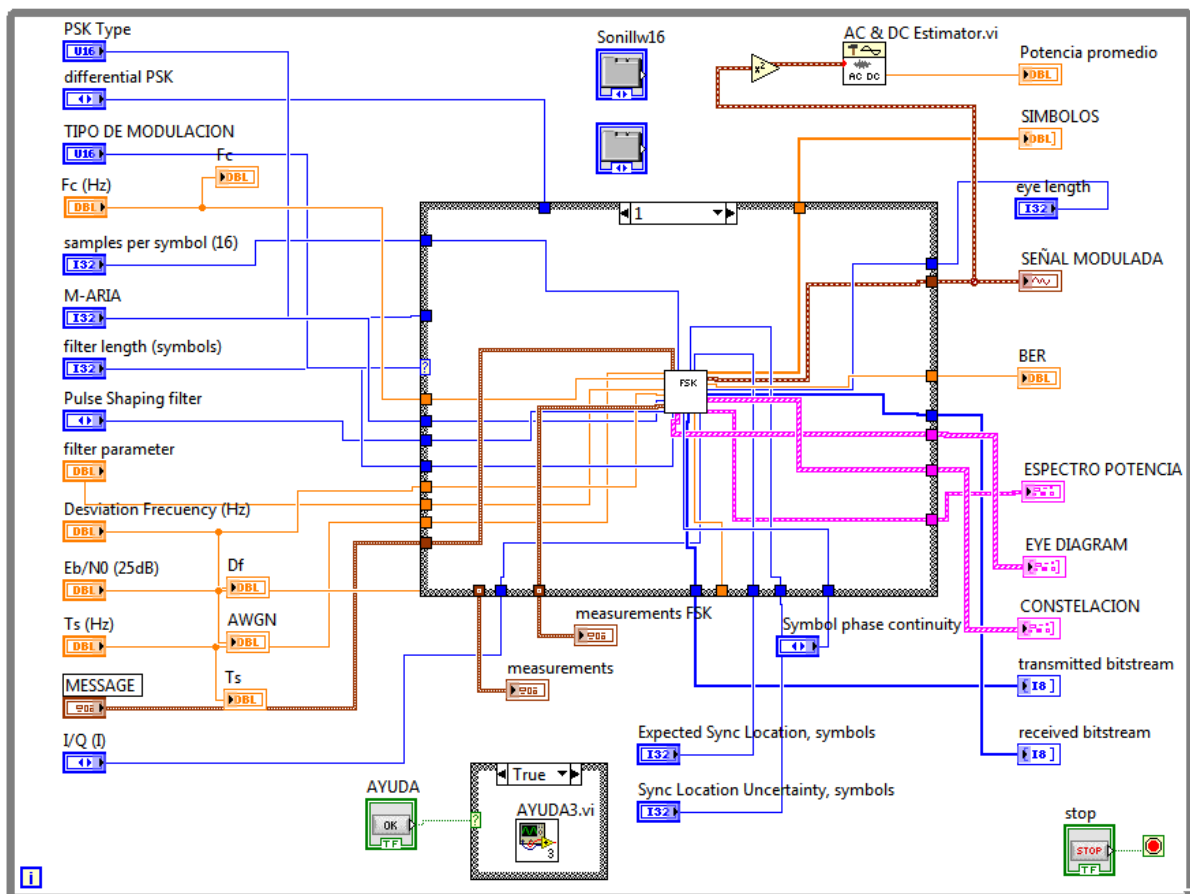


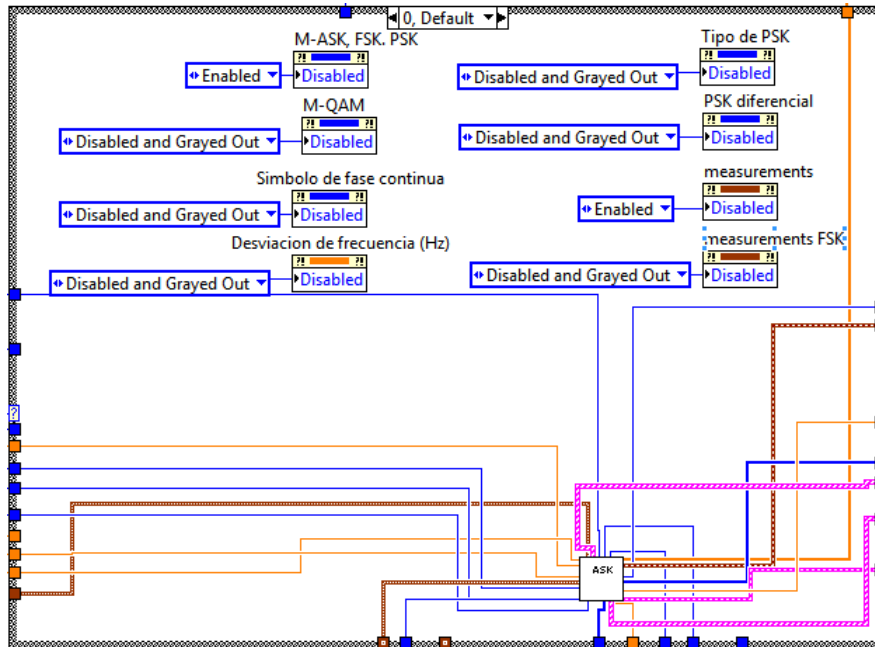
Figura 6.132: Diagrama en bloques del simulador de señales en Pasa Banda

Fuente: Por el investigador

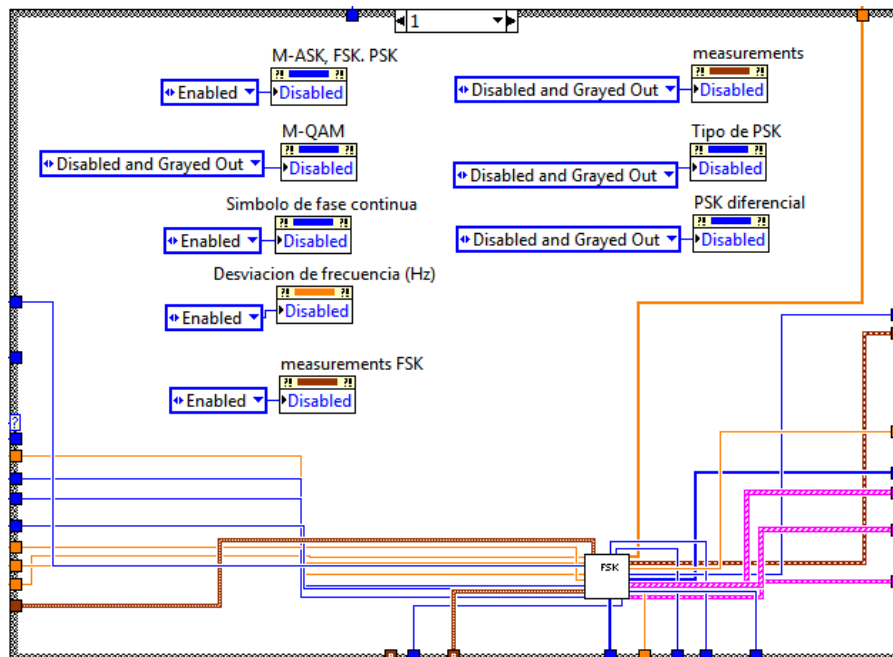
En la figura 6.133 muestra los SubVIs dentro de cada *Case* con unas funciones adicionales llamados *Disable* que permiten activar o desactivar algunos elementos

del VI Principal, según las condiciones que sean necesarios para cada tipo de modulación digital.

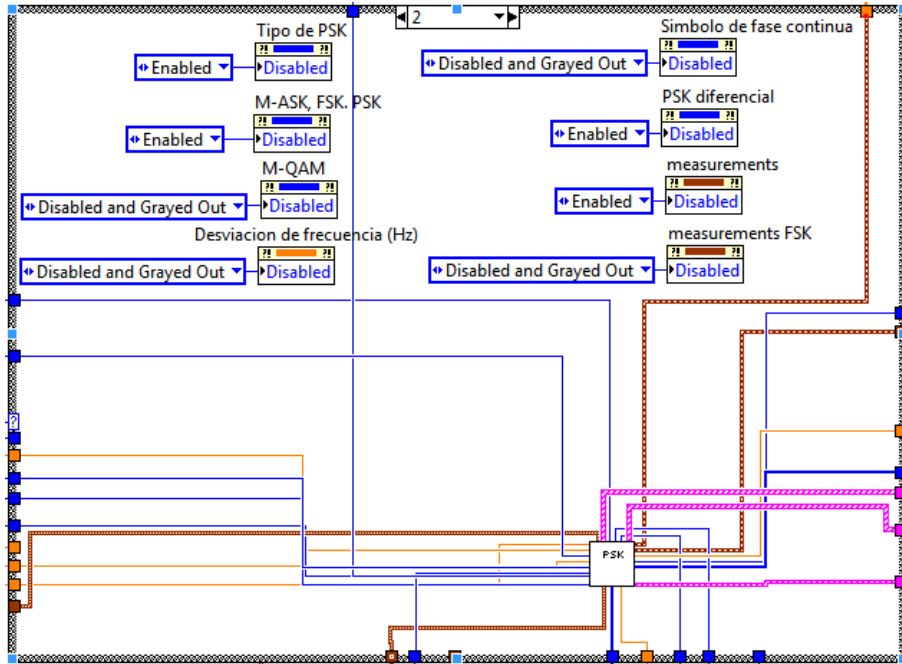
Case 0: ASK



Case 1: FSK



Case 2: PSK



Case 3: QAM

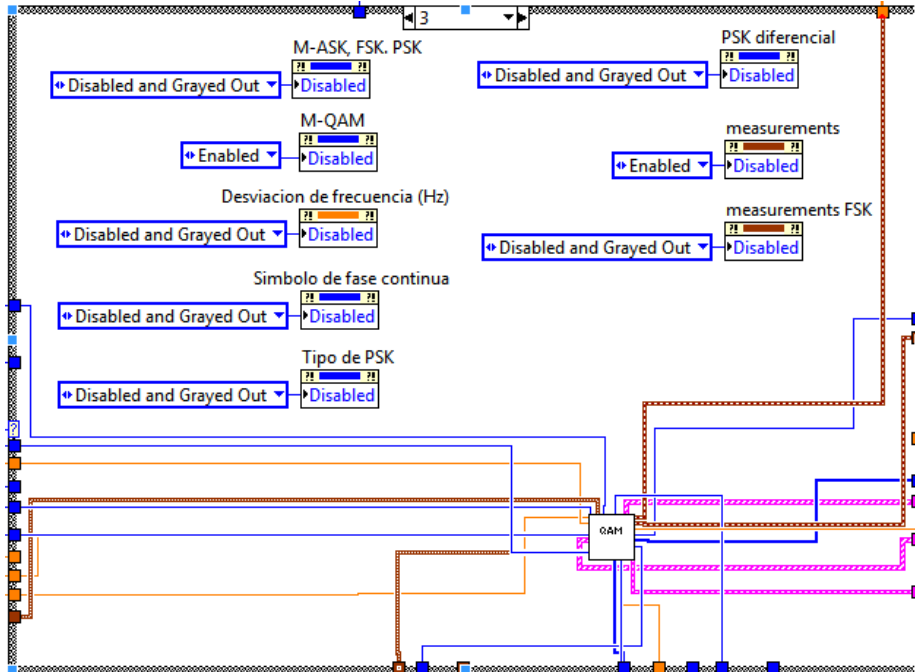


Figura 6.133: SubVIs dentro del Case Structure

Fuente: Por el investigador

6.9 Presupuesto

El costo del software labview 2011 (versión 11.0) no fue necesario debido a que la FISEI, de la Universidad Técnica de Ambato ya tiene este software con su respectiva licencia. Sin embargo, es importante conocer que la licencia de labview debe ser actualizado anualmente con un costo adicional como se muestra en la tabla 6.1.

Tabla 6.1: Costo de la licencia de labview
Fuente: Por el investigador

N	DETALLE	CANTIDAD	VALOR UNITARIO (USD)	VALOR ANUAL (USD)	VALOR TOTAL (USD)
1	Licencia de Labview 2011	1	3000	3000	3000

CAPITULO VII

7. CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Mediante el desarrollo de esta propuesta, se obtuvo una solución innovadora y de última tecnología, para el mejoramiento de las prácticas en los laboratorios de la FISEI.
- Las simulaciones de las técnicas de modulación-demodulación digital y las codificaciones de línea con sus respectivos elementos, permitieron obtener resultados óptimos en el momento de su ejecución.
- Al ejecutar el proyecto propuesto, se presentó un tiempo mínimo de retardo al visualizar los resultados.

RECOMENDACIONES

- Se recomienda el uso del software Labview como una herramienta capaz de resolver de alguna manera los problemas de carencia de equipos de laboratorio en la FISEI.
- Se debe utilizar la herramienta de ayuda que se tiene para cada aplicación del laboratorio virtual, con la finalidad de comprobar que los resultados de las simulaciones sean los correctos.
- Para la ejecución del laboratorio virtual, se necesita tener un computador con sistema operativo Window Xp o superior, memoria RAM 2GB y procesador Dual Core de 1.3 Ghz.

BIBLIOGRAFIA

- ✓ GABRIUNAS, V. Apuntes de electrónica. Publicada 1999.
- ✓ GUTIERREZ. José. Herramientas de ayuda al diseño simulación y control de procesos industriales. Publicado 1995.
- ✓ JONES, H. Virtual reality applications. Londres, Academic Press. Publicada (1995)
- ✓ LEON, W. Couch II. Sistemas de comunicaciones análogas y digitales Quinta Edición. Publicada 1998.
- ✓ LEOPOLDO, Javier. Proyecto de Grado: “Integración del Downconverter NI 5660 a través de instrumentos virtuales bajo labview para generar diseños de referencia y procedimientos aplicables a sistemas de telecomunicaciones”. Publicado 2010.
- ✓ MISCHA Swartz. Transmisión de información. Modulación y ruido.
- ✓ SANTA CRUZ, Oscar. Apuntes de Electrónica III. Publicada 2008.

LINKOGRAFIA

- APLICACIONES DE MATLAB. Publicado en Noviembre del 2009. http://www.lawebdelprogramador.com/foros/Matlab/1155014Aplicaciones_Matlab.html
- BRICEÑOJ. COMUNICACIONES DIGITALES. Publicado en Mayo del2010.<http://webdelprofesor.ula.ve/ingenieria/jbriceno/transmisiones/com diP2.pdf>
- BARNETO. García. TRABAJO DE INVESTIGACION MODULO III. http://sig.utpl.edu.ec/sigutpl/Staftpro/realidad/re_virtual.PDF
- CURSO BÁSICO DE LABVIEW. <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>
- INSTRUMENTACIÓN VIRTUAL CON LABVIEW. <http://laimbio08.escet.urjc.es/docencia/IE/Introduccion%20LabVIEW%20-%20Seis%20Horas.pdf>
- LABORATORIO VIRTUAL Modificada 14 de Enero del 2011. <http://www.tropinature.com/cvitjmn/publications/educdist/labvirt/evollab4.pdf>.

- MODULACION DIGITAL. http://www.slideshare.net/2unefa_grupo_0_02/modulaci3n-digital.
- MODULACI3N.<http://200.69.103.48/comunidad/profesores/jruiz/jairocd/texto/usm/cd/modulacion.pdf>
- MARCO ANTONIO JIMENEZ ESTRADA. TECNICAS DE MODULACION DIGITAL Publicado el 04 Octubre del 2010. <http://es.scribd.com/doc/38611884/Tecnicas-de-Modulacion-Digital>
- MODULACION. Modificada el 08 de Agosto del 2011.http://es.wikipedia.org/wiki/Modulaci3n_%28telecomunicaci3n%29
- MODULACION Y DEMODULACION. Publicado el 27 de Febrero del 2008. <http://raudivevoice.blogspot.es/1204114740/>
- MODULACION ANALOGA Y DIGITAL.<http://profesores.is.es/cuelaing.edu.co/asignaturas/rdc1-lab/Labs/Lab%205.pdf>
- Rivera. Javier. PROGRAMACI3N EN C. <http://aurea.es/wp-content/uploads/1-introduccionlenguajec.pdf>
- SEÑALES ANAL3GICAS Y DIGITALES. Modificada el 08 de Noviembre del 2011. <http://www.mitecnologico.com/Main/SeñalesAnalogicasYDigitales>
- SISTEMAS DE TELECOMUNICACIONES. <http://www.uv.es/hertz/hertz/Docencia/teoria/Trasmdigital.pdf>
- SANCHEZ. Carlos. CURSO DE UTILIZACION PRACTICA DE MATLAB. Publicado Julio del 2006. <http://biocomp.cnb.csic.es/~cos/s/Docencia/MATLAB/ApuntesGrande.pdf>
- TRANSMISION DE DATOS. <http://hcdsc.gov.ar/biblioteca/ISES/educacion/ciencias%20de%20la%20comunicacion/Transmision%20de%20datos.pdf>.
- TUTORIAL DE LABVIEW. http://www.gte.us.es/ASIGN/IE_4T/Tutorial%20de%20Labview.pdf
- TÉCNICAS DE MODULACI3N. <http://blearning.itmina.edu.mx/dep/sada/carreras/Ingenieria%20en%20Sistemas%20Computacionales/5to%20>

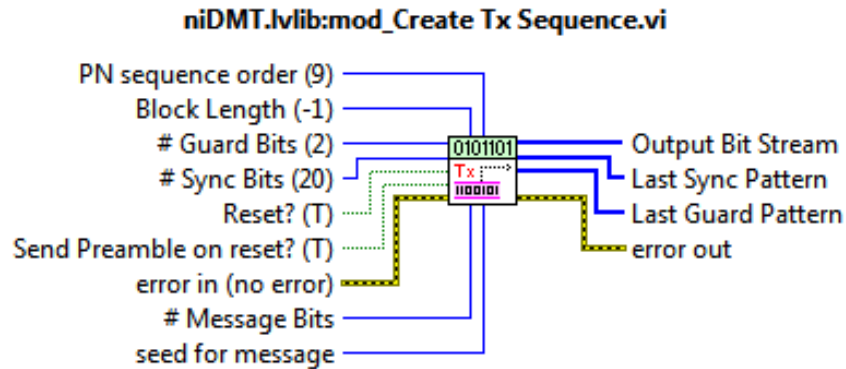
Semestre/Teoria%20de%20las%20Telecomunicaciones/teoria_telecomunicaciones/CAPITULO%20II.pdf

- TUTORIAL DE LABVIEW. http://www.gte.us.es/ASIGN/IE_4T/Tutorial%20de%20Labview.pdf
- WIKIPEDIA. Lenguaje C. Modificada el 31 de Julio del 2012. http://es.wikipedia.org/wiki/C_%28lenguaje_de_programaci%C3%B3n%29
- WIKIPEDIA. Lenguaje C#. Modificada el 30 Julio del 2012. http://es.wikipedia.org/wiki/C_Sharp

ANEXOS

ANEXO 1: Descripción de los terminales de cada uno de los VIs utilizados en el simulador de modulaciones digitales.

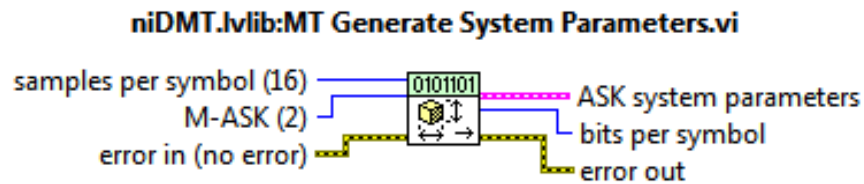
CREATE TX SEQUENCE.VI



Descripción de los terminales:

- **PN sequence order.-** genera una secuencia de bits ordenados en dependencia de los parámetros como: numero de bits de guardia, sincronismo de bits y los bits del mensaje.
- **Block length.-** es un terminal para el bloque de longitud.
- **# Guard Bits.-** numero de bits de guardia.
- **# Sync Bits.-** permite el sincronismo de los bits generados.
- **Reset?.-** reinicia el VI.
- **Send Preamble on reset?.-** envía el preámbulo de reinicio.
- **#Message Bits.-** numero de bits de mensaje.
- **Seed for message.-** muestras para el mensaje.
- **Output bit stream.-** salida de la cadena de bits.
- **Last Sync Pattern.-** patrón de la ultima sincronización.
- **Last Guard Pattern.-** patrón de la ultima guardia.
- **Error in y out.-** permite corregir errores de entrada y salida.

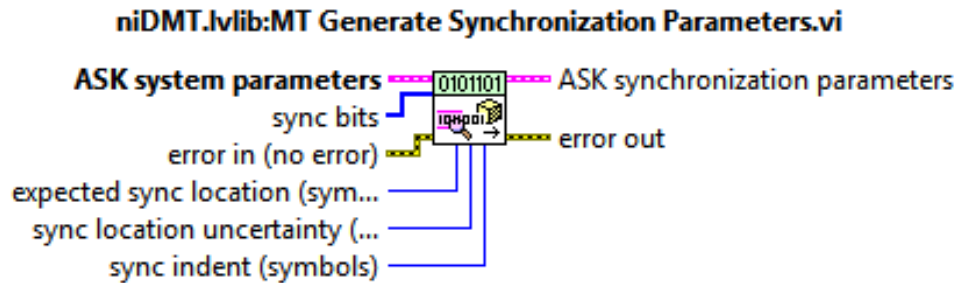
MT GENERATE SYSTEM PARAMETER ASK .VI



Descripción de los terminales:

- **samples per symbol.-** especifica un número par positivo, de las muestras dedicadas a cada símbolo. Multiplica este valor por la velocidad de símbolos para determinar la frecuencia de muestreo. El valor predeterminado es 16.
- **M-ASK.-** especifica el número *M-aria*, que es el número de estados distintos, que representan los símbolos en la forma de onda compleja, modulada en banda base. El valor por defecto es 2.
- **Error in** (sin errores), puede aceptar la información, error de cableado, de anteriores VIs. Utilice esta información, para decidir si la funcionalidad debe ser anulada en el caso de los errores de otros VIs.
- **ASK System Parameters.-** devuelve los valores de los parámetros, que definen el sistema ASK. Conecte este grupo de parámetros del sistema a los parámetros de la *MT Modulate ASK.vi* o el *MT Demodulate ASK.vi*.
- **bits per symbol.-** devuelve el número de bits representados por cada símbolo. Este valor es igual a $\text{Log}_2(M)$, donde M es el orden de la modulación (por ejemplo, para 16-ASK, $M = 16$).
- **error out.-** pasa la información de error o advertencia de un VI, a otros VIs.

MT GENERATE SYNCHRONIZATION PARAMETERS ASK.VI



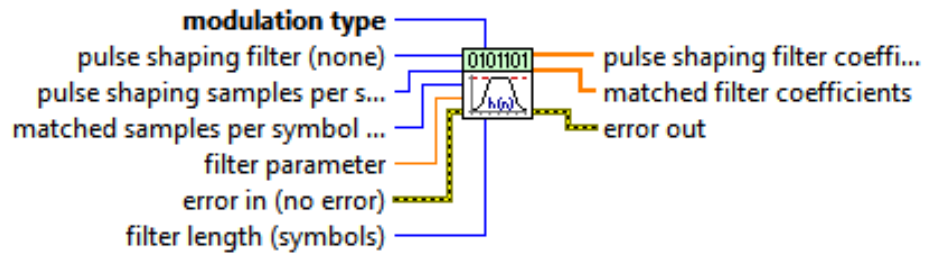
Descripción de los terminales:

- **ASK system parameters.**- especifica los valores de los parámetros del sistema ASK. Se conecta con el *Mt Generate Parameter System ASK.vi* para retornar los parámetros del sistema a este VI.
- **sync bits.**- especifica un array de bits de sincronización, que se utilizan para generar el grupo de parámetros de sincronización ASK. El valor predeterminado es 0.
- **expected sync location.**- especifica la ubicación prevista de los parámetros de sincronización de símbolos. El valor predeterminado es -1.
- **sync location uncertainty.**- especifica el número de símbolos antes o después de la ubicación de la sincronización, que podría ser tal vez, en el primer símbolo de la secuencia de sincronización situado. El valor por defecto es 10
- **sync indent.**- especifica la distancia que la secuencia de sincronización tiene sangría en el bloque de información. La distancia es el número de símbolos demoduladas, procedentes de la secuencia de sincronización.
- **ASK synchronization parameters.**- devuelve parámetros de sincronización para pasar a la demodulación.

Nota: El VI polimórfico, elegido es de tipo **bits array**, debido a que necesitamos trabajar con bits, no con matriz de número (**number array**). Los terminales: **error in** y **error out**, ya no se describe, por cumplir las mismas funciones para todos los VI.

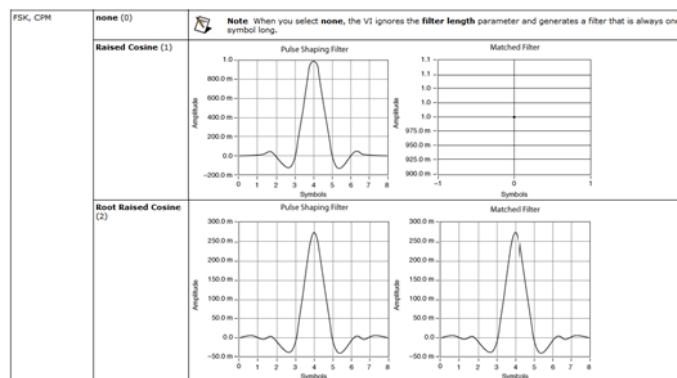
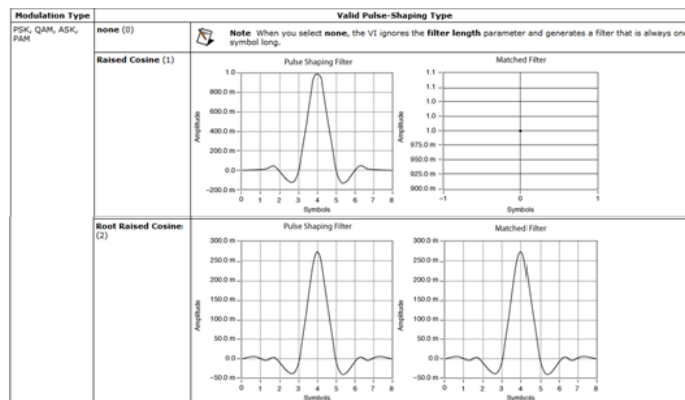
MT GENERATE FILTER COEFFICIENTS

niDMT.lvlib:MT Generate Filter Coefficients.vi



Descripción de los terminales:

- **modulation type.**- selecciona el tipo de modulación ASK, FSK,...QAM.
- **pulse shaping filter (none).**- especifica la clase de filtro a generar. La selección aquí, debe tener relación, con el tipo de modulación a usar como se muestra en la tabla siguiente. El valor predeterminado es **none**.



Nota: En nuestro proyecto, se ha utilizado los siguientes tipos de filtro como son: none, Raised Cosine y Root Raised Cosine.

- **pulse shaping samples per symbol.-** especifica el número de muestras deseadas por símbolo para el *pulse shaping filter*. Si el *pulse shaping filter*, es usada para la demodulación, este valor del parámetro debe coincidir con *pulse shaping samples per symbol*, de los parámetros en *demodulate.vi*. El valor predeterminado es 16.
- **matched samples per symbol.-** especifica el número de muestras deseadas por símbolo para la demodulación filtro adaptado. Este valor de parámetro debe coincidir con los parámetros del *samples per symbol*, para el *demodulate digital.vi*. El valor predeterminado es 16.
- **filter parameter.-** especifica ya sea alfa (atenuación de coseno elevado y el filtro raíz cuadrada de coseno elevado), o BT (el producto del ancho de banda -3 dB y el período de símbolo para un filtro Gaussiano). Este parámetro se ignora cuando el *pulse shaping filter parameter* es puesto en none. El valor predeterminado es 0,5.
- **filter length.-** especifica la longitud deseada del *pulse shaping filter*, en símbolos. La longitud del filtro adaptado puede variar según el tipo de filtro y el tipo de modulación. Los valores válidos son > 1 . El valor predeterminado es 8.
- **pulse shaping filter coefficient.-** devuelve un array ordenado de coeficientes del filtro correspondiente a la respuesta deseada del filtro para el *pulse shaping filter* usado en la modulación. El número de coeficientes depende de las características del *pulse shaping filter*, especificadas en el *pulse shaping filter parameter*.
- **matched filter coefficients.-** devuelve un conjunto ordenado de coeficientes de filtro, correspondiente al filtro deseado para la demodulación.

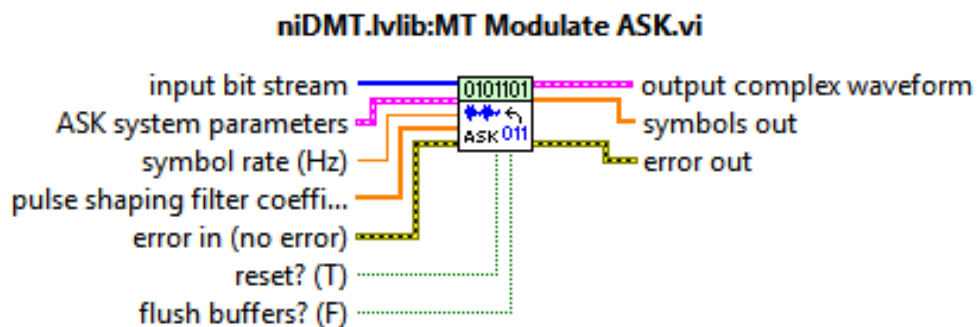
Los cálculos de los coeficientes de filtro son usados durante la modulación para reducir el ancho de banda de la señal transmitida y durante la demodulación para reducir la interferencia entre símbolos. El *pulse shaping filter* puede ser utilizado tanto en la transmisión o para la demodulación de señales moduladas PSK, FSK, MSK, ASK, PAM, QAM y CPM. El *matched filter*, sólo se utiliza para la demodulación.

El VI, calcula la respuesta impulsiva del filtro utilizando las siguientes fórmulas:

Filter Type	Formula
Raised Cosine	$h(t) = \text{sinc}\left(\frac{t}{T}\right) \frac{\cos\left(\frac{\pi\alpha t}{T}\right)}{1 - \frac{4\pi^2\alpha^2 t^2}{T^2}}$ <p>where T is the symbol period and α is the filter rolloff factor.</p>
Root Raised Cosine	$h(t) = \begin{cases} 1 - \alpha + \frac{4\alpha}{\pi}, & t = 0 \\ \frac{\alpha}{\sqrt{2}} \left[\left(1 + \frac{2}{\pi}\right) \sin\left(\frac{\pi}{4\alpha}\right) + \left(1 - \frac{2}{\pi}\right) \cos\left(\frac{\pi}{4\alpha}\right) \right], & t = \pm \frac{T}{4\alpha} \\ \frac{\sin\left(\pi \frac{t}{T} (1 - \alpha)\right) + 4\alpha \frac{t}{T} \cos\left(\pi \frac{t}{T} (1 + \alpha)\right)}{\pi \frac{t}{T} \left(1 - \left(\frac{4\alpha t}{T}\right)^2\right)}, & \text{otherwise} \end{cases}$ <p>where T is the symbol period and α is the filter rolloff factor.</p>

MT MODULATE ASK

En la modulación ASK, implica el mapeo de la cadena de bits de entrada para los símbolos ASK, interpolando por la inserción de cero, y aplicando el *pulse shaping filter* para los datos interpolados I / Q.



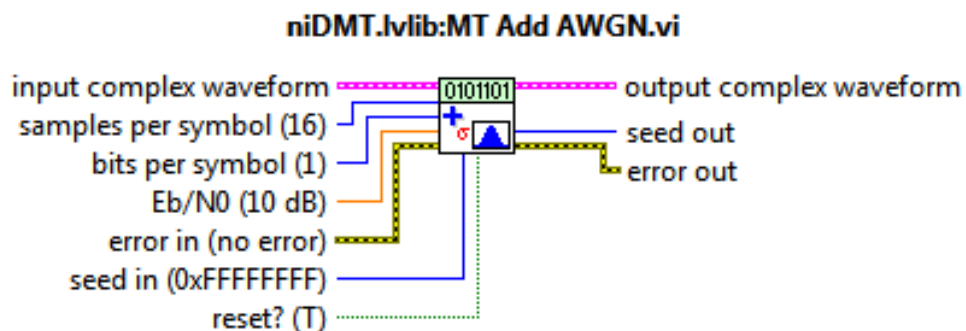
Descripción de los terminales:

- **input bit stream.**- especifica la secuencia de bits de información a ser modulada.
- **ASK system parameters.**- especifica los valores de los parámetros que definen el sistema ASK. Se debe conectar los cluster del *ASK system parameters* al *MT Generate System Parameter.vi*. No altere los valores.

- **symbol rate.**- especifica la velocidad del símbolo que desea, en hertz (Hz). Este valor debe ser positivo. El valor predeterminado es 1,00.
- **pulse shaping filter coefficient.**- especifica un array ordenado que contiene los coeficientes de filtro para el *pulse shaping filter*. Conecte el *pulse shaping filter coefficients* del **MT Generate filter.vi** a estos parámetros. Cuando generemos los coeficientes del filtro, asegúrese que el valor de los parámetros del *pulse shaping samples per symbol* del **MT Generate filter coefficients.vi**, sea igual al valor del *samples per symbol* del **ASK System Parameter**.
- **flush buffers.**- especifica si las muestras son forzados a salir de la forma de onda modulada que se ven afectados por el pulse shaping filter de retardo. El valor predeterminado es FALSE.
- **output complex waveform.**- devuelve la forma de onda ASK, modulado en banda base.
- **symbols out.**- devuelve el array de símbolos asignados antes de *pulse shaping* aplicado. El array representa el valor complejo de cada símbolo asignado.

MT ADD AWGN.vi

Este VI retorna una forma de onda, señal+ruido, con un E_b/N_0 especificado por el usuario, donde E_b representa la energía por bit, y N_0 representa la varianza del ruido.



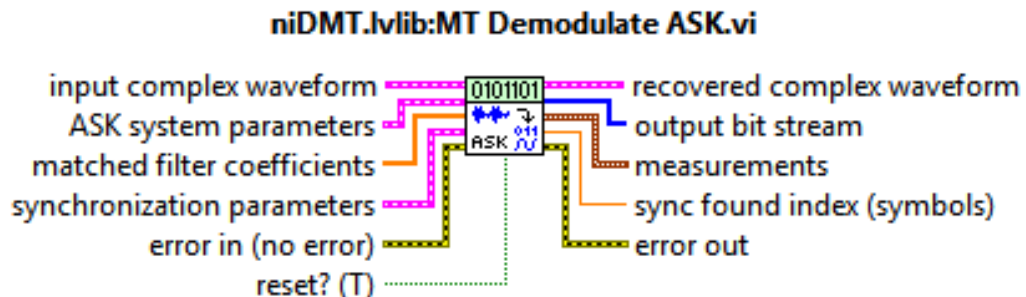
Descripción de los terminales:

- **input complex waveform.**- especifica los datos modulados de la forma de onda compleja, en banda base.

- **samples per symbol (16).**- especifica el número de muestras por la información del símbolo en la entrada, de la forma de onda compleja. El valor por defecto es 16.
- **bits per symbol.**- especifica el número de bits por símbolo y se conecta al *MT Generate System Parameter*. Por ejemplo, el número de *bits per symbol* en la modulación por desplazamiento de fase en cuadratura (QPSK) es 2. El valor defecto es 1.
- **Eb/N0 (10 dB).**- especifica el Eb/N0 deseado, de la onda de salida compleja en dB. El valor por defecto es de 10.0 dB.
- **Seed in (0xFFFFFFFF).**- especifica cómo el generador de muestra de ruido es reservado. Cuando el valor del parámetro es > 0, el generador de AWGN es regenerado. El valor por defecto es 0xFFFFFFFF (-1).
- **output complex waveform.**- devuelve la señal-ruido en los datos de la forma de onda en banda base.

MT DEMODULATE ASK

Este VI, intenta remover la portadora y el desplazamiento de fase mediante el bloqueo de la señal portadora.

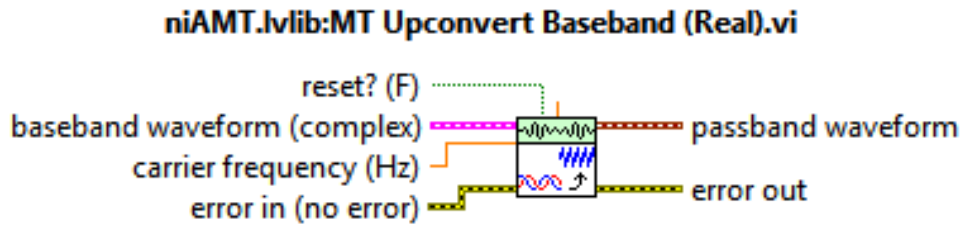


Descripción de los terminales:

- **input complex waveform.**- especifica la forma de onda compleja, de la modulación ASK, con datos en banda base.
- **ASK system parameters.**- especifica los valores que definen los parámetros del sistema ASK. Conecte este terminal al *ASK system parameters* del *MT Generate System Parameters*. No alterar los valores.

- **matched filter coefficients.**- especifica un array ordenado, conteniendo los *matched filter coefficients*, deseados. Conecte el *matched filter coefficients parameter* del *MT Generate Filter Coefficients.vi*, a este parámetro. Cuando vaya generando los *filter coefficients*, asegurese que el valor del *matched samples per symbol parameter* del *MT Generate Filter Coefficients.vi*, sea igual al valor de los elementos del *samples per symbol* del *ASK system parameters*.
- **synchronization parameters.**- especifica los valores de los parámetros, describiendo la secuencia de sincronización y el rango de los bits sobre la cual buscar esta secuencia.
- **recovered complex waveform.**- devuelve la corrección de la alineación de tiempo y sobremuestreo de datos, de la forma de onda compleja, después del filtro adaptado. El desplazamiento de frecuencia y correcciones de fase de compensación, son valores escalares aplicadas a todo el bloque.
- **output bit stream.**- devuelve la secuencia de bits demodulados, de la información.
- **measurements.**- devuelve los resultados de las mediciones realizadas por el demodulador ASK.
 - **frequency offset.**- devuelve la frecuencia de la portadora de la medición de offset, en hertz (Hz). El desplazamiento de frecuencia medida, se elimina de la forma de onda compleja recuperada.
 - **frequency drift.**- devuelve la medición de la desviación de la frecuencia portadora, en hertz (Hz).
 - **phase offset.**- retorna la fase medido en grados. La medición phase offset, se elimina de la forma de onda compleja recuperada.
- **sync found index (symbols).**- devuelve el índice símbolo dentro de la forma de onda complejo de entrada, donde se encontró la correlación máxima de la secuencia de sincronización pico.

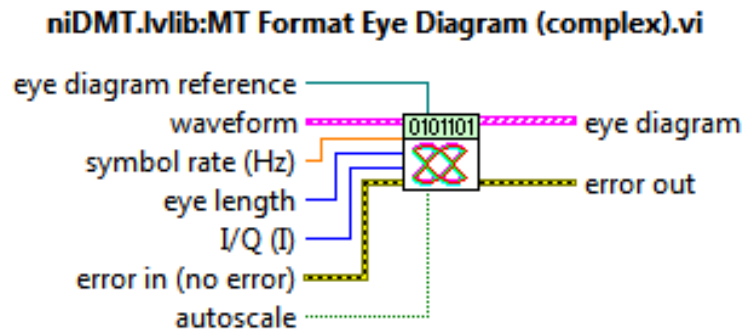
MT UPCONVERTER BASEBAND (Real).vi



Descripción de los terminales:

- **baseband waveform.-** especifica los datos de la señal compleja envolvente para la conversión ascendente, a su equivalente en pasa banda.
- **carrier frequency.-** especifica la frecuencia central pasa banda, en hertz (Hz). La señal entrante se convierte en el centro alrededor de esta frecuencia.
- **passband waveform.-** devuelve la señal convertida en pasabanda.

MT FORMAT EYE DIAGRAM

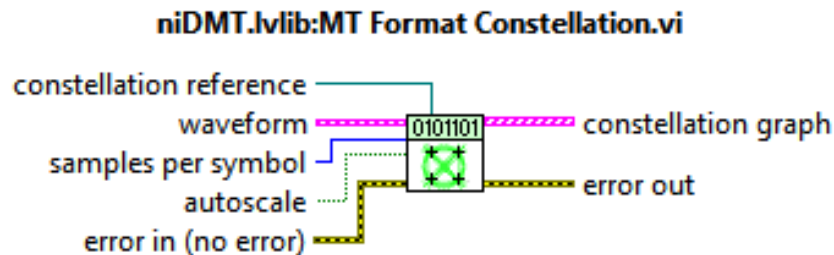


Descripción de los terminales:

- **eye diagram reference.-** es la referencia para el gráfico conectado al parámetro del diagrama de ojo.
- **waveform.-** especifica los datos recuperados de la modulación, para trazar el diagrama de ojo.

- **symbol rate (Hz).**- especifica la velocidad de símbolo que desea, en hertz (Hz). El valor predeterminado es 1.0.
- **eye length.**- especifica el número de periodos del símbolo en la escala horizontal del diagrama de ojo. El valor predeterminado es 1.
- **I/Q.**- especifica si debe generarse I de datos o Q para el diagrama de ojo.
- **autoscale.**- especifica si se debe aplicar auto escala a la gráfica que se hace referencia en el control del diagrama de referencia. El valor por defecto, es false (sin auto escala).
- **eye diagram.**- devuelve una matriz de segmentos de la forma de onda para la visualización del diagrama de ojos. Cada segmento tiene una longitud de tiempo (longitud del ojo x período de símbolo) segundos. Cablee este parámetro directamente a un waveform graph.

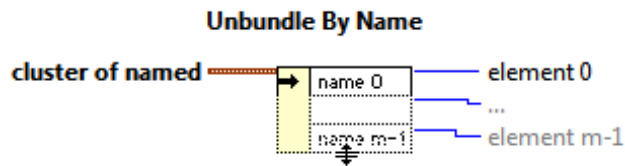
MT FORMAT CONSTELLATION.vi



Descripción de los terminales:

- **constellation reference.**- especifica una referencia al cableado del parámetro gráfico de constelación.
- **waveform.**- especifica los datos recuperados de la modulación, para trazar como una constelación.
- **samples per symbol.**- especifica el número de muestras por símbolo en la forma de onda recuperada. El valor por defecto es 0.
- **autoescale.**- Especifica si se debe aplicar autoescala a la gráfica referida en el control de referencia, de la constelación. El valor predeterminado es falso (no automático de escala).
- **constellation graph.**- devuelve una matriz de dos elementos de datos XY. El primer elemento es la constelación, el segundo elemento es los datos de sobremuestreo. Cablee este parámetro directamente a un control gráfico XY.

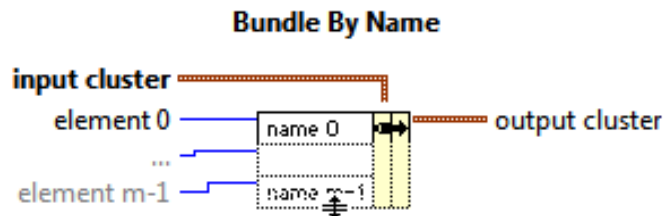
UNBUNDLE BY NAME



Descripción de los terminales:

- **cluster of named:** es el clúster que contiene los elementos al que desea acceder.
- **element:** es el elemento de la entrada del clúster llamado con el nombre de 0 .. m-1. Puede acceder a los elementos sólo con las propias etiquetas. Para seleccionar el nombre del elemento, haga clic en una terminal de nombre y seleccione un nombre del menú contextual.

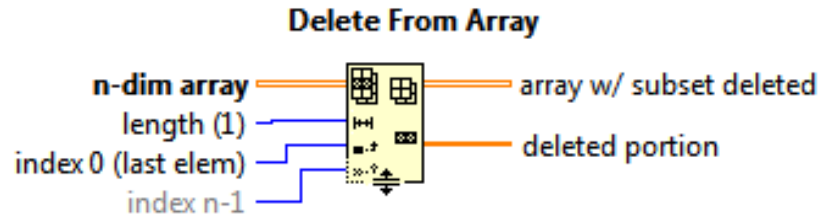
BUNDLE BY NAME



Descripción de los terminales:

- **input cluster:** es el clúster cuyos elementos se desea reemplazar. Al menos un elemento de la entrada del clúster, debe tener propia etiqueta.
- **element:** son los elementos de la entrada del clúster, que se desea reemplazar por el nombre. Sólo se puede sustituir los elementos con sus propias etiquetas.
- **output cluster:** es el clúster, cuyo valor es el clúster resultante.

DELETE FROM ARRAY

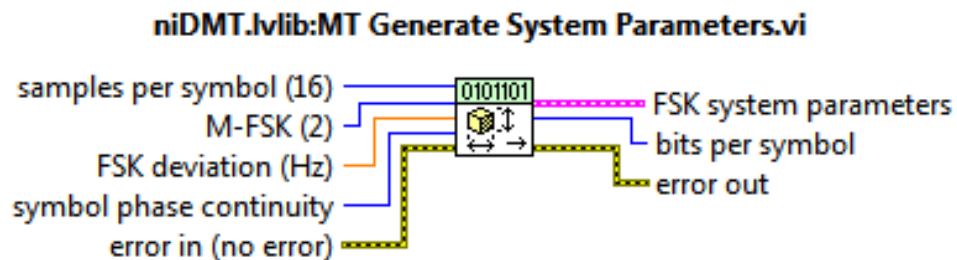


Descripción de los terminales:

- **n-dim array** es el array de la que desea eliminar el elemento (s), fila (s), columna (s), página (s), y así sucesivamente. Esta entrada puede ser un arreglo de n dimensiones de cualquier tipo.
- **length** determina cuántos elementos, filas, columnas o páginas para eliminar.
- **index 0..n-1** especifica lo que se desea eliminar del array, como un elemento, fila, columna o página. Un arreglo de funciones en los datos de LabVIEW accede a un array en orden de filas.
- **array w/ subset deleted** es el array devuelta, con el elemento eliminado, fila (s), columna (s), o página (s).
- **deleted portion** es el elemento eliminado o array.

Nota: Los VIs que no se describen sus terminales, es debido a que cumplen con las mismas características de los anteriores instrumentos estudiados.

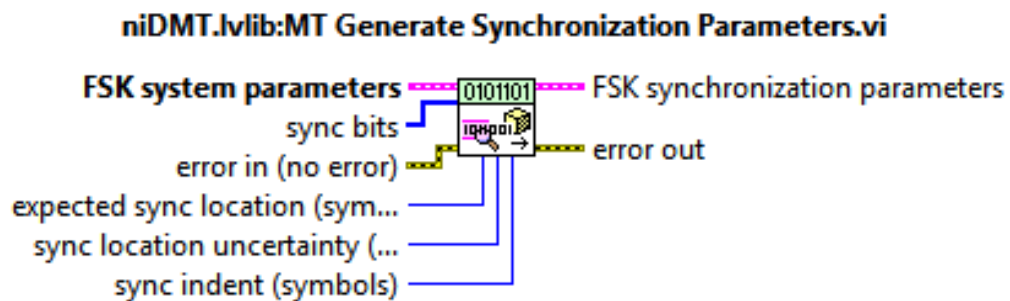
MT GENERATE SYSTEM PARAMETERS FSK.VI



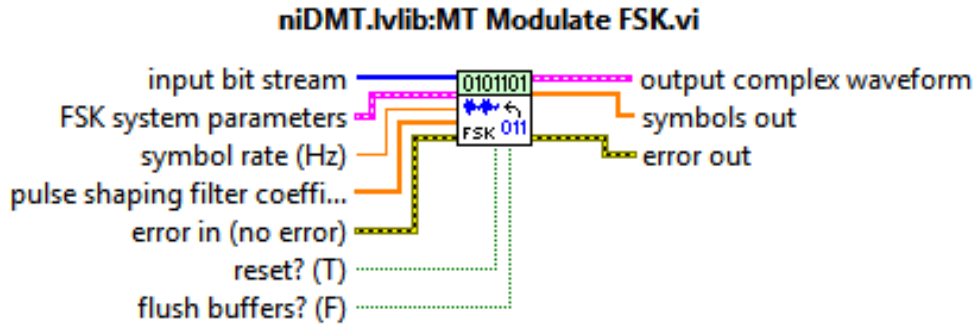
Descripción de los terminales:

- **M-FSK (2):** especifica el número M-aria, que es el número de distintas desviaciones de frecuencia, para usar como símbolos. Este valor debe ser una fuente positiva de 2. El valor predeterminado es 2.
- **FSK deviation (Hz):** especifica la máxima desviación de la frecuencia FSK. En las frecuencias de banda de base, las desviaciones de los símbolos individuales se espacian uniformemente en el intervalo $[-f_d, f_d]$, donde f_d representa la desviación de frecuencia. El valor predeterminado es de 15.000.
- **symbol phase continuity:** especifica si las transiciones de fase entre los símbolos son continuas:
 - **continuous (0):** Especifica las transiciones de fase continua entre los símbolos. Este valor es el valor predeterminado.
 - **discontinuous (1):** Especifica transiciones discontinuas de fase entre símbolos, es decir, DPFSK.

MT GENERATE SYNCHRONIZATION PARAMETERS FSK.VI

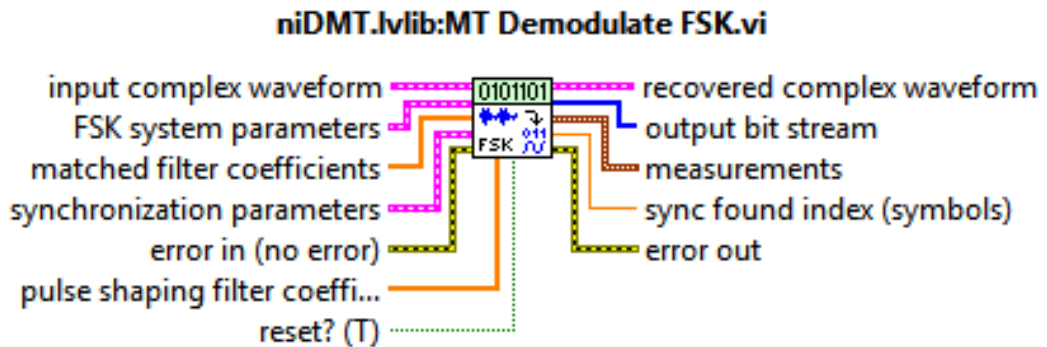


MT MODULATE FSK.VI



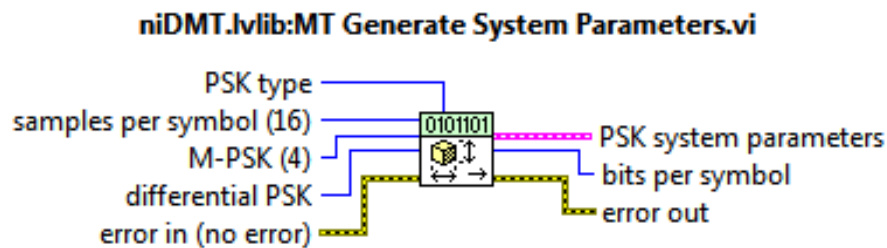
MT DEMODULATE FSK.VI

Este VI tiene un terminal más que el anterior demodulador ASK y es el *pulse shaping filter coefficients*: especifica un array ordenado que contiene el pulse shaping filter coefficients deseado. Este parámetro se utiliza internamente para reproducir la forma de onda ideal para hacer las mediciones.



MT GENERATE SYSTEM PARAMETER PSK.VI

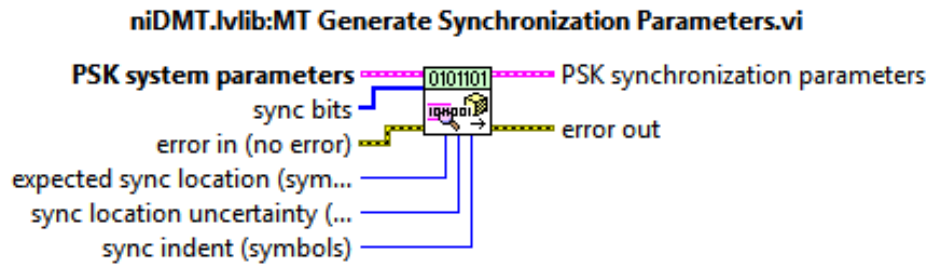
En la Modulación y demodulación PSK, también se utilizarán los mismos elementos o VIs con la diferencia que el selector polimórfico sea PSK. Algunos terminales cumplen con las mismas descripciones del ASK, y los que haga falta, lo explicaremos a continuación.



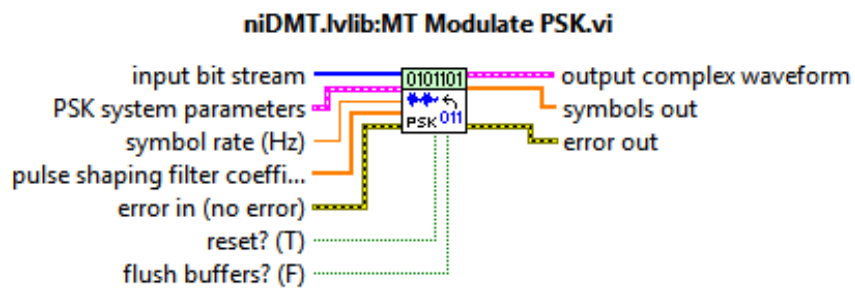
Descripción de terminales:

- **M-PSK (4)** especifica el número *M-aria*, que es el número de estados distintos, que representan los símbolos en la forma de onda compleja, modulada en banda base. El valor por defecto es 4.
- **differential PSK** especifica cómo la modulación PSK representa símbolos. La operación diferencial se utiliza para implementar formatos PSK tales como diferencial de cuadratura PSK (DQPSK) y $\pi / 4$ DQPSK.
 - **disable** (0) Los símbolos se representan como puntos de la constelación. Este valor es predeterminado.
 - **enable** (1) Los símbolos se representan como las transiciones entre puntos de la constelación.
- **PSK type** especifica el tipo de modulación PSK.
 - **Normal:** establece el tipo de modulación para PSK regular.
 - **Shifted:** gira la constelación por π / M cada símbolo.
 - **Offset:** Este esquema de modulación es una forma de manipulación por desplazamiento de fase en la que cuatro ángulos diferentes de fase se utilizan.

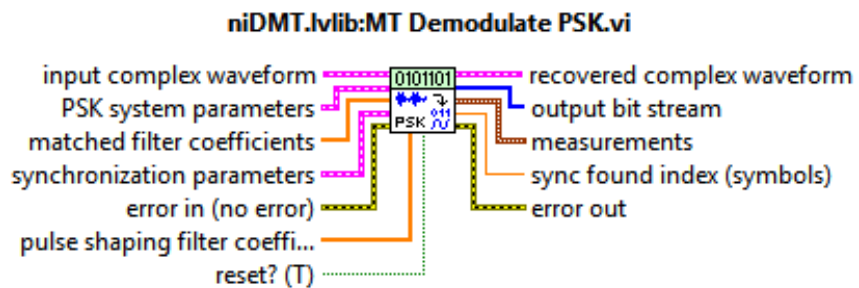
MT GENERATE SYNCHRONIZATION PARAMETERS PSK.VI



MT MODULATE PSK.VI

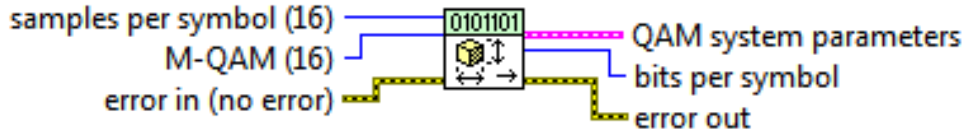


MT DEMODULATE PSK.VI



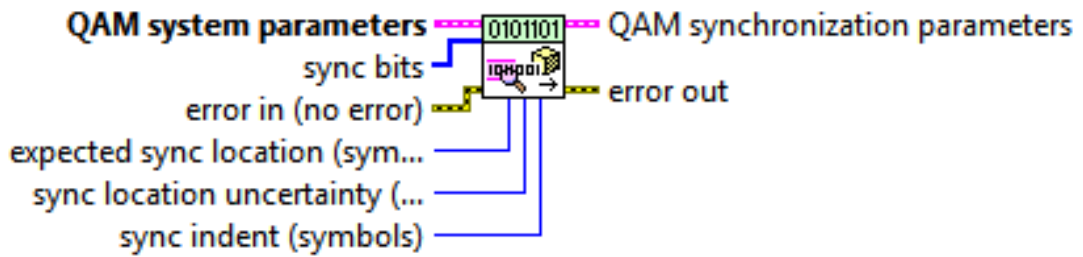
MT GENERATE SYSTEM PARAMETERS QAM.VI

niDMT.lvlib:MT Generate System Parameters.vi



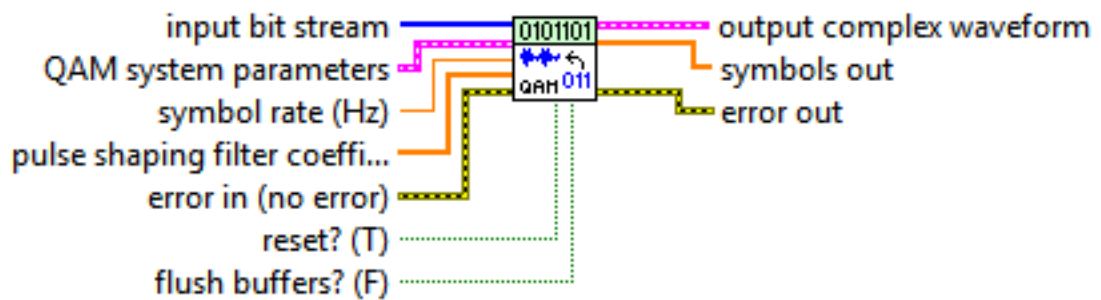
MT GENERATE SYNCHRONIZATION PARAMETERS QAM.VI

niDMT.lvlib:MT Generate Synchronization Parameters.vi

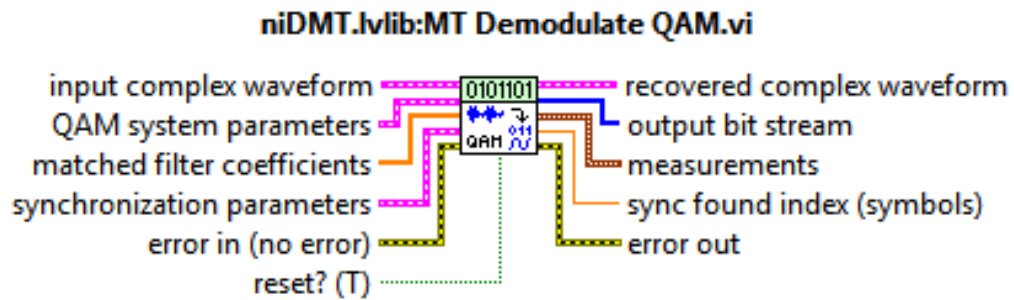


MT MODULATE QAM.VI

niDMT.lvlib:MT Modulate QAM.vi

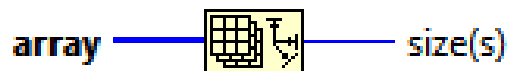


MT DEMODULATE QAM.VI



ARRAY SIZE

Array Size

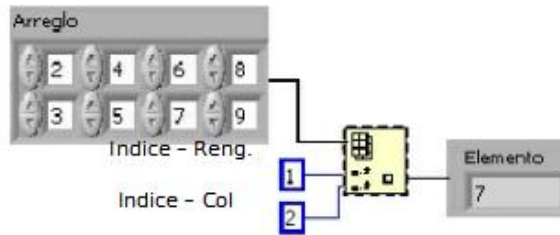


Descripción de los terminales:

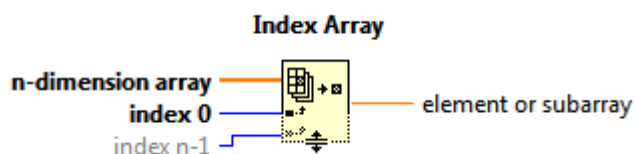
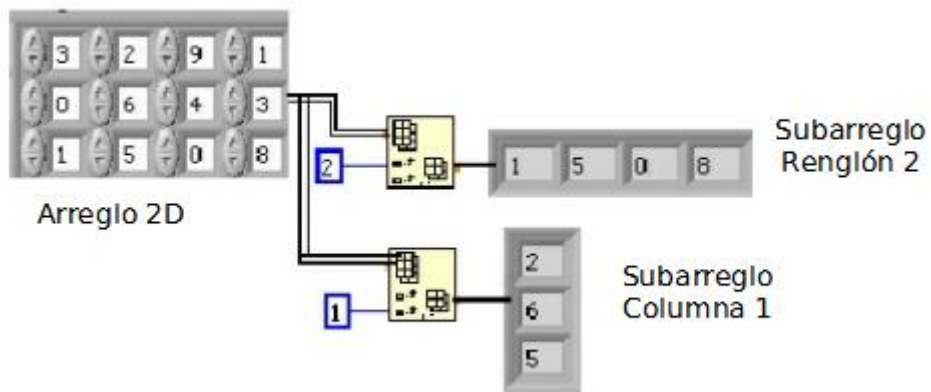
- **array:** puede ser una matriz de n-dimensión de cualquier tipo.
- **size:** es un entero de 32 bits si el array es de una dimensión (1D). Si la matriz es multidimensional, el valor devuelto es una matriz de 1D en la que cada elemento es un entero de 32 bits que representa el número de elementos en la dimensión correspondiente de la matriz.

INDEX ARRAY

Como el ejemplo muestra la función Index Array accedendo el elemento del segundo renglón, tercer columna de una arreglo bidimensional.



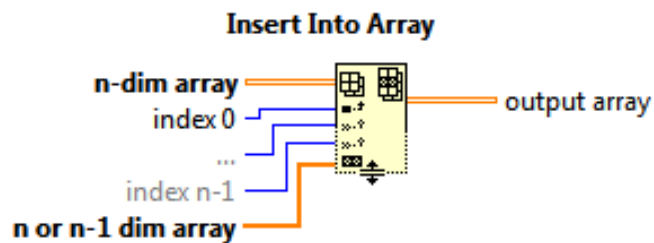
Si bien usted puede usar la función Index Array para extraer un elemento escalar de un arreglo, también se puede usar esta función para extraer un renglón o una columna de un arreglo multidimensional, y formar un subarreglo. Index Array realiza lo anterior automáticamente. Si se desea extraer un renglón de un arreglo multidimensional, se conecta una constante en la terminal Índice - Reng. (mostrada en la figura anterior), para indicar el renglón que se desea extraer. Ahora, la terminal Índice - Col se deja sin conectar. Así mismo, si se desea extraer una columna, se indica el número de ésta en la terminal Índice - Col y se deja sin conectar la terminal Índice-Reng



Descripción de los terminales:

- **n-dimension array:** puede ser una matriz n-dimensional de cualquier tipo. Si n-dimensión de la matriz es un conjunto vacío, elemento o submatriz devuelve el valor predeterminado del tipo de datos definido por la matriz.
- **index 0:** debe ser numérico. El número de entradas de índice coincide con el número de dimensiones de matriz n-dimensión.

INSERT INTO ARRAY

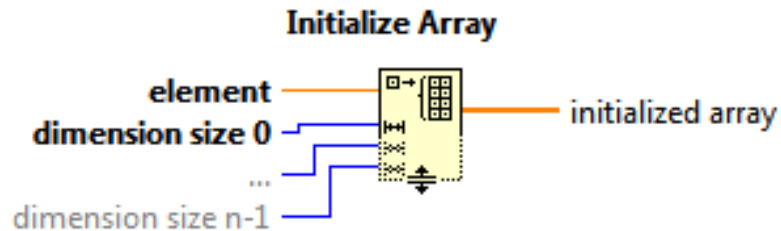
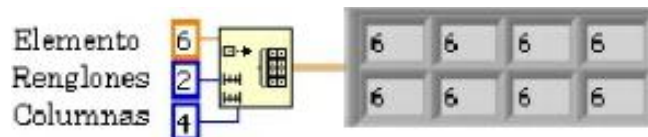


Descripción de los terminales:

- **n-dim array:** es el array en la que desea insertar un elemento, fila, columna, en la página, y así sucesivamente. Esta entrada puede ser un arreglo de n dimensiones de cualquier tipo.
- **index 0...n-1:** especifica el punto en el array en el que desea insertar el elemento, fila, columna, en la página, y así sucesivamente. Puede conectar una sola entrada de índice.
- **output array:** esta función devuelve con el elemento insertado (s), fila (s), columna (s), o página (s).

INITIALIZE ARRAY

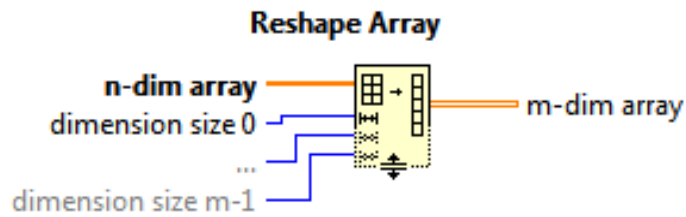
El tamaño del icono de esta función puede modificarse de tal manera que corresponda al número de dimensiones del arreglo de salida. El siguiente ejemplo muestra un arreglo bidimensional de 4 x 2 elementos, inicializado con los valores 2 y 4 respectivamente.



Descripción de los terminales:

- **element:** es el valor que se utiliza para inicializar todos los elementos de la matriz inicializada.
- **dimensión size 0..n-1:** podría ser un numero. La función crea una matriz vacía si *dimension size* es 0. Usted debe tener terminales con un tamaño n-dimensión.
- **initialized array:** es una matriz del mismo tipo al que esta conectado al terminal *element*.

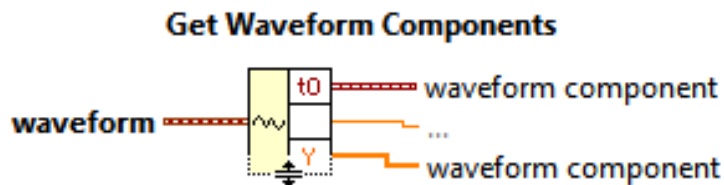
RESHAPE ARRAY



Descripción de los terminales:

- **n-dim array:** puede ser un array n-dimensional de cualquier tipo.
- **dimensión size 0:** especifica las dimensiones de **m-dim array** y podría ser numerico. La función crea un array vacío si este terminal es 0. Puede tener el terminal **m dimension size** *m*-dimensiones.
- **m-dim array:** la función rellena la nueva matriz con el valor predeterminado del tipo de datos de *n-dim array*.

GET WAVEFORM COMPONENTS

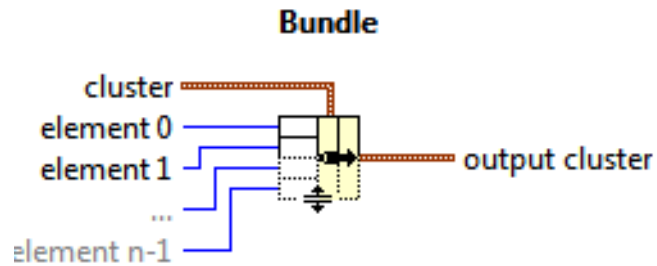


Descripción de terminales:

- **waveform:** es la forma de onda de la cual usted quiere recuperar sus componentes.
- **t0:** devuelve el tiempo de activación de la forma de onda.
- **dt** devuelve el intervalo de tiempo en segundos entre puntos de datos en la forma de onda
- **Y** devuelve los valores de los datos del *waveform*.

- **attributes** devuelve los nombres y valores de todos los atributos de forma de onda. También puede usar el *Get Waveform Attribute VI* para recuperar los nombres y valores de todos los atributos o el valor de un atributo único.

BUNDLE



Descripción de terminales:

- **cluster:** es el grupo cuyo valor desea cambiar. Si no conectar esta entrada, la función crea un clúster.
- **element 0..n-1** acepta cualquier tipo de datos.
- **output clúster:** es el resultante del clúster.

NUMBER TO BOOLEAN ARRAY

Number To Boolean Array



Descripción de terminales:

- **number:** puede ser un número.
- **Boolean array:** puede tener 8, 16, 32, o 64 elementos si el **number** es un entero. Si **number** es un número de punto fijo, el array booleano puede tener entre 1 y 64 elementos.

BOOLEAN TO (0,1)

Boolean To (0,1)

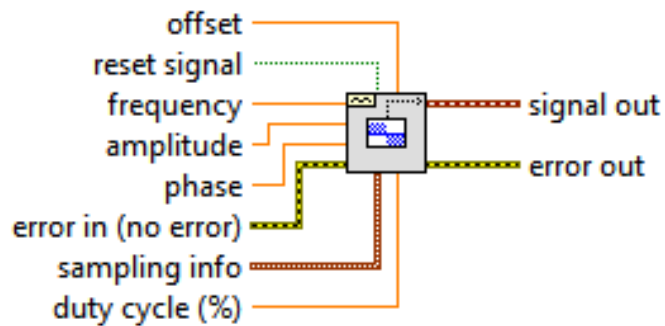


Descripción de terminales:

- **Boolean:** puede ser un escalar, una matriz, un conjunto de valores booleanos, una serie de grupos de valores booleanos, y así sucesivamente.
- **0,1:** es 0 si booleano es FALSO y 1 si booleano es VERDADERO.

SQUARE WAVEFORM

NI_MABase.lvlib:Square Waveform.vi



Descripción de terminales:

- **offset:** es el desplazamiento DC de la señal. El valor predeterminado es 0,0.
- **reset signal:** si es TRUE, se restablece la fase con el valor de control de fase y la marca de tiempo a cero. El valor predeterminado es FALSE.
- **frequency:** es la frecuencia de la forma de onda en unidades de hertz. El valor por defecto es 10.
- **amplitude:** es la amplitud de la forma de onda. La amplitud es también el voltaje pico. El valor por defecto es 1.0.

- **phase:** es la fase inicial, en grados, de la forma de onda. El valor por defecto es 0. El VI ignora la fase si **reset signal** is FALSE.
- **sampling info:** contiene información de muestreo.
 - **Fs:** es la frecuencia de muestreo en muestras por segundo. El valor predeterminado es de 1000.
 - **#s:** es el número de muestras en la forma de onda. El valor por defecto es 1000.
- **duty cycle:** es el porcentaje de tiempo de una onda cuadrada sigue siendo alta en comparación a baja altura sobre un período. El valor predeterminado es 50.
- **signal out:** es la forma de onda generada.

ANEXO 2: Manual de usuario del simulador de modulaciones digitales.

MANUAL DE USUARIO

SIMULADOR DE MODULACIONES DIGITALES

1. Menú Inicio

A continuación los pasos para ejecutar el programa:

1. Corremos el ejecutable del simulador de modulaciones digitales.
2. Elegimos el botón **MENU** para abrir el menú principal del simulador de modulaciones digitales y si desea abortar la aplicación clic en el botón **EXIT**.



2. Menú Principal

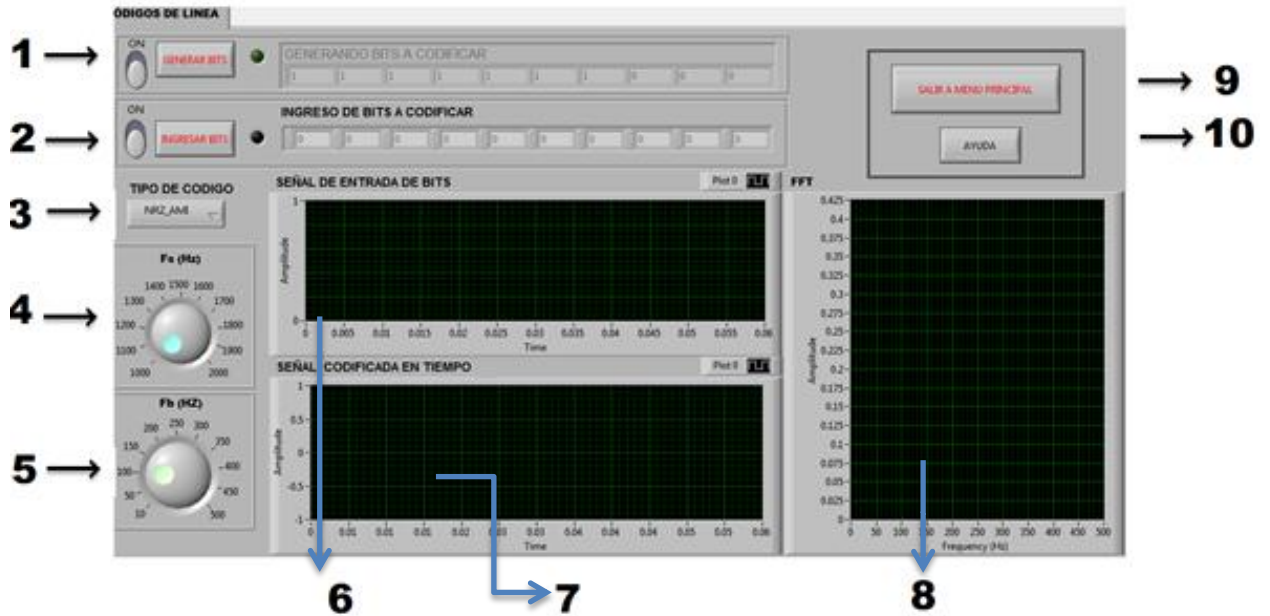
En esta ventana, se tiene 3 opciones que son:

1. **Señales Banda Base:** En esta opción se puede ingresar al instrumento virtual, donde se realiza los diferentes códigos de línea como son: NRZ_U (No retorno a cero unipolar), NRZ_P (No retorno a cero polar), RZ_U (Retorno a cero unipolar), RZ_B (Retorno a cero bipolar), AMI_NRZ (No retorno a cero e inversión alternada de marca), AMI_RZ (Retorno a cero e inversión alternada de marca) y el código Manchester.
2. **Señales Pasa Banda:** En esta opción se abre el instrumento virtual, encargado de simular las modulaciones y demodulaciones digitales. Entre ellas tenemos: ASK (Modulación desplazamiento amplitud), FSK (Modulación desplazamiento en frecuencia), PSK (Modulación desplazamiento en fase) y QAM (Modulación de amplitud en cuadratura), cada una con sus respectivos parámetros de entrada.
3. **Atrás:** Permite regresar a la ventana INICIO.



3. Simulador de señales en Banda Base

Como se puede ver en la siguiente figura, se muestra todos los **controles** e **indicadores**, así como también dos opciones: *salir a menú principal* y *ayuda*. A continuación se explica cada elemento de este instrumento virtual:



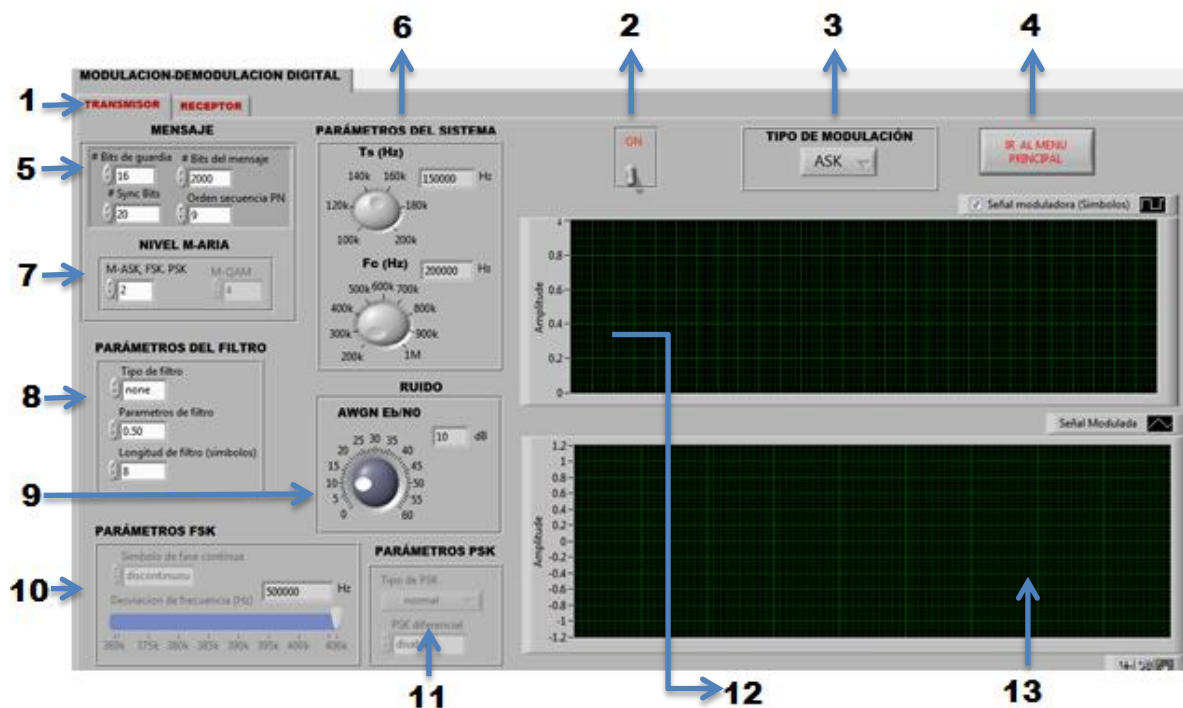
1. **Generar bits:** El control booleano debe estar en **ON**, para activar el ingreso de los bits a codificar de manera aleatoria, cada vez que se presione el botón *generar bits a codificar*. El indicador denominado *generando bits a codificar*, permite visualizar cada uno de los bits generados automáticamente, es decir los 10 bits.
2. **Ingresar bits:** El control booleano debe estar en **ON**, para activar el ingreso de los bits a codificar manualmente, una vez presionado el botón *ingresar bits*. Al tener activado este modo de ingreso de bits a codificar, el usuario podrá ingresar cada uno de los bits (entre 0, 1 lógico) para su posterior codificación de línea.
3. **Tipo de Código:** Es un control que muestra un menú con todos los tipos de código de línea desarrollados en el presente proyecto.
4. **Frecuencia de muestreo:** Control encargado de variar la frecuencia a la que se desea muestrear la señal.
5. **Frecuencia de bits:** Es un control que permite variar, la rapidez de la señal los bits codificados en tiempo con la señal de entrada de bits.

6. **Señal de entrada de bits:** Es un indicador grafico que muestra la señal de entrada de bits formada a partir de los bits ingresados, ya sea aleatoria o manualmente.
7. **Señal codificada en tiempo:** Es un indicador grafico, encargada de visualizar la respectiva señal codificada en tiempo, según el tipo de código de línea seleccionado.
8. **Espectro de frecuencia:** En este indicador se muestra una gráfica de amplitud contra frecuencia de la señal codificada en tiempo.
9. **Salir a menú principal:** Permite regresar nuevamente al Menú Principal en donde se puede escoger la otra opción que seria ingresar al simulador de señales pasa banda o salir.
10. **Ayuda:** Es donde se puede obtener información teórica acerca de las características de cada código de línea y sus formas de onda.

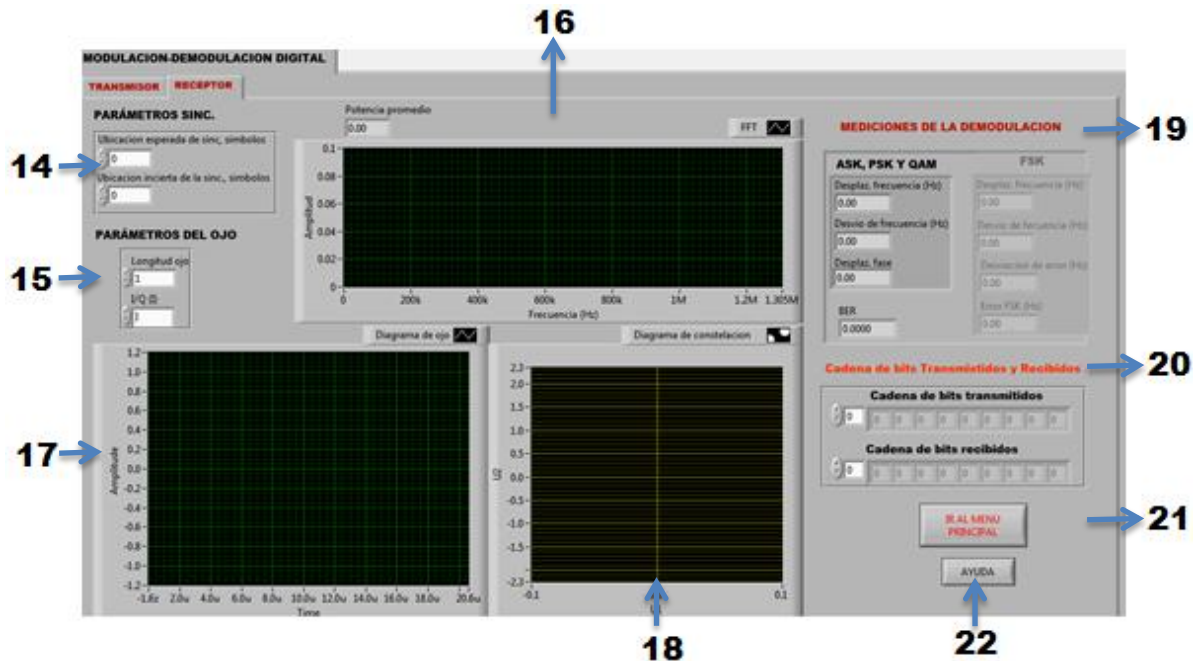
4. Señales Pasa Banda

Continuando con el manual de usuario, se detalla cada uno de los elementos que conforma el simulador de señales en pasa banda.

Transmisor



Receptor



1. **PESTAÑAS DE CONTROL** que permiten navegar por dos ventanas que son: **Transmisor y Receptor**.
2. Botón **ON**, es utilizado para activar la visualización de las graficas correspondientes al VI Principal.
3. **TIPO DE MODULACIÓN** es un control que despliega un menú para elegir el tipo de modulación digital (ASK, FSK, PSK y QAM) que se requiere simular.
4. **IR AL MENU PRINCIPAL** es un botón que permite retornar a la ventana denominado menú principal.
5. **MENSAJE**, son los controles que crean el mensaje o conocida como señal moduladora, estos controles ya se encuentran con los valores por defecto y son:
 - **#Bits de Guardia**: numero de bits de guardia. El valor por defecto es 16.
 - **#Sync Bits**: control para especificar el número de bits que se quieren sincronizar. El valor por defecto es 20.
 - **#Bits mensaje**: ingrese el número de bits del mensaje. El valor por defecto es 2000.
 - **Orden secuencia PN**: genera una secuencia de bits ordenados según el valor especificado. El valor por defecto es 9.

6. **PARÁMETROS DEL SISTEMA**, se refiere a dos controles muy importantes al momento de realizar una cierta modulación digital y estos son:

- **Tasa de símbolos (Ts):** Permite variar la velocidad del símbolo que desea, en Hertz (Hz). Este valor debe ser positivo. El valor predeterminado es 1,00.
- **Frecuencia portadora (Fc):** Se lo utiliza para manipular la frecuencia de la señal portadora. Especifica la frecuencia central pasa banda, en hertz (Hz). La señal entrante se convierte en el centro alrededor de esta frecuencia.

7. Controles que tienen como función principal, el ingreso del **NIVEL M-ARIO** de la señal moduladora, cabe recordar que M-ario, representa la cantidad de condiciones posibles (**símbolos**) para una determinada cantidad de variables binarias consideradas. La ecuación que se aplica para relacionar el número de bits con el numero de símbolos es la siguiente:

$$M = 2^N$$

Donde:

N: Cantidad de bits codificados

M: Cantidad de condiciones posibles de salida con N bits

Ejemplo:

Si se toman dos bits codificados juntos entonces $M = 2^2 = 4$. En general:

N	M
1	2
2	4
3	8
4	16

8. **Parámetros de filtro**, tiene que ver con los controles del filtrado:

- **Tipo de filtro:** control para seleccionar la clase de filtro a generar.
- **Parámetro de filtro:** especifica ya sea alfa (atenuación de coseno elevado y el filtro raíz cuadrada de coseno elevado), o BT (el producto del ancho de banda -3 dB y el período de símbolo para un filtro Gaussiano). El valor predeterminado es 0.5

- **Longitud de filtro:** especifica la longitud deseada del tipo de filtro, en símbolos. Los valores válidos son > 1 . El valor predeterminado es 8.
 - **Muestras por símbolo:** especifica el número de muestras deseadas por símbolo para la demodulación filtro adaptado. El valor predeterminado es 16.
9. Control del Ruido Blanco Gaussiano Aditivo (**AWGN**), es utilizado para simular como el ruido afecta a la señal transmitida.
10. **PARÁMETROS FSK**, son los controles que se consideran como parámetros adicionales de la modulación FSK como son: **símbolo de fase continua** que permite escoger dos opciones continuo o discontinuo y el otro control es la **desviación de frecuencia** medido en Hertz.
11. **PARÁMETROS PSK**: Son los parámetros adicionales para la modulación PSK y entre ellos tenemos:
- **Tipo de PSK:** En este conjunto de controles se muestra un menú que permite seleccionar el tipo de modulación: normal, shifted y offsed.
 - **Normal:** establece el tipo de modulación para PSK regular.
 - **Shifted:** gira la constelación por π / M cada símbolo.
 - **Offset:** Este esquema de modulación es una forma de manipulación por desplazamiento de fase en la que cuatro ángulos diferentes de fase se utilizan.
 - **PSK Diferencial:** Permite activar o desactivar un PSK diferencial. La operación diferencial se utiliza para implementar formatos PSK tales como diferencial de cuadratura PSK (DQPSK) y $\pi / 4$ DQPSK.
 - **disable (0):** en esta opción los símbolos se representan como puntos de la constelación. Este valor es predeterminado.
 - **enable (1):** activado esta opción los símbolos se representan como las transiciones entre puntos de la constelación
12. **SEÑAL MODULADORA**, es una ventana en donde se visualiza una señal digital de acuerdo a los parámetros ingresados en los controles del **MENSAJE**.

13. SEÑAL MODULADA, es una ventana en donde se muestra la señal modulada, según el tipo de modulación seleccionada.

14. PARÁMETROS DE SINCRONIZACIÓN: Son dos controles encargados de controlar que los bits de transmisión sean los mismos que los bits de recepción:

- **Ubicación esperada de Sincronización, símbolos:** control que especifica la ubicación prevista de los parámetros de sincronización de símbolos. El valor predeterminado es -1.
- **Ubicación incierta de sincronización, símbolos:** control que especifica el número de símbolos antes o después de la ubicación de la sincronización, que podría ser tal vez, en el primer símbolo de la secuencia de sincronización situado. El valor por defecto es 10

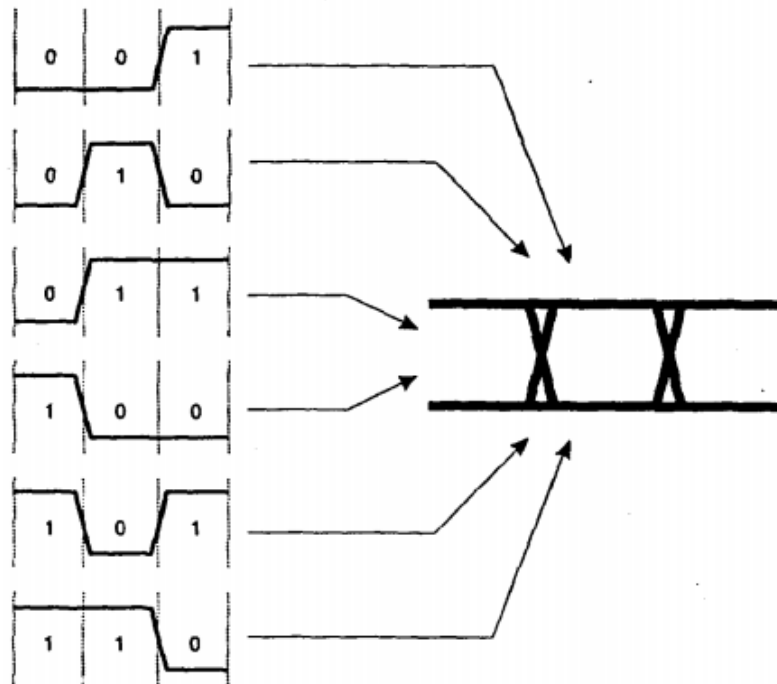
15. PARÁMETROS DE OJO: Muestra los controles de los parámetros del diagrama de ojo. Estos controles son:

- **Longitud de ojo:** control que especifica el número de periodos del símbolo en la escala horizontal del diagrama de ojo.
- **I/Q:** control que especifica si debe generarse I (fase) de datos o Q (cuadratura) para el diagrama de ojo.

16. ESPECTRO DE FRECUENCIA: Es un indicador gráfico que visualiza la señal modulada en función de la amplitud con respecto a la frecuencia. También en la parte superior izquierda de este indicador se muestra la potencia promedio de la señal modulada.

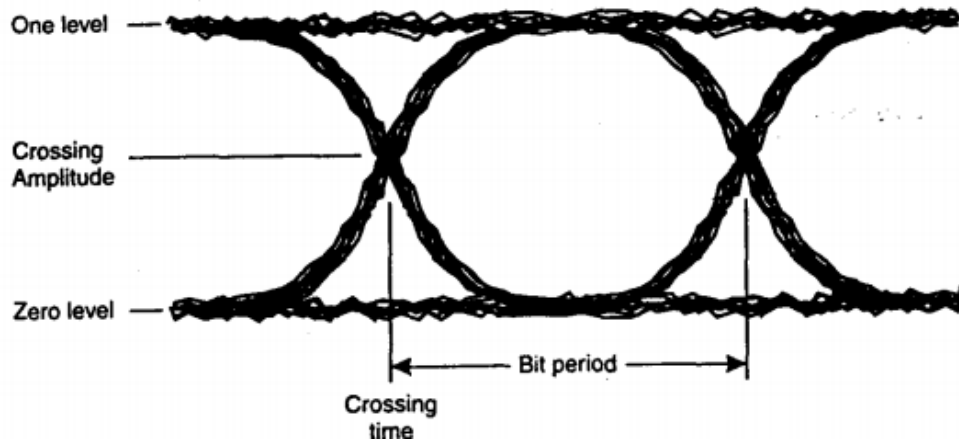
17. DIAGRAMA DE OJO: es una ventana utilizada para el análisis del comportamiento de los enlaces de transmisión. Permite analizar las formas de onda de los pulsos que se propagan en un enlace de comunicaciones, para lograr observar sus formas, desfases, niveles de ruido, potencias de las señales y con ello apreciar la distorsión del canal, la severidad del ruido o interferencia y los errores de sincronismo en el receptor.

Por ejemplo en una secuencia de 3 bits tenemos una cantidad total de 8 combinaciones posibles, las que pueden ser observadas en la siguiente figura. Se puede observar que en la figura no se consideran las cadenas de 3 unos y 3 ceros consecutivos, ya que, debido a la superposición de las otras combinaciones, quedan determinadas implícitamente.



Las propiedades del diagrama de ojo son:

- **One level:** Corresponde a la medición del valor promedio del nivel de uno lógico.
- **Zero level:** Corresponde a la medida del valor medio del nivel cero lógico.
- **Eye Crossing:** Consiste de dos partes, crossing time y crossing amplitud. El crossing time se refiere al tiempo en el que se produce la apertura del ojo y su posterior cierre, mientras que el crossing amplitud, esta referido al nivel de voltaje en la cual se produce la apertura del ojo y su posterior cierre. Considerando estos dos parámetros se define el **Bit Period**, que corresponde al periodo entre la apertura y cierre del ojo.



18. DIAGRAMA DE CONSTELACIÓN: Es un diagrama de constelación que representa un esquema de modulación digital en el plano complejo. Los ejes real e imaginario suelen ser llamados I (por In-phase) y Q (por quadrature). Los puntos en la constelación representan símbolos de modulación los que componen el alfabeto, es decir todas las "palabras" que podrán usarse en un intercambio de información.

19. MEDICIONES DE LA DEMODULACIÓN: En esta parte, se obtiene los valores o resultados de la demodulación digital y que explicamos a continuación:

- **Desplazamiento frecuencia:** devuelve la frecuencia de la portadora de la medición de offset, en hertz (Hz). El desplazamiento de frecuencia medida, se elimina de la forma de onda compleja recuperada.
- **Desviación de frecuencia:** devuelve la medición de la desviación de la frecuencia portadora, en hertz (Hz).
- **Desplazamiento fase:** retorna la fase medido en grados. La medición desplazamiento de fase, se elimina de la forma de onda compleja recuperada.
- **BER:** es un indicador que muestra la cantidad de errores en toda línea de transmisión de datos, y se define como la probabilidad de que un bit se reciba erróneamente.

Para los resultados de la demodulación FSK, se adiciona dos indicadores más de las expuestas anteriormente:

- **Desviación frecuencia:** especifica la máxima desviación de la frecuencia FSK. El valor por defecto es 15.
- **FSK error:** muestra la medición del Error de FSK. Su formula es: $FSK\ error(\%) = [FSK\ error(Hz)/maximum\ ideal\ frequency\ deviation] \times 100$

20. Cadena de bits de Transmisión y recepción: son indicadores numéricos, que nos ayudan a verificar si la información (bits) fue recuperada o no, dependiendo de todos los parámetros del transmisor como del receptor.

21. Menú Principal: Es un botón creado con el propósito de retornar a la ventana del menú principal.

22. Ayuda: Este botón permite abrir una ventana, en donde se visualiza la información que ayuda a entender el comportamiento de las modulaciones digitales.

ANEXO 3: Encuesta realizada a los alumnos 7mo, 8vo y 9no semestre de la carrera de electrónica y comunicaciones en FISEI.

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

Encuesta dirigida para los estudiantes de la carrera de Ingeniería en Electrónica y Comunicaciones

OBJETIVO: Recolectar información para la implementación de un laboratorio virtual de modulación y demodulación digital, que se tiene para las prácticas de laboratorio e investigación

INSTRUCTIVO: Marque con una X su respuesta adecuada a sus conocimientos en Electrónica y Comunicaciones.

1. Considera usted importantes los sistemas de comunicaciones digitales?

- Si
- No

2. Sabe como es el principio de funcionamiento de una comunicación digital?

- Si
- No

3. Como le gustaría que fueran sus clases de comunicaciones digitales?

- Practicas
- Teóricas
- Ambos términos

Porque:

4. Conoce usted instrumentos para medir y visualizar los diferentes métodos de modulaciones digitales?

- Si
- No

Cuales: _____

5. Usted, tiene conocimientos de lo que es un laboratorio o instrumento virtual?

- Si
- No

6. Está de acuerdo, que las aplicaciones de un laboratorio virtual permiten mejorar la enseñanza-aprendizaje en el área de comunicaciones digitales?
- **Si**
 - **No**
7. Considera importante que la FISEI posea instrumentos con los que se pueda realizar prácticas de Modulaciones y Demodulaciones digitales?
- **Si**
 - **No**

ANEXO 4: Encuesta realizada a los docentes del área de electrónica y comunicaciones en FISEI.

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

Encuesta dirigida para los docentes del área de comunicaciones de la carrera en Ingeniería en Electrónica y Comunicaciones.

INSTRUCTIVO: Procure ser lo más objetivo en honor a la verdad.

Marque con una X en la respuesta adecuada, que usted crea conveniente.

- Considera usted importante realizar prácticas de modulaciones y demodulaciones digitales en los laboratorios de electrónica?
 - Si
 - No

- El Software, utilizado en los laboratorios de electrónica de la FISEI, cuenta con sus respectivas licencias y son óptimas para simular diferentes aplicaciones de comunicaciones digitales.
 - Si
 - No
 - Ninguno

- Cuenta los laboratorios de electrónica con equipos, capaces de generar, medir o visualizar las modulaciones y demodulaciones digitales para realizar practicas de calidad?
 - Si
 - No

- Cree usted, que una manera de solucionar la carencia de practicas por falta de equipos en los temas de modulaciones digitales, es haciendo uso de laboratorios virtuales?
 - Si
 - No
 - Ninguno

Porque: _____

- Cree usted, que variar de manera practica los parámetros de una señal digital y verificando los resultados, son de gran interés para entender las modulaciones digitales?
 - **Si**
 - **No**

- Considera importante que la FISEI posea un laboratorio virtual, en la que se pueda realizar las Modulaciones y demodulaciones digitales, con el fin de mejorar la enseñanza-aprendizaje?
 - **Si**
 - **No**

ANEXO 5: Ficha de observación realizada al laboratorista del área de electrónica y comunicaciones en FISEI.

UNIVERSIDAD TECNICA DE AMBATO
FISEI

Ficha de observación dirigida a los laboratoristas de la Carrera en Electrónica y Comunicaciones.

Objetivo: Recolectar información sobre los equipos y los instrumentos que se tiene para las prácticas de laboratorio e investigación.

PARAMETROS	SI	NO
Se cuenta con las suficientes herramientas de software y hardware para las prácticas de modulaciones digitales, en los laboratorios de la FISEI.		
Los programas que posee la FISEI, son actualizados para el desarrollo de la investigación.		
Existen equipos con varias interfaces, que permitan interactuar con programas como Labview, etc.		
Es necesario tener laboratorios para diferentes aplicaciones en el área de comunicaciones digitales.		
Los instrumentos de medición en los laboratorios satisfacen las necesidades de los docentes.		
Los equipos de los laboratorios de electrónica, en la FISEI, cumplen con los requerimientos de los estudiantes, a la hora de realizar prácticas de comunicaciones digitales.		
En el laboratorio, las prácticas de modulación y demodulación digital son difíciles de ejecutar por la falta de equipos.		
Sería una gran contribución si se implementara herramientas virtuales, para el análisis de las modulaciones digitales en los laboratorios de electrónica de la FISEI.		
TOTAL		

