

The background of the entire image is a photograph of a large satellite dish antenna against a dark, star-filled night sky. The dish is angled upwards, and the background shows a gradient from deep blue at the top to orange and red at the horizon.

Telecomunicaciones en tu idioma

Notas de un estudiante

Jose Antonio Hancco M.

Copyright © 2022 Jose Hancco

LIBRO LIBRE DE USOS

[HTTPS://GITHUB.COM/YASPERTERIAN](https://github.com/YASPERTERIAN)

Con licencia de Creative Commons Attribution-NonCommercial 3.0 Unported License (la “Licencia”). No puede usar este archivo excepto de conformidad con la Licencia. Puede obtener una copia de la Licencia en <http://creativecommons.org/licenses/by-nc/3.0>. A menos que lo exija la ley aplicable o se acuerde por escrito, el software distribuido bajo la Licencia se distribuye “TAL CUAL”, SIN GARANTÍAS NI CONDICIONES DE NINGÚN TIPO, ya sea expresa o implícita. Consulte la Licencia para conocer el idioma específico que rige los permisos y las limitaciones en virtud de la Licencia.

Primera edición, septiembre 2021

Si existe algún error, crees que una sección se puede mejorar o dar cualquier tipo de *feedback* acerca del libro no dudes y mándame un correo a jhanccoma@unsa.edu.pe, te responderé lo más pronto que pueda y gracias por mejorar este libro de todos y para todos.

Ya sabes cómo funciona esto. Coges un libro, saltas a la dedicatoria y descubres que, una vez más, el autor ha dedicado su libro a alguien que no eres tú.

Pero esta vez no será así...

Dedico este libro a mi yo del pasado, quien ha sacrificado tiempo para poder escribir este libro para quienes lo necesiten pues como cualquier carrera, no es difícil los temas que tienes que aprender, sino que te lo explican mal y sin ganas, ganas desganadas que destruyen las tuyas.



Índice general

I

Métodos Matemáticos I

1	Ecuaciones Diferenciales de primer orden	13
1.1	Conceptos introductorios	13
1.1.1	Tipos	13
1.1.2	Orden y grado	14
1.2	Variables separables	15
1.3	Ecuaciones diferenciales lineales	18
1.4	Ecuaciones Diferenciales Exactas	20
1.5	Ecuaciones Diferenciales No Exactas	22
1.6	Ecuaciones por sustitución	24
1.6.1	Ecuaciones Homogéneas	24
1.6.2	Ecuaciones Diferenciales de Bernoulli	26

II

Sistemas Digitales

2	Lógica digital	31
2.1	Compuertas lógicas	31
2.1.1	Compuerta NOT	31
2.1.2	Compuerta AND	32
2.1.3	Compuerta OR	33
2.1.4	Compuerta NAND	34
2.1.5	Compuerta NOR	35
2.1.6	Compuerta OR exclusiva-XOR	36

2.1.7	Compuerta NOR exclusiva-XNOR	37
2.2	Formas de expresar funciones booleanas	37
2.2.1	Suma de productos-SOP	37
2.2.2	Productos de sumas-POS	37
2.2.3	Conversión de SOP a POS	39
2.2.4	Forma canónica	40
2.2.5	Forma normalizada	40
2.2.6	¿Cómo pasar de la forma normalizada a la canónica?	40
2.3	Axiomas	41
2.4	Teoremas del álgebra de conmutación	41
2.4.1	Dualidad	42
2.5	Mapas de Karnaugh	44
2.5.1	Mapas de Karnaugh-4 variables	44
2.5.2	Agrupación de términos	46
2.5.3	Mapas de Karnaugh-5 variables	50
2.5.4	Mapas de Karnaugh-6 variables	51
2.6	Parámetros TTL y CMOS	51
2.6.1	Tensión de alimentación continua	51
2.6.2	Nivel lógico CMOS +5V	52
2.6.3	Nivel lógico CMOS +3V	52
2.6.4	Nivel lógico TTL	53
2.6.5	Inmunidad y Margen de ruido	53
2.6.6	Disipación de potencia	54
2.6.7	Carga y Fan-out	55
3	Circuitos Lógicos	57
3.1	Sumador	57
3.1.1	Semi-sumador	57
3.1.2	Sumador completo	58
3.1.3	Sumador paralelo	59
3.1.4	Sumador 74LS283	59
3.1.5	Sumador Acarreo Anticipado	60
3.2	Codificador y Decodificador	60
3.2.1	Encoder	60
3.2.2	Decoder	62
3.3	Multiplexor y Demultiplexor	64
3.3.1	Multiplexor	64
3.3.2	Demultiplexor	65
3.4	Latch	66
3.4.1	Latch S-R(Set-Reset)	67
3.4.2	Símbolo lógico	68
3.4.3	Latch: Eliminador de rebote	68
3.4.4	Latch con entrada de habilitación	70
3.4.5	Latch D con entrada de habilitación	70

3.5	Flip-Flops	71
3.5.1	Flip Flop tipo S-R disparado por flanco	71
3.5.2	Flip Flop D disparado por flanco	73
3.5.3	Flip Flop J-K disparado por flanco	74
3.5.4	Entradas asíncronas de inicialización y borrado	75
3.6	División de frecuencia	76
3.7	Monoestables	78
4	Circuitos Lógicos Secuenciales	81
4.1	Máquina de estados	81
4.1.1	Modelo de Mealy	82
4.2	Grafos de estados	84

III	Smart cities	
5	Conociendo la ESP8266	95
5.1	ESP8266	95
5.2	Instalación Windows:Materiales	96
5.3	Instalación	96
5.4	Equivalente al Hola Mundo pero en placas	96
5.5	Conectar a una red Wi-Fi	97
5.6	Crear una red WiFi	99
6	Servicios en la red	101
6.1	Crear un servidor local	101
6.2	Lectura Analógica	105
6.3	Proyecto explicativo 1	106
6.3.1	Parte 1	107
6.3.2	Parte 2	108
6.3.3	Parte 3	110
6.4	Sensor de temperatura y humedad DHT11	111
7	Proyecto final	115
7.1	Software	115
7.1.1	Instalación de Node.js	115
7.1.2	Instalación de Mosquitto	116
7.2	Hardware	120
7.3	Back-end	121
7.3.1	Xammp-PHP My admin	122
7.3.2	Xammp-Htdocs	124
7.3.3	Node-RED	124
7.4	Front-end	125

8	Unidad I	137
8.1	Theorems	137
8.1.1	Several equations	137
8.1.2	Single Line	137
8.2	Definitions	137
8.3	Notations	138
8.4	Remarks	138
8.5	Corollaries	138
8.6	Propositions	138
8.6.1	Several equations	138
8.6.2	Single Line	138
8.7	Examples	138
8.7.1	Equation and Text	139
8.7.2	Paragraph of Text	139
8.8	Exercises	139
8.9	Problems	139
8.10	Vocabulary	139
8.11	Table	139
8.12	Figure	139
9	Unidad 2	141
9.1	Modulación por ancho de pulso-PWM	141

10	Introducción y memoria	145
10.0.1	Repaso análisis vectorial	145

11	Telefonía	149
11.1	Evolución de la telefonía	149
11.2	Comutadores	150
11.3	Repaso: Señales analógicas y digitales	150
11.3.1	Ventajas y desventajas de las señales analógicas	150
11.3.2	Ventajas y desventajas de las señales digitales	151
11.4	Multiplexación	151
11.4.1	¿Para que sirve?	151
11.4.2	Multiplexación por división de tiempo o TDM (<i>time-division multiplexing</i>) ..	152
11.4.3	Multiplexación por división de frecuencia o FDM (<i>frequency-division multiplexing</i>)	
	152	

11.4.4	Multiplexación por división de longitud de onda o WDM (<i>wavelength-division multiplexing</i>)	153
11.4.5	Multiplexación por Estadística por División de Tiempo o STDM(<i>Statistical time-division multiplexing</i>)	153
11.5	Técnicas de conmutación	154
11.5.1	Comutación por circuitos	154
11.5.2	Comutador por paquetes	155
11.6	Red de telefonía publica	156
11.6.1	Red de telefonía publica conmutada-PSTN	156
11.6.2	Características	157
11.6.3	Componentes	157
11.7	Señalización telefónica	161
11.8	Planta interna y externa	162
11.8.1	Planta Interna	162
11.8.2	Planta externa	163
12	Herramientas para el análisis telefónico	167
12.1	Tráfico Telefónico	167
12.1.1	Variaciones	167
12.2	Medidas de tráfico telefónico	169
12.3	Tipos de tráfico	172
12.4	Accesibilidad	173
12.5	Grade of service(GoS)	175
12.6	Diseño de redes	175
12.7	Encaminamiento de tráfico	179
12.8	Erlang-B	181
12.8.1	Erlang-B para un enlace	183
12.8.2	Erlang-B para N enlaces	184
13	Telefonía IP	187
13.0.1	Redes de computadoras	187
13.0.2	PDU	187
13.0.3	Arquitectura de red	188
13.0.4	Tipos de servicios	191
13.1	Repaso: TCP/IP	193
13.1.1	Capa de transporte	193
13.1.2	Capa de internet	194
13.2	VoIP	197
13.3	Digitización de la voz	197
13.3.1	¿Como se digitaliza de la voz?	197
13.3.2	Transporte de Voz	199
13.4	Componentes de VoIP	200
13.4.1	Terminal IP	200
13.4.2	Centralita IP	201
13.4.3	Gateway	201
13.4.4	Adaptador FXS	202

13.4.5	Unidad de control MCU	202
13.5	Tecnología IP	202
13.5.1	DSL	202
13.5.2	RDSI	203
13.5.3	LMDS	203
13.5.4	GSM	203
13.6	Vocabulary	205

VII

Instrumentación 1

14	Unidad I	209
-----------	-----------------------	------------

VIII

Anexos

14.1	Ejercicios resueltos U2-Sistemas de telefonía	213
14.2	Ejercicios resueltos U2-Sistemas de telefonía	214
14.3	Ejercicios resueltos U2-Sistemas de telefonía	215
14.4	Ejercicios resueltos U2-Sistemas de telefonía 2	216
14.5	Ejercicios resueltos U2-Sistemas de telefonía 2	217
14.6	Ejercicios resueltos U2-Sistemas de telefonía 2	218
14.7	Ejercicios resueltos U2-Encaminamiento de tráfico telefónico	219
14.8	Ejercicios resueltos U2-Encaminamiento de tráfico telefónico	220
14.9	Ejercicios resueltos U2-Encaminamiento de tráfico telefónico	221
14.10	Ejercicios resueltos U2-Erlang B	222
14.11	Ejercicios resueltos U2-Erlang B	223
14.12	Ejercicios resueltos U2-Erlang B	224

IX

Extras

15	Artificios matemáticos	227
	Bibliography	229
	Articles	229
	Books	229
	Index	231



Métodos Matemáticos I

1 Ecuaciones Diferenciales de primer orden

13

- 1.1 Conceptos introductorios
- 1.2 Variables separables
- 1.3 Ecuaciones diferenciales lineales
- 1.4 Ecuaciones Diferenciales Exactas
- 1.5 Ecuaciones Diferenciales No Exactas
- 1.6 Ecuaciones por sustitución

1. Ecuaciones Diferenciales de primer orden

En este capítulo aprenderemos como resolver ecuaciones diferenciales. Para ello es necesario saber como siempre álgebra media-avanzada, cálculo diferencial e integral y sobre todo habilidad matemática.

1.1 Conceptos introductorios

Una ecuación diferencial es una ecuación derivada de una función desconocida(variable dependiente) respecto a una o más variables independientes.

$$\frac{dy}{dx} + 5 = e$$

Donde:

y es nuestra variable **dependiente**.

x es nuestra variable **independiente**.

1.1.1 Tipos

Existen dos tipos de ecuaciones diferenciales.

Ecuación diferencial Ordinaria

Son de la forma:

$$dy = 2dx$$

$$\frac{dy}{dx} = 2x$$

$$\frac{dy}{dx} = 2xy + 3$$

Donde nuestra prioridad es encontrar **y=F(x)**.

Ecuación diferencial Parcial

Estas ecuaciones son de la forma:

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = xy$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Donde nuestra tarea es hallar $\mathbf{u}=\mathbf{F(x,y)}$

1.1.2 Orden y grado

Debemos dejar en claro algunos términos claro antes de proceder: Orden y Grado.

Orden

El orden de una ED¹ es la derivada de mayor orden que aparece.

Grado

El grado de una ecuación diferencial (ya sea ordinaria o parcial) es el exponente de la mayor derivada contenida en la ecuación.

$$\underbrace{\left(\frac{d^3y}{dx^3}\right)^2}_{orden3} - 2\left(\frac{dy}{dx}\right)^4 + xy = 0$$

Definición 1.1.1 — Homogeneidad en funciones. Una función en homogénea si todos los argumentos se multiplican por un factor constante, entonces el valor de la función resulta ser un cierto número de veces el factor multiplicativo elevado a una potencia. Dicha potencia es el grado de la función homogénea.

$$G(tx,ty) = t^n \cdot G(x,y)^a \quad (1.1)$$

Una función es homogénea si al multiplicarla por t nos devuelve la misma función multiplicada t a la n veces, donde n es el grado de homogeneidad.

^aSe usará letras mayúsculas para representar funciones en su forma natural y letras minúsculas como las derivadas de las mismas.

Ejercicio 1.1 Comprobar si la función $F(x)$ es homogénea y si lo es, indicar su grado.

$$F(x,y) = \sqrt{x^2 + y^2}$$

Solución:

Multiplicamos por un mismo factor t .

$$F(tx,ty) = \sqrt{t^2x^2 + t^2y^2}$$

¹Desde ahora ecuación diferencial se abreviará como ED

Factorizamos t de la expresión:

$$F(x,y) = t \underbrace{\sqrt{x^2 + y^2}}_{G(x,y)}$$

$$F(x,y) = t \cdot G(x,y)$$

La función es homogénea de grado 1

■

1.2 Variables separables

Son las más sencillas de identificar, para poder resolverlas es necesario hacer un despeje para "acomodar" los términos e integrar a ambos lados de la expresión.

Definición 1.2.1 — ED de variables separables. La forma estándar de estas ecuaciones es:

$$\frac{dy}{dx} = F(x,y) = \frac{g(x)}{h(y)}$$

$$\frac{dy}{dx} = \frac{g(x)}{h(y)}$$

$$h(y)dy = g(x)dx \rightarrow \int h(y)dy = \int g(x)dx + C$$

Después de la integración llegamos a:

$$H(y) = G(X) + C$$

También pueden verse de la forma diferencial:

$$G(x,y)dx + H(x,y)dy = 0 \quad (1.2)$$

Ambas soluciones se presentan normalmente de la manera implícita.

Ejercicio 1.2

$$\begin{aligned} y \cdot \ln(x) \frac{dx}{dy} &= \left(\frac{y+1}{x} \right)^2 \\ \underbrace{\int x^2 \cdot \ln(x) dx}_1 &= \underbrace{\int \frac{(y+1)^2}{y} dy}_2 \end{aligned} \quad (1.3)$$

Resolviendo por integrales por partes para 1:

$$\begin{aligned}
 u &= \ln(x) & du &= \frac{1}{x} dx \\
 dv &= x^2 & v &= \frac{x^3}{3} \\
 \ln(x) \cdot \frac{x^3}{3} - \int \frac{x^3}{3} \cdot \frac{1}{x} dx & & & \\
 \frac{x^3 \cdot \ln(x)}{3} - \frac{x^3}{9} & & & \quad (1.4)
 \end{aligned}$$

Resolviendo normalmente para 2:

$$\begin{aligned}
 \int \left(\frac{y^2}{y} + \frac{2y}{y} + \frac{1}{y} \right) dy & \\
 \frac{y^2}{2} + 2y + \ln(y) & \quad (1.5)
 \end{aligned}$$

Juntando los resultados de 1 y 2:

$$\boxed{\frac{x^3 \ln(x)}{3} - \frac{x^3}{9} = \frac{y^2}{2} + 2y + \ln(y) + C} \quad (1.6)$$

■

Ejercicio 1.3 Ejercicio ED de primer orden con Condiciones Iniciales^a.

$$\begin{aligned}
 \frac{dy}{dt} + ty &= y; \quad y(1) = 3 \\
 \frac{dy}{dt} &= y - ty \\
 \frac{dy}{dt} &= y(1-t) \\
 \int \frac{1}{y} dy &= \int (1-t) dt \\
 \ln(y) &= t - \frac{t^2}{2} + C
 \end{aligned} \quad (1.7)$$

De por sí, la ecuación 1.7 es la respuesta, pero por algo nos dan las C.I. Recordemos que y es una función que depende de t , en otras palabras: $Y(t)$. Con la C.I. que nos dieron, nos quiere decir que cuando $t=1$, y vale 3. Bajo este pensamiento podemos seguir con el procedimiento (usando las ecuaciones 15.1 y 15.2) para hallar C para la ecuación 1.7.

$$\begin{aligned}
 y &= e^{t - \frac{t^2}{2} + C}; y(1) = 3 \\
 3 &= e^{1 - \frac{1^2}{2} + c} \\
 \ln(3) &= \frac{1}{2} + C \\
 \ln(3) - \frac{1}{2} &= C
 \end{aligned} \tag{1.8}$$

Una vez despejada la ecuación de 1.8 podemos reemplazarla en la ecuación 1.7:

$$\boxed{\ln(y) = t - \frac{t^2}{2} + \ln(3) - \frac{1}{2}} \tag{1.9}$$

^aDesde ahora se abreviará como C.I.

Ejercicio 1.4 Resolver la siguiente E.D.:

$$\frac{dy}{dx} = \frac{xy + 3x - y - 3}{xy - 2x + 4y - 8} \tag{1.10}$$

Factorizando y separando la ecuación 1.10:

$$\begin{aligned}
 \frac{dy}{dx} &= \frac{x(y+3) - (y+3)}{x(y-2) + 4(y-2)} \\
 \frac{dy}{dx} &= \frac{(y+3)(x-1)}{(y-2)(x+4)} \\
 \frac{y-2}{y+3} dy &= \frac{x-1}{x+4} dx
 \end{aligned} \tag{1.11}$$

Desarrollando las divisiones de ambos lados:

$$\begin{aligned}
 \int 1 - \frac{5}{y+3} dy &= \int 1 - \frac{5}{x+4} dx \\
 y - 5 \ln|y+3| &= x - 5 \ln|x+4| + C
 \end{aligned} \tag{1.12}$$

Usando propiedad de los logaritmos(15.3):

$$\ln|x+4|^5 - \ln|y+3|^5 = x - y + C \tag{1.13}$$

Usando la propiedad de logaritmos(15.4):

$$\begin{aligned}
 \ln\left(\frac{x+4}{y+3}\right)^5 &= x - y + C \\
 \left(\frac{x+4}{y+3}\right)^5 &= e^{x-y} \cdot e^C
 \end{aligned} \tag{1.14}$$

En este caso, una exponencial de c es posible solo expresarla como una exponencial con exponente 1:

$$\left(\frac{x+4}{y+3}\right)^5 = e^{x-y} \cdot e \quad (1.15)$$

■

1.3 Ecuaciones diferenciales lineales

Definición 1.3.1 — E.D.L.. Forma estándar:

$$\frac{dy}{dx} + P(x)y = Q(x) \quad (1.16)$$

Si $Q(x)=0$, la E.D. 1.16 es **homogénea**

Si $Q(x) \neq 0$, la E.D. 1.16 **NO es homogénea**.

Para resolver este tipo de E.D. necesario un cambio de variable que se demuestra de la siguiente manera:

$$\frac{d}{dx} \left[e^{\int P(x)dx} \cdot y \right] \quad (1.17)$$

Derivando el producto de funciones:

$$e^{\int P(x)dx} \cdot y + e^{\int P(x)dx} \cdot \frac{dy}{dx}$$

Acomodando los términos:

$$\underbrace{e^{\int P(x)dx} \left[P(x)y + \frac{dy}{dx} \right]}_{Ecuacion1,17} \quad (1.18)$$

Tengamos en cuenta la igualdad entre las ecuaciones 1.17 y 1.18, por lo que es posible reemplazarla:

$$\underbrace{\left[P(x)y + \frac{dy}{dx} \right]}_{Ecuacion1,17} e^{\int P(x)dx} = Q(x) \cdot e^{\int P(x)dx}$$

$$\frac{d}{dx} \left[e^{\int P(x)dx} \cdot y \right] = Q(x) \cdot e^{\int P(x)dx}$$

Integrando a ambos lados

$$\begin{aligned} \int \frac{d}{dx} \left[e^{\int P(x)dx} \cdot y \right] dx &= \int Q(x) \cdot e^{\int P(x)dx} dx \\ e^{\int P(x)dx} \cdot y &= \int Q(x) \cdot e^{\int P(x)dx} dx \end{aligned} \quad (1.19)$$

Despejando y :

$$y = \frac{\int Q(x) \cdot e^{\int P(x)dx} dx}{e^{\int P(x)dx}} + C$$

Haremos un cambio de variable:

$$U(x) = e^{\int P(x)dx} \quad (1.20)$$

Haciendo el cambio de variable de la ecuación 1.3 en el resultado de la ecuación 1.19 se obtiene:

$$\boxed{U(x)Y(x) = \int Q(x)U(x)dx + C} \quad (1.21)$$

En conclusión, para resolver una ecuación diferencial, se usa la ecuación 1.21, teniendo en cuenta el cambio de variable de la ecuación 1.20.

Ejercicio 1.5 Resolver la siguiente E.D.L.:

$$(1-x^2) \frac{dy}{dx} - x = -xy$$

Para resolver esta E.D.L. es necesario darle la forma a la ecuación, osea ordenar los términos y encontrar las funciones P(x) y Q(x) de acuerdo a la ecuación 1.16. En este caso tenemos que aislar la diferencial ($\frac{dy}{dx}$), es por ello que dividimos todo entre ese término que acompaña a la diferencial. Luego hemos intercambiado los términos para tener del lado izquierdo la diferencial y una función que multiplique a y, mientras que del lado derecho tenemos que tener solo una función:

$$\begin{aligned} (1-x^2) \frac{dy}{dx} - x &= -xy \\ \frac{dy}{dx} + \frac{x}{(1-x^2)}y &= \frac{x}{(1-x^2)} \end{aligned} \quad (1.22)$$

$$P(x) = \frac{x}{(1-x^2)} \quad Q(x) = \frac{x}{(1-x^2)}$$

Una vez que tenemos identificamos las funciones P(x) y Q(x), tenemos que usar el cambio de variable(ecuación 1.20). Antes de seguir resolviendo la ecuación principal, resolvamos el cambio de variable. Para ello integramos la función P(x) como manda la ecuación 1.20:

$$\begin{aligned} \int \frac{x}{(1-x^2)} dx &= -\frac{1}{2} \ln|1-x^2| \\ \therefore e^{\int P(x)dx} &= (1-x^2)^{-\frac{1}{2}} = U(x) \end{aligned}$$

La solución anterior, si es un poco complicada de comprender, se uso la propiedad 15.5. Una vez que tenemos la función U(x), volvemos a la resolución principal, usando la función U(x) procedemos con la ecuación 1.21 (despejamos Y(x), para ello dividimos a ambos entre U(x)):

$$\begin{aligned} Y(x) &= \frac{\int \frac{x}{(1-x^2)} \cdot \frac{1}{\sqrt{1-x^2}} dx}{\frac{1}{\sqrt{1-x^2}}} + C \\ Y(x) &= \frac{\int \frac{x}{(1-x^2)^{\frac{3}{2}}} dx}{\frac{1}{\sqrt{1-x^2}}} + C \\ Y(x) &= \frac{-\frac{1}{2} \cdot 2(1-x^2)^{-\frac{1}{2}} + C}{(1-x^2)^{-\frac{1}{2}}} \end{aligned}$$

Separando el denominador para cada término del numerador:

$$Y(x) = 1 + C(1 - x^2)^{\frac{1}{2}} \quad (1.23)$$

■

1.4 Ecuaciones Diferenciales Exactas

Este tipo de ecuaciones toman como referencia a las ecuaciones diferenciales de primer orden.

Definición 1.4.1 — Ecuaciones Diferenciales exactas. En base a la ecuación 1.2, se busca la siguiente forma estándar para poder resolver:

$$M(x,y)dx + N(x,y)dy = 0 \quad (1.24)$$

Será exacta si cumple con la siguiente condición:

$$\frac{\partial M(x,y)}{\partial y} = \frac{\partial N(x,y)}{\partial x} \quad (1.25)$$

¿Cómo resolverla?

Si hemos comprobado que nuestra ecuación diferencial es exacta usando la ecuación 1.34, entonces existe una función $F(x,y)$ tal que:

$$\frac{\partial F(x,y)}{\partial x} = M(x,y) \quad (1.26a)$$

$$\frac{\partial F(x,y)}{\partial y} = N(x,y) \quad (1.26b)$$

En la ecuación 1.26a podemos determinar $F(x,y)$ si integramos $M(x,y)$ respecto a x , obviamente manteniendo a y como una constante. Si lo hacemos obtendremos:

$$F(x,y) = \int M(x,y)dx + G(y) \quad (1.27)$$

Añadimos la función $G(y)$ porque es una función de y implícita, es decir existe ahí, pero como derivamos en función a x , esta función $G(y)$ desaparece porque y es considerada constante.

De la ecuación 1.27, ahora derivamos respecto a y e igualamos este resultado con la ecuación 1.26b:

$$\frac{\partial F(x,y)}{\partial y} = \frac{\partial}{\partial y} \left(\int M(x,y)dx \right) + g'(y) = N(x,y) \quad (1.28a)$$

$$g'(y) = N(x,y) - \frac{\partial}{\partial y} \left(\int M(x,y)dx \right) \quad (1.28b)$$

Ahora integraremos la ecuación 1.28b respecto a y y reemplazamos en la ecuación 1.27. Con eso ya estaría resuelta la ecuación diferencial exacta. La solución general se da de la siguiente manera:

$$F(x,y) = C$$

Ejercicio 1.6 Resolver:

$$2y^2x - 3dx + 2yx^2 + 4dy = 0$$

Antes de todo, identifiquemos las funciones M y N en nuestra E.D.E:

$$\underbrace{2y^2x - 3dx}_{M(x,y)} + \underbrace{2yx^2 + 4dy}_{N(x,y)} = 0$$

Trata de guiarte por las diferenciales(dx y dy) para identificarlas bien. Ahora tenemos que comprobar que se trata de una E.D.E:

$$\frac{\partial M}{\partial y} = 4xy = \frac{\partial N}{\partial x}, \therefore \text{es exacta.} \quad (1.29)$$

Como es exacta, existe una función F(x,y), así que seguimos los pasos descritos anteriormente. Empezamos con la ecuación 1.26a:

$$\begin{aligned} \frac{\partial F(x,y)}{\partial x} &= 2y^2x - 3 \\ F(x,y) &= \int 2y^2x - 3dx + G(y) \\ F(x,y) &= y^2x^2 - 3x + G(y) \end{aligned} \quad (1.30)$$

Ya tenemos la respuesta, pero debemos hallar G(y) para poder reemplazarla en 1.30. En consecuencia seguimos con el paso detallado en la ecuación 1.26b:

$$\frac{\partial F(x,y)}{\partial y} = 2yx^2 + 4 \quad (1.31)$$

Lo que vamos a hacer es lo siguiente: Nosotros ya tenemos F(x,y) (ecuación 1.30) pero nos falta hallar G(y), es por ello que hallamos el término de N(x,y)(ecuación 1.26b). Ese término nos dice que si realizamos la derivada parcial respecto a y a la función F(x,y) obtendremos N(x,y). Nosotros tenemos F(x,y) y tenemos N(x,y); y tenemos una función desconocida que depende de y(G(y)), por consecuencia, hallaremos esa función desconocida:

$$\frac{\partial F(x,y)}{\partial y} = 2yx^2 + g'(y) = 2yx^2 + 4 \quad (1.32a)$$

$$g'(y) = 4 \quad (1.32b)$$

$$\int g'(y)dy = \int 4dy \quad (1.32c)$$

$$G(y) = 4y \quad (1.32d)$$

¿Qué se hizo?

Por si te cuesta entender, vamos a detallar:

- Ecuación 1.32a:** Hallamos el término correspondiente a la expresión 1.26b, que es la expresión 1.31; si nos centramos solo en esta expresión nos dice que: la derivada parcial de F(x,y) respecto a y es N(x,y). Nosotros sabemos cuánto vale N(x,y) pero no sabemos cuándo vale F(x,y). Pero, si nos fijamos bien, sabemos cuánto vale F(x,y)(ecuación 1.30), y si la derivamos podemos igualar a la expresión 1.31.

2. Ecuación 1.32b y 1.32c: Ya hallamos $g'(x)$, así que solo nos queda integrar **respecto a y** porque es una función de y .

Ahora que tenemos todo: solo nos queda reemplazar en la ecuación 1.30 el valor de la función incógnita $G(y)$ hallada en 1.32d:

$$\boxed{F(x,y) = y^2x^2 - 3x + 4y = C} \quad (1.33)$$

■

- R** Se puede cambiar el orden de resolver, es decir, primero hemos resuelto la expresión 1.26a y luego la expresión 1.26b, pero también es válido (y debería salir lo mismo) si haces primero la expresión 1.26b y luego la expresión 1.26a

1.5 Ecuaciones Diferenciales No Exactas

Este tipo de E.D.N.E son reducibles a E.D.E mediante un factor integrante.

Definición 1.5.1 — Ecuaciones Diferenciales exactas. En base a la ecuación 1.2, se busca la siguiente forma estándar para poder resolver:

$$M(x,y)dx + N(x,y)dy = 0$$

Será exacta si cumple con la siguiente condición:

$$\frac{\partial M(x,y)}{\partial y} \neq \frac{\partial N(x,y)}{\partial x} \quad (1.34)$$

¿Cómo resolver?

Tenemos que buscar un F.I.² para que la E.D.N.E se haga E.D.E.

Si:

$$\frac{1}{N(x,y)} \left(\frac{\partial M(x,y)}{\partial y} - \frac{\partial N(x,y)}{\partial x} \right) \quad (1.35)$$

devuelve como resultado una función que **SOLO** depende de x ; entonces el F.I. es:

$$U(x) = e^{\int \frac{1}{N(x,y)} \left(\frac{\partial M(x,y)}{\partial y} - \frac{\partial N(x,y)}{\partial x} \right) dx} \quad (1.36)$$

Si:

$$\frac{1}{M(x,y)} \left(\frac{\partial N(x,y)}{\partial x} - \frac{\partial M(x,y)}{\partial y} \right) \quad (1.37)$$

devuelve como resultado una función que **SOLO** depende de y ; entonces el F.I. es:

$$U(y) = e^{\int \frac{1}{M(x,y)} \left(\frac{\partial N(x,y)}{\partial x} - \frac{\partial M(x,y)}{\partial y} \right) dy} \quad (1.38)$$

Cuando se hallada multiplicamos a toda la E.D.N.E por F.I.; haciendo esto la E.D pasará a ser una E.D.E. y se procede a resolver como tal.

²Abreviación de Factor Integrante

Ejercicio 1.7 Resolver:

$$xydx + (2x^2 + 3y^2 - 20)dy = 0$$

Solución:

Queda de más que antes de todo, es necesario identificar $M(x,y)$ y $N(x,y)$:

$$\underbrace{xy}_{M(x,y)} dx + \underbrace{(2x^2 + 3y^2 - 20)}_{N(x,y)} dy = 0$$

Como primer paso, tenemos que ver si se trata de una E.D exacta o no exacta:

$$\left. \begin{array}{l} \frac{\partial M(x,y)}{\partial y} = x \\ \frac{\partial N(x,y)}{\partial x} = 4x \end{array} \right\} \neq \therefore \text{La E.D. no es exacta} \quad (1.39)$$

Procedemos con la solución de esta E.D.N.E., por lo tanto tendremos que hallar el F.I.. Empezaremos con la ecuación 1.35^a

$$\frac{1}{(2x^2 + 3y^2 - 20)} \left(\frac{\partial(xy)}{\partial y} - \frac{\partial(2x^2 + 3y^2 - 20)}{\partial x} \right) = \frac{-3x}{2x^2 + 3y^2 - 20}$$

Aquí tenemos un problema, se supone que la función resultante solo debe depender de x , pero la nuestra depende de y también. Descartamos esta.

Como esta primera comprobación para hallar F.I. no cumple la condición probamos con la otra(ecuación 1.37):

$$\frac{1}{xy} \left(\frac{\partial(2x^2 + 3y^2 - 20)}{\partial x} - \frac{\partial(xy)}{\partial y} \right) = \frac{3x}{xy} = \frac{3}{y}$$

Este condicionante para hallar el F.I. solo depende de y , así que nos sirve y por ende, trabajaremos con ese mismo, por lo que ahora hallaremos el F.I. según la expresión 1.38:

$$U(y) = e^{\int \frac{3}{x} dy} = y^3 \quad (1.40)$$

Cuando se halla encontrado el F.I. multiplicamos a toda la E.D.N.E por $U(y)$ para obtener una E.D.E.

$$\begin{aligned} & \textcolor{red}{y^3} (xydx + (2x^2 + 3y^2 - 20)dy) = 0 \\ & xy^4 dx + (2x^2y^3 + 3y^5 - 20y^3) = 0 \end{aligned} \quad (1.41)$$

Si en este momento procedemos a comprobar si es exacta, obtendremos:

$$\frac{\partial M}{\partial y} = 4xy^3 = \frac{\partial N}{\partial x}, \therefore \text{es exacta.}$$

Ahora podemos resolverla como lo hacemos para las E.D.E.:

$$F(x,y) = \int xy^4 dx + G(y)$$

$$F(x,y) = \frac{y^4 x^2}{2} + G(y)$$

Derivamos F(x,y) respecto a y e igualamos a N(x,y)

$$\cancel{2y^3x^2} + g'(y) = 2x^2y^3 + 3y^5 - 20y^3$$

$$g'(y) = 3y^5 - 20y^3$$

$$G(y) = y^6 - 5y^4$$

$$F(x,y) = \frac{y^4 x^2}{2} + \frac{y^6}{2} - 5y^4 = C$$

Multiplicando por dos a toda la expresión, teniendo en cuenta que 2 es constante:

$$\boxed{y^4 x^2 + y^6 - 10y^4 = C} \quad (1.42)$$

^aPuedes empezar con la ecuación 1.35 o 1.37, de notas formas, si una no cumple la condición usas la otra.

■

1.6 Ecuaciones por sustitución

1.6.1 Ecuaciones Homogéneas

Recordando lo aprendido en la ecuación 1.1, una ecuación de la forma:

$$M(x,y)dx + N(x,y)dy = 0$$

será homogénea si M(x,y) y N(x,y) son homogéneas del mismo grado:

$$M(xt, yt) = t^n M(x, y) \quad N(xt, yt) = t^n N(x, y)$$

o:

$$M(x, y) = x^n M(1, u) \quad N(x, y) = x^n N(1, u)$$

$$u = \frac{y}{x}$$

Al final, tenemos que devolver el cambio de variable u . Una ecuación diferencial homogénea siempre puede reducirse a una ecuación de variable separable por medio de una sustitución algebraica.

¿Cómo se hace?

Sea la ecuación diferencial de la forma 1.24 y sea homogénea. Se reduce a una E.D de variable separable usando cualquiera de las sustituciones:

$$y = u \cdot x \quad (1.43a)$$

$$x = v \cdot y \quad (1.43b)$$

$$u = \frac{y}{x} \quad (1.43c)$$

u y v son las nuevas variables dependientes.

R $x=v \cdot y$ se elige cuando $M(x,y)$ es de estructura más simple que la de $N(x,y)$.

Si elegimos $y=u \cdot x$ con su derivada:

$$y = u \cdot x \rightarrow dy = udx + xdu \quad (1.44)$$

Si reemplazamos 1.44 en 1.24:

$$M(x, ux)dx + N(x, ux) \underbrace{(udx + xdu)}_{dy} = 0$$

Aplicando la propiedad de homogeneidad a M y N es posible escribir:

$$x^n M(1, u)dx + x^n uN(1, u)dx + x^n xN(1, u)du = 0$$

$$x^n [M(1, u) + uN(1, u)]dx + x^n N(1, u)du = 0$$

Dividiendo a toda la expresión entre x^n

$$[M(1, u) + uN(1, u)]dx + N(1, u)du = 0$$

Si hacemos una división a nuestra conveniencia para arreglar la E.D.

$$\frac{dx}{x} + \frac{N(1, u)du}{M(1, u) + uN(1, u)} = 0$$

R El procedimiento anterior no debe ser memorizado, puesto que es solo una ejemplo de como se debe realizar las sustituciones. El procedimiento debe hacerse por completo.

La sustitución por $x=v \cdot y$ se realiza de manera similar.

Observaciones:

1. En la práctica, la sustitución de $x=v \cdot y$ se elige cuando la función $M(x,y)$ sea de estructura más simple que $N(x,y)$.
2. Cuando no hay diferencia apreciable entre M y N , se puede usar cualquiera de los dos sustituciones.
3. Si al resolver la que escogimos vemos que se torna complicada(algebraicamente) usamos la otra sustitución.

Ejercicio 1.8 Resolver:

$$(x - y)dx + xdy = 0 \quad (1.45)$$

Si hacemos las propiedades de la homogeneidad, notaremos que son homogéneas en grado 1. Elegiremos:

$$y=u \cdot x \rightarrow dy = udx + xdu \quad (1.46)$$

Reemplazando 1.8 en 1.8:

$$(x - ux)dx + x(udx + xdu) = 0$$

Dividimos toda la expresión entre x :

$$(1 - u)dx + (udx + xdu) = 0$$

Agrupando algebraicamente:

$$(1 - u + u)dx + xdu = 0 \longrightarrow dx + xdu = 0$$

Dividimos entre x^a :

$$\frac{dx}{x} + du = 0$$

Esta expresión ya es una E.D. de variable separable:

$$\int \frac{dx}{x} + \int du = \int x \\ \ln|x| + u = c$$

Devolviendo la variable u :

$$\boxed{\ln|x| + \frac{y}{x} = C}$$

^aEste paso depende de la función, recuerda que tenemos que buscar una E.D. de variable separable.



1.6.2 Ecuaciones Diferenciales de Bernoulli

Otro tipo de ecuación que se resuelven por sustitución son las ecuaciones diferenciales de Bernoulli.

Definición 1.6.1 La forma estándar de las E.D. de Bernoulli es:

$$\frac{dy}{dx} + P(x) \cdot y = Q(x) \cdot y^n \quad (1.47a)$$

$$\frac{dx}{dy} + P(y) \cdot x = Q(y) \cdot x^n \quad (1.47b)$$



En algunos casos, pueden haber funciones que acompañen a las $\frac{dy}{dx}$, pero ya se trata de un trabajo algebraico en llegar a las expresiones 1.47.

En ambos casos de las formas estándares 1.47, si $n=0$, $y=1$ en la ecuación 1.47a o $x=1$ en la ecuación 1.47b nos encontramos ante una E.D.L. y no hay problemas porque ya sabemos como resolver. Sin embargo si $n \neq 0$, $y \neq 1$ en la ecuación 1.47a o $x \neq 1$ en la ecuación 1.47b nos encontramos frente a un E.D.B³, estas E.D. se resuelven aplicando una sustitución.

El cambio de variable es:

$$W = 1^{1-n} \quad (1.48)$$

¿Cómo se llega a ese cambio de variable?⁴

$$\frac{dy}{dx} + P(x)y = Q(x)y^n$$

³Abreviación de Ecuación Diferencial de Bernoulli.

⁴Obviamente en esta caso $y \neq 1$ y $n \neq 0$.

Dividimos entre y^n :

$$y^{-n} \frac{dy}{dx} + P(x)y^{1-n} = Q(x) \quad (1.49)$$

Haciendo un cambio de variable con su derivada:

$$\begin{aligned} W &= y^{1-n} \rightarrow \frac{dw}{dx} = (1-n)y^{-n} \cdot \frac{dy}{dx} \\ \frac{1}{(1-n)} \cdot \frac{dw}{dx} &= y^{-n} \cdot \frac{dy}{dx} \end{aligned} \quad (1.50)$$

Sustituyendo la expresión 1.50 en 1.49:

$$\frac{1}{(1-n)} \cdot \frac{dw}{dx} + P(x)W = Q(x)$$

Multiplicando toda la expresión por (1-n):

$$\frac{dw}{dx} + \underbrace{(1-n)P(x)}_M W = \underbrace{(1-n)Q(x)}_N$$

Ahora es una E.D.L. y se procede con su procedimiento aprendido.

Ejercicio 1.9 Resolver:

$$(4-x^2) \frac{dy}{dx} + 4y = (2+x)y^2$$

Buscamos una expresión similar a 1.47a:

$$\frac{dy}{dx} + \frac{4}{(4-x^2)}y = \frac{2+x}{(4-x^2)}y^2$$

Dividimos entre y^2 :

$$y^{-2} \frac{dy}{dx} + \frac{4}{(4-x^2)}y^{-1} = \frac{2+x}{(4-x^2)} \quad (1.51)$$

Efectuamos el cambio de variable W:

$$\begin{aligned} W &= y^{1-2} \rightarrow \frac{dw}{dx} = (1-2)y^{-2} \cdot \frac{dy}{dx} \\ -\frac{dw}{dx} &= y^{-2} \cdot \frac{dy}{dx} \end{aligned} \quad (1.52)$$

Reemplazando la expresión 1.52 en 1.51:

$$\begin{aligned} -\frac{dw}{dx} + \frac{4W}{(4-x^2)} &= \frac{2+x}{4-x^2} \\ \frac{dw}{dx} + \frac{4}{(x^2-4)} \cdot W &= \frac{1}{x-2} \end{aligned}$$

Ahora se procede a resolver como una E.D.L.

$$U(x) = e^{\int \frac{4}{(x-2)} dx}$$

$$U(x) = e^{\ln|x+2|}$$

$$U(x) = \frac{x-2}{x+2}$$

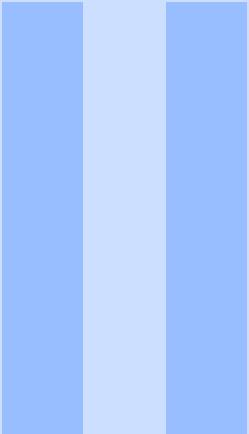
$$\frac{x-2}{x+2} \cdot W(x) = \int \frac{1}{(x-2)} \cdot \frac{(x-2)}{(x+2)} dx + C$$

$$W(x) = [\ln|x+2| + C] \cdot \frac{x+2}{x-2}$$

Devolviendo el cambio de variable $W = y^{(1-2)}$:

$$\boxed{y^{-1} = [\ln|x+2| + C] \cdot \frac{x+2}{x-2}} \quad (1.53)$$

■



Sistemas Digitales

2	Lógica digital	31
2.1	Compuertas lógicas	
2.2	Formas de expresar funciones booleanas	
2.3	Axiomas	
2.4	Teoremas del álgebra de conmutación	
2.5	Mapas de Karnaugh	
2.6	Parámetros TTL y CMOS	
3	Circuitos Lógicos	57
3.1	Sumador	
3.2	Codificador y Decodificador	
3.3	Multiplexor y Demultiplexor	
3.4	Latch	
3.5	Flip-Flops	
3.6	División de frecuencia	
3.7	Monoestables	
4	Circuitos Lógicos Secuenciales	81
4.1	Máquina de estados	
4.2	Grafos de estados	

2. Lógica digital

Un sistema digital binario es un conjunto de dispositivos que son destinados a la generación, transmisión, manejo, procesamiento y almacenamiento de señales digitales.

2.1 Compuertas lógicas

2.1.1 Compuerta NOT

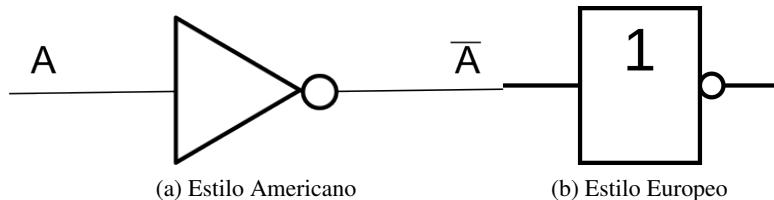


Figura 2.1: Simbología compuerta NOT

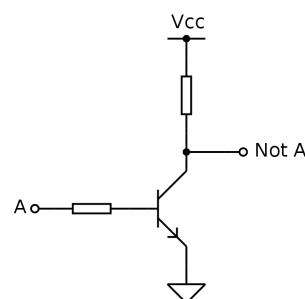


Figura 2.2: Implementación electrónica: Compuerta NOT

IN	OUT
0	1
1	0

Cuadro 2.1: Tabla de verdad de la compuerta NOT

2.1.2 Compuerta AND

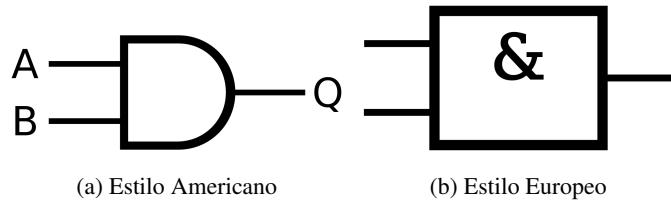


Figura 2.3: Simbología compuerta AND

Transistor AND Gate

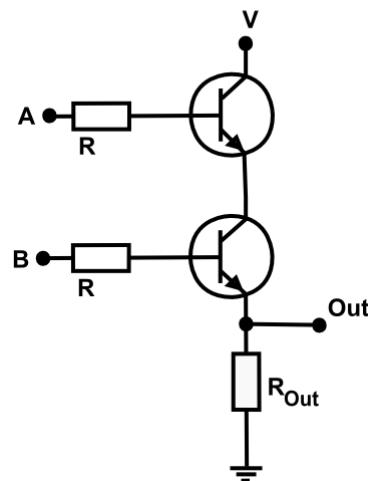


Figura 2.4: Implementación electrónica: Compuerta AND

IN		OUT
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Cuadro 2.2: Tabla de verdad de la compuerta AND

Se puede caracterizar como:

$$A \times B \times C \times \dots = X$$

La salida será 1 solo si todas las entradas son 1.

2.1.3 Compuerta OR

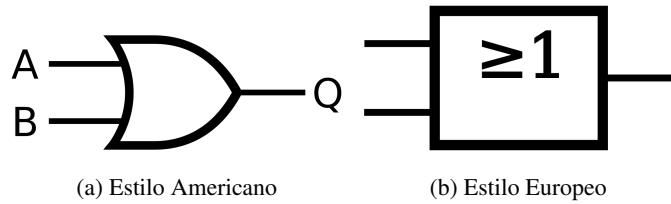


Figura 2.5: Simbología compuerta OR

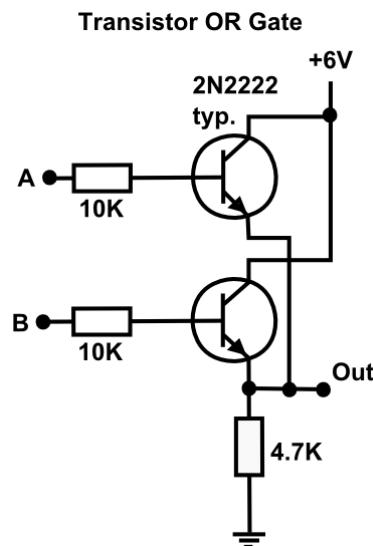


Figura 2.6: Implementación electrónica: Compuerta OR

IN		OUT
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Cuadro 2.3: Tabla de verdad de la compuerta OR

$$A + B + C + \dots = X$$

La salida será 0 solo si todas las entradas son 0.

2.1.4 Compuerta NAND

Solo cuando todas sus entradas son ALTO, la salida será BAJO.

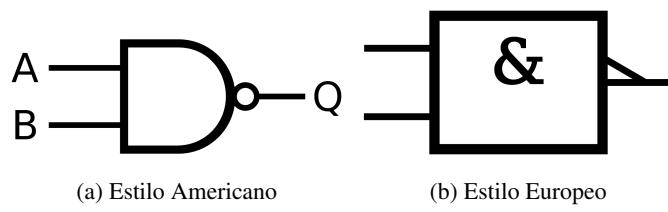


Figura 2.7: Simbología compuerta NAND

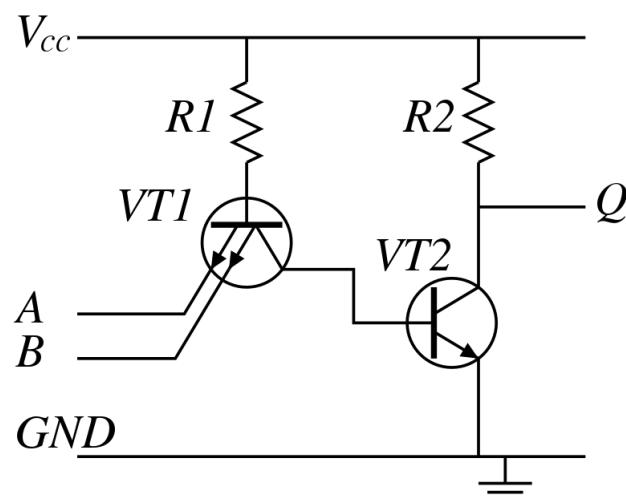


Figura 2.8: Implementación electrónica: Compuerta NAND

IN		OUT
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Cuadro 2.4: Tabla de verdad de la compuerta NAND

2.1.5 Compuerta NOR

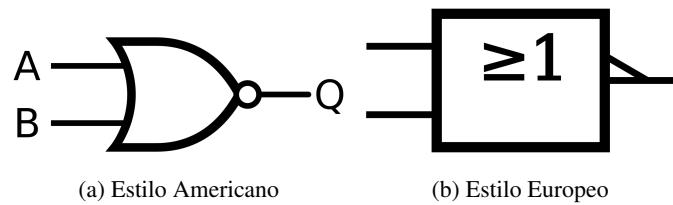


Figura 2.9: Simbología compuerta NOR

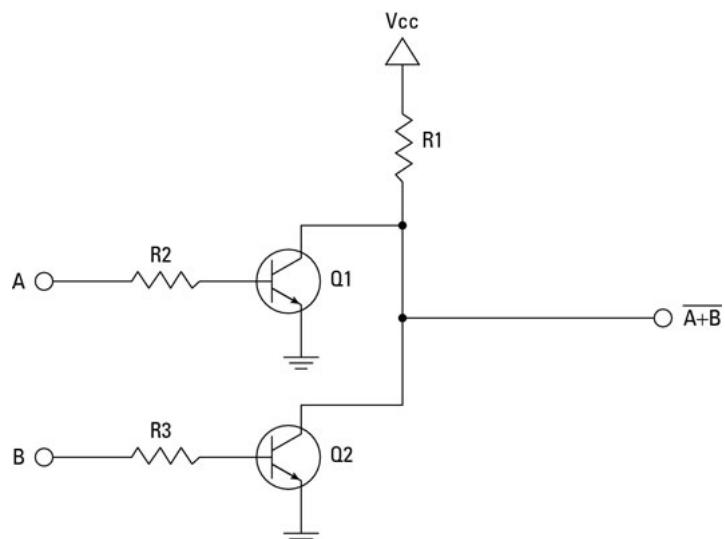


Figura 2.10: Implementación electrónica: Compuerta NOR

IN		OUT
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

Cuadro 2.5: Tabla de verdad de la compuerta NOR

Solo si todas las entradas son 0, la salida será 1.

2.1.6 Compuerta OR exclusiva-XOR

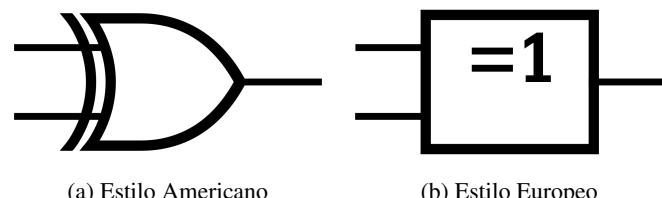


Figura 2.11: Simbología compuerta XOR

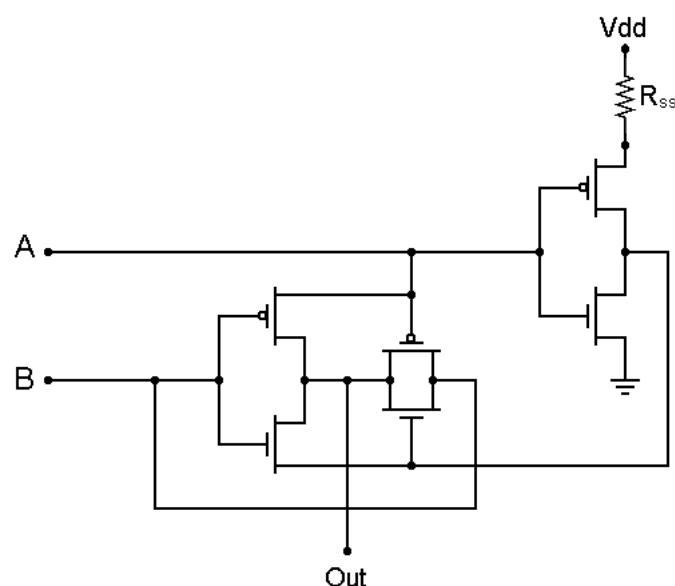


Figura 2.12: Implementación electrónica: Compuerta XOR

IN		OUT
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Cuadro 2.6: Tabla de verdad de la compuerta XOR

2.1.7 Compuerta NOR exclusiva-XNOR

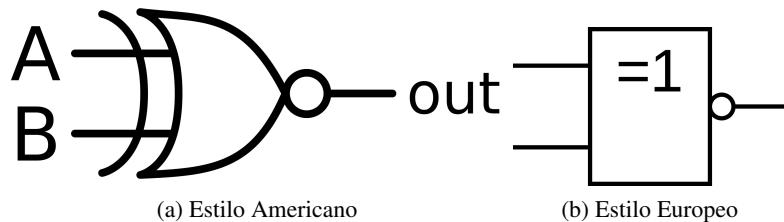


Figura 2.13: Simbología compuerta XNOR

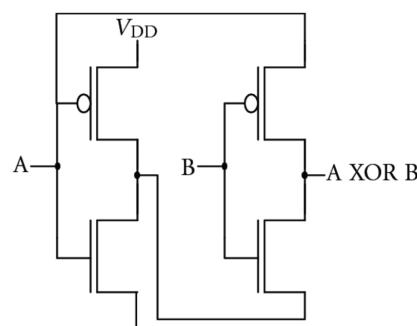


Figura 2.14: Implementación electrónica: Compuerta XNOR

IN		OUT
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Cuadro 2.7: Tabla de verdad de la compuerta XNOR

2.2 Formas de expresar funciones booleanas

Existen formas de expresar las funciones booleanas: como **suma de productos** o como **productos de sumas**.

2.2.1 Suma de productos-SOP

Suma(**OR**) de términos(**AND**), formados por varias variables complementadas o no. Aquí nos importan las salidas con nivel **ALTO**. Como entradas, un nivel alto será $A=1$ mientras que un nivel bajo es $A'=0$.

$$f(a, b, c) = \bar{a}bc + a\bar{b}\bar{c} + abc + c$$

2.2.2 Productos de sumas-POS

Productos(**AND**) de sumas(**OR**) formadas por varias variables complementadas o no. Aquí nos importan las salidas con nivel **BAJO**. Como entradas, un nivel alto será $A'=1$ mientras que un nivel

abajo es A=0.

$$F(a,b,c) = (A+B+C)(A+B'+C')(C'+A)$$



La terminología de la negación puede tener varias formas, en este libro usaremos dos: \bar{A} o A' . Ambos significan lo mismo. Asimismo, para el POS usaré letras mayúsculas y para SOP usaré letras minúsculas. Lo más importante es que entiendas como funciona todo así la terminología no será un problema.

Se detallará un ejemplo para poder explicar los SOP y POS para 3 variables. La tabla puede ser expandida para más variables pero todo funcionará bajo la misma lógica.

F	A	B	C	Min término	Notación minterm	Max término	Notación maxterm
0	0	0	0	$A'B'C'$	m_0	$A+B+C$	M_0
1	0	0	1	$A'B'C$	m_1	$A+B+C'$	M_1
2	0	1	0	$A'BC'$	m_2	$A+B'+C$	M_2
3	0	1	1	$A'BC$	m_3	$A+B'+C'$	M_3
4	1	0	0	$AB'C'$	m_4	$A'+B+C$	M_4
5	1	0	1	$AB'C$	m_5	$A'+B+C'$	M_5
6	1	1	0	ABC'	m_6	$A'+B'+C$	M_6
7	1	1	1	ABC	m_7	$A'+B'+C'$	M_7

Cuadro 2.8: Notación de *minterm* y *maxterm*.

Ejercicio 2.1 Se presenta la tabla 2.9, expresar la misma tabla como SOP y como POS.

SOP:

En SOP, solo nos fijamos en las salidas de nivel ALTO(1). Una vez que las hayamos ubicado escribimos las filas tal y como están expresando las entradas como un producto. Por ejemplo, el primer ALTO se encuentra en la posición de F_1 , donde A y B son nivel BAJO y C es nivel ALTO. Escribimos, según la teoría, donde ALTO es A y BAJO es A' , el primer sumando como: $A'B'C$. Repetimos el mismo paso para F_3, F_4, F_5, F_6 y F_7 puesto que tienen un nivel ALTO como salida. Si seguimos la lógica que se vio y sumando cada sumando obtendremos:

$$f(a,b,c) = \underbrace{A'B'C}_{F_1} + \underbrace{A'BC}_{F_3} + \underbrace{AB'C}_{F_5} + \underbrace{ABC'}_{F_6} + \underbrace{ABC}_{F_7}$$

Esta expresión también puede ser reducida usando notación dicha en la tabla 2.8. Como la tabla y nuestro ejemplo son de 3 variables de entrada, solo buscamos la equivalencia dentro de la columna *Notación minterm*, la forma reducida sería:

$$f(a,b,c) = m_1 + m_3 + m_5 + m_6 + m_7$$

POS:

Para POS, en vez de fijarnos en los niveles ALTOS, nos fijamos en los niveles BAJOS y seguimos la misma lógica vista en SOP, sin embargo, ahora en vez de cada fila sea expresada como un producto y luego sumemos todos, en este caso cada fila se expresa como una suma, y al agrupar todos los términos será mediante una suma. Así, los ceros están en las ubicaciones F_0, F_2 y F_4 ,

entonces, según la teoría 1 será A' y 0 será A:

$$F(A, B, C) = \underbrace{(A + B + C)}_{F_0} \cdot \underbrace{(A + B' + C)}_{F_2} \cdot \underbrace{(A' + B + C)}_{F_4}$$

Según la tabla 2.8 también puede ser simplificado:

$$F(A, B, C) = M_0 M_2 M_5$$

■

F_1	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Cuadro 2.9: Tabla ejemplo 1

2.2.3 Conversión de SOP a POS

A veces es necesario expresar de una forma u otra, es por eso que usa la conversión de SOP a POS. Se explicará usando un ejemplo:

$$f(x, y, z) = x'y'z' + x'yz + x'yz' + xy'z$$

- Evaluamos los binarios de los sumando, recordando las equivalencias:

$$f(x, y, z) = \underbrace{x'y'z'}_{\begin{matrix} 000 \\ 0 \end{matrix}} + \underbrace{x'yz}_{\begin{matrix} 011 \\ 3 \end{matrix}} + \underbrace{x'yz'}_{\begin{matrix} 010 \\ 2 \end{matrix}} + \underbrace{xy'z}_{\begin{matrix} 101 \\ 5 \end{matrix}}$$

El color rojo es su equivalente en binario mientras que el azul lo es en decimal.

- Identificamos los términos faltantes, como es una función lógica de 3 variables: $2^3 = 8$, en el paso uso, con la ayuda de nuestra tabla 2.8 logramos identificar a los términos 0,3,2,5; por consecuente nos faltan los términos 1, 4, 6 y 7 cuyos binarios son:

- 1=001
- 4=100
- 6=110
- 7=111

- Ahora escribimos esos mismos términos que faltan(1, 4, 6 y 7) como POS:

$$f(a, b, c) = \underbrace{(A + B + C')}_{\begin{matrix} 1 \\ 1 \end{matrix}} \cdot \underbrace{(A' + B + C)}_{\begin{matrix} 4 \\ 4 \end{matrix}} \cdot \underbrace{(A' + B' + C)}_{\begin{matrix} 6 \\ 6 \end{matrix}} \cdot \underbrace{(A' + B' + C')}_{\begin{matrix} 7 \\ 7 \end{matrix}}$$

2.2.4 Forma canónica

Función en donde cada sumando o multiplicando aparecen todas las entradas, sin importar si están negada o no; en otras palabras, si se tiene una función de tres variables $f(a,b,c)$, una expresión canónica puede ser:

$$f(a,b,c) = ab'c + a'b'c + abc + a'b'c'$$

2.2.5 Forma normalizada

Reducción de la forma canónica mediante el álgebra de commutación. De tener, por ejemplo, una función $f(a,b,c)$, una forma normalizada sería:

$$f(a,b,c) = a + bc' + a'c$$

Vemos que en cada sumando o producto no están presentes todas las variables.

2.2.6 ¿Cómo pasar de la forma normalizada a la canónica?

1. **Identificar** los sumandos de **NO** estén completos, usaremos de ejemplo la siguiente función lógica: $f(a,b,c) = a + bc + \overline{abc}$ ¹

$$f(a,b,c) = \textcolor{red}{a} + \textcolor{blue}{bc} + \textcolor{blue}{\overline{abc}}$$

Los términos en color **rojo** son los términos que están incompletos, es decir, no están presentes todas las variables a excepción del tercer sumando de **azul**.

2. **Completar** los términos incompletos. Aunque haya una teoría detrás de todo, lo más rápido es completar los términos incompletos con todas las variaciones posibles. En este caso tenemos 3 variables, así que procedemos:

- **a:** En el caso de tener un solo término, en nuestro ejemplo a , sabemos que a este sumando le faltan 2 variables: b y c . Entonces repetimos el término unitario (a en nuestro caso) 2^n veces, donde n es el número de variables faltantes; en este caso es $n=2$ porque falta b y c . Una vez que hayamos repetidos 2^n veces la variable unitaria, a cada una de ellas escribimos las posibles combinaciones que pueden tomar las dos variables restantes:

$$\begin{array}{ccc} \textcolor{red}{a} & \bar{b} & \bar{c} \\ \textcolor{red}{a} & \bar{b} & c \\ \textcolor{red}{a} & b & \bar{c} \\ \textcolor{red}{a} & b & c \end{array}$$

Hemos generado 4 términos.

- **bc:** Repetimos el mismo paso anterior, ahora n será 1, por lo tanto $2^n = 2$, repetimos dos veces el término bc y completamos con las posibles variantes del término faltante:

$$\begin{array}{ccc} \textcolor{red}{b} & \textcolor{red}{c} & \bar{a} \\ b & c & a \end{array}$$

Hemos generado 2 términos.

El procedimiento será el mismo para más variables, sin embargo para más variables será más largo el proceso.

¹Otra manera de expresar la negación aparte de \bar{A} , A' es $/A$

3. **Sumamos** todos los términos hallados, incluso los que no hemos tocado (\overline{abc}), al sumarlos todos debemos tener el cuenta el teorema 2.4.3, en otras palabras, términos repetidos solo se colocan una vez.



Es común que escuchemos diferentes formas de expresar un circuito lógico. Una común es expresión booleana: Esta es una forma de expresar la salida como las operaciones que se realizan. Por ejemplo, sean las entradas A y B, la salida de una compuerta AND es $A \cdot B$, la compuerta OR se expresa como $A+B$, la compuerta XOR \oplus y así existen otros operadores para las demás compuertas y circuitos lógicos.

2.3 Axiomas

Los axiomas son conjuntos mínimos de definiciones básicas que suponemos como verdaderas:

1. $X=0$ si $X \neq 1 \vee X=1$ si $X \neq 0$
2. Si $X=0 \Rightarrow X'=1 \wedge$ Si $X=1 \Rightarrow X'=0$
3. $0 \cdot 0=0 \wedge 1+1=1$
4. $1 \cdot 1=1 \wedge 0+0=0$
5. $0 \cdot 1=1 \cdot 0=0 \wedge 1+0=0+1=1$

2.4 Teoremas del álgebra de conmutación

Los teoremas siguientes son la base de la simplificación de circuitos complejos:

Theorem 2.4.1 — Identidad.

$$X + 0 = X \tag{2.1}$$

$$X \cdot 1 = X \tag{2.2}$$

Theorem 2.4.2 — Elemento nulo.

$$X + 1 = 1 \tag{2.3}$$

$$X \cdot 0 = 0 \tag{2.4}$$

Theorem 2.4.3 — Potencias idénticas.

$$X + X = X \tag{2.5}$$

$$X \cdot X = X \tag{2.6}$$

Theorem 2.4.4 — Involución.

$$(X')' = X \tag{2.7}$$

Theorem 2.4.5 — Complementos.

$$X + X' = 1 \tag{2.8}$$

$$X \cdot X' = 0 \tag{2.9}$$

Theorem 2.4.6 — Comutatividad.

$$X + Y = Y + X \quad (2.10)$$

$$X \cdot Y = Y \cdot X \quad (2.11)$$

Theorem 2.4.7 — Asociatividad.

$$(X + Y) + Z = X + (Y + Z) \quad (2.12)$$

$$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z) \quad (2.13)$$

Theorem 2.4.8 — Distributividad.

$$X \cdot Y + X \cdot Z = X \cdot (Y + Z) \quad (2.14)$$

$$(X + Y)(X + Z) = X + Y \cdot Z \quad (2.15)$$

Theorem 2.4.9 — Cubierta.

$$X + X \cdot Y = X \quad (2.16)$$

$$X(X + Y) = X \quad (2.17)$$

Theorem 2.4.10 — Combinación.

$$X \cdot Y + X \cdot Y' = X \quad (2.18)$$

$$(X + Y)(X + Y') = X \quad (2.19)$$

Theorem 2.4.11 — Consenso.

$$X \cdot Y + X' \cdot Z = X \cdot Y + X' \cdot Z \quad (2.20)$$

$$(X + Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z) \quad (2.21)$$

Theorem 2.4.12 — DeMorgan.

$$\overline{(X + Y)} = \overline{X} \cdot \overline{Y} \quad (2.22)$$

$$\overline{(X \cdot Y)} = \overline{X} + \overline{Y} \quad (2.23)$$

2.4.1 Dualidad

Cualquier teorema o identidad en el álgebra de conmutación continua siendo verdadero si tanto 0 y 1 como · y + son intercambiados en todas partes.

Ejercicio 2.2 Dada la tabla 2.10, encuentre las formas canónicas de la forma POS y SOP:

Solución

Salida R

POS:

$$R = \underbrace{(X + Y + Z)}_{M_0} \underbrace{(X + Y + Z')}_{M_1} \underbrace{(X + Y' + Z)}_{M_2} \underbrace{(X' + Y + Z)}_{M_4}$$

$$R = \prod M(0, 1, 2, 4)$$

SOP:

$$R = \underbrace{X'YZ}_{m_3} + \underbrace{XY'Z}_{m_5} + \underbrace{XYZ'}_{m_6} + \underbrace{XYZ}_{m_7}$$

$$R = \sum m(3, 5, 6, 7)$$

Salida S

POS:

$$R = \underbrace{(X + Y + Z)}_{M_0} \underbrace{(X + Y' + Z')}_{M_3} \underbrace{(X' + Y + Z')}_{M_5} \underbrace{(X' + Y' + Z)}_{M_6}$$

$$R = \prod M(0, 3, 5, 6)$$

SOP:

$$R = \underbrace{X'Y'Z}_{m_1} + \underbrace{X'YZ'}_{m_2} + \underbrace{XYZ'}_{m_4} + \underbrace{XYZ}_{m_7}$$

$$R = \sum m(1, 2, 4, 7)$$

■

F	X	Y	Z	R	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Cuadro 2.10: Ejercicio 2

Corolario 2.4.13 — Llenado de tablas de verdad. Antes de pasar a los Mapas de Karnaugh, hacer las tablas de verdad son fáciles independientemente del número de variables. Se ubica todas las variables en la tabla y se procede a llenar cada columna de izquierda a derecha, sin embargo, en las filas intercalando ceros y unos de acuerdo a la posición de la columna restada en una unidad. Por, ejemplo para la columna número 1(de derecha a izquierda), se intercalan

ceros y unos cada 2^0 posiciones siempre empezando desde 0; en otras palabras en la columna escribiremos 0-1-0-1-0-1-0....

Para la columna 2, los ceros y unos se intercalan cada 2^1 posiciones; en otras palabras escribiremos 0-0-1-1-0-0-1-1....Para la columna 3, 2^2 ; 0-0-0-1-1-1-0-0-0....

El proceso funciona para n variables independiente de el número de variables.

2.5 Mapas de Karnaugh



Importante: En todo el desarrollo de este capítulo el bit más significativo será A y el menos significativo será D,E,F dependiendo del número de variables.

Los mapas de Karnaugh son una forma de expresar las tablas de verdad de varias variables. Nos permite no escribir muchas combinaciones para expresar varias funciones. Los mapas de Karnaugh tienen la siguientes formas:

	<i>B</i>	
	0	1
<i>A</i>	0	
1		

2.5.1 Mapas de Karnaugh-4 variables

Las posiciones se detallan en el siguiente cuadro:

	<i>C</i>	
	0	1
<i>AB</i>	00	
	01	
	11	
	10	
	0	1
	2	3
	6	7
	4	5

Las posiciones reales cambian de acuerdo al número de variables, las posiciones reales para 4 variables es:

		CD			
		00	01	11	10
AB	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

Las posiciones reales se usan en base a los unos y ceros ubicados en base a su posición en el cuadro 2.8.

No es necesario memorizar las posiciones, puesto que las posiciones reales no son más que la forma decimal de las variables. Del cuadro de 4 variables visto antes, por ejemplo, cada número será de la forma **ABCD**: el cero es **0000**, la posición 1 es **0001**, la posición 15 es **1111**. Este cambio se orden en los términos 2-3, 6-7, 14-15 y 10-11 se produce porque usamos la codificación Gray.

Es importante decir que los laterales del los mapas de Karnaugh son cíclicos, es decir:

- Los cuadros limítrofes del cuadro 14 son, 6 por arriba, 15 por la izquierda, 10 por abajo y 12 por la derecha.
- Los cuadros limítrofes del cuadro 10 son, 14 por arriba, 11 por la izquierda, 2 por abajo y 8 por la derecha.
- Los cuadros limítrofes del cuadro 8 son, 12 por arriba, 10 por la izquierda, 0 por abajo y 9 por la derecha.

Recordemos que cuando trabajamos con SOP solo nos importa los 1 y para POS solo importa los 0. Ademas el bit que esta a la izquierda es el **Bit más significativo(MSB)** y hacia la derecha se encuentra el **Bit menos significativo(LSB)**.

- R** En los mapas de Karnaugh sí y solo sí se agrupan en grupos de base 2(1, 2, 4, 8...). Siempre se prioriza agrupar la mayor parte de, se pueden compartir bits entre grupos pero siempre agrupando la mayor cantidad de bits.

Ejercicio 2.3 Expresar la siguiente expresión en un mapa de Karnaugh:

$$Z = \overline{ABC}D + \overline{A}\overline{B}\overline{C}D + A\overline{B}\overline{C}D + ABCD + A\overline{B}CD + \overline{ABC}D + A\overline{B}\overline{C}\overline{D}$$

Solución:

Tenemos dos opciones, la más fácil es pasar cada sumando a binario pero es requisito saber los números decimales en el sistema binario. Para ello tenemos que recodar las equivalencias de acuerdo si estamos frente a un SOP o POS, de ahí, son alterar el orden del bit más significativo tan solo escribimos ceros y unos:

$$Z = 0011 + 0100 + 1101 + 1111 + 1100 + 0001 + 1010$$

Una vez que lo hemos pasado a binario, tenemos que pasarlo a decimal para saber las casillas

donde colocaremos unos, si fuera POS, sabremos las casillas donde colocaremos ceros:

$$Z = 3 + 4 + 13 + 15 + 12 + 1 + 10$$

Ya tenemos los números de casillas donde tenemos que colocar unos. Dibujamos nuestro mapa y colocamos unos **SOLO** en las casillas que nos resultó de pasar de binario a decimal. En las demás casillas en blanco colocaremos ceros.

		CD			
		00	01	11	10
AB	00	0	1	1	0
	01	1	0	0	0
	11	1	1	1	0
	10	0	0	0	1

■

2.5.2 Agrupación de términos

Algo importante con los mapas de Karnaugh es la agrupación de términos que se mencionó anteriormente. La agrupación se hace una vez terminada la realización de los mapas de Karnaugh, se explicará con un ejemplo:

Dada la función lógica:

$$Z = A + \bar{C}D + A\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D}$$

Al contrario del ejemplo anterior, esta forma no es de la forma canónica, sino que es de la forma normalizada. Lo que se haría normalmente es escribir todas las posibles combinaciones para cada sumando; por ejemplo, para el primer sumando(A) le faltan 3 términos, escribiríamos 2^3 combinaciones faltantes, para el segundo sumando sería 2^2 combinaciones faltantes y así sucesivamente, es un método valido pero lento. Hay un método más sencillo que lo explicaré luego, ahora primero vamos a agrupar. La función lógica Z , generará el siguiente mapa de Karnaugh:

		CD				
		00	01	11	10	
AB		00	0	1	0	0
		01	0	1	0	1
		11	1	1	1	1
		10	1	1	1	1

Agrupamos, teniendo en cuenta que tenemos que agrupar la mayor cantidad de dígitos posibles y es valido compartir bits, la agrupación correcta es:

		CD				
		00	01	11	10	
AB		00	0	1	0	0
		01	0	1	0	1
		11	1	1	1	1
		10	1	1	1	1

Puedes apreciar que el sector verde tiene una intersección con el sector rosado, es posible agrupar los dos unos que están fuera de esa intersección pero siempre tienes que tratar de agrupar la mayor cantidad posible, puesto que los mapas de Karnaugh son un método para **simplificar** funciones lógicas. Con el sector amarillo no tenemos de otra que agrupar de dos, ya que 3 es un número invalido de agrupación.

La función tiene 4 sumandos pero la agrupación en el mapa de Karnaugh solo tenemos 3 sectores, eso quiere decir que es puede simplificar la función Z:

- **Sector verde:** Concéntrate en el sector azul, la columna que no dice: que tanto C y D no están cambiando su valor, siempre están siendo 0 y 1. Ademas fíjate que pasa con A y B a lo largo de esa columna: están cambiando todos sus valores. Podemos decir que: *En el sector azul, los valores de C y D se mantienen estáticos pero A y B son valores cambiantes para todo C y D.* En conclusión, escribiremos el primer sumando como:

$$\bar{C}D$$

Solo es CD porque los valores para A y B no nos importan ya que estamos "englobando" todas sus variantes(de A y B).

- **Sector rojo:** Si nos centramos en el sector A, es un caso distinto. Analizamos el comportamiento para A: siempre será 1. Para B: Depende, pues toma tanto valor 0 como 1. C y D:

No se puede decir mucho de ellos puesto que varían totalmente, en otras palabras: no nos importa sus variantes pues el sector cubre todas ellas. En consecuencia, la única variable “estable” es A valiendo 1, pues si A vale 1 las variables no nos importan su comportamiento ya que se engloba todo, es por ello que el segundo sumando resultante de este sector es:

A

- **Sector amarillo:** Este es el sector más pequeño, así que lo analizaremos: Para A: Si cambia, puede ser 0 como 1. Para B: No cambia, se mantiene estático valiendo 1. Para C y D: no cambia, sí o sí tiene que ser C=1 y D=0. En consecuencia, el valor de A no nos importa por que puede ser 0 como 1 pero los demás valores son importante porque delimitan el sector, el tercer sumando será:

BC \bar{D}

Como resultado final, la expresión de 4 sumando puede ser expresada como 3 sumando, siendo la nueva función simplificada:

$$Z = A + \bar{C}D + BC\bar{D}$$

Ejercicio 2.4 Expresar la siguiente función lógica en mapa de Karnaugh:

$$Z = \bar{C}D + A\bar{B}CD + A\bar{B}\bar{C}\bar{D} + AB\bar{C}\bar{D} + BCD$$

Solución:

Lo haremos de la forma larga por si no le agarras truco a los mapas de Karnaugh. Escribimos sus formas binarias y completamos los sumandos incompletos para hallar las formas binarias ya que esta la forma normalizada:

$$Z = \bar{C}D + \underbrace{A\bar{B}CD}_{1010} + \underbrace{A\bar{B}\bar{C}\bar{D}}_{1000} + \underbrace{AB\bar{C}\bar{D}}_{1100} + BCD$$

$$\bar{A}\bar{B}\bar{C}D(0001)$$

$$\bar{A}B\bar{C}D(0111)$$

$$\bar{A}\bar{B}C\bar{D}(0101)$$

$$A\bar{B}CD(1111)$$

$$A\bar{B}\bar{C}D(1001)$$

$$AB\bar{C}D(1101)$$

Ahora puedes ubicar en todas esas posiciones **unos**, puedes pasarlo a decimal si gustas o trabajar tal como están expresadas en binario recordando la forma ABCD. Lo expresamos en su mapa de Karnaugh y agrupamos:

		CD			
		00	01	11	10
AB	00	0	1	0	0
	01	0	1	1	0
	11	1	1	1	0
	10	1	1	0	1

En el mapa de Karnaugh se muestran 4 sectores, por lo tanto, 4 sumandos:

$$Z = \bar{C}D + A\bar{C} + BD + A\bar{B}\bar{D}$$

Solo céntrate en las variables que no cambian, las estáticas pues las dinámicas que cambian no nos importan tanto, pues se sobreentiende que ya están agrupadas.

Si se desea también se puede expresar como POS, es absolutamente lo mismo pero nos centramos en los **ceros** y agrupamos como siempre:

		CD			
		00	01	11	10
AB	00	0	1	0	0
	01	0	1	1	0
	11	1	1	1	0
	10	1	1	0	1

La expresión en SOP es:

$$Z = (A + D) \cdot (B + \bar{C} + \bar{D}) \cdot (\bar{B} + \bar{C} + D)$$

Nota que ahora $A=0$ y $\bar{A}=1$, el desarrollo es el mismo que SOP, solo ahora cada factor contiene sumas y se unen todos los factores mediante un producto. ■



En necesario no olvidar lo que caracteriza a SOP y POS:

SOP es una **SUMA** donde cada sumando esta compuesto por **FACTORES**, aquí $A=1$ y $\bar{A}=0$.

Por el otro lado POS es un **PRODUCTO** donde cada factor esta compuesto por **SUMAS**, aquí $A=0$ y $\bar{A}=1$.

Corolario 2.5.1 — Forma de sumatoria y productoria. Si recordamos la formas de expresar las funciones lógicas mediante sumatorias ($\sum(1, 2, 3, 5)$) o productorias ($\prod(1, 2, 3, 5)$), solo colocamos **unos** en las posiciones indicadas por la sumatoria y **ceros** en las posiciones indicadas por la productoria; ademas llenamos con sus negadas en los cuadros que restan.

2.5.3 Mapas de Karnaugh-5 variables

Se expresa en dos tablas. Se presenta la numeración para un mapa de 5 variables:

		DE						DE				
		00	01	11	10			00	01	11	10	
BC		00	0	1	3	2			16	17	19	18
		01	4	5	7	6			20	21	23	22
		11	12	13	15	14			28	29	31	30
		10	8	9	11	10			24	25	27	26
A=0						A=1						

Para poder expresar en SOP y POS la tabla E=0 se superpone encima de la tabla E=1

		DE						DE				
		00	01	11	10			00	01	11	10	
BC		00				1						1
		01		1	1	1			1	1	1	1
		11	1	1		1						1
		10	1	1		1			1			1
A=0						A=1						

Imaginariamente solapa A=1 sobre A=0, notaremos que hay sectores que coinciden como el sector verde con el rojo y amarillo con celeste. Podemos escribir el sector verde y rojo como una sola expresión: no nos importa A, no nos importa B y C, solo tenemos que definir bien D y E pues son fijos:

$$D\bar{E}$$

Para los sectores amarillo y celeste: E no nos importa puesto que puede ser 0 como 1, B y C si importan pues son estables, no cambian. D tampoco nos importa, puede ser 0 como 1, sin embargo, E si importa ya que si o si tiene que ser 1:

$$\bar{B}CE$$

Los otros sectores(azul y rosado) son únicos ya que no tienen coincidencia así que se escribirán de forma única:

$$\overline{BDA}$$

$$\overline{BCDE}$$

Siendo la expresión final de la tabla:

$$Z = D\bar{E} + \bar{B}CE + \overline{BDA} + \overline{BCDE}$$

Si se realiza el álgebra de conmutación el término de la casilla número 25 desaparecerá.

2.5.4 Mapas de Karnaugh-6 variables

Mientras más variables se tengan, los mapas de karnaugh dejan de ser útiles debido a su complejidad y se tornan complejos, un mapa de 6 variable es lo complicado que puedes trabajar, aunque se puede con más no es común verlos. La numeración de las casillas es:

EF				EF					
	00	01	11	10	00	01	11	10	
CD	00	0	1	3	2	16	17	19	18
	01	4	5	7	6	20	21	23	22
	11	12	13	15	14	28	29	31	30
	10	8	9	11	10	24	25	27	26
AB= 00				AB= 01					
CD	00	32	33	35	34	48	49	51	50
	01	36	37	39	38	52	53	55	54
	11	44	45	47	46	60	61	63	62
	10	40	41	43	42	56	57	59	58
AB= 10				AB= 11					



El MSB es A y el LSB es F, y las casillas como se dijo es la representación decimal de los bits en binario, por ejemplo: la casilla 56 ocupa la casilla 111000(ABCDEF) que si lo pasas a decimal es 56.

2.6 Parámetros TTL y CMOS

2.6.1 Tensión de alimentación continua

TTL En continua(DC), para los *Transistor-Transistor Logic* la alimentación es de +5V.

CMOS En DC, para los *Complementary Metal Oxide Semiconductor* existen en +5V, +3.3V, +2.5V y +1.2V

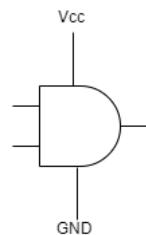


Figura 2.15: En los diagramas se omite los pines Vcc y GND.

2.6.2 Nivel lógico CMOS +5V

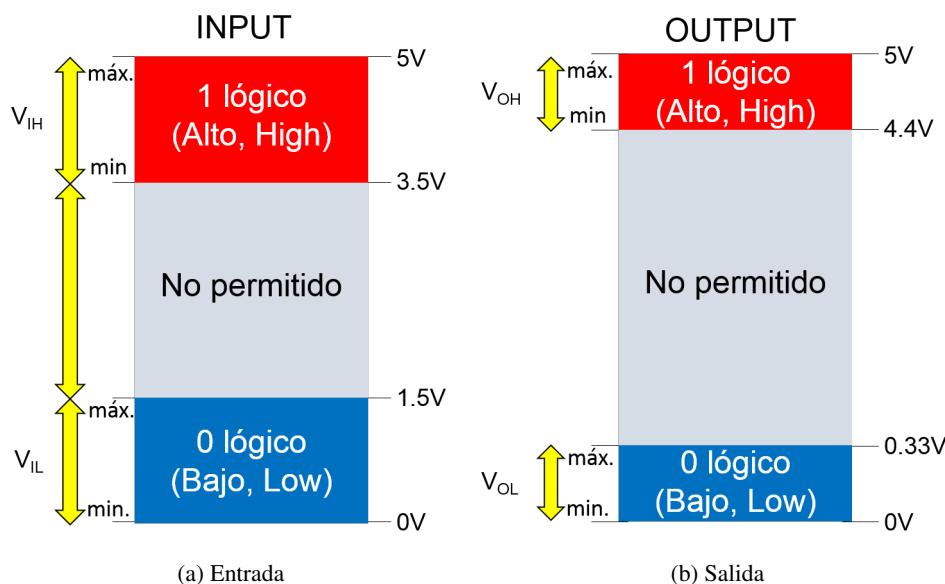


Figura 2.16: Nivel lógico CMOS a +5V.

2.6.3 Nivel lógico CMOS +3V

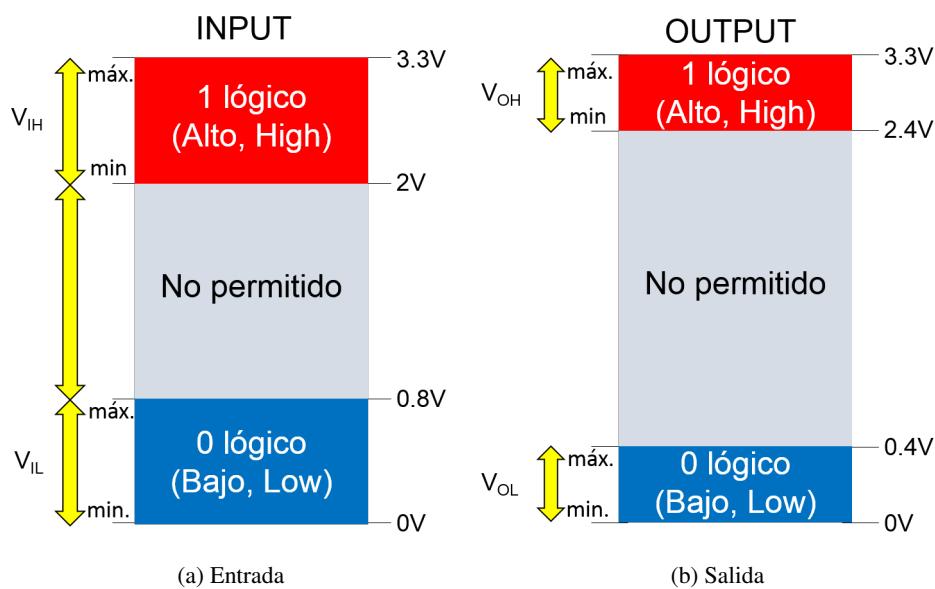


Figura 2.17: Nivel lógico CMOS a +3.3V.

2.6.4 Nivel lógico TTL

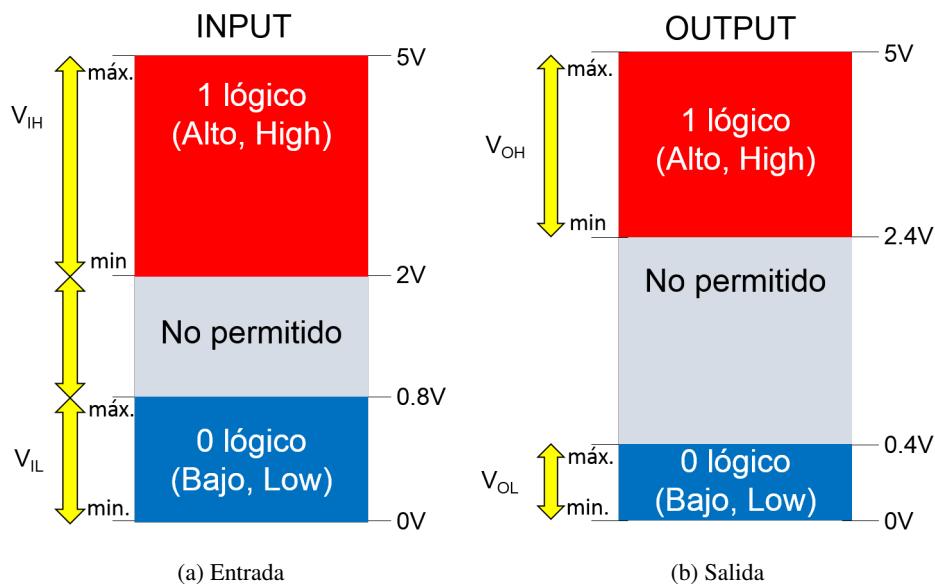


Figura 2.18: Nivel lógico TTL a +3.3V.

2.6.5 Inmunidad y Margen de ruido

Capacidad de tolerar ciertas fluctuaciones de tensión no deseadas en sus entradas sin que cambie. La medida de inmunidad de ruido es el margen de ruido. Para determinado circuito se especifican dos valores:

V_{NH} Margen de ruido alto.

V_{NL} Margen de ruido bajo.

$$V_{NH} = V_{OH(min)} - V_{IH(min)} \quad (2.24a)$$

$$V_{NL} = V_{IL(mx)} - V_{OL(mx)} \quad (2.24b)$$

2.6.6 Disipación de potencia

Cuando el estado de salida de la puerta es a nivel alto, circula la corriente I_{CCH} y cuando es a nivel bajo circula una corriente I_{CCL} .

$$I_{CC} = \frac{I_{CCH} + I_{CCL}}{2} \quad (2.25)$$

Ejercicio 2.5 Por una determinada puerta circulan $2\mu A$ cuando su salida esta a nivel Alto y $3,6\mu A$ cuando esta a nivel bajo. ¿Cuál es la disipación de potencia media si:

- La puerta esta en un estado de salida estático alto?
- La puerta esta en un estado de salida estático bajo?
- V_{cc} es 5V y la puerta funciona con un ciclo de trabajo de 25% en alto?

Solución:

1. **Puerta estática en estado alto:**

$$P = I \times V$$

$$P = 2\mu A \times 5V$$

$$P = 10\mu W$$

2. **Puerta estática en estado bajo:**

$$P = I \times V$$

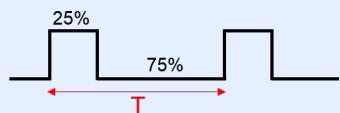
$$P = 3,6\mu A \times 10V$$

$$P = 18\mu W$$

La potencia media es:

$$P_m = \frac{10\mu W + 18\mu W}{2} = 14\mu W$$

3. **V_{CC} es 5 V y nivel alto a 25%:** 25% duty cycle es otra forma de expresar, esta forma es usada en PWM.



$$P = 25\%(10\mu W) + 75\%(14\mu W)$$

$$P = 2,5\mu W + 13,5\mu W$$

$$P = 16W$$

La disposición en un circuito TTL es esencialmente constante dentro de su rango de frecuencias de operación. Para los CMOS depende de la frecuencia.

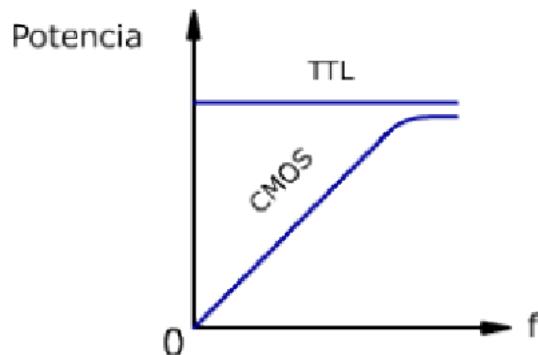


Figura 2.19: Curva disipación de potencia.

2.6.7 Carga y Fan-out

Cuando la salida de una compuerta es la entrada de otra, esto se convierte en carga. Fan-out es el límite de cuantas puertas se puede colocar.

3. Circuitos Lógicos

Existen dos tipos de circuitos lógicos:

Combinacional Donde la salida solo depende de la entrada actual.

Secuencial Donde la salida depende de la entrada actual y la secuencia anterior.

3.1 Sumador

3.1.1 Semi-sumador

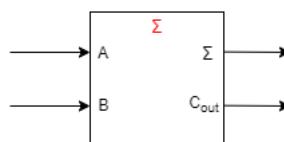


Figura 3.1: Semi-sumador

A y B Bits de entrada.

Σ Suma.

C_{out} Acarreo de salida.

Su tabla de verdad es la siguiente:

A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Cuadro 3.1: Tabla de verdad: semi-sumador

A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Cuadro 3.2: Tabla de verdad: Sumador completo.

Las dos entradas dos A y B, las suma en binario de ambas entradas es la salida Σ y el acarreo¹(lo que comúnmente llamamos: "lo que llevamos") es la salida C_{out} . Ese circuito solo depende de las entradas actuales por si mismo. Para implementar este circuito, se tiene que implementar el siguiente circuito lógico:

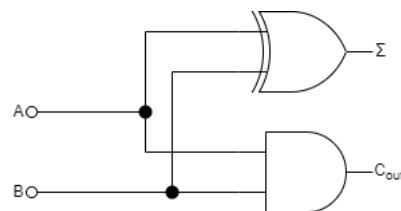
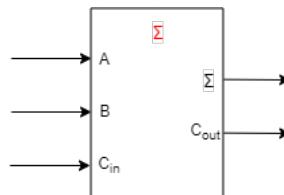


Figura 3.2: Circuito lógico semi sumador

3.1.2 Sumador completo

Figura 3.3: Sumador completo
Donde:

A y B Bits de entrada.

C_{in} Acarreo de entrada.

Σ Suma.

C_{out} Acarreo de salida.

La tabla de verdad del sumador completo es: Simplificando por mapas de Karnaugh, se obtiene el siguiente circuito lógico:

¹Por ejemplo, $5+3=8$, no existe acarreo. Pero si sumamos $8+9$, sería 4 con acarreo de 1, en total 14. El acarreo depende del sistema de numeración. en base 10, el valor máximo es 9 ya que 10 será 0 con acarreo de 1; en base 2 ocurre lo mismo lo máximo será 10(2 en base 10).

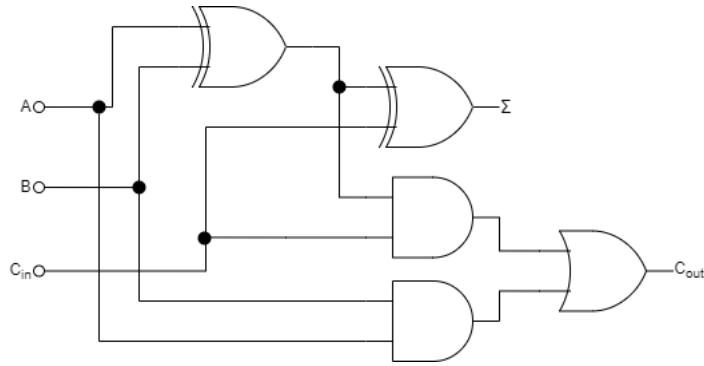


Figura 3.4: Circuito lógico sumador completo

3.1.3 Sumador paralelo

Si implementamos los sumadores completo en cascada se obtiene el sumador en paralelo, que tiene la capacidad de sumar números con más dígitos.

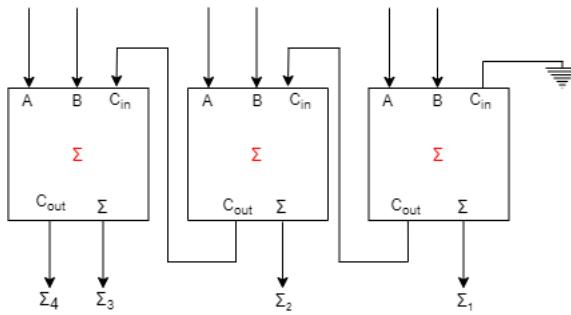


Figura 3.5: Sumador paralelo

3.1.4 Sumador 74LS283

Este es un circuito que sirve como sumador de dos números de 4 bits, se puede encadenar más circuitos en forma de cascada para expandir el tamaño. Para poder usar este sumador se debe seguir los siguientes pasos:

1. **Identificar los números:** En nuestro ejemplo, usaremos B=22 y A=9.
2. **Pasar números a binario:** Si se desea la suma, solo se pasa a binario. Si se desea una resta, por ejemplo 22+(-9), en necesario pasar a binario ambos números y el número negativo se tiene que negar usando complemento a 1.
3. **Colocar los números en las entradas:** Para la suma: B=22=0001 0110 y A=9=0000 1001 en caso de la suma, en caso de resta: B=22=0001 0110 y A=-9=1111 0110. En este caso el tamaño de cada número es 1 byte, y están agrupados en grupos de 4 bits. Colocamos en las entradas en grupos de 4 bits empezando con el MSB.

Para la suma:

- **B(1-4):** Colocamos 0001.
- **A(1-4):** Colocamos 0000.
- **B(5-8):** Colocamos 0110.
- **A(5-8):** Colocamos 1001.
- **Interruptor Suma/Resta:** Este es un interruptor que commuta entre una suma y resta, como sumaremos 22+9 lo dejamos en 0.

Para la resta:

- **B(1-4):** Colocamos 0001.
- **A(1-4):** Colocamos 1111.
- **B(5-8):** Colocamos 0110.
- **A(5-8):** Colocamos 0110.
- **Interruptor Suma/Resta:** Al tratarse de una resta $22+(-9)$ lo dejamos en 1.

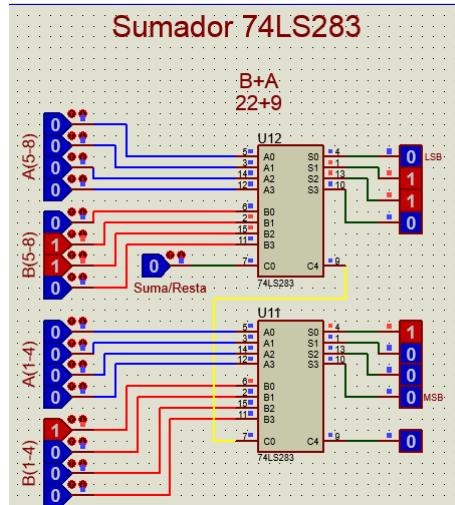


Figura 3.6: Sumador 74LS283 en proteus.

3.1.5 Sumador Acarreo Anticipado

La propagación de acarreo tiene lugar cuando el acarreo de entrada se transmite como acarreo de salida. El acarreo de entrada puede ser propagado por el sumador completo cuando uno o ambos bits de entrada son igual a 1. El acarreo propagado(C_p) se expresa como la función OR de los bits de entrada.

$$C_p = A + B$$

La generación tiene lugar cuando el sumador completo genera internamente un acarreo de salida. Solo cuando ambos bits de entrada son 1 se genera internamente un acarreo de salida. Solo cuando ambos bits de entrada son 1 se genera un acarreo, el acarreo acelerado C_g se expresa como la función AND de los bits de entrada A y B.

$$C_g = A \cdot B$$

3.2 Codificador y Decodificador

Es el proceso de asignar a cada entrada una combinación única de bits.

3.2.1 Encoder

Son circuitos combinacionales con 2^n entradas máximas y n salidas, en donde las filas de las entradas van a tener **solo un dato que cambia** y en la salida aparece un código asignado a esas entradas.

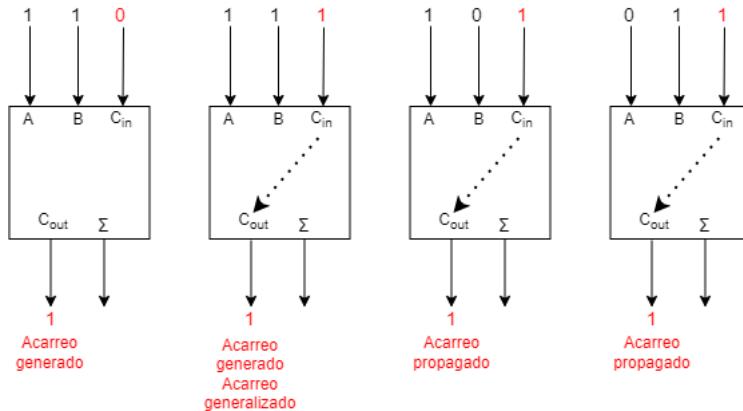


Figura 3.7: Sumador anticipado.

Procedimiento

- Determine cual es el único valor que cambia en la entrada.
 - Las salidas se leen por columnas tomando cada dato como un término completo.
- Caso 1: En cada fila hay un dato que cambia.** En un teclado se utilizan teclas del 0 al 3 y se requiere que en las salidas entregue los números codificados en binario en las entradas. La

IN				OUT	
T ₀	T ₁	T ₂	T ₃	B ₀	B ₁
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Cuadro 3.3: Tabla de verdad codificador 3 botones.

realización de esta tabla cumple con: $2^2 = 4$ entradas y 2 salidas. El dato que cambia es la entrada 1, si lo escribimos como POS y SOP:

$$B_1 = T_1 + T_3$$

$$B_2 = T_2 + T_3$$

$$B_1 = \overline{T_0} \cdot \overline{T_2}$$

$$B_2 = \overline{T_0} \cdot \overline{T_1}$$

- Caso 2: Hay una fila donde ningún dato cambia.**

IN			OUT	
D ₀	D ₁	D ₂	B	A
0	1	1	0	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Cuadro 3.4: Codificador: una fila no cambia.

IN			OUT	
D_0	D_1	D_2	B	A
0	1	1	0	0
1	0	1	1	1
1	1	0	1	0
0	0	0	0	1

$$A = \overline{D_1} + (D_0 \cdot D_1 \cdot D_2)$$

$$B = \overline{D_2} + (D_0 \cdot D_1 \cdot D_2)$$

$$A = D_0 \cdot D_2$$

$$B = D_0 \cdot D_1$$

- Hay una fila donde todos los datos cambian.

$$A = \overline{D_1} + (\overline{D_0} \cdot \overline{D_1} \cdot \overline{D_2})$$

$$B = \overline{D_1} + \overline{D_2}$$

$$A = D_0 \cdot D_2$$

$$B = D_0 \cdot (D_0 + D_1 + D_2)$$

Tipos:

- Sin prioridad: Solamente una entrada puede ser activada en cada instante.

IN								OUT		
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Y_2	Y_1	Y_0
0	1	1	1	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	0	0	1
1	1	0	1	1	1	1	1	0	1	0
1	1	1	0	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1

$$Y_0 = \overline{A_1} + \overline{A_3} + \overline{A_5} + \overline{A_7} \therefore Y_0 = \overline{A_1 \times A_3 \times A_5 \times A_7}$$

$$Y_1 = \overline{A_2} + \overline{A_3} + \overline{A_6} + \overline{A_7} \therefore Y_1 = \overline{A_2 \times A_3 \times A_6 \times A_7}$$

$$Y_2 = \overline{A_4} + \overline{A_5} + \overline{A_6} + \overline{A_7} \therefore Y_0 = \overline{A_4 \times A_5 \times A_6 \times A_7}$$

Las entradas se activan con 0 lógico.

- Con prioridad: Codifica la entrada activa de mayor valor decimal son tener en cuenta los demás. Un ejemplo con prioridad es el IC74147, donde todas las entradas se activan con 0 lógico y las todas las salidas están negadas, es decir, se activan con 0 lógico.

3.2.2 Decoder

Al contrario del encoder, el decoder es lo contrario: de n entradas se puede manipular 2^n entradas. El número en binario se escribe empezando por el A1 (que es más significativo). Los

decodificadores tienen su propia tabla de la verdad que representa los estados posibles de entrada y los respectivos valores de las salidas para cada uno de esos estados.

De esta tabla de la verdad podemos sacar el circuito lógico combinacional con puertas lógicas, circuito lógico para construir nuestro decodificador.

Ejercicio 3.1 Diseñe 2 circuitos que al activarse cada uno de los pulsadores en la salida se visualicen los siguientes números en binario:

1. Pulsador 1: 14
2. Pulsador 2: 50
3. Pulsador 3: 23
4. Pulsador 4: 72

Solución:

Primero, vamos a separar decenas y unidades de cada número y diseñar una tabla de verdad para cada uno. Solo poseemos 4 pulsadores, por lo tanto, vamos a necesitar 3 salidas puesto que los números de las decenas (1, 5, 2, 7) expresadas en binario necesitan 3 dígitos; entonces para $n=3$ salidas nos podemos permitir $2^3 = 8$ entradas, pero solo haremos uso de 4 entradas. Nuestro circuito constará de dos sub-circuitos: uno se encargará de mostrar los dígitos de las decenas y el otro mostrará los dígitos de las unidades. Desarrollamos la tabla de verdad para las decenas.

INPUT				OUTPUT				
P_1	P_2	P_3	P_4	D_2	D_1	D_0	N_0	
1	0	0	0	0	0	1	1	
0	1	0	0	1	0	1	5	
0	0	1	0	0	1	0	2	
0	0	0	1	1	1	1	7	

Una vez que hayamos desarrollado nuestra tabla de verdad las simplificamos usando POS o SOP, es preciso decir que ambos son posibles pero como hablamos de **SIMPLIFICAR** usaremos el que menos términos genere, en otras palabras: la salida D_0 posee 3 salidas en 1(SOP), esa misma salida posee 1 salida en 0(POS), entonces nos combiene POS pues solo será un término comprado a los 3 que nos generaría la otra forma. Aplicando POS a la tabla de las decenas:

$$D_0 = \overline{P_3}$$

$$D_1 = \overline{P_1} \times \overline{P_2}$$

$$D_2 = \overline{P_1} \times \overline{P_3}$$

Lo mismo haremos con las unidades, sin embargo en esta tabla notamos que nos conviene expresarlo como SOP pues en cada salida hay menos salidas 1 que salidas 0, por lo tanto, simplificando mediante SOP la tabla de las unidades:

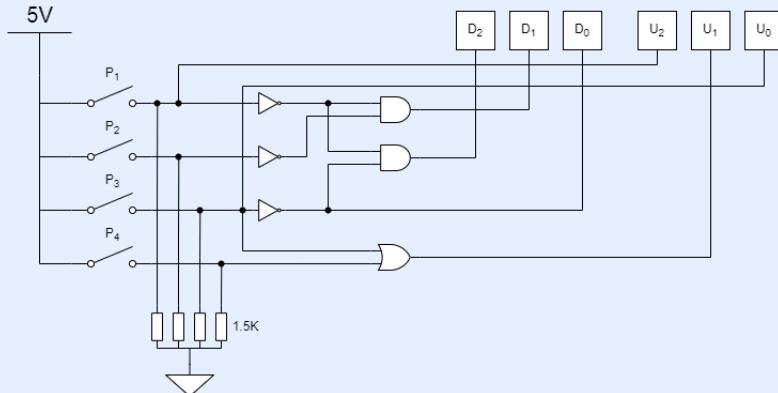
INPUT				OUTPUT				
P_1	P_2	P_3	P_4	D_2	D_1	D_0	N_0	
1	0	0	0	1	0	0	4	
0	1	0	0	0	0	0	0	
0	0	1	0	0	1	1	3	
0	0	0	1	0	1	0	2	

$$U_0 = P_3$$

$$U_1 = P_3 + P_4$$

$$U_2 = P_1$$

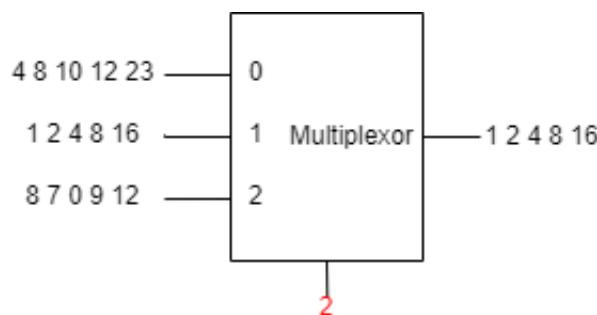
Ya tenemos las expresiones, antes de realizar el circuito esta de más aclarar que el bit más significativo es en el caso de las decenas D_2 y en el caso de las unidades es U_2 . Tenemos las expresiones booleanas de cada tabla así que solo queda realizar su circuito:



3.3 Multiplexor y Demultiplexor

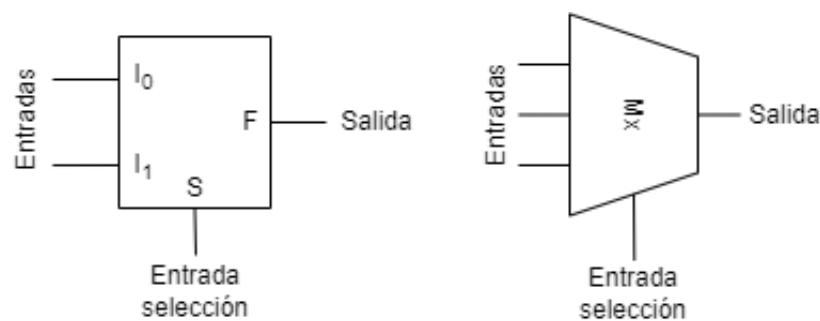
3.3.1 Multiplexor

Circuito combinacional al que entran varios canales y solo sale uno de ellos.



Multiplexor Simple

Existen dos posibles simbologías para los multiplexores:



S	I_1	I_0	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

		$I_1 I_0$				
		00	01	11	10	
S		0	0	1	1	0
		1	0	0	1	1

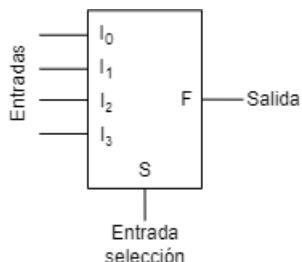
$$F = \bar{S} \cdot I_0 + S \cdot I_1$$



Es usual que en las tablas de verdad, la entrada que se encuentra a la izquierda es el más significativo(MSB) mientras que el que se encuentra más a la derecha es el menos significativo(LSB).

Si analizamos la tabla: vemos que la entrada de selección S, poseerá dos estados; si analizamos cuando S=0 vemos que la salida F estará activada siempre y cuando la entrada I_0 tenga la entrada activa, mientras que si I_0 posee una entrada 0, no importa el valor que posea la entrada I_1 (alto o bajo) la salida F siempre será 0. Esto nos quiere decir que cuando S=0, la salida F solo le importará los estados de la entrada I_0 pero no de I_1 . Cuando la entrada de selección S=1, notaremos le mismo comportamiento, la salida F solo tomará en cuenta los estados de la entrada I_1 puesto que la entrada I_0 no nos importa. En conclusión, el la entrada de control S es capaz de seleccionar entre dos entradas.

Multiplexor 2 entradas selección

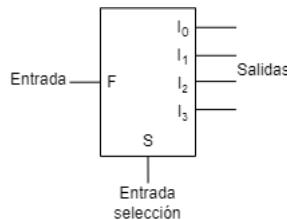


S_1	S_0	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

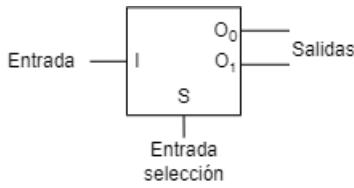
$$F = \bar{S}_1 \cdot \bar{S}_0 \cdot I_0 + \bar{S}_1 \cdot S_0 \cdot I_1 + S_1 \cdot \bar{S}_0 \cdot I_2 + S_1 \cdot S_0 \cdot I_3$$

3.3.2 Demultiplexor

Al igual que el multiplexor puede poseer varias entradas, varios canales a diferentes bits.

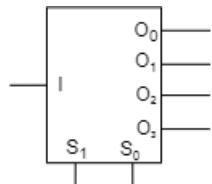


La expresión booleana para el multiplexor más simple:



S	I	O ₁	O ₀
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

$$F = \overline{S_1} \cdot \overline{S_0} \cdot I_0 + \overline{S_1} \cdot S_0 \cdot I_1 + S_1 \cdot \overline{S_0} \cdot I_2 + S_1 \cdot S_0 \cdot I_3$$



S ₁	S ₀	O ₃	O ₂	O ₁	O ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

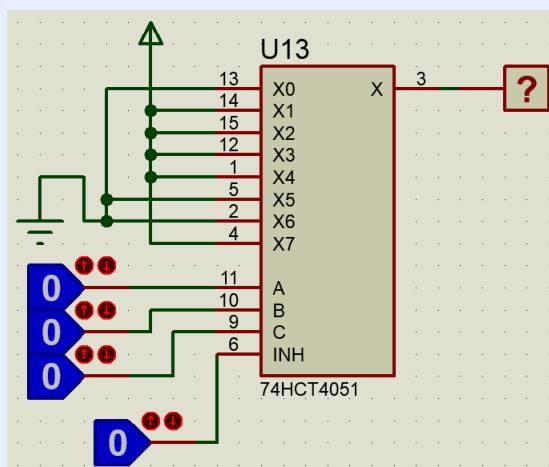
$$O_0 = \overline{S_1} \cdot \overline{S_0} \cdot I \quad O_1 = \overline{S_1} \cdot S_0 \cdot I \\ O_2 = S_1 \cdot \overline{S_0} \cdot I \quad O_3 = S_1 \cdot S_0 \cdot I$$

Ejercicio 3.2 Usando el IC 74HCT4051, expresar la siguiente expresión de un demultiplexor:

$$F = C\bar{B}\bar{A} + \bar{C}B\bar{A} + \bar{C}BA + CBA + \bar{C}\bar{B}A$$

Solución:

Para poder implementarlo con el IC, necesitamos que la expresión este en su forma canónica. Una vez que este en su forma canónica, dependiendo si esta como SOP o POS nos indicará los valores donde será 1 y 0 respectivamente. Como estamos frente a un SOP, los valores decimales de los sumandos(pasando a binario los bits, donde MSB es C y LSB es A) nos dan los números de las entradas en las cuales deben tener entradas altas y las restantes entrada baja.



Cada vez que en las entradas de selección se logre colocar cualquiera de los sumandos de la función F la salida será 1. ■

3.4 Latch

El latch(cerrojo) es un tipo de dispositivo de almacenamiento temporal de dos estados(biestable).

3.4.1 Latch S-R(Set-Reset)

Dispositivo lógico biestable o multivibrador. Un Latch S-R con entrada activa a un nivel ALTO, se compone de dos puertas NOR o NAND acopladas.

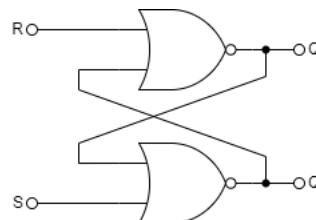


Figura 3.8: Latch S-R con entrada activa a nivel alto.

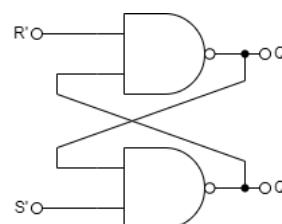


Figura 3.9: Latch $\bar{S}-\bar{R}$ con entrada activa a nivel bajo.

Comportamiento

En Q , cuando S este en nivel alto y R en estado bajo, la salida Q cambia a nivel alto(Estado inicial $Q=0$).

Una vez que nos encontramos en este estado, sea cual sea la entrada de S , la salida Q no cambiará:

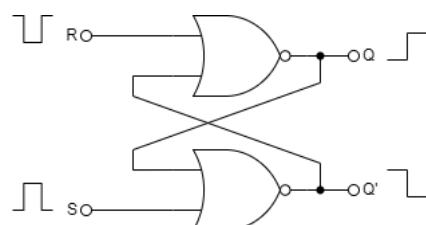


Figura 3.10: Set=1, Reset=0 y $Q=1$.

Cambiará solo si R cambia a estado alto.

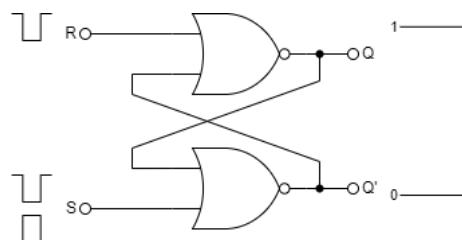


Figura 3.11: Set=1/0, Reset=0 y $Q=1$.

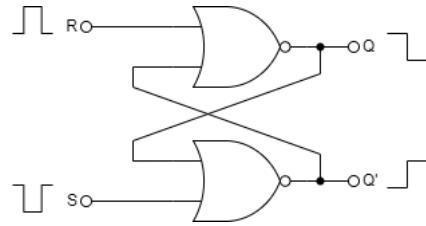


Figura 3.12: Set=0, Reset=1 y Q=0.

Condiciones especiales

Los latch poseen algunas restricciones que son importantes conocerlas para tener precaución al momento de implementar circuito.

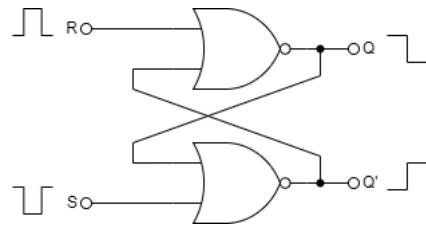


Figura 3.13: Si S=0 y R=0, las salidas no cambian.

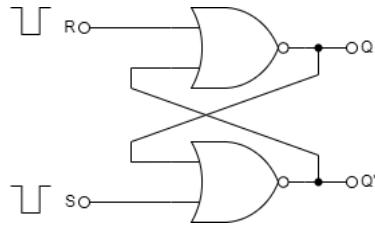


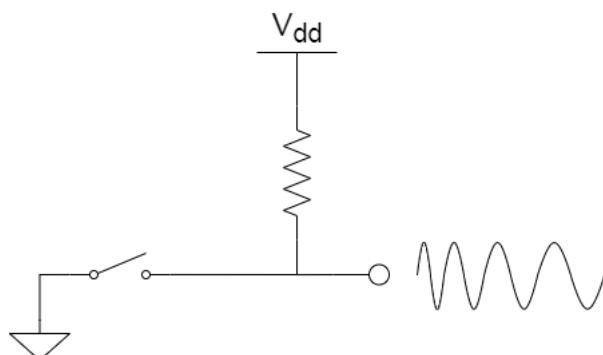
Figura 3.14: Si S=1 y R=1, las salidas presentan un estado no válido: indeterminado.

3.4.2 Símbolo lógico

La simbología del latch es el siguiente: El diagrama de tiempos de un Latch tiene el siguiente comportamiento ejemplificado:

3.4.3 Latch: Eliminador de rebote

Se presenta un circuito con interruptor:



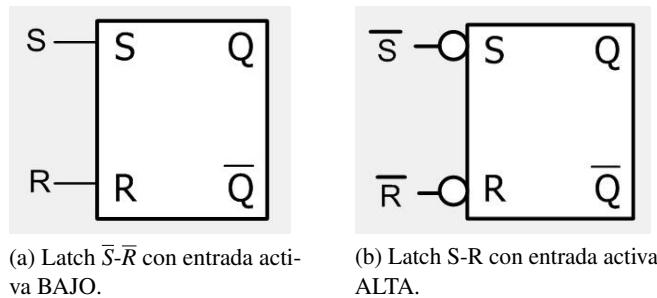


Figura 3.15: Latch simbología

Input		Output		Comentarios
\bar{S}	\bar{R}	Q	\bar{Q}	
1	1	NC	NC	No cambia, permanece en el mismo estado
0	1	1	0	Latch estado SET
1	0	0	1	Latch estado Reset
1	1	1	1	Condición NO valida

Cuadro 3.5: Tabla de verdad: Latch

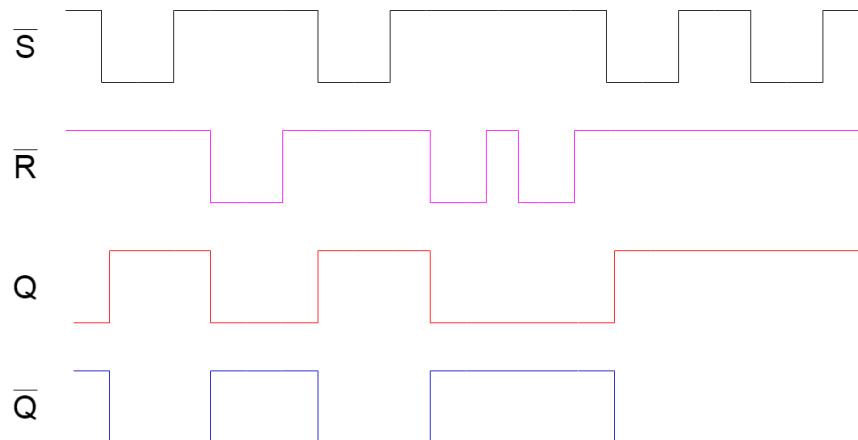
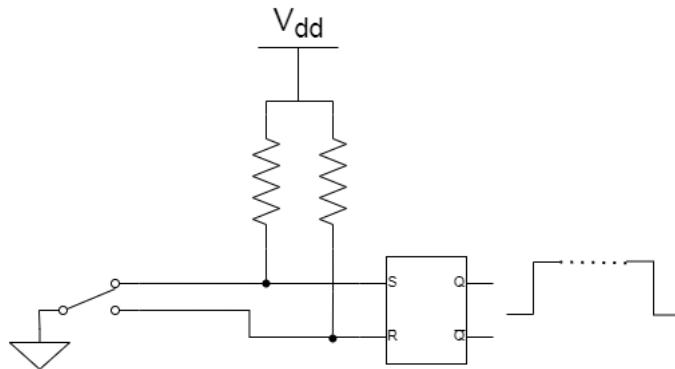


Figura 3.16: Diagrama de tiempos Latch.

Vemos que la salida no es estable, esta lectura inestable puede ocasionar errores de lectura o falsos altos y bajos. Por eso se añade un latch como eliminador de rebote:



3.4.4 Latch con entrada de habilitación

El latch no cambiará de estado hasta que la entrada EN este en nivel ALTO, mientras este en ALTO, las salidas serán controlados por las entradas S y R. Una situación invalida es cuando R y S están a nivel alto simultáneamente. Diagrama de tiempos de un Latch con entrada enable:

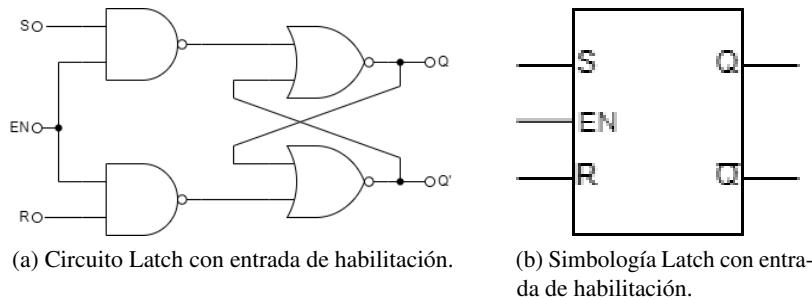


Figura 3.17: Latch con entrada de habilitación.

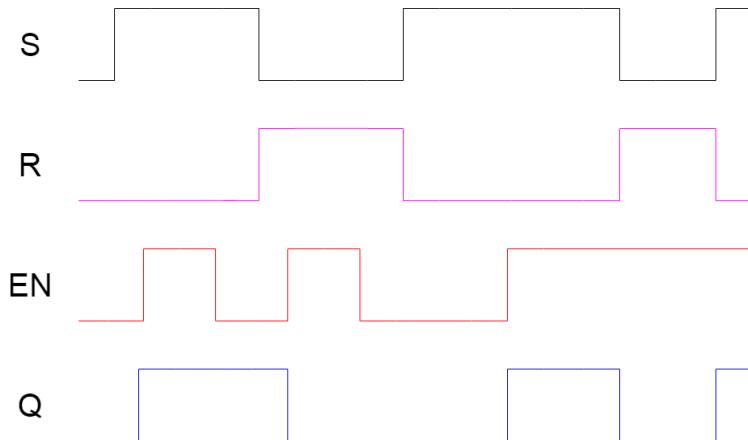


Figura 3.18: Diagrama de tiempos latch S-R con entrada de habilitación.

3.4.5 Latch D con entrada de habilitación

A diferencia de latch S-R con entrada de habilitación, el latch D solo tiene una entrada que recibe el nombre de entrada de datos(D) y ademas de la habilitación(EN). Mientras EN este en estado ALTO, la salida Q es una copia de D, si EN pasa a estado BAJO se mantiene el estado anterior.

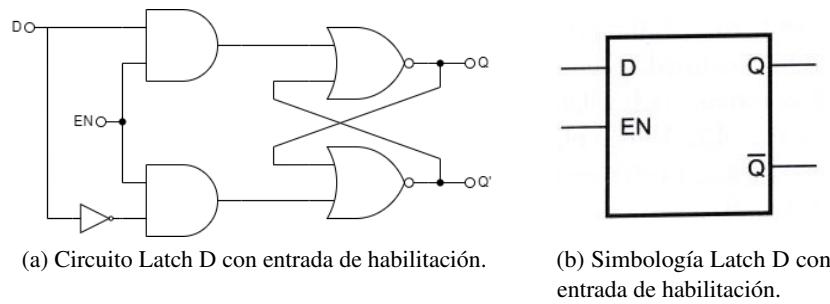


Figura 3.19: Latch D con entrada de habilitación.

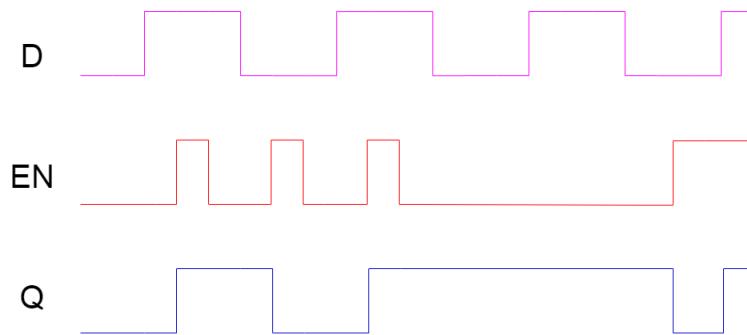


Figura 3.20: Diagrama de tiempos latch D con entrada de habilitación.

3.5 Flip-Flops

Son dispositivos síncronos de dos estados, multivibradores biestables. La salida cambia de estado únicamente en un instante específico de una entrada reloj(CLK). Los cambios se producen sincronizadamente con el reloj. Normalmente son **disparados por flanco**.

Una identificación es por si símbolo lógico lo da un triángulo que se encuentra en la entrada clock. El triángulo de denuncia indicador entrada dinámica.

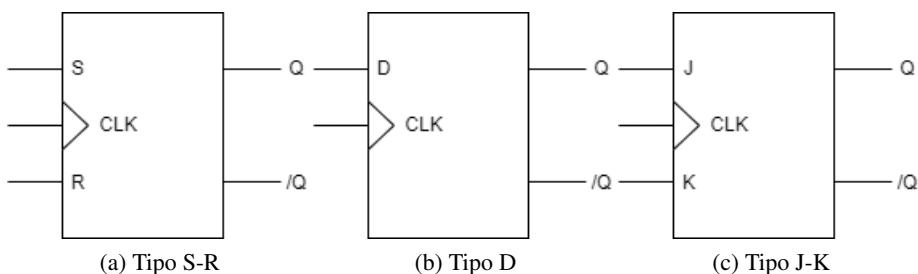


Figura 3.21: Tipos de Flip Flips

3.5.1 Flip Flop tipo S-R disparado por flanco

Los estados son iguales a los latch, lo diferente aquí es la entrada reloj y el flanco, hay de dos tipos: disparado por flanco ascendente o descendente.

R	S	D	E	Q_{SR}	$Q_{S\bar{T}e}$	Q_D
1	0	0	0	1	1	0
0	1	0	0	1	1	0
0	1	1	0	1	1	0
1	0	0	1	0	0	0
1	0	0	1	0	0	0
1	0	0	0	0	0	0
0	1	1	1	1	1	1
0	1	1	0	1	1	1
1	0	0	0	0	1	1
0	0	1	1	0	No definido	1
1	0	1	1	0	0	1
1	0	0	1	0	0	0
0	1	0	1	1	1	0
0	1	1	1	1	1	1
1	0	1	1	0	0	1
0	1	1	0	1	0	1
1	1	0	0	No definido	0	1
0	1	1	1	1	1	1
1	0	1	1	0	0	1
1	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	1	1	0
0	1	0	1	1	1	0
1	0	1	1	0	0	1
1	0	0	0	0	0	1
1	0	1	1	0	0	1

Cuadro 3.6: Ejemplo de tren de bits de latch S-R, latch S-R con enable y Latch D con enable.

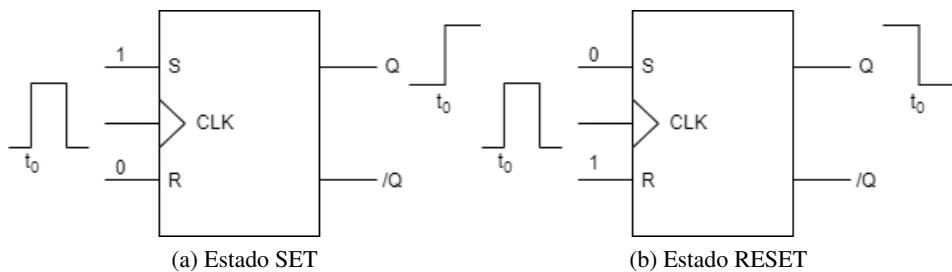
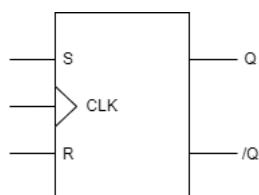


Figura 3.22: Estados



INPUT			OUTPUT		
S	R	CLK	Q	\bar{Q}	Comentarios
0	0	X	Q_0	\bar{Q}_0	No cambia
0	1	\uparrow	0	1	RESET
1	0	\uparrow	1	0	SET
1	1	\uparrow	?	?	No valida

- \uparrow : Transición del reloj de nivel bajo.
- X: Irrelevante.
- Q_0 : Nivel de salida previo a la transición del reloj.

Para el flanco descendente positivo es lo mismo solo que $CLK=\downarrow$.

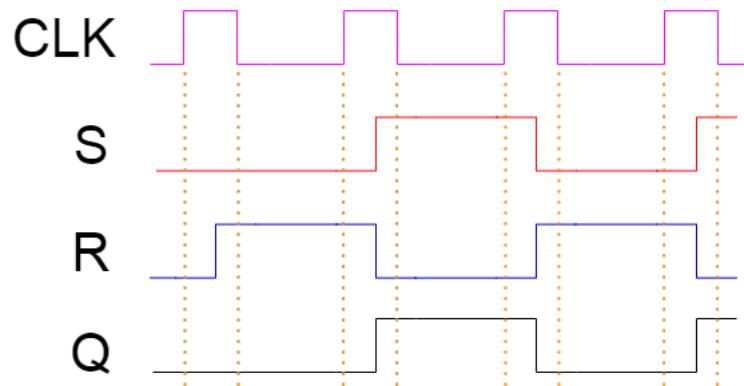


Figura 3.23: Diagrama de tiempos flip flop S-R disparado por flanco descendente.

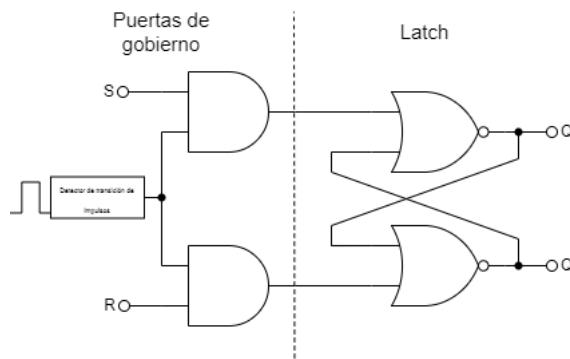


Figura 3.24: Composición de un Flip-Flop

3.5.2 Flip Flop D disparado por flanco

Resulta muy útil cuando se necesita almacenar un único bit de dato.

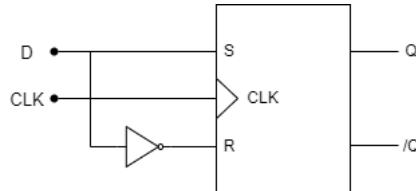


Figura 3.25: Símbolo Flip Flop D disparado por flanco.

INPUT		OUTPUT		
D	CLK	Q	\bar{Q}	Comentario
1	\uparrow	1	0	SET(almacena 1)
0	\uparrow	0	1	RESET(almacena 0)

Cuadro 3.7: Tabla de verdad Flip Flop tipo D disparado por flanco

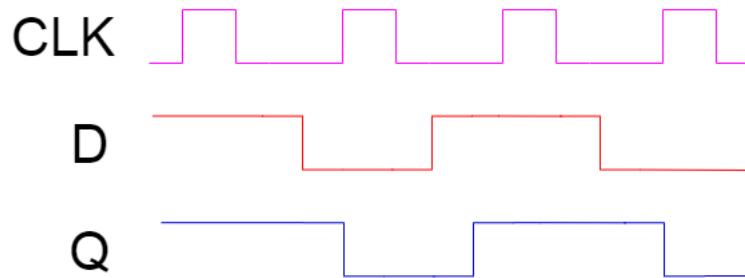


Figura 3.26: Diagrama de tiempos

3.5.3 Flip Flop J-K disparado por flanco

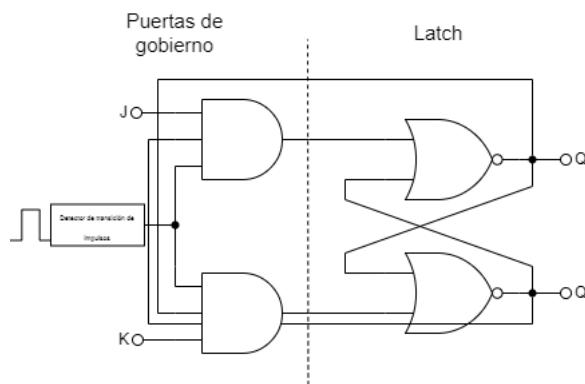


Figura 3.27: Símbolo Flip Flop J-K disparado por flanco.

- R** De acuerdo con la tabla de verdad, cuando las entradas J y K están a nivel lógico 1, a cada borde activo en la entrada de reloj, la salida del biestable cambia de estado. A este modo de funcionamiento se le denomina modo de basculación (toggle en inglés).

J	K	CLK	Q	\bar{Q}	Comentario
0	0	↑	Q_0	\bar{Q}_0	No cambio
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	Q_0	\bar{Q}_0	Basculación

Cuadro 3.8: Tabla de verdad Flip Flop J-K disparado por flanco

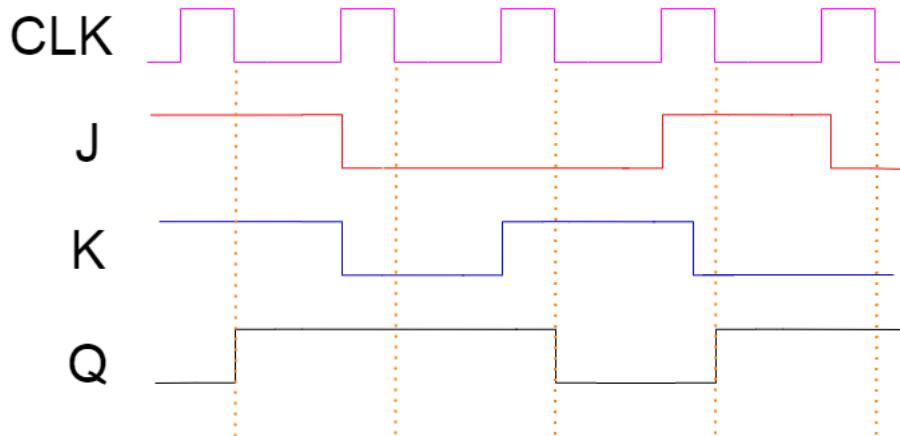


Figura 3.28: Ejemplo: flanco descendente

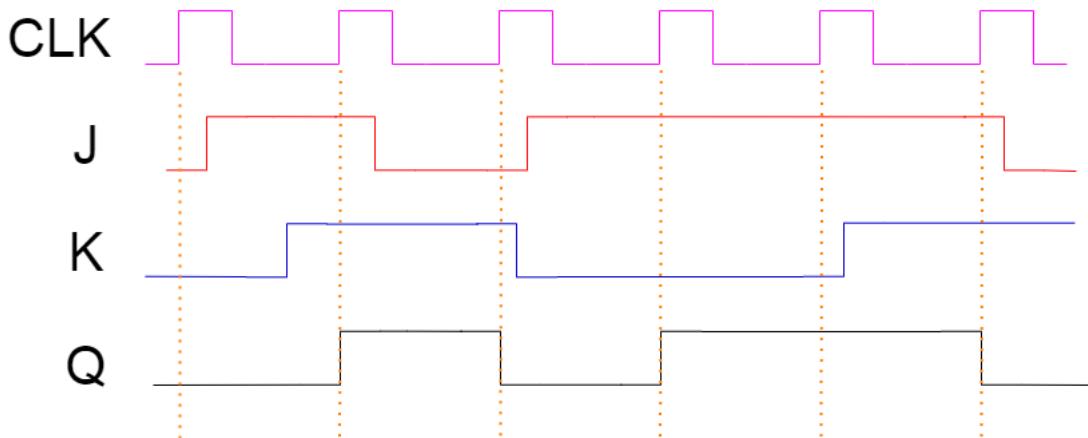


Figura 3.29: Ejemplo: flanco ascendente

3.5.4 Entradas asíncronas de inicialización y borrado

La mayoría de los circuitos integrados tienen también otras entradas asíncronas. Generalmente son *Preset* (PRE), *clear* (CLC), activación directa (SD) y desactivación directa (RD).

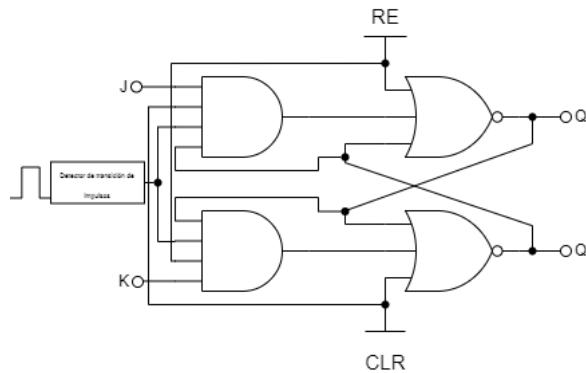


Figura 3.30: Implementación de entradas asíncronas.

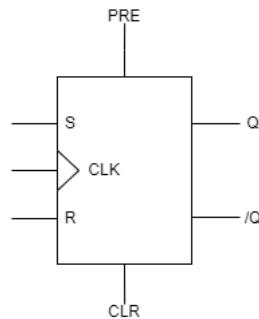


Figura 3.31: Simbología de las entradas PREset y CLeaR.

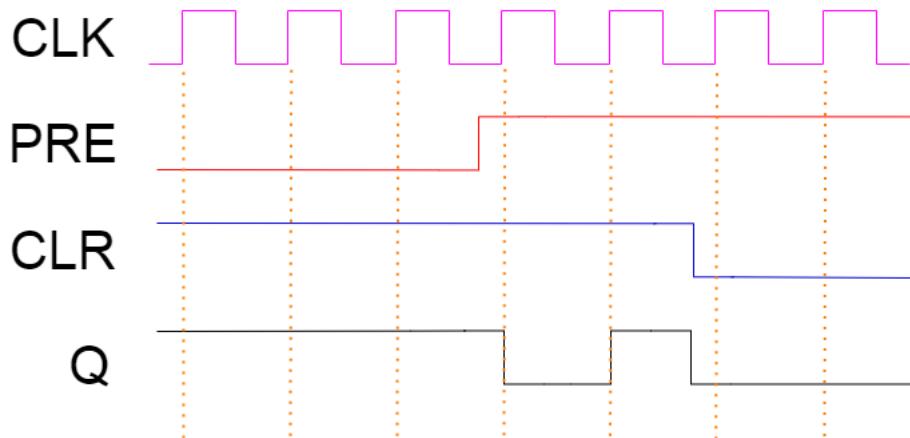
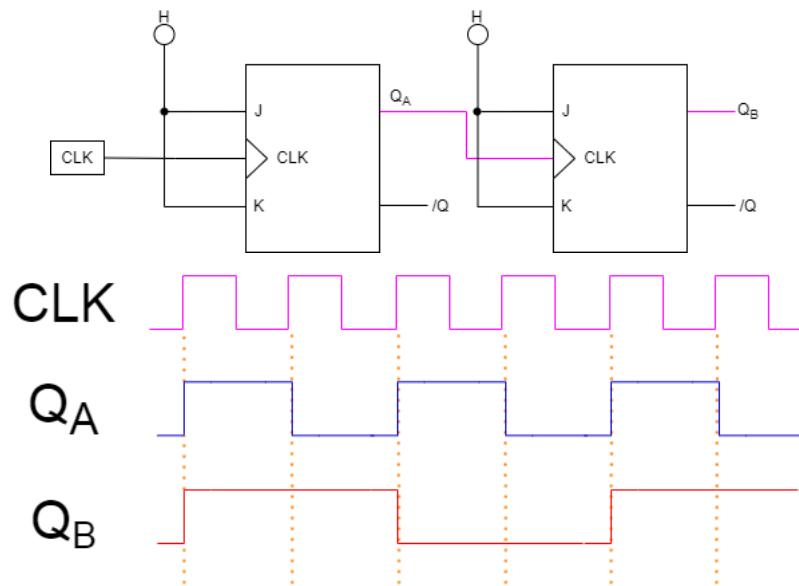


Figura 3.32: Diagrama de tiempos de las entradas PRE y CLR.

3.6 División de frecuencia

Usando Flip-Flops J-K se consigue dividir la frecuencia de reloj por 4.



3.7 Monoestables

Dispositivos multivibradores que solo tienen un único estado estable. Normalmente, un monoestable se encuentra en su estado estable, cambiando a su estado inestable solo cuando se dispara. Una vez que se ha disparado, el monoestable permanece en su estado inestable durante un determinado intervalo de tiempo, volviendo a continuación a su estado estable.

El tiempo que este dispositivo permanece en el estado inestable determina la anchura del impulso de salida. Nótese que los monoestables tienen que esperar hasta que se termine el tiempo definido, mientras que los monoestables disparables al ser activado antes que termine el tiempo definido se vuelve a inicializar el tiempo.



(a) Periodo shoot menor que Q.



(b) Periodo shoot mayor que Q.

Figura 3.33: Monoestable no redisparable

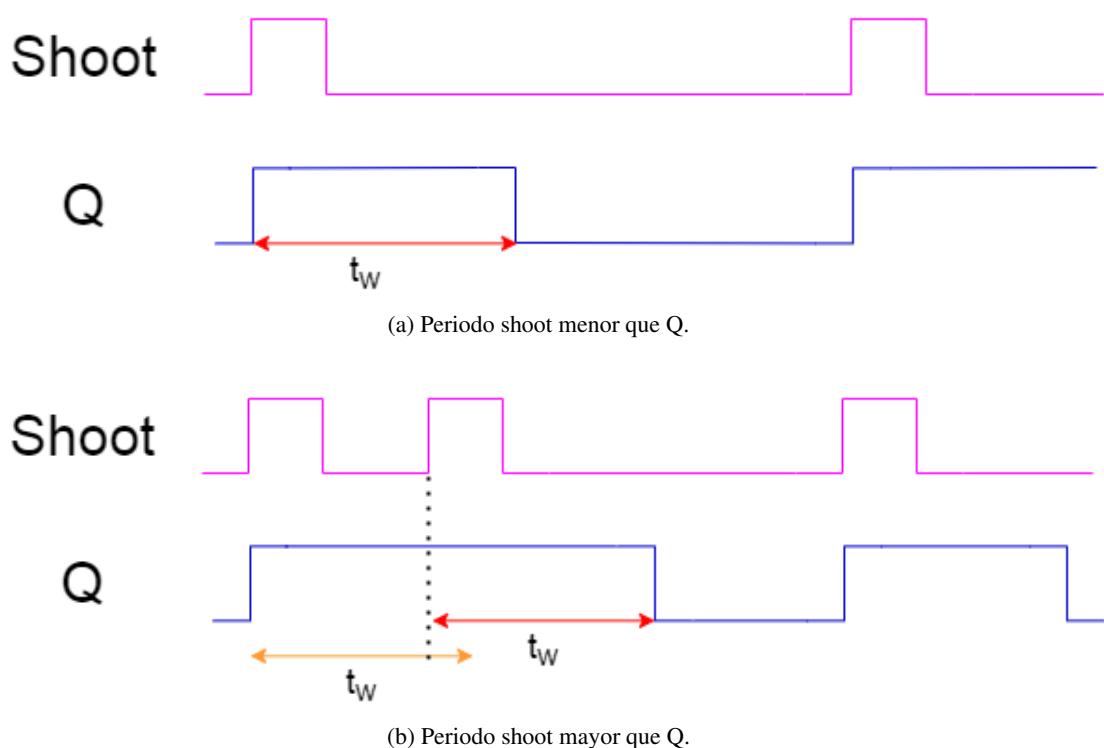


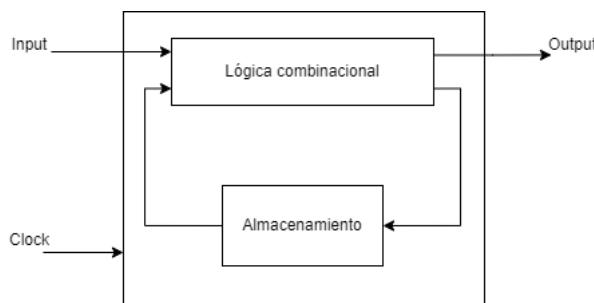
Figura 3.34: Monoestable redisparable

4. Circuitos Lógicos Secuenciales

Circuito Lógico Combinacional cuya salida depende de los valores actuales y pasados de las señales de entrada.

Componente:

- Señal de entrada y salida(binaria).
- Señal de reloj (binaria periodica).
- Lógica combinacional(determina la salida y el próximo estado)
- Almacenamiento (mantiene información sobre el estado actual)



4.1 Máquina de estados

Solo hablaremos de los Circuitos Secuenciales Síncronos.

Red de combinacionales y biestables conectados entre si. Puede haber caminos cíclicos, pero tienen que atravesar al menos un biestable. Todos los biestables usan la misma señal reloj. Las señales de entrada se sincronizan con el mismo reloj.

Tiempo del reloj debe ser mayor o igual al tiempo de propagación del flip flop más el tiempo de propagación del CLC¹(ecuación 4.1).

$$T_c \geq T_p(FF) + T_p(CLC) \quad (4.1)$$

¹Abreviatura de Circuito Lógico Combinacional

Existen dos tipos de máquinas de estados: **Mealy** y **Moore**. Son parecidas pero no iguales. Veremos la de Mealy puesto que la de Moore será más fácil. Notesé la diferencia entre ambas, la única

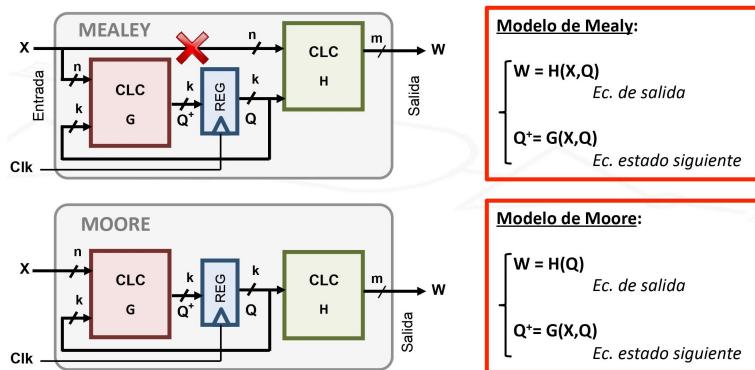


Figura 4.1: Tipos de máquina de estado

diferencia es la entrada, en Moore solo se dirige a CLC G mientras que en el de Mealy de dirige tanto al CLC G como al CLC H.

4.1.1 Modelo de Mealy

Cualquier CLS² puede expresarse como: Agrupando todos los circuitos en un único CLC(en la figura 4.1, solo contariamos con REG y CLC H) y los biestables(flip flop) en único REG. Es posible que existan más CLC(en la figura 4.1 se presenta dos CLC,G y H).

Datos que queremos caracterizar

- Número de entradas: $X=(x_{n-1}, x_{n-2}, \dots, x_1, x_0)$
- Número de salidas $W=(W_{n-1}, W_{n-2}, \dots, W_1, W_0)$
- Número de estados $Q=(q_{n-1}, q_{n-2}, \dots, q_1, q_0)$

Por eso necesitamos caracterizar:

- Tabla de verdad de W .
- Tabla de verdad de Q^+ .
- Estado inicial de Q

Puede ser un poco complicado de entender, vamos a hacer un ejercicio para entender mejor.

Ejercicio 4.1 Con:

- $X=(X_0)$
- $W=(W_1, W_0)$
- $Q=(q_1, q_0)$
- Estado inicial= $(0,0)$

Antes de todo, analicemos: Tenemos X que representa las entradas, W que representa las salidas, Q es el número de estados y el estado inicial, que representa el estado en el que el circuito siempre va empezar. Primero, mirando la forma de un CLC de Mealy (4.1), poseemos dos CLC: G y H. Primero empecemos con G: Este CLC G es un circuito de transición, puesto que no es la salida, sino que las salidas se almacenan para que el siguiente CLC funcione. ¿Como armamos la tabla de verdad?

Primero, como siempre identificamos las entradas: Tenemos X como entrada y Q el número de estados, podemos concluir que nuestras tablas deben tener 3 entradas: q_1, q_0 y x_0 ; escribimos todas las combinaciones pero ¿Cómo sabes que salidas habrá? Las salidas serán las mismas que

²Circuito Lógico Secuencial

el número de estados, se les diferencia con un símbolo de suma encima ellos que significa que el estado ha evolucionado. La **tabla de transición** se vería de la siguiente manera:

q_1	q_0	x_0	q_1^+	q_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	0
1	1	0	x	x
1	1	1	x	x

Las x significa que puede tomar cualquier valor puesto que esos valores no nos importan. En este caso, las salidas q^+ serán puestas con esos números a manera de ejemplo pero pueden ser distintos valores, depende de tu propósito. Con esto ya hemos terminado nuestra **tabla de transición** $Q^+ G$. Antes de pasar el CLC H entendamos el CLC G:

¿Qué nos quiere decir?

Nos da como estado inicial $(0,0)$ teniendo la forma de $Q(q_1, q_0)$. Nuestro punto de partida será $(0,0)$. Ahora ese estado inicial $(0,0)$ evolucionará, cambiará a otro estado (el que queramos, ya depende de nuestro propósito), en este ejemplo el estado inicial tiene dos posibles evoluciones dependiendo si x_0 es 0 o 1. Si $x_0=0$, el estado inicial $(0,0)$ evoluciona a $(0,0)$ mientras que si $x_0=1$, el estado $(0,0)$ evoluciona a $(0,1)$. Listo, ahora recordemos que el CLC G tiene en secuencia el CLC H que lo veremos luego y tiene al REGistro; es decir, las salidas (q_1^+, q_0^+) serán las entradas (q_1, q_0) . Expliquemos este paso, si tenemos $(0,0)$ y $x_0=0$, es estado de evolución será $(0,0)$, ese mismo estado será las entradas (q_1, q_0) , y entraremos en un bucle infinito, si $x_0=1$, las entradas $(0,0)$ evolucionan a $(0,1)$, vemos que aquí podemos avanzar en la tabla, ahora salida $(0,1)$ será nuestra entrada y estaremos antes otros dos casos; ahora tenemos como estado inicial $(0,1)$, si $x_0=0$ la salida será $(1,0)$ que significa que avanzaremos en la tabla pero si $x_0=1$ no avanzaremos, nos quedaremos en ese estado infinitamente. En otras palabras, las salidas al ser las mismas que las entradas, van avanzando en su evolución, eso depende de nuestra entrada x (no siempre, depende de nuestro propósito). Si seguimos bajo esta lógica, llegaremos hasta el penúltimo estado, donde la entrada $(1,0)$ permanecerá en el mismo estado si $x_0=0$, y estará en ese bucle hasta que $x_0=1$, donde evolucionará a $(0,0)$ y se volverá a repetir el ciclo, con esto se asegurar un ciclo evolutivo cerrado. **¿Y el CLC H?**

Es posible añadir más CLC, si añadimos un CLC que vaya después del REG, las entradas del CLC H (el que viene después del REG) serán las salidas del CLC G. Podemos decir que las salidas del CLC G serán las entradas del mismo CLC G (se autoalimenta con respecto a las entradas) y a su vez serán las entradas del CLC H, pero el CLC H no tiene retroalimentación, solo depende de las entradas y los estados almacenados (salidas del CLC G que se almacenan en REG).

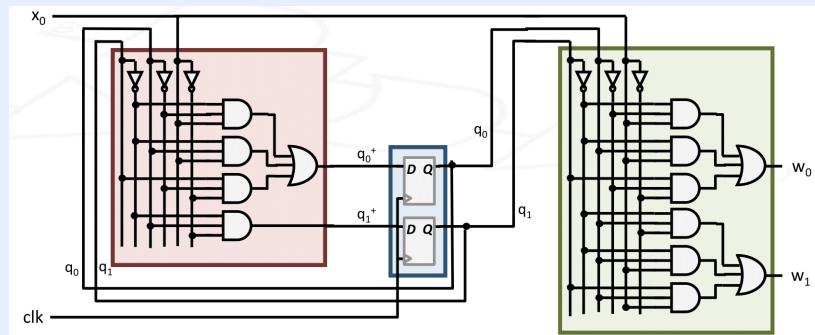
Dada la **Tabla de verdad de la salida**, en este ejemplo las definiremos de la siguiente manera^a

q_1	q_0	x_0	w_1	w_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	0

Si te fijas bien, las salidas de la tabla de transición Q^+ (CLC G) están contenidas en las entradas de la tabla de verdad de W(Tabla de verdad de la salida), y las salidas son independientes, puesto que estas serán las salidas de nuestra maquina de estado.

¿Y como armo el circuito?

Para armar el circuito se tiene que simplificar ambas tablas a SOP o POS, según nos sea mejor, una vez que hayamos logrado simplificar lo más que podamos procedemos a implementar el circuito, el circuito de nuestro ejemplo sería:



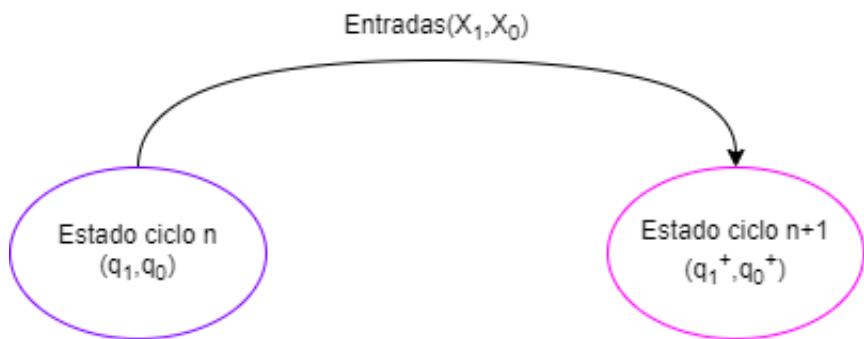
Según lo visto se puede completar la siguiente tabla sabiendo el estado inicial es (0,0):

q_0	0	0	0	1	1	0	1
q_1	0	0	0	0	0	1	0
Ciclo	n	n+1	n+2	n+3	n+4	n+5	
X_0	0	1	1	0	1	1	
w_0	0	1	1	0	0	1	
w_1	0	0	1	1	0	1	
q_0^+	0	1	1	0	1	1	
q_1^+	0	0	0	1	0	0	

^aOjo, no siempre saldrá así, ya depende de nuestro proyecto o nuestras necesidades.

4.2 Grafos de estados

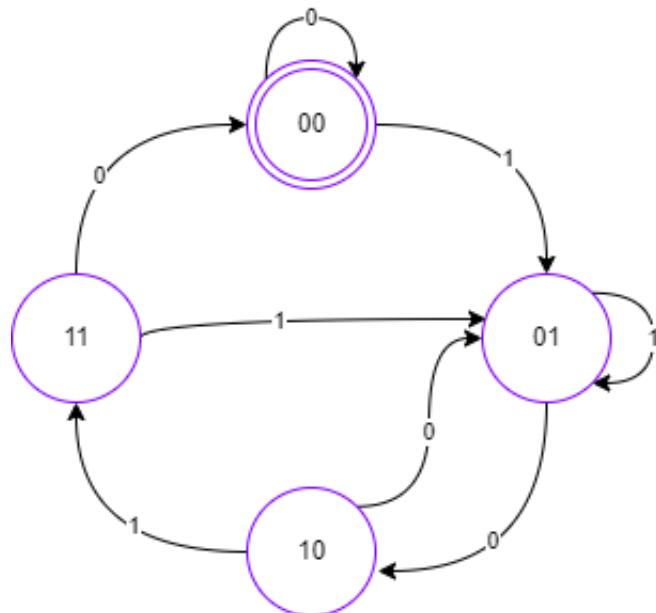
Grafos de estados o diagramas de estado son una forma de expresar las tablas de transición, el formato es el siguiente:



Dada la tabla de transición:

q_1	q_0	x_0	q_1^+	q_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

Su grafo correspondiente es:



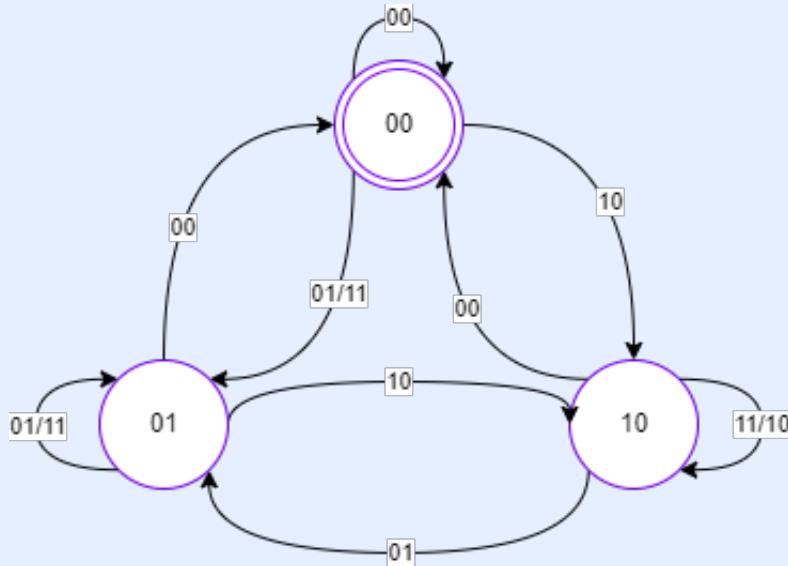
El **estado inicial** es representado con el doble círculo.

Ejercicio 4.2 Dada la tabla de transición de un CLS, dibuja su grafo de estados:

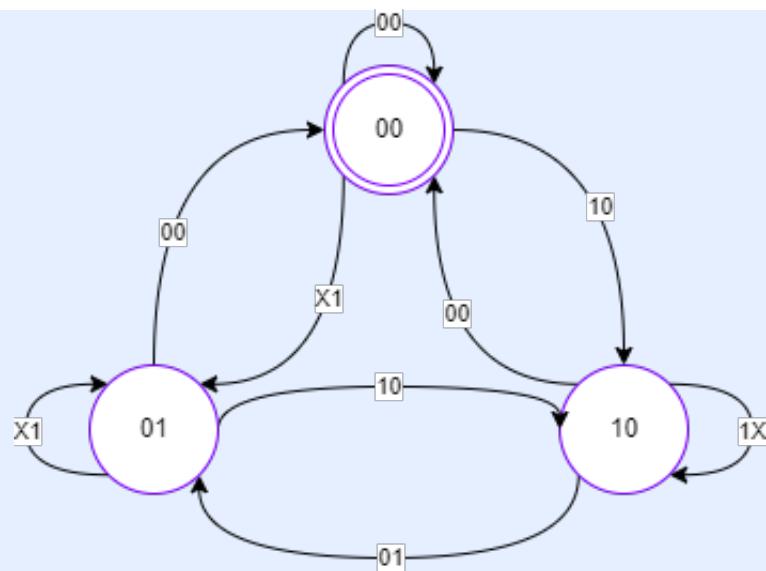
q_1	q_0	x_1	x_0	q_1^+	q_0^+
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Solución:

Nos fijamos en la salidas y en las entradas, notamos que en las salidas son: (0,0), (0,1) y (1,0) en las entradas tenemos las mismas combinaciones excepto (1,1). Como conclusión de este análisis podemos decir que la entrada (1,1) no se usa, ya que ninguna salida hace que lleguemos a este estado, por ende nuestros estados son: (0,0), (0,1) y (1,0). Una vez tengamos los estados definidos procedemos a hacer las flechas basandonos en las entradas x_1 y x_0 .



Nota que algunas flechas tiene dos estados, por ejemplo: 01/11 o 11/10. Cuando tenemos estos casos podemos simplificarlo ya que considerando la forma que tienen (x_1, x_0), no nos importa el valor de x_1 y no nos importa x_0 respectivamente en cada caso, así que podemos escribir una x indicando no importa el estado de alguna entrada, simplificando quedaría así:

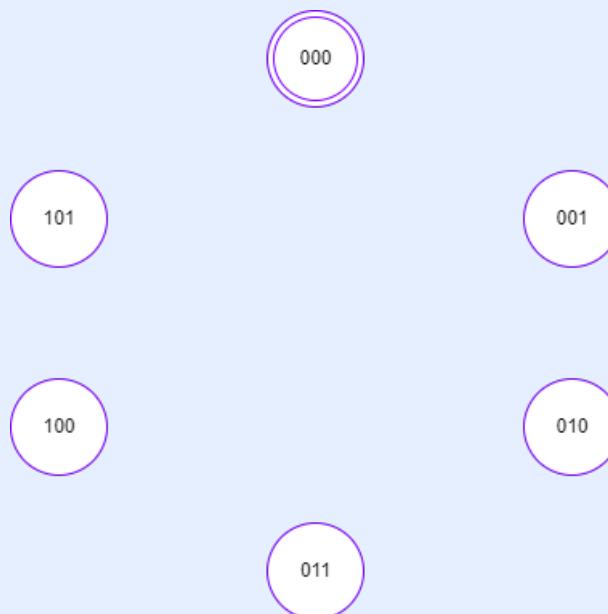


Aplicando la teoría a un ejemplo más real:

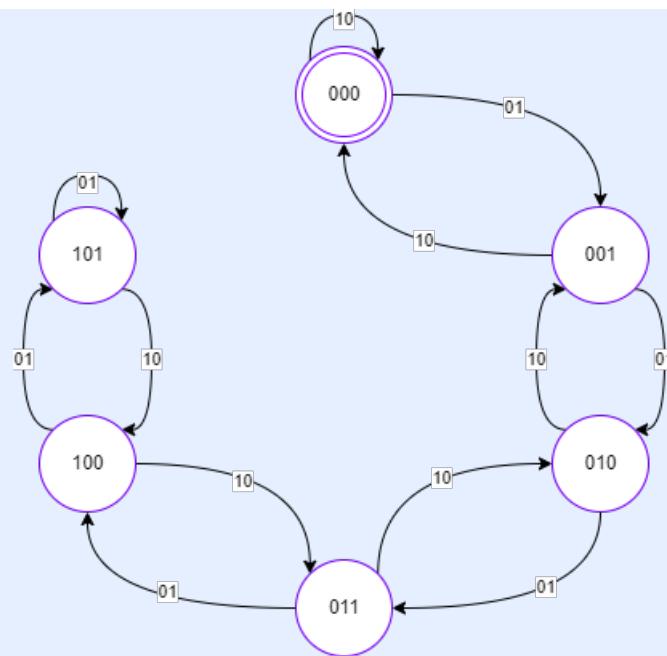
Ejercicio 4.3 Tenemos un contador desde 0 al 5, con entradas x_1 y x_0 . Hacer el grafos teniendo en cuenta que $x_1=+1$ y $x_0=-1$ con estado inicial en 0,0.

Solución:

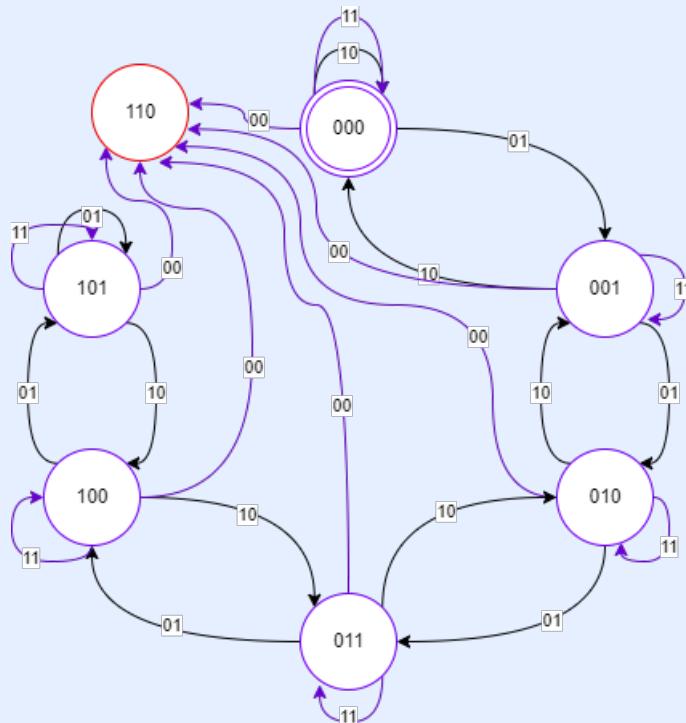
Desde el 0 al 5 tenemos 6 números, por lo tanto haremos que cada número en binario sea un estado teniendo un total de 6, así que necesitaremos $2^3=8$ estados(como solo necesitamos 6 dos estados no serán usados):



El problema nos dice que si la combinación de las entradas es (0,1) significa que sumamos uno y si es (1,0) restamos uno. Sabiendo esto vamos a unir los estados con flechas:

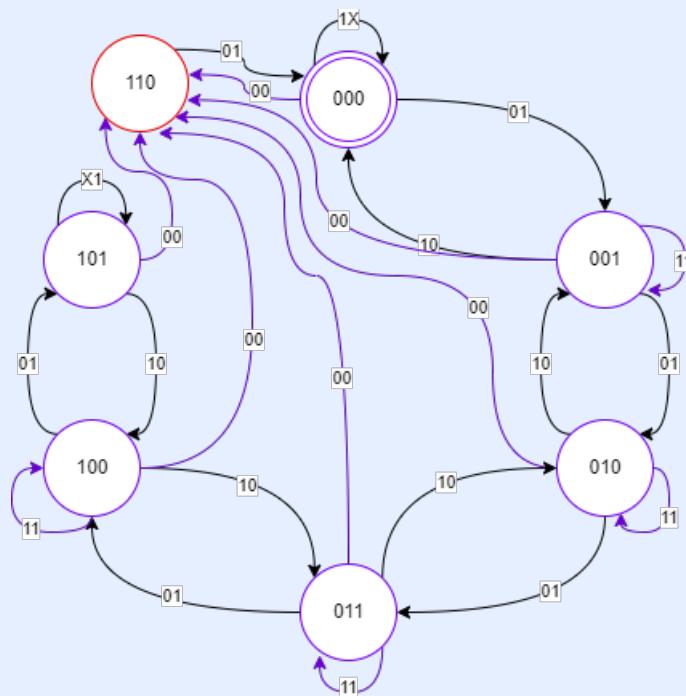


Listo, tenemos las sumas y las restas, nota que cuando llegue al estado 000 y restamos se queda en el mismo estado; lo mismo que si sumamos al último estado (101) se quedará en el mismo estado, pero podemos mejorar este modelo. Definiremos lo siguiente: que cuando las entradas sean (1,1) nos mantengamos en el mismo estado, ademas añadiremos un estado nulo que lo definiremos como (110^a):



Ya hemos implementado el estado OFF (estado 110), con el estado (1,1) nos mantenemos en el mismo estado, con el estado (0,0) apagamos el circuito. Se puede mejorar los estados donde hay

dos entradas que nos llevan al mismo estado y aumentamos un flecha para salir del estado OFF:



Resumiendo, sabiendo las entradas (x_1, x_0):

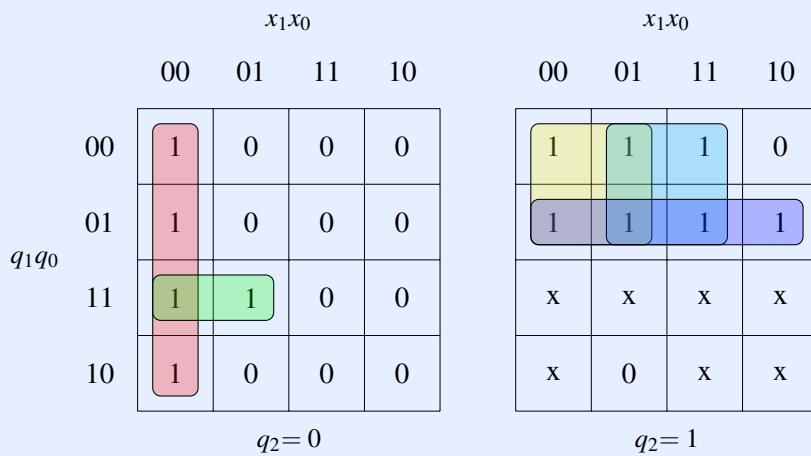
- (0,1) incremento
- (1,0) decremento
- (1,1) ni uno pulsado
- (0,0) apagamos el circuito

La tabla de verdad sería la siguiente:

Estado	q_2	q_1	q_0	x_1	x_0	q_2^+	q_1^+	q_0^+
Estado 0	0	0	0	0	0	1	1	0
	0	0	0	0	1	0	0	1
	0	0	0	1	0	0	0	0
	0	0	0	1	1	0	0	0
Estado 1	0	0	1	0	0	1	1	0
	0	0	1	0	1	0	1	0
	0	0	1	1	0	0	0	0
	0	0	1	1	1	0	0	1
Estado 2	0	1	0	0	0	1	1	0
	0	1	0	0	1	0	1	1
	0	1	0	1	0	0	0	1
	0	1	0	1	1	0	1	0
Estado 3	0	1	1	0	0	1	1	0
	0	1	1	0	1	1	0	0
	0	1	1	1	0	0	1	0
	0	1	1	1	1	0	1	1
Estado 4	1	0	0	0	0	1	1	0
	1	0	0	0	1	1	0	1
	1	0	0	1	0	0	1	1
	1	0	0	1	1	1	0	0
Estado 5	1	0	1	0	0	1	1	1
	1	0	1	0	1	1	0	1
	1	0	1	1	0	1	0	0
	1	0	1	1	1	1	0	1
Estado OFF	1	1	0	0	0	x	x	x
	1	1	0	0	1	0	0	0
	1	1	0	1	0	x	x	x
	1	1	0	1	1	x	x	x

Ya solo nos queda simplificar las salidas q^+ :

Mapa de Karnaugh para q_2^+



Mapa de Karnaugh para q_1^+

		x_1x_0						x_1x_0			
		00	01	11	10			00	01	11	10
		00	1	0	0	0	00	1	0	0	1
		01	1	1	0	0	01	1	0	0	0
		11	1	0	1	1	11	x	x	x	x
		10	1	1	1	0	10	x	0	x	x

$q_2 = 0$ $q_2 = 1$

Mapa de Karnaugh para q_0^+

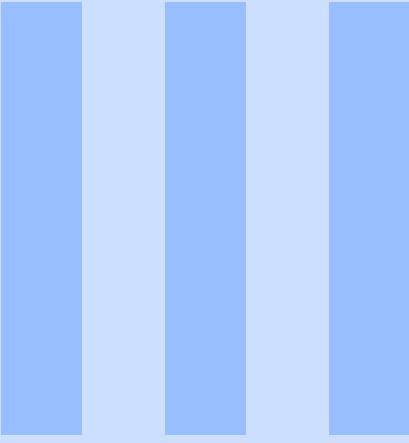
		x_1x_0						x_1x_0			
		00	01	11	10			00	01	11	10
		00	0	1	0	0	00	0	1	0	1
		01	0	0	1	0	01	1	1	1	0
		11	0	0	1	0	11	x	x	x	x
		10	0	1	0	1	10	x	0	x	x

$q_2 = 0$ $q_2 = 1$

Ya solo nos queda expresar en la forma de funciones lógicas y podemos implementar el circuito.



^aAunque este es el binario del número 7, definiremos que este estado sea neutro/apagado/off.



Smart cities

5	Conociendo la ESP8266	95
5.1	ESP8266	
5.2	Instalación Windows:Materiales	
5.3	Instalación	
5.4	Equivalente al Hola Mundo pero en placas	
5.5	Conectar a una red Wi-Fi	
5.6	Crear una red WiFi	
6	Servicios en la red	101
6.1	Crear un servidor local	
6.2	Lectura Analógica	
6.3	Proyecto explicativo 1	
6.4	Sensor de temperatura y humedad DHT11	
7	Proyecto final	115
7.1	Software	
7.2	Hardware	
7.3	Back-end	
7.4	Front-end	

5. Conociendo la ESP8266

Todos los códigos estarán disponibles en **GitHub**, puedes acceder desde este link: <https://github.com/Yasperterian/4rduino> o desde este código QR:

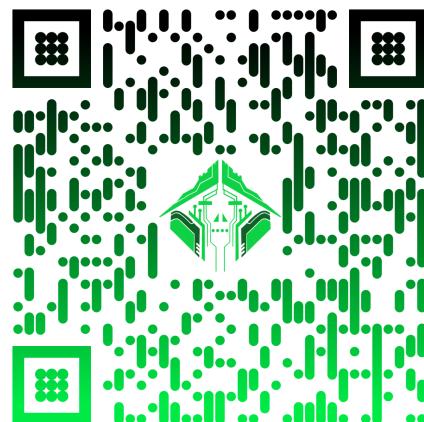


Figura 5.1: Repositorio con todos los programas

5.1 ESP8266

NodeMCU ESP8266 es una plataforma de desarrollo similar a **Arduino** especialmente orientada al Internet de las cosas (IoT). La placa NodeMcu v2 ESP8266 tiene como núcleo al **SoM ESP-12E** que a su vez está basado en el **SoC Wi-Fi ESP8266**, integra además el conversor USB-Serial TTL CP2102 y conector micro-USB necesario para la programación y comunicación a PC. NodeMcu v2 ESP8266 está diseñado especialmente para trabajar montado en



Figura 5.2: Microcontrolador ESP8266 NodeMCU

protoboard o soldado sobre una placa. Posee un regulador de voltaje de 3.3V en placa, esto permite alimentar la placa directamente del puerto micro-USB o por los pines 5V y GND. Los pines de entradas/salidas (GPIO) trabajan a 3.3V por lo que para conexión a sistemas de 5V es necesario utilizar conversores de nivel como: Conversor de nivel 3.3-5V 4CH o Conversor de nivel bidireccional 8CH - TXS0108E.

5.2 Instalación Windows:Materiales

Software IDE arduino Disponible desde <https://www.arduino.cc/en/software>

Cable micro usb Que sea capaz de transmitir datos, no solo energía.

Driver del ESP8266 No siempre es necesario. Si windows no lo reconoce por si solo, es necesario que instales el driver correspondiente antes de todo.¹

5.3 Instalación

1. Descargar e instalar el IDE de arduino, la primera vez que lo abras te preguntará sobre el *Firewall de windows*, permite el acceso².
2. Una vez instalado, conectar tu ESP8266 por cable usb, windows lo debe reconocer así que espera a que se instale el driver automáticamente.
3. Nos dirigimos a *Archivo>Preferencias>Gestor de URLs Adicionales de Tarjetas* y pegamos esta dirección: https://arduino.esp8266.com/stable/package_esp8266com_index.json³. Le damos en OK.
4. Vamos a *Herramientas>Placa:"Arduino/Genuino UNO">Gestor de Tarjetas*⁴
5. Introducimos en el buscador **ESP8266**, donde nos salga el resultado de la búsqueda clickamos en instalar. Esperemos que termine de instalar.
6. Vamos a *Herramientas>Placas>Gestor de tarjetas>ESP8266 Boards(3.0.2)>NodeMCU 1.0 (ESP-12E Module)*
7. Vamos a *Herramientas>Upload Speed*, y (recomendablemente)seleccionamos **115200**(*También asegurarte de fijar la misma cantidad usando Serial.begin(115200) en el código*);ademas que con el **Monitor serie** tengan el mismo número de baudios).
8. *Herramientas>Puerto* y seleccionamos el puerto que aparece de la forma **COMxx**, donde xx son números. Al abrir el IDE de Arduino siempre fijate que este seleccionada la COM correcta.

5.4 Equivalente al Hola Mundo pero en placas

Como en la programación de computadoras es un clásico que nuestro primer programa sea el famoso "Hola mundo", dentro del mundo de programación de placas existe su equivalente: Hacer parpadear un LED.

1. Nos dirigimos a *Archivo>Ejemplos>01.Basics>Blink*.
2. Se abrirá una ventana nueva. Luego buscamos dentro del código *void setup()* y justo por encima de ella escribimos: **#define LED_BUILTIN 2**. El número dos significa el puerto del LED, puede ser 1 en algunos casos particulares, se recomienda leer los puertos leds de tu placa.

¹Puedes probar el driver <https://parzibyte.me/blog/2020/02/09/instalar-driver-esp8266-windows/>

²Si deseas cambiar el lenguaje a español ve a: *File>Preferences>Editor Lenguaje>Español(Spanish)*. Presionamos OK y reiniciamos el Arduino IDE

³Repositorio oficial: <https://github.com/esp8266/arduino>

⁴Puede tener el nombre de: Placa"*Nombre genérico*"

3. Le damos click en la flecha que apunta a la derecha en la parte superior(subir) y esperamos que compile.
4. Esperamos que termine de descargar y compilar, en consecuencia, tenemos que ver nuestro LED parpadear con una frecuencia de 1 segundo.

```

1 // the setup function runs once when you press reset or power the board
2 #define LED_BUILTIN 2
3 void setup() {
4     // initialize digital pin LED_BUILTIN as an output.
5     pinMode(LED_BUILTIN, OUTPUT);
6 }
7 // the loop function runs over and over again forever
8 void loop() {
9     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the
10    voltage level)
11    delay(1000);                  // wait for a second
12    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the
13    voltage LOW
14    delay(1000);                  // wait for a second
15 }
```



Figura 5.3:
QR pro-
gram:
1.Blink

Código 5.1: Blink Program

5.5 Conectar a una red Wi-Fi

Una de las funciones más elementales e importantes del ESP8266 es la posibilidad de conectarse a internet para realizar proyectos de IoT(*Internet of things*). Así que, como primer paso vamos a conectarnos a nuestra red Wifi:

```

1 #include <ESP8266WiFi.h>
2
3 String ssid    = "FIWI";
4 String password = "21040411";
5
6 byte cont = 0;
7 byte max_intentos = 50;
8
9 void setup() {
10     // Inicia Serial
11     Serial.begin(115200);
12     Serial.println("\n");
13     // Conexion WIFI
14     WiFi.begin(ssid, password);
15     while (WiFi.status() != WL_CONNECTED and cont < max_intentos) {
16         //Cuenta hasta 50 si no se puede conectar lo cancela
17         cont++;
18         delay(500);
19         Serial.print(".");
20     }
21     Serial.println("!!!");
22
23     if (cont < max_intentos) { //Si se conecto
24         Serial.println("*****");
25         Serial.print("Conectado a la red WiFi: ");
26         Serial.println(WiFi.SSID());
```



Figura 5.4:
QR pro-
gram:
2.Wifi-
Connect

```

27     Serial.print("IP: ");
28     Serial.println(WiFi.localIP());
29     Serial.print("macAddress: ");
30     Serial.println(WiFi.macAddress());
31     Serial.println("*****");
32 }
33 else { //No se conecto
34     Serial.println("-----");
35     Serial.println("Error de conexion");
36     Serial.println("-----");
37 }
38 }
39
40 void loop() {
41 }
```

Código 5.2: Conectarse a una red WiFi

Debemos cambiar algunas cosas en nuestro programa:

ssid Línea 3. Colocamos el nombre de nuestra red wifi, en el ejemplo la red se llama *FIWI*, asi que le cambias al nombre de tu red.

password Línea 4. Aquí colocaremos la contraseña de nuestra red, en el ejemplo es *21040411*, colocas la tuya.

Las lineas de **byte**(Lineas 6-7) son dos variables que van contar el número de intentos en los que va tratar de conectarse a la red, esta inicializada en 0 y el límite es 50 intentos. Ademas el programa tiene algunos mensajes para mostrarse en el display, como no poseemos ni un display por el momento usaremos una función del mismo IDE de arduino llamado **Monitor serie**. Tenemos que tener el *monitor serie*⁵ abierto antes de correr el programa, puesto que el código nos mostrará información acerca del estado de la placa.

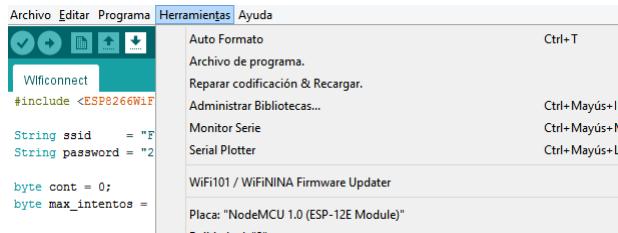


Figura 5.5: Ubicación del Monitor serie

⁵En el monitor serie asegurate de tener seleccionada: sin ajuste de linea y tener seleccionada la cantidad correcta de baudios, que coincide con el definido en el programa

5.6 Crear una red WiFi

Nuestra ESP8266, como hemos visto, se puede conectar a una red wifi pero también puede crear su propia red wifi para que dispositivos se conecten a ella. En este apartado veremos como podemos crear una red wifi:

```
1 #include <ESP8266WiFi.h>
2 const char* ssid="WIFI_ESP8266";//Name of your red
3 const char* password="12345678";//Your password, remember, at least 8
   characters
4 void setup() {
5   Serial.begin(115200);
6   Serial.print("\nSetting Ap");//Status message, you can view it in
     monitor serie
7   WiFi.softAP(ssid, password);
8   Serial.println("WiFi is ready");
9 }
10 void loop() {
11   int device= WiFi.softAPgetStationNum();//It counts numbers of users
     connected to your red
12   Serial.printf("Devices connected =%d\n",device);
13   delay(5000);
14 }
```

Código 5.3: Conectarse a una red WiFi



Figura 5.6:
QR pro-
gram:
3.Create-
WiFi



6. Servicios en la red

6.1 Crear un servidor local

Para crear esto utilizaremos el lenguaje HTML, hay varias formas pero esta es la básica para empezar.¹. Puedes hacerlo en **Sublime text**² o en **VScode**³, en nuestro caso usaremos VScode, luego copiamos el código en nuestro editor. **OJO: Hacemos este código solo para ver como funcionan las etiquetas en el "lenguaje"HTML.**

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <style>
6     html {
7       font-family: Helvetica;
8       display: inline-block;
9       margin:
10      0px auto;
11      text-align: center;
12    }
13
14   .button {
15     background-color: #195B6A;
16     border: none;
17     color:
18     white;
19     padding: 16px 40px;
20     text-decoration: none;
21     font-size: 30px;
22     margin: 2px;
```



Figura 6.1:
QR pro-
gram:
4.Server
local On
Off led

¹Si da error, puedes ponerle una tilde en la i de titulo en linea 33

²<https://www.sublimetext.com/>

³<https://code.visualstudio.com/>

```

23         cursor: pointer;
24     }
25
26     .button2 {
27         background-color: #77878A;
28     }
29 </style>
30 </head>
31
32 <body>
33     <h1>Titulo</h1>
34     <p>GPIO 5 </p>
35     <p><button class="button">ON</button></p>
36     <p>GPIO 4 </p>
37     <p><button class="button button2">OFF</button></p>
38 </body>
39
40 </html>

```

Código 6.1: Conectarse a una red WiFi

Cuando lo hemos copiado, damos click derecho en cualquiera parte del editor y le damos en *Dar formato al texto* como en la figura 6.2: Luego de ello, guardamos.

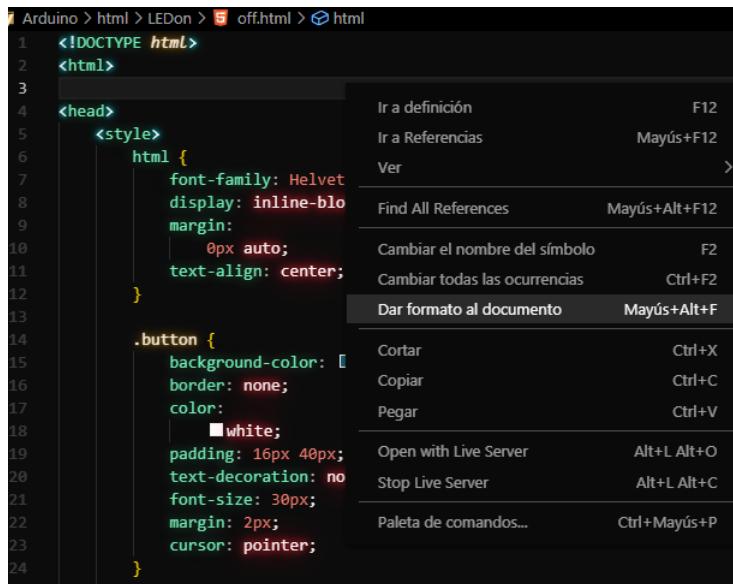


Figura 6.2: Dar formato al texto

Tenemos la primera parte. Ahora nos vamos a arduino y abrimos el programa llamado *ejemplo.ino*⁴ modificamos las lineas 3 y 4 donde nos piden nuestras credenciales de nuestra red WiFi. Una vez hecho abrimos el Monitor serie 5.5 y ejecutamos nuestro programa(no esta de más decir que te fijes que el puerto correcto esta seleccionado⁵) Implementamos el circuito de la figura 6.6, una vez implementado ejecutamos el programa. Al ejecutar observamos en nuestro Monitor que después de buscar nuestra red nos dirá su nombre y su IP como en la figura 6.3. Nos dará nuestra IP y lo copiamos

⁴Escanear el código qr 6.1 para conseguir el programa

⁵Ver item 8 de la parte 1.3 Instalación



Figura 6.3: Dar formato al texto

en nuestro navegador, por ejemplo: nuestra IP es 192.123.0.123, tal como está, lo copiamos e ingresamos.

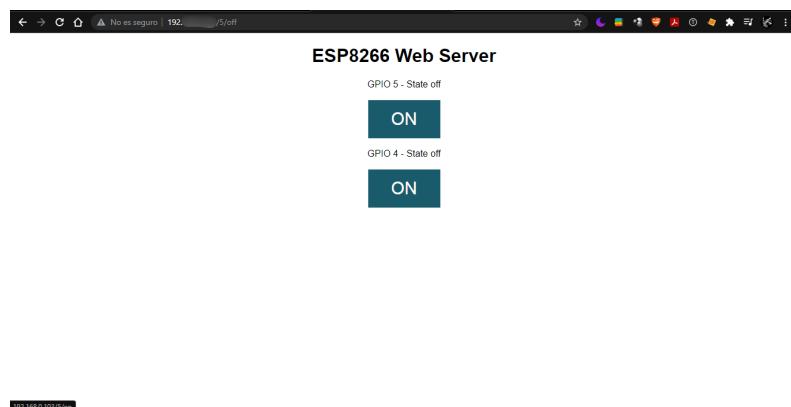


Figura 6.4: Página de ON y OFF

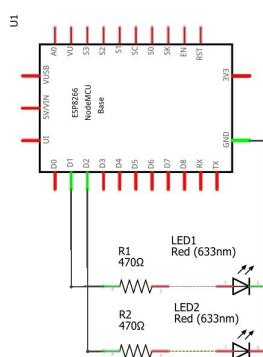


Figura 6.5: Diagrama de la conexión leds

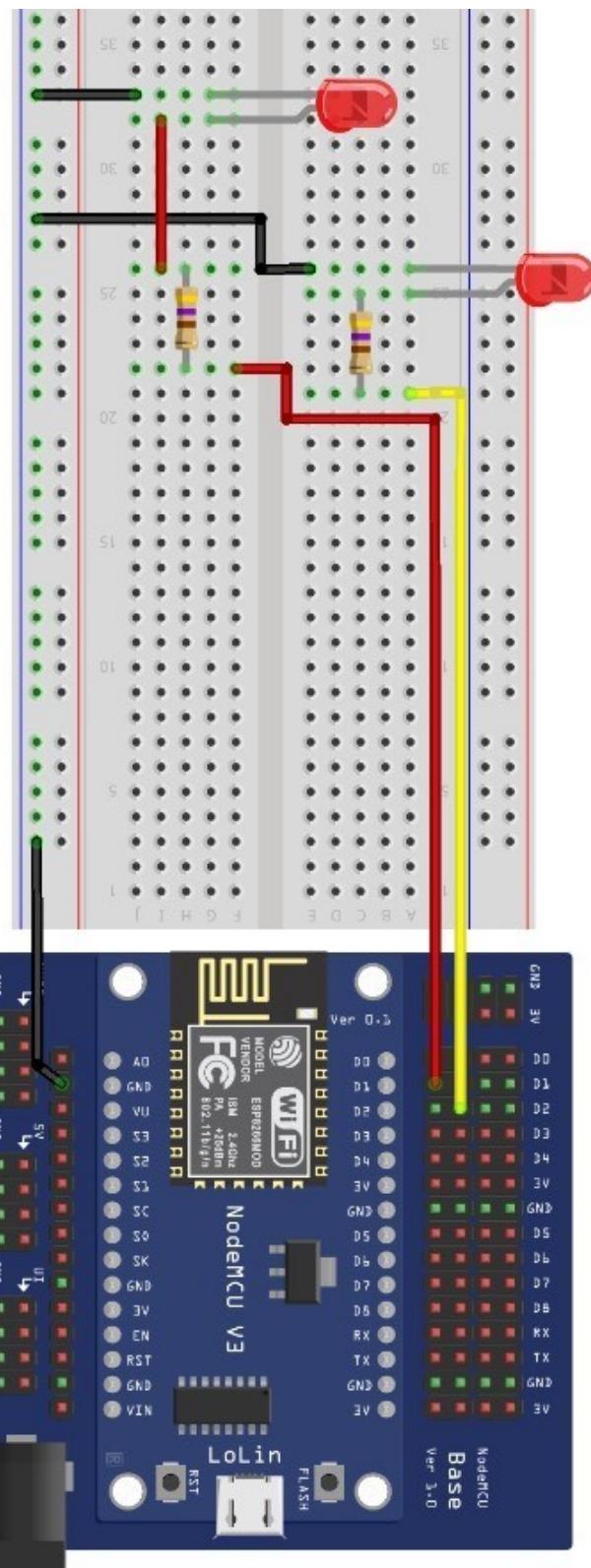


Figura 6.6: Diagrama de la conexión leds implementado, las resistencias pueden variar desde los 100 Ω hasta los 500 Ω .

6.2 Lectura Analógica

También podemos usar nuestra ESP8266 para medir dispositivos analógicos como por ejemplo, medir la resistencia de un potenciómetro: ver gráficamente como cambia este valor. En nuevo documento copiamos el siguiente programa:

```

1 const int analogInPin=A0;
2 int SensorValue=0;
3
4 void setup() {
5     // put your setup code here, to run once:
6 Serial.begin(115200);
7 }
8
9 void loop() {
10    // put your main code here, to run repeatedly:
11    SensorValue=analogRead(analogInPin);
12    Serial.println(SensorValue);
13 }
```

Código 6.2: Medir la resistencia de un potenciómetro

Y implementamos el circuito de la figura 6.8:



Figura 6.7:
QR pro-
gram:
5.GraphA-
nalog

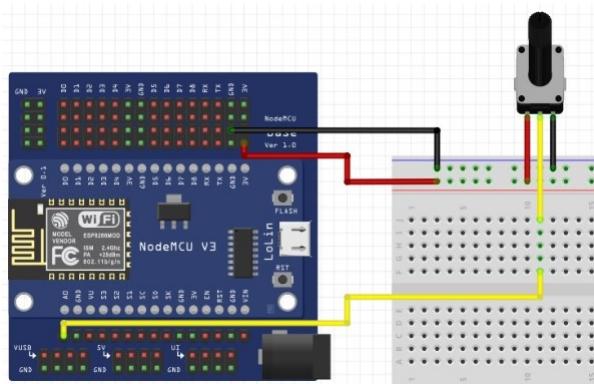


Figura 6.8: Diagrama de la conexión para el potenciómetro(presta mucha atención)

Una vez que tenemos todo conectado corremos el programa entramos a la siguiente ruta:

Herramientas>Serial Plotter(Imagen 6.9)

Luego de ingresar al Serial Plotter podremos ver el estado de la resistencia de nuestro potenciómetro desde el 0 hasta 1024.

Los pines de entrada analógica permiten leer valores analógicos que se convertirán en valores dentro del rango de 0-1024(Image6.10), ($2^{\text{elevado a } 10}$). Se trata de una conversión analógica digital de 10 bits de resolución, donde el menor valor es: **0000000000** y el máximo valor es **1111111111** ademas las señales analógicas no deben superar los 5V.

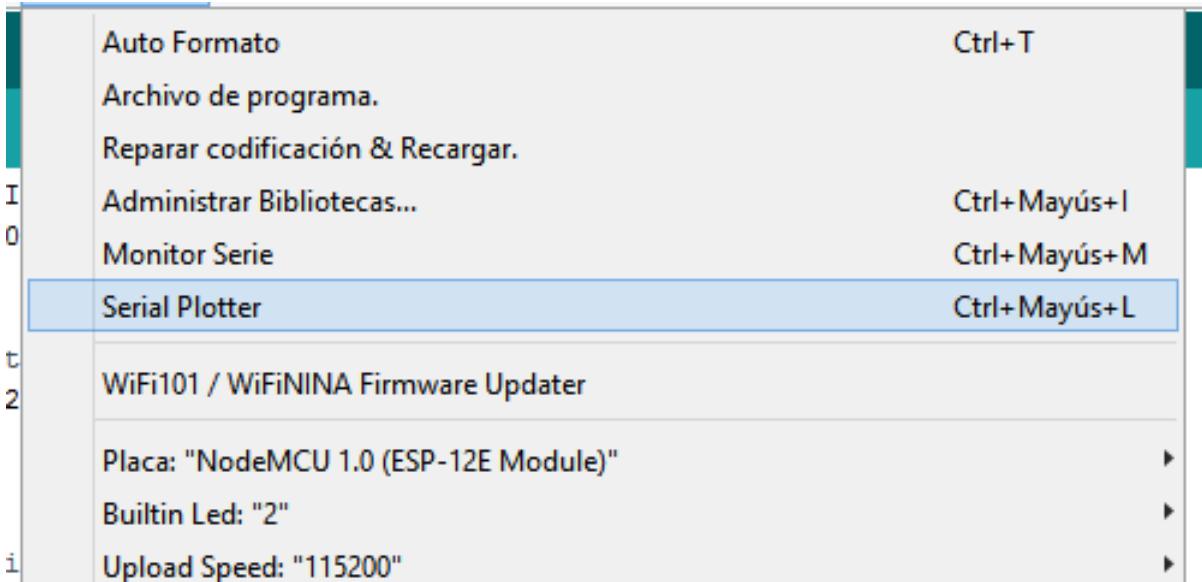


Figura 6.9: Ubicación del Serial Plotter

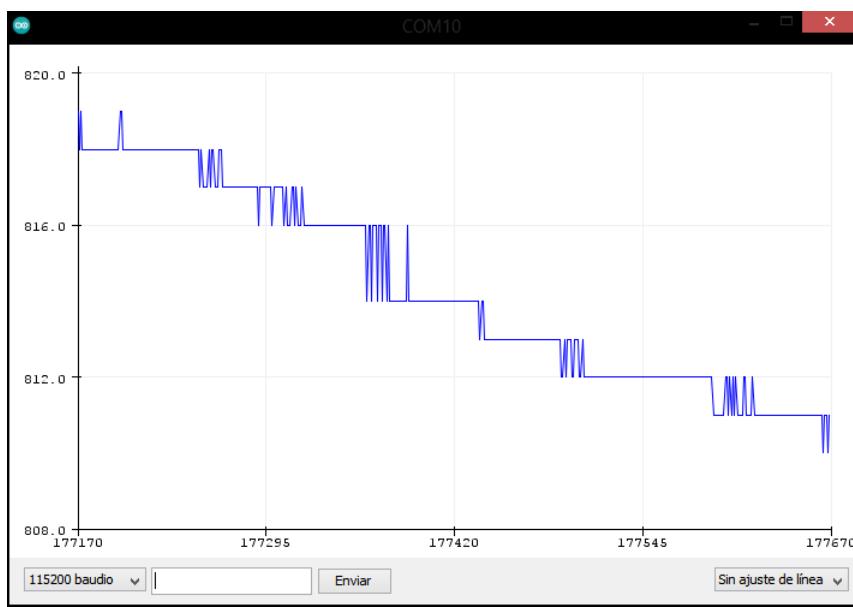


Figura 6.10: Serial Plotter

6.3 Proyecto explicativo 1

Ahora analizaremos un proyecto acerca de bases de dato, IoT y servidores locales. Antes de todo necesitamos tener instalado en nuestro computador **Xammp**, para ello no vamos a su pagina oficial y los descargamos. Una vez instalado, lo abrimos y nos aparecerá una ventana como esta(figura 6.11).

Por ahora solo necesitamos **Apache** y **MySQL**. Si es necesario, podemos correrlo como administrador. A veces es necesario que los check estén marcados, no siempre. Le damos al botón de start para Apache y para MySQL. Esperamos que nos de la confirmación y listo. En nuestro navegador podemos ingresar a las dirección localhost(figura 6.12) y debe aparecer una pantalla del Xammp: Descargamos el archivo rar⁶, extraemos y tendremos dos carpetas: terminal y web.

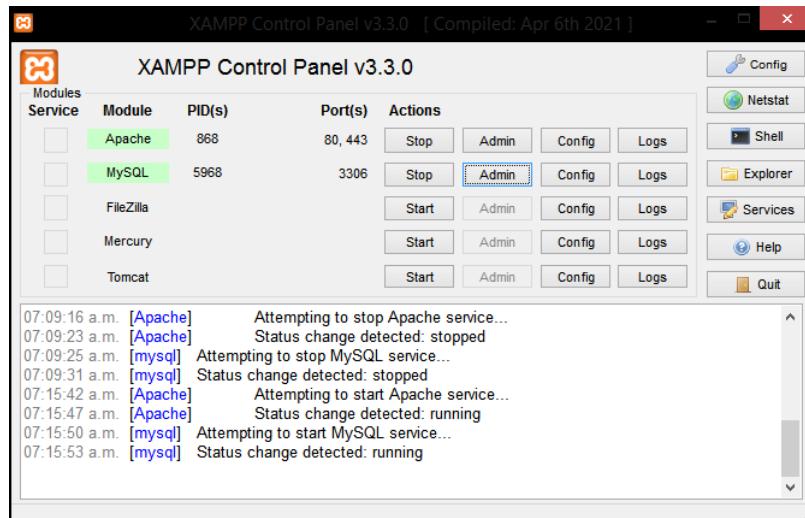


Figura 6.11: Xampp interfaz de server

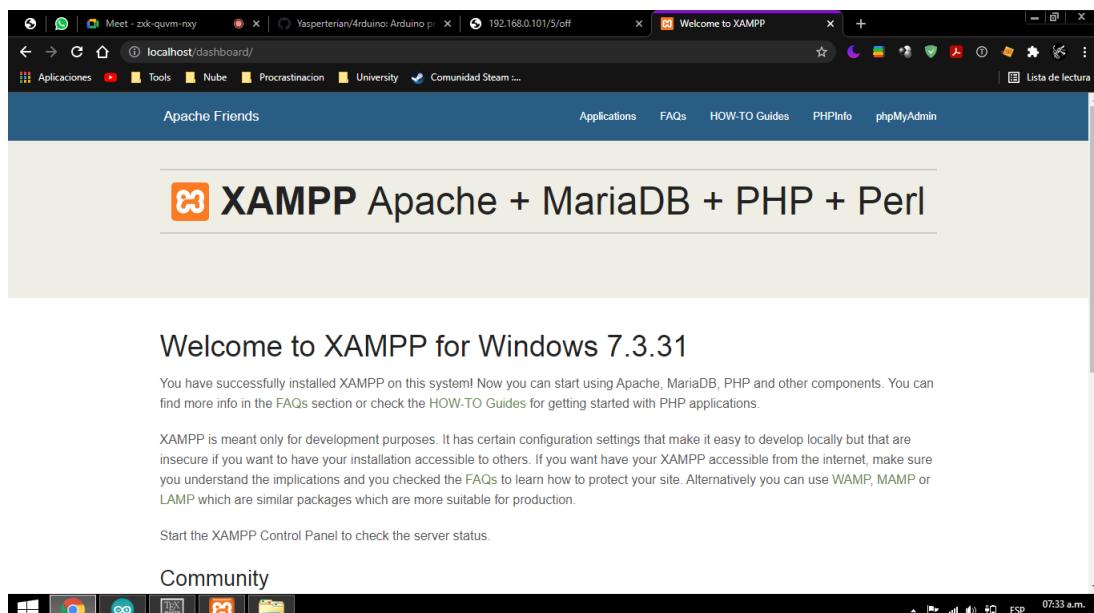


Figura 6.12: Local host en Google Chrome

Mientras avancemos tendremos que hacer algunos cambios a los archivos.

6.3.1 Parte 1

Abrimos el archivo que esta ubicado en *6. Proyecto explicativo 1>Terminal>wifi_get>wifi_get.ino*, cambiamos nuestras credenciales de la red wifi. Ahora necesitamos nuestra IP para conectarnos a la red. Para ello vamos abrir nuestro CMD(windows + R, escribimos *cmd* y enter), cuando abra escribimos **ipconfig** y buscamos **Direccion IPv4**(figura 6.14) del adaptador de red, es dirección la pegamos en la linea 44 reemplazando a la anterior(figura 6.15). Volvemos al IDE de arduino y nos vamos a la linea 44, revisamos que tenga el siguiente formato: *dirección IP/nombre de la carpeta*

⁶<https://www.dropbox.com/s/hvxl74ye9pk1slc/6.Proyecto%20explicativo%201.rar?dl=0>



Figura 6.13: QR program: 6.Proyecto explicativo 1

```
Selezionar C:\Windows\system32\cmd.exe
Adaptador de LAN inalámbrica Conexión de área local* 4:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . .
Adaptador de LAN inalámbrica Wi-Fi:
  Sufijo DNS específico para la conexión. . . .
  Vínculo: dirección IPv6 local. . . . . : fe80::54a9:c8c0:260c:70be%4
  Dirección IPv4. . . . . : 192.168.0.104
  Máscara de subred . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . : 192.168.0.1
Adaptador de Ethernet Ethernet:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . .
Adaptador de Ethernet VirtualBox Host-Only Network:
  Sufijo DNS específico para la conexión. . . .
  Vínculo: dirección IPv6 local. . . . . : fe80::1ca7:7c3:83c0:9221%17
  Dirección IPv4. . . . . : 192.168.56.1
  Máscara de subred . . . . . : 255.255.255.0
```

Figura 6.14: IP desde el CMD

que creamos/nombre de la API/usuarios y contraseña⁷.

Nos vamos a *C>xampp>htdocs*, y creamos la carpeta **practica1**, entramos en la carpeta y pegamos **api_get.php** ubicada en **6.Proyecto explicativo1>web>api_get.php**

Corremos el programa y entramos a la dirección URL de la linea 44(toda la URL)(figura 6.15) y lo ponemos en nuestro navegador. Si abrimos el archivo *api_get.php* con VScode o sublime text

```
Selezionar C:\Windows\system32\cmd.exe
Adaptador de LAN inalámbrica Conexión de área local* 4:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . .
Adaptador de LAN inalámbrica Wi-Fi:
  Sufijo DNS específico para la conexión. . . .
  Vínculo: dirección IPv6 local. . . . . : fe80::54a9:c8c0:260c:70be%4
  Dirección IPv4. . . . . : 192.168.0.104
  Máscara de subred . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . : 192.168.0.1
Adaptador de Ethernet Ethernet:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . .
Adaptador de Ethernet VirtualBox Host-Only Network:
  Sufijo DNS específico para la conexión. . . .
  Vínculo: dirección IPv6 local. . . . . : fe80::1ca7:7c3:83c0:9221%17
  Dirección IPv4. . . . . : 192.168.56.1
  Máscara de subred . . . . . : 255.255.255.0
32 rial.println("Conectado con éxito, mi IP es:");
33 rial.println(WiFi.localIP());
34
35
36
37 cle de ejecución*/
38 loop() {
39
40 WiFi.status() == WL_CONNECTED) { //Comprobamos el estado de la conexión WiFi
41
42 HTTPClient http;
43 /Indicamos el destino
44 http.begin(wifilet,"http://192.168.0.104/humedad/api_get.php?user=Efrain&pass=12345");
45 /Preparamos el header text/plain si solo vamos a enviar texto plano sin un paradigma llav
46 http.addHeader("Content-Type", "plain-text");
47 /Enviamos el post pasándole, los datos que queremos enviar.
48 /esta función nos devuelve un código que guardamos en un int)
49 int codigo_respuesta = http.GET();
50
51 if(codigo_respuesta>0){
52 //Transmitimos el número de respuesta
```

Figura 6.15: Atento con la linea 44

podemos editar la respuesta de la API.(figura 6.16)

6.3.2 Parte 2

Para la parte 2, copiamos la API de la carpeta *6.Proyecto explicativo 1>web>api_post.php* a nuestra carpeta del servidor(*C>xampp>htdocs>practica1*), igual que la parte anterior. En el programa de arduino hacemos los mismos cambios(credenciales de red y link con la IP de la linea

⁷el usuario y contraseña no es relevante aún, puedes poner cualquier cosa, pero solo puedes cambiar la palabra "joss" y "12345"

```

3
4  ?php
5
6  $user = $_GET['user'];
7  $password = $_GET['pass'];
8
9  echo "Hola esta es la respuesta a GET: ".$user." tu clave es ".$password;
10 |
11 ?|
12

```

Figura 6.16: Editar respuesta de la API

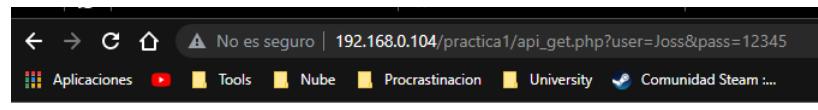


Figura 6.17: Resultado final Parte 1

44(figura 6.18)).

La diferencia está en como obtiene los datos, en la parte 1 lo mandamos a través del link. Mientras que para la parte 2, le estamos mandando por medio de una cadena **string** que esta incluida en el código, y esta nos la devolverá a la ESP8266 los datos de usuario y password mediante el **monitor series**.

```

1
2 ; http;
3 ; el destino
4 i(wifiClient, "http://192.168.0.104/practica1/api_get.php?user=Efrain&pass=12345");
5 s; el header text/plain si solo vamos a enviar texto plano sin un paradigma llave:valor.
6 :ader("Content-Type", "plain-text");
7 el post pasándole, los datos que queremos enviar.
8 i:cción nos devuelve un código que guardamos en un int)
9 ;_respuesta = http.GET();
^

```

Figura 6.18: Modificando URL

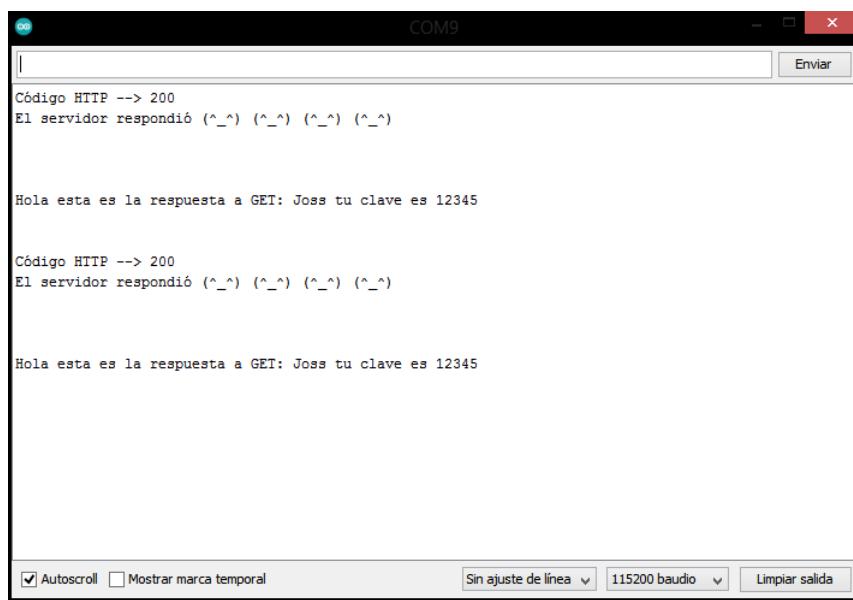


Figura 6.19: Usuario monitor series

6.3.3 Parte 3

Importante: Realizar primero la sección "Sensor de temperatura y humedad DHT11".

Para la parte 3 tenemos que abrir el sql, para ello en el administrador de servidores ponemos **admin**(figura 6.20). Creamos una base de datos, nos dirigimos a Nueva(figura 6.23)>Ponemos

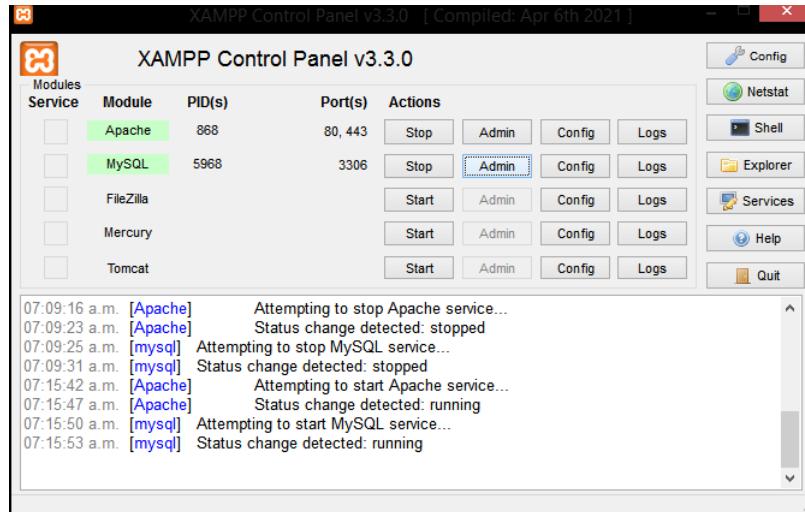


Figura 6.20: Localizar MySQL

un nombre y creamos, seleccionamos nuestra base de datos(imagen 6.21) que esta ubicada en *6. Proyecto explicativo 1>web>schema.sql*(figura 6.22), continuar y ya estará. En el panel izquierdo aparecerá nuestra base de datos. De la carpeta del proyecto copiamos *receive_data.php* a la carpeta

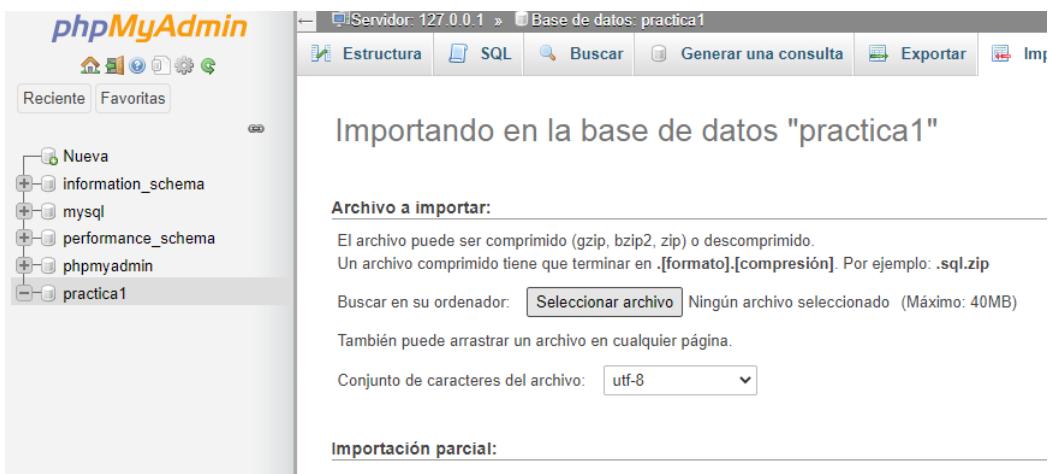


Figura 6.21: Importando SQL

del servidor. Lo abrimos, y modificamos el **dbname** que por defecto esta en **schema**, le ponemos el nombre de nuestra carpeta(practical1 en nuestro caso(figura 6.24) y guardamos.

Ahora abrimos el programa **Send_Data_MySQL.ino** y modificamos lo mismo que los anteriores(credenciales, nuestra IP, carpeta del servidor(linea del comentario *Indicamos el destino-Modificar* debe parecido a esto pero con tus datos: `http.begin(wifiClient, "http://192.168.0.104/practica1/receive_data.php");`)

Nos fijamos bien en la conexión de los puertos del sensor de temperatura y humedad(**La instalación de este sensor se detalla más adelante**). Una vez que tenemos todo listo, corremos el programa y

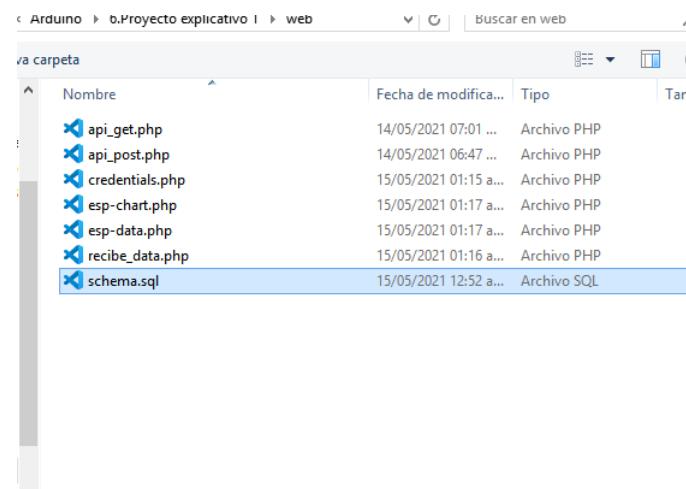


Figura 6.22: Seleccionar un archivo sql



Figura 6.23: Creando una base de datos

```
$servername = "localhost";
$dbname = "practica1"; // replace with dbname
$password = ""; // replace with password
$username = "root"; // replace with username

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $temperature = floatval($_POST["temperature"]);
    $humidity = floatval($_POST["humidity"]);
    $connection = new mysqli($servername, $username, $password);
    if ($connection->connect_error) die("Connection failed: " . $connection->connect_error);
}
```

Figura 6.24: Cambiar Scheme

nos vamos a nuestro servidor, nos dirigimos a **practica 1>sensor data** y veremos(figura6.25) que los datos se registrarán en nuestro servidor local.

6.4 Sensor de temperatura y humedad DHT11

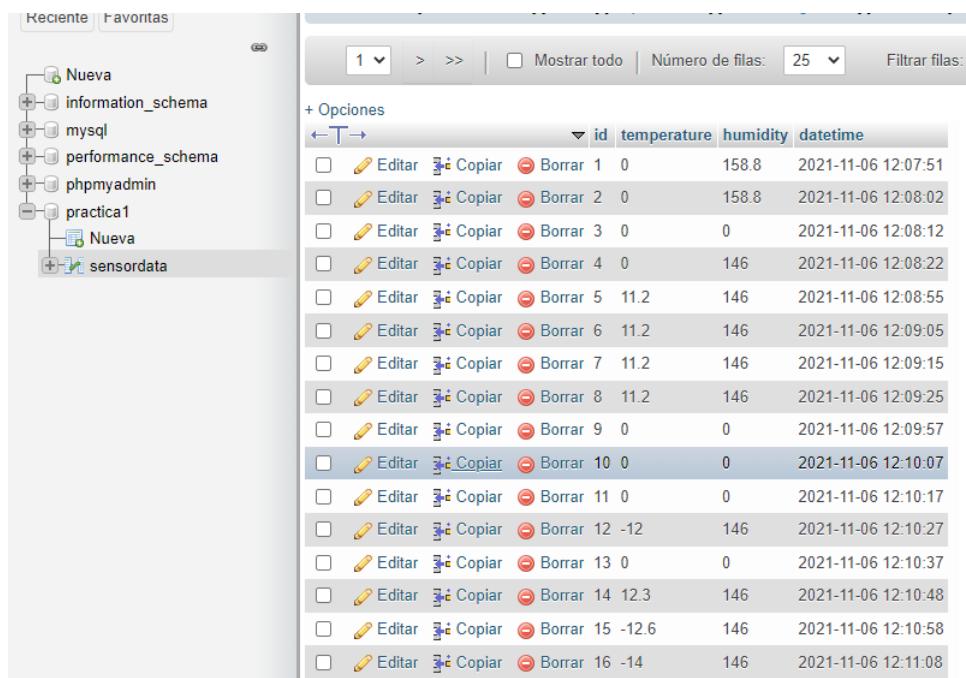
Instalamos las librerías desde la IDE de arduino, para eso vamos a **Programa>Incluir Librería>Administrar bibliotecas** y buscamos DHT11(figura 6.26), uno hecho por Adafruit.

Probamos el programa testSensorDHT11.ino y vemos si todo funciona correctamente, debe leer la temperatura y la humedad y temperatura en grados farenheit y mostrará en el **monitor serie**.

Implementamos el diagrama de la figura 6.29 encima del diagrama de la figura 6.6, es preferible cambiar el color de los leds, rojo que este conectado al puerto D2 y verde al puerto D1. Si da error de lectura, asegúrate que este bien conectado el DHT11, a veces no hace contacto bien con la protoboard.



Figura 6.27: QR
program: Test
DHT11



The screenshot shows the MySQL Workbench interface. On the left, the database structure is displayed with databases like information_schema, mysql, performance_schema, phpmyadmin, practica1, and sensordata. The sensordata database is selected. On the right, a table named 'practica1' is shown with columns id, temperature, humidity, and datetime. The table contains 16 rows of data.

	<input type="button" value="Editar"/>	<input type="button" value="Copiar"/>	<input type="button" value="Borrar"/>	id	temperature	humidity	datetime
				1	0	158.8	2021-11-06 12:07:51
				2	0	158.8	2021-11-06 12:08:02
				3	0	0	2021-11-06 12:08:12
				4	0	146	2021-11-06 12:08:22
				5	11.2	146	2021-11-06 12:08:55
				6	11.2	146	2021-11-06 12:09:05
				7	11.2	146	2021-11-06 12:09:15
				8	11.2	146	2021-11-06 12:09:25
				9	0	0	2021-11-06 12:09:57
				10	0	0	2021-11-06 12:10:07
				11	0	0	2021-11-06 12:10:17
				12	-12	146	2021-11-06 12:10:27
				13	0	0	2021-11-06 12:10:37
				14	12.3	146	2021-11-06 12:10:48
				15	-12.6	146	2021-11-06 12:10:58
				16	-14	146	2021-11-06 12:11:08

Figura 6.25: Datos en la nube local

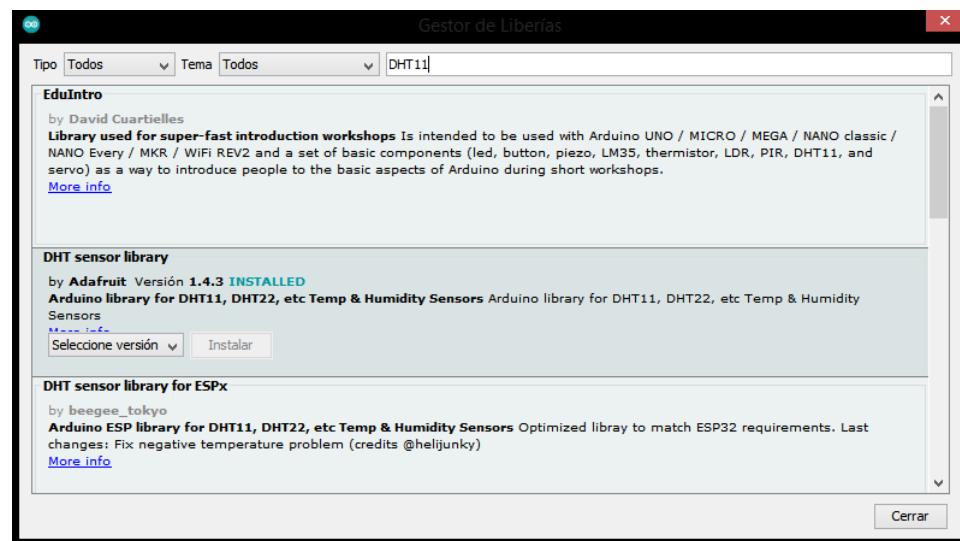


Figura 6.26: Librerías del DHT11

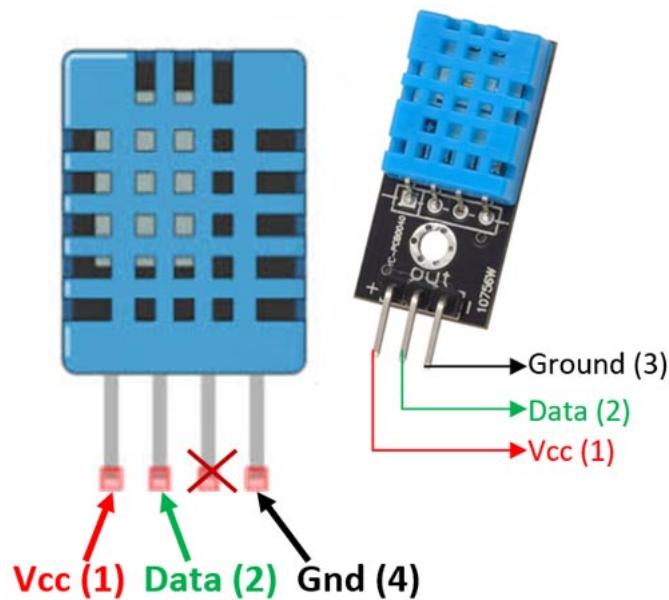


Figura 6.28: DHT11 pinout

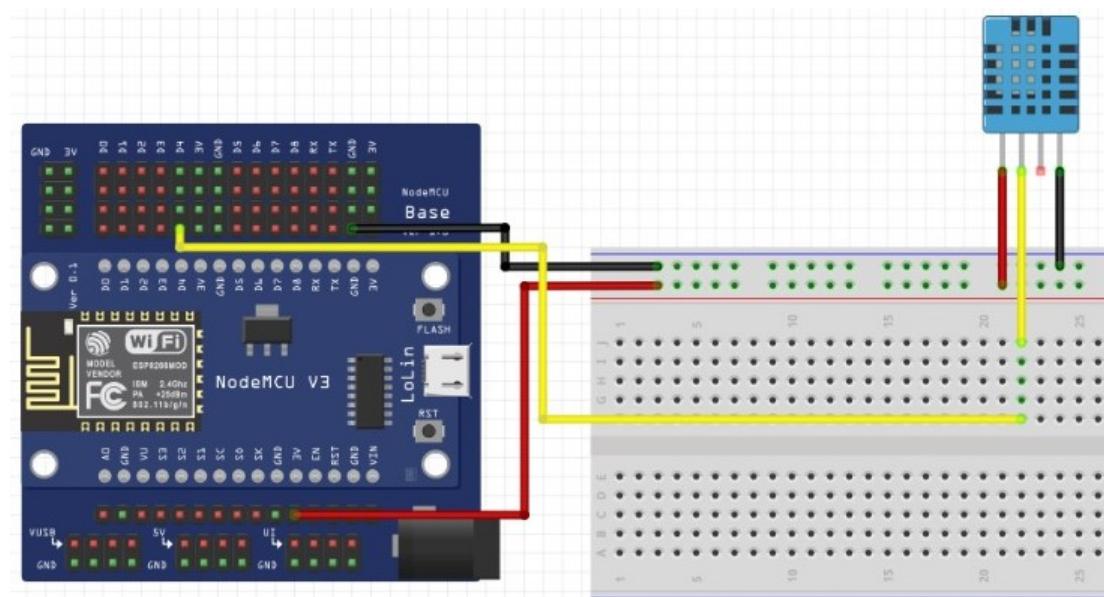


Figura 6.29: Diagrama de la conexión para el DHT11



7. Proyecto final

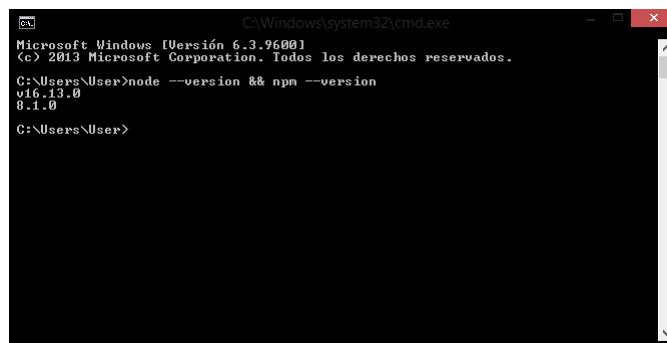
7.1 Software

Para este proyecto necesitaremos instalar algunas cosas, por ahora tenemos que instalar **node.js** y **Mosquitto**. No te explicaré a profundo sobre que van o que hacen cada tecnología, eso ya depende de ti.

7.1.1 Instalación de Node.js

Descargamos el paquete de instalación de <https://nodejs.org/en/>, la que tu quieras pero preferible la más estable, en mi caso yo trabajaré con la 16.13.0 LTS. Abriremos el archivo *.msi* y lo ejecutamos, aceptamos todo e instalamos. Una vez finalizada abrimos nuestra ventana de comandos¹, dentro del CMD² escribimos: **node --version && npm --version** y presionas enter, esperas y te debe dar las versiones que actualmente tienes en tu PC(figura 7.1).

Si llegamos aqui sin problemas tenemos el node.js instalado perfectamente. Luego de ello, sin salir



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.
C:\Users\User>node --version && npm --version
v16.13.0
8.1.0
C:\Users\User>
```

Figura 7.1: Versión Node red

del CMD abierto escribimos: **npm install -g --unsafe-perm node-red**, con esto instalaremos Node

¹windows+R, escribimos cmd y presioamos enter

²para powershell escribes: node --version; npm --version

red en nuestra computadora. Esperamos a que instale.

Cuando ya se haya instalado cerramos el CMD y lo volvemos a abrir, para correr el node red escribimos el siguiente comando: node-red³, esperamos a que se lance el programa. Una vez lanzado, buscamos la linea donde nos dice en que dirección esta corriendo el server(figura 7.2) y entramos a esa dirección desde nuestro navegador. Aprovechando que estamos dentro de la interfaz

```

node red
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.
C:\Users\user>node-red
28 Nov 17:58:16 - [info] Welcome to Node-RED
=====
28 Nov 17:58:16 - [info] Node-RED version: v0.2.1.3
28 Nov 17:58:16 - [info] Node.js version: v6.13.0
28 Nov 17:58:16 - [info] Windows_NT 6.3.9600 x64 LE
28 Nov 17:58:22 - [info] Loading palette nodes
28 Nov 17:59:22 - [info] Dashboard version 3.0.4 started at /ui
28 Nov 17:59:25 - [info] Settings file : C:\Users\user\.node-red\settings.json
28 Nov 17:59:25 - [info] Context store : 'default' (module=memory)
28 Nov 17:59:25 - [info] User directory : C:\Users\user\.node-red
28 Nov 17:59:25 - [info] Flow history enabled: true (maxHistory=1000, maxEvents=1000, enabled=false)
28 Nov 17:59:25 - [info] Flows file : C:\Users\user\.node-red\flows.json
28 Nov 17:59:25 - [info] Server now running at http://127.0.0.1:1880
28 Nov 17:59:25 - [warn]

Your flow credentials file is encrypted using a system-generated key.
  
```

Figura 7.2: Localizar el servidor actual de Node red

del Node red, instalaremos algunas paletas que nos serán utiles. Para instalar paletas vamos a las tres barras horizontales de la esquina superior derecha >Manage palettes >install; y buscamos *node-red-contrib-ibm-watson-iot*, *node-red-dashboard*. Por ahora debemos tener 4 palettes instaladas(figura 7.3). Si queremos salir de Node red, cerramos la pestaña donde esta abierto la interfaz de node

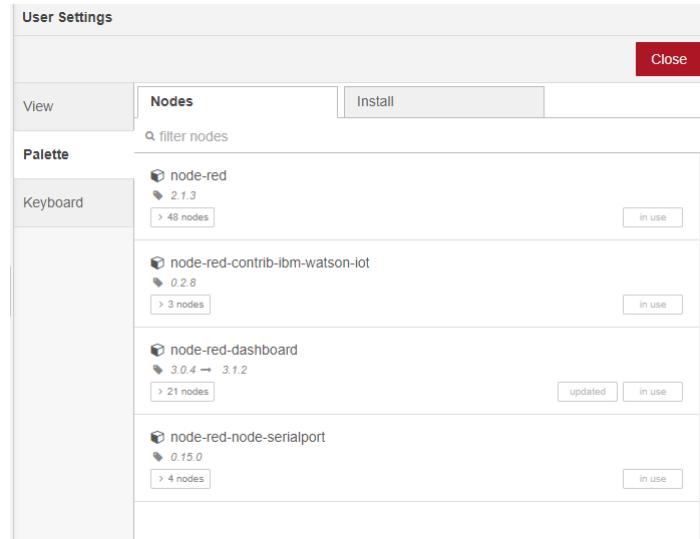


Figura 7.3: Paletas necesarias en node red

red; luego nos vamos al CMD donde se esta ejecutando node red y presionamos ctrl+c, con esto se detendrá y se cerrará.

7.1.2 Instalación de Mosquitto

Para la instalación de mosquitto, descagamos el instalador desde <https://mosquitto.org/download/>, cuando lo instalamos tenemos que tener por lo minimo marcados la casilla de **files** y la casilla **Service**. Ojo: acuerdate bien de donde lo estan instalando, en mi caso lo estoy estando el Archivos de programa, una carpeta en el disco local C. Procedemos con la instalación y esperamos.

³De no correr, prueba subir de directorio hasta el C:/, para subir tienes que escribir le siguiente comando: cd..

Una vez instalado cerramos todo y abrimos el CMD, tenemos que irnos hacia donde hemos instalado el Mosquitto. En la mayoría de los casos cuando abres el CMD estarás en la ruta: **C:\user\User**, para ir donde a la carpeta donde está instalado Mosquitto, tenemos que subir hasta el disco C y bajar. Tipeamos el comando **cd..** pasa subir un nivel, ahora nos encontramos en **C:\user**, tipeamos el comando una vez más y nos encontramos en **C:**.

Ahora que estamos en el disco C, nos dijimos a dónde hemos instalado el mosquito(en mi caso: C>Archivos de programa>mosquitto) usando el comando **dir**. Entonces tipeamos el comando: **dir** y presionamos enter, podremos ver los archivos en esa carpeta, en mi caso esta la carpeta **Program Files**, por lo que escribimos: **cd Program Files⁴** y damos enter, ahora usando de nuevo el comando **dir** para ver si efectivamente está la carpeta mosquito. Si no esta sube de nivel y sigue buscando, si esta usa el comando **cd mosquito** y con esto estamos dentro de la carpeta mosquito. Para asegurarnos que estamos en la carpeta correcta escribimos el comando **mosquitto -v**(figura 7.4). Dejamos todo por un momento, vamos a la ubicación donde hemos instalado el mosquito mediante

```

C:\Windows\system32\cmd.exe - mosquitto -v
13/12/2015 01:49 p.m. <DIR> Windows Multimedia Platform
15/06/2015 05:08 p.m. <DIR> Windows NT
29/09/2013 10:47 p.m. <DIR> Windows Photo Viewer
13/12/2015 01:49 p.m. <DIR> Windows Portable Devices
22/08/2013 10:51 a.m. <DIR> WindowsPowerShell
          0 archivos          0 bytes
        41 dirs 81,822,203,904 bytes libres

C:>cd "Program Files"
C:\Program Files>cd mosquito
C:\Program Files\mosquitto>mosquitto -v
1638142518: mosquitto version 2.0.14 starting
1638142518: Using default config.
1638142518: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1638142518: Create a configuration file which defines a listener to allow remote
connections.
1638142518: For more details see https://mosquitto.org/documentation/authentication-
methods/
1638142518: Opening ipv4 listen socket on port 1883.
1638142518: Opening ipv6 listen socket on port 1983.
1638142518: mosquitto version 2.0.14 running

```

Figura 7.4: Ver la versión de mosquito

nuestro explorador de archivos, y dentro de la carpeta donde hemos instalado mosquito buscamos el archivo **mosquitto.conf** y lo abrimos con VScode o Sublime Text; tenemos que configurar, porque si no te diste cuenta, en la figura 7.4 nos dice que las conexiones solo serán posibles para clientes DENTRO de esta máquina; pero obviamente nuestra ESP8266 no está dentro de la máquina, es por ello que editamos este archivo de configuración. Entonces añadimos las dos primeras líneas de la figura 7.5 y guardamos, si nos dice error al guardar en VScode, nos da opciones, escogemos guardar como administrador, esperamos y aceptamos la solicitud que nos pedirá. Cerramos el explorador de

```

mosquitto.conf • bpsk.py U
C: > Program Files > mosquito > mosquitto.conf
1  listener 1883#añadimos esta linea
2  allow_anonymous true#añadimos esta linea
3  |
4  # Config file for mosquitto
5  #
6  # See mosquitto.conf(5) for more information.
7  #
8  # Default values are shown, uncomment to change.
9  #

```

Figura 7.5: Editando archivo de configuración

archivos y volvemos al CMD de mosquito. Presionamos **ctrl+c** para detener mosquito y ahora usando el comando **mosquitto -c mosquitto.conf -v** hacemos que mosquito inicie usando el archivo de configuración que acabamos de modificar.

En este punto realmente ver es siguiente lista de reproducción <https://youtube.com/playlist?list=PLcmn4ItelQPYrMOJ06rCerzIcbaGYdUPz6>

⁴puedes escribir solo cd pro y apretar tab las veces que sea necesaria para que windows lo autocomplete

Para los trabajos de aquí en adelante, antes de programar tenemos que tener corriendo al mismo tiempo Node Red como Mosquitto como ya hemos visto en estos dos pasos.

Conexión de Mosquitto a Node red desde ESP8266

Antes de todo, tenemos que saber nuestra IP de nuestro adaptador sea de Ethernet o Wifi, para ello abrimos una consola de CMD y escribimos el comando: `ipconfig /all`, y uso mediante red inalámbrica, por eso busco mi IP(figura 7.6) de ese adaptador, anótate esa dirección IP porque será importante. Una vez hemos puesto a correr Mosquitto y Node Red(se tienen que ejecutar desde

```

Selezionare C:\Windows\system32\cmd.exe
DHCP habilitado : no
Configuración automática habilitada : sí
Adaptador de LAN inalámbrica Wi-Fi:
  Sufijo DNS específico para la conexión . . . . . : no
  Descripción . . . . . : Qualcomm Atheros AR9485 Wireless Network Adapter
  Dirección física . . . . . : D0-53-49-9A-27-BE
  DHCP habilitado . . . . . : sí
  Configuración automática habilitada . . . . . : sí
  Dirección IP local . . . . . : fe80::5492:8c0:260a:70be%4<Preferido>
  Dirección IP . . . . . : 192.168.0.104<Preferido>
  Máscara de subred . . . . . : 255.255.255.0
  Concesión obtenida . . . . . : domingo, 28 de noviembre de 2021 08:13:05 p.m.
  La concesión expira . . . . . : domingo, 28 de noviembre de 2021 10:10:48 p.m.
  Puerto de enlace predeterminada . . . . . : 192.168.0.1
  Servidor DHCP . . . . . : 192.168.0.1
  IDLID DHCPv6 . . . . . : 131093321
  DUID de cliente DHCPv6 . . . . . : 00-01-00-01-1D-11-C8-40-F0-76-1C-8C-7A-6D
  Servidores DNS . . . . . : 8.8.8.8
                                         8.8.4.4

```

Figura 7.6: IP de nuestro computador

CMD independientes) nos vamos a Node red y buscamos en las paletas del lateral izquierdo el Nodo **mqtt in** y lo arrastramos a nuestro entorno de trabajo, le damos doble click y lo configuramos como la figura 7.7, en server que dice *Add new mqtt-broker*, damos click al icono de lapiz de su derecha y lo configuramos como la figura 7.8. Ahora, al nodo de *mqtt in* le añadimos los siguientes

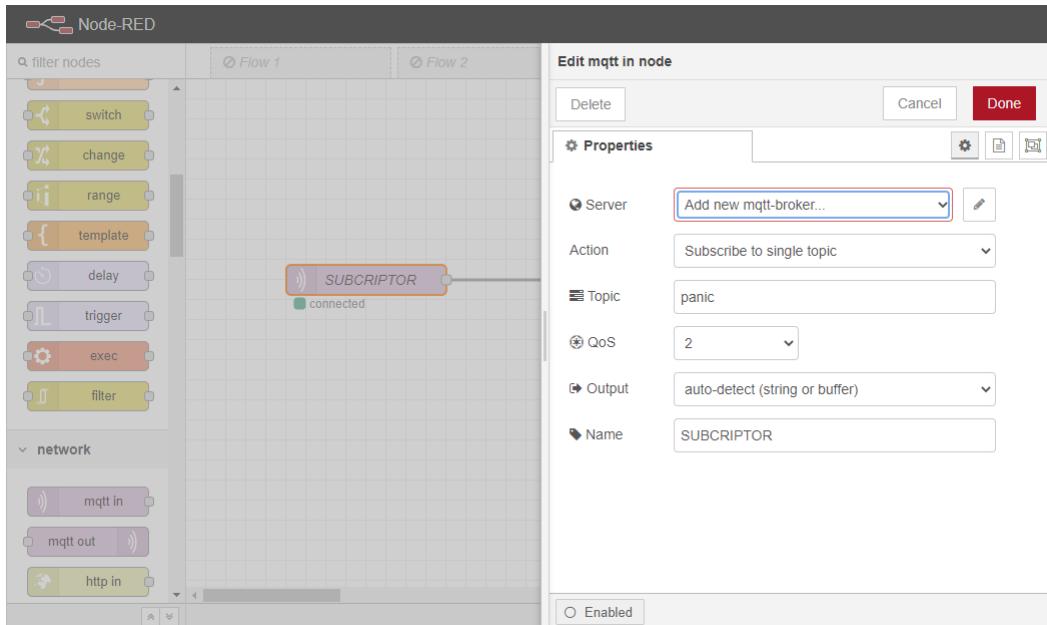


Figura 7.7: Configuración de mqtt in

nodos(ver figura 7.9 y figura 7.10):

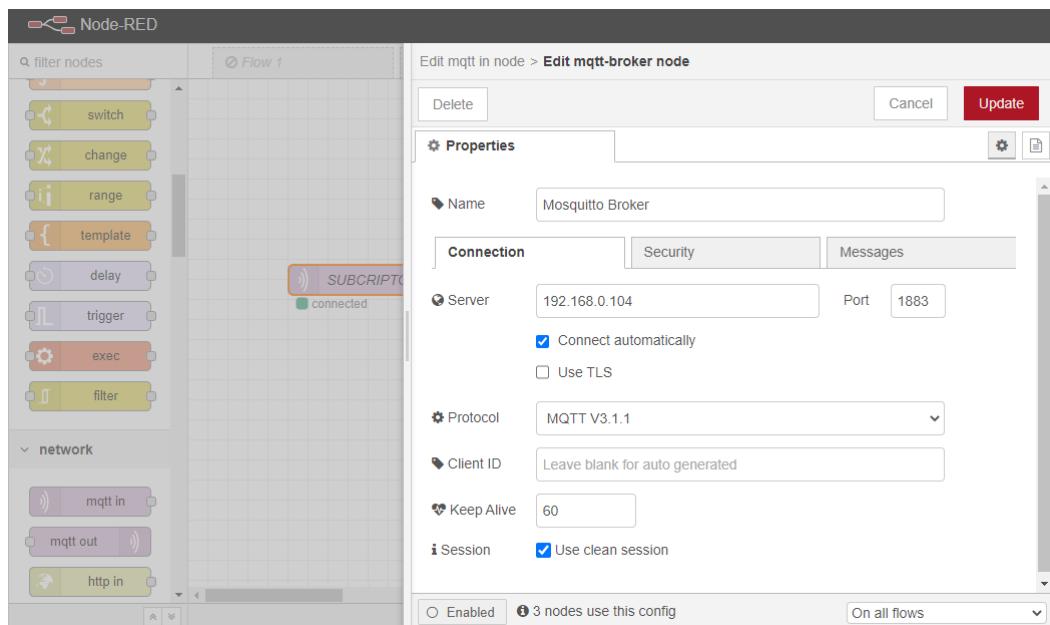


Figura 7.8: Server es nuestro IP y port es 1880 u 1883. No esta de más ver las IP de nuestro ordenador con **netstat -an** Una vez que lo termines de crear, selecciona este broker siempre en el apartado server de la figura 7.7

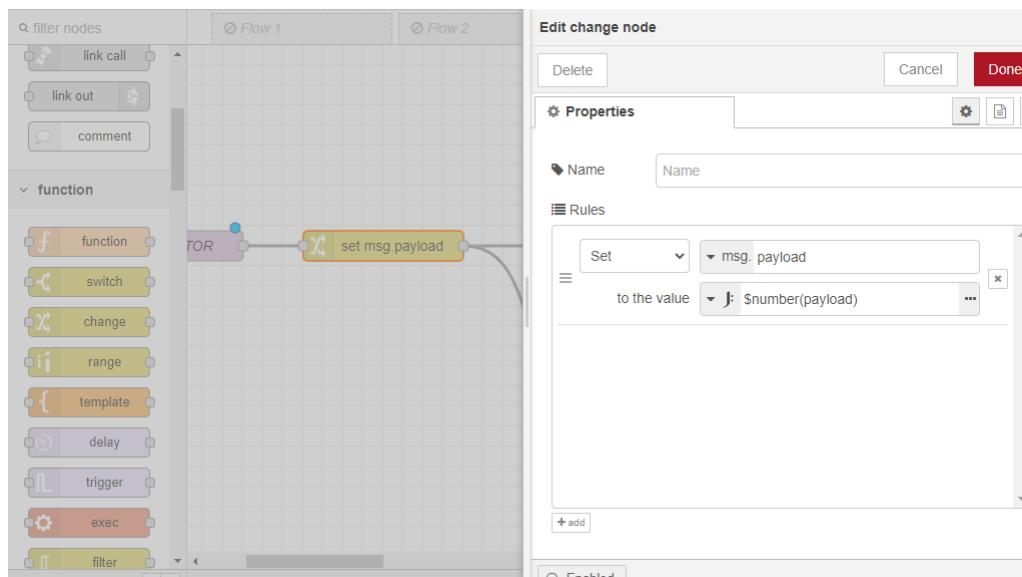


Figura 7.9: Configuración nodo *change*

El proyecto esta disponible en GITHUB. Así que vamos a dividir la explicación del proyecto en **Back-end** y **Front-end**.

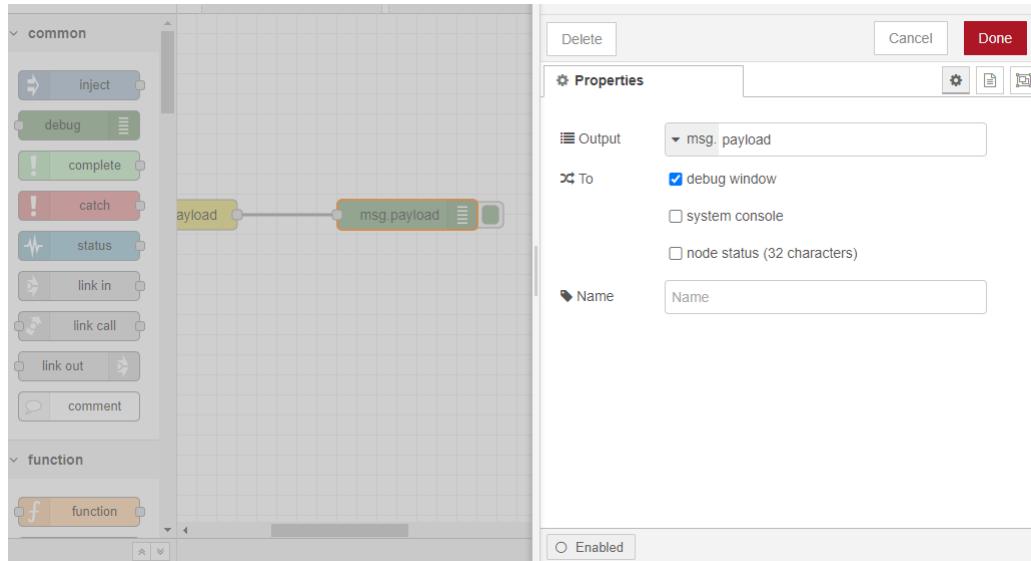


Figura 7.10: Configuración del nodo *debug*

7.2 Hardware

Los componentes requeridos son:

1. 1 ESP-8266 MCU
2. 1 I2C LCD
3. 1 LCD 16x2
4. 4 LEDs(rojo, verde, azul, amarillo)
5. 4 resistencias 470Ω
6. 1 timer 555
7. 2 resistencias $47k\Omega$
8. 1 botón momentáneo 2 pines
9. 1 metro de cables de colores

Se tiene que desplegar el circuito de la figura 7.11

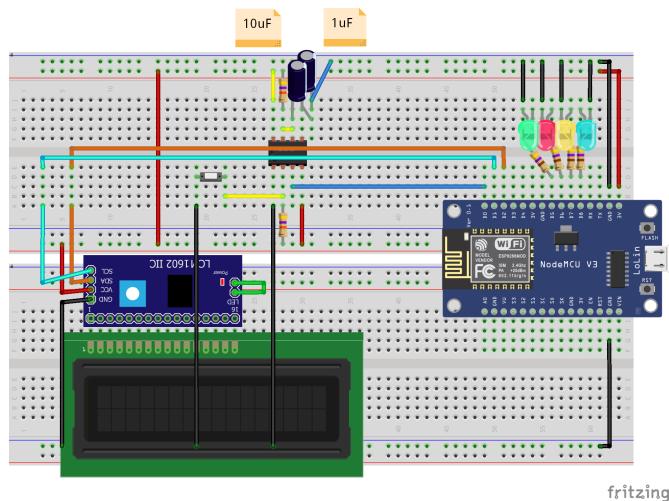


Figura 7.11: Circuito botón de pánico

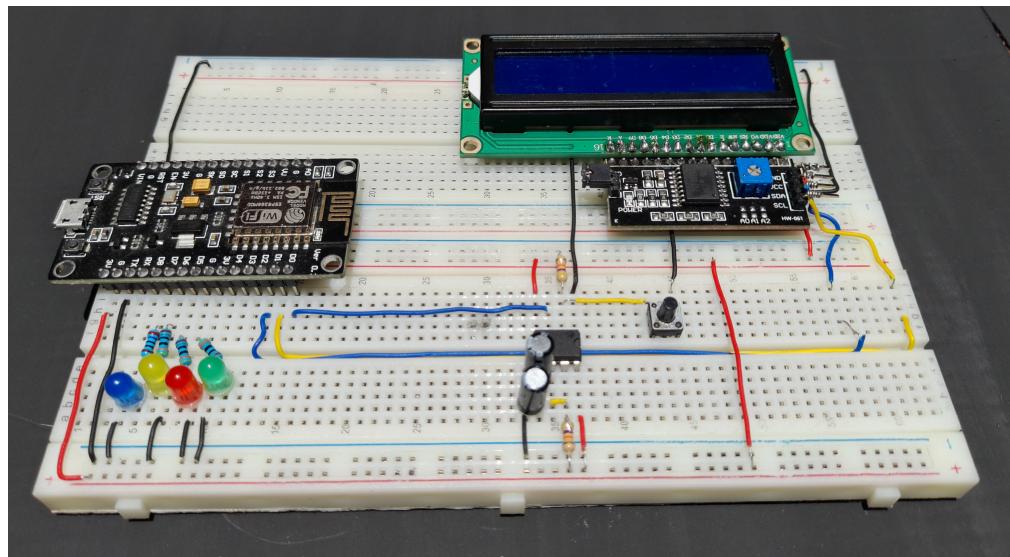


Figura 7.12: Circuito terminado-Vista trasera

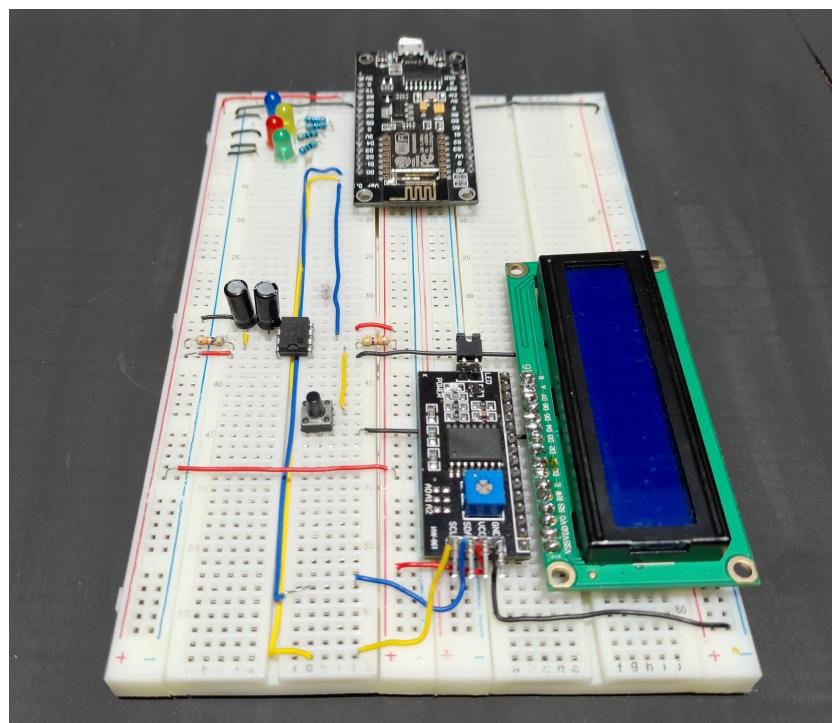


Figura 7.13: Circuito terminado-Vista izquierda

7.3 Back-end

En la parte del back-end, descargaremos el archivo *7.PanicButton.ino*. El código está comentado para comprender qué hace el programa. En cuanto a nuestra computadora, donde ya tenemos el Node-red, Xammp y Mosquitto instalado, prepararemos el entorno para que la ESP-8266 pueda enviar datos. Ojo: respeta las mayúsculas y todos los parámetros que se establecen.

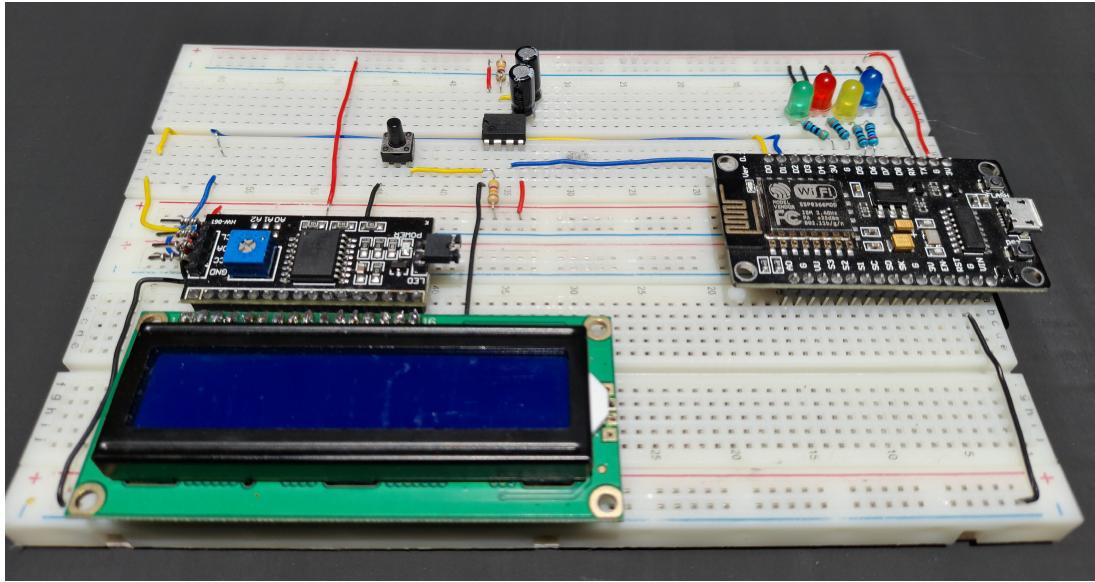


Figura 7.14: Circuito terminado-Vista frontal

7.3.1 Xammp-PHP My admin

Entramos en *localhost:/phpmyadmin*, y crearemos 3 bases de datos: para el GPS, para los usuarios y para los estados del botón.

Base de datos: GPS

Crearemos la base de datos en base a los siguientes parámetros:

Resumiendo los pasos mostrados en las imágenes⁵, podemos decir que: Se ha creado un database

Base de datos	Colejamiento	Acción
gpsdata	utf8mb4_general_ci	[+] Seleccionar privilegios
information_schema	utf8_general_ci	[+] Seleccionar privilegios
mysql	utf8mb4_general_ci	[+] Seleccionar privilegios
performance_schema	utf8_general_ci	[+] Seleccionar privilegios
phpmyadmin	utf8_bin	[+] Seleccionar privilegios
php_login_database	utf8_unicode_ci	[+] Seleccionar privilegios
practica1	utf8mb4_general_ci	[+] Seleccionar privilegios
statusbutton	utf8mb4_general_ci	[+] Seleccionar privilegios

Total: 8

Figura 7.15: Creación de la database

Estructura | SQL | Buscar | Generar una consulta | Exportar | Importar

No se han encontrado tablas en la base de datos.

Crear tabla

Nombre: puntosgps | Número de columnas: 5

Continuar

Figura 7.16: Creación de la tabla dentro del database

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice	A_I	Comentarios
id	INT		Ninguno			<input checked="" type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
Nombre	CHAR	32	Ninguno			<input type="checkbox"/>	---	<input type="checkbox"/>	
LAT	INT	32	Ninguno			<input type="checkbox"/>	---	<input type="checkbox"/>	
LON	INT	32	Ninguno			<input type="checkbox"/>	---	<input type="checkbox"/>	
time	INT	32	Ninguno			<input type="checkbox"/>	---	<input type="checkbox"/>	

Figura 7.17: Creación de las columnas de la tabla del database

de nombre **gpsdata**, que contiene una tabla llamada **puntosgps**. La tabla **puntosgps** tiene las siguientes columnas(7.18):

1. id: Con parámetro de creación A_I(auto incrementar)
2. Nombre:Tipo varchar, tamaño 32.
3. LAT:Tipo float, tamaño 32.
4. LON:Tipo float, tamaño 32.
5. time: Tipo timestamp, predeterminado CURRENT_TIMESTAMP

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)		No	Ninguna			AUTO_INCREMENT	Cambiar Eliminar Más
2	Nombre	varchar(32)	utf8mb4_general_ci	No	Ninguna				Cambiar Eliminar Más
3	LAT	float		No	Ninguna				Cambiar Eliminar Más
4	LON	float		No	Ninguna				Cambiar Eliminar Más
5	time	timestamp		No	current_timestamp()			ON UPDATE CURRENT_TIMESTAMP()	Cambiar Eliminar Más

Figura 7.18: Resumen de la creación del database: GPS

Base de datos: Usuarios

Para la base de datos, tenemos que nombrar a nuestra base: **php_login_database**, y dentro de ella crear una tabla llamada **users** con los siguientes parámetros(figura 7.19)

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)		No	Ninguna			AUTO_INCREMENT	Cambiar Eliminar Más
2	email	varchar(200)	utf8_unicode_ci	No	Ninguna				Cambiar Eliminar Más
3	password	varchar(200)	utf8_unicode_ci	No	Ninguna				Cambiar Eliminar Más

Figura 7.19: Resumen de la creación del database: Usuarios

⁵Los datos de las imágenes 7.15, 7.16 y 7.17 son erróneos, tener en cuenta ello.

Base de datos: Estados

Nombrada en phpmyadmin como: **statusbutton**, con una tabla dentro de ella, llamada **historial**, que debe tener las siguientes columnas(figura 7.20).

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	status01	int(11)			No	Ninguna			Cambiar Eliminar Más
3	tiempo	datetime			Sí	current_timestamp()			Cambiar Eliminar Más

Figura 7.20: Resumen de la creación del database: Estados

7.3.2 Xammp-Htdocs

En C>xampp>htdocs tenemos que creamos una carpeta llamada **panicbuttonserver**, y dentro de ella pegamos nuestros archivos que se encuentran en la carpeta *panicbuttonserver* dentro del repositorio de GITHUB.⁶. Como resultado debe ser como la figura 7.21

Nombre	Fecha de modifica...	Tipo	Tamaño
assets	11/12/2021 08:58 ...	Carpetas de archivos	
database.php	12/12/2021 05:01 ...	Archivo PHP	1 KB
forgotten.php	12/12/2021 12:36 ...	Archivo PHP	0 KB
index.php	28/12/2021 05:29 ...	Archivo PHP	2 KB
info.php	28/12/2021 09:56 ...	Archivo PHP	3 KB
logout.php	12/12/2021 06:17 ...	Archivo PHP	1 KB
main.php	28/12/2021 05:30 ...	Archivo PHP	2 KB
negro.php	21/12/2021 07:58 a...	Archivo PHP	0 KB
signup.php	27/12/2021 05:54 ...	Archivo PHP	2 KB

Figura 7.21: Resumen de la creación de la carpeta htdocs.

7.3.3 Node-RED

En node-red(Windows+R y escribimos node-red y presionamos enter, luego entramos la dirección que nos devuelve en la consola)vamos a configurar dos nodos paralelos:

1. Status
2. GPS

Antes de desplegar los nodos, tenemos que tener instalados los siguientes paquetes de nodos:

- node-red-contrib-ui-led
- node-red-contrib-web-worldmap
- node-red-dashboard
- node-red-node-email
- node-red-node-mysql
- node-red-node-ui-table

Nodos estados del botón

Nodo coordenadas gps

Igual se deja un archivo JSON llamado *flows.json* en la carpeta del botón de pánico para que lo puedas importar a node-red.

⁶<https://github.com/Yasperterian/4rduino>

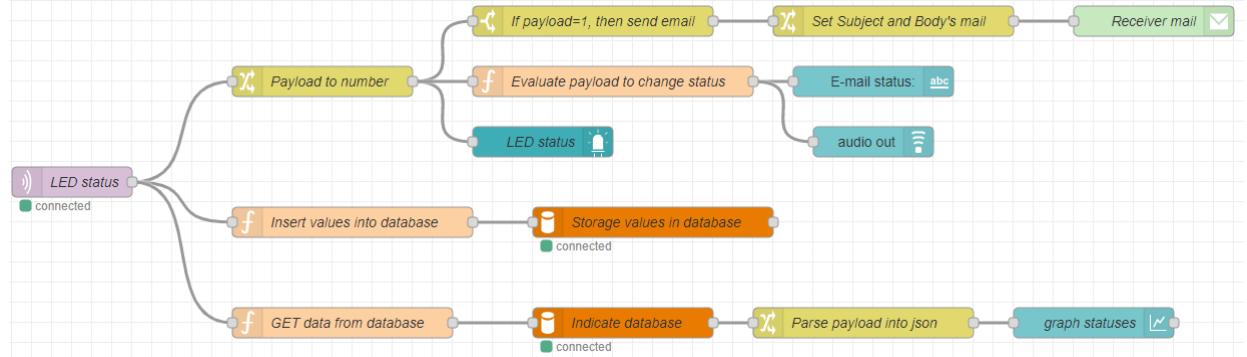


Figura 7.22: Flujo de datos: status button

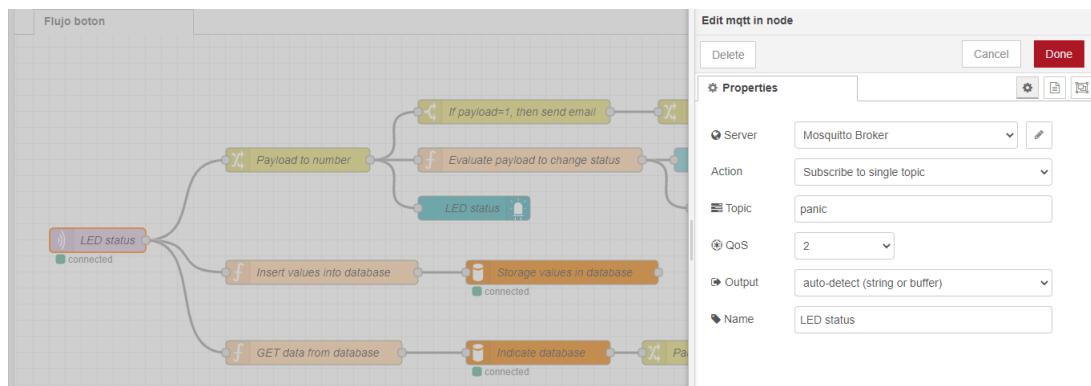


Figura 7.23: Nodo: LED status

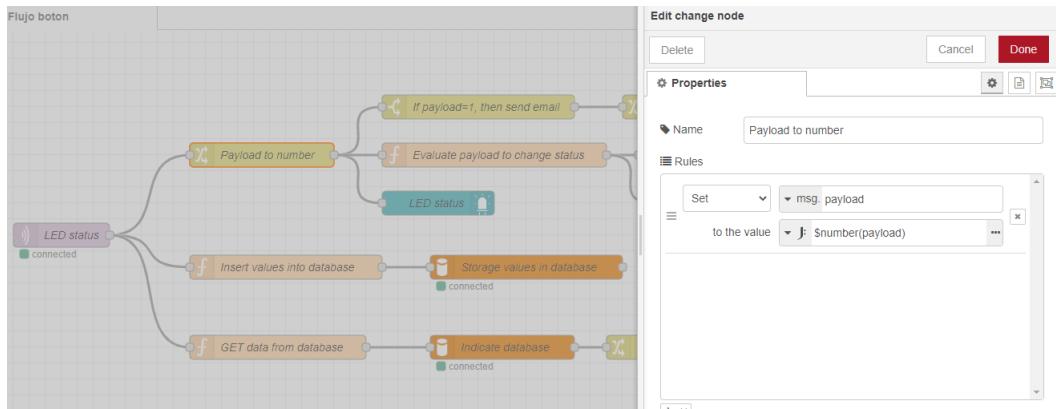
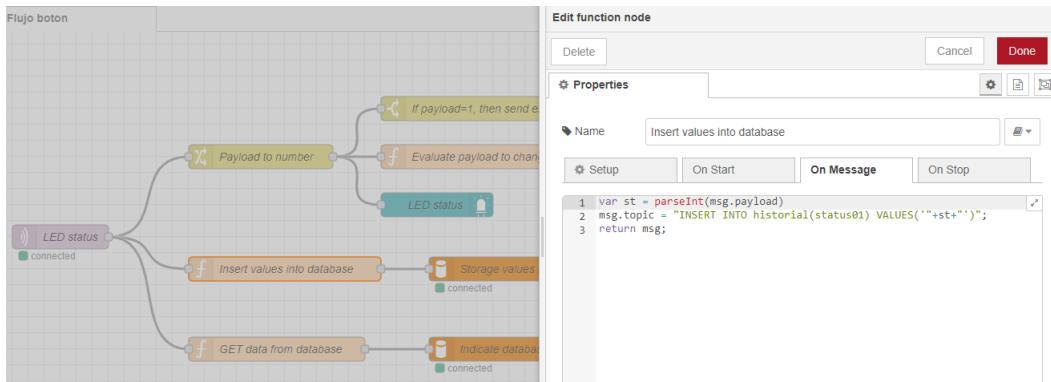
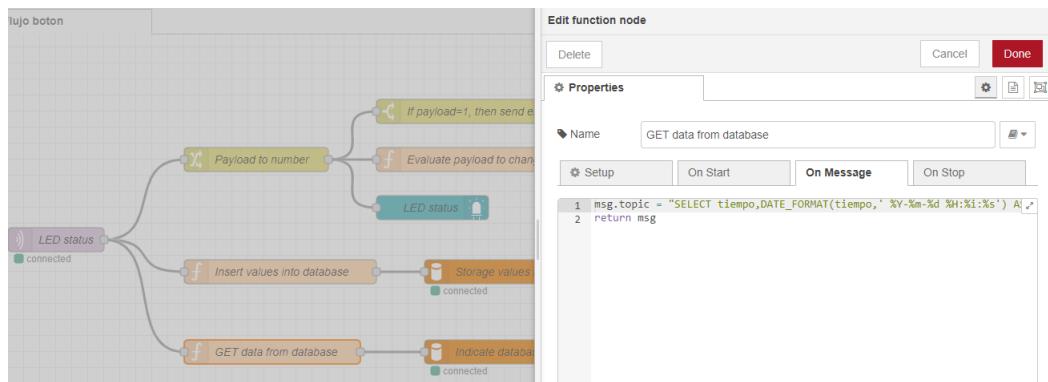
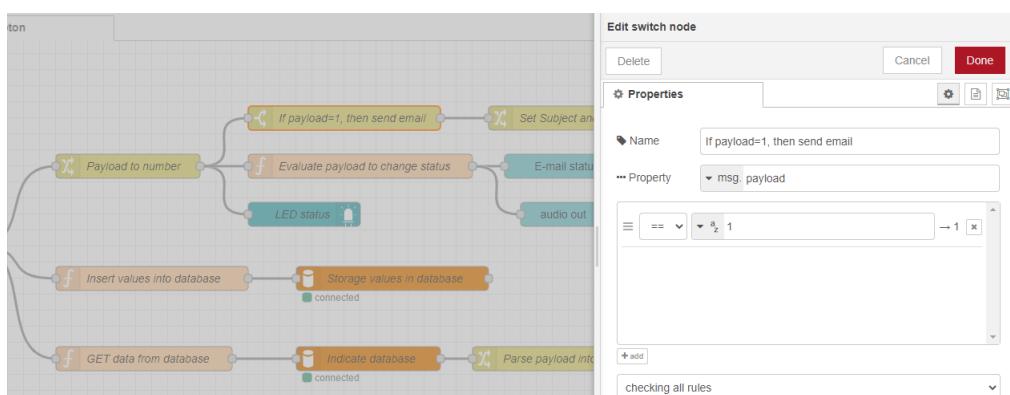
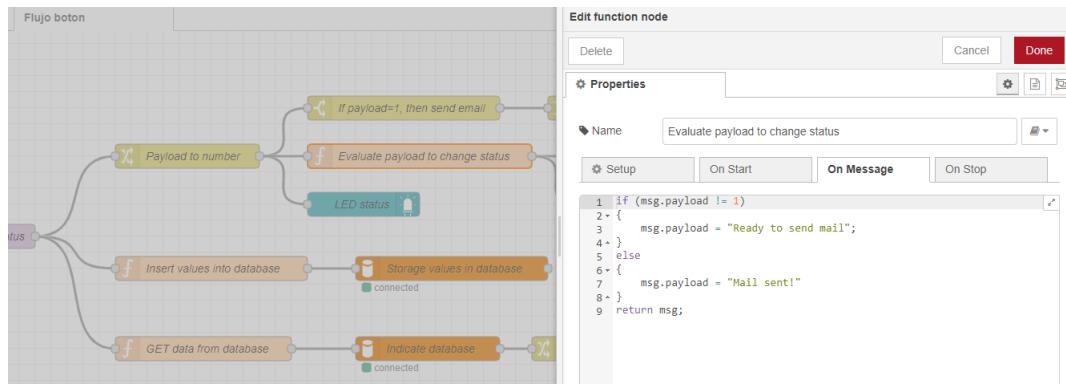
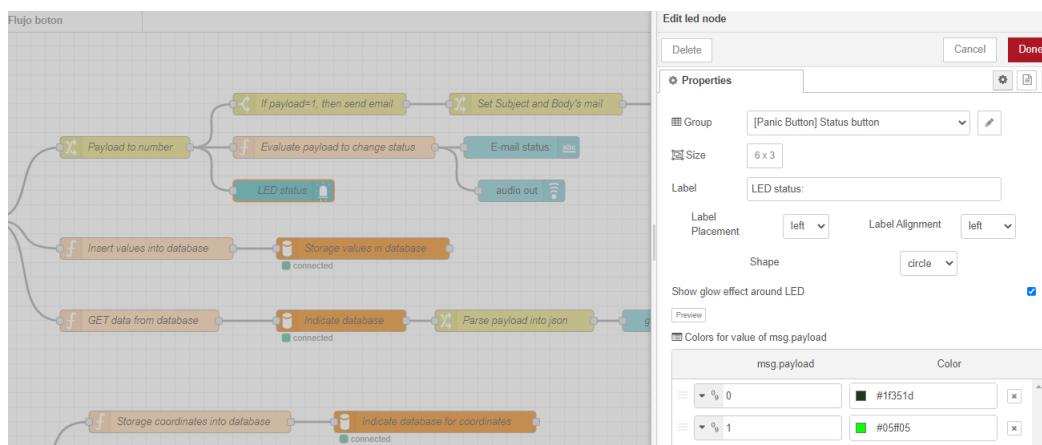
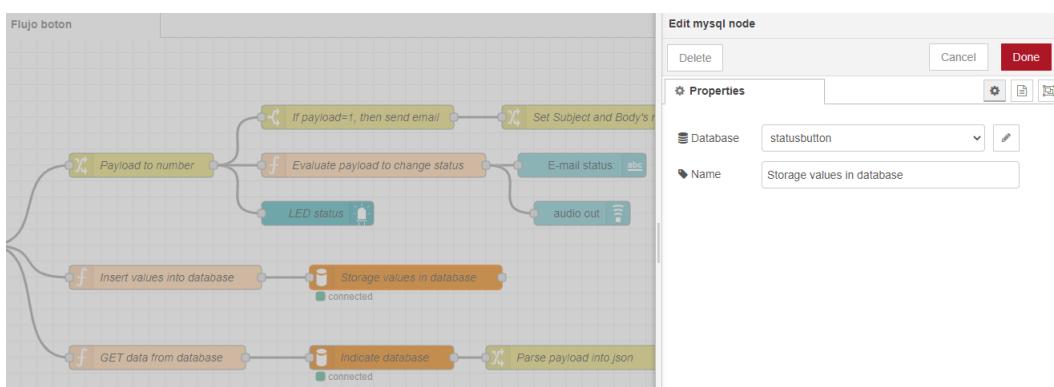


Figura 7.24: Nodo: Change node(payload to number)

7.4 Front-end

En cuanto al front-end, ya todo esta configurado y listo para usarse. Tan solo hay que copiar los archivos como se dijo en la carpeta `htdocs` y poder acceder a ellos desde `localhost`. Se tendrá como pagina principal `http://localhost/panicbuttonserver/`. Se puede usar un editor de texto para poder cambiar textos dentro de cada archivo CSS. Hay algunos apartados que no funcionan como configuración o recuperar cuenta.

Figura 7.25: Nodo: function node(*insert values into database*)Figura 7.26: Nodo: function node(*GET data from database*)Figura 7.27: Nodo: switch node(*If payload=1, then send email*)

Figura 7.28: Nodo: function node(*Evaluate payload to change status*)Figura 7.29: Nodo: led node(*LED status*)Figura 7.30: Nodo: mysql node(*Storage values in database*)

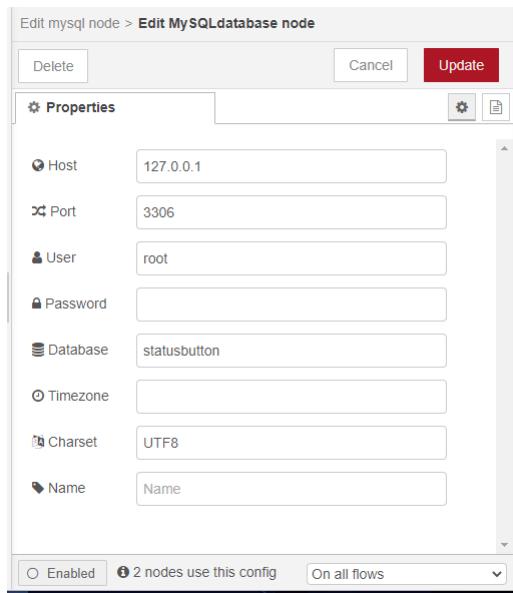
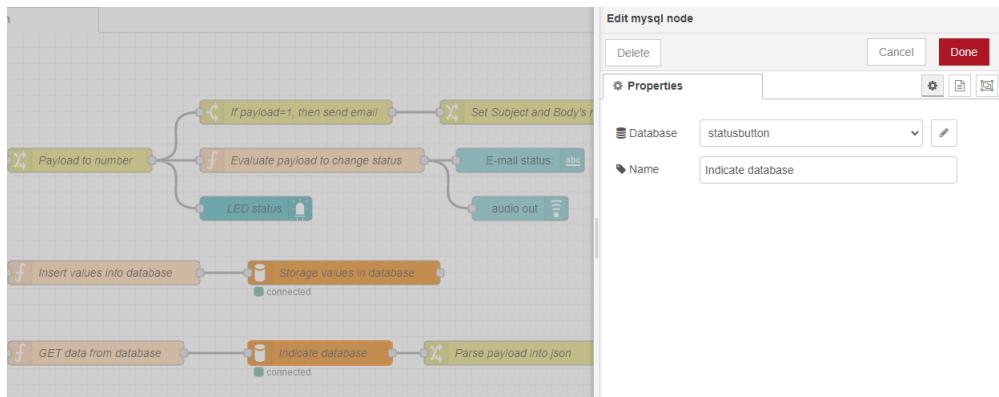
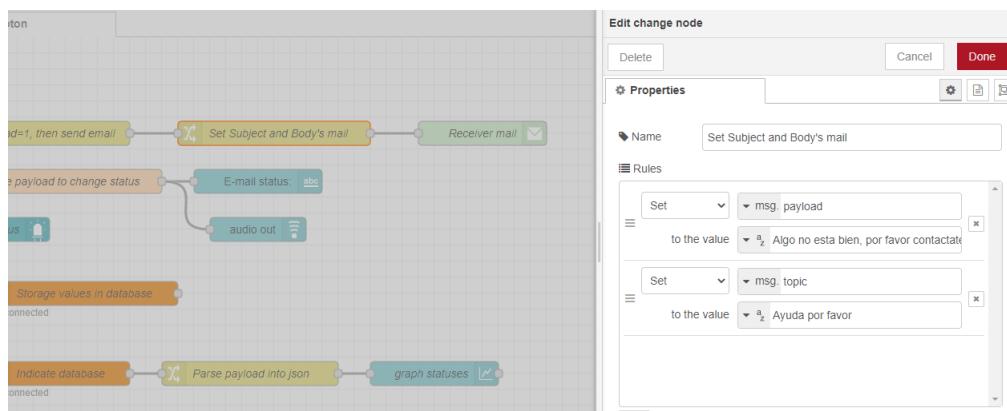
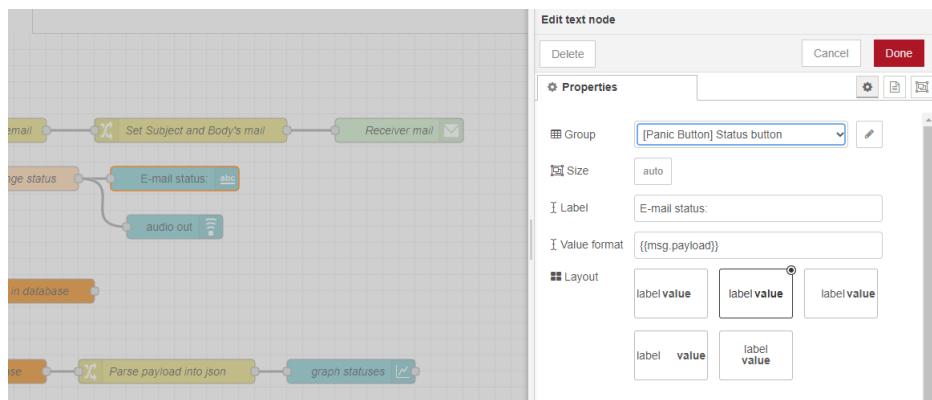
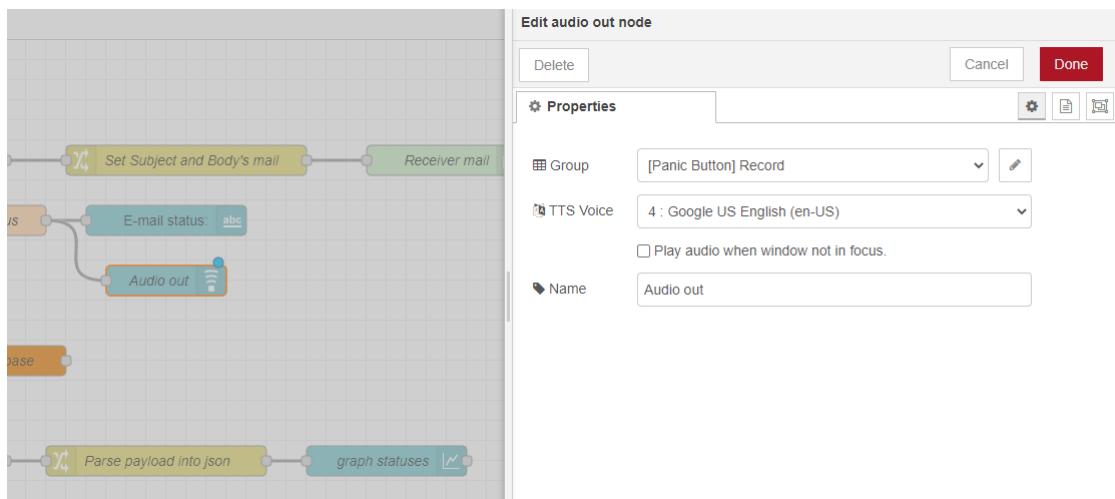
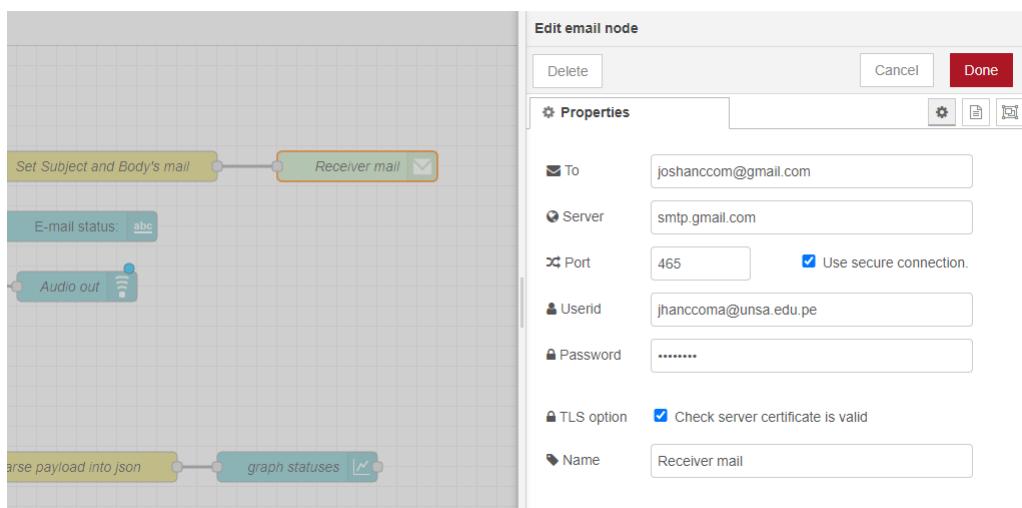


Figura 7.31: Configuración del database: statusbutton

Figura 7.32: Nodo: mysql node(*Indicate database*)Figura 7.33: Nodo: change node(*Set Subject and Body's mail*)

Figura 7.34: Nodo: text node(*E-mail status*)Figura 7.35: Nodo: audio out node(*Audio out*)Figura 7.36: Nodo: email node(*Receiver mail*), el password debe ser del correo desde donde se va enviar el correo.

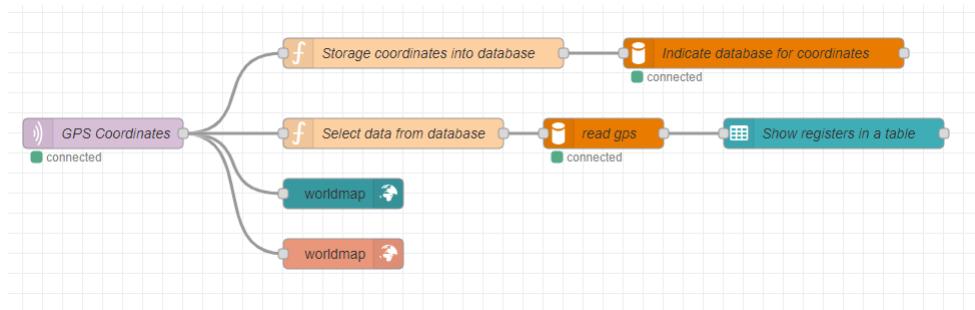


Figura 7.37: Flujo de datos: coordenadas gps

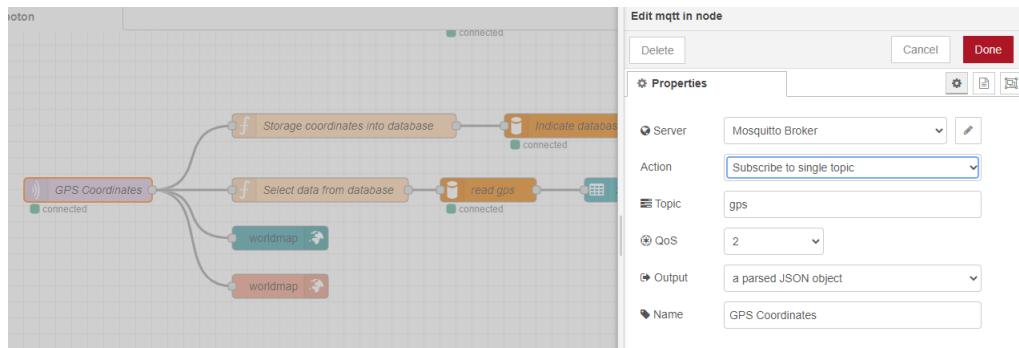
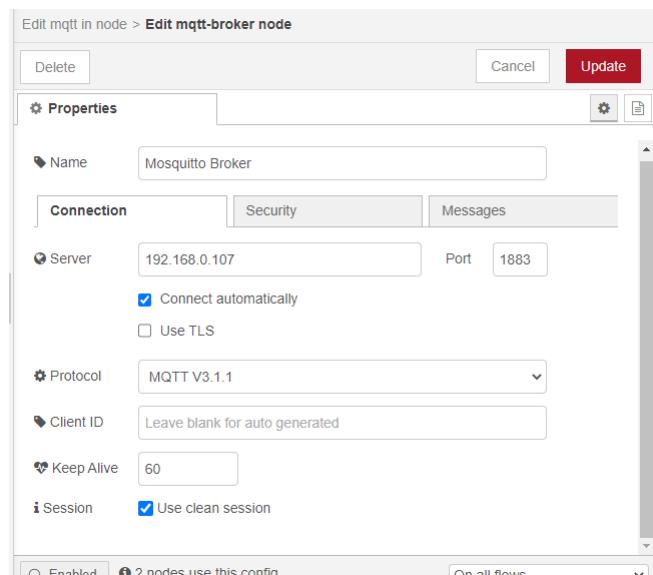
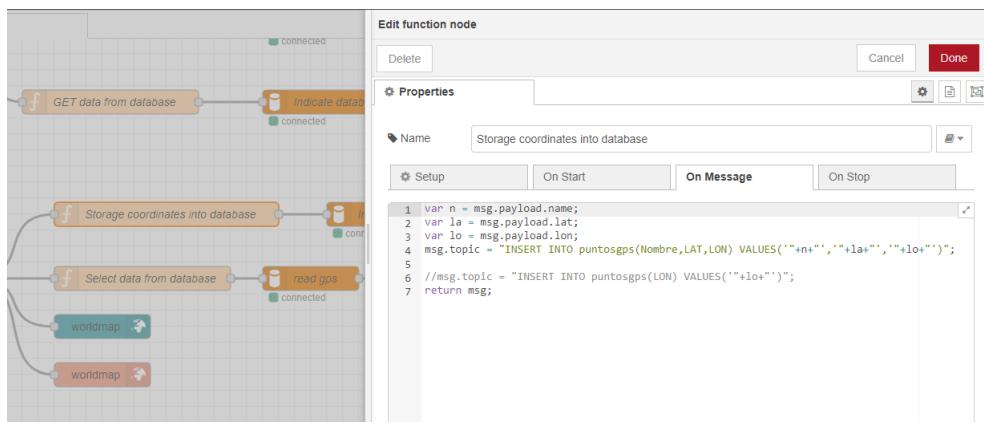
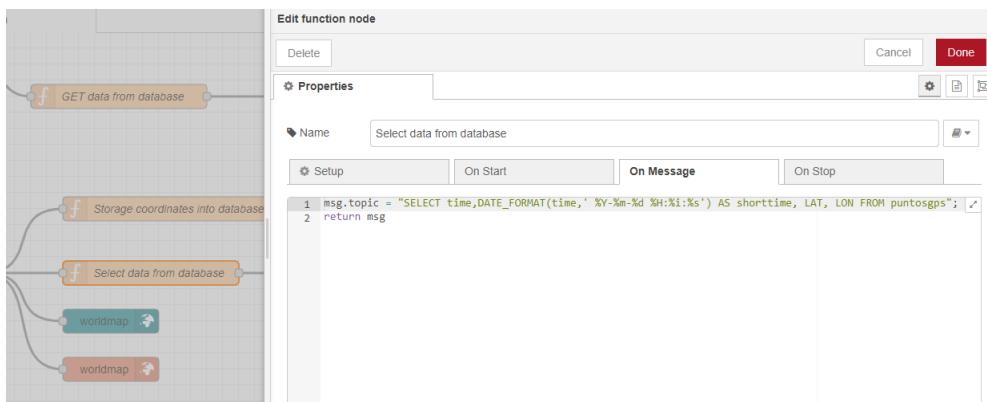
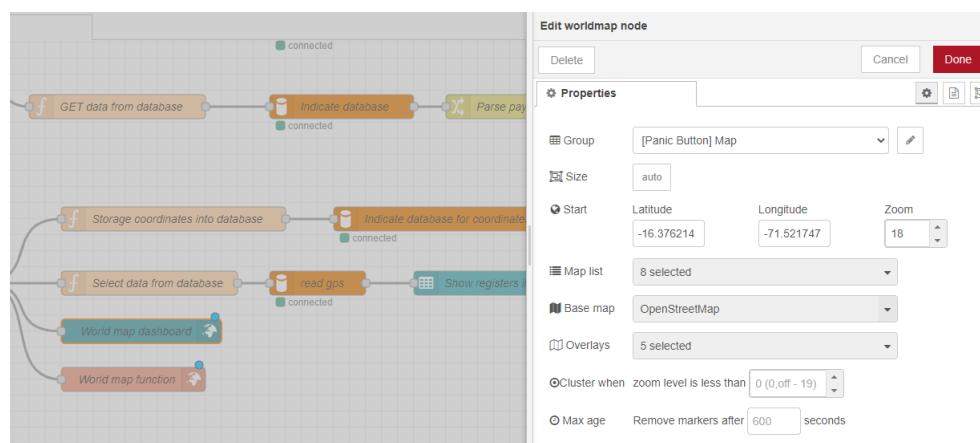
Figura 7.38: Nodo: mqtt node(*SGPS Coordinates*)

Figura 7.39: Mosquitto broker, se necesario para la figura 7.23

Figura 7.40: Nodo: function node(*Storage coordinates into database*)Figura 7.41: Nodo: function node(*Select data from database*)Figura 7.42: Nodo: worldmap node(*World map dashboard*)

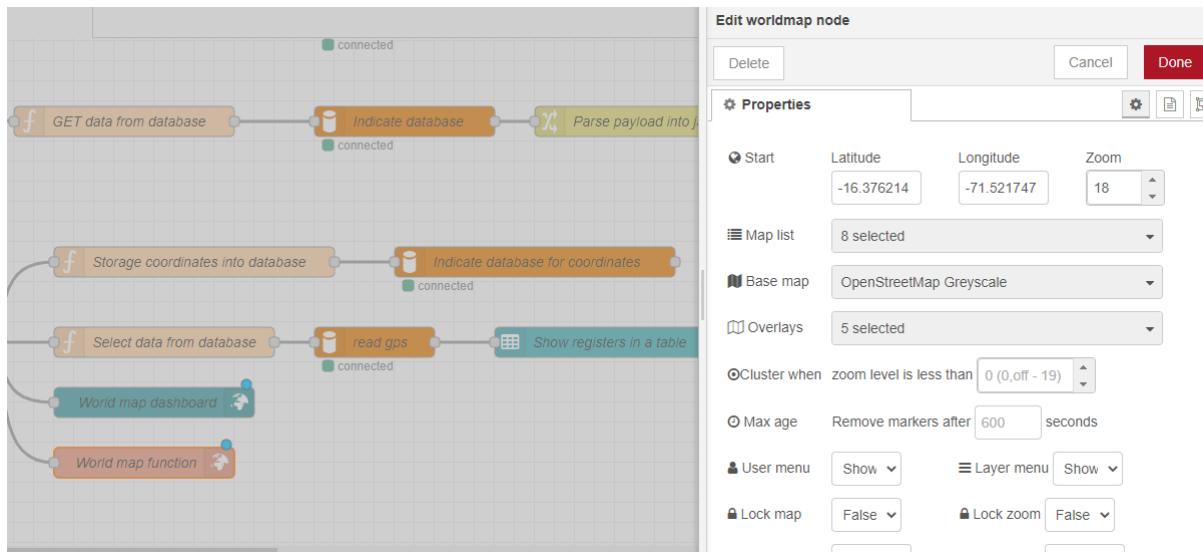
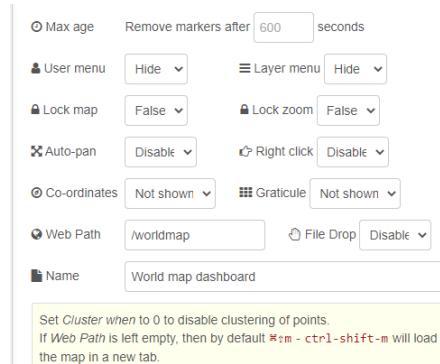
Figura 7.43: Nodo: worldmap node(*World map function*)

Figura 7.44: Nodo: worldmap node propiedades para la figura 7.42

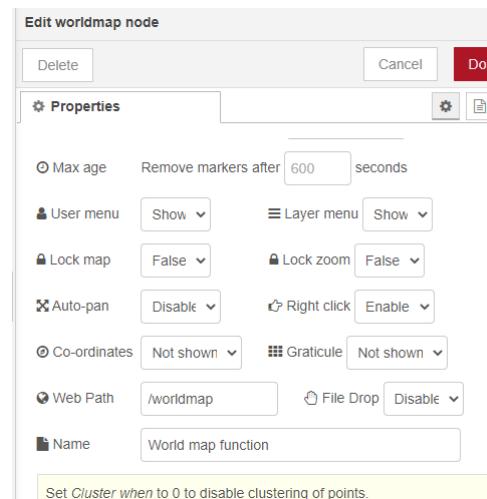
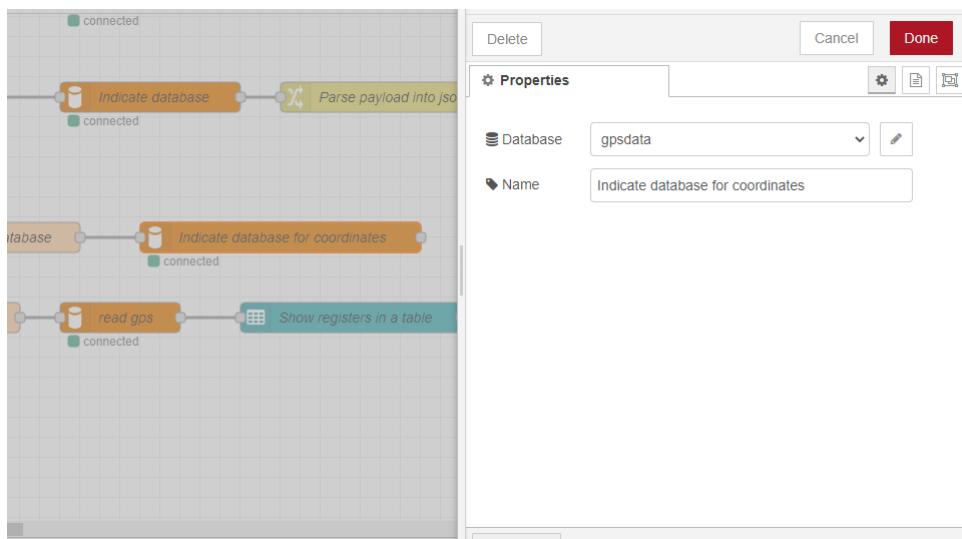
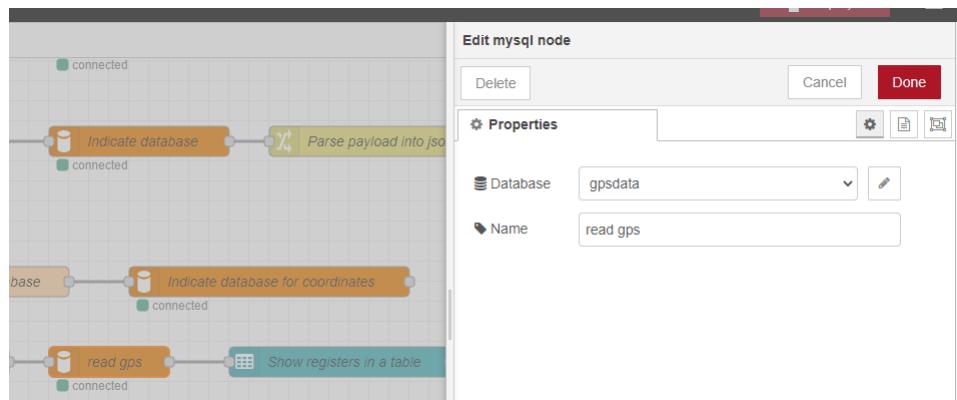


Figura 7.45: Nodo: worldmap node propiedades para la figura 7.43

Figura 7.46: Nodo: mysql node(*Indicate database for coordinates*)Figura 7.47: Nodo: mysql node(*read gps*)

A detailed view of the 'Edit MySQLdatabase node' dialog. It includes fields for Host (127.0.0.1), Port (3306), User (root), Password (empty), Database (gpsdata), Timezone (empty), Charset (UTF8), and a Name field (empty). There are 'Delete', 'Cancel', and 'Update' buttons at the top right, and a 'Properties' tab with a gear icon and a save icon.

Figura 7.48: Nodo:edit mysql node para gps

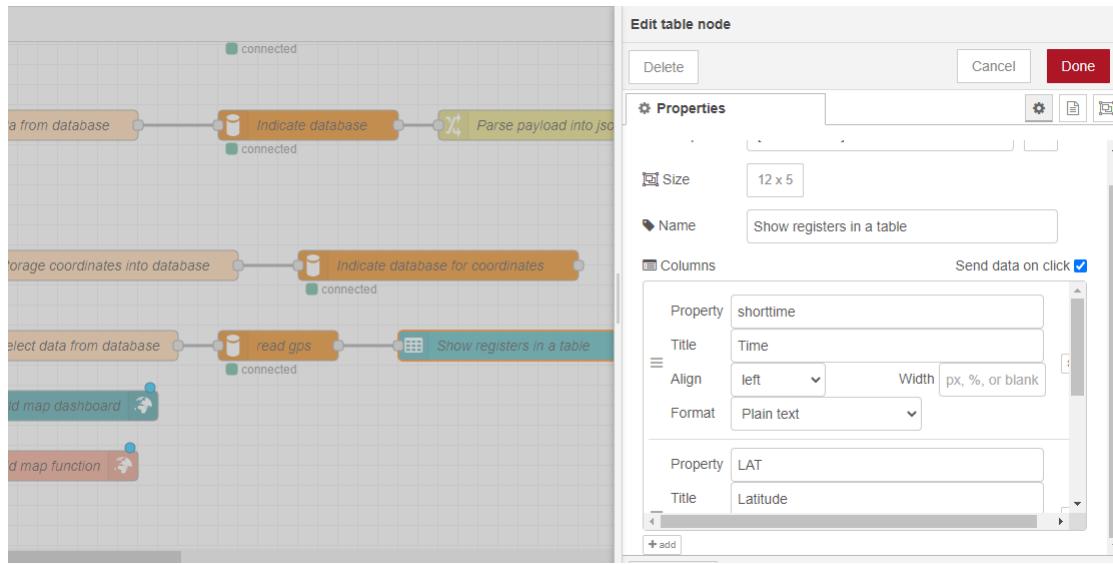


Figura 7.49: Nodo:edit table node para gps

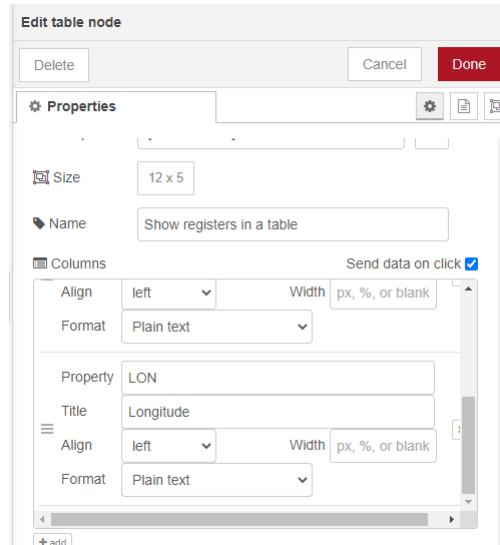


Figura 7.50: Nodo:edit table node para gps. Parte 2

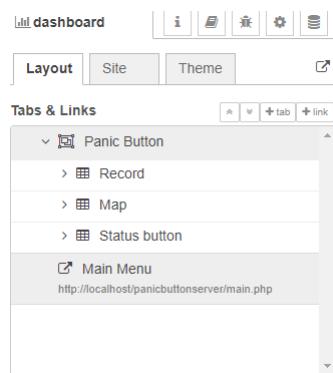


Figura 7.51: Resultado final de la organización del dashboard.

IV Telecommunicaciones 2

8	Unidad I	137
8.1	Theorems	
8.2	Definitions	
8.3	Notations	
8.4	Remarks	
8.5	Corollaries	
8.6	Propositions	
8.7	Examples	
8.8	Exercises	
8.9	Problems	
8.10	Vocabulary	
8.11	Table	
8.12	Figure	
9	Unidad 2	141
9.1	Modulación por ancho de pulso-PWM	

8. Unidad I

8.1 Theorems

This is an example of theorems.

8.1.1 Several equations

This is a theorem consisting of several equations.

Theorem 8.1.1 — Name of the theorem. In $E = \mathbb{R}^n$ all norms are equivalent. It has the properties:

$$|||\mathbf{x}|| - ||\mathbf{y}||| \leq ||\mathbf{x} - \mathbf{y}|| \quad (8.1)$$

$$||\sum_{i=1}^n \mathbf{x}_i|| \leq \sum_{i=1}^n ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \quad (8.2)$$

8.1.2 Single Line

This is a theorem consisting of just one line.

Theorem 8.1.2 A set $\mathcal{D}(G)$ is dense in $L^2(G)$, $|\cdot|_0$.

8.2 Definitions

This is an example of a definition. A definition could be mathematical or it could define a concept.

Definición 8.2.1 — Definition name. Given a vector space E , a norm on E is an application,

denoted $||\cdot||$, E in $\mathbb{R}^+ = [0, +\infty[$ such that:

$$||\mathbf{x}|| = 0 \Rightarrow \mathbf{x} = \mathbf{0} \quad (8.3)$$

$$||\lambda \mathbf{x}|| = |\lambda| \cdot ||\mathbf{x}|| \quad (8.4)$$

$$||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \quad (8.5)$$

8.3 Notations

Notación 8.1. Given an open subset G of \mathbb{R}^n , the set of functions φ are:

1. Bounded support G ;
2. Infinitely differentiable;

a vector space is denoted by $\mathcal{D}(G)$.

8.4 Remarks

This is an example of a remark.



The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

8.5 Corollaries

This is an example of a corollary.

Corolario 8.5.1 — Corollary name. The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

8.6 Propositions

This is an example of propositions.

8.6.1 Several equations

Proposición 8.6.1 — Proposition name. It has the properties:

$$|||\mathbf{x}|| - ||\mathbf{y}||| \leq ||\mathbf{x} - \mathbf{y}|| \quad (8.6)$$

$$||\sum_{i=1}^n \mathbf{x}_i|| \leq \sum_{i=1}^n ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \quad (8.7)$$

8.6.2 Single Line

Proposición 8.6.2 Let $f, g \in L^2(G)$; if $\forall \varphi \in \mathcal{D}(G)$, $(f, \varphi)_0 = (g, \varphi)_0$ then $f = g$.

8.7 Examples

This is an example of examples.

8.7.1 Equation and Text

■ **Ejemplo 8.1** Let $G = \{x \in \mathbb{R}^2 : |x| < 3\}$ and denoted by: $x^0 = (1, 1)$; consider the function:

$$f(x) = \begin{cases} e^{|x|} & \text{si } |x - x^0| \leq 1/2 \\ 0 & \text{si } |x - x^0| > 1/2 \end{cases} \quad (8.8)$$

The function f has bounded support, we can take $A = \{x \in \mathbb{R}^2 : |x - x^0| \leq 1/2 + \varepsilon\}$ for all $\varepsilon \in]0; 5/2 - \sqrt{2}[$. ■

8.7.2 Paragraph of Text

■ **Ejemplo 8.2 — Example name.** Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. ■

8.8 Exercises

This is an example of an exercise.

Ejercicio 8.1 This is a good place to ask a question to test learning progress or further cement ideas into students' minds. ■

8.9 Problems

Problema 8.1 What is the average airspeed velocity of an unladen swallow?

8.10 Vocabulary

Vocabulario 8.1 — Word. Definition of word.

8.11 Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Cuadro 8.1: Table caption

Referencing Table 8.1 in-text automatically.

8.12 Figure

Referencing Figure 8.1 in-text automatically.

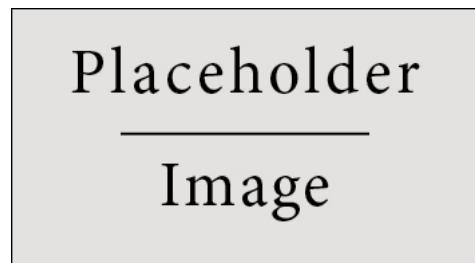


Figura 8.1: Figure caption

9. Unidad 2

9.1 Modulación por ancho de pulso-PWM

PWM es una técnica que se usa para transmitir señales analógicas cuya señal portadora será digital. En esta técnica se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

Definición 9.1.1 — Duty Cicle. El ciclo de trabajo (*duty cycle*) de una señal periódica es el ancho de su parte positiva, en relación con el período. Está expresado en porcentaje, por tanto, un duty cycle de 10 % indica que está 10 de 100 a nivel alto.

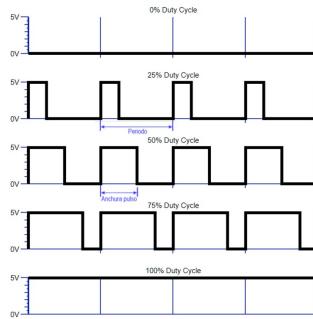
$$\text{Duty cycle} = \frac{t}{T} \quad (9.1)$$

Donde:

t: Tiempo en parte positiva.

T: Periodo, tiempo total.

Básicamente, consiste en activar una salida digital durante un tiempo y mantenerla apagada durante el resto, generando así pulsos positivos que se repiten de manera constante. Por tanto, la frecuencia es constante (es decir, el tiempo entre disparo de pulsos), mientras que se hace variar la anchura del pulso, el duty cycle. El promedio de esta tensión de salida, a lo largo del tiempo, será igual al valor analógico deseado. Lo que caracteriza a PWM es la capacidad de controlar los pulsos, controlando el voltaje de la señal. Para ello la señal moduladora (una señal analógica) hará modificaciones sobre la señal portadora (mayormente una señal *sawtooth*) con la finalidad de, en vez de tener una amplitud infinita solo tendrá valores entre 5v-0v (puede ser modifiable). Si lo ponemos en un ejemplo: podríamos controlar la velocidad de un moto apagándolo y prendiéndolo tan rápido que teóricamente sabríamos que lo estamos apagando pero experimentalmente sería imperceptible.

Figura 9.1: Diferentes *Duty Cycles*

Definición 9.1.2 — Voltaje promedio de PWM. El voltaje promedio de una señal modulada por PWM podrá ser calculada por:

$$V_p = V_{cc} \cdot \frac{DC}{100} \quad (9.2)$$

Donde:

V_{cc} : La entrada de alimentación de un dispositivo.(Voltios)

DC: Duty cycle.(porcentaje)

V_p : Voltaje promedio.

Ejercicio 9.1 Se desea un voltaje promedio de 1V con voltaje de entrada de 5V, calcular el tiempo que la señal pwm estará apagada.

Solución:

Nos pide el tiempo, aunque no nos da el tiempo para calcularlo en segundos(usingo la ecuación 9.1), pero podemos hallarlo en porcentaje. Por ello, usando la ecuación 9.2 reemplazamos nuestros datos, tenemos que hallar DC:

$$\begin{aligned} 1V &= 5V \cdot \frac{DC}{100} \\ \frac{1V \cdot 100}{5V} &= DC \\ DC &= 20 \end{aligned} \quad (9.3)$$

El 20% hallado es el duty cycle, es el tiempo que valdrá 5V, nos pide el tiempo que estará apagada o donde valdrá 0V, el 80% restante del periodo total. ■

V

Campos electromagnéticos

10. Introducción y memoria

10.0.1 Repaso análisis vectorial

Sistema de coordenadas cartesiano(ortogonal)

Definimos un punto arbitrario, (x,y) teniendo las respectivas consideraciones:

- $x & y$ positivas para derecha y arriba respectivamente.
- $x & y$ negativas para la izquierda y abajo respectivamente.

Coordenadas polares

Un punto arbitrario se define mediante las coordenadas polares planas (r,θ) , donde:

- r es la longitud de la línea que une el origen con el punto
- θ es el ángulo entre dicha línea y un eje fijo (normalmente el x)

Es común asumir que θ es medida en sentido contrario a las agujas del reloj y normalmente respecto al eje x positivo.

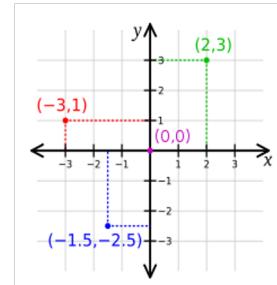


Figura 10.1: Sistema de coordenadas rectangulares

De polares a cartesianas

$$x = r \times \cos(\theta)$$

$$y = r \times \sin(\theta)$$

De cartesianas a polares

$$\tan(\theta) = \frac{y}{x}$$

$$r = \sqrt{x^2 + y^2}$$

- **Ejemplo 10.1** Expresar el campo vectorial dado en coordenadas cartesianas:

$$A(x,y,z) = \frac{(2x^2 + y^2)(-y\hat{a}_x + x\hat{a}_y)}{(1 + x^2 + y^2)(x^2 + y^2)}$$

■

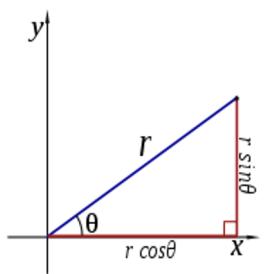


Figura 10.2: Sistema de coordenadas polares

VI

Sistemas de telefonía

11 Telefonía 149

- 11.1 Evolución de la telefonía
- 11.2 Conmutadores
- 11.3 Repaso: Señales analógicas y digitales
- 11.4 Multiplexación
- 11.5 Técnicas de conmutación
- 11.6 Red de telefonía pública
- 11.7 Señalización telefónica
- 11.8 Planta interna y externa

12 Herramientas para el análisis telefónico 167

- 12.1 Tráfico Telefónico
- 12.2 Medidas de tráfico telefónico
- 12.3 Tipos de tráfico
- 12.4 Accesibilidad
- 12.5 *Grade of service(GoS)*
- 12.6 Diseño de redes
- 12.7 Encaminamiento de tráfico
- 12.8 Erlang-B

13 Telefonía IP 187

- 13.1 Repaso: TCP/IP
- 13.2 VoIP
- 13.3 Digitización de la voz
- 13.4 Componentes de VoIP
- 13.5 Tecnología IP
- 13.6 Vocabulary



11. Telefonía

11.1 Evolución de la telefonía

La era de las telecomunicaciones se inicia en el siglo XIX con la invención del telégrafo en los años 1830's.

El teléfono se patentó en 1876 por Graham Bell, y con ello fue posible transportar la voz humana de un lugar a otro. Inicialmente las comunicaciones eran punto a punto, los dispositivos terminales debían conectarse directamente. Ya se torna complicado cuando aparecen más teléfonos



Figura 11.1: Enlace cableado de dos teléfonos

en la red, por ejemplo con 4 teléfonos el cableado se elevaba considerablemente:

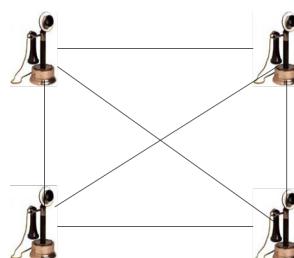


Figura 11.2: Cableado con 4 teléfonos

La ecuación que describe este comportamiento y la cantidad necesaria de cableado para poder establecer la conexión entre todos los equipos es el siguiente:

Theorem 11.1.1 — Cantidad de enlaces. Sea N el número de dispositivos a conectar, se crearán E enlaces para que todos estén interconectados:

$$E = \frac{N \times (N - 1)}{2} \quad (11.1)$$

11.2 Conmutadores

Debido a este problema es que se crean los conmutadores: “Comutación es el proceso de crear enlaces virtuales entre los equipos para permitirles la comunicación”

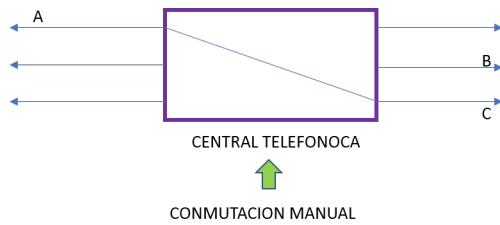


Figura 11.3: Principio básico de los conmutadores

Al principio estos eran manuales, tal cual vemos en las películas: una señorita operadora nos recibe y dándole al lugar donde queremos llamar nos conecta; sin embargo ahora se ha automatizado permitiendo menos personal, eficiencia y rapidez.//Cuando juntamos ambos temas vistos se produce un sistema de telefonía como se ve en la imagen, cada enlace entre cada elemento recibe un nombre en específico.

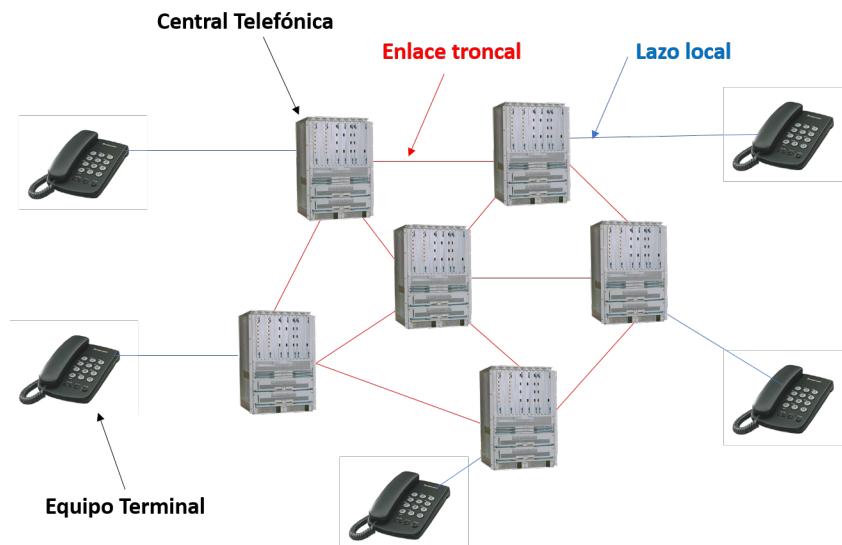


Figura 11.4: Principio básico de los conmutadores

11.3 Repaso: Señales analógicas y digitales

11.3.1 Ventajas y desventajas de las señales analógicas

- El procesamiento de las señales analógicas es relativamente más simple que las señales digitales, los circuitos electrónicos involucrados en la interpretación de la señal son más

Señal analógica	Señal digital
Varían su amplitud continuamente	Su amplitud es constante en un lapso de tiempo
Varían continuamente, todo el tiempo cambian	Cambian después de un periodo de tiempo
Pueden representar infinitos valores	Representan valores discretos (finitos)
Son muy afectadas por el ruido (ruido es todo aquello que perturba a la señal)	Tienen mayor tolerancia al ruido
No se pueden regenerar	Es posible regenerarlas (por lo tanto las señales digitales pueden viajar a mayores distancias con regeneradores intermedios)

Cuadro 11.1: Cuadro comparativo de las características principales entre las señales analógicas y señales digitales

simples y por lo tanto de menor coste.

- La transmisión de la señal es ligeramente más simple, aunque la eficacia dependerá del alcance de la señal en el medio.
- La principal desventaja es la eficacia y las posibles interferencias que se puedan generar en la emisión y recepción de la señal.

11.3.2 Ventajas y desventajas de las señales digitales

- Ante la atenuación, la señal digital puede ser amplificada y al mismo tiempo reconstruida gracias a los sistemas de regeneración de señales.
- La señal digital permite la multigeneración infinita sin pérdidas de calidad.
- Como principal desventaja se puede comentar que el procesamiento es más complejo, aunque a medida que la tecnología avanza este diferencial se reduce cada vez más.
- La señal digital requiere mayor ancho de banda para ser transmitida que la analógica.
- Se necesita una conversión analógica-digital previa y una decodificación posterior, en el momento de la recepción.
- La transmisión de señales digitales requiere una sincronización precisa entre los tiempos del reloj de transmisor, con respecto a los del receptor. Un desfase cambia la señal recibida con respecto a la que fue transmitida.

11.4 Multiplexación

La multiplexación es el proceso de transmisión simultánea de dos o más señales individuales a través de un único canal de comunicación, ya sea por cable o inalámbrico.

Es una técnica que permite la transmisión de señales de información y cuya idea consiste en ocupar un canal (normalmente de gran capacidad) de transmisión a partir de distintas fuentes.

11.4.1 ¿Para qué sirve?

Aumenta el número de canales de comunicación para poder transmitir más información. En las telecomunicaciones se usa la multiplexación para dividir las señales en el medio por el que vayan a viajar dentro del espectro electromagnético. Por ejemplo para poder enviar varios canales de voz por un mismo medio podemos utilizar multiplexación en el tiempo o en la frecuencia. En otras palabras: es compartir la capacidad de transmisión de datos sobre un mismo enlace para aumentar la eficiencia, además sirve para minimizar la cantidad de líneas físicas requeridas y maximizar el uso del ancho de banda de los medios.

El mayor uso de la multiplexación se da en el sistema telefónico, donde millones de llamadas se multiplexan en cables, líneas de fibra óptica de larga distancia, satélites y canales inalámbricos.

La multiplexación aumenta la capacidad del operador telefónico para gestionar más llamadas, minimizando los costos del sistema y el uso del espectro.



Figura 11.5: Multiplexación: esquema

Multiplexor	Demultiplexor
Combina los datos de las líneas de entrada. Los transmite a través de un enlace de mayor capacidad.	Acepta la cadena multiplexada. Separa los datos de acuerdo al canal del que provienen y los distribuye a la línea apropiada de salida.

Cuadro 11.2: Función de cada parte de la multiplexación

11.4.2 Multiplexación por división de tiempo o TDM (*time-division multiplexing*)

Consiste en asignar a cada usuario, durante unas determinadas ranuras o fracciones de tiempo", la totalidad del ancho de banda disponible.

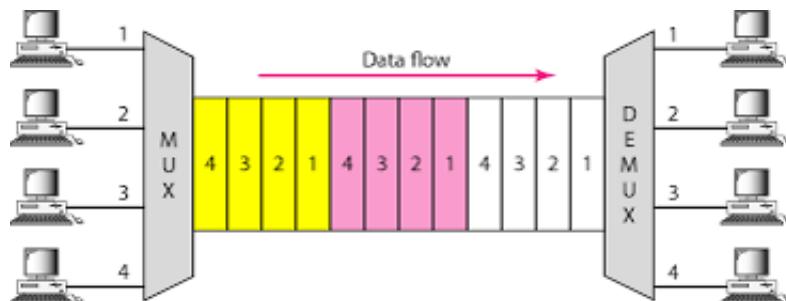


Figura 11.6: TDM exemplificación

11.4.3 Multiplexación por división de frecuencia o FDM (*frequency-division multiplexing*)

Esta técnica que consiste en dividir mediante filtros el espectro de frecuencias del canal de transmisión y desplazar la señal a transmitir dentro del margen del espectro correspondiente mediante modulaciones, de tal forma que cada usuario tiene posesión exclusiva de su banda de frecuencias. Algunas características del FDM:

- El ancho de banda del medio debe ser mayor que le ancho de banda de la señal transmitida. Capacidad de transmisión de varias señales a la vez.
- La señal lógica trasmisita a través del medio es analógica. La señal recibida puede ser analógica o digital.
- Para la comunicación análoga el ruido tiene menos efecto.

Ventajas	Desventajas
<p>El uso de la capacidad es alto. Combina los datos de las líneas de entrada.</p> <p>Cada uno para ampliar el número de usuarios en un sistema en un coste bajo.</p>	<p>La sensibilidad frente a otro problema de usuario es alta.</p> <p>El coste inicial es alto y la complejidad técnica es más</p>

Cuadro 11.3: TDM: Ventajas y desventajas

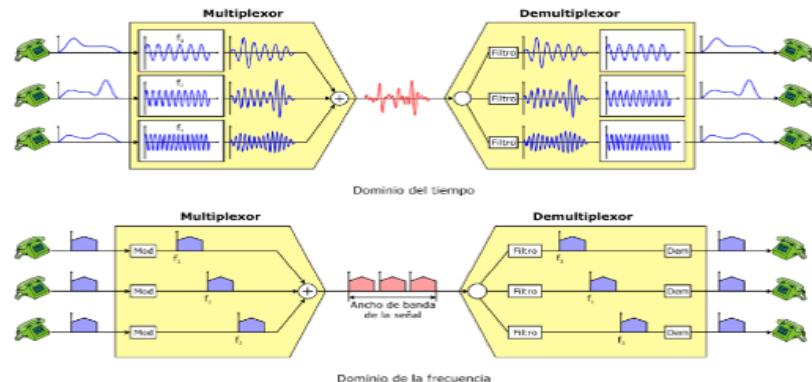


Figura 11.7: FDM exemplificación

11.4.4 Multiplexación por división de longitud de onda o WDM (*wavelength-division multiplexing*)

Se quieren combinar múltiples haces de luz dentro de una única luz en el multiplexor, Hacer a operación inversa en el demultiplexor. Combinar y dividir haces de luz se resuelve fácilmente mediante un prisma. Un prisma curva un rayo de luz basándose en el ángulo de incidencia y la frecuencia

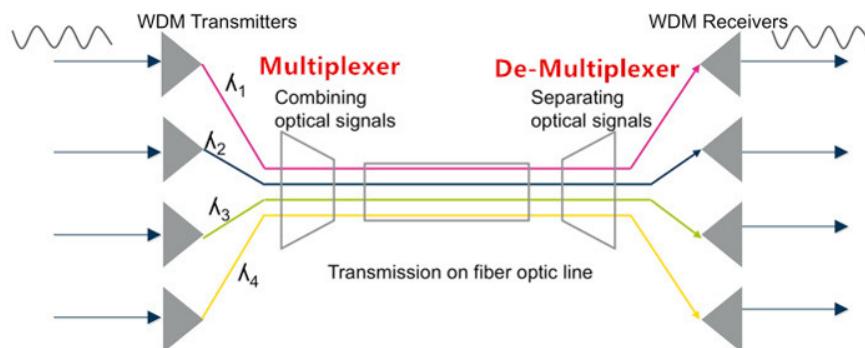


Figura 11.8: WDM exemplificación

11.4.5 Multiplexación por Estadística por División de Tiempo o STDM(*Statistical time-division multiplexing*)

Es un canal de comunicaciones que se divide en un número arbitrario de las secuencias de datos de velocidad de transmisión de bites variables de los canales digitales.

Trabaja con señales digitales transmite los datos de los canales que en cada momento tengan información disponible para transmitir, lo que mejora la eficiencia del canal de transmisión.

Ventajas	Desventajas
<p>El usuario puede ser añadido al sistema, simplemente añadiendo otro par de modulador de transmisor y receptor.</p> <p>El sistema de FDM apoya el flujo de dúplex total de información que es requerido por la mayor parte de la aplicación.</p>	<p>En el sistema FDM, el coste inicial es alto y un problema para un usuario puede afectar a veces a otros.</p> <p>En el sistema FDM, cada usuario requiere una frecuencia de portador precisa.</p>

Cuadro 11.4: TDM: Ventajas y desventajas

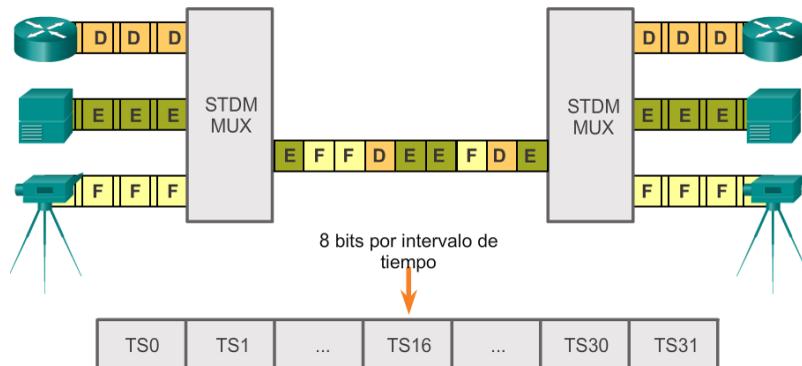


Figura 11.9: STDM ejemplificación

11.5 Técnicas de conmutación

11.5.1 Conmutación por circuitos

Esta multiplexación implica un camino dedicado entre el transmisor y el receptor de datos mediante una TDM.

Esta conmutación posee 3 fases:

1. Inicio de conexión
2. Transferencia de datos
3. Liberación de la conexión

Una vez que se establece la llamada a través de los conmutadores se puede llegar a la conclusión que la comunicación es bilateral entre dos teléfonos como lo propuso Graham Bell. El número de B

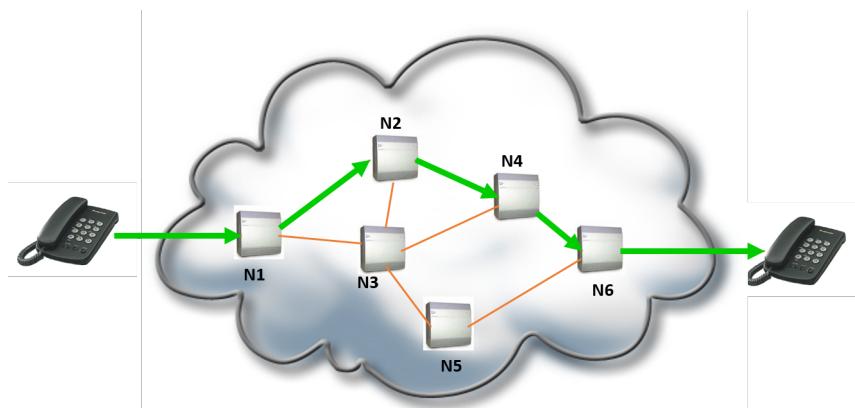


Figura 11.10: Comutación por circuitos

se transmite una sola vez, al inicio de la conexión, antes de la conversación. La conexión se realiza de extremo a extremo, luego pasa la información de usuario donde el retardo de la información casi 0 seg. El enlace es utilizado solo para paso de la información entre el usuario A y B, por lo tanto los nodos no requieren memoria para almacenar la información del usuario puesto que una vez que el enlace culmina esta listo para que sea usado por otro par de usuarios para una nueva conexión. Es importante resaltar que los nodos no realizan corrección de error.

11.5.2 Comutador por paquetes

A diferencia del anterior, esta forma no tiene una conexión directa entre ambos puntos, sino que la información viaja en paquetes, estos paquetes son independientes entre si; es por eso que la información completa sobre las direcciones de origen y destino se envían con el paquete.

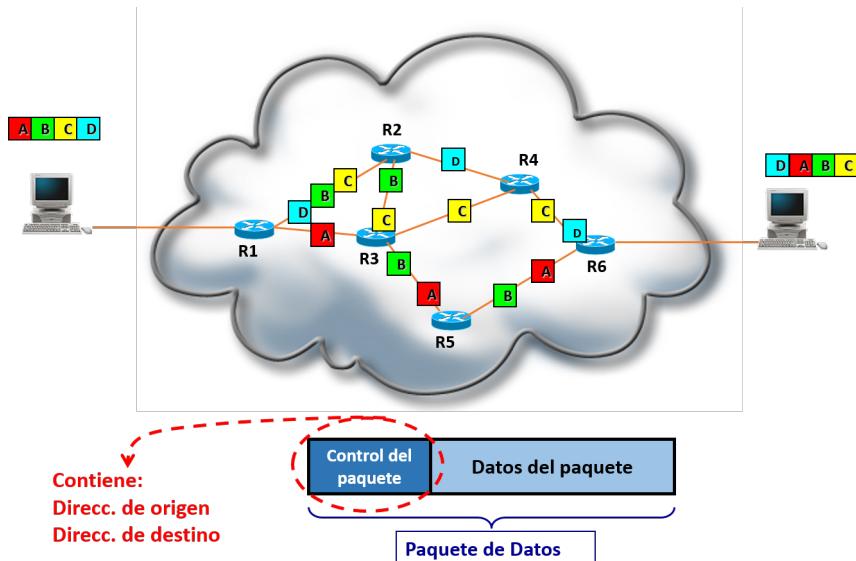


Figura 11.11: Comutación por paquetes

- El enlace es utilizado para varias comunicaciones.
- El número del terminal de destino se transmite en cada paquete.
- No necesariamente se realiza la conexión de extremo a extremo para transmitir la información.
- Utiliza protocolos. Ejm: TCP, UDP, IP
- Retardo de la información considerable

Resumen comparativo

Carácteristicas	Circuitos	Paquetes
Tiempo para establecer el trayecto.	Según la señalización utilizada.	No existe fase de establecimiento.
Retardo de transmisión End to End.	Despreciable.	Existe en toda comunicación, para tiempo real es negativo
Asignación de circuitos.	Único y exclusivo para cada comunicación	Compartido por otras comunicaciones simultaneas.
Identificación del destino.	Sólo en la fase de establecimiento.	Se incluye un identificador en cada paquete
Necesidad de almacenar en la red.	No es necesario.	Si, en los nodos de la red.
Flexibilidad de la red.	Encaminamiento alternativo.	Gran flexibilidad.

Cuadro 11.5: Cuadro comparativo entre conmutación por circuito y por paquete

11.6 Red de telefonía pública

11.6.1 Red de telefonía pública conmutada-PSTN

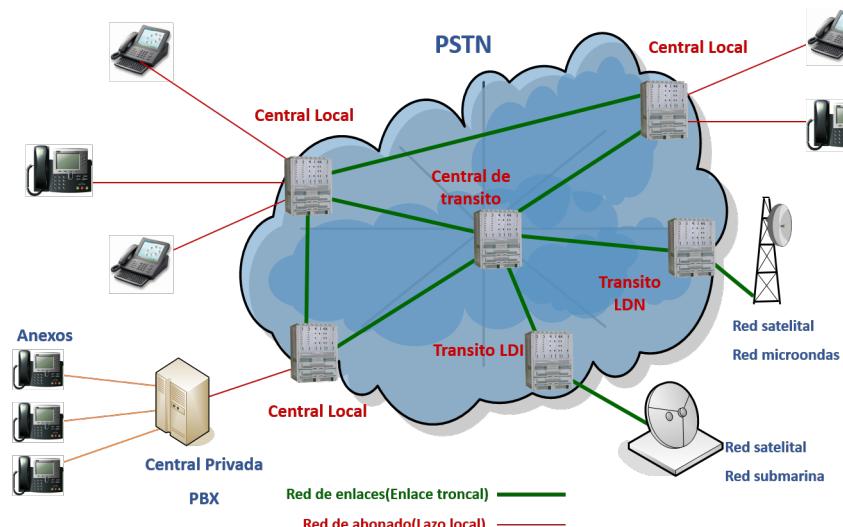


Figura 11.12: Red de telefonía

Compuesta por:

- El equipo terminal: normalmente un teléfono(Analógicos / Digitales).
- La central de conmutación: Que puede ser pública o privada (PBX)(Privada y Públicas)
- El bucle de abonado o usuario.(Lazo abonado)
- Las troncales de comunicación.(Troncales Analógicas y Digitales)

La Red Telefónica Pública Conmutada o **PSTN**, es una red pública basada en conmutación de circuitos tradicional, diseñada principalmente para la transmisión de voz en tiempo real. En un principio estaba basada únicamente en sistemas analógicos, POST¹.

¹(Plain Old Telephone Service) hace referencia al servicio teléfono tradicional conformado por dispositivos analógicos

Actualmente ha evolucionado a sistemas digitales e incluso basados en IP.

La PSTN, está formada por un conjunto de dispositivos de conmutación (centrales telefónicas) y enlaces (troncales / líneas de abonado) que transportan la voz de los abonados distribuidos en un área geográfica, optimizada para comunicaciones de voz en tiempo real.

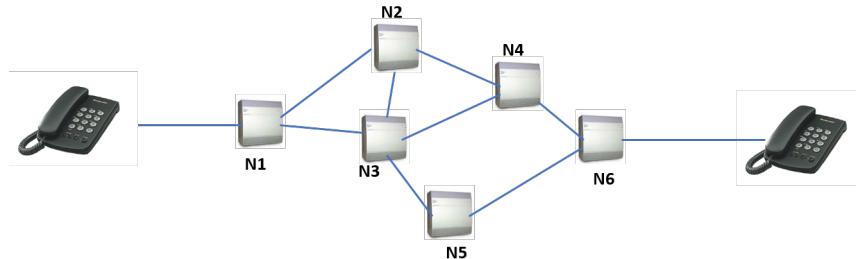


Figura 11.13: Red de telefonía pública commutada

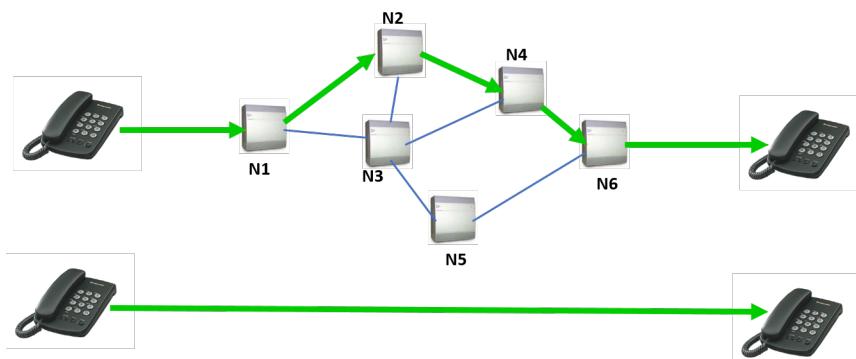


Figura 11.14: Apreciamos que llegamos al mismo punto inicial, enlace bidireccional

11.6.2 Características

- El bucle de abonado(usuario) es generalmente analógico.
- Ofrece a cada usuario un circuito (bucle de abonado) para señales analógicas con una banda base de 4KHz para cada conversación entre dos dispositivos terminales.
- Los enlaces troncales son en la mayoría digitales.
- Ancho de banda en los troncales de 64Kbps.
- Conexión dúplex circuit-switched.
- Movilidad limitada.

11.6.3 Componentes

Equipos Terminales

Son los dispositivo que le permite al usuario establecer una comunicación. Cada terminal es utilizado por los usuarios (abonados) para transmitir voz u otro tipo de datos a otros terminales que están conectados a la red. Ejemplos: teléfono, fax, modem.

Pueden ser analógicos o digitales, se conectan a las centrales telefónicas a través del lazo de abonado.

Los símbolos utilizados para representar diferentes tipos de aparatos y terminales telefónicos son los siguientes:

que utilizan el cableado de cobre.

Elemento	Símbolo
Aparato telefónico en general.	A simple rectangle with a curved top representing a handset.
Aparato telefónico con dispositivo de marcación.	Two rectangles side-by-side, one containing three dots and the other containing a grid of dots representing a dial pad.
Aparato telefónico con batería local.	A rectangle with a cross inside, representing internal battery power.

Figura 11.15: Símbolos representativos

Central telefónica

Es el nodo que establece, mantiene y termina las conexiones (llamadas) entre 2 o más usuarios. También se le conoce como equipo o central de conmutación. Se encarga de la señalización así como de facilitar la información sobre su progreso.



Figura 11.16: Diagrama del accionamiento de una central telefónica

Elemento	Símbolo
Dispositivo de conmutación (Forma 1).	Un cuadro que contiene una 'X' invertida.
Dispositivo de conmutación con líneas de entrada y de salida.	Un cuadro que contiene una 'X' invertida y tiene tres líneas de conexión saliendo de su lado derecho.
Dispositivo de conmutación (Forma 2).	Un cuadro que contiene una 'T' vertical.

Figura 11.17: Simbología de una central telefónica (conmutador)

Red de enlace

Esta formado por las troncales que enlazan las centrales de conmutación, pueden tener miles de kilómetros y transportan gran cantidad de llamadas multiplexadas.

Troncales

Interconectan centrales de conmutación telefónica, y llevan las señales de voz permitiendo la comunicación de los usuarios.

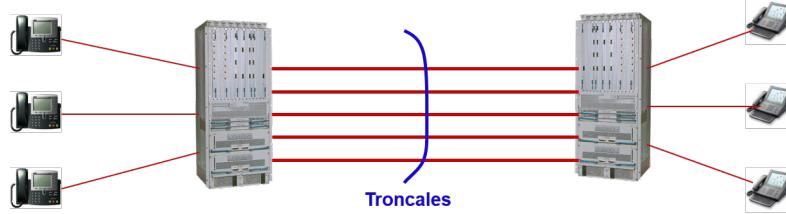


Figura 11.18: Diagrama iconográfico de las troncales

Lazo de abonado o local

El lazo de abonado interconecta a la central telefónica con el terminal del usuario. Presenta las siguientes características:

- Red analógica
- 6 a 7 Km de distancia (usuario a la central)
- Gran cantidad de cableado
- Red que transporta poco tráfico
- 2 hilos

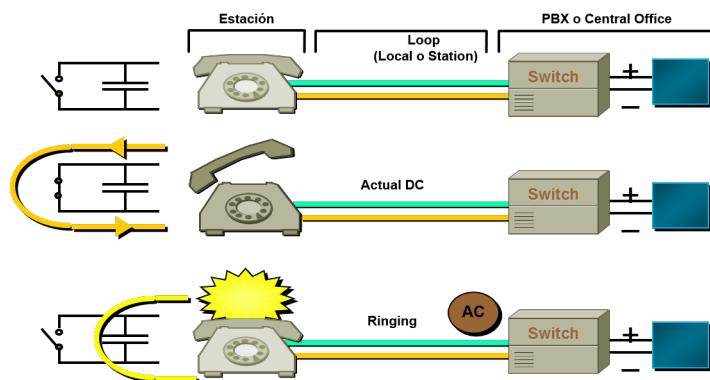


Figura 11.19: Lazo de abonado o bucle de abonado, va desde la central de comutación hasta el abonado.

El lazo de abonado transporta la voz telefónica dentro del rango de frecuencias de 300 a 3400 Hz.

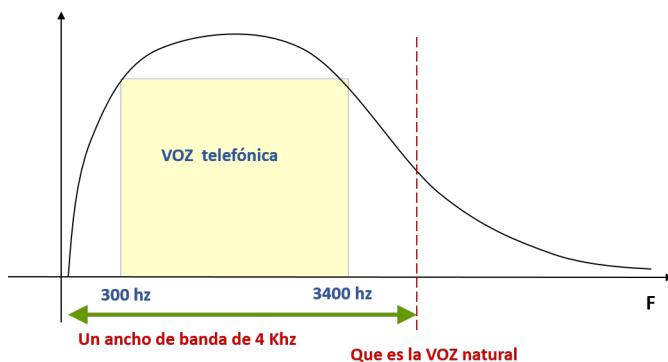


Figura 11.20: Rango de frecuencias

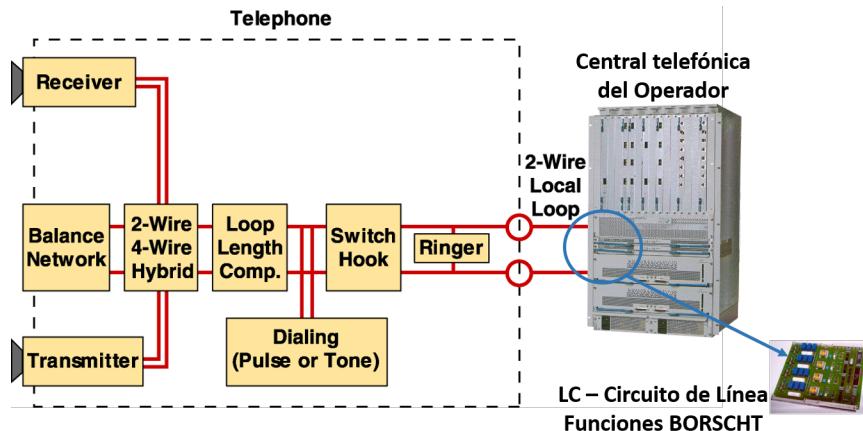


Figura 11.21: Diagrama de bloques de un teléfono analógico

SLIC y BORSCHT

Cada teléfono conectado a la central telefónica está provisto de un grupo de circuitos básicos que alimentan al teléfono y proporcionan todas las funciones básicas: llamada, tono de marcación y supervisión de la marcación. Estos circuitos colectivamente se llaman interfase del suscriptor o circuitos de interfase de línea del suscriptor (SLIC, subscriber line interface circuits).

La SLIC proporciona siete funciones básicas llamadas BORSCHT (acrónimo de las funciones batería, protección de sobrevoltaje, llamada, supervisión, codificación, bobina híbrida y prueba). Las funciones del BORSCHT son:

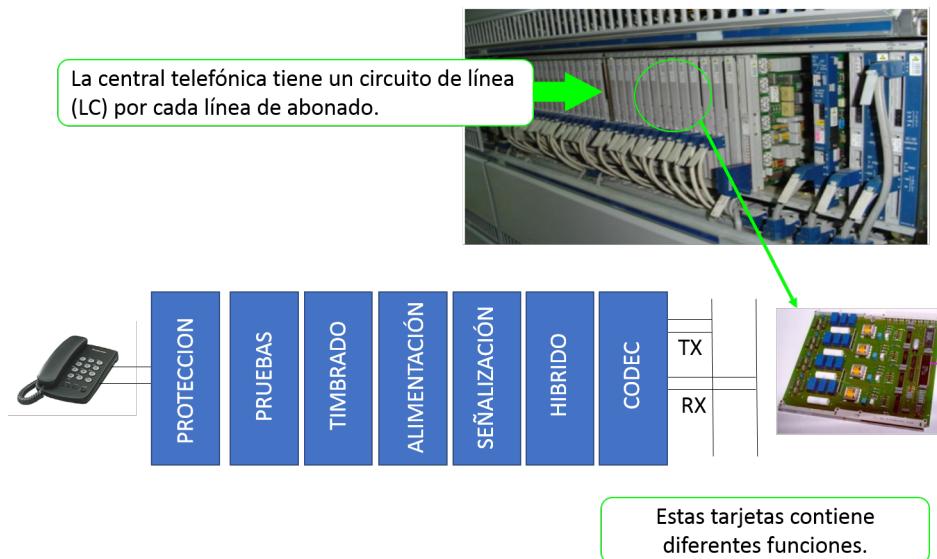


Figura 11.22: Diagrama de bloques de un teléfono analógico

Battery Hace referencia al voltaje de alimentación de -48V DC entre los dos conductores que llegan al terminal, para transmitir la voz y generar la corriente de señalización.

Overvoltage Protection La planta externa aérea se expone a sobrecorrientes por lo que esta función hace referencia a la protección

Ringing Generación de la corriente de timbrado.

Signaling También conocido como supervisión, esta función hace referencia a la detección de la corriente de colgado y descolgado

Coding Hace referencia a la codificación y decodificación. Incluye funciones de conversión A/D y D/A.

Hybrid Circuito híbrido que convierte el ingreso de 2 hilos a 4 hilos.

Test Debe ser posible conectar el lazo local y el terminal al equipo de prueba en la oficina central.

11.7 Señalización telefónica

A manera de recordar, tenemos el conocimiento acerca de los hilos necesarios para las troncales analógicas:

- 2 hilos para transmitir(T_x)
- 2 hilos para recibir(R_x)
- 2 hilos para señalizar

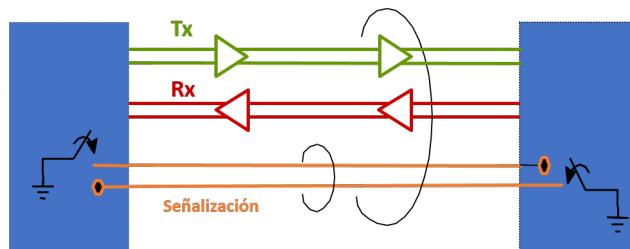


Figura 11.23: Hilos de las troncales analógicas

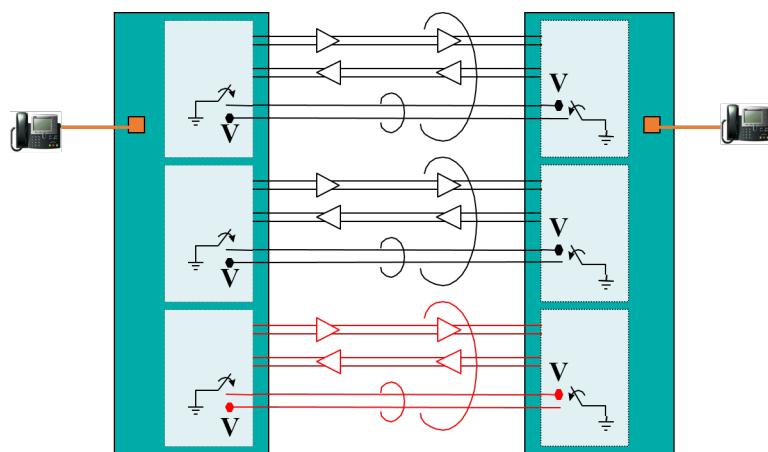


Figura 11.24: Hilos en la troncal

Ejercicio 11.1 En el diseño de una red telefónica se ha establecido que la ruta directa entre las centrales A y B debe ser capaz de mantener en promedio unas 86 llamadas simultáneas.

¿Cuántos pares de hilos de cobre debe tener el cable que se instale en esta ruta?

¿Cuántos de esos pares se van a dedicar a señalización? **Solución:**

Con un promedio de 86 llamadas simultáneas es necesario por cada llamada 3 pares de hilos de cobre, por lo tanto:

$$N \text{ de pares de cobre} = \text{numero de llamadas} \times 3$$

$$N = 86 \times 3 = 258 \quad (11.2)$$

Necesitamos 258 pares de cobre.

Cada llamada requiere 3 pares de cobre, es decir, un tercio de los pares totales por llamada(3 pares o 6 cables de cobre). Por lo tanto:

$$T_s = \frac{258}{3} = 86 \quad (11.3)$$

Son 86 pares de cobre dedicados a la señalización. ■

11.8 Planta interna y externa

Se forma de los siguientes componentes:

11.8.1 Planta Interna

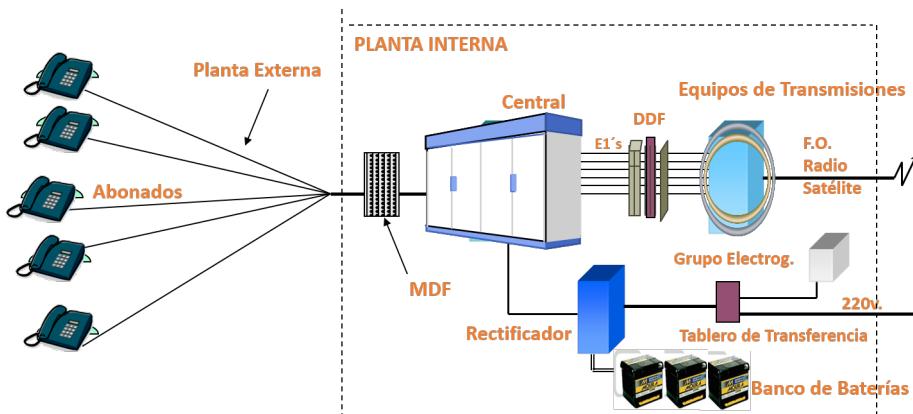


Figura 11.25: Diagrama planta interna

- **Distribuidor general y fosa de cables:** Unen la planta interior con la exterior.



Figura 11.26: Main Distribution Frame: Une la planta externa con la interna

- **Equipos de transmisión:** Se encarga de enviar las señales eléctricas.



Figura 11.27: Fosa de cables:Recibe los cables por galerías subterráneas



Figura 11.28: Equipos de transmisión:Pueden ser de cobre o fibra óptica

- **Equipos de conmutación:**Sirven para que la señal de los equipos de transmisión llegue a los receptores.



Figura 11.29: Comutadores:Establecen la llamada y determinan la ruta.

- **Planta de fuerza y banco de baterías:**Proveen la energía necesaria para que los equipos funcionen.

11.8.2 Planta externa

La planta externa está constituida por la canalización de red principal y red secundaria:

- **Canalización.** Es la red subterránea por la que se envían los cables de cobre.
- **Cajas de distribución.** Conectan la red principal con la red secundaria, mediante puentes.
- **Postes.** Sirven para tender los cables de cobre.

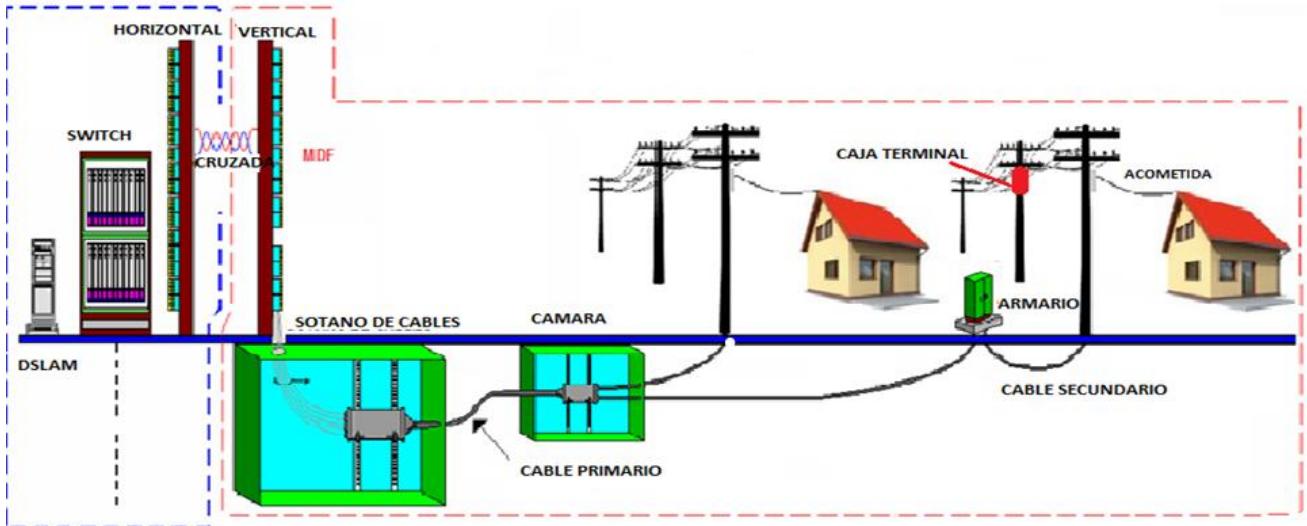


Figura 11.30: Planta externa

- **Cajas terminales.** Son un punto de dispersión de la red. Hasta aquí llegan los cables de cobre.

Tipos de planta externa:

Rígidas Son aquellas en las que todos los conductores se prolongan eléctricamente desde el repartidor principal hasta el punto de distribución (caja terminal) mediante empalmes cerrados. Las redes de este tipo son económicas a condición de que la densidad telefónica sea reducida o de que las líneas de abonado sean muy cortas, pues, de lo contrario, los gastos de instalación y explotación de los puntos de sub repartición (cualquier punto entre la central y el punto de distribución) serían superiores a las economías de pares que se conseguirán con una red flexible. Son adecuadas para las zonas rurales y para las zonas urbanas inmediatamente próximas a la central (zona de servicio directo).

Flexibles Cuando se intercalan entre el repartidor principal y los elementos de distribución puntos de subreplicación (armarios) se obtiene una estructura denominada red flexible, de esta manera la red de línea de abonado queda dividida en dos secciones distintas: Sección de cables principales y Sección de cable de distribución.

Son adecuadas para las zonas de demanda no definida, zonas urbanas, zonas de largo alcance.

Las redes semirigidas Combinan el uso de empalmes y el uso de armarios para la distribución del cableado de red hacia los abonados.

Es así como la planta externa forma el conjunto de elementos e instalaciones que sirven de vínculo entre el abonado y su correspondiente central, como así también el vínculo entre dos centrales.

Elementos de la planta externa

MDF Es el elemento principal de distribución de líneas, y que se encuentra ubicado en el edificio de la central, se conectan los cables de enlace, de red directa y los cables alimentadores. Los blocks de conexión de líneas, son de 100 pares y se instalan en forma vertical para los cables de planta externa y horizontal para los cables de planta interna

Armarios de distribución Elemento subrepartidor de líneas, que por medio de puentes, permite conectar un par primario con cualquier par secundario. Puente: Par de conductores (24 AWG), que se usa en los MDF y armarios, como medio de conexión entre los diferentes blocks. Las capacidades se obtienen sumando número de pares primarios más los pares secundarios.

Canalización Es el sistema construido en forma subterránea, para permitir la instalación, retiro, protección

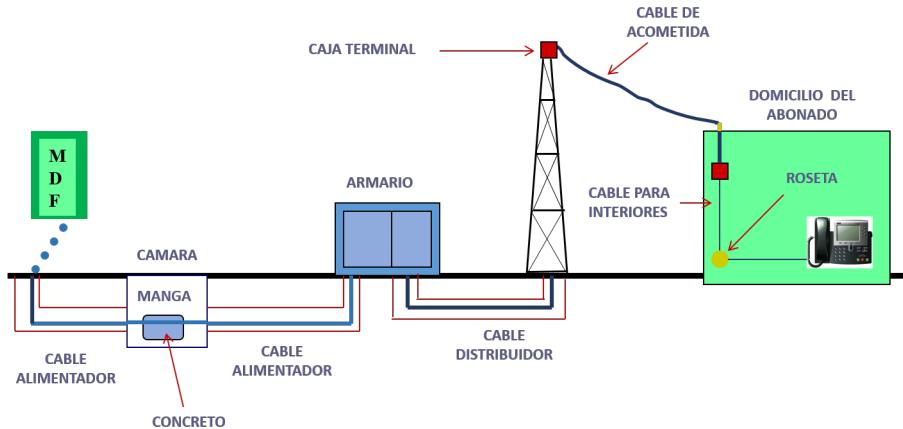


Figura 11.31: Planta externa diagrama

y mantenimiento de los cables; en una red de servicios de telecomunicaciones, a partir del túnel de cables. Las canalizaciones están conformadas por **Cámaras** que son pozos subterráneos destinados a alojar los empalmes y facilitar, la instalación y cambio de dirección de los cables; y **Ducterías** que son ductos que conecta cámaras entre si, una cámara con el túnel de cables, una cámara a subida a poste, fachada , edificios o base de un armario.

Postes Elemento de soporte que se utiliza para la instalación de los cables de redes de telecomunicaciones en forma aérea. Se tiene dos tipos: Los de **madera** de 7mt y de **concreto** de 9,11 y 15m.

Características:

1. Poseen 3 niveles de líneas de cables
2. Los postes de concreto se usan en redes urbanas de gran tamaño y de madera para zonas rurales.
3. La separación entre poste y poste es variable de 30 a 70 mts.
4. Los postes pueden tener 2 líneas de cables por nivel.



12. Herramientas para el análisis telefónico

12.1 Tráfico Telefónico

En el diseño de sistemas telefónicos un factor muy importante es la ingeniería de tráfico ya que juega un papel muy importante en buscar la solución optima en función del costo y la eficiencia del diseño de la red. El tráfico telefónico esta ligado al concepto de ocupación, por lo que decimos que un circuito esta cursando tráfico cuando esta ocupado. El tráfico telefónico es medible, en términos de tiempo, que se entiende en tiempo de ocupación. Este depende del número de llamadas y de la duración de las mismas. Esta de más decir que el tráfico es aleatorio, por lo que naturaleza estadística.

12.1.1 Variaciones

Existen variaciones que depende del tiempo en el que se muestrea, por ejemplo tenemos las variaciones diarias, semanales, anuales y accidentales

Variaciones Diarias

A nivel diario, se puede medir a lo largo de un día el comportamiento de un abonado, es decir, posee ciertos hábitos que caracterizan un día: hora de almuerzo, break del trabajo, salida, hora de trabajo, etc.

Variación Semanal

Con el mismo usuario o con un grupo más grande se puede caracterizar una semana, los hábitos al igual que el diario son los actores de todo: días de trabajo, días de relajo, viernes por la noche, domingo de descanso, etc.

Variación anual

La anual se caracteriza o se ve afectada por épocas o estaciones; por ejemplo verano es caracterizado por las vacaciones, por lo que las llamadas crecen a comparación de actividades normales(dependiendo de la actividad).

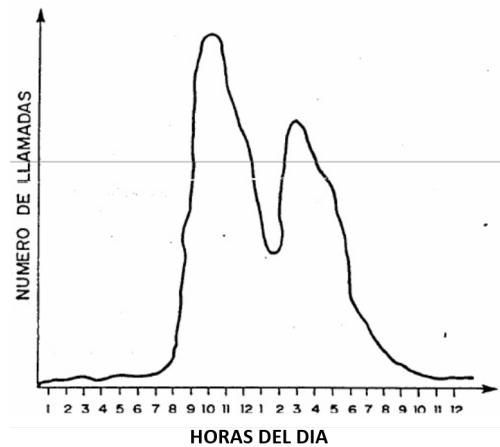


Figura 12.1: Variación diaria

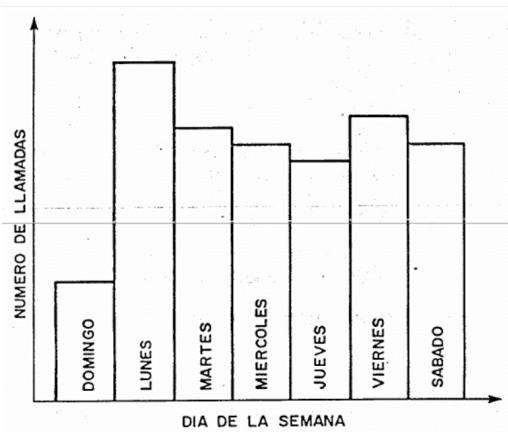


Figura 12.2: Variación semanal

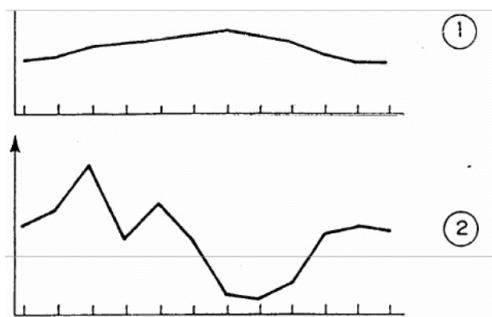


Figura 12.3: a)Variación anual en verano b)Variación anual en actividad industrial

Variación accidental

Los eventos aleatorios son eventos que afectarán nuestra variación, hay unos que pueden llamarse aleatorios que se pueden prever pero no del todo(elecciones políticas, conciertos, festivales, etc) y otros totalmente impensables(crisis políticas en un país, guerras, pandemias, etc)

12.2 Medidas de tráfico telefónico

Volumen del tráfico(V_t)

El tráfico telefónico está ligado al concepto de ocupación y puede medirse en términos de tiempo. Por lo que se puede concluir que se refiere al tiempo de ocupación de un circuito telefónico. Es decir, si un teléfono está ocupado durante una hora al día, el volumen de tráfico cursado durante dicho día por el teléfono es de una hora, y si un teléfono está ocupado durante una hora a la semana, el volumen de tráfico cursado durante la semana por el teléfono es también de una hora. Este concepto se puede expandir a un conjunto de circuitos y no solo restringido a un solo sistema.

Definición 12.2.1 — Volumen del tráfico. Dado un cierto número de llamadas(valor adimensional) y una duración media de cada llamada(horas, minutos o segundos); es posible calcular el **volumen del tráfico** de la siguiente manera:

$$V_t = n \times d \quad (12.1)$$

Donde:

n:número de llamadas.

d:duración promedio de cada llamada.

Ejercicio 12.1 Una central telefónica tiene 3 troncales.

1. Ha cursado 5 llamadas cada una con un promedio de duración de 10 min.
2. Ha cursado 8 llamadas cada una con un promedio de duración de 5 min.
3. Ha cursado 3 llamadas cada una con un promedio de duración de 7min.

Calcular los volúmenes de tráfico individuales y total de toda la troncal.

Solución

Nos pide calcular volumen, por lo que tenemos que extraer tan solo dos datos para cada uno: número de llamadas y su duración promedio y aplicar la ecuación 12.1

1. Troncal 1: 5 llamadas con duración de 10 min:

$$V_t = 5 \text{ llamadas} \times 10 \text{ min}$$

$$V_t = 50 \text{ min}$$

2. Troncal 2: 8 llamadas con duración de 5 min:

$$V_t = 8 \text{ llamadas} \times 5 \text{ min}$$

$$V_t = 40 \text{ min}$$

3. Troncal 3: 3 llamadas con duración de 7 min:

$$V_t = 3 \text{ llamadas} \times 7 \text{ min}$$

$$V_t = 21 \text{ min}$$

4. Troncal total: tan solo sumamos los volúmenes individuales:

$$V_t = 50 \text{ min} + 40 \text{ min} + 21 \text{ min}$$

$$V_t = 111 \text{ min}$$

Algunas unidades para medir el volumen han sido creadas para su mejor compresión:

- **Llamada reducida(LLR)** es un tiempo de ocupación de 2 min.

$$1LLR = 2\text{min} = 120\text{s}$$

- **Centum Call Seconds(CCS)** es un tiempo de ocupación de 100s.

$$1CCS = 100\text{s}$$

- La relación entre ellos está dada por

$$1LLR = 1,2CCS \rightarrow 1CCS = \frac{1}{1,2LLR} = 0,833LLR$$

Tiempo de observación(T_o)

El volumen de tráfico, por sí solo, no es una magnitud capaz de dar idea del grado de ocupación. Para tener una idea completa del grado de ocupación de un circuito o grupo de circuitos, necesitamos además saber el periodo durante el cual se ha observado el circuito o grupo de circuitos. A este periodo de tiempo se le denomina tiempo de observación, o tiempo de referencia

Intensidad de tráfico(I_t)

La intensidad de tráfico es una medida más completa por lo que no solo toma en cuenta el volumen de tráfico sino también el tiempo de observación.

Definición 12.2.2 — Intensidad de tráfico. La intensidad de tráfico si nos da una medida del grado de ocupación. La intensidad de tráfico, se expresa en ERLANGS. Esta definida como:

$$I_t = \frac{V_t}{T_o} \quad (12.2)$$

Donde:

V_t :volumen de tráfico.

T_o :Tiempo de observación.

R Para que la intensidad pueda ser medida en Erlangs es esencial que tanto el volumen de tráfico y tiempo de observación estén en horas.

Erlang

El Erlang es independiente del tiempo de observación , por ejemplo, si se tiene un canal de datos que cursa un tráfico de 1 hora en un periodo de 1 hora se dice que la intensidad de tráfico es 1 Erlang, ahora si se tiene otro canal que cursa un tráfico de 1 día en un periodo de 1 día también se dirá que hay una intensidad de tráfico de 1 Erlang. Por lo general se toma 1 hora de observación, por ejemplo, si una oficina tiene dos operadores de teléfonos y ambos están ocupados durante todo el tiempo, esto representa 2 Erlangs de tráfico, o si un canal de radio está ocupado durante treinta minutos en una hora se dice que soporta un tráfico de 0.5 Erlangs. De forma alternativa, un Erlang puede ser considerado como "multiplicador de utilización"por unidad de tiempo, así un uso del 100 % corresponde a 1 Erlang, una utilización de 200 % son 2 Erlangs, y así sucesivamente. Por ejemplo, si el uso total del móvil en un área por hora es de 180 minutos, esto representa $180/60 = 3$ Erlangs.

Ejercicio 12.2 5 circuitos de una central de conmutación, cursan cada uno de ellos 32 llamadas de una duración media de 2 minutos cada una.

1. Calcular el volumen de tráfico en minutos.
2. Calcular el volumen de tráfico en LLR.
3. Calcular el volumen de tráfico en CCS.

Solución

1. Para cada circuito se halla el volumen de tráfico, pero tenemos en cuenta que son 5 circuitos, por lo que podemos incluir esta cantidad en la ecuación, es como hallar individualmente cada volumen y luego sumar todos los circuitos:

$$V_t = 5 \times 32 \times 2 = 320\text{min}$$

2. Para transformar en LLR dividimos entre 2, puesto que nuestro volumen está en min, así que es posible pasarlo a LLR:

$$V_t = \frac{320\text{min}}{2\text{min}} = 120\text{LLR}$$

3. Para pasarlo a CCS tenemos que asegurarnos que están en segundos, de ahí recién podemos hacer la conversión:

$$V_t = 320\text{min} \times \frac{60\text{s}}{1\text{min}} = 19200\text{s}$$

Una vez que están en segundos recien podemos hacer la conversión:

$$V_t = \frac{19200\text{s}}{100\text{s}} = 192\text{CCS}$$

■

Ejercicio 12.3 Calcular la intensidad de tráfico por un aparato telefónico, que cursa 27 llamadas de una duración media de 5 minutos, en un día.

Solución

$$V_t = 27 \times 5 = 135\text{min}$$

Para recalcar, vamos a expresar en Erlangs, así que es necesario que todo este en horas.

$$V_t = \frac{135\text{min}}{60} = 2,25\text{h}$$

Ahora podemos calcular la intensidad de tráfico en Erlangs, es por ello que usamos 1 día=24 horas:

$$I_t = \frac{2,25}{24} = 0,09375\text{ Erlangs} \quad (12.3)$$

■

Hora cargada

En la práctica suele tomarse como Tiempo de Observación, la denominada hora cargada (HC); es el período de 60 minutos consecutivos durante los cuales el volumen de tráfico es máximo. Al ser una hora de uso, es común modificar las anteriores medidas(LLR y CCS)para que puedan ser usadas con esta hora.

- **Llamada reducida en la hora cargada(LLR/HC)**es la intensidad de tráfico correspondiente a un volumen de tráfico de una llamada reducida (LLR) cursada por un órgano o grupo de órganos, o por un circuito o grupo de circuitos, durante la hora cargada.
- **CIENTOS DE SEGUNDOS EN LA HORA CARGADA (CCS/HC)**la intensidad de tráfico correspondiente a un volumen de tráfico de 100 segundos (1CCS) cursado por un órgano o grupo de órganos, o por un circuito o grupo de circuitos, durante la hora cargada.
- **Equivalencias:**
 - 1 Erlang=30 LLR/HC=36 CCS/HC
 - 1 LLR/HC = 0.033 ERLANGS = 1.2 CCS/HC
 - 1 CCS/HC = 0.028 ERLANGS = 0.833 LLR/HC

Inclusive hay nuevas unidades como **Segundos de llamada(CS)** y **Minutos de llamada(CM)** que tambien son usados como medida de la intensidad:

$$1\text{Erlang} = 36\text{CCS} = 3600\text{CS} = 60\text{CM}$$

Continuando con la hora cargada, como su nombre lo dice: Es la hora en el que el volumen de tráfico es el máximo: Se elige la hora de mayor tráfico, lo que asegura un suficiente número de

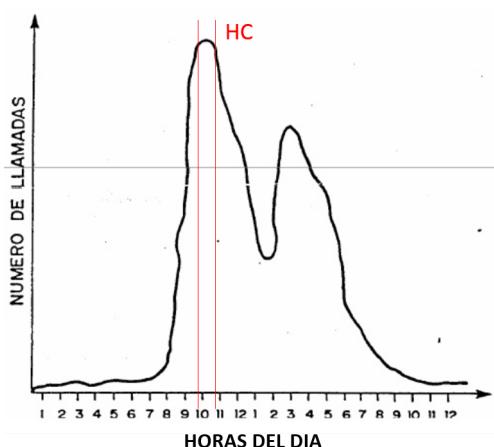


Figura 12.4: Representación de la Hora Cargada

elementos para poder cursar todas las llamadas que se presenten en los restantes períodos de menor carga.No obstante, se toma la intensidad de tráfico media a lo largo de dicha hora carga. Esto quiere decir que habrá en este período momentos de intensidad mayor, durante los cuales se rechazarán las llamadas que no puedan ser atendidas a causa de la saturación. En general, cada día del año tiene su HC y pueden presentarse variaciones de unos días a otros, tanto en la situación de dicha HC como en la intensidad de tráfico que se curse en ella.

Para fines de dimensionamiento de grupos de circuitos, conviene tomar una hora cargada de referencia, de forma que la intensidad de tráfico durante ésta sea el promedio de la que generalmente se cursa en las horas cargadas de la época de mayor carga. Mas no deben considerarse HC anormales(eventos aleatorios únicos) como una crisis, un evento único, etc.

En la práctica, se simplificará la determinación si se consideran períodos completos de 15 minutos y se hace una lectura global de cada período; la HC que se determine de esta forma será la correspondiente a los cuatro períodos consecutivos de 15 minutos que dan una suma mayor.

12.3 Tipos de tráfico

Existen varios tipos de tráfico:

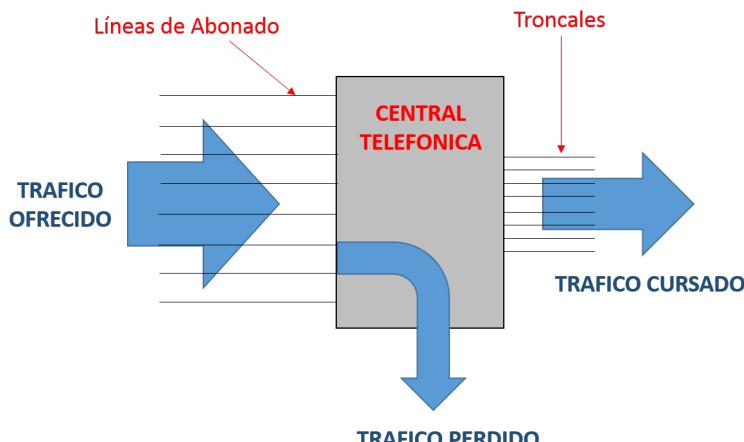


Figura 12.5: Diagrama de tipos de tráfico

- Tráfico ofrecido:** es el tráfico que los abonados intentan cursar y lo ofrecen al conjunto de circuitos de la central para ser cursados. En otras palabras, es el total de llamadas que ingresan a un circuito.
- Tráfico cursado:** Es el tráfico que logró alcanzar un circuito, en otras palabras es el tráfico atendido por los grupos de circuitos cuando se han establecido las llamadas.
- Tráfico desbordamiento:** es el tráfico que no pudo ser cursado por los grupos de circuitos directos de la central y puede ser entregado a otro grupo de circuitos de desborde. Este tráfico aún está en el circuito y no son rechazados, sino que están a la espera que uno circuito se desocupe para poder ingresar en su lugar.
- Tráfico Perdido:** es el tráfico de desbordamiento que no pudo ser cursado. El tráfico perdido es la diferencia entre el tráfico ofrecido y el tráfico cursado y se puede reducir aumentando la capacidad del sistema. Ahora sí, este es el tráfico que ha sido rechazado y fue perdido(colgado). El tráfico perdido es el tráfico que no pudo ser cursado al encontrar el sistema en congestión. De la imagen 12.5 se puede deducir que:

$$\text{Tráfico Ofrecido} - \text{Tráfico Perdido} = \text{Tráfico Cursado} \quad (12.4)$$

El tráfico ofrecido es un concepto teórico y se utiliza sólo para propósitos de planificación teórica. Sólo el tráfico cursado es medida en la práctica y por supuesto depende de la capacidad de la red o sistema.

Desde el punto de vista económico, la capacidad de la red siempre será menor que el tráfico ofrecido, esto se debe al carácter aleatorio de las comunicaciones lo que produce que en condiciones normales de funcionamiento sólo un porcentaje de los usuarios de la red solicitan recursos a la misma.

12.4 Accesibilidad

El rendimiento de un grupo de circuitos de salida, queda determinado esencialmente por la cantidad de líneas de salida del grupo que pueda alcanzarse.

Es decir cuantas líneas de salida pueden alcanzarse desde una línea de entrada. Esa cantidad se llama **accesibilidad**.

Accesibilidad completa

Cuando las líneas de entrada pueden acceder a cualquiera de las líneas de salida disponibles:

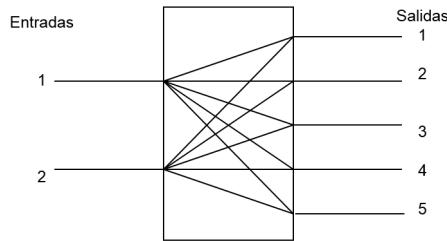


Figura 12.6: Accesibilidad completa de grado 5

Accesibilidad Limitada

Cuando el numero de líneas de salida que pueden ser alcanzadas por una línea de entrada es inferior al total, es decir, se restringen las salidas para una o más entradas, en la figura 12.7 vemos que las salidas K para el usuario de la linea 1 y 2 es K=3 aunque el sistema tiene un grado de accesibilidad 6(porque tiene 6 salidas). Es por ello que llegamos a una generalidad:

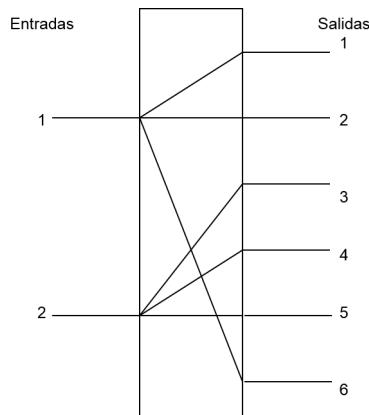


Figura 12.7: Accesibilidad limitada de grado 6

Definición 12.4.1 La accesibilidad(salidas) de cada usuario individual debe ser menor o igual a la accesibilidad(salidas) total.

$$K_u \leq N \quad (12.5)$$

Donde:

K_u =número de salidas del u-ésimo usuario.

N=Salidas(accesibilidad) total.

Sistemas de pérdida

En los sistemas de pérdidas el abonado que solicita una comunicación y encuentra congestión recibe el tono de ocupado que lo obliga a cancelar o "perder"su comunicación para repetir posteriormente.

Sistema a Espera

En el sistema de espera, si el abonado trata de establecer una comunicación no existiendo enlaces libres, no recibe el tono de ocupado en forma inmediata sino que se le permite esperar hasta que surjan las disponibilidades necesarias, vale decir que el intento se almacena sin necesidad de repetir la llamada.

12.5 Grade of service(GoS)

Nos da una idea de la "calidad" que reciben nuestros usuarios.

En sistemas con pérdida, se define como la probabilidad de pérdida (o probabilidad de congestión). Coincide con la probabilidad de que un usuario, al realizar una llamada, se encuentre con el sistema sin recursos.

En sistemas de demora se suele relacionar con la probabilidad de esperar para disponer de un recurso.

También se puede definir como el tiempo medio de espera antes de obtener un recurso para cursar la llamada.

En un escenario real sin bloqueo en una central, es necesario proporcionar tantas líneas como abonados haya, esta opción es la mejor pero no es económico debido a que no siempre se estarán usando el 100 % de las redes, así que se pone una cantidad ideal(después de un estudio del sistema) y barata al mismo tiempo. Por lo tanto, algunas llamadas tienen que ser rechazadas y reintentadas cuando las líneas están siendo utilizadas por otros abonados.

Definición 12.5.1 El grado de servicio se refiere a la proporción de llamadas fallidas en relación con el número total de llamadas.

También se define como la proporción de tráfico perdido respecto al tráfico ofrecido.

$$GoS = \frac{T_p}{T_o} = \frac{C_p}{C_t} \quad (12.6)$$

Donde:

T_p : Tráfico perdido

T_o : Tráfico ofrecido

C_p : Llamadas perdidas

C_t : Llamadas totales

R Es obvio que el número de llamadas perdidas debe ser el menor número posible mientras que el número total de llamadas tiene que ser el máximo posible, es así que llegamos a la conclusión que un número óptimo de GoS tiene que ser el menor posible como 0.002, donde podemos inferir que por cada 1000 llamadas podemos perder 2.

R Para que GoS será cero se requiere que la cantidad de abonados sea igual a la cantidad de servidores. Es preciso señalar que el GoS estará entre **0 y 1 por ser una probabilidad**.

El GoS se aplica a una conexión de terminal a terminal. Pero normalmente un centro de conmutación se divide en los siguientes pares de componentes para poder analizarlos:

- Una llamada interna (abonado a central de conmutación).
- Una llamada saliente a la red troncal (central de conmutación a troncal).
- La red troncal (troncal a troncal).
- Una llamada de terminación (central de conmutación a abonado).

Ver ejercicios:¹

12.6 Diseño de redes

Para el encaminamiento de redes se tiene dos posibles escenarios: linea directa o linea alternativa. La red mostrada en la figura 12.8 es un sistema de ejemplo debido a su simplicidad; en la vida real,

¹Ver ejercicios resueltos 14.3 y 14.6

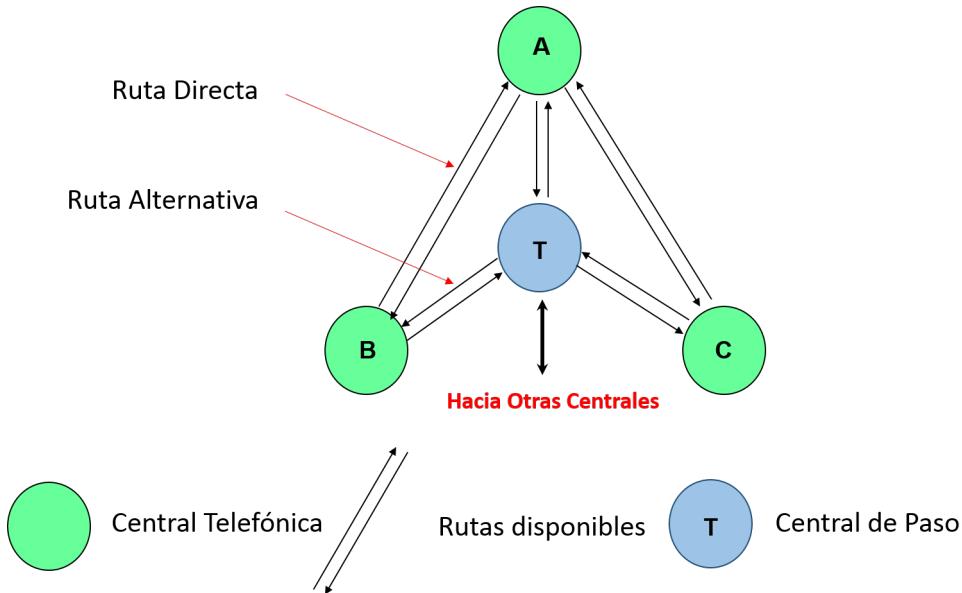


Figura 12.8: Redes de encaminamiento

redes gigantes serían muy laboriosas de graficar, es por ello que diagramas como la imagen 12.8 se expresa de otra manera. A continuación se detalla la forma equivalente de la figura 12.8. Para ello es necesario recordar un poco de la figura 12.5.

Imaginemos que nos encontramos en la estructura de la figura 12.9, tenemos un "círcuito de agua", donde el propósito es llegar de A hacia B. El agua(*tráfico ofrecido*) será vertida por la entrada de A, el agua avanzará y se encontrará con la primera bifurcación(*troncales directas A-B*): una parte del agua caerá y seguirá haciendo todo el recorrido(*tráfico de desbordamiento*) mientras que la otra llegará de frente a B(*tráfico cursado*), hasta aquí decimos que no nos importa el agua que logró alcanzar el atajo, pues esta agua ya logró el cometido: llegar de A a B, sigamos analizando el flujo de agua que no logró alcanzar este atajo.

Agua que no logró alcanzar este atajo de A a B aún tiene una oportunidad: llegar a otro atajo(*troncales hacia B*). Es este punto tenemos el mismo escenario anterior; no todos van a lograr llegar a B, habrá un flujo que logre pasar este atajo(*Tráfico cursado*) y otro simplemente será desecharido(*tráfico perdido*).

Aprenderemos la ilustración que ilustra el encaminamiento de tráfico desde una determinada central frente las demás, como flechas y troncales salientes de la misma(figura 12.10a).

Viendo la figura 12.10b vemos que consta(generalmente) de 2 niveles y 3 rutas de intensidad de tráfico, entonces podemos generalizar:

- Ae** Tráfico ofrecido
- C** Troncales directas de A-B
- m** Tráfico desbordamiento
- S** Troncales hacia B
- A'** Tráfico perdido

Sabiendo esto y usando el ejemplo de la figura 12.8 podemos expresar todas las maneras posibles de ruteo, en otras palabras, las diferentes formas en las que A,C y T pueden llegar a B; B, T y A pueden llegar a C y B, C y T pueden llegar a A.

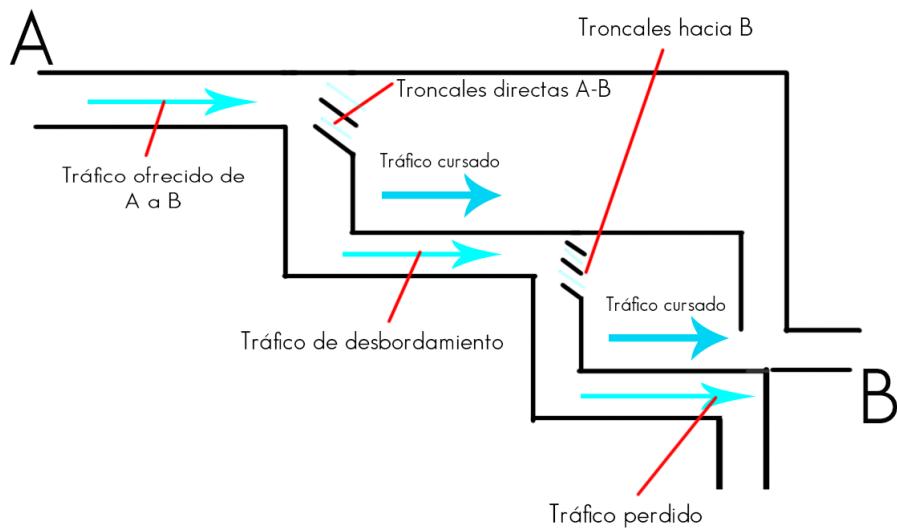


Figura 12.9: Analogía gráfico equivalente

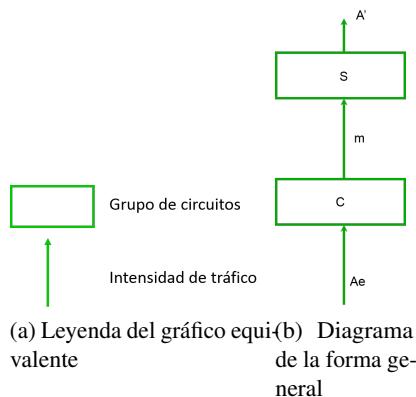


Figura 12.10: Generalidades

Ejercicio 12.4 Hallar la ruta de encaminamiento para A de la figura 12.8:

Solución

Viendo la figura tenemos 3 orígenes de tráfico: B, C y T. Por lo tanto habrá 3 "fuentes" de tráfico que irán hacia A.

Para B-A:

Vemos una ruta directa B-A, como son centrales telefónicas asumimos que, entre las dos existen una troncal directa, por que existe una ruta directa. Si hay una ruta directa se aplica la ruta común" de la figura 12.10b entonces de A a B decimos que existe una ruta donde el tráfico pasará por las troncales directas A-B y luego pasará por cualquier troncal hacia B.

Para C-A

Para este caso vemos que es lo mismo que el caso B-A, habrá una linea ruta directa que empieza de C para ir a A. Como es directa aplicamos el diagrama de la figura 12.10b.

Con eso tenemos dos rutas cubiertas.

Para T-A

Sabemos por la figura 12.10b que nuestro tráfico va primero por la central directa A-B(C), y luego va por la central hacia B(S), es así que de la figura del ejercicio T no es una central, al contrario, es una **central de paso**, es por eso que ahora no vamos a representar el camino T-A como lo solíamos hacer usando de referencia el esquema de 12.10b, sino que haremos una pequeña variación. Como esta es la central alternativa, ignoraremos totalmente las troncales directas A-B, lo hacemos por que esta troncal ya hemos analizado debido a que en este ejemplo ya hay una ruta directa B-A. Entonces solo nos quedaría: el tráfico de desbordamiento, troncales hacia A(no confundir con troncales directas B-A) y el tráfico perdido.

Tenemos 3 diagramas por separado, pero algo en común de los 3 son las troncales hacia A, podemos decir, de manera coloquial que, vamos a factorizar estas troncales para unir los 3 diagramas. El resultado final sería la figura 12.11 ■

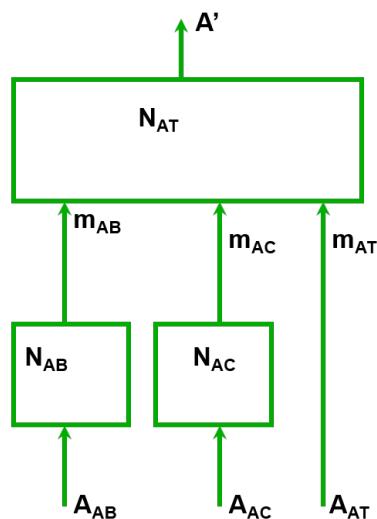


Figura 12.11: Ejemplo para A

Si quieres practicar, puedes hacerlo con las otras centrales(B, C y T)

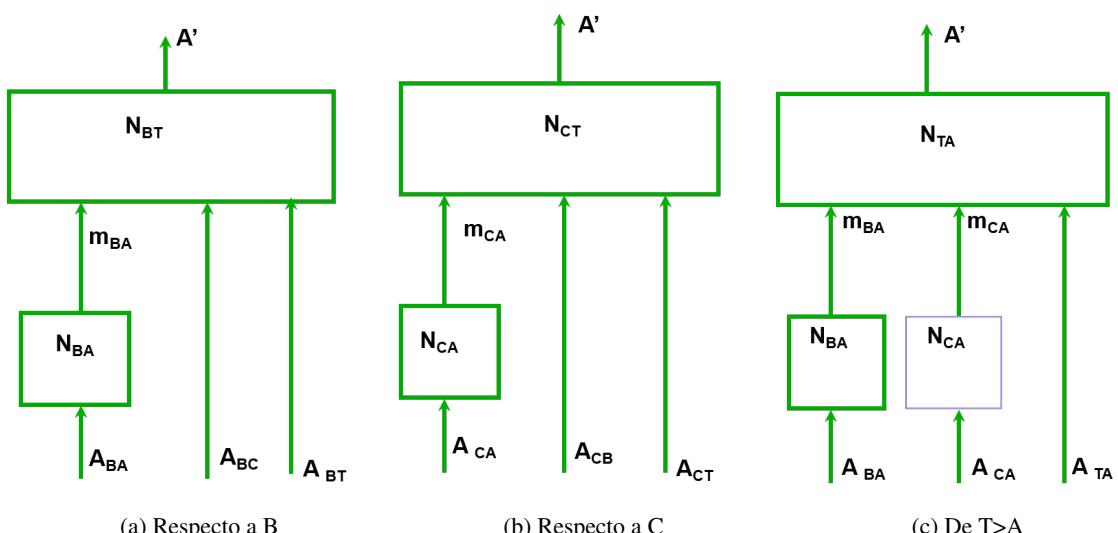


Figura 12.12: Gráficos de práctica resueltos

12.7 Encaminamiento de tráfico

Cuando nuestro tráfico telefónico va hacia una troncal y luego hacia nuestro destino es probable que no exista una ruta directa, ya sea por varios factores: que la troncal directa este saturada, que no tenga mucha capacidad, que haya habido un problema, etc. Entonces, no podemos utilizar esta "vía directa", por ende pasamos de ser tráfico ofrecido a tráfico de desbordamiento, y ¿Cómo se desborda un río?: el agua buscará una ruta alternativa para que pueda volver a su cauce, igualmente en nuestro caso: nuestro tráfico de desbordamiento tendrá que buscar más rutas para poder llegar y tratar en lo posible de no perder esta llamada.

Pero no podemos redirigir el tráfico como sea, existen ciertos criterios que se definen para poder lograr un encaminamiento óptimo, puede ser un poco difícil de entender pero trataré de explicarlo detalladamente:

Criterios de encaminamiento:

Estos son los 3 criterios que usaremos, léelos lentamente y trata de entender:

1. **Criterio 1:** Si entre la central donde se decide el encaminamiento y la central de destino final de la llamada existe una sección directa, se encaminará el tráfico por esta como primera opción.

Aquellas llamadas que no puedan ser cursadas por esta sección directa (TRAFICO DE DESBORDAMIENTO), lo harán por sección final correspondiente.

2. **Criterio 2:** Si existe sección directa entre la central donde se decide el encaminamiento y una central que sea, por red Jerárquica, de rango superior a la central de destino final de la llamada, se cursara por esta, y el tráfico de desbordamiento se cursara por la sección final correspondiente.

Si existen varias secciones directas que cumplan esta condición, se hará por la más corta.

El tráfico que una sección directa no pueda cursar, nunca se cursara por otra sección directa, sino por sección final.

3. **Criterio 3:** Se encamina el tráfico como única opción por la sección final correspondiente. El tráfico que dicha sección final no pueda cursar, no podrá encaminarse y constituirá lo que se denomina TRAFICO PERDIDO.

Estos 3 criterios jugarán bajo las siguientes reglas:

1. El encaminamiento se hace desde el abonado A al abonado B, haciendo los tránsitos necesarios en centrales intermedias.
2. Cada central toma **uno y solo uno** de los criterios: si no puede tomar el primer criterio tomará el segundo, si no puede con el segundo tomará el tercero.
3. Cuando se aplique el primer o segundo criterio, tendrá que estudiarse por separado cada uno de los caminos que se obtienen como opción.
4. La alternativa a una sección directa **nunca** es otra sección directa.
5. Las centrales que toman decisiones de encaminamiento dependen del origen y destino de la llamada, y de las decisiones de encaminamiento previas.
6. Se tomarán con preferencia las secciones directas por los que los encaminamientos sean más cortos
7. Solo se puede elegir una ruta directa en los casos:
 - Si está conectada directamente a la central de destino.
 - Si la central de destino está conectada a un nodo en jerarquía superior por una final.
 - No se puede cursar dos secciones directas para llegar a un destino, solo se toma una por recorrido.
8. Si un nodo tiene varias opciones de rutas finales se preferir la de desbordamiento.

Lo sé, es mucha teoría pero no se entiende. Primero antes de hacer un ejercicio tenemos que tener en cuenta ¿Cuál es la jerarquía de centrales(figura 12.13)?

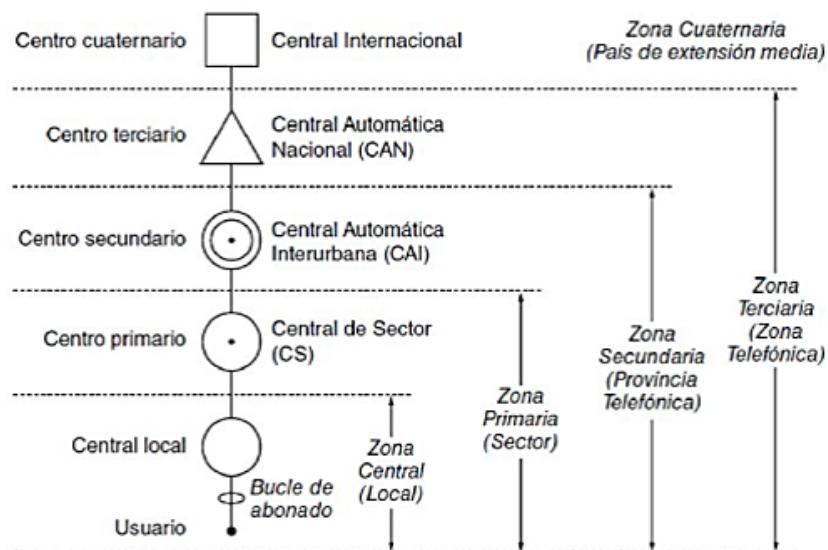


Figura 12.13: Jerarquía de centrales

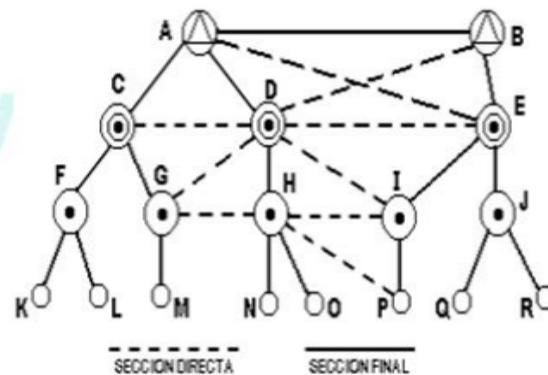


Figura 12.14: Encaminamiento de ejemplo

Ejercicio 12.5 Dada la figura 12.14, encaminar una llamada desde N hasta P de la red.

Solución:

Primera opción: Felizmente podemos usar el primer criterio, tenemos todo para ello: N es usuario, como único camino tenemos la sección final hacia H(centro primario); que nos dice el primer criterio:...entre la central donde se decide el encaminamiento y la central de destino final de la llamada existe una sección directa, se encaminara el tráfico por esta como primera opción...; nuestra central(la única) es H, la central de destino es P, ¿Existe una sección directa? Sí, entonces nos encaminamos por esa ruta, por lo tanto como primera alternativa tenemos:

$$N > H > P$$

Segunda opción: ¿Qué pasaría si H esté saturada? no podríamos pasar por ahí, nos rechazaría el paso, así que buscamos otra ruta alternativa: Estamos en H, varados, ¿A dónde vamos?, le criterio 1 ya no podemos utilizarlo porque H está "prohibida" para cruzar así que leemos el criterio 2:*Si existe sección directa entre la central donde se decide el encaminamiento(en este caso, estamos en D por que H nos ha encaminado a otra central, ademas tenemos en cuenta la*

4ta regla, si o si llegamos a D) y una central que sea, por red jerárquica, de rango superior a la central de destino final de la llamada(en nuestro caso, algo con más rango jerárquico a P es I, y existe la sección directa mencionada el principio), se cursará por esta, y el tráfico de desbordamiento se cursará por la sección final correspondiente(que es la sección directa I-P).. Así tenemos la segunda opción:

$$N > H > D > I > P$$

Tercera opción: Nos ponemos en el caso que la central D está saturada, nos encaminará a otra de más rango jerárquico, **NO** es posible ir de D a C o de D a E(porque tienen el mismo rango), así que pasamos a A. En A usamos el segundo criterio, ¿por qué? pues el primer criterio no se puede, no existe una sección directa desde A hacia P, recordemos que hemos subido de jerarquía en la *segunda opción*(P tuvo que subir a I **solo** por que tenemos sección final y es prácticamente lo mismo, en otras palabras, llegamos a I y automáticamente llegamos a P por sección final).

En A usamos el segundo criterio, lo usamos por que tenemos una **sección directa**, y es de menor rango(A es mayor que E), ademas como nuestra "central final" es I, vemos que entre E e I hay una **sección final**, realmente podemos usar el segundo criterio en todo su esplendor.

$$N > H > D > A > E > I > P$$

- (R) Vemos que A y B están conectados mediante una **sección final**, esto nos quiere decir que pueden ser conmutables, en otras palabras, podemos ir de A a E como el tercer caso pero como están unidas por una **recta final**, la tenemos que tener en cuenta.
- (R) No podemos usar D>E por que ya usamos la sección directa D>I, y como dice la regla 4, tenemos prohibido usar esa sección directa

Es así que, por esa aclaración tenemos como resultado final:

$$N > H > D > A > B > E > I > P$$

Esta respuesta se ve graficada en el imagen 12.17. ■

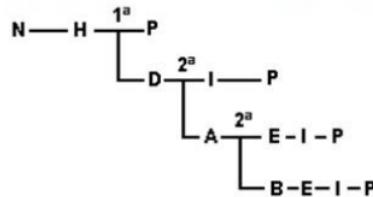


Figura 12.15: Gráfica de la solución

Como siempre, tendrás algunos ejercicios resueltos en el anexo 14.9

12.8 Erlang-B

Analizamos el tráfico no por que sí, sino porque tenemos que diseñar centrales capaces de poder soportar todas(o al menos la mayoría) las llamadas y hacer que la probabilidad de pérdida sera la

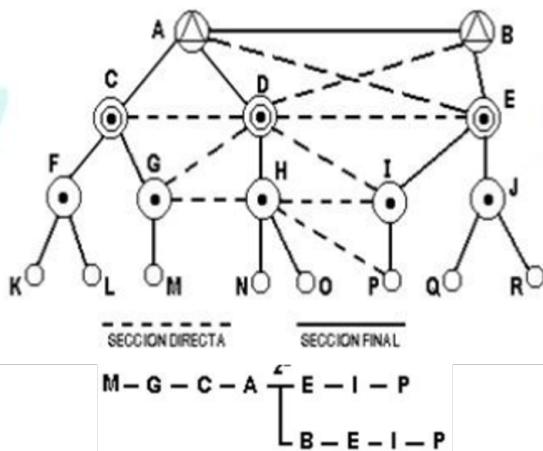


Figura 12.16: Encaminamiento de M a P

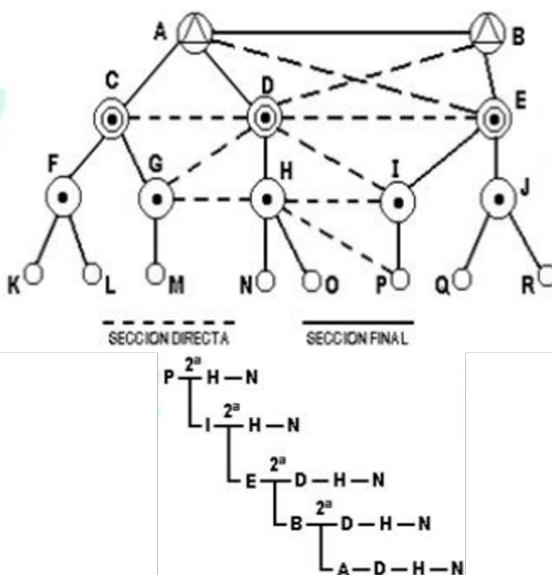


Figura 12.17: Encaminamiento de P a N

mínima.

Es así que Erlang (matemático danés) estudió el comportamiento del tráfico telefónico. En 1917 publicó su trabajo. Realizó las siguientes hipótesis:

- Las llamadas se generan aleatoriamente y solamente una cada vez (modelo de Poisson).
- La probabilidad de que en una ruta estén ocupados un determinado número de enlaces no varía en el tiempo. Es decir, existe equilibrio estadístico.
- Las llamadas que llegan a la ruta y no encuentran ningún enlace libre se pierden, sin esperar ningún tiempo y sin ocupar ningún enlace. No se producen reintentos.
- La duración de las llamadas sigue una distribución exponencial negativa. La probabilidad de que en un tiempo t no llegue ninguna llamada es $e^{-\lambda t}$
- El número de fuentes es infinito. En la práctica, es mucho mayor que el número de enlaces.

Definición 12.8.1 — Intensidad de tráfico II. Existen dos expresiones que dan origen a la

ecuación 12.2, las ecuaciones primitivas son:

$$I_t = \frac{\sum_{i=1}^n h_i}{T} \quad (12.7)$$

$$h = \frac{\sum_{i=1}^n h_i}{c} \quad (12.8)$$

Donde:

I_t : Intensidad de tráfico en Erlangs.

T: Tiempo de observación.

h_i : Duración de cada llamada individual.

n: Número total de llamadas.

h: tiempo medio de la toma de enlace.

Despejando las ecuaciones 12.7 y 12.8(igualando las sumatorias de los numeradores):

$$I \times T = n \times h$$

Y despejando la **intensidad de tráfico(I)** obtenemos:

$$I = \frac{n \times h}{T}; eq 12,2$$

12.8.1 Erlang-B para un enlace

En Erlang-B para un enlace, se relacionan 3 parámetros:

1. I_T : Intensidad de tráfico.
2. N: Número de enlaces de ruta.
3. **B**: Probabilidad de que una llamada encuentre todos los enlaces ocupados y no pueda cursarse (**probabilidad de bloqueo**); exacto, es el GoS.

Suponemos una ruta con infinitos enlaces. El procedimiento de toma de enlaces de esta ruta consiste en intentar siempre en primer lugar el primer enlace. Si está ocupado se intenta el segundo, y así sucesivamente hasta encontrar un enlace libre. En otras palabras, la llamada intentará el primer enlace, ¿Está ocupado?, intentará con el segundo enlace, ¿Está ocupado?, intentará el tercero y así sucesivamente. Esta de mas decir que, si una llamada tuvo que irse al segundo enlace(o cualquiera) y llega otra llamada y ve que esta vació el primer enlace, pues irá por el primer enlace.

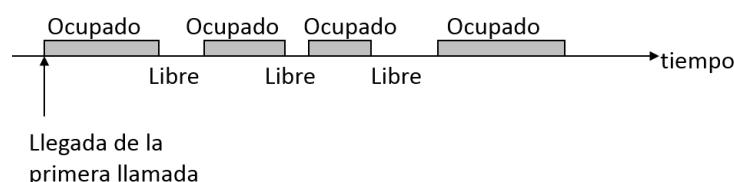


Figura 12.18: Primer enlace: rechazará llamadas mientras este ocupado, caso contrario la aceptará

Formula Erlang-B para un enlace

Definición 12.8.2 — Probabilidad B. Probabilidad de que una llamada encuentre el primer enlace bloqueado es igual a la proporción entre tiempo ocupado y total:

$$B = \frac{I_t}{1 + I_t} \quad (12.9)$$

Donde:

I_t : Intensidad de tráfico B: Probabilidad de que una llamada encuentre el primer enlace bloqueado

$$B = \frac{A}{1+A}$$

A	B	B (%)	Tráfico cursado (Erlangs)
0.01	0.01	1%	0.0099
1	0.5	50%	0.5
99	0.99	99%	0.99

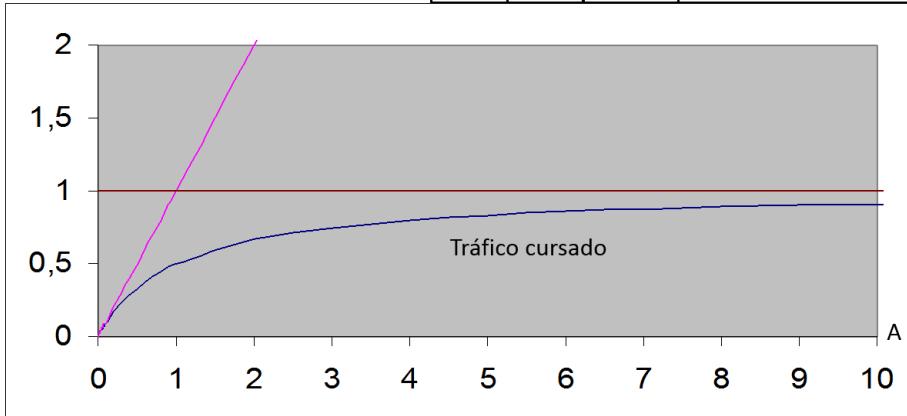


Figura 12.19: Primer enlace: rechazará llamadas mientras este ocupado, caso contrario la aceptará

Algunos puntos a resaltar son:

- Para obtener una probabilidad de bloqueo aceptable (1 % o menor) el tráfico ofrecido debe ser mucho menor que la “capacidad aparente” (0.01 Erlangs frente a 1 Erlang).
- Para obtener una ocupación muy alta hay que ofrecer mucho tráfico, el cual se pierde (bloquea) casi todo.
- Por muy poco tráfico que se ofrezca, siempre hay una probabilidad de perdida (bloqueo).
- El máximo tráfico que se puede cursar es 1 Erlang.

12.8.2 Erlang-B para N enlaces

Posee esta forma:

Definición 12.8.3 — Erlang para N enlaces. Definida por:

$$p(x+1) = \frac{I_t}{x+1} \quad (12.10)$$

La ecuación 12.10 se puede generalizar de la siguiente manera:

$$\begin{aligned} p(1) &= \frac{I_t}{1} \times p(0) \\ p(2) &= \frac{I_t}{2} \times p(1) = \frac{I_t}{2} \times \frac{I_t}{1} \times p(0) \\ p(3) &= \frac{I_t}{3} \times p(2) = \frac{I_t}{3} \times \frac{I_t}{2} \times \frac{I_t}{1} \times p(0) \\ p(x) &= \frac{I_t^x}{x!} \times p(0) \end{aligned} \quad (12.11)$$

Donde:

$p(x)$: probabilidad de que haya x enlaces ocupados

I_t : Intensidad de tráfico

A continuación haremos una gráfica para representar la ecuación 12.11, solo para graficar usaremos $I_t=5$ (usaremos x , solo cambiamos de variable) y $p(0)=0.02$:

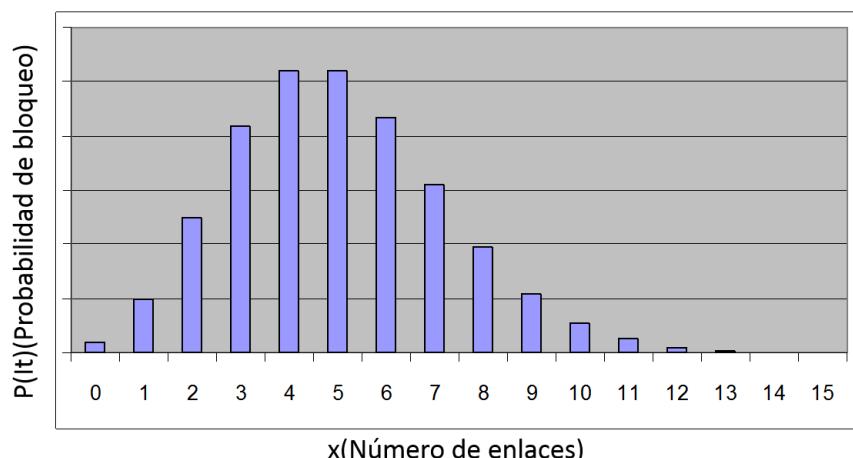


Figura 12.20: Gráfica de la ecuación 12.11 con $I_t(x)=5$ y $p(0)=0.02$

Distribución de Poisson

Para calcular la probabilidad de bloqueo se tiene una expresión para cierto enlace x ; a su vez se suscitan dos casos particulares: x es un número cualquiera entero y cuando x es igual al número total de enlaces.

Definición 12.8.4 — Distribución de Poisson para X específico. La distribución de probabilidad se denomina distribución de Poisson truncada o distribución de pérdidas de Erlang, y se define por:

$$P(x) = \frac{I_t^x}{x! \sum_{k=0}^N \frac{I_t^k}{k!}} \quad (12.12)$$

Se lee: *La probabilidad de que el enlace x este ocupada es*

Donde:

x : troncal escogida.

N : Número de troncales.

I_t : Intensidad de tráfico(Erlangs).

¿Qué pasaría si $x=N$? La ecuación 12.12 cambiará ligeramente:

Definición 12.8.5 — Distribución de Poisson cuando X es igual a N. En particular, cuando $x = N$, la probabilidad de pérdida viene dada por:

$$P(x) = P(N) = B(N, I_t) = \frac{A^N}{N! \sum_{k=0}^N \frac{I_t^k}{k!}} \quad (12.13)$$

Donde:

I_t : Intensidad de tráfico.

N: Número total de troncales.

B: Probabilidad de pérdida o bloqueo(Gos).

La expresión 12.13 iterativa se expresa de forma extendida de la siguiente manera:

$$B(N, I_t) = \frac{I_t^N}{N! (1 + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots + \frac{A^N}{N!})} \quad (12.14)$$

R Este resultado se conoce como fórmula de Erlang del primer tipo, fórmula de Erlang-B o fórmula de pérdida de Erlang.

La ecuación especifica la probabilidad de bloqueo para un sistema con llegadas aleatorias desde una fuente infinita y distribuciones de tiempo de espera arbitrarias.

La fórmula Erlang B da la congestión temporal del sistema y relaciona la probabilidad de bloqueo con el tráfico ofrecido y el número de líneas troncales.

Los valores de $B(N, I_t)$ obtenidos, se han representado frente a los Erlangs de tráfico A ofrecidos para diferentes valores del número de líneas N. Se introducen dos medidas: Número medio de troncales

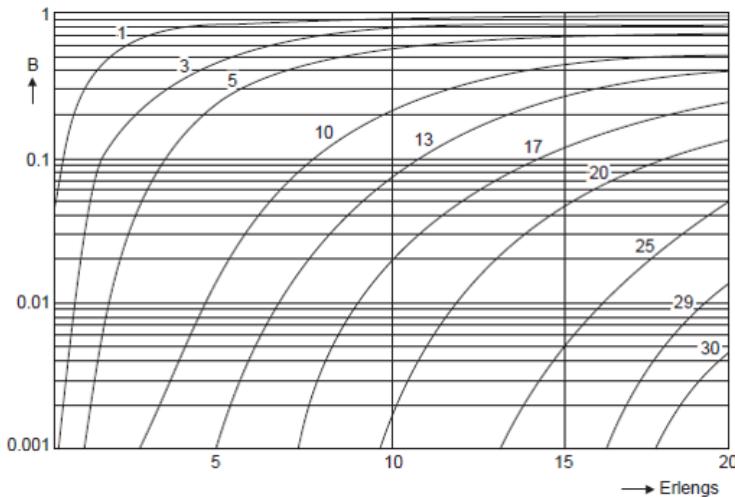


Figura 12.21: Gráfica de B en función de I_t (Erlangs) para distintos N

ocupadas y ocupación de una troncal, ambas ecuaciones se detallan a continuación:

Definición 12.8.6 — Número medio de troncales ocupadas(carga cursada). El número medio de troncales ocupadas u ocupación se define como la **carga cursada**:

$$I'_t = I_t (1 - B(N, I_t)) \quad (12.15)$$

Así, la carga cursada es la posición de la carga ofrecida que no se pierde del sistema. La carga cursada por línea se conoce como ocupación de la red.

Definición 12.8.7 — Ocupación de una troncal. La ocupación de la red ρ es una medida del grado de utilización de un grupo de líneas y se denomina a veces **factor de utilización**. Definida por:

$$\rho = \frac{I'_t}{N} = \frac{I_t (1 - B(N, I_t))}{N} \quad (12.16)$$

13. Telefonía IP

VoIP es un acrónimo de *Voice Over Internet Protocol* (Voz sobre Protocolo de Internet), o en términos más comunes, *Servicio Telefónico a través de Internet*. Dale una leída a 11.5 antes de seguir.

13.0.1 Redes de computadoras

Las redes de computadoras trabajan con tramas y paquetes, a manera de ejemplificar: Para ello,

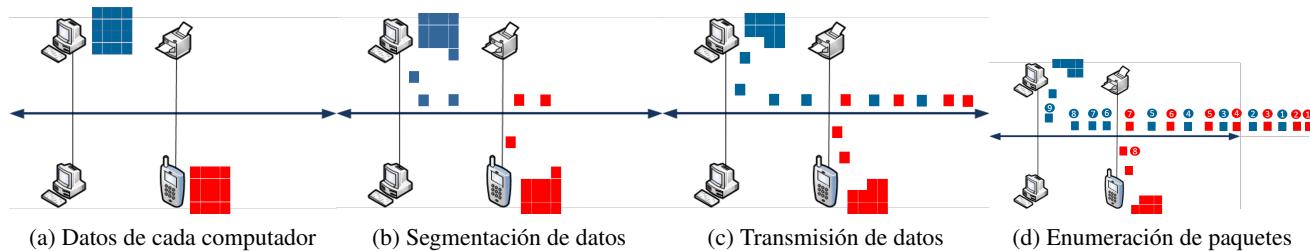


Figura 13.1: Enviar datos de computadoras por el mismo canal

es necesario recordar sobre que trata la segmentación y la multiplexación.

Segmentación Dividir la información en piezas mas pequeñas, para hacer mas fácil el manejo de la data dentro de la red. Además es posible aprovechar mejor el ancho de banda.

Multiplexación Al dividir la información en piezas mas pequeñas es posible además enviar diferentes conversaciones a través del mismo medio.

Hablando teóricamente es muy sencillo, pero ¿Cómo nos aseguramos que cada cuadradito "de color ira a donde debe ir?". Para ello existen protocolos como el **PDU**(Protocol Data Unit).

13.0.2 PDU

Una **Protocol data unit** (PDU) es un término de interconexión de sistema abierto (OSI) utilizado en telecomunicaciones que se refiere a un grupo de información agregada o eliminada por



Figura 13.2: Representación alusiva a segmentación y multiplexación

una capa del modelo OSI. Cada capa del modelo utiliza la PDU para comunicarse e intercambiar información, que solo puede ser leída por la capa de pares en el dispositivo receptor y luego se pasa a la siguiente capa superior después de la extracción. En palabras simples, cada capa tiene un "lenguaje distinto", por eso, nuestra cabecera mientras pasa de capa en capa, cada una de estas le añade una pequeña información propia para que cuando llegué a la misma capa pero en otro sistema este lo pueda entender e interpretar. Recordar que la información irá desde la capa de aplicación hasta la capa física en el emisor, mientras que en el receptor irá desde la capa física hasta la capa de aplicación. Es debido a esto que los añadidos a nuestra cabezera a lo largo de las capas en el emisor desaparecerán cuando lleguen al receptor. Si cogemos solo un solo nivel, se puede decir que:

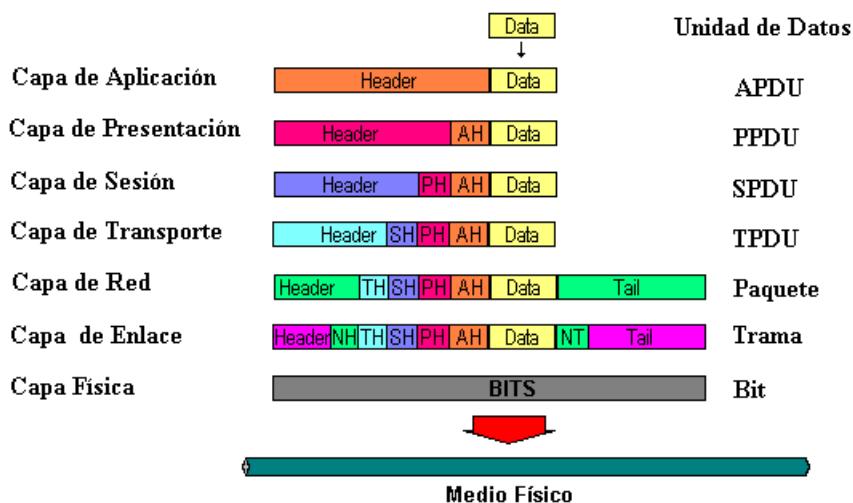


Figura 13.3: PDU

13.0.3 Arquitectura de red

La arquitectura de una red esta dada por las capas y sus funciones, así como los protocolos que las implementan. Dentro de la arquitectura se definen interfaces que actúan entre las capas como medio de comunicación. Las interfaces ponen a disposición operaciones y servicios para las capas superiores. Los protocolos implementan estas operaciones y servicios y escuchan en las interfaces las peticiones por atender. La lista de estos protocolos es conocida como pila de protocolos.

Es común trabajar con capa porque los modelos de capas reducen la complejidad de diseño e implementación de soluciones de comunicación; haciendo que cada capa cumpla funciones definidas, que son ofrecidas como servicios a las capas superiores. Los modelos de capas permiten realizar cambios y mejoras en una capa sin tener que afectar las funciones de las demás. Cada capa encapsula sus funciones lo que mantiene una estructura simple para el resto de capas, que solo conocen los servicios a los que pueden acceder, pero no necesitan saber como son implementados.

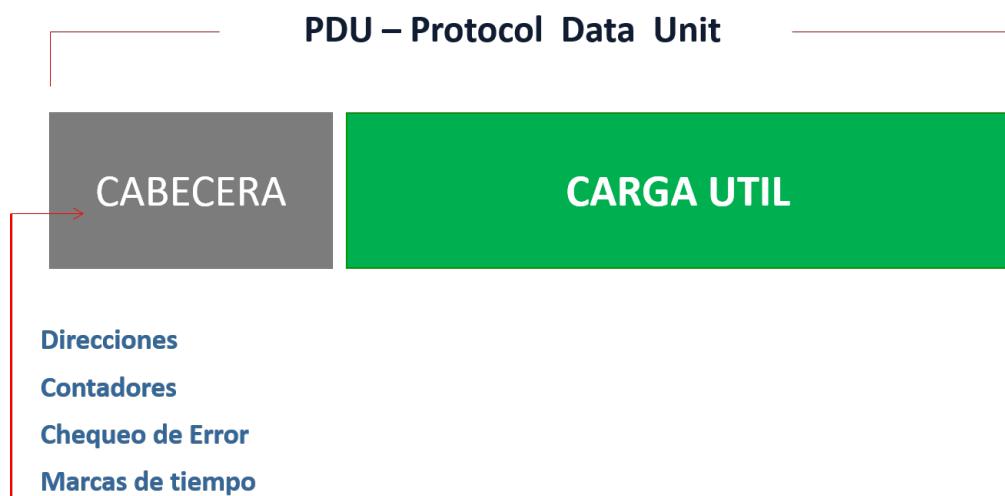


Figura 13.4: PDU añade cabezales en cada capa propia ellos

No se transfieren datos de manera directa desde la capa n de una máquina a la capa n de otra máquina, sino que cada capa pasa los datos y la información de control a la capa inmediatamente inferior, hasta que se alcanza a la capa más baja. Debajo de la capa 1 se encuentra el medio físico a través del cual ocurre la comunicación real. La estructura general de verlas es como se gráfica en la figura Existen dos arquitecturas de redes importantes: el modelo de referencia OSI y el modelo de

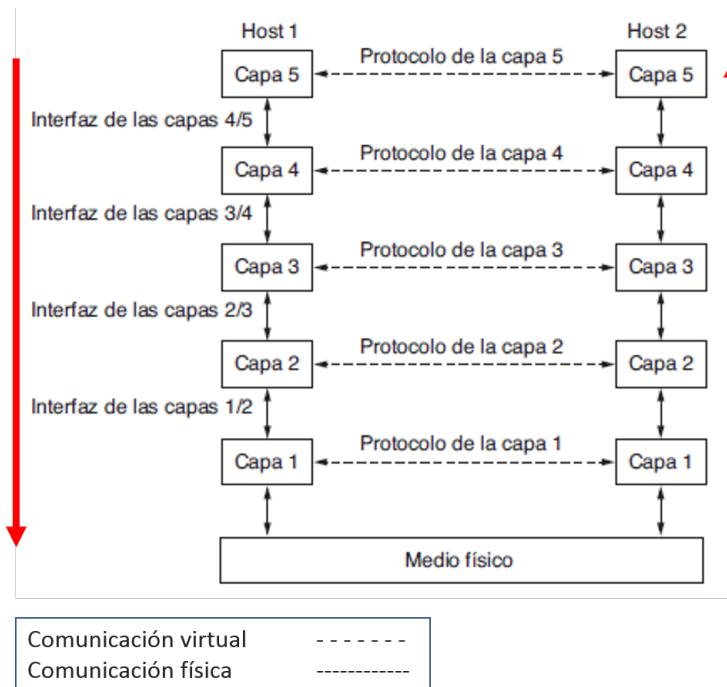


Figura 13.5: Estructura general de capas: A veces pueden tener más o menos capas como el protocolo OSI(7 capas) o TCP/IP(4 capas)

referencia TCP/IP. Aunque ya casi no se utilizan los protocolos asociados con el modelo OSI, el modelo en sí es bastante general y sigue siendo válido; asimismo, las características en cada nivel siguen siendo muy importantes. El modelo TCP/IP tiene las propiedades opuestas: el modelo en sí

no se utiliza mucho, pero los protocolos son usados ampliamente.

¿Por qué razones los protocolos del modelo OSI no se adoptaron en forma generalizada?

A principios de 1980 el desarrollo de redes originó desorden en muchos sentidos. Se produjo un enorme crecimiento en la cantidad y tamaño de las redes. A medida que las empresas tomaban conciencia de las ventajas de usar tecnologías de conexión, las redes se agregaban o expandían a casi la misma velocidad con la que se introducían las nuevas tecnologías de red, solo había un problema: cada empresa creaba una nueva "tecnología" de comunicación entre redes, esto hacía que existiesen muchas variantes por lo que dificultó la comunicación entre varios sistemas. Debido a su problema de incompatibilidad de redes, la Organización Internacional para la Estandarización (ISO) investigó nuevos modelos de conexión como la red de Digital Equipment Corporation (DECnet), la Arquitectura de Sistemas de Red (Systems Network Architecture) y TCP/IP a fin de encontrar un conjunto de reglas aplicables de forma general a todas las redes. Con base en esta investigación, la ISO desarrolló un "modelo de red" que ayuda a los fabricantes a crear redes que sean compatibles con otras redes. Existen diversos protocolos de acuerdo a cómo se espera que sea la comunicación. Este conjunto de protocolos se denomina TCP/IP. TCP/IP se ha convertido en el estándar de-facto para la conexión en red corporativa. Las redes TCP/IP son ampliamente escalables, para lo que TCP/IP puede utilizarse tanto para redes pequeñas como grandes.

¿Cuáles son los problemas que tiene el modelo TCP/IP?

- El modelo no distingue bien entre servicios, interfaces y protocolos, lo cual afecta al diseño de nuevas tecnologías en base a TCP/IP.
- Es más difícil de configurar y mantener a pesar de tener menos capas.
- Es algo más lento en redes con un volumen de tráfico medio bajo, puede ser más rápido en redes con un volumen de tráfico grande donde haya que enrutar un gran número de tramas.
- Peor rendimiento para uso en servidores de fichero e impresión



Figura 13.6: La información más la cabecera de cada capa pasa a la siguiente. La información es encapsulada en cada capa hasta salir como una cadena de bits por el medio físico.

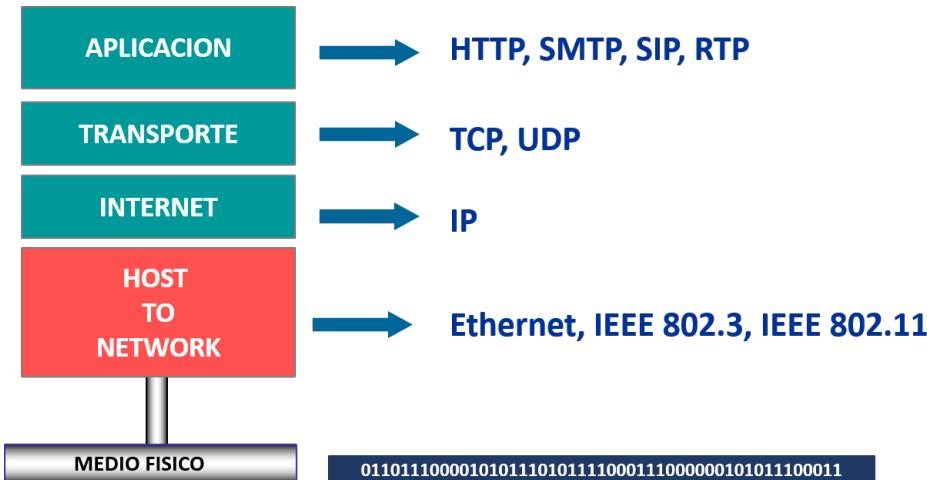


Figura 13.7: Capas del modelo TCP/IP: resaltando que el SIP es el más importante para telefonía IP. Lo que más interesa de este modelo no son las capas sino los protocolos usados por cada una de ellas

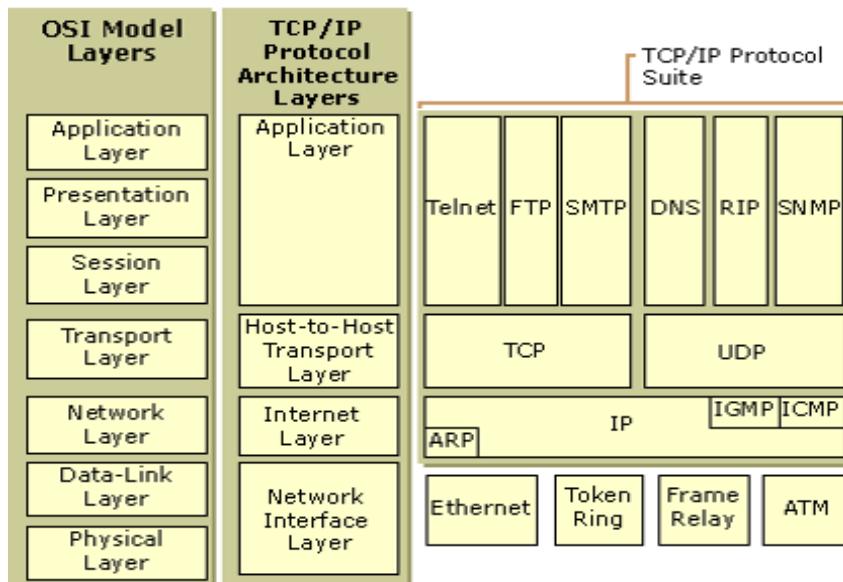


Figura 13.8: Comparación entre ambos modelos: OSI vs TCP/IP y los protocolos de cada capa

13.0.4 Tipos de servicios

Orientado a la conexión

Es un servicio que garantiza el control del flujo de mensajes: inicia, mantiene y termina las comunicaciones entre el cliente y el servidor. Está modelado a partir del sistema telefónico. Para hablar con alguien se levanta el auricular, se marca el número, hablamos y después se cuelga. En algunos casos al establecer una conexión, el emisor, el receptor y la subred llevan a cabo una negociación en cuanto a los parámetros que se van a usar, como el tamaño máximo del mensaje, la calidad requerida del servicio y demás cuestiones relacionadas. También es llamado **círculo**, ese es otro nombre para una conexión con recursos asociados.

Características:

- Es confiable.

- Se establece un circuito virtual.
- El estado de los mensajes es conocido en todo momento.
- Se tiene un control de secuencia de mensajes.
- Existe confirmación y retransmisión de mensajes.
- Se mantiene un control de flujo.
- Se genera overhead adicional.
- Es lento.

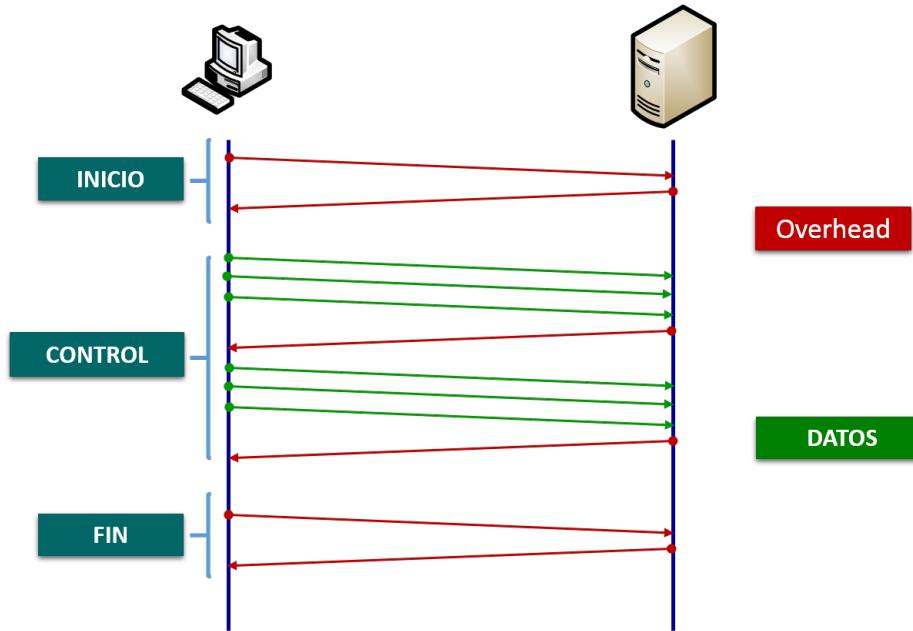


Figura 13.9: Orientado a la conexión: El usuario del servicio establece primero una conexión, la utiliza y después la libera.

Un ejemplo en la que es apropiado un servicio orientado a la conexión es la transferencia de archivos. El propietario del archivo desea estar seguro de que todos los bits lleguen correctamente y en el mismo orden en el que se enviaron. Sería poco aceptable un servicio que ocasionalmente pierda unos cuantos bits, incluso aunque fuera mucho más rápido.

No orientado a la conexión

Es un servicio que solo se centra en el envío de mensajes, pero no garantiza el control del flujo de mensajes, ni el estado de los mensajes. Está modelado a partir del sistema postal. Cada mensaje lleva la dirección de destino completa, y cada uno es enrutado hacia los nodos intermedios, en forma independiente a todos los mensajes subsecuentes. Hay distintos nombres para los mensajes en diferentes contextos: un paquete es un mensaje en la capa de red. A este servicio también se le conoce como servicio de **datagramas**.

Características

- No es confiable.
- El estado de los mensajes no es relevante.
- No hay control de secuencia de mensajes.
- No existe confirmación y ni retransmisión de mensajes.
- No interesa el control de flujo.
- Es rápido.

Por ejemplo, los emisores de correo electrónico basura (spammers) envían su correo a muchos destinatarios. El emisor no quiere tener que pasar por el problema de establecer una conexión con

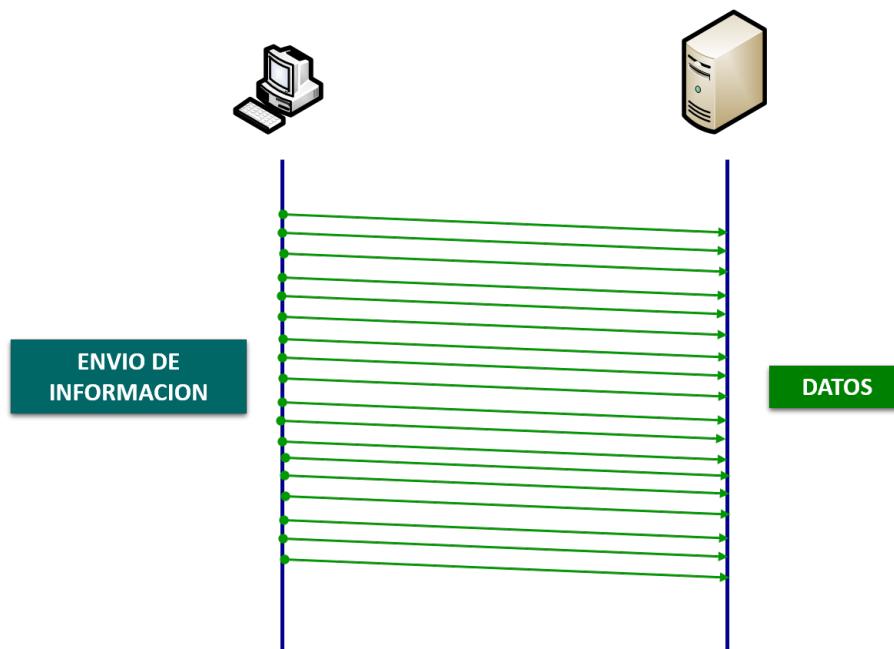


Figura 13.10: NO orientado a la conexión

un destinatario sólo para enviarle un mensaje. Tampoco es esencial una entrega cien por ciento confiable, sobre todo si eso es más costoso. Todo lo que se requiere es una forma de enviar un solo mensaje que tenga muy alta probabilidad de llegar, aunque sin garantías.

Tipo de servicio	Servicio	Ejemplo
Orientado a la conexión	Flujo de mensajes confiable. Flujo de bytes confiable. Conexión no confiable.	Secuencia de páginas. Descarga películas. Voz sobre IP
Sin conexión	Datagrama no confiable. Datagrama confirmación no recepción. Solicitud-respuesta	Correo electrónico basura. Mensajería de texto. Consulta de una base de datos.

Cuadro 13.1: Ejemplos de los servicios Orientado y NO orientado a la conexión

13.1 Repaso: TCP/IP

A manera de recordar las capas vista en el capítulo de Infraestructura vamos a repasar las capas del modelo TCP/IP.

13.1.1 Capa de transporte

Un solo dispositivo puede utilizar varios servicios como e-mail, web, mensajería instantánea, etc. La información de cada aplicación debe ser empaquetada, transportada y entregada a la aplicación apropiada en el dispositivo destino. Para conseguir entregar los flujos de datos a las aplicaciones correctas, la capa de transporte debe identificar la aplicación destino. Para esto se utiliza un identificador el numero de puerto. Además, cada flujo de datos debe ser enviado en piezas

que sean manejables a través de la red, la capa de transporte agrega a cada pieza la información necesaria para ser enviada hasta el otro extremo y ser desensamblada. En la figura 13.9 es ejemplo

TPC/IP	UDP
Confiabilidad.	Rapidez y bajo overhead.
Confirmación de paquetes	Sin confirmación de paquetes
Retransmisión de paquetes perdidos.	Sin retransmisión de paquetes.
Entrega de la información de acuerdo al orden de envío.	Entrega de la información como llega.
En tiempo real sensible al retraso.	No sensible al retraso.

Cuadro 13.2: Capa de transporte-Tipo de servicio

es el protocolo TCP/IP mientras que para la figura 13.10 un claro ejemplo es el protocolo UDP. Por encima de la capa de transporte se encuentra la capa de aplicación, capa que contiene los protocolos de alto nivel como TELNET(terminal virtual), FTP(transferencia de archivos) y SMTP(correo electrónico)

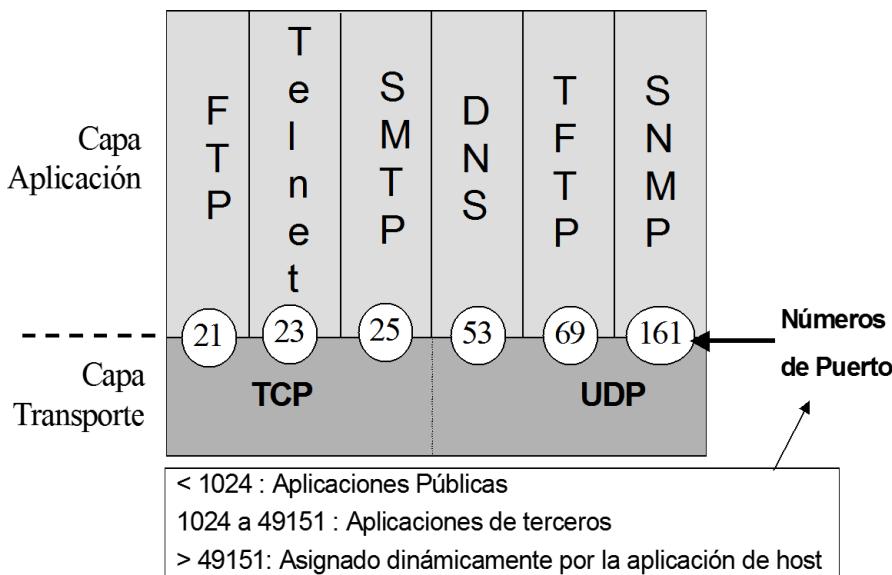


Figura 13.11: Direccionamiento de puertos

13.1.2 Capa de internet

Se encarga de combinar varios enlaces múltiples en redes, y redes de redes en interredes. La capa de red provee el mecanismo necesario para que los dispositivos terminales puedan enviar paquetes de un extremo a otro dentro de una trayectoria a través de diferentes redes. Para cumplir esta tarea utiliza los siguientes mecanismos:

- Direccionamiento.
- Enrutamiento.

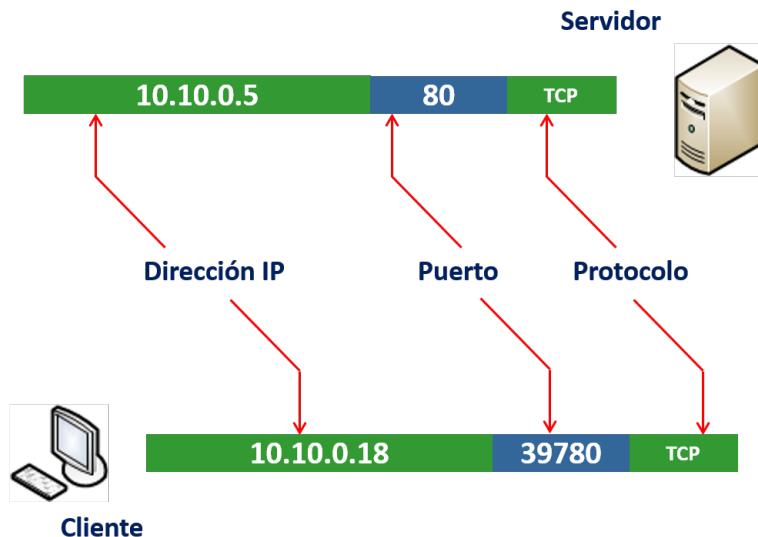


Figura 13.12: Representación de la conformación de paquetes en la capa de transporte.

Direccionamiento

La capa de red provee el mecanismo de direccionamiento para que los paquetes viajen de un extremo a otro a través de un conjunto de redes. Como cada paquete lleva una parte de un flujo mas grande de información hacia un dispositivo terminal remoto este ultimo debe tener una dirección única. Es por eso que en este nivel todos los hosts que forman parte de la red deben tener una dirección, para ser ubicados.

Enrutamiento

Una vez que tenemos direcciones asignadas a todos los dispositivos de la red; la capa de red debe proveer un servicio que guíe a los paquetes a seguir la ruta hacia su destino. El origen y el destino no están siempre dentro de la misma red; por lo que los paquetes viajaran a través de varias redes diferentes. Los dispositivos intermedios que conectan las redes son los ruteadores, su rol es encontrar caminos hacia otras redes y dirigir a los paquetes a través de ellos.

IPv4

IPv4 es actualmente la versión mas utilizada de IP, es el único protocolo utilizado en capa 3 para mover la información de los usuarios a través de internet. Fue diseñado como un protocolo de bajo overhead; proveyendo funciones que son necesarias para entregar un paquete desde el origen en una red hasta el destino en otra red a través de un sistema interconectado de redes. Su diseño no se ocupa de monitorear y administrar el flujo de paquetes; esto es dejado a protocolos de capa superiores. El protocolo IPv4 provee un modelo de direccionamiento jerárquico, basado en direcciones de 32 bits, esto permite que los dispositivos puedan emplear lógica digital para realizar operaciones de comparación entre otras. Sin embargo, para que nosotros podamos recordar y utilizar estas direcciones utilizamos un formato decimal para representarlas: cuatro números decimales separados por puntos. Donde cada uno de estos números decimales representa un numero binario de ocho bits.

1. Clase A: este bloque de direcciones puede soportar redes extremadamente grandes con más de 16 millones de hosts. Como condición, el primer bit del primer octeto debe mantenerse en “0”, lo que implica que el primer octeto solo puede tener 128 subredes.
2. Clase B: este bloque de direcciones puede soportar redes medianamente grandes con más de 65 000 hosts. Como condición, los dos primeros bits del primer octeto deben mantenerse

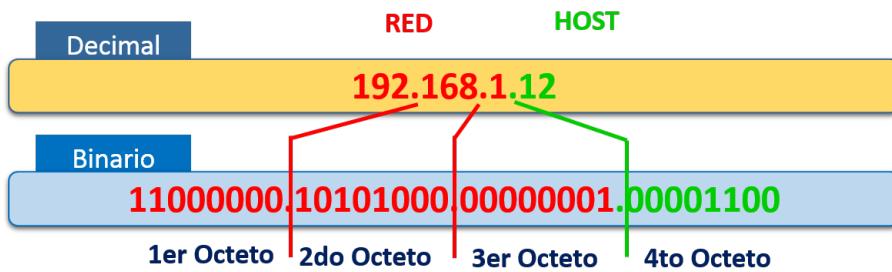
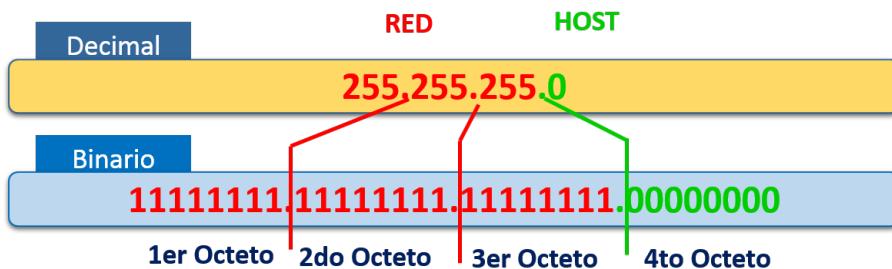
**Mascara:**

Figura 13.13: Las direcciones IPv4 están divididas en dos bloques, un numero de bits comenzando por los mas significativos indican la red. En el ejemplo los primeros 24 bits y finalizando con los bits menos significativos determinan el numero de Host. En el ejemplo los últimos 8 bits.

CLASE	DIRECCIONES DISPONIBLES		CANTIDAD DE REDES	CANTIDAD DE HOSTS	APLICACIÓN
	DESDE	HASTA			
A	0.0.0.0	127.255.255.255	128*	16.777.214	Redes grandes
B	128.0.0.0	191.255.255.255	16.384	65.534	Redes medianas
C	192.0.0.0	223.255.255.255	2.097.152	254	Redes pequeñas
D	224.0.0.0	239.255.255.255	no aplica	no aplica	Multicast
E	240.0.0.0	255.255.255.255	no aplica	no aplica	Investigación

* El intervalo 127.0.0.0 a 127.255.255.255 está reservado como dirección loopback y no se utiliza.

Figura 13.14: Clases de IPv4

en “10”. Este bloque distribuye mejor el espacio de direcciones que la clase A, ya que tiene aprox. La misma cantidad de redes que de hosts.

3. Clase C: este bloque de direcciones puede soportar redes con más de 254 hosts. Como condición, los tres primeros bits del primer octeto deben mantenerse en “110”. Puede proveer direcciones a aprox. 2 millones de redes.

Según el RFC 3330 describe las direcciones ipv4 especiales asignadas por la IANA. Entre ellas podemos identificar las direcciones públicas y privadas.

- **Publicas:** La gran mayoría de direcciones unicast IPv4 son públicas, es decir son direcciones designadas para ser accesibles desde internet.
- **Privadas:** Son direcciones unicast IPv4 que pueden utilizarse dentro de redes privadas que no tienen acceso directo a internet y por lo que es posible reutilizarlas entre una red y otra. Los bloques de direcciones privados son: -10.0.0.0 – 10.255.255.255 (10.0.0.0 /8)
-172.16.0.0 – 172.31.255.255 (172.16.0.0 /12)
-192.168.0.0 – 192.168.255.255 (172.16.0.0 /24)

13.2 VoIP

Voz sobre el Protocolo de Internet o VoIP se refiere a la transmisión del tráfico de voz sobre redes de datos utilizando el protocolo IP. Es decir utilizaremos la pila de protocolos TCP/IP y sus funcionalidades para que la voz pueda ir de un extremo a otro de la red.

A priori esta idea nos permite obtener conclusiones inmediatas, y de la misma forma apreciar las ventajas de esta tecnología, entre otras:

- Puede utilizarse aprovechando la infraestructura creada para una red local o de Internet, con independencia de los medios de transmisión utilizados, permitiendo mantener varias conversaciones simultáneas.
- En empresas no requiere de una instalación simultánea de cableado telefónico para conectar los teléfonos a la centralita o PBX.
- Permite comunicarnos con otra persona sin importarnos la distancia ni el lugar en el que nos encontramos, no siendo necesario utilizar el sistema de telefonía tradicional para efectuar llamadas internacionales, con el consiguiente ahorro económico.
- No requiere de grandes inversiones para su puesta en funcionamiento. Con una computadora con tarjeta de sonido, altavoces, micrófono y el software apropiado, sería suficiente. Cualquier otro dispositivo como un móvil de última generación, preparado para utilizarse en una red TCP/IP, nos serviría, de ahí el carácter portable de dicha tecnología.
- Al basarse en la arquitectura TCP/IP, no sólo se puede utilizar para la transmisión de voz, permitiendo también el envío de imágenes, vídeo o texto, tal y como lo hacemos a través de una red o Internet.
- Hace más eficiente el ancho de banda, utilizando técnicas de compresión y evitando el envío de información durante los espacios de silencio que se producen durante las conversaciones, ya que carece de sentido.

13.3 Digitización de la voz

Las desventajas sobre las señales analógicas usadas en la telefonía convencional. Estas desventajas son:

- La voz análoga no se propaga bien a largas distancias.
- La señal decae con la distancia.
- Ruido de fondo llega a ser comparable a la señal.
- Amplificación de la señal también amplifica el ruido.

Debido a esto, es una gran alternativa la digitalización para contra-restar estos efectos, así se logra discriminar mejor la señal del ruido. Hoy en día, a voz se transmite siempre en forma digital, excepto en el “local loop” (excepto ISDN que extiende la voz digital hasta el abonado). Para realizar la digitalización, al igual que en la modulación donde se utilizan moduladores, aquí se usa **codecs**.

13.3.1 ¿Cómo se digitaliza la voz?

Para poder digitalizar la voz se utiliza el famoso **Teorema de Nyquist**, que nos dice que:

Definición 13.3.1 La frecuencia de muestro es dos veces la frecuencia máxima de la señal a muestrear:

$$f_s \geq 2 * f_a \quad (13.1)$$

Donde:

f_s : Frecuencia de muestreo. f_a : Frecuencia de la señal a muestrear(frecuencia analógica)

A manera de repaso de este tema los pasos para la digitalización se hace de la siguiente manera:

Muestreo

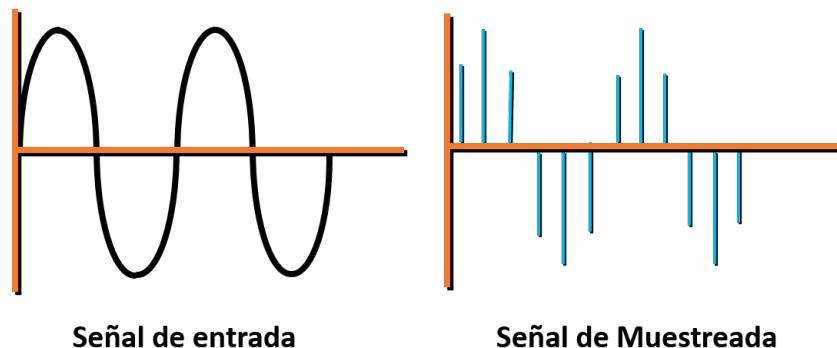


Figura 13.15: Muestreo de una señal seno.

Tomar muestras cada cierto tiempo de la señal a muestrear, mientras más muestras se tiende a perder menor información.

Cuantificación

Como hemos tomado muestras a lo largo del tiempo en el muestreo, tenemos que cuantificar, en otras palabras, tenemos que elegir qué valores tomarán los valores del eje Y. La cuantificación es usado para representar el nivel eléctrico de cada muestra. Dependiendo del número de niveles se necesitará un número específico de bits. Si tenemos planeado cuantificar en 8 niveles necesitamos

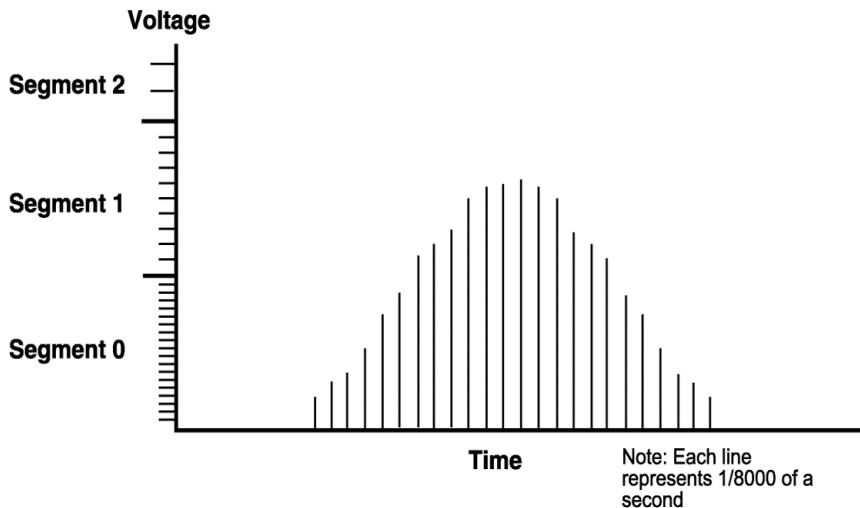


Figura 13.16: Cuantificación del muestreo de una señal seno.

3 bits, que quiere decir esto:

Tenemos 8 niveles: 1, 2, 3, 4, 5, 6, 7 y 8. Como estamos cuantificando recordemos que solo usamos ceros y unos, además tienen que ser únicos, es por ello que usando el sistema binario se asigna una combinación única a cada uno.

1. Nivel 1: 000
2. Nivel 2: 001
3. Nivel 3: 010
4. Nivel 4: 011
5. Nivel 5: 100
6. Nivel 6: 101

- 7. Nivel 7: 110
- 8. Nivel 8: 111

Se ha asignado un código binario único a cada uno, este número también es:

$$\text{Bits necesarios} = 2^n, \text{ donde } n \text{ es el número de niveles.} \quad (13.2)$$

Codificación

De acuerdo al número de bits seleccionado durante el proceso de cuantificación se establecen los códigos que representarán los niveles y se asignan a las diferentes muestras. Una forma común de codificar en PCM (*Pulse Code Modulation*), utiliza codificación por forma de onda y se basa en el teorema de Nyquist:

Definición 13.3.2 — Tasa de transmisión binaria. También llamada **tasa de transmisión de bits**, define el número de bits que se transmiten por unidad de tiempo a través de un sistema de transmisión digital o entre dos dispositivos digitales. Así pues, es la velocidad de transferencia de datos.

$$V_{bin} = f_s \times nbm \quad (13.3)$$

Donde:

V_{bin} : Tasa de transmisión de bits.

f_s : Frecuencia de muestreo.

nbm: Número de bits por muestra.

Ejercicio 13.1 Con un rango de frecuencia de voz de 4kHz, y una resolución de 8 bits por muestra, se puede calcular, la transmisión binaria, primero, usando la ecuación 13.1 obtenemos:

$$f_s \geq 2 \times 4\text{kHz}$$

$$f_s 8\text{kHz}$$

$$V_{bin} = 8\text{KHz} \times 8\text{bits/m}$$

$$V_{bin} = 8000\text{m/s} \times 8\text{bits/m}$$

$$V_{bin} = 64\text{kbytes}$$

$$(13.4)$$

Así entonces, usando la ecuación 13.3:

$$V_{bin} = 8\text{KHz} \times 8\text{bits/m}$$

$$V_{bin} = 8000\text{m/s} \times 8\text{bits/m}$$

$$V_{bin} = 64\text{kbytes}$$

$$(13.5)$$

CODECs

Los CODECS nos dan la posibilidad de utilizar menos ancho de banda por canal de voz. Permitiéndonos pasar más canales de voz (llamadas) por la red.

- G.711(ulaw / alaw) → 64Kbps
- G.726 → 32 Kbps
- G.729 → 8 Kbps (Necesita de licencia)
- GSM → 13 Kbps

13.3.2 Transporte de Voz

Una vez digitalizada la voz en un mensaje a través de CODECS, éste se divide en fragmentos más pequeños o paquetes para enviarlos al destinatario de la conversación. VoIP debe establecer unos mecanismos que al menos aseguren que la voz codificada y comprimida, se entregue a su destinatario en un intervalo de tiempo determinado y con la calidad suficiente. En caso contrario

Codec	Bandwidth	Sample period	Frame size	Frames/ packet	Ethernet Bandwidth
G.711 (PCM)	64 kbps	20 ms	160	1	95.2 kbps
G.723.1A (ACELP)	5.3 kbps	30 ms	20	1	26.1 kbps
G.723.1A (MP-MLQ)	6.4 kbps	30 ms	24	1	27.2 kbps
G.726 (ADPCM)	32 kbps	20 ms	80	1	63.2 kbps
G.728 (LD-CELP)	16 kbps	2.5 ms	5	4	78.4 kbps
G.729a (CS-CELP)	8 kbps	10 ms	10	2	39.2 kbps
AMR-WB/G.722.2 (ACELP)	6.6 kbps	20 ms	17	1	38.0 kbps

Figura 13.17: COCECs.

la conversación tendría cortes y por tanto se degradaría la calidad del servicio. Para garantizar la transmisión, VoIP se integra en la arquitectura TCP/IP utilizando una serie de protocolos en la capa de aplicación, que recurren al protocolo de la capa de transporte UDP. El protocolo UDP no ejerce control sobre la transmisión como TCP, haciendo que sea más rápido y por tanto recomendable para la transmisión de voz en tiempo real.

La combinación de protocolos que permiten la transmisión de paquetes de voz en cuestión de nanosegundos, son los siguientes:

NTP (Network time protocol o Protocolo de tiempo de red): este protocolo se encarga de controlar que los fragmentos de voz digitalizada llegan a su destinatario en el intervalo de tiempo requerido.

RTP (Real-time transport protocol o Protocolo de transporte en tiempo real): proporciona funciones de transporte de red extremo a extremo, encapsulando la voz digitalizada en paquetes. Se transporta generalmente sobre otros protocolos y NO se usa un puerto definido *well-known port*, teniendo al 5004 como puerto por defecto, además que debe ser par y el puerto de control debe ser el siguiente valor(impar).

RTCP (Real-time transport control protocol o Protocolo de control de transporte en tiempo real): este protocolo se complementa con RTP, realizando un seguimiento del reparto de paquetes de voz digitalizada mediante el envío periódico de paquetes de control, permitiendo gestionar la calidad del servicio (QoS).

13.4 Componentes de VoIP

Las redes VoIP se basan en elementos de hardware y software que podemos encontrar en la implantación de un sistema. Las redes VoIP se basan principalmente en los siguientes elementos:

- Red de telefonía
- Terminal IP (teléfono IP)
- Gatekeepers (centralitas IP)
- Gateway IP (puerta de enlace IP)

13.4.1 Terminal IP

Permiten conectarse directamente a la red de datos, ya que poseen al menos un puerto para establecer la conexión. Los más habituales disponen de un conector hembra RJ45, el cual se conectarán mediante un latiguillo a una toma de red de pared. Los terminales IP necesitan de alimentación eléctrica suministrada mediante la red eléctrica, aunque también puede obtener la energía eléctrica mediante la red Ethernet utilizando tecnología PoE (Power over Ethernet o Alimentación sobre Ethernet)

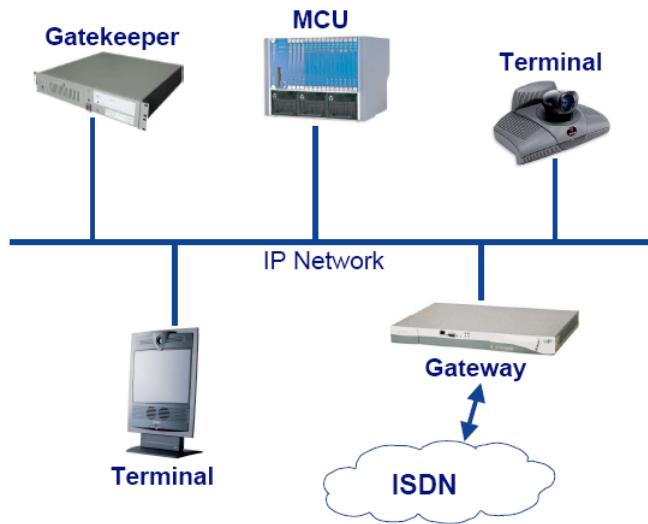


Figura 13.18: Componentes de VoIP

- Son puntos finales de la Comunicación.
- Comunicación bidireccional en tiempo real.
- Q.931 para señalización de llamada.
- H.245, negocia el uso del canal y capacidades.
- RAS (Registration/Admission/Status) ⇒ Gatekeeper.
- RTP/RTCP para secuenciar paquetes de Audio y Video.

13.4.2 Centralita IP

Las centralitas IP o IP-PBX (Private Branch eXchange o Centralita Secundaria Privada), son unas centralitas telefónicas para gestionar las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica de la RTB u otra centralita de proveedor de VoIP. La PBX se encarga de centralizar todas las conexiones, pero solo cuando existe, pues no es necesaria.

Están conectadas directamente a la RTB por medio de líneas troncales en un extremo, y a la red VoIP en otro (con gateway, router, switch, hub...).

Son un sistema informático compuesto por un servidor VoIP, con software de servidor VoIP (por ejemplo Asterix PBX para Linux, GNUGk, OpenH323...) que remplaza todas las funciones de una centralita hardware de telefonía convencional. Su función es dar servicio desde una LAN a la RTC (y viceversa) o de internet a la RTC. Una centralita evita conectar varios teléfonos de una oficina o edificio de manera separada a la red RTC, y evita la necesidad de que las comunicaciones entre teléfonos internos deban salir a la red RTC, con lo que se ahorran muchos gastos fijos y variables. Además, la PBX permite gestionar llamadas directas a todos los teléfonos asignando extensiones locales (de la red VoIP) al número público (del proveedor de telefonía tradicional o VoIP). Otra de las ventajas es que las centralitas físicas tienen una vida útil de unos 15 años, en cambio las centralitas software son para siempre.

13.4.3 Gateway

La característica principal de VoIP es poder transmitir voz a través de una red de datos, aunque a veces el destinatario no se encuentra en ella y es necesario el acceso a otro tipo de redes, como una RTC (Red Telefónica Conmutada) o una red de telefonía móvil.

La función de un gateway es posibilitar el acceso a otras redes, generalmente de carácter público,

reduciendo en ocasiones el coste de las facturas.

En una organización o empresa, estos dispositivos pueden conectarse al PBX de forma que todos los usuarios dispongan de acceso a otras redes.

FXS y FXO son los nombres de los puertos usados por las líneas telefónicas analógicas o teléfonos.

FXS – Foreign eXchange Subscriber (Interfaz de suscriptor externo): Es el puerto que envía la línea analógica al suscriptor. En otras palabras, es el enchufe de la pared que envía tono de marcado, corriente para la batería y tensión de llamada.

Gateway FXS

Los gateways FXS se usan para conectar una o más líneas de una centralita tradicional con una centralita IP o proveedor telefónico VoIP. Se necesitará un gateway FXS cuando desea conectar los puertos FXO (que normalmente se conectan a la empresa telefónica) a la Internet o centralita VoIP. **FXO – Foreign eXchange Office (Interfaz de central externa):** Es el puerto que recibe la línea analógica. Es un enchufe del teléfono, aparato de fax o el enchufe de su centralita telefónica analógica. Envía una indicación de colgado/descolgado (cierre de bucle).

Gateway FXO

Para conectar líneas telefónicas analógicas con una centralita IP, se necesita un gateway FXO. Ello le permitirá conectar el puerto FXS con el puerto FXO del gateway, que luego convierte la línea telefónica analógica en una llamada VoIP.

13.4.4 Adaptador FXS

El adaptador FXS se usa para conectar un teléfono analógico o aparato de fax a un sistema telefónico VoIP o a un prestador VoIP. Se necesitará para conectar el puerto FXO del teléfono/fax con el adaptador.

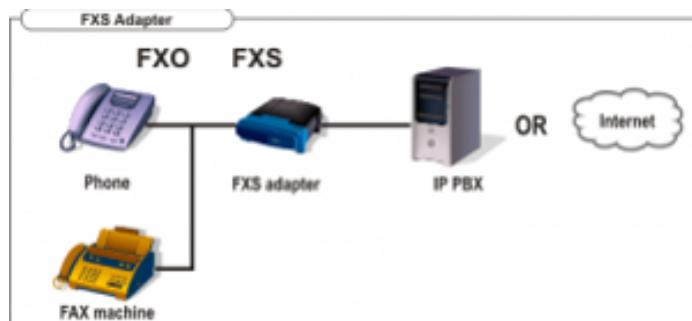


Figura 13.19: Adaptador FXS

FXO y FXS son siempre pares, es decir, similar a un enchufe macho/hembra.

13.4.5 Unidad de control MCU

Un MCU esta formado por el Multipoint Controller y por el Multipoint procesor.

1. MP: Comutación y mezclado de las tramas multimedia.
2. MC: Señalización de llamadas, control de conferencias

13.5 Tecnología IP

13.5.1 DSL

DSL (línea de suscriptor digital) es tecnología de banda ancha que proporciona transmisión de información a alta velocidad (hasta 7.1 Mbps) a través de una línea telefónica existente. Las

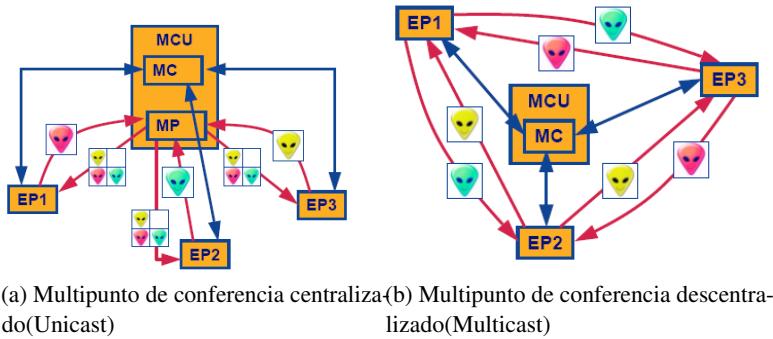


Figura 13.20: Tráfico multimedia

velocidades son hasta 50 veces más rápidas que las de un módem por dial-up estándar de 28.8 Kbps. Además, el servicio DSL es instantáneo - con sólo un clic tendrás una conexión a Internet superrápida. Está basada en utilizar la RTB para la transmisión de información a alta velocidad. Una variante de estas redes es ADSL (Asymmetric Digital Subscriber Line o Línea Digital Asimétrica de Abonado) y se llama asimétrica porque la velocidad de transmisión en un sentido es menor que en otro.

Como contrapartida, también existen las líneas SDSL (Symmetric Digital Subscriber Line o Línea Digital Simétrica de Abonado), donde la comunicación se realiza a la misma velocidad en ambos sentidos. ADSL utiliza circuitos integrados ASP (Advanced Signal Processor o Procesador de Señales Avanzado) para eliminar electrónicamente todas las interferencias producidas en la comunicación. Las redes ADSL pueden utilizar diferentes protocolos de transporte de datos, como TCP/IP o ATM. ATM permite la difusión de películas y la videoconferencia de alta calidad.

13.5.2 RDSI

La red RDSI dispone de su propio cableado, permitiendo no sólo la transmisión de voz, sino de datos, de imagen y sonido en tiempo real, etc. Para permitir diferentes tipos de transmisión, se utiliza la técnica de multiplexación creando canales independientes para ser transmitidos por un único medio. Cuando se contrata una conexión RDSI, los terminales a utilizar deben ser digitales, aunque también se puede utilizar un dispositivo denominado TA (Terminal Adapter o Adaptador de Terminal) para conectar terminales analógicos, realizando la conversión analógico-digital de las señales.

13.5.3 LMDS

LMDS (Local Multipoint Distribution System o Sistema Local de Distribución Multipunto) es un sistema que utiliza el aire como medio de transmisión para la comunicación desde una estación base a los abonados del servicio. Permite la transmisión de voz, datos, Internet y vídeo mediante la emisión de ondas radioeléctricas de alta frecuencia (28 - 40Ghz), y se suele utilizar para cubrir zonas geográficas donde es difícil el acceso mediante cable. A mayor frecuencia las velocidades son más altas (hasta 8Mbps), pero la zona de cobertura es menor.

13.5.4 GSM

GSM (Global System for Mobile communications o Sistema Global para las Comunicaciones Móviles) es un estándar que utiliza la commutación de circuitos para la comunicación entre teléfonos móviles, y que realiza la transmisión de voz y señalización en formato digital. Para su funcionamiento se utiliza una red compleja de estaciones base o BS (Base Station) que dan cobertura a un

área geográfica determinada, denominada celda o célula. Los terminales de usuario buscan siempre una estación base a la que conectarse utilizando una banda de frecuencia comprendido entre 900 MHz y 1800 MHz.

13.6 Vocabulary

Vocabulario 13.1 — Congestión. Es cuando todos los circuitos capaces de cursar tráfico en una central están ocupados.

VII

Instrumentación 1



A photograph of a complex industrial piping system. The pipes are primarily blue and yellow, with various valves, fittings, and sensors attached. The pipes are mounted on a white metal structure, likely a support beam or a wall. The background is slightly blurred, showing more of the industrial environment.

14. Unidad I

IDK

Anexos

- 14.1 Ejercicios resueltos U2-Sistemas de telefonía
- 14.2 Ejercicios resueltos U2-Sistemas de telefonía
- 14.3 Ejercicios resueltos U2-Sistemas de telefonía
- 14.4 Ejercicios resueltos U2-Sistemas de telefonía 2
- 14.5 Ejercicios resueltos U2-Sistemas de telefonía 2
- 14.6 Ejercicios resueltos U2-Sistemas de telefonía 2
- 14.7 Ejercicios resueltos U2-Encaminamiento de tráfico telefónico
- 14.8 Ejercicios resueltos U2-Encaminamiento de tráfico telefónico
- 14.9 Ejercicios resueltos U2-Encaminamiento de tráfico telefónico
- 14.10 Ejercicios resueltos U2-Erlang B
- 14.11 Ejercicios resueltos U2-Erlang B
- 14.12 Ejercicios resueltos U2-Erlang B

14.1 Ejercicios resueltos U2-Sistemas de telefonía

1. Problemas propuestos

- 1.1. Una oficina realiza un promedio de 43 llamadas al día, cada llamada tiene una duración en promedio de 8 minutos, si el horario de oficina es de 8:00 am a 4:00 pm y se asume que las llamadas se realizan de manera proporcional en todo el horario, calcular la intensidad de tráfico.

Datos:

Número de llamadas(n): 43 llamadas/día

Tiempo promedio(t_p): 8min

Tiempo de observación(t_0)=8 horas

Resolución

$$V_t = n * t$$

$$V_t = 43 * 8$$

$$V_t = 344 \text{ min} \quad (1)$$

$$I_t = \frac{V_t}{t_o}$$

$$I_t = \frac{344 \text{ min}}{8 \text{ hours}} * \frac{1 \text{ hour}}{60 \text{ min}}$$

$$I_t = 0,716E \quad (2)$$

- 1.2. Determinar la intensidad de tráfico, si una línea está ocupada durante una hora y 100 usuarios solicitan una llamada con una duración promedio de 3 minutos.

Datos:

Número de llamadas(n): 100 llamadas

Tiempo promedio(t_p): 3min

Tiempo de observación(t_0)=1 horas

Resolución

$$V_t = n * t$$

$$V_t = 100 * 3$$

$$V_t = 300 \text{ min} \quad (3)$$

14.2 Ejercicios resueltos U2-Sistemas de telefonía

$$\begin{aligned}
 I_t &= \frac{V_t}{t_o} \\
 I_t &= \frac{300min}{1hours} * \frac{1hour}{60min} \\
 I_t &= 5Erlangs
 \end{aligned} \tag{4}$$

- 1.3. Consideremos un centro de atención al cliente ONLINE vía telefónica. El tiempo completo de la fase de marcado se asume a ser de 20 s, incluye la invitación a marcar y el anuncio de todas las opciones. Se esperan 10.000 llamadas en la hora pico. Calcular la intensidad de tráfico.

Datos:

Número de llamadas(n): 10000 llamadas

Tiempo promedio(t_p): 20 seg

Tiempo de observación(t_0)=1 horas

Resolución

$$\begin{aligned}
 V_t &= n * t \\
 V_t &= 10000 * 20seg * \frac{1min}{60seg} \\
 V_t &= 3333,3min \\
 I_t &= \frac{V_t}{t_o} \\
 I_t &= \frac{3333,3min}{1hours} * \frac{1hour}{60min} \\
 I_t &= 55,555Erlangs
 \end{aligned} \tag{5}$$

- 1.4. En un grupo de 25 líneas se cursaron 150 llamadas en media hora, con un promedio de duración de 3 minutos. Calcular el tráfico total y por línea.

Datos:

Número de llamadas(n): 150 llamadas

Tiempo promedio(t_p): 3min

Tiempo de observación(t_0)=0.5 horas

Líneas: 5

Resolución

14.3 Ejercicios resueltos U2-Sistemas de telefonía

$$V_t = n * t$$

$$V_t = 150 * 3$$

$$V_t = 450 \text{ min} \quad (7)$$

$$I_t = \frac{V_t}{t_o}$$

$$I_t = \frac{450 \text{ min}}{0,5 \text{ hours}} * \frac{1 \text{ hour}}{60 \text{ min}}$$

$$I_t = 15E \quad (8)$$

$$Traffic_{line} = \frac{I_t}{N_l}$$

$$Traffic_{line} = \frac{15E}{25} \quad (9)$$

$$Traffic_{line} = 0,6E \quad (10)$$

14.4 Ejercicios resueltos U2-Sistemas de telefonía 2

Problem 1

Considerar un grupo de 1200 abonados que generan 600 llamadas durante la hora cargada. El tiempo medio de espera es de 2.2 minutos. ¿Cuál es el tráfico ofrecido en erlangs, LLR y CCS.

Solution

- a) El tráfico ofrecido está definido por:

$$V_t = n * d \quad (1)$$

Usando la ecuación 1:

$$V_t = 600 * 2,2;$$

$$V_t = 1320 \text{ min} \rightarrow 22 \text{ Horas}$$

La intensidad, descrita como:

$$I_t = \frac{V_t}{T_0} \quad (2)$$

$$I_t = A_0 = \frac{22}{1}$$

- b) En LLR:

$$1 \text{ Erlang} = 33 \text{ LLR} \quad (3)$$

$$A_0 = 22 * 30$$

$$A_0 = 660 \text{ LLR}$$

- c) En CCS:

$$1 \text{ Erlang} = 36 \text{ CCS} \quad (4)$$

$$A_0 = 22 * 36$$

$$A_0 = 792 \text{ CCS}$$

Problem 2

Durante la hora cargada, se ofrecen 1200 llamadas a un enlace de transmisión, pero seis de ellas no se transmiten (se pierden). El tiempo medio de duración de la llamada es de 3 minutos. Calcular:

- a) el tráfico ofrecido
- b) el tráfico cursado
- c) el tráfico perdido

14.5 Ejercicios resueltos U2-Sistemas de telefonía 2

Solution

a) Usando la ecuación 1 entonces:

$$V_t = 1200 * 3$$

$$V_t = 3600\text{min} \rightarrow 60\text{horas}$$

Con la ecuación 2:

$$I_t = \frac{60}{1}$$

$$I_t = A_o = 60 \text{ Erlangs}$$

b) Usando la ecuación 1 entonces:

$$V_t = 1194 * 3$$

$$V_t = 3582\text{min} \rightarrow 59,7\text{horas}$$

Con la ecuación 2:

$$I_t = \frac{59,7}{1}$$

$$I_t = 59,7 \text{ Erlangs}$$

$$A_c = 59,7 \text{ Erlangs}$$

c) Usando la ecuación 1 entonces:

$$V_t = 6 * 3$$

$$V_t = 18\text{min} \rightarrow 3\text{horas}$$

Con la ecuación 2:

$$I_t = \frac{0,3}{1}$$

$$I_t = 0,3 \text{ Erlangs}$$

Problem 3

Durante la hora cargada, se ofrecieron 1400 llamadas a un grupo de troncales y se perdieron 14. La duración media de las llamadas es de 3 minutos. Encontrar:

- a) el tráfico ofrecido
- b) el tráfico cursado
- c) GoS
- d) duración del periodo de congestión

14.6 Ejercicios resueltos U2-Sistemas de telefonía 2

Solution

a)

$$V_T = 1400 * 3$$

$$V_T = 4200 \text{min} \leftrightarrow 70 \text{Horas}$$

$$I_t = \frac{70}{1} = 70 \text{ ERLANGS}$$

b) Usando la ecuación 1

$$V_T = 14 * 3$$

$$V_T = 42 \text{min} \leftrightarrow 0,7 \text{Horas}$$

Aplicando la ecuación 2

$$I_t = \frac{0,7}{1} = 70 \text{ ERLANGS}$$

$$A_c = 70 \text{Erlangs} - 0,7 \text{Erlangs} = 69,3 \text{Erlangs}$$

c) Tenemos dos datos importantes: 14(tráfico perdido) y 1400(tráfico ofrecido)

$$GoS = \frac{A_p}{A_o} \quad (5)$$

reemplazando los datos en 5

$$GoS = \frac{14}{1400}$$

$$GoS = 0,01$$

d) El congestionamiento esta dado por:

$$\text{Congestionamiento} = GoS * T_0 \quad (6)$$

Reemplazando en 6:

$$\text{Congestionamiento} = 0,01 * 3600s$$

$$\text{Congestionamiento} = 36s$$

Problem 4

La capacidad de un help desk se ha diseñado para ofrecer un GoS de 2% durante la hora cargada. El número total de llamadas ofrecidas durante este intervalo de tiempo es de 300 (solicitudes de ayuda).

a) ¿Cuál es el número total de llamadas perdidas?

b) ¿Durante cuántos minutos está congestionado el servicio de asistencia durante la hora cargada?

14.7 Ejercicios resueltos U2-Encaminamiento de tráfico telefónico

Problem 1

Para la red telefónica de la figura y en virtud del Plan de Encaminamiento, llenar la tabla siguiente:

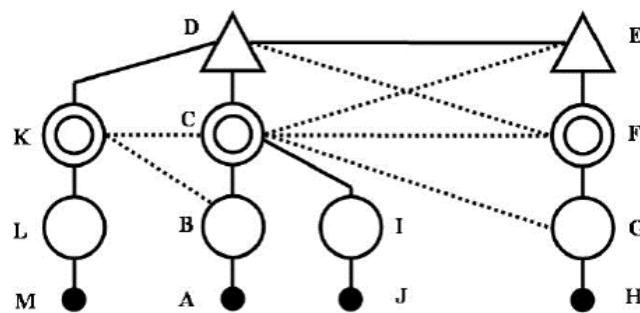


Figura 1: Problema 1

Solution

Central Origen	Central Destino	Ruta directa	Ruta alternativa 1	Ruta alternativa 2	Ruta final
A	H	ABCGH	ABCDFGH	-	ABCDEFGH
H	A	HGCBA	HGFBCBA	HGFEDCBA	HGFEDCBA
M	H	MLKDFGH	MLKDFGH	-	MLKDEFGH
H	M	HGFDKLM	HGFCKLM	HGFECKLM	HGFEDKLM
A	M	ABKLM	ABCKLM	-	ABCDKLM
M	A	MLKBA	-	-	MLKDCBA
M	J	MLKCIJ	-	-	MLKDCIJ

Problem 2

Para la red telefónica de la figura y en virtud del Plan de Encaminamiento, llenar la tabla siguiente:

14.8 Ejercicios resueltos U2-Encaminamiento de tráfico telefónico

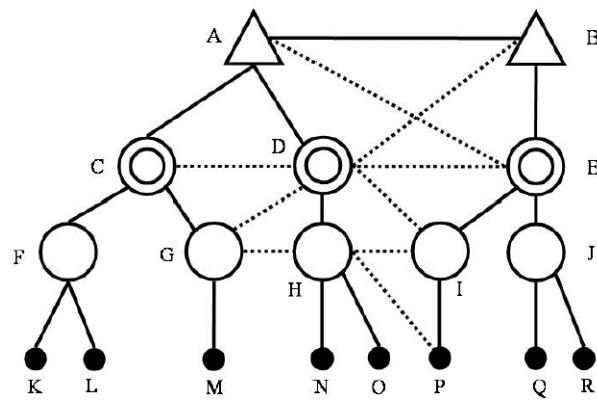


Figura 2: Problema 2

Solution

	De N a P	De P a N
Ruta directa	NHP	PHN
Ruta alternativa 1	NHDIP	PIHN
Ruta alternativa 2	NHDAEIP	PIEDHN
Ruta alternativa 3	-	PIEBDHN
Ruta final	NHDABEIP	PIEBADHN

Problem 3

Indicar las rutas posibles con origen en A y destino en la central K.

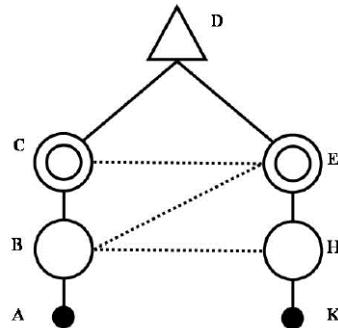


Figura 3: Problema 3

14.9 Ejercicios resueltos U2-Encaminamiento de tráfico telefónico**Solution**

Ruta	De A a K
Ruta directa	ABHK
Ruta alternativa 1	ABCEHK
Ruta final	ABCDEHK

14.10 Ejercicios resueltos U2-Erlang B

Problem 1

Consideremos un grupo troncal con una carga ofrecida de 4.5 erlangs y una probabilidad de bloqueo de 0.01.

Si el tráfico ofrecido aumenta a 13 erlangs, para mantener la misma probabilidad de bloqueo a) Encuentre el número de troncales necesarias. b) Calcule también la ocupación de las troncales.

Solution

a) Encuentre el número de troncales necesarias.

Para ellos usamos la ecuación:

$$B(N, A) = \frac{A^N}{N! \sum_{k=0}^N \left(\frac{A^k}{k!}\right)} \quad (1)$$

Como tenemos que averiguar el número de troncales, crearemos un código sencillo para esto, donde N será la variable independiente. Usando su gráfica, nos movemos hacia una probabilidad de 0.01, apreciamos que necesitamos **aproximadamente 22 troncales**¹.

b) Calcule también la ocupación de las troncales. Para ello usamos la siguiente ecuación:

$$\rho = \frac{A(1 - B(N, A))}{N} \quad (2)$$

Reemplazando en la ecuación 2

$$\begin{aligned} \rho &= \frac{A(1 - B(N, A))}{N} \\ &= \frac{13(1 - 0,01)}{22} \\ &= 0,585 \end{aligned} \quad (3)$$

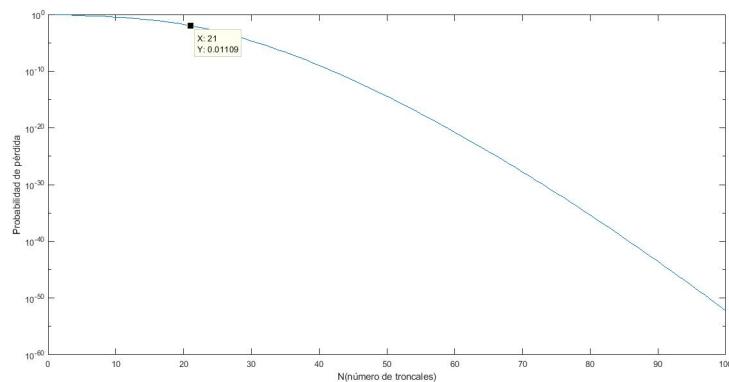


Figura 1: Análisis gráfico

¹Es mejor tener sistemas de redundancia

14.11 Ejercicios resueltos U2-Erlang B

```

1  for i=1:length(N)
2      s=0;
3      for k=0:i
4          g=(13^k)/factorial(k);
5          s=s+g;
6      end
7      B(i)=(13^N(i))/(factorial(N(i))*s);
8  end
9 semilogy(N,B)

```

Código 1: Código 1

Problem 2

A un grupo de 7 troncales se le ofrecen 4E de tráfico, hallar: **a)** El grado de servicio **b)** La probabilidad de que sólo una troncal esté ocupada **c)** La probabilidad de que sólo una troncal esté libre **d)** La probabilidad de que al menos una troncal esté libre.

Solution

a) El grado de servicio

El grado de servicio es la probabilidad de bloqueo, usando la ecuación 1:

$$\begin{aligned}
 B(N, A) &= \frac{A^N}{N! \sum_{k=0}^N \left(\frac{A^k}{k!}\right)} \\
 &= \frac{4^7}{7! \sum_{k=0}^7 \left(\frac{4^k}{k!}\right)} \\
 &= 0,062
 \end{aligned} \tag{4}$$

b) La probabilidad de que sólo una troncal esté ocupada

$$\begin{aligned}
 P(1) &= \frac{\frac{4^1}{1!}}{\sum_{k=0}^7 \frac{4^k}{k!}} \\
 &= 0,077
 \end{aligned} \tag{5}$$

c) La probabilidad de que sólo una troncal esté libre

$$\begin{aligned}
 P(6) &= \frac{\frac{4^6}{6!}}{\sum_{k=0}^7 \frac{4^k}{k!}} \\
 &= 0,1172
 \end{aligned} \tag{6}$$

14.12 Ejercicios resueltos U2-Erlang B

d) La probabilidad de que al menos una troncal esté libre.

$$\begin{aligned} P(x < 5) &= 1 - P(7) \\ &= 1 - GoS \\ &= 1 - 0,063 \\ &= 0,937 \end{aligned} \tag{7}$$

IX

Extras

15 Artificios matemáticos	227
Bibliography	229
Articles	
Books	
Index	231

15. Artificios matemáticos

En esta sección se detallaran trucos, repasos, propiedades que son vitales saberlas pero, típico del ser humano, a veces se nos olvida.

$$e^{\ln|x|} = x \quad (15.1)$$

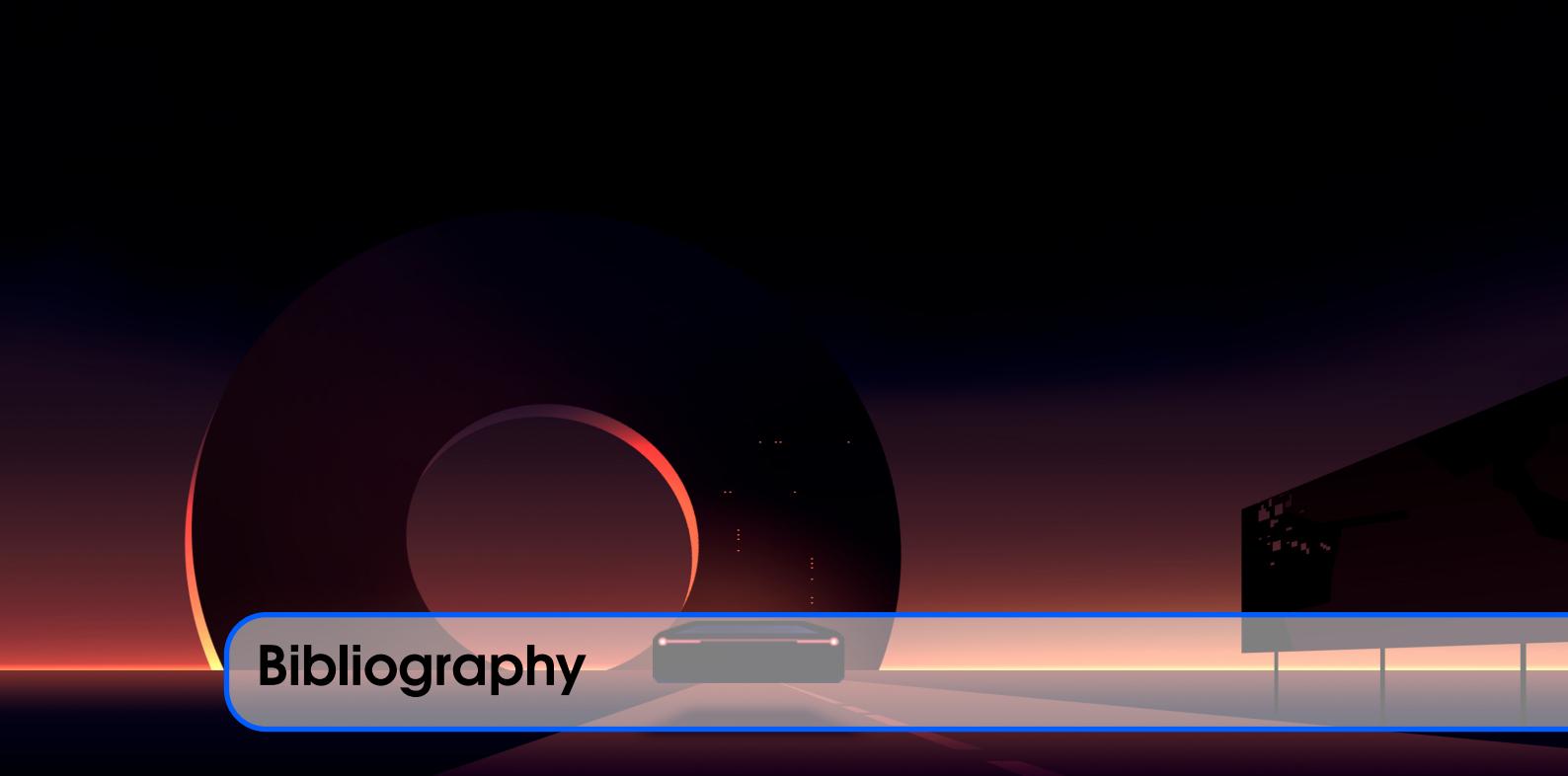
$$\ln(e^x) = x \frac{\ln(e)}{1} = x \quad (15.2)$$

$$\log_a(b) = b \cdot \log a \quad (15.3)$$

$$\log_c(a) - \log_c(b) = \log_c\left(\frac{a}{b}\right) \quad (15.4)$$

$$\log(a)^b = b \cdot \log(a) \quad (15.5)$$

$$\log(\sqrt[b]{a}) = \frac{\log(a)}{b} \quad (15.6)$$



Bibliography

Articles

- [1] James Smith. "Article title". En: 14.6 (mar. de 2013), páginas 1-8.

Books

- [2] John Smith. *Book title*. 1.^a edición. Volumen 3. 2. City: Publisher, ene. de 2012, páginas 123-200.

Índice alfabético

C

Citation	8
Corollaries	10

D

Definitions	9
-------------------	---

E

Examples	10
Equation and Text.....	10
Paragraph of Text	11
Exercises.....	11

F

Figure	15
--------------	----

L

Lists	8
Bullet Points	8
Descriptions and Definitions	8
Numbered List.....	8

N

Notations	10
-----------------	----

P

Paragraphs of Text	7
Problems	11
Propositions	10
Several Equations.....	10
Single Line.....	10

R

Remarks	10
---------------	----

T

Table	15
Theorems	9
Several Equations.....	9
Single Line.....	9

V

Vocabulary	11
------------------	----