

## ARQUITECTURA DEL COMPUTADOR

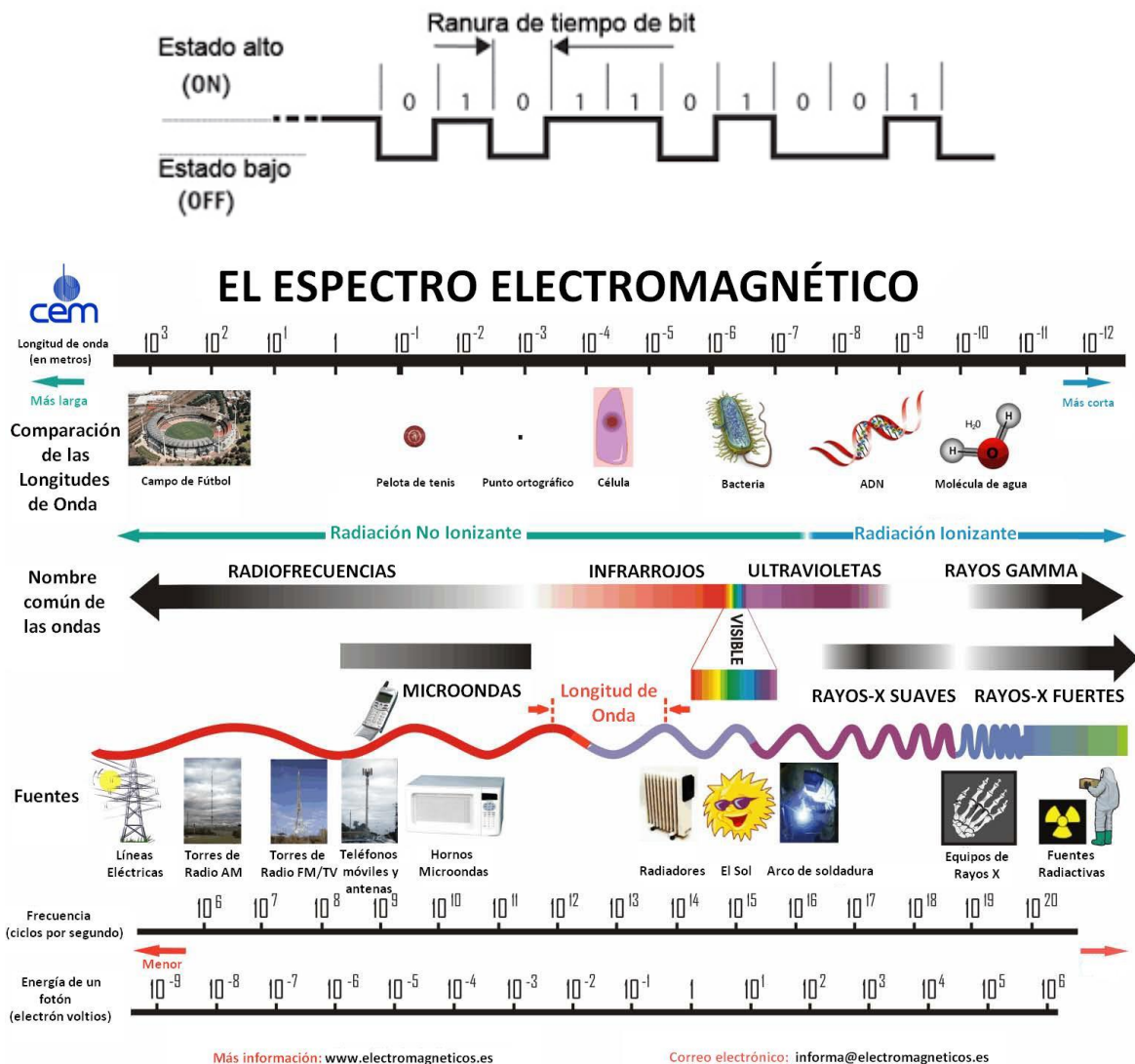
### bit

La palabra bit fue utilizada por primera vez en la década de los 30, sorprendentemente, para designar partes de información (bits of information). Simplificando, un bit es exactamente eso: una combinación de dos dígitos que se junta con otros dígitos del mismo tipo para generar la información completa.

Leer <https://www.bell-labs.com/claude-shannon/>

USA (30/abril/1916 – 24/febrero/2001) [https://es.wikipedia.org/wiki/Claude\\_Elwood\\_Shannon](https://es.wikipedia.org/wiki/Claude_Elwood_Shannon)

Toda la memoria del ordenador se compone de dispositivos electrónicos que pueden adoptar únicamente dos estados, que representamos matemáticamente por 0 y 1. Cualquiera de estas unidades de información se denomina **bit**, contracción de «**binary digit**» en inglés.

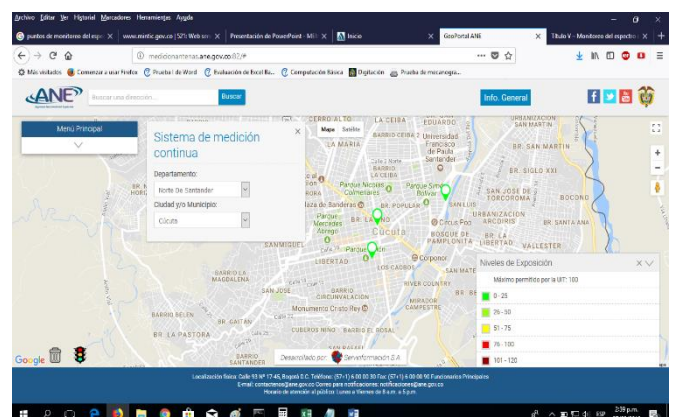
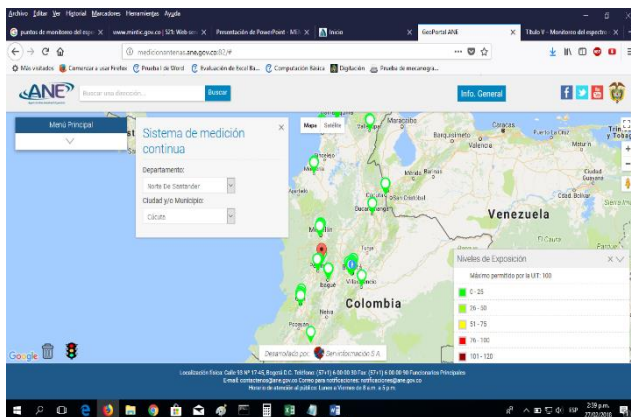


### IMPORTANTE espectro en COLOMBIA

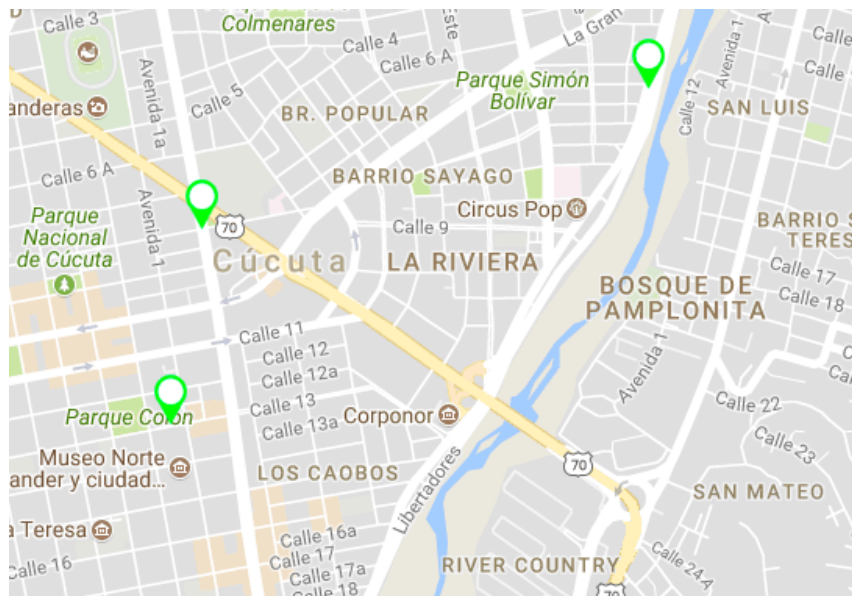
Las frecuencias del espectro electromagnético usadas para los servicios de difusión y servicios móviles, de policía, bomberos, radioastronomía, meteorología y fijos.” Este “(...) no es un concepto estático, pues a medida que avanza la tecnología se aumentan (o disminuyen) rangos de frecuencia utilizados en comunicaciones, y corresponde al estado de avance tecnológico.

<http://www.ane.gov.co/index.php>

### Puntos de Monitoreo del Espectro en Colombia (CNM centro nacional de monitoreo)



<http://medicionantenas.ane.gov.co:82/>



PRADOS	LATINO	LA PLAYA
Ciudad: Cúcuta	Ciudad: Cúcuta	Ciudad: Cúcuta
Dirección: Avenida Libertadores 6AN-09	Dirección: Avenida 0 Calle 8	Dirección: Calle 13 1-57
Fecha Ultima Medición: 2017-12-04 21:00:00	Fecha Ultima Medición: 2018-02-07 09:24:00	Fecha Ultima Medición: 2017-12-27 01:24:00
Valor Ultima Medición: 0.37	Valor Ultima Medición: 2.4	Valor Ultima Medición: 1.09

**REPRESENTACIÓN DE UN BIT (MEDIDAS)****NIBBLE:** 4 bits**BYTE**

Cada grupo de 8 bits se conoce como byte u octeto. Es la unidad de almacenamiento en memoria, la cual está constituida por un elevado número de posiciones que almacenan bytes. La cantidad de memoria de que dispone un sistema se mide en: **(24 Oct 1922 – Aleman - Leer Werner Buchholz en 1957) “Mordisco”**

**1 Byte = 4 bits para el IBM 7030**



- Kilobytes (1 Kb = 1024 bytes)
- Megabytes (1 Mb = 1024 Kb)
- Gigabytes (1 Gb = 1024 Mb)
- Terabytes (1 Tb = 1024 Gb)
- Petabytes (1 Pb = 1024 Tb).
- Exabytes (1 Eb = 1024 Pb).
- Zetabytes (1 Zb = 1024 Eb).
- Yotabytes (1 Yb = 1024 Zb).
- Brontobytes = (1Bb = 1024 Yb)
- GeopBytes = (1 Gpb = 1024 Bb)
- Saganbytes = (1 Sb = 1024 Gbp)
- Jotabyte = ( 1Jb = 1024 Sb )

Unidades básicas de información (en bytes)				
Prefijos del Sistema Internacional			Prefijo binario	
Múltiplo - (Símbolo)	Estándar SI	Binario	Múltiplo - (Símbolo)	Valor
kilobyte (kB)	$10^3$	$2^{10}$	kibibyte (KiB)	$2^{10}$
megabyte (MB)	$10^6$	$2^{20}$	mebibyte (MiB)	$2^{20}$
gigabyte (GB)	$10^9$	$2^{30}$	gibibyte (GiB)	$2^{30}$
terabyte (TB)	$10^{12}$	$2^{40}$	tebibyte (TiB)	$2^{40}$
petabyte (PB)	$10^{15}$	$2^{50}$	pebibyte (PiB)	$2^{50}$
exabyte (EB)	$10^{18}$	$2^{60}$	exbibyte (EiB)	$2^{60}$
zettabyte (ZB)	$10^{21}$	$2^{70}$	zebibyte (ZiB)	$2^{70}$
yottabyte (YB)	$10^{24}$	$2^{80}$	yobibyte (YiB)	$2^{80}$

**PALABRA:** 2 Bytes o a partir de la arquitectura del PC se puede decir que es 4 u 8 Bytes, para nuestra materia sería 16 bits, o 32 o 64 bits.

**PALABRA DOBLE:** 4 Bytes**PALABRA CUADRUPLE:** 8 Bytes**PÁRRAFO:** 16 Bytes**IMPORTANTE LEER:**

<https://cs.stackexchange.com/questions/67684/does-a-byte-contain-8-bits-or-9/67688>



**SISTEMAS NUMERICOS**

EGIPCIO HIERATICO	I	U	III	W	Y	Z	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474</
----------------------	---	---	-----	---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	--------

Otros ejemplos:  $12_{(10)}$  ,  $23456_{(10)}$

Los números decimales se clasifican en unidades, decenas, centenas, etc por tanto siempre leemos de izquierda a derecha mas no su valor de posición el cual se lee de derecha a izquierda.

Ejemplo:  $694_{(10)}$  Descripción en notación expandida sin valor decimal solo potencias.

$6 = 10^2$   $9 = 10^1$   $4 = 10^0$  por tanto los valores de posición son:

Para el número 6 es de  $10^2$  , para el número 9 es de  $10^1$  y para el número 4 es de  $10^0$

Valor facial es aquel valor del número decimal en si, ósea el mismo número pero sin base.

Leyes de potencia

Todo número elevado a la potencia 1 da como resultado el mismo número de base.

Todo número elevado a una potencia 0 da como resultado uno.

Operaciones: Suma, Resta, Multiplicación y División.

### **SISTEMAS BINARIO (SISTEMA DIADICO)**

Sistema utilizado por todos los computadores, debido a que toda la información la maneja a través de código binario el cual conforma el lenguaje de máquina (Assembler o ensamblador). Trabaja en base 2, por tanto utiliza solo dos dígitos para representarse

El cero (0) y el uno (1).

Este sistema es el manejado además por dispositivos eléctricos, mecánicos, electrónicos, etc, representando dos estados posibles en cualquier sistema (abierto y cerrado) ON/OFF.

Sistema abierto es todo aquel que no conduce o no permite el flujo de corriente eléctrica, mientras que un circuito cerrado si los permite, debido a que todos sus dispositivos se encuentran en línea.

Ejemplo:  $10011_{(2)} = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1$   
 $16 + 0 + 0 + 2 + 1$   
 $19_{(10)}$

Los valores de posición son:  $2^4$   $2^3$   $2^2$   $2^1$   $2^0$

Otros ejemplos:  $111_{(2)}$  ,  $1000_{(2)}$

*Tabla de potencias*

$2^0$	$2^1$	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	$2^7$	$2^8$	$2^9$	$2^{10}$
1	2	4	8	16	32	64	128	256	512	1024

Tipo	Sin signo
1 byte	255
2 bytes	65.535
4 bytes	4.294.967.295
8 bytes	18.446.744.073.709.551.615

Tipo	Positivo	Negativo
1 byte	127	-128
2 bytes	32.767	-32.768
4 bytes	2.147.483.647	-2.147.483.648
8 bytes	9.223.372.036.854.775.807	-9.223.372.036.854.775.808

**Operaciones****SUMA**

Tabla de adición binaria

0+0=0	0+1=1	1+0=1	1+1=0 llevo 1	1+1+1=1 llevo 1
-------	-------	-------	---------------	-----------------

Ejemplos:  $10111_{(2)} + 110_{(2)} = 11101_{(2)}$  $11101_{(2)} + 1111_{(2)} = 101100_{(2)}$ Ejercicios  $11011111_{(2)} + 10000_{(2)} =$  $10011111_{(2)} + 1110_{(2)} =$  $10111_{(2)} + 110_{(2)} + 11101_{(2)} =$ **RESTA**

Tabal de sustracción binaria

0-0=0	0-1=1 y resto 1 a la izquierda	1-0=1	1-1=0
-------	--------------------------------	-------	-------

Para realizar la resta binaria se puede hacer de dos formas, por el método tradicional o por el método de complementos.

*Método Tradicional*Ejemplos:  $10111_{(2)} - 110_{(2)} = 10001_{(2)}$  $11101_{(2)} - 1111_{(2)} = 1110_{(2)}$ Ejercicios  $11011111_{(2)} - 10000_{(2)} =$  $10011111_{(2)} - 1110_{(2)} =$  $10111_{(2)} - 110_{(2)} =$ *Método Complementos*

Para realizar este método debemos seguir los siguientes pasos:

- Hallar el complemento del sustraendo (Cambiar unos por ceros y ceros por unos).
- Sumar el minuendo con el complemento del sustraendo.
- Transferir el primer uno que se encuentre más a la izquierda (en caso sobrante) y lo sumamos a la derecha.

Ejemplos:

 $10111_{(2)} - 110_{(2)} = 10001_{(2)}$ Paso A Como faltan números para igualar tamaño rellenamos de ceros  $00110 = 11001$ Paso B  $10111 + 11001 = 110000$ Paso C  $10000 + 1 = 10001$  $11101_{(2)} - 1111_{(2)} = 1110_{(2)}$ Paso A  $01111 = 10000$ Paso B  $11101 + 10000 = 101101$ Paso C  $01101 + 1 = 1110$ Ejercicios  $11011111_{(2)} - 10000_{(2)} =$  $10011111_{(2)} - 1110_{(2)} =$  $10111_{(2)} - 110_{(2)} =$ **MULTIPLICACIÓN**

Se realiza de forma idéntica al sistema decimal a excepción de la suma de los resultados de la multiplicación la cual se debe realizar en forma binaria.

 $10111_{(2)} * 110_{(2)} = 10001010_{(2)}$ **DIVISIÓN**

Se realiza a través de restar sucesivas del divisor, ya que el único digito no cero es uno.

$$1010001_{(2)} / 11_{(2)} = 11011_{(2)}$$

### **NÚMEROS NEGATIVOS BINARIOS**

Un número negativo posee el bit más a la izquierda con valor de 1, el cual indica su estado o valor (+/-). Para esto no es que a todo número le cambiemos su valor del bit izquierdo para ser negativo ya que  $11111111_{(2)}$  es 255 y no es negativo, para saber si es negativo debemos basarnos en el complemento a dos. Este método consiste en invertir todos los números y sumarle un 1.

Ejemplo:  $0.11111111_{(2)} = 255$   
 Invertido  $0.00000000_{(2)}$   
 Sumamos 1  $1_{(2)}$   
 Resultado  $1.00000001_{(2)} = -255$  algo parecido a  $256 - 1 = 255$

Tabla de los números negativos

$00000011 = +3$   
 $00000010 = +2$   
 $00000001 = +1$   
 $00000000 = 0$   
 $11111111 = -1$   
 $11111110 = -2$   
 $11111101 = -3$   
 $11111100 = -4$   
 $11111011 = -5$

CASO IMPORTANTE es el número 128, por ejemplo:

128 en 1 Bytes es igual a  $10000000_{(2)}$

Si lo queremos convertir a -128 tendríamos:

Primero: Complemento 1 de 128 que es igual a  $01111111_{(2)}$

Segundo: Complemento 2 del mismo número, que sería sumarle 1 a este resultado del paso

Primero, por lo tanto

$01111111_{(2)}$

+  $1_{(2)}$

$10000000_{(2)}$  que es -128, siendo este resultado igual a 128 positivo, por lo tanto se puede diferenciar estos dos número pero con un tamaño más grande de 2 Bytes. Por ejemplo:

$00000000 \ 10000000_{(2)}$   
 Complemento 1:  $11111111 \ 01111111_{(2)}$   
 Complemento 2: +  $1_{(2)}$   


---

 $11111111 \ 10000000_{(2)}$

En caso de Octal y Hexadecimal sería de la siguiente manera

-  $35_{(8)}$  el 3 le presta 8 unidades al 5 para convertir en 13 y así poder restarle 7  
 $77_{(8)}$  luego el 3 pasa a ser 2 a quien se le debe restar 7 pero como no tiene quien le preste el 0 pasa a ser 7

$77736_{(8)}$  RTA que es un número negativo por el 7 con que inicia el número, serían tantos 7 como el tamaño de la variable que la guarda lo permita

### **MÉTODO OCTAL**

Sistema numérico que trabaja en base 8, por tanto utiliza solo 8 dígitos para representarse

Entre el cero (0) y el siete (7).

La cantidad de dígitos representados el cual es igual a 8 hace representativo este número e potencias de 2 como 2 elevado a la tres, por tanto su representación en sistema binario hace que sea en grupo de tres bits.

Sistema octal	Sistema binario	Sistema octal	Sistema binario
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Ejemplo:  $21341_{(8)} = 2 \cdot 8^4 + 1 \cdot 8^3 + 3 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 = 2 \cdot 4096 + 1 \cdot 512 + 3 \cdot 64 + 4 \cdot 8 + 1 \cdot 1$   
 $8192 + 512 + 192 + 32 + 1$   
 $8929_{(10)}$

#### Tabla de potencias

$8^0$	$8^1$	$8^2$	$8^3$	$8^4$	$8^5$	$8^6$
1	8	64	512	4096	32768	262144

#### Operaciones

##### SUMA

Se realiza sumando los dos números a trabajar, teniendo el resultado de la anterior suma le restamos a ese resultado el múltiplo (no la potencia) de la base octal que se aproxime más a esta cantidad sin sobrepasar su valor, y el valor a que se lleve el múltiplo de la base octal lo sumamos al operando del lado derecho de la operación.

Ejemplos:  $75462_{(8)} + 2564_{(8)} = 100246_{(8)}$   
 $456612_{(8)} + 335264_{(8)} = 1014076_{(8)}$   
 Ejercicios  $653211_{(8)} + 15642_{(8)} = 671053_{(8)}$

##### RESTA

Para realizar la resta octal se puede hacer de dos formas, por el método tradicional o por el método de complementos.

##### Método Tradicional

Ejemplos:  $67452_{(8)} - 3215_{(8)} = 64235_{(8)}$  Cuando se va a prestar se presta una cantidad, ocho unidades  
 $56743_{(8)} - 42531_{(8)} = 14212_{(8)}$

##### Método Complementos

Para realizar este método debemos seguir los siguientes pasos:

- Le resto a un número de 7 igual al sustraendo, y luego al resultado le sumamos 1 para obtener el complemento del sustraendo.
- Sumar el minuendo con el complemento del sustraendo.
- Eliminamos el primer uno que se encuentre más a la izquierda (en caso sobrante).

Ejemplos:

$$65423_{(8)} - 4445_{(8)} = 10001_{(8)}$$

Paso A  $7777 - 4445 = 3332 + 1 = 3333$

Paso B  $65423 + 3333 = 60756$

#### MULTIPLICACIÓN

Se realiza de forma idéntica al sistema decimal a excepción de la suma de los resultados de la multiplicación la cual se debe realizar en forma octal y teniendo en cuenta que al multiplicar el valor resultante no existe en este sistema se debe restar el múltiplo mas cercano y sumar la posición del múltiplo al siguiente valor.

$$7456541_{(8)} \cdot 235_{(8)} = 2247642175_{(8)}$$

#### DIVISIÓN

Se realiza a través de la conversión al sistema decimal del dividendo momentáneo y el divisor general, en la cual el valor obtenido en el cociente de la división decimal es el valor que se coloca como cociente de



la división octal. Este Cociente lo multiplicamos por el divisor octal aplicando las propiedades del multiplicación y dicho valor obtenido se lo restamos al dividendo aplicando también las propiedades de la resta octal.

$$7654_{(8)} / 25_{(8)} = 277_{(8)}$$

$$4567_{(8)} / 52_{(8)} = 71_{(8)}$$

1.  $76=62$   $25=21$   $62/21=2$
2.  $25*2=52$   $76-52=24$
3.  $245=165$   $165/21=7$
4.  $25*7= 223$   $245-223=22$
5.  $224= 148$   $148/21=7$
6.  $25*7= 223$   $224-223=001$

### MÉTODO HEXADECIMAL

Sistema numérico que trabaja en base 16, por tanto utiliza solo 16 dígitos para representarse

Entre el cero (0) - nueve (9) y A hasta la F.

La cantidad de dígitos representados el cual es igual a 16 hace representativo este numero e potencias de 2 como 2 elevado a la cuatro, por tanto su representación en sistema binario hace que sea en grupo de cuatro bits.

$$\begin{aligned} \text{Ejemplo: } 2351_{(16)} &= 2*16^3 + 3*16^2 + 5*16^1 + 1*16^0 = 2*4096 + 3*256 + 5*16 + 1*1 \\ &8192 + 768 + 80 + 1 \\ &9041_{(10)} \end{aligned}$$

### Operaciones

#### SUMA

Se realiza sumando los dos números a trabajar, teniendo el resultado de la anterior suma le restamos a ese resultado el múltiplo (no la potencia) de la base hexadecimal que se aproxime más a esta cantidad sin sobrepasar su valor, y el valor a que se lleve el múltiplo de la base hexadecimal lo sumamos al operando del lado derecho de la operación.

$$\begin{aligned} \text{Ejemplos: } CB3467_{(16)} + AF_{(16)} &= CB3516_{(16)} \\ 2FDCA8_{(16)} + 458FE_{(16)} &= 3435A6_{(16)} \end{aligned}$$

#### RESTA

Para realizar la resta hexadecimal se puede hacer de dos formas, por el método tradicional o por el método de complementos.

#### Método Tradicional

$$\begin{aligned} \text{Ejemplos: } 5FE2_{(16)} - 98D_{(16)} &= 5655_{(16)} \text{ Cuando se va a prestar se presta una cantidad, 16 unidades} \\ FFFF_{(16)} - EEA2_{(16)} &= 115D_{(16)} \end{aligned}$$

#### Método Complementos

Para realizar este método debemos seguir los siguientes pasos:

- A) Le resto a un número de 15 igual al sustraendo, y luego al resultado le sumamos 1 para obtener el complemento del sustraendo.
- B) Sumar el minuendo con el complemento del sustraendo.
- C) Eliminamos el primer uno que se encuentre más a la izquierda (en caso sobrante).

Ejemplos:

$$72A4_{(16)} - 4E86_{(16)} = 241E_{(16)}$$

$$\text{Paso A } 15151515 - 4E86 = 11179 + 1 = 111710 = B17A$$

$$\text{Paso B } 72A4 + B17A = 241E_{(16)}$$

### MULTIPLICACIÓN

Se realiza de forma idéntica al sistema decimal a excepción de la suma de los resultados de la multiplicación la cual se debe realizar en forma hexadecimal.

$$5641_{(16)} * 35_{(16)} = 11DB75_{(16)}$$

## DIVISIÓN

Se realiza a través de la conversión al sistema decimal del dividendo momentáneo y el divisor general, en la cual el valor obtenido en el cociente de la división decimal es el valor que se coloca como cociente de la división hexadecimal. Este Cociente lo multiplicamos por el divisor octal aplicando las propiedades de la multiplicación y dicho valor obtenido se lo restamos al dividendo aplicando también las propiedades de la resta octal.

$$ABCDF_{(16)} / 9_{(16)} = 1316E_{(8)}$$

1. A=10 9=9 10/9=1
2. 9\*1=9 A-9=1
3. 1B=27 27/9=3
4. 9\*3= 1B(27) 1B-1B=0
5. C= 12 12/9=1
6. 9\*1=9 C-9=3
7. 3D=60 60/9= 6
8. 9\*6=36 (54) 3D-36=7
9. 7F=127 127/9=E(14)
10. 9\*E=7F 7F-7F=0

Potencias de 2	Valor decimal
2 <sub>10</sub>	1024
2 <sub>9</sub>	512
2 <sub>8</sub>	256
2 <sub>7</sub>	128
2 <sub>6</sub>	64
2 <sub>5</sub>	32
2 <sub>4</sub>	16
2 <sub>3</sub>	8
2 <sub>2</sub>	4
2 <sub>1</sub>	2
2 <sub>0</sub>	1
2 <sub>-1</sub>	1/2=0.5
2 <sub>-2</sub>	1/4=0.25
2 <sub>-3</sub>	1/8=0.125
2 <sub>-4</sub>	1/16=0.0625
2 <sub>-5</sub>	1/32=0.03125
2 <sub>-6</sub>	1/64=0.015625

Potencias de 8	Valor decimal
8 <sub>3</sub>	1/512=0.001953125
8 <sub>2</sub>	1/64=0.015625
8 <sub>1</sub>	1/8=0.125
8 <sub>0</sub>	1
8 <sub>1</sub>	8
8 <sub>2</sub>	64
8 <sub>3</sub>	512
8 <sub>4</sub>	4096
8 <sub>5</sub>	32768

Potencias de 16	Valor decimal
16 <sub>3</sub>	1/4096=0.000244
16 <sub>2</sub>	1/256=0.003906
16 <sub>1</sub>	1/16=0.125
16 <sub>0</sub>	1
16 <sub>1</sub>	16
16 <sub>2</sub>	256
16 <sub>3</sub>	4096
16 <sub>4</sub>	65536
16 <sub>5</sub>	1048576

## CONVERSIONES ENTRE LOS DISTINTOS SISTEMAS NUMERICOS

### BINARIO-DECIMAL

Se realiza utilizando la expresión en notación expandida

$$\text{Ejemplo: } 10011_{(2)} = 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 1*16 + 0*8 + 0*4 + 1*2 + 1*1$$

$$16 + 0 + 0 + 2 + 1$$

$$19_{(10)}$$

### DECIMAL-BINARIO

Se realiza a través de divisiones sucesivas entre dos (2), tomando el residuo como parte de la conversión y el cociente como dividendo de la próxima división. El resultado es la agrupación de los residuos y en forma inversa es su resultado.

Ejemplo:  $98 = 1100010$

### **BINARIO-OCTAL**

Se realiza agrupando de izquierda a derecha en grupo de tres bits consecutivos el número binario para luego utilizando la tabla de tres bits del sistema binario darle a cada uno su representación octal.

Ejemplo:  $7654 = 111\ 110\ 101\ 100 = 11110101100$

### **OCTAL-BINARIO**

Se realiza utilizando la tabla de tres bits del sistema octal, con el valor correspondiente a cada uno de los elementos del sistema octal se escribe su representación binaria.

Ejemplo:  $7654 = 111\ 110\ 101\ 100 = 11110101100$

### **BINARIO-HEXADECIMAL**

Se realiza agrupando de izquierda a derecha en grupo de cuatro bits consecutivos el número binario para luego utilizando la tabla de cuatro bits del sistema binario darle a cada uno su representación hexadecimal.

Ejemplo:  $10101001011100 = 0010\ 1010\ 0101\ 1100 = 2A5C$

### **HEXADECIMAL-BINARIO**

Se realiza utilizando la tabla de cuatro bits del sistema hexadecimal, con el valor correspondiente a cada uno de los elementos del sistema hexadecimal se escribe su representación binaria.

Ejemplo:  $2A5C = 0010\ 1010\ 0101\ 1100 = 10101001011100$

### **OCTAL-DECIMAL**

Se realiza utilizando la expresión en notación expandida

Ejemplo:  $5725_{(8)} = 5 \cdot 8^3 + 7 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 = 3029$

### **DECIMAL-OCTAL**

Se realiza a través de divisiones sucesivas entre ocho (8), tomando el residuo como parte de la conversión y el cociente como dividendo de la próxima división. El resultado es la agrupación de los residuos y en forma inversa es su resultado.

Ejemplo:  $98 = 1100010$

### **OCTAL-HEXADECIMAL**

Se realiza representando el número octal en forma binaria (agrupación en tres bits), dicho valor en binario se pasa a forma hexadecimal (agrupando en cuatro bits).

Ejemplo:  $47231 = 100\ 111\ 010\ 011\ 001 = 0100\ 1110\ 1001\ 1001 = 4E99$

### **HEXADECIMAL-OCTAL**

Se realiza representando el número hexadecimal en forma binaria (agrupación en cuatro bits), dicho valor en binario se pasa a forma octal (agrupando en tres bits).

Ejemplo:  $4E99 = 0100\ 1110\ 1001\ 1001 = 100\ 111\ 010\ 011\ 001 = 47231$

### **HEXADECIMAL-DECIMAL**

Se realiza utilizando la expresión en notación expandida

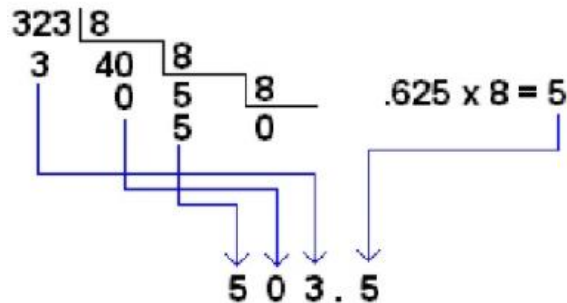
Ejemplo:  $2351_{(16)} = 2 \cdot 16^3 + 3 \cdot 16^2 + 5 \cdot 16^1 + 1 \cdot 16^0 = 2 \cdot 4096 + 3 \cdot 256 + 5 \cdot 16 + 1 \cdot 1 = 9041$

### **DECIMAL-HEXADECIMAL**

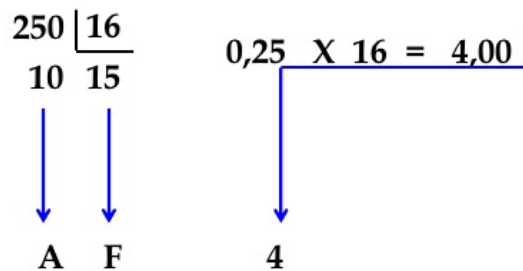
Se realiza por medio de divisiones sucesivas entre diez y seis, tomando el residuo como parte de la conversión y el cociente como dividendo de la próxima división. El resultado es la agrupación de los residuos y en forma inversa es su resultado. Ejemplo:  $9041 = 2351$

#### PARTE DECIMAL ENTRE SISTEMAS

Convertir **323.625** a Octal



Transformar = 250.25



Resultado= FA.4

#### Websites para conversiones

[http://www.periodni.com/es/sistema\\_de\\_numeracion\\_convertidor.html](http://www.periodni.com/es/sistema_de_numeracion_convertidor.html)

[http://wims.unice.fr/wims/es\\_tool~number~baseconv.es.html](http://wims.unice.fr/wims/es_tool~number~baseconv.es.html)

#### Websites complementarias

<https://es.khanacademy.org/math/pre-algebra/applying-math-reasoning-topic/alternate-number-bases/v/number-systems-introduction>

[https://es.wikipedia.org/wiki/Sistema\\_binario](https://es.wikipedia.org/wiki/Sistema_binario)

<http://www.areatecnologia.com/sistema-binario.htm>

<https://www.disfrutalasmatematicas.com/numeros/>

#### Videos complementarios

1. UTPL Conversión de sistemas numéricos  
<https://www.youtube.com/watch?v=l6uSJdm-uus>

2. Conversiones sistemas numéricos #1 Introducción, métodos y conceptos básicos.

<https://www.youtube.com/watch?v=jdp91beSm5g>

3. Conversiones entre sistemas numéricos #2 Cifras decimales en cadenas numéricas.

<https://www.youtube.com/watch?v=h7tEf4pnANc>

4. Conversiones entre sistemas numericos #3 Metodos alternativos Decimal a binario

<https://www.youtube.com/watch?v=k-1AK5s6M-k>

5. Conversiones entre sistemas numéricos #4 Compresión y descompresión Binaria. Hexadecimal y octal.

<https://www.youtube.com/watch?v=Q2ddN4l89Z4>

## EJERCICIOS

- 1) *Averigua y escribe el código ASCII correspondiente, tanto en decimal como en binario, a las letras de tu nombre y apellidos. Distinguir entre mayúsculas/minúsculas, y sin acentos.*

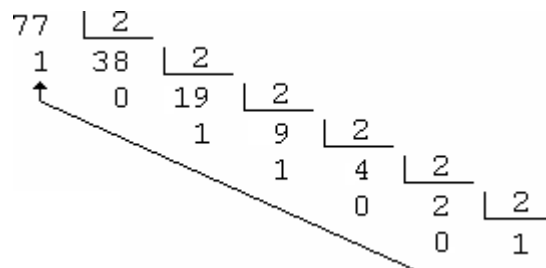
Nota: Al final de este documento podéis encontrar la tabla ASCII de los caracteres imprimibles.

LETRA	DECIMAL	BINARIO
M	77	1001101
a	97	1100001
n	110	1101110
u	117	1110101
e	101	1100101
l	108	1101100

Para cada letra del nombre extraemos su correspondencia en decimal utilizando la tabla ASCII. Una vez tenemos todos los valores decimales del nombre procedemos a convertir cada valor decimal a binario.

Como ejemplo, veamos la conversión de la primera letra del nombre (M) a su valor en binario.

- *Conversión a binario del número  $77_{10}$*

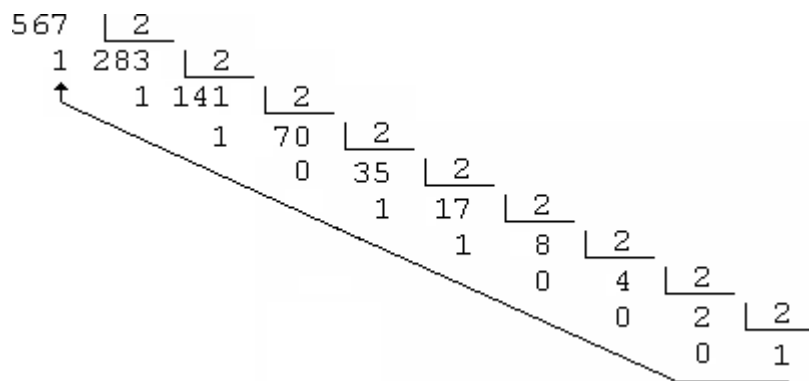


El resultado, siguiendo todos los restos empezando por el cociente de más a la derecha, nos da el resultado final en binario. Es decir, **1001101<sub>2</sub>**. La conversión para el resto de letras se realizaría siguiendo este mismo proceso.

- 2) *Realiza la conversión a binario del número decimal 567.*



Siguiendo el mismo proceso que en el ejercicio anterior, tenemos que dividir sucesivamente el  $567_{10}$  entre 2 hasta que lleguemos a un número que ya no sea divisible entre 2.



Siguiendo todos los restos obtenidos, empezando por el cociente de más a la derecha, nos da el resultado final en binario. Es decir, **1001110110<sub>2</sub>**.

**3) Realiza la conversión tanto a decimal como a hexadecimal del número binario 1001110110.**

▪ **Conversion a decimal**

En esta conversión cada cifra será multiplicada por la base del sistema de numeración ( $b=2$ ) elevada a una potencia que dependerá de la posición de esa cifra en el número a convertir. Empezando por la posición de más a la derecha la primera cifra se multiplica por  $2^0$ , la segunda por la derecha se multiplica por  $2^1$ , y así sucesivamente.

$$1001110110_2 = 1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$1001110110_2 = 1 \times 2^9 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 \text{ (obviamos los ceros)}$$

$$1001110110_2 = 512 + 64 + 32 + 16 + 4 + 2$$

$$1001110110_2 = \mathbf{630_{10}}$$

▪ **Conversión a hexadecimal**

Para la conversión a hexadecimal el número a convertir se divide en grupos de 4 bits (empezando por la derecha). Si el último grupo no tiene 4 bits se le añaden tantos ceros a la izquierda como sea necesario hasta completar el grupo.

Por tanto, el número  $1001110110_2$  lo dividimos en 3 grupos de 4 bits cada uno de ellos; a saber: 10, 0111 y 0110. Como el último grupo no llega a los 4 bits lo rellenamos con ceros a la izquierda, quedando los 3 grupos de 4 bits como: 0010, 0111 y 0110.

BINARIO	DECIMAL	HEXADECIMAL
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4

0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

**Tabla 1: Conversión directa entre binario, decimal y hexadecimal**

Después de esa división, la conversión es directa (ver Tabla 1) ya que a cada grupo de 4 bits ( $2^4 = 16$  posibles valores = los que tiene el alfabeto hexadecimal) le corresponde un valor en el alfabeto hexadecimal.

La conversión de cada grupo, si no sabemos la conversión directa entre el binario y el hexadecimal, se haría de la siguiente manera:

$$0010_2 = 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1 \times 2^1 = 2_{10} = 2_{16}$$

$$0111_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 4 + 2 + 1 = 7_{10} = 7_{16}$$

$$0110_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1 \times 2^2 + 1 \times 2^1 = 4 + 2 = 6_{10} = 6_{16}$$

Por tanto, como podemos ver en la siguiente figura, al  $0010_2$  le corresponde el valor hexadecimal 2, al  $0111_2$  el valor hexadecimal 7 y al  $0110_2$  el valor hexadecimal 6.

0010	0111	0110	← binario
⋮	⋮	⋮	
2	7	6	← hexadecimal

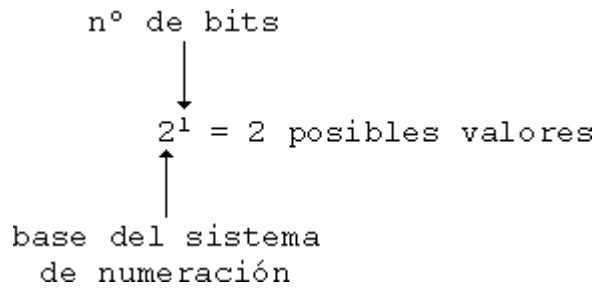
El resultado final es:  $1001110110_2 = 276_{16}$ .

- 4) El sistema octal es un sistema de numeración en base 8 ( $b=8$ ) cuyo alfabeto es { 0, 1, 2, 3, 4, 5, 6, 7 }. Construir una tabla con la representación de los 32 primeros números en los sistemas de numeración hexadecimal, decimal, octal y binario.

BINARIO	OCTAL	DECIMAL	HEXADECIMAL	BINARIO	OCTAL	DECIMAL	HEXADECIMAL
0000	00	00	00	10000	20	16	10
0001	01	01	01	10001	21	17	11
0010	02	02	02	10010	22	18	12
0011	03	03	03	10011	23	19	13
0100	04	04	04	10100	24	20	14
0101	05	05	05	10101	25	21	15
0110	06	06	06	10110	26	22	16
0111	07	07	07	10111	27	23	17
1000	10	08	08	11000	30	24	18
1001	11	09	09	11001	31	25	19
1010	12	10	0A	11010	32	26	1A
1011	13	11	0B	11011	33	27	1B
1100	14	12	0C	11100	34	28	1C
1101	15	13	0D	11101	35	29	1D
1110	16	14	0E	11110	36	30	1E
1111	17	15	0F	11111	37	31	1F

**Tabla 2: Los 32 primeros números en varios sistemas de numeración**

Como se puede comprobar en la Tabla 2, hay cierta relación en el número de bits que necesitan los diferentes sistemas de numeración vistos en clase. Con un bit, el sistema binario puede codificar hasta 2 valores; el 0 y el 1.



Con 2 bits se pueden llegar a codificar hasta  $2^2 = 4$  posibles valores diferentes:

<b>BINARIO</b> 2 DÍGITOS	<b>DECIMAL</b>
00	0
01	1
10	2
11	3

**Tabla 3: Los 4 posibles valores que se pueden codificar con 2 bits**

Por tanto, un sistema de numeración cuyo alfabeto fuese { 0, 1, 2, 3 } (4 posibles valores diferentes) podría codificarse en un ordenador utilizando únicamente 2 bits. Con 3 bits se pueden codificar hasta  $2^3 = 8$  posibles valores diferentes:

<i>BINARIO</i> 3 DÍGITOS	<b>OCTAL</b> 1 DÍGITO	<b>DECIMAL</b>
000	0	0
001	1	1
010	2	2
011	3	3
100	4	4
101	5	5
110	6	6
111	7	7

**Tabla 4: Los 8 posibles valores que se pueden codificar con 3 bits**

Por tanto, con 3 bits podemos codificar todos los posibles valores del sistema de numeración octal (b=8); es decir, todos los posibles valores de su alfabeto { 0, 1, 2, 3, 4, 5, 6, 7 } (8 posibles valores diferentes).

Por último, con 4 bits se pueden codificar hasta  $2^4 = 16$  posibles valores diferentes; es decir, los mismos posibles valores que tiene el sistema hexadecimal (b=16). Ver la Tabla 5.

*BINARIO*  
4 DÍGITOS

*HEXADECIMAL*  
1 DÍGITO

**DECIMAL**

0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

**Tabla 5: Los 16 posibles valores que se pueden codificar con 4 bits**

Entonces, como podemos ver en las dos tablas anteriores:

- 3 dígitos binarios equivalen a 1 dígito octal  $\rightarrow 2^3 = 8^1 = 8$  posibles valores.
- 4 dígitos binarios equivalen a 1 dígito hexadecimal  $\rightarrow 2^4 = 16^1 = 16$  posibles valores.

**5)** *Intenta realizar la conversión a decimal del número octal 325.*

El proceso de conversión a decimal es igual que si hiciéramos la conversión de binario a decimal, pero en este caso la base del sistema de numeración es 8 (b=8) en lugar de 2.

$$325_8 = 3 \times 8^2 + 2 \times 8^1 + 5 \times 8^0 = 3 \times 64 + 2 \times 8 + 5 \times 1 = 192 + 16 + 5 = 213_{10}$$

**6)** *¿Serías capaz de escribir el proceso de conversión entre números en octal y binario, y viceversa? (nota: utiliza como punto de partida el proceso que se realiza entre hexadecimal y binario).*

En el sistema hexadecimal dividimos el número binario a convertir en grupos de 4 bits ya que, como hemos visto en el ejercicio 4 (ver Tabla 5), con 4 bits podemos representar 16 valores diferentes; es decir, todo el alfabeto de valores hexadecimales.

Por tanto, para la conversión de números binarios al sistema octal (8 posibles valores diferentes) podemos hacerlo dividiendo ese número en grupos de 3 bits, como hemos visto en

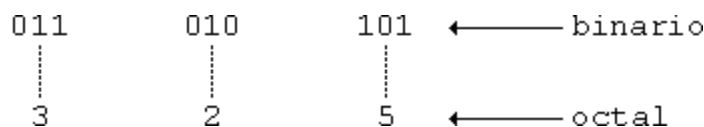


el ejercicio 4 (ver Tabla 4), y añadiendo al último grupo todos los ceros a la izquierda que sean necesarios.

Pongamos como ejemplo el número binario 11010101. Para su conversión al sistema de numeración octal, vamos dividiendo ese número en grupos de 3 empezando por la derecha; es decir, obtenemos 3 grupos diferentes: 11, 010 y 101. Al último grupo le añadimos un 0 a la izquierda para rellenar ese grupo hasta que tenga 3 dígitos.

Por tanto, tenemos los 3 grupos de 3 dígitos: 011, 010 y 101.

Después de esa división la conversión es directa (ver Tabla 4), obteniendo finalmente, como podemos ver en la siguiente figura, el valor 325 en el sistema octal (mismo valor que en el ejercicio 5).



La conversión de cada grupo, si no sabemos la conversión directa entre el binario y el octal, se haría de la siguiente manera:

$$011_2 = 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1 \times 2^1 + 1 \times 2^0 = 2 + 1 = 3_{10}$$

$$010_2 = 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1 \times 2^1 = 2_{10}$$

$$101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 2^2 + 1 \times 2^0 = 4 + 1 = 5_{10}$$

**7)** ¿Cuál es el siguiente número hexadecimal al 19F? a) 2A0 b) 200 c) 1A0

Si empezamos por la cifra de más a la derecha, la F, tenemos que el siguiente número es el **0** ya que la F es la última cifra del alfabeto hexadecimal. Extrapolándolo a nuestro sistema decimal, lo mismo ocurre cuando del número 9 pasamos directamente al 0, debido a que ya no existen más números en el alfabeto decimal.

Al cambiar de la F al 0 tenemos que sumar 1 (el acarreo) a la siguiente cifra; al igual que ocurre en el sistema decimal. En este caso, al 9 le sumamos 1. El resultado en hexadecimal es **A**, que correspondería con el número 10 en decimal. Como pasar del 9 a la A no supone volver al primer valor del alfabeto, la tercera cifra no sufre ninguna modificación (no hay acarreo). Por tanto se quedaría el **1**.

Solución: El siguiente número al 19F<sub>16</sub> corresponde con el 1A0<sub>16</sub> (**opción C**).

**8)** ¿Cuántos bytes tienen 16 MB? Y, ¿cuántos bits?

Si 1 MB corresponde a 1024 KB entonces 16 MB serán 16×1024 KB. Es decir, 16 MB corresponden a 16384 KB en total. A su vez, 1 KB corresponde a 1024 bytes; por lo que 16384 KB serán 16384×1024 bytes. Es decir, **16777216 bytes** en total.

Por otro lado, como 1 byte corresponde a 8 bits entonces 16777216 bytes serán 16777216×8 bits. Es decir, 16 MB corresponde a **134217728 bits** en total.

Resumiendo:  $16 \text{ MB} = 16 \times 1024 \text{ KB} = 16384 \text{ KB} = 16384 \times 1024 \text{ bytes} = 16777216 \text{ bytes} = 16777216 \times 8 \text{ bits} = 134217728 \text{ bits}$

**9)** Si tengo 2 módulos de memoria con 512 MB y 1024 MB, ¿cuántos GB de memoria tengo en total?

La suma total de ambos módulos sería  $512 \text{ MB} + 1024 \text{ MB} = 1536 \text{ MB}$ . Por otro

lado, si 1 GB corresponde a 1024 MB entonces 1536 MB serán:

$$1536 \text{ MB} / 1024 \text{ MB} = \mathbf{1,5 \text{ GB}}$$

**10)** Tenemos un disco duro con una capacidad total de 20 GB. Si cada bloque en los que ese disco duro se divide tiene 4 KB, ¿cuántos bloques hay en total?

Lo primero, tendremos que convertir la capacidad del disco duro de GB a KB. Recordemos que 1 GB corresponde a 1024 MB (primera conversión), por lo que 20 GB serán:

$$20 \times 1024 = 20480 \text{ MB}$$

A su vez, 1 MB equivale a 1024 KB (segunda conversión). Por tanto, si tenemos un disco duro de 20480 MB, en total tendremos también:

$$20480 \times 1024 = 20971520 \text{ KB}$$

Finalmente, si cada bloque en los que el disco duro se divide ocupa 4 KB, tendremos un total de: 20971520

$$\text{KB} / 4 \text{ KB} = \mathbf{5242880 \text{ bloques}}$$

TABLA ASCII

El código 32 es el espacio en blanco. Los códigos del 33 al 126 se conocen como **caracteres imprimibles**, y representan letras, dígitos, signos de puntuación, etc.

DECIMAL	CARÁCTER	DECIMAL	CARÁCTER	DECIMAL	CARÁCTER
32	<i>espacio</i>	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_	127	<i>delete</i>