

Ciclo Formativo de Grado Superior de Administración de Sistemas Informáticos en red



Módulo Profesional: **LMSGI**

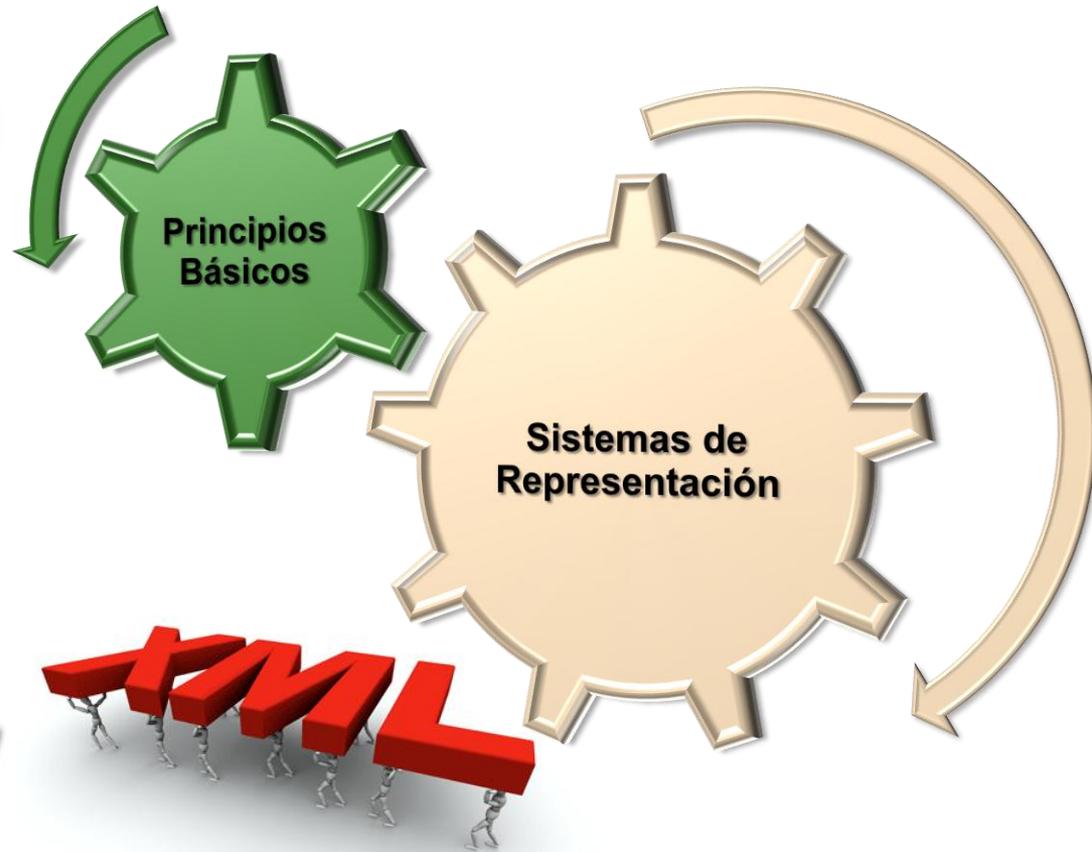
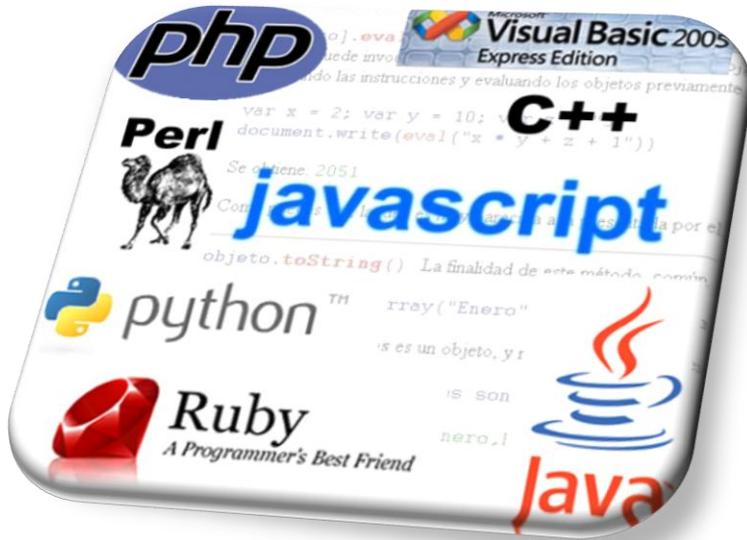
UT 2. Introducción a los Lenguajes de Programación

*Departamento de Informática y Comunicación
IES San Juan Bosco (Lorca-Murcia)
Profesor: Juan Antonio López Quesada*



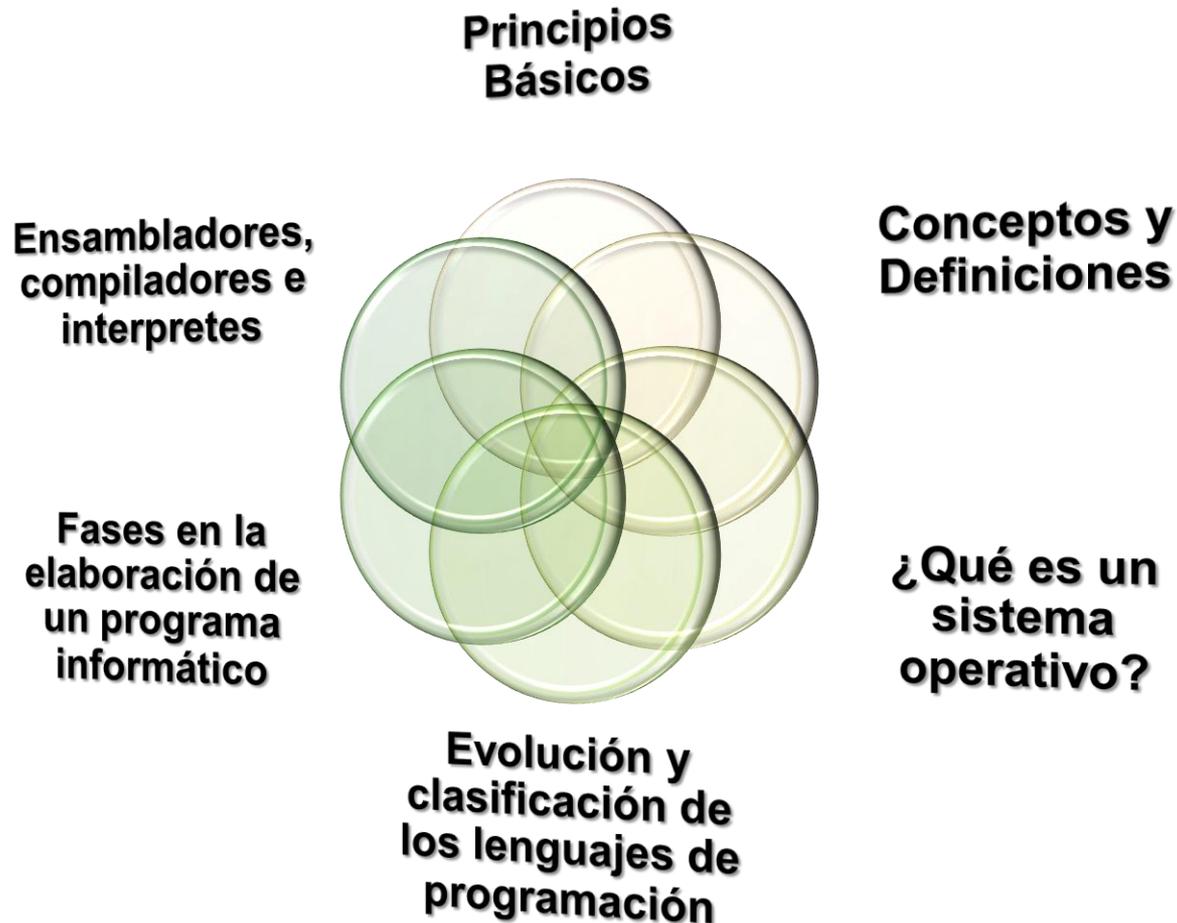


Índice de Contenidos



A a B b C c D d E
N O Y Δ ≡ Ж ѣ џ
س ع ك あ に 廿
文 毳 𠂇 𠂈 𠂉 𠂊 𠂋 𠂌 𠂍 𠂎 𠂏 𠂐 𠂑 𠂒 𠂓 𠂔 𠂕 𠂖 𠂗 𠂘 𠂙 𠂚 𠂛 𠂜 𠂝 𠂞 𠂟 𠂠 𠂡 𠂢 𠂣 𠂤 𠂥 𠂦 𠂧 𠂨 𠂩 𠂪 𠂫 𠂬 𠂭 𠂮 𠂯 𠂰 𠂱 𠂲 𠂳 𠂴 𠂵 𠂶 𠂷 𠂸 𠂹 𠂺 𠂻 𠂼 𠂽 𠂾 𠂿 𠃀 𠃁 𠃂 𠃃 𠃄 𠃅 𠃆 𠃇 𠃈 𠃉 𠃊 𠃋 𠃌 𠃍 𠃎 𠃏 𠃐 𠃑 𠃒 𠃓 𠃔 𠃕 𠃖 𠃗 𠃘 𠃙 𠃚 𠃛 𠃜 𠃝 𠃞 𠃟 𠃠 𠃡 𠃢 𠃣 𠃤 𠃥 𠃦 𠃧 𠃨 𠃩 𠃪 𠃫 𠃬 𠃭 𠃮 𠃯 𠃰 𠃱 𠃲 𠃳 𠃴 𠃵 𠃶 𠃷 𠃸 𠃹 𠃺 𠃻 𠃼 𠃽 𠃾 𠃿 𠄀 𠄁 𠄂 𠄃 𠄄 𠄅 𠄆 𠄇 𠄈 𠄉 𠄊 𠄋 𠄌 𠄍 𠄎 𠄏 𠄐 𠄑 𠄒 𠄓 𠄔 𠄕 𠄖 𠄗 𠄘 𠄙 𠄚 𠄛 𠄜 𠄝 𠄞 𠄟 𠄠 𠄡 𠄢 𠄣 𠄤 𠄥 𠄦 𠄧 𠄨 𠄩 𠄪 𠄫 𠄬 𠄭 𠄮 𠄯 𠄰 𠄱 𠄲 𠄳 𠄴 𠄵 𠄶 𠄷 𠄸 𠄹 𠄺 𠄻 𠄼 𠄽 𠄾 𠄿 𠅀 𠅁 𠅂 𠅃 𠅄 𠅅 𠅆 𠅇 𠅈 𠅉 𠅊 𠅋 𠅌 𠅍 𠅎 𠅏 𠅐 𠅑 𠅒 𠅓 𠅔 𠅕 𠅖 𠅗 𠅘 𠅙 𠅚 𠅛 𠅜 𠅝 𠅞 𠅟 𠅠 𠅡 𠅢 𠅣 𠅤 𠅥 𠅦 𠅧 𠅨 𠅩 𠅪 𠅫 𠅬 𠅭 𠅮 𠅯 𠅰 𠅱 𠅲 𠅳 𠅴 𠅵 𠅶 𠅷 𠅸 𠅹 𠅺 𠅻 𠅼 𠅽 𠅾 𠅿 𠆀 𠆁 𠆂 𠆃 𠆄 𠆅 𠆆 𠆇 𠆈 𠆉 𠆊 𠆋 𠆌 𠆍 𠆎 𠆏 𠆐 𠆑 𠆒 𠆓 𠆔 𠆕 𠆖 𠆗 𠆘 𠆙 𠆚 𠆛 𠆜 𠆝 𠆞 𠆟 𠆠 𠆡 𠆢 𠆣 𠆤 𠆥 𠆦 𠆧 𠆨 𠆩 𠆪 𠆫 𠆬 𠆭 𠆮 𠆯 𠆰 𠆱 𠆲 𠆳 𠆴 𠆵 𠆶 𠆷 𠆸 𠆹 𠆺 𠆻 𠆼 𠆽 𠆾 𠆿 𠇀 𠇁 𠇂 𠇃 𠇄 𠇅 𠇆 𠇇 𠇈 𠇉 𠇊 𠇋 𠇌 𠇍 𠇎 𠇏 𠇐 𠇑 𠇒 𠇓 𠇔 𠇕 𠇖 𠇗 𠇘 𠇙 𠇚 𠇛 𠇜 𠇝 𠇞 𠇟 𠇠 𠇡 𠇢 𠇣 𠇤 𠇥 𠇦 𠇧 𠇨 𠇩 𠇪 𠇫 𠇬 𠇭 𠇮 𠇯 𠇰 𠇱 𠇲 𠇳 𠇴 𠇵 𠇶 𠇷 𠇸 𠇹 𠇺 𠇻 𠇼 𠇽 𠇾 𠇿 𠈀 𠈁 𠈂 𠈃 𠈄 𠈅 𠈆 𠈇 𠈈 𠈉 𠈊 𠈋 𠈌 𠈍 𠈎 𠈏 𠈐 𠈑 𠈒 𠈓 𠈔 𠈕 𠈖 𠈗 𠈘 𠈙 𠈚 𠈛 𠈜 𠈝 𠈞 𠈟 𠈠 𠈡 𠈢 𠈣 𠈤 𠈥 𠈦 𠈧 𠈨 𠈩 𠈪 𠈫 𠈬 𠈭 𠈮 𠈯 𠈰 𠈱 𠈲 𠈳 𠈴 𠈵 𠈶 𠈷 𠈸 𠈹 𠈺 𠈻 𠈼 𠈽 𠈾 𠈿 𠉀 𠉁 𠉂 𠉃 𠉄 𠉅 𠉆 𠉇 𠉈 𠉉 𠉊 𠉋 𠉌 𠉍 𠉎 𠉏 𠉐 𠉑 𠉒 𠉓 𠉔 𠉕 𠉖 𠉗 𠉘 𠉙 𠉚 𠉛 𠉜 𠉝 𠉞 𠉟 𠉠 𠉡 𠉢 𠉣 𠉤 𠉥 𠉦 𠉧 𠉨 𠉩 𠉪 𠉫 𠉬 𠉭 𠉮 𠉯 𠉰 𠉱 𠉲 𠉳 𠉴 𠉵 𠉶 𠉷 𠉸 𠉹 𠉺 𠉻 𠉼 𠉽 𠉾 𠉿 𠊀 𠊁 𠊂 𠊃 𠊄 𠊅 𠊆 𠊇 𠊈 𠊉 𠊊 𠊋 𠊌 𠊍 𠊎 𠊏 𠊐 𠊑 𠊒 𠊓 𠊔 𠊕 𠊖 𠊗 𠊘 𠊙 𠊚 𠊛 𠊜 𠊝 𠊞 𠊟 𠊠 𠊡 𠊢 𠊣 𠊤 𠊥 𠊦 𠊧 𠊨 𠊩 𠊪 𠊫 𠊬 𠊭 𠊮 𠊯 𠊰 𠊱 𠊲 𠊳 𠊴 𠊵 𠊶 𠊷 𠊸 𠊹 𠊺 𠊻 𠊼 𠊽 𠊾 𠊿 𠋀 𠋁 𠋂 𠋃 𠋄 𠋅 𠋆 𠋇 𠋈 𠋉 𠋊 𠋋 𠋌 𠋍 𠋎 𠋏 𠋐 𠋑 𠋒 𠋓 𠋔 𠋕 𠋖 𠋗 𠋘 𠋙 𠋚 𠋛 𠋜 𠋝 𠋞 𠋟 𠋠 𠋡 𠋢 𠋣 𠋤 𠋥 𠋦 𠋧 𠋨 𠋩 𠋪 𠋫 𠋬 𠋭 𠋮 𠋯 𠋰 𠋱 𠋲 𠋳 𠋴 𠋵 𠋶 𠋷 𠋸 𠋹 𠋺 𠋻 𠋼 𠋽 𠋾 𠋿 𠌀 𠌁 𠌂 𠌃 𠌄 𠌅 𠌆 𠌇 𠌈 𠌉 𠌊 𠌋 𠌌 𠌍 𠌎 𠌏 𠌐 𠌑 𠌒 𠌓 𠌔 𠌕 𠌖 𠌗 𠌘 𠌙 𠌚 𠌛 𠌜 𠌝 𠌞 𠌟 𠌠 𠌡 𠌢 𠌣 𠌤 𠌥 𠌦 𠌧 𠌨 𠌩 𠌪 𠌫 𠌬 𠌭 𠌮 𠌯 𠌰 𠌱 𠌲 𠌳 𠌴 𠌵 𠌶 𠌷 𠌸 𠌹 𠌺 𠌻 𠌼 𠌽 𠌾 𠌿 𠍀 𠍁 𠍂 𠍃 𠍄 𠍅 𠍆 𠍇 𠍈 𠍉 𠍊 𠍋 𠍌 𠍍 𠍎 𠍏 𠍐 𠍑 𠍒 𠍓 𠍔 𠍕 𠍖 𠍗 𠍘 𠍙 𠍚 𠍛 𠍜 𠍝 𠍞 𠍟 𠍠 𠍡 𠍢 𠍣 𠍤 𠍥 𠍦 𠍧 𠍨 𠍩 𠍪 𠍫 𠍬 𠍭 𠍮 𠍯 𠍰 𠍱 𠍲 𠍳 𠍴 𠍵 𠍶 𠍷 𠍸 𠍹 𠍺 𠍻 𠍼 𠍽 𠍾 𠍿 𠎀 𠎁 𠎂 𠎃 𠎄 𠎅 𠎆 𠎇 𠎈 𠎉 𠎊 𠎋 𠎌 𠎍 𠎎 𠎏 𠎐 𠎑 𠎒 𠎓 𠎔 𠎕 𠎖 𠎗 𠎘 𠎙 𠎚 𠎛 𠎜 𠎝 𠎞 𠎟 𠎠 𠎡 𠎢 𠎣 𠎤 𠎥 𠎦 𠎧 𠎨 𠎩 𠎪 𠎫 𠎬 𠎭 𠎮 𠎯 𠎰 𠎱 𠎲 𠎳 𠎴 𠎵 𠎶 𠎷 𠎸 𠎹 𠎺 𠎻 𠎼 𠎽 𠎾 𠎿 𠏀 𠏁 𠏂 𠏃 𠏄 𠏅 𠏆 𠏇 𠏈 𠏉 𠏊 𠏋 𠏌 𠏍 𠏎 𠏏 𠏐 𠏑 𠏒 𠏓 𠏔 𠏕 𠏖 𠏗 𠏘 𠏙 𠏚 𠏛 𠏜 𠏝 𠏞 𠏟 𠏠 𠏡 𠏢 𠏣 𠏤 𠏥 𠏦 𠏧 𠏨 𠏩 𠏪 𠏫 𠏬 𠏭 𠏮 𠏯 𠏰 𠏱 𠏲 𠏳 𠏴 𠏵 𠏶 𠏷 𠏸 𠏹 𠏺 𠏻 𠏼 𠏽 𠏾 𠏿 𠐀 𠐁 𠐂 𠐃 𠐄 𠐅 𠐆 𠐇 𠐈 𠐉 𠐊 𠐋 𠐌 𠐍 𠐎 𠐏 𠐐 𠐑 𠐒 𠐓 𠐔 𠐕 𠐖 𠐗 𠐘 𠐙 𠐚 𠐛 𠐜 𠐝 𠐞 𠐟 𠐠 𠐡 𠐢 𠐣 𠐤 𠐥 𠐦 𠐧 𠐨 𠐩 𠐪 𠐫 𠐬 𠐭 𠐮 𠐯 𠐰 𠐱 𠐲 𠐳 𠐴 𠐵 𠐶 𠐷 𠐸 𠐹 𠐺 𠐻 𠐼 𠐽 𠐾 𠐿 𠑀 𠑁 𠑂 𠑃 𠑄 𠑅 𠑆 𠑇 𠑈 𠑉 𠑊 𠑋 𠑌 𠑍 𠑎 𠑏 𠑐 𠑑 𠑒 𠑓 𠑔 𠑕 𠑖 𠑗 𠑘 𠑙 𠑚 𠑛 𠑜 𠑝 𠑞 𠑟 𠑠 𠑡 𠑢 𠑣 𠑤 𠑥 𠑦 𠑧 𠑨 𠑩 𠑪 𠑫 𠑬 𠑭 𠑮 𠑯 𠑰 𠑱 𠑲 𠑳 𠑴 𠑵 𠑶 𠑷 𠑸 𠑹 𠑺 𠑻 𠑼 𠑽 𠑾 𠑿 𠒀 𠒁 𠒂 𠒃 𠒄 𠒅 𠒆 𠒇 𠒈 𠒉 𠒊 𠒋 𠒌 𠒍 𠒎 𠒏 𠒐 𠒑 𠒒 𠒓 𠒔 𠒕 𠒖 𠒗 𠒘 𠒙 𠒚 𠒛 𠒜 𠒝 𠒞 𠒟 𠒠 𠒡 𠒢 𠒣 𠒤 𠒥 𠒦 𠒧 𠒨 𠒩 𠒪 𠒫 𠒬 𠒭 𠒮 𠒯 𠒰 𠒱 𠒲 𠒳 𠒴 𠒵 𠒶 𠒷 𠒸 𠒹 𠒺 𠒻 𠒼 𠒽 𠒾 𠒿 𠓀 𠓁 𠓂 𠓃 𠓄 𠓅 𠓆 𠓇 𠓈 𠓉 𠓊 𠓋 𠓌 𠓍 𠓎 𠓏 𠓐 𠓑 𠓒 𠓓 𠓔 𠓕 𠓖 𠓗 𠓘 𠓙 𠓚 𠓛 𠓜 𠓝 𠓞 𠓟 𠓠 𠓡 𠓢 𠓣 𠓤 𠓥 𠓦 𠓧 𠓨 𠓩 𠓪 𠓫 𠓬 𠓭 𠓮 𠓯 𠓰 𠓱 𠓲 𠓳 𠓴 𠓵 𠓶 𠓷 𠓸 𠓹 𠓺 𠓻 𠓼 𠓽 𠓾 𠓿 𠔀 𠔁 𠔂 𠔃 𠔄 𠔅 𠔆 𠔇 𠔈 𠔉 𠔊 𠔋 𠔌 𠔍 𠔎 𠔏 𠔐 𠔑 𠔒 𠔓 𠔔 𠔕 𠔖 𠔗 𠔘 𠔙 𠔚 𠔛 𠔜 𠔝 𠔞 𠔟 𠔠 𠔡 𠔢 𠔣 𠔤 𠔥 𠔦 𠔧 𠔨 𠔩 𠔪 𠔫 𠔬 𠔭 𠔮 𠔯 𠔰 𠔱 𠔲 𠔳 𠔴 𠔵 𠔶 𠔷 𠔸 𠔹 𠔺 𠔻 𠔼 𠔽 𠔾 𠔿 𠕀 𠕁 𠕂 𠕃 𠕄 𠕅 𠕆 𠕇 𠕈 𠕉 𠕊 𠕋 𠕌 𠕍 𠕎 𠕏 𠕐 𠕑 𠕒 𠕓 𠕔 𠕕 𠕖 𠕗 𠕘 𠕙 𠕚 𠕛 𠕜 𠕝 𠕞 𠕟 𠕠 𠕡 𠕢 𠕣 𠕤 𠕥 𠕦 𠕧 𠕨 𠕩 𠕪 𠕫 𠕬 𠕭 𠕮 𠕯 𠕰 𠕱 𠕲 𠕳 𠕴 𠕵 𠕶 𠕷 𠕸 𠕹 𠕺 𠕻 𠕼 𠕽 𠕾 𠕿 𠖀 𠖁 𠖂 𠖃 𠖄 𠖅 𠖆 𠖇 𠖈 𠖉 𠖊 𠖋 𠖌 𠖍 𠖎 𠖏 𠖐 𠖑 𠖒 𠖓 𠖔 𠖕 𠖖 𠖗 𠖘 𠖙 𠖚 𠖛 𠖜 𠖝 𠖞 𠖟 𠖠 𠖡 𠖢 𠖣 𠖤 𠖥 𠖦 𠖧 𠖨 𠖩 𠖪 𠖫 𠖬 𠖭 𠖮 𠖯 𠖰 𠖱 𠖲 𠖳 𠖴 𠖵 𠖶 𠖷 𠖸 𠖹 𠖺 𠖻 𠖼 𠖽 𠖾 𠖿 𠗀 𠗁 𠗂 𠗃 𠗄 𠗅 𠗆 𠗇 𠗈 𠗉 𠗊 𠗋 𠗌 𠗍 𠗎 𠗏 𠗐 𠗑 𠗒 𠗓 𠗔 𠗕 𠗖 𠗗 𠗘 𠗙 𠗚 𠗛 𠗜 𠗝 𠗞 𠗟 𠗠 𠗡 𠗢 𠗣 𠗤 𠗥 𠗦 𠗧 𠗨 𠗩 𠗪 𠗫 𠗬 𠗭 𠗮 𠗯 𠗰 𠗱 𠗲 𠗳 𠗴 𠗵 𠗶 𠗷 𠗸 𠗹 𠗺 𠗻 𠗼 𠗽 𠗾 𠗿 𠘀 𠘁 𠘂 𠘃 𠘄 𠘅 𠘆 𠘇 𠘈 𠘉 𠘊 𠘋 𠘌 𠘍 𠘎 𠘏 𠘐 𠘑 𠘒 𠘓 𠘔 𠘕 𠘖 𠘗 𠘘 𠘙 𠘚 𠘛 𠘜 𠘝 𠘞 𠘟 𠘠 𠘡 𠘢 𠘣 𠘤 𠘥 𠘦 𠘧 𠘨 𠘩 𠘪 𠘫 𠘬 𠘭 𠘮 𠘯 𠘰 𠘱 𠘲 𠘳 𠘴 𠘵 𠘶 𠘷 𠘸 𠘹 𠘺 𠘻 𠘼 𠘽 𠘾 𠘿 𠙀 𠙁 𠙂 𠙃 𠙄 𠙅 𠙆 𠙇 𠙈 𠙉 𠙊 𠙋 𠙌 𠙍 𠙎 𠙏 𠙐 𠙑 𠙒 𠙓 𠙔 𠙕 𠙖 𠙗 𠙘 𠙙 𠙚 𠙛 𠙜 𠙝 𠙞 𠙟 𠙠 𠙡 𠙢 𠙣 𠙤 𠙥 𠙦 𠙧 𠙨 𠙩 𠙪 𠙫 𠙬 𠙭 𠙮 𠙯 𠙰 𠙱 𠙲 𠙳 𠙴 𠙵 𠙶 𠙷 𠙸 𠙹 𠙺 𠙻 𠙼 𠙽 𠙾 𠙿 𠚀 𠚁 𠚂 𠚃 𠚄 𠚅 𠚆 𠚇 𠚈 𠚉 𠚊 𠚋 𠚌 𠚍 𠚎 𠚏 𠚐 𠚑 𠚒 𠚓 𠚔 𠚕 𠚖 𠚗 𠚘 𠚙 𠚚 𠚛 𠚜 𠚝 𠚞 𠚟 𠚠 𠚡 𠚢 𠚣 𠚤 𠚥 𠚦 𠚧 𠚨 𠚩 𠚪 𠚫 𠚬 𠚭 𠚮 𠚯 𠚰 𠚱 𠚲 𠚳 𠚴 𠚵 𠚶 𠚷 𠚸 𠚹 𠚺 𠚻 𠚼 𠚽 𠚾 𠚿 𠛀 𠛁 𠛂 𠛃 𠛄 𠛅 𠛆 𠛇 𠛈 𠛉 𠛊 𠛋 𠛌 𠛍 𠛎 𠛏 𠛐 𠛑 𠛒 𠛓 𠛔 𠛕 𠛖 𠛗 𠛘 𠛙 𠛚 𠛛 𠛜 𠛝 𠛞 𠛟 𠛠 𠛡 𠛢 𠛣 𠛤 𠛥 𠛦 𠛧 𠛨 𠛩 𠛪 𠛫 𠛬 𠛭 𠛮 𠛯 𠛰 𠛱 𠛲 𠛳 𠛴 𠛵 𠛶 𠛷 𠛸 𠛹 𠛺 𠛻 𠛼 𠛽 𠛾 𠛿 𠜀 𠜁 𠜂 𠜃 𠜄 𠜅 𠜆 𠜇 𠜈 𠜉 𠜊 𠜋 𠜌 𠜍 𠜎 𠜏 𠜐 𠜑 𠜒 𠜓 𠜔 𠜕 𠜖 𠜗 𠜘 𠜙 𠜚 𠜛 𠜜 𠜝 𠜞 𠜟 𠜠 𠜡 𠜢 𠜣 𠜤 𠜥 𠜦 𠜧 𠜨 𠜩 𠜪 𠜫 𠜬 𠜭 𠜮 𠜯 𠜰 𠜱 𠜲 𠜳 𠜴 𠜵 𠜶 𠜷 𠜸 𠜹 𠜺 𠜻 𠜼 𠜽 𠜾 𠜿 𠝀 𠝁 𠝂 𠝃 𠝄 𠝅 𠝆 𠝇 𠝈 𠝉 𠝊 𠝋 𠝌 𠝍 𠝎 𠝏 𠝐 𠝑 𠝒 𠝓 𠝔 𠝕 𠝖 𠝗 𠝘 𠝙 𠝚 𠝛 𠝜 𠝝 𠝞 𠝟 𠝠 𠝡 𠝢 𠝣 𠝤 𠝥 𠝦 𠝧 𠝨 𠝩 𠝪 𠝫 𠝬 𠝭 𠝮 𠝯 𠝰 𠝱 𠝲 𠝳 𠝴 𠝵 𠝶 𠝷 𠝸 𠝹 𠝺 𠝻 𠝼 𠝽 𠝾 𠝿 𠞀 𠞁 𠞂 𠞃 𠞄 𠞅 𠞆 𠞇 𠞈 𠞉 𠞊 𠞋 𠞌 𠞍 𠞎 𠞏 𠞐 𠞑 𠞒 𠞓 𠞔 𠞕 𠞖 𠞗 𠞘 𠞙 𠞚 𠞛 𠞜 𠞝 𠞞 𠞟 𠞠 𠞡 𠞢 𠞣 𠞤 𠞥 𠞦 𠞧 𠞨 𠞩 𠞪 𠞫 𠞬 𠞭 𠞮 𠞯 𠞰 𠞱 𠞲 𠞳 𠞴 𠞵 𠞶 𠞷 𠞸 𠞹 𠞺 𠞻 𠞼 𠞽 𠞾 𠞿 𠟀 𠟁 𠟂 𠟃 𠟄 𠟅 𠟆 𠟇 𠟈 𠟉 𠟊 𠟋 𠟌 𠟍 𠟎 𠟏 𠟐 𠟑 𠟒 𠟓 𠟔 𠟕 𠟖 𠟗 𠟘 𠟙 𠟚 𠟛 𠟜 𠟝 𠟞 𠟟 𠟠 𠟡 𠟢 𠟣 𠟤 𠟥 𠟦 𠟧 𠟨 𠟩 𠟪 𠟫 𠟬 𠟭 𠟮 𠟯 𠟰 𠟱 𠟲 𠟳 𠟴 𠟵 𠟶 𠟷 𠟸 𠟹 𠟺 𠟻 𠟼 𠟽 𠟾 𠟿 𠠀 𠠁 𠠂 𠠃 𠠄 𠠅 𠠆 𠠇 𠠈 𠠉 𠠊 𠠋 𠠌 𠠍 𠠎 𠠏 𠠐 𠠑 𠠒 𠠓 𠠔 𠠕 𠠖 𠠗 𠠘 𠠙 𠠚 𠠛 𠠜 𠠝 𠠞 𠠟 𠠠 𠠡 𠠢 𠠣 𠠤 𠠥 𠠦 𠠧 𠠨 𠠩 𠠪 𠠫 𠠬 𠠭 𠠮 𠠯 𠠰 𠠱 𠠲 𠠳 𠠴 𠠵 𠠶 𠠷 𠠸 𠠹 𠠺 𠠻 𠠼 𠠽 𠠾 𠠿 𠡀 𠡁 𠡂 𠡃 𠡄 𠡅 𠡆 𠡇 𠡈 𠡉 𠡊 𠡋 𠡌 𠡍 𠡎 𠡏 𠡐 𠡑 𠡒 𠡓 𠡔 𠡕 𠡖 𠡗 𠡘 𠡙 𠡚 𠡛 𠡜 𠡝 𠡞 𠡟 𠡠 𠡡 𠡢 𠡣 𠡤 𠡥 𠡦 𠡧 𠡨 𠡩 𠡪 𠡫 𠡬 𠡭 𠡮 𠡯 𠡰 𠡱 𠡲 𠡳 𠡴 𠡵 𠡶 𠡷 𠡸 𠡹 𠡺 𠡻 𠡼 𠡽 𠡾 𠡿 𠢀 𠢁 𠢂 𠢃 𠢄 𠢅 𠢆 𠢇 𠢈 𠢉 𠢊 𠢋 𠢌 𠢍 𠢎 𠢏 𠢐 𠢑 𠢒 𠢓 𠢔 𠢕 𠢖 𠢗 𠢘 𠢙 𠢚 𠢛 𠢜 𠢝 𠢞 𠢟 𠢠 𠢡 𠢢 𠢣 𠢤 𠢥 𠢦 𠢧 𠢨 𠢩 𠢪 𠢫 𠢬 𠢭 𠢮 𠢯 𠢰 𠢱 𠢲 𠢳 𠢴 𠢵 𠢶 𠢷 𠢸 𠢹 𠢺 𠢻 𠢼 𠢽 𠢾 𠢿 𠣀 𠣁 𠣂 𠣃 𠣄 𠣅 𠣆 𠣇 𠣈 𠣉 𠣊 𠣋 𠣌 𠣍 𠣎 𠣏 𠣐 𠣑 𠣒 𠣓 𠣔 𠣕 𠣖 𠣗 𠣘 𠣙 𠣚 𠣛 𠣜 𠣝 𠣞 𠣟 𠣠 𠣡 𠣢 𠣣 𠣤 𠣥 𠣦 𠣧 𠣨 𠣩 𠣪 𠣫 𠣬 𠣭 𠣮 𠣯 𠣰 𠣱 𠣲 𠣳 𠣴 𠣵 𠣶 𠣷 𠣸 𠣹 𠣺 𠣻 𠣼 𠣽 𠣾 𠣿 𠤀 𠤁 𠤂 𠤃 𠤄 𠤅 𠤆 𠤇 𠤈 𠤉 𠤊 𠤋 𠤌 𠤍 𠤎 𠤏 𠤐 𠤑 𠤒 𠤓 𠤔 𠤕 𠤖 𠤗 𠤘 𠤙 𠤚 𠤛 𠤜 𠤝 𠤞 𠤟 𠤠 𠤡 𠤢 𠤣 𠤤 𠤥 𠤦 𠤧 𠤨 𠤩 𠤪 𠤫 𠤬 𠤭 𠤮 𠤯 𠤰 𠤱 𠤲 𠤳 𠤴 𠤵 𠤶 𠤷 𠤸 𠤹 𠤺 𠤻 𠤼 𠤽 𠤾 𠤿 𠥀 𠥁 𠥂 𠥃 𠥄 𠥅 𠥆 𠥇 𠥈 𠥉 𠥊 𠥋 𠥌 𠥍 𠥎 𠥏 𠥐 𠥑 𠥒 𠥓 𠥔 𠥕 𠥖 𠥗 𠥘 𠥙 𠥚 𠥛 𠥜 𠥝 𠥞 𠥟 𠥠 𠥡 𠥢 𠥣 𠥤 𠥥 𠥦 𠥧 𠥨 𠥩 𠥪 𠥫 𠥬 𠥭 𠥮 𠥯 𠥰 𠥱 𠥲 𠥳 𠥴 𠥵 𠥶 𠥷 𠥸 𠥹 𠥺 𠥻 𠥼 𠥽 𠥾 𠥿 𠦀 𠦁 𠦂 𠦃 𠦄 𠦅 𠦆 𠦇 𠦈 𠦉 𠦊 𠦋 𠦌 𠦍 𠦎 𠦏 𠦐 𠦑 𠦒 𠦓 𠦔 𠦕 𠦖 𠦗 𠦘 𠦙 𠦚 𠦛 𠦜 𠦝 𠦞 𠦟 𠦠 𠦡 𠦢 𠦣 𠦤 𠦥 𠦦 𠦧 𠦨 𠦩 𠦪 𠦫 𠦬 𠦭 𠦮 𠦯 𠦰 𠦱 𠦲 𠦳 𠦴 𠦵 𠦶 𠦷 𠦸 𠦹 𠦺 𠦻 𠦼 𠦽 𠦾 𠦿 𠧀 𠧁 𠧂 𠧃 𠧄 𠧅 𠧆 𠧇 𠧈 𠧉 𠧊 𠧋 𠧌 𠧍 𠧎 𠧏 𠧐 𠧑 𠧒 𠧓 𠧔 𠧕 𠧖 𠧗 𠧘 𠧙 𠧚 𠧛 𠧜 𠧝 𠧞 𠧟 𠧠 𠧡 𠧢 𠧣 𠧤 𠧥 𠧦

Índice de Contenidos – Principios Básicos



Abstract/Resumen:

Se pretende conocer las funciones que puede realizar un ordenador, saber qué es un lenguaje de programación y los distintos tipos que existen y conocer la estructura conceptual de un ordenador. Por lo tanto una visión general del marco conceptual inicial en el que nos vamos a mover, poniendo de manifiesto la importancia de la programación como herramienta que impregna la actividad informática.



Principios Básicos

El ordenador es la herramienta que permite procesar automáticamente la información, facilitando en gran medida su organización, tratamiento, transmisión y almacenamiento.

El término Informática ha evolucionado a lo largo del tiempo. Al principio, se definió como la ciencia que se encargaba de estudiar el tratamiento automático de la información. Procede de la unión de dos palabras: información y automática.

En la actualidad, esta palabra engloba muchos más conceptos, ya que se aplica a la mayoría de los campos técnicos y científicos. La informática, hoy por hoy, se utiliza en la mayoría de los campos y actividades en las que el ser humano es partícipe. Se utiliza en medicina, aviación, construcción, diseño de automóviles, edificios, etc. y, por supuesto, se utiliza de forma personal para cubrir necesidades que hace unos años no existían.

Mediante **la Informática se gestiona información en forma de programas y datos** con los que es posible poner en marcha complejos sistemas electrónicos, mecánicos, etc. como, por ejemplo, controlar maquinaria, orientar y poner en órbita naves espaciales, controlar los ciclos de producción de cosechas, etc. Su desarrollo ha sido espectacular convirtiéndose en una herramienta imprescindible en comunicaciones, telefonía, Internet, vigilancia, control de tráfico, .. En resumen en todos los ámbitos de nuestra vida.

Conceptos y Definiciones

Datos

Son todos aquellos elementos considerados como unidades de tratamiento dentro de un sistema de proceso de datos. Los datos pueden ser básicamente de dos tipos, denominados datos de entrada que son aquellos pendientes de proceso o elaborar y datos de salida que son aquellos obtenidos tras el proceso de los datos de entrada. Al conjunto de los datos se le denomina información.

Tipo de Datos

Es una restricción impuesta para la interpretación, manipulación, representación de datos. Tipos de datos comunes en lenguajes de programación son los tipos primitivos (enteros, caracteres, ... etc.).

Conceptos y Definiciones

Programas

Son conjuntos finitos de órdenes diseñados y creados a través del razonamiento lógico y almacenados en ficheros de texto respetando la sintaxis de un determinado lenguaje de programación. Estos conjuntos de ordenes se trasmiten al ordenador para la realización y ejecución de tareas concretas

Aplicación informática

Es un conjunto de programas enlazados o relacionados entre sí.

Lenguaje de Programación

Conjunto finito de reglas léxicas, sintácticas y semánticas permitiendo al programador construir estructuras denominadas sentencias o instrucciones con las que elaborar programas que resuelven problemas concretos.

Conceptos y Definiciones

Sistema informático

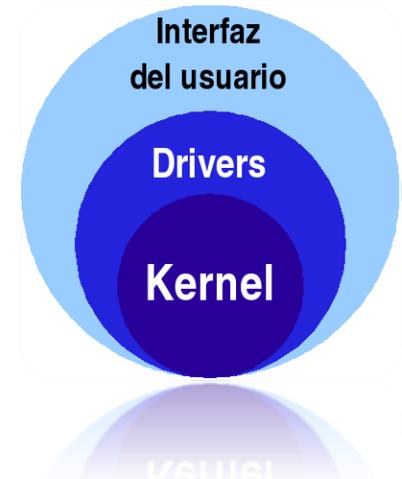
Es un conjunto de elementos que permiten procesar información por medio de equipos informáticos y cuya finalidad es la de obtener nueva información a partir de la ya existente. Todo sistema informático queda dividido de forma global en cuatro capas o niveles que son:

- ✦ Hardware
- ✦ Sistema operativo
- ✦ Programas de aplicación
- ✦ Recursos humanos, que son aquellas personas encargadas del desarrollo, implantación, explotación y mantenimiento de un sistema informático.



Sistema Operativo

- El objetivo fundamental de los sistemas operativos es **gestionar y administrar los recursos** del Sistema convirtiéndose en el mediador entre el usuario y el Hardware.
- Un sistema operativo es una capa compleja entre el hardware y el usuario, concebible también como una **máquina virtual**, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo. Por ejemplo, un usuario normal simplemente abre los archivos grabados en un disco, sin preocuparse por la disposición de los bits en el medio físico, los tiempos de espera del motor del disco, la posición del cabezal, el acceso de otros usuarios, etc.



Conceptos y Definiciones

Hardware - HW

Se denomina así a la parte física de un sistema informático, por ejemplo, un disco duro, un monitor, una tarjeta gráfica, cables, etc.

Los principales elementos que constituyen el hardware son:

Unidad central de proceso (**UCP, CPU**), es el verdadero cerebro de la maquina, y la encargada de controlar, coordinar y realizar todas las operaciones de un sistema informático.

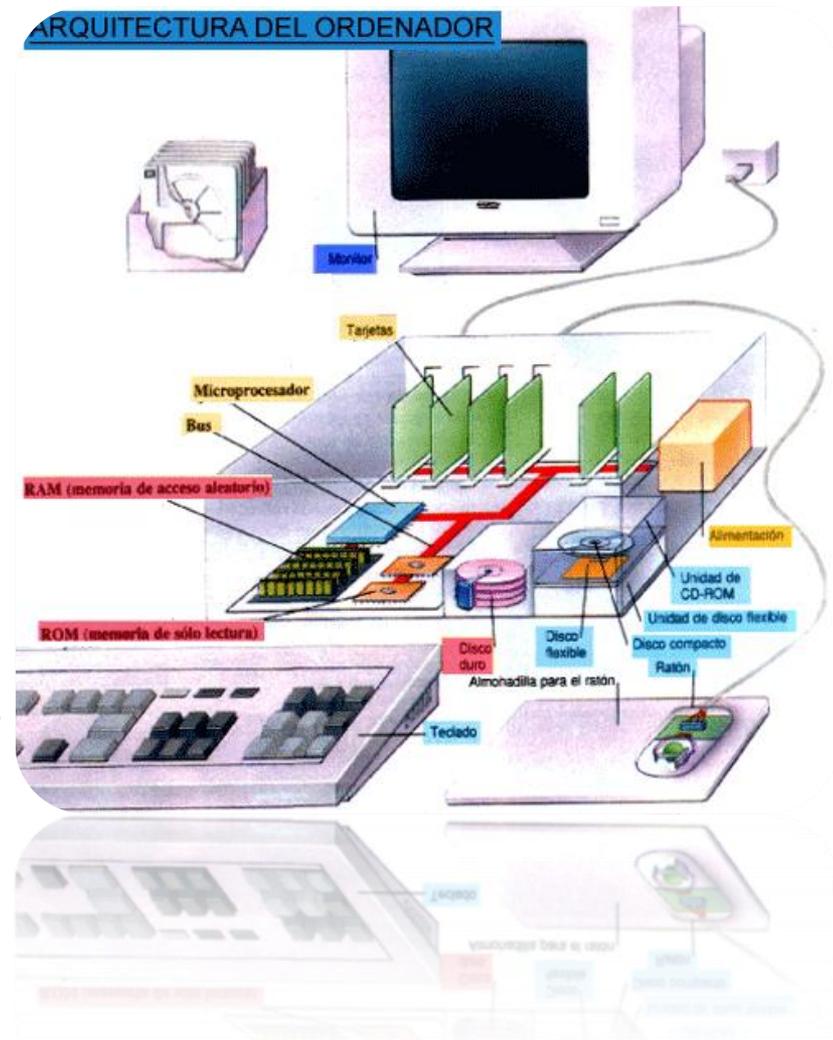
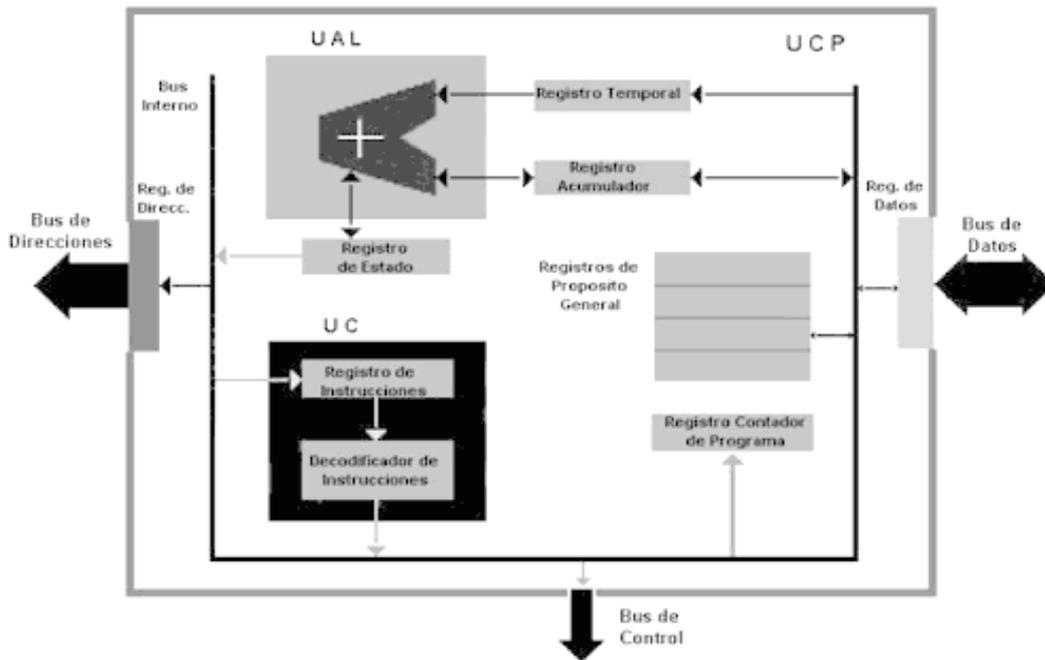
Los componentes básicos de una CPU son:

Unidad de control (**UC**): Es la parte capaz de detectar por medio de señales eléctricas el estado de cada uno de los elementos conectados al ordenador y gobierna las unidades de entrada, salida y entrada / salida, además de interpretar y ejecutar las instrucciones que constituyen los programas.

Unidad aritmético lógica (**U.A.L**): Es la parte del procesador encargada de realizar todas aquellas operaciones de tipo aritmético y tipo lógico.

Memoria: Se pueden diferenciar principalmente dos tipos de memoria, en primer lugar, memoria central o RAM. También se conoce comúnmente con el nombre de memoria principal, y es la parte del sistema donde se almacenan temporalmente los programas que se van a ejecutar junto con los datos que queremos procesar. En segundo lugar, la memoria auxiliar, son dispositivos de almacenamiento masivo de información. Los datos y programas pueden quedar almacenados de forma permanente, dando opción al usuario a recuperarlos en próximas sesiones de trabajo.

Conceptos y Definiciones



Conceptos y Definiciones

Periféricos

Un ordenador solo es capaz de ejecutar órdenes y de realizar operaciones tan básicas como sumar, restar, multiplicar y dividir valores numéricos, comparar valores numéricos o alfanuméricos, almacenar o recuperar la información.

Por tanto, su potencia y capacidad de cálculo dependerá básicamente de su eficacia, fiabilidad, rapidez y precisión, así como de la memoria disponible (tanto principal como auxiliar (la principal es la RAM y la auxiliar los soportes de almacenamiento permanente)). Con la combinación de operaciones básicas, un ordenador puede llegar a realizar operaciones o cálculos verdaderamente complejos, pero siempre existirá una estrecha dependencia de la máquina con el hombre, y sin la cual el ordenador es una herramienta carente de utilidad, pues el hombre es quien marca las pautas para su correcto funcionamiento a través de la lógica y el razonamiento.

En definitiva, un ordenador solo es capaz de aceptar datos de entrada, procesarlo y facilitar datos o resultados de salida. Los datos son introducidos o recuperados de un ordenador a través de los denominados periféricos o dispositivos externos de entrada / salida, que son los elementos destinados a auxiliar a la CPU (Unidad Central de Proceso) y establece una estrecha relación entre dicha unidad y el mundo exterior.

Conceptos y Definiciones

Según la función que desempeñen los periféricos se pueden clasificar en:

Periféricos de entrada

Son aquellos capaces de suministrar o introducir datos y ordenes a un sistema informático, entre los más usuales destacan el teclado, el ratón, lectores de código de barra, escáner óptico, etc.

Periféricos de salida

Son los encargados de sacar al exterior los resultados de proceso u operaciones realizadas por la CPU. Entre las más usuales destacan la pantalla con todas sus variantes, la impresora, el plotter.

Periféricos de entrada / salida

Son aquellos capaces de realizar cualquiera de las dos operaciones anteriores indistintamente, entre ellos destacan las unidades de disco duro, unidades de CD-ROM, de DVD, Streamer, disqueteras.

Periféricos de comunicación con otro sistema físico u ordenador.

Son aquellos encargados de establecer y facilitar el trasiego o intercambio de información entre dos o más ordenadores y bien entre un ordenador y otro sistema físico. Un claro ejemplo de este tipo de periféricos sería el MODEM, encargado de convertir señales digitales en señales analógicas y viceversa, un dispositivo ADSL/Cable MODEM....

Conceptos y Definiciones

Software - SW

Es la parte lógica o funcional de un sistema informático. Es decir la parte que hace funcionar al conjunto de elementos electrónicos de un ordenador permitiendo y coordinando cada uno de los componentes físicos. El software se divide principalmente en: **el software básico (BIOS), el sistema operativo (OS/2, Ubuntu, Leopard, Windows ...)** y **el software de aplicación (Macromedia, Office, Openffice ...)**.



¿Qué es un sistema operativo?

Un sistema operativo puede definirse como un conjunto de programas encargados de hacer posible el manejo del ordenador y la utilización de programas de aplicación, actuando como intermediario entre el usuario y el hardware.

En definitiva, se puede decir que la función de un sistema operativo es la de controlar el flujo de información entre dispositivos al mismo tiempo que facilitar la interactividad del usuario con la maquina de manera cómoda, así como la asignación de tareas y coordinación del funcionamiento interno del ordenador de manera eficaz, eficiente y segura.

Tipos de sistema operativo

Atendiendo al número de usuarios que puedan ser atendidos simultáneamente, el número de tareas o programas que se pueden ejecutar al mismo tiempo, el número de procesadores soportados y el tiempo de respuesta, los sistemas operativos se pueden clasificar en:

Sistemas monoprogramados: Se caracterizan por permitir la ejecución de un solo programa cada vez, por lo que no permitirá la ejecución de otro hasta que no finalice el anterior. El programa que se desea ejecutar es cargado en memoria y permanece en ella hasta que finaliza, adueñándose de la totalidad de los recursos del sistema, y que en este tipo de sistemas operativos no se da opción a que un segundo proceso participe de los mismos.

¿Qué es un sistema operativo?

Sistemas multiprogramados: También reciben el nombre de sistemas multitarea y se caracterizan básicamente por ser sistemas que aprovechan los tiempos de inactividad o los tiempos muertos de la CPU permitiendo la ejecución “simultanea” de varios programas, así como la utilización y el trabajo del procesador. El proceso es sencillo, se cargan en memoria varios programas y se divide el tiempo de proceso que la CPU dedicará a cada uno, lo que permite la ejecución alternativa de todos ellos, esto es lo que se denomina concurrencia de procesos.

Sistemas de multiproceso: Los sistemas operativos capaces de funcionar bajo ordenadores cuya arquitectura soporta 2 o más procesadores reciben el nombre de sistemas de multiproceso. A esta forma de trabajo se la suele llamar de proceso paralelo y asegura una mayor velocidad y seguridad.

Sistemas de tiempo compartido: Es lo que se conoce como multiprogramación interactiva. En este tipo de sistemas cada vez que los usuarios se conectan al ordenador, abren una sesión, que es el periodo de tiempo transcurrido desde que el usuario se conecta hasta que se despide. Este hecho crea un proceso capaz de atender a ese usuario facilitándole la comunicación con el sistema operativo.

¿Qué es un sistema operativo?

Sistemas de tiempo real: el tiempo de respuesta es el periodo de tiempo transcurrido desde que se realiza una petición o solicitud al sistema hasta que este responde. Si ese periodo de tiempo es muy breve (entre 1 msg. y 1 sg.) hablamos de tiempo real. Estos sistemas también considerados multiprogramados e interactivos, caracterizados por su rápida reacción y por manejar información que debe ser continuamente actualizada según los cambios producidos en tiempo real, por lo que requieren grandes restricciones en el tiempo de respuesta.

Sistemas monousuario: Son sistemas muy simples que solo permiten el acceso a un usuario cada vez, por lo que no se requieren ningún tipo de restricción o control en la gestión de los usuarios conectados. Este tipo de sistemas pueden basarse tanto en la monoprogramación como en la multiprogramación y suelen ser principalmente ordenadores personales.

Sistemas multiusuario: Son sistemas que se basan frecuentemente en sistemas multiprogramados permitiendo el acceso de varios usuarios a la vez. Estos tienen la posibilidad de ejecutar varios programas de forma concurrente, lo que permite rentabilizar al máximo el rendimiento del procesador.

Evolución y clasificación de lenguajes

Un lenguaje de programación es una notación o conjunto de símbolos y caracteres combinados entre sí de acuerdo con una sintaxis para posibilitar la transmisión y ejecución de instrucciones y datos a la CPU. Dichos símbolos y caracteres son traducidos a un conjunto de señales eléctricas representadas en código binario. La razón de convertir esos símbolos y caracteres a ceros y unos se debe a que el microprocesador solamente entiende ese lenguaje, que es el lenguaje binario o código máquina.



Evolución y clasificación de lenguajes

Lenguajes de bajo nivel

Son aquellos que por su característica se encuentran más próximos a la arquitectura de la maquina, englobándose en este tipo el lenguaje maquina y el lenguaje ensamblador.

Lenguaje maquina, se caracteriza principalmente por:

- ◆ *Ser considerado el primer lenguaje de programación.*
- ◆ *Ser el único lenguaje inteligible directamente por un ordenador.*
- ◆ *Basarse en la combinación de dos únicos símbolos, el 0 y el 1, denominados bits.*
- ◆ *Ser propio de un determinado procesador, es decir, que cada procesador tiene su propio y particular lenguaje maquina, que no podrá ser entendido por cualquier otro.*

Evolución y clasificación de lenguajes

Lenguaje ensamblador: Surge como sustituto del lenguaje maquina y esta basado en el uso de nemotécnicos (palabras abreviadas procedentes del inglés formadas por letras y a veces números). La programación en lenguaje ensamblador precisa de un amplio conocimiento sobre la constitución, estructura, y funcionamiento interno de un ordenador, así como un hábil manejo de los códigos y sistemas de numeración, en especial el binario y el hexadecimal.

```
-u 100 1a
OCFD:0100 BA0B01
OCFD:0103 B409
OCFD:0105 CD21
OCFD:0107 B400
OCFD:0109 CD21

MOV DX,010B
MOV AH,09
INT 21
MOV AH,00
INT 21

-d 10b 13f
OCFD:0100 20 65 73 74 65 20 65 73-20 75 6E 20 70 72 6F 67
OCFD:0110 72 61 6D 61 20 68 65 63-68 6F 20 65 6E 20 61 73
OCFD:0120 73 65 6D 62 6C 65 72 20-70 61 72 61 20 6C 61 20
OCFD:0130 57 69 6B 69 70 65 64 69-61 24
OCFD:0140
```

Hola,
este es un prog
rama hecho en as
sembler para la
Wikipedia\$

Evolución y clasificación de lenguajes

Lenguajes de alto nivel

Son aquellos que por sus características se encuentran más próximos al usuario o programador y se consideran como tales lenguajes como por ejemplo Basic, Pascal, Cobol y C. **Una de las características más importantes de estos lenguajes es que son independientes de la arquitectura del ordenador**, lo que implica que los programas desarrollados en lenguajes de alto nivel pueden ser ejecutados sobre ordenadores con distinto procesador. Este hecho hace que el programador no necesite poseer **amplios conocimientos sobre el funcionamiento interno del ordenador** que esta programando. Por otro lado cabe destacar una mayor facilidad en el desarrollo, depuración y mantenimiento de los programas.

```
parenthesis
  declare variable (optional)
    initialize      test      increment or
    ↓              ↓        ↓
for(int x = 0; x < 100; x++) {
    println(x); // prints 0 to 99
}
```

Evolución y clasificación de lenguajes

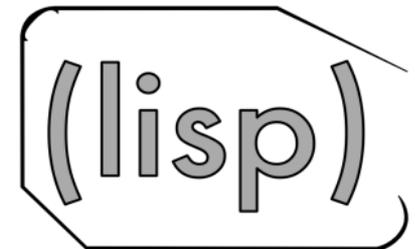
Lenguajes imperativos/estructurados. Utilizan instrucciones como unidad de trabajo de los programas (Cobol, Pascal, C, Ada).

Lenguajes declarativos. Los programas se construyen mediante descripciones de funciones o expresiones lógicas (Lisp, Prolog). Dentro de este punto tenemos los lenguajes lógicos basados en la lógica de primer orden y los lenguajes funcionales basados en *lambda-Calculo*.

Lenguajes orientados a objetos. La unidad de proceso es el objeto (instanciación de Clase la cual pertenece a una Jerarquía) y en el se incluyen los datos (variables) y las operaciones que actúan sobre ellos (Smalltalk, C++, java).

Lenguajes orientados al problema. Diseñados para problemas específicos, principalmente de gestión, suelen ser generadores de aplicaciones.

Lenguajes de Marcas. *Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento/datos, incorporando etiquetas o marcas que contienen/describen información adicional acerca de la estructura de dicho texto/datos o su presentación.*



Evolución y clasificación de lenguajes

Otra clasificación que se puede hacer es la de atendiendo al desarrollo de los lenguajes desde la aparición de las computadoras, que sigue un cierto paralelismo con las generaciones establecidas en la evolución de las mismas:

Primera generación. Lenguajes maquina y ensambladores.

Segunda generación. Primeros lenguajes de alto nivel imperativo (FROTRAN, COBOL).

Tercera generación. Lenguajes de alto nivel imperativo. Son los mas utilizados y siguen vigentes en la actualidad (ALGOL 8, PL/I, PASCAL, MODULA).

Cuarta generación. Orientados básicamente a las aplicaciones de gestión y al manejo de bases de datos (NATURAL, SQL).

Quinta generación. Orientados a la inteligencia artificial y al procesamiento de los lenguajes naturales (LISP, PROLOG).



Fases en la elaboración de un programa informático.

El desarrollo de una aplicación o conjunto de programas para obtener una solución informática a un determinado problema se basa en un concepto denominado ciclo de vida, que establece una serie de etapas o fases que hay que seguir secuencialmente y de forma ordenada cuando se desea desarrollar un determinado producto de software.

Estas fases son las siguientes: **análisis, diseño, codificación o construcción, explotación y mantenimiento.**

Análisis

En esta fase se establece cual es el producto que se va a desarrollar, siendo necesario especificar los procesos y estructuras de datos que se van a emplear para satisfacer la necesidad del usuario, por lo que debe existir una gran comunicación entre el usuario y el analista para conocer todas las necesidades y restricciones en el desarrollo de la aplicación.

Diseño

En esta fase se alcanza una solución óptima, detallada y con la mayor precisión posible para el desarrollo de la aplicación, teniendo en cuenta los recursos físicos del sistema (tipo de ordenador, periféricos, comunicaciones, etc.). y los recursos lógicos (sistema operativo, compiladores, bases de datos, etc.).

Fases en la elaboración de un programa informático.

Codificación

Consiste en la traducción de la solución obtenida a un determinado lenguaje de programación, basándonos en las especificaciones del diseño.

Explotación

En esta fase se realiza la implementación de la aplicación en el entorno operativo o sistema físico donde va a funcionar habitualmente y su puesto en marcha para obtener un funcionamiento normal de todo el sistema.

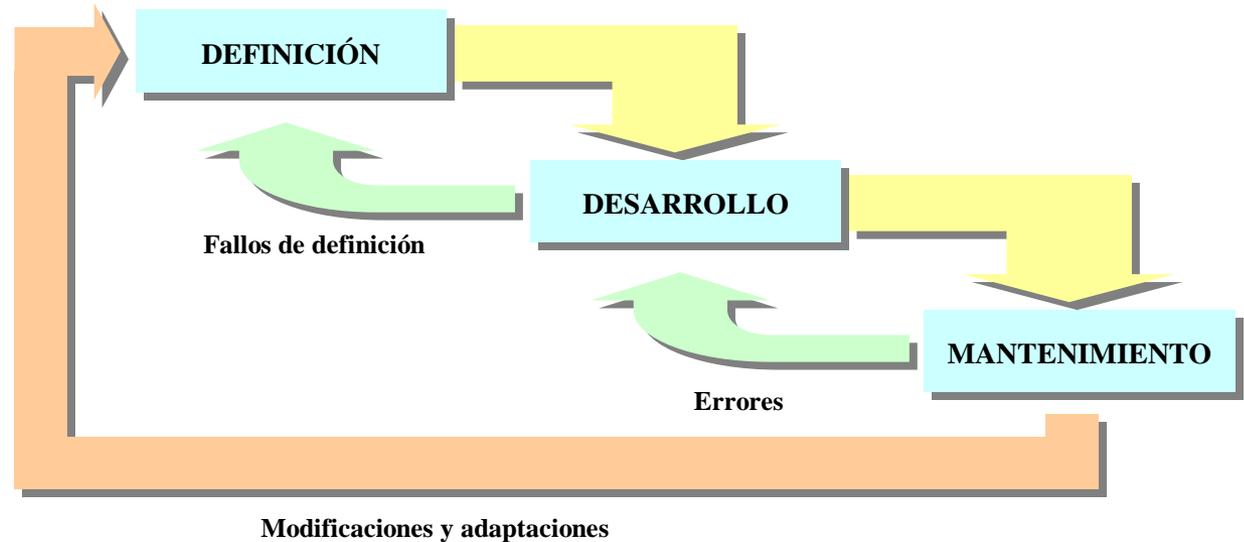
Algunas de las actividades que se realizan son: instalación de los programas, pruebas globales del sistema, traspaso de la información del sistema anterior al nuevo y por último, eliminación del sistema anterior.

Esta fase se complementará con la información del usuario con respecto al sistema y el uso de la aplicación.

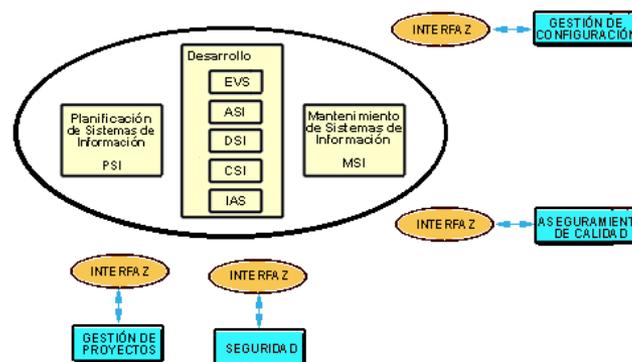
Mantenimiento

Esta fase completa el ciclo de vida y en ella se realizan las correcciones necesarias para resolver errores y deficiencias del producto desarrollado, existiendo la posibilidad de que ciertas acciones de esta fase puedan reiniciar el ciclo de vida.

Fases en la elaboración de un programa informático.



En este ámbito nos encontramos con **Métrica 3 un completa Metodología del desarrollo del SW.**



Ensambladores, compiladores e interpretes

Cuando programamos en un lenguaje distinto al lenguaje maquina, los programas diseñados deben ser traducidos a código binario, para que así las instrucciones que componen dicho programa puedan ser entendidas y ejecutadas por la UCP. Los programas encargados de llevar a cabo esta tarea reciben el nombre de traductores de lenguaje. Como tarea de guía al proceso de traducción, se comprueba que el programa fuente esta correctamente escrito de acuerdo con la sintaxis del lenguaje.

Ensambladores

Los ensambladores son los encargados de transformar o traducir los programas escritos en ensamblador a su equivalente en código maquina o binario para que pueda ser ejecutado por la UCP.

Intérpretes

Un intérprete es un programa encargado de procesar y traducir cada instrucción o sentencia de un programa escrito en un lenguaje de alto nivel a código maquina y después ejecutarla, es decir, que el procesador ejecuta la orden o instrucción una vez traducida y después de comprobar que no existe error alguno de sintaxis.

Ensambladores, compiladores e interpretes

Por tanto, un intérprete lee, examina, traduce y ejecuta una a una todas las instrucciones del programa fuente.

Los intérpretes suelen proporcionar un editor a través del cual se puede escribir el correspondiente programa fuente, facilitando así la edición y su posterior interpretación y ejecución, lo cual permitirá la comprobación y corrección de los posibles errores producidos durante el desarrollo del programa.

Compiladores

Un compilador es un programa traductor que a diferencia de los intérpretes, lleva a cabo la fase de traducción de dos formas, primero traduce completamente el programa fuente a código máquina y seguidamente lo ejecuta. Un programa compilado no necesita traducciones del programa fuente en ejecuciones sucesivas, lo que hace mucho más rápido el tiempo de ejecución. Se puede decir por tanto que los compiladores presentan todo lo bueno de los ensambladores y los intérpretes.

Ensambladores, compiladores e interpretes

Las etapas del proceso de compilación son:

Edición: Consiste en la escritura del programa utilizando un lenguaje de programación precisamente seleccionado y su posterior grabación en un soporte de almacenamiento permanente. La edición del programa debe realizarse mediante un editor de textos que puede formar parte o no del compilador. En esta fase se obtiene el denominado programa fuente.

Compilación: En esta fase se traduce el programa fuente a su equivalente en código maquina, obteniendo en caso de no producirse ningún error el denominado programa o modulo objeto. En caso de producirse errores, el compilador los mostrará utilizando los mensajes correspondientes, lo cual nos permitirá corregir el programa fuente y proceder de nuevo a su compilación.

Linkado: Esta fase también recibe el nombre de montaje y consiste en unir o enlazar el programa objeto con determinadas rutinas internas del lenguaje. Si el método de compilación es modular se enlazarán los distintos módulos para obtener así el programa ejecutable.

Ejecución: Esta fase consiste en la llamada del programa ejecutable a través del sistema operativo. Inicialmente se debe comprobar el buen funcionamiento del programa mediante el uso de unos juegos de pruebas que especifican los resultados que se desean obtener en función de unos determinados datos de entrada.

Tecnologías WEB – W3C (www.w3c.es/org)

El **World Wide Web Consortium**, abreviado **W3C**, es un consorcio internacional que produce recomendaciones para la World Wide Web -*sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet (conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial)* -. Está dirigida por Tim Berners-Lee, el creador original de **URL** (*Uniform Resource Locator*, Localizador Uniforme de Recursos -), **HTTP** (*HyperText Transfer Protocol*, Protocolo de Transferencia de HiperTexto) y **HTML** (*Lenguaje de Marcado de HiperTexto*) que son las principales tecnologías sobre las que se basa la Web.



Cuestiones y Ejercicios



WORLD WIDE WEB
CONSORTIUM®

Oficina Española

- + Define Ordenador.
- + Componentes de un Ordenador.
- + Operaciones básicas que realiza un ordenador.
- + Definición de Periférico.
- + Tipos de Periféricos.
- + Define dato e información.
- + Define programa y lenguaje de programación.
- + Componentes de un sistema de información.
- + Define Sistema de Información.
- + Pormenoriza los componentes de un PC.
- + Función del registro de instrucción.
- + Define sistema Operativo.
- + Clasifica los Sistemas Operativos.
- + Describe las características de algún sistema operativo.
- + Describe las características de tú sistema operativo.
- + Define Lenguaje de Programación.



- + Diferencia entre lenguaje máquina y lenguaje ensamblador.
- + Diferencia entre los lenguajes de bajo nivel y lenguajes de alto nivel.
- + Concepto de Ciclo de Vida.
- + Actividades de la etapa de Análisis.
- + Define Compilador.
- + Etapas de compilación.
- + Define linkado.
- + Resumen en un diagrama los tipos de Lenguajes de Programación.
- + Diferencias entre Compilación y Traducción.
- + ¿Qué significa W3C?. Indica su importancia.
- + ¿Qué son los Estándares Web?.
- + Define HTML , XHTML y CSS.
- + Recopila web referencias de XHTML y CSS.
- + Define y describe javascript y recopila 2 webreferencias.
- + Recopila guías rápidas de (x)html, css y javascript.
- + Herramientas CASE para el desarrollo de entornos Web.
- + Recopila guías rápidas o webreferencias de uso del entorno de desarrollo Macromedia Dreamweaver.

Índice de Contenidos

Sistemas de Representación

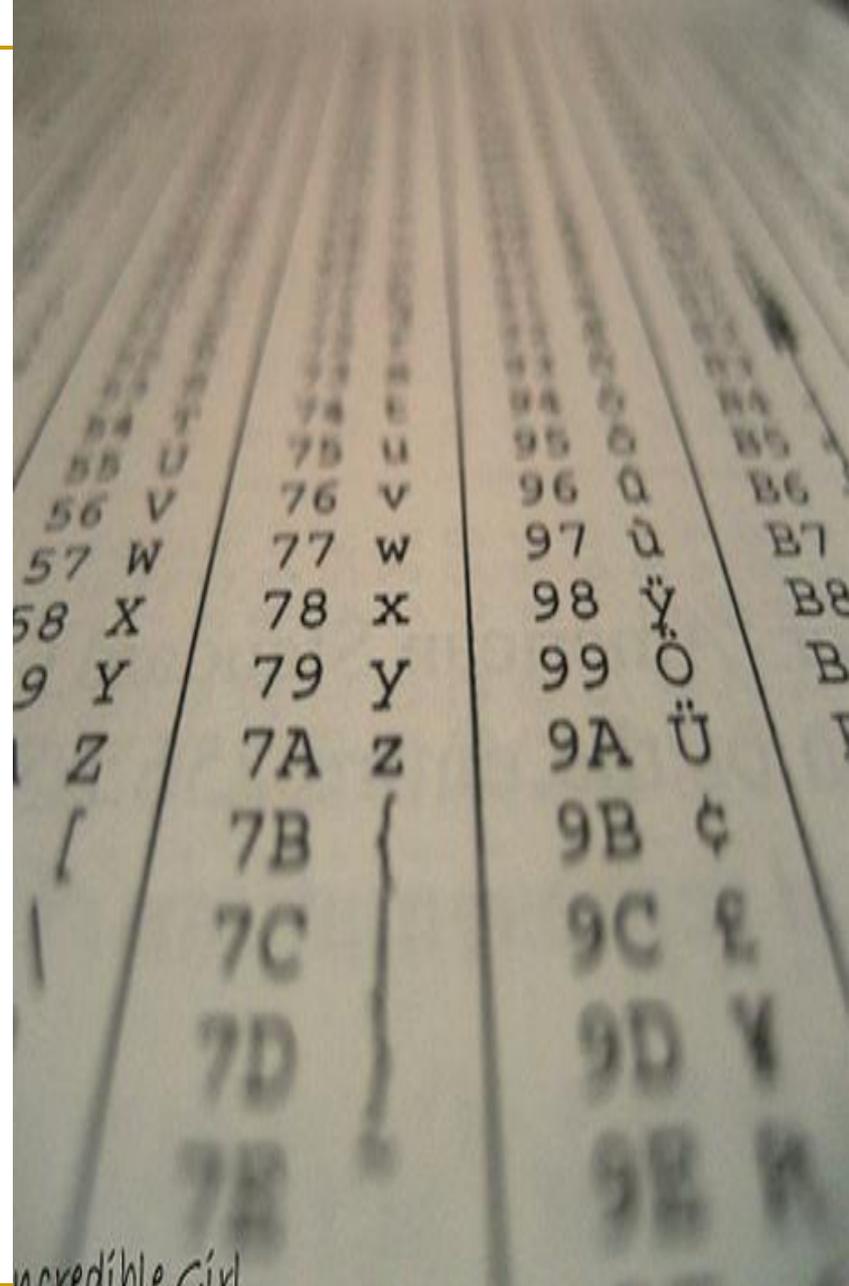


Abstract/Resumen:

La ejecución de un programa implica la manipulación de datos, según especifica un conjunto ordenado de instrucciones (un programa).

Para que el ordenador/PC ejecute un programa es preciso proporcionarle dos elementos: **instrucciones** que forman el programa y los **datos** con los que debe operar dichas instrucciones.

Uno de los aspectos más importantes relacionado con los datos, es cómo representarlos .



Sistema Decimal

Un dato es toda aquella información característica de una entidad, que es susceptible de tratamiento en un programa informático. Para todo ordenador, internamente, toda la información es numérica (combinaciones de ceros y unos - código binario). Por ello, para su correcto tratamiento, todos los datos que forman dicha información deben estar representados de tal forma que los programas sean capaces de realizar las debidas operaciones sobre ellos, sin dar lugar a error o fallos por parte del sistema.

El sistema decimal es un sistema posicional, ya que el significado de un símbolo depende fundamentalmente de su posición relativa al símbolo coma (,), denominado *coma decimal*, que en caso de ausencia se supone colocada implícitamente a la derecha.

Utiliza como base el 10, que corresponde al número de símbolos que comprenden para la representación de cantidades; estos símbolos (también denominados dígitos) son:

0 1 2 3 4 5 6 7 8 9

Sistema Decimal

Una determinada cifra, que se denominará *número decimal*, se puede expresar de la siguiente forma:

$$N^{\circ} = \sum_{i=-d}^n (\text{dígito})_i * (\text{base})^i$$

Donde:

base = 10

i = posición respecto a la coma

d = n.º de dígitos a la derecha de la coma,

n = n.º de dígitos a la izquierda de la coma - 1,

dígito = cada uno de los que componen el número

La fórmula responde al **Teorema Fundamental de la Numeración** que será descrito en el siguiente punto.

El sistema decimal es un sistema posicional como ya hemos dicho, ya que el mismo dígito puede variar su valor de acuerdo a su posición.

Ej.:

1000	mil
100	cien
10	diez
1	uno
0,1	un décimo
0,01	un centésimo

Teorema fundamental de la numeración

El teorema fundamental de la numeración dice:

“El valor en el sistema decimal de una cantidad expresada en otro sistema cualquiera de numeración, viene dado por la fórmula:

$$\dots + X_4 * B^4 + X_3 * B^3 + X_2 * B^2 + X_1 * B^1 + X_0 * B^0 + X_{-1} * B^{-1} + X_{-2} * B^{-2} + X_{-3} * B^{-3} + \dots”$$

donde **X** es el dígito y **B** la base.

Ejemplo:

Supongamos la cantidad $3221,03_4$ esta expresada en base 4 (ver subíndice al final de la cantidad), dicha base utiliza para representar cantidades los dígitos 0, 1, 2 y 3. ¿Cuál será el valor correspondiente en el sistema decimal?

$$3 * 4^3 + 2 * 4^2 + 2 * 4^1 + 1 * 4^0 + 0 * 4^{-1} + 3 * 4^{-2} = 3 * 64 + 2 * 16 + 2 * 4 + 1 * 1 + 0 * 0,25 + 3 * 0,0645 = 233,1875$$

Sistema Binario

Por razones técnicas, la mayoría de los circuitos electrónicos que conforman un ordenador solo puede detectar la presencia o ausencia de tensión en el circuito. Si a la presencia de tensión en un punto del circuito le asignamos el valor 1 y a la ausencia de la misma el valor 0 (a esta lógica se la denomina **lógica positiva**). Caso contrario la denominaremos **lógica negativa**.

Por las razones antes vistas, ya que el hardware por el momento solo reconoce estos dos estados fue necesario crear un sistema de numeración basado en estos dos valores (0, 1), al cual se lo denominó Binario, y cuya base por lo tanto es 2 (números de dígitos del sistema).

En computación cada dígito de un número representado en este sistema se denomina **bit** (contracción de **binary digit**).

Como múltiplos del bit hallamos:

8 bits	≡	Byte (palabra)	B (10110110)
1024 bytes	≡	1 kilobyte	KB
1024 KB	≡	1 Megabyte	MB
1024 MB	≡	1 Gigabyte	GB
1024 GB	≡	1 Terabyte	TB
1024 TB	≡	1 Petabyte	PB
1024 PB	≡	1 Zettabyte	ZB
1024 ZB	≡	1 Yottabyte	YB

Sistema Binario

Con n bits puedo representar 2^n objetos del universo-dominio del problema

Para representar n objetos del dominio del problema necesito \log_2^n bits

Utilizando calculadora

Una forma simple para calcular el $\log_2(n)$ en una calculadora que no posee la función \log_2 es utilizar el **logaritmo natural** (base e , indicado como \ln) o el **logaritmo común** (base 10 indicado como \log), los cuales se encuentran en la mayoría de las calculadoras científicas. La fórmula para esto es:

$$\log_2(n) = \frac{\ln(n)}{\ln(2)} = \frac{\log(n)}{\log(2)}$$

Sistema Octal

Es un sistema cuya base es el número 8, es decir, utiliza 8 símbolos para la representación de un valor cualquiera. Estos símbolos son:

0 1 2 3 4 5 6 7

Este es un sistema también posicional, de aritmética muy similar al decimal. Su utilización comenzó como sistema de salida de las computadoras ya que para representar un valor la cantidad de símbolos que necesita es menor que el binario y la conversión entre ambos sistemas es muy sencilla de implementar.

Sistema Hexadecimal

Es un sistema cuya base es el número 16, es decir, utiliza 16 símbolos para la representación de un valor cualquiera. Estos símbolos son:

0 1 2 3 4 5 6 7 8 9 A B C D E F

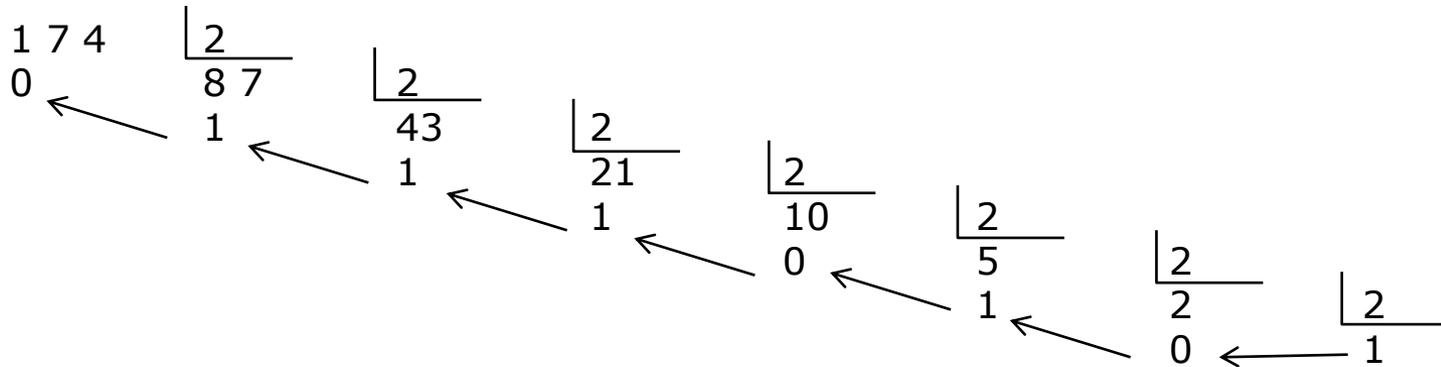
Este es otro sistema posicional, de característica similar al octal. Su uso fue adoptado por idénticas razones que el octal.

Conversión entre los distintos sistemas

Conversión decimal a binario

Para convertir un número decimal entero a binario, este debe ser dividido por dos y repetir el proceso con sus cocientes hasta que el cociente tome el valor 1. La unión de todos restos escritos en orden inverso encabezados por el último cociente, nos dará el valor expresado en binario.

Ej. : Convertir el número 174 a binario



$$= 10101110_2$$

Conversión entre los distintos sistemas

Conversión decimal a binario

Para convertir una fracción decimal a binario, esta fracción debe ser multiplicada por dos y tomamos la parte entera del resultado, repetimos el proceso con la parte fraccionaria del resultado anterior, dándonos una nueva parte entera, y así sucesivamente hasta que la parte fraccionaria se haga 0 (cero) o que tengamos suficientes decimales que nos permita estar debajo de un determinado error.

Ej. : Convertir el número 0,90625 a fracción binaria

$$\begin{array}{rcl} 0,90625 * 2 = & \downarrow & 1,8125 \\ 0,8125 * 2 = & & 1,625 \\ 0,625 * 2 = & & 1,25 \\ 0,25 * 2 = & & 0,5 \\ 0,5 * 2 = & & 1, \\ 0,90625_{10} = & & 0,11101_2 \end{array}$$

Si se desea convertir un número que tiene parte entera y decimal a binario, se deberá operar cada parte por separado como ya se ha visto, y luego obtener la suma de los resultados.

Por ejemplo:

$$174,90625_{10} = 10101110,11101_2$$

Conversión entre los distintos sistemas

Conversión binario a decimal

Para realizar esta conversión se utiliza como base el teorema fundamental de la numeración.

El método práctico consiste en multiplicar cada uno de los términos por potencias crecientes de 2 a partir de la coma decimal y hacia la izquierda, y realizar la suma de las operaciones.

Por ejemplo:

Pasar a decimal el binario 10101110_2

1	0	1	0	1	1	1	0	
							→	$0 * 2^0 = 0$
							→	$1 * 2^1 = 2$
							→	$1 * 2^2 = 4$
							→	$1 * 2^3 = 8$
							→	$0 * 2^4 = 0$
							→	$1 * 2^5 = 32$
							→	$0 * 2^6 = 0$
							→	$1 * 2^7 = 128$
								174

$$10101110_2 = 174_{10}$$

Conversión entre los distintos sistemas

Conversión binario a decimal

En los casos de números que posean parte entera y decimal se recomienda el uso del teorema fundamental de la numeración.

Ej.: Convertir $1101,011_2$ a base 10

Para pasar a base 10 deberemos hacer:

$$\begin{aligned}1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2} + 1 * 2^{-3} &= \\1 * 8 + 1 * 4 + 0 + 1 * 1 + 0 + 1 * 0,25 + 1 * 0,125 &= \\8 + 4 + 0 + 1 + 0 + 0,25 + 0,125 &= 13,375\end{aligned}$$

$$1101,011_2 = 13,375_{10}$$

Conversión entre los distintos sistemas

Conversión octal a binario

Al ser la base del octal (8) potencia de la base binaria (2^3), la transformación de una base a la otra se hace en forma directa dígito a dígito. Cada dígito octal será reemplazado por 3 dígitos binarios (3 por ser la potencia que relaciona ambas bases), según la tabla que tenemos a continuación.

Ej.:

Convertir a binario el número $276,534_8$

2	7	6,	5	3	4
010	111	110,	101	011	100

Octal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$$276,534_8 = 10111110,1010111_2$$

Como se puede ver los ceros al comienzo se han quitado, igual que los ceros que se hallan a la derecha de la coma (ya que no tienen ningún sentido).

Conversión entre los distintos sistemas

Conversión binario a octal

Esta conversión es similar a la anterior, pero cada tres símbolos binarios corresponde uno octal. Para realizar correctamente esta conversión el **número de dígitos** a la derecha de la coma decimal debe ser múltiplo de 3 si no lo fuera deberá **agregarse al final** del número tantos ceros como sea necesario. Idéntico caso será a la izquierda de la coma, en dicho caso los ceros se agregan al principio del número.

Ej.

Convertir el binario 10101011,0011 a octal.

010	101	011,	001	100
2	5	3,	1	4

0 cero agregado al número para permitir la correcta conversión.

$$10101011,0011_2 = 253,14_8$$

Conversión entre los distintos sistemas

Conversión hexadecimal a binario

Por idénticas razones que el caso anterior ($16 = 2^4$), la transformación de una base a la otra se hace en forma directa dígito a dígito. Cada dígito hexadecimal será reemplazado por 4 dígitos binarios (4 por ser la potencia que relaciona ambas bases), según la tabla que tenemos a continuación.

Hexadecimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Hexadecimal	Binario
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Ej.:

Convertir a binario el número $5A8,39C_{16}$

5	A	8,	3	9	C
0101	1010	1000,	0011	1001	1100

$5A8,39C_{16} = 10110101000,0011100111_2$

Como se puede ver otra vez los ceros al comienzo se han quitado, igual que los ceros que se hallan a la derecha de la coma (ya que no tienen ningún sentido).

Conversión entre los distintos sistemas

Conversión binario a hexadecimal

Esta conversión es similar a la conversión a octal, pero en lugar de tres, serán cuatro símbolos binarios los que corresponde a un hexadecimal. Para realizar correctamente esta conversión el **número de dígitos** a la derecha de la coma decimal debe ser múltiplo de 4 si no lo fuera deberá **agregarse al final** del número tantos ceros como sea necesario. Idéntico caso será a la izquierda de la coma, en dicho caso los ceros se agregan al principio del número.

Ej.

Convertir el binario 1010101011,00111 a hexadecimal.

0010	1010	1011,	0011	1000
2	A	B,	3	8

0 cero agregado al número para permitir la correcta conversión.

$$1010101011,00111_2 = 2AB,38_{(16)}$$

Conversión entre los distintos sistemas

Conversión decimal a octal o hexadecimal

Para cualquiera de estos dos casos se hará en forma similar a la explicada para convertir de decimal a binario 1.7.1. Pero se deberá tener en cuenta que la base ya no es 2, sino 8 o 16 según corresponda. (Dividir por 8 o 16)

Conversión octal o hexadecimal a decimal

Para cualquiera de estos dos casos se deberá usar el teorema fundamental de la numeración, teniendo en cuenta base que corresponda (8 o 16 según el caso).

Conversión octal a hexadecimal o hexadecimal a octal.

Estas conversiones no son posibles en una forma directa. Para realizar cualquiera de ellas se deberá usar el pasaje a otra base como paso intermedio.

Por ejemplo octal \Leftrightarrow decimal \Leftrightarrow hexadecimal

octal \Leftrightarrow binario \Leftrightarrow hexadecimal

Se recomienda como metodología de trabajo esta última, porque al ser las operaciones de conversión más sencillas disminuye la probabilidad de error. Además no existe la posibilidad de errores de redondeo.

Representación de números enteros

Existen diversas formas de representar un número entero en un ordenador (todos en sistema binario), entre ellas son:



En todos los casos se considera que tenemos un número limitado de dígitos para cada elemento numérico. El número de dígitos disponibles lo representa N (8, 16, 32, 64 o sea 1, 2, 3, 4... Bytes).

Representación interna de la información:

codificación alfanumérica

Todas las letras, números y símbolos tienen asignados valores codificados con 1 y 0.

Tres juegos de códigos comunes son:

ASCII (para las PCs y UNIX y DOS computadoras) - **American Standard Code for Information Interchange** — *Código Estadounidense Estándar para el Intercambio de Información*

EBCDIC (para IBM System 390 computadoras de gran porte) - **Extended Binary Coded Decimal Interchange Code**

Unicode (para Windows NT y exploradores de Web reciente)

El código **ASCII** utiliza 7 bits por carácter, permitiendo 128 diferentes caracteres, lo que es suficiente para el alfabeto en letras mayúsculas y minúsculas y los símbolos de una máquina de escribir corriente además de algunas combinaciones reservadas para su uso interno. Un código ASCII extendido usa 8 bits por carácter lo que añade otros 128 caracteres posibles. Este juego de códigos más amplio permite que se agreguen los símbolos de lenguajes extranjeros y varios símbolos gráficos. El **Unicode** usa 16 bits por carácter, por lo que para los mismos caracteres insume el doble de espacio que con el código ASCII. Pero Unicode puede manejar muchos más caracteres. La meta de Unicode es representar cada elemento usado en la escritura de cualquier idioma del planeta.

Representación interna de la información: codificación alfanumérica

El código ASCII – www.elCodigoASCII.com.ar

sigla en inglés de American Standard Code for Information Interchange
(Código Estadounidense Estándar para el Intercambio de Información)

Caracteres de control ASCII				Caracteres ASCII imprimibles									ASCII extendido											
DEC	HEX	Simbolo ASCII		DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo			
00	00h	NULL	(carácter nulo)	32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
01	01h	SOH	(inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	ü	161	A1h	í	193	C1h	ł	225	E1h	õ
02	02h	STX	(inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	é	162	A2h	ó	194	C2h	Ł	226	E2h	ö
03	03h	ETX	(fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	ú	195	C3h	ł	227	E3h	ÿ
04	04h	EOT	(fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ñ	196	C4h	ł	228	E4h	ÿ
05	05h	ENQ	(enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	å	165	A5h	Ñ	197	C5h	ł	229	E5h	ÿ
06	06h	ACK	(acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	ä	166	A6h	ª	198	C6h	ł	230	E6h	ÿ
07	07h	BEL	(timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	º	199	C7h	ł	231	E7h	ÿ
08	08h	BS	(retroceso)	40	28h	(72	48h	H	104	68h	h	136	88h	è	168	A8h	¿	200	C8h	ł	232	E8h	ÿ
09	09h	HT	(tab horizontal)	41	29h)	73	49h	I	105	69h	i	137	89h	ë	169	A9h	®	201	C9h	ł	233	E9h	ÿ
10	0Ah	LF	(salto de línea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	è	170	AAh	¬	202	CAh	ł	234	EAh	ÿ
11	0Bh	VT	(tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ï	171	ABh	½	203	CBh	ł	235	EBh	ÿ
12	0Ch	FF	(form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	î	172	ACh	¼	204	CCh	ł	236	ECh	ÿ
13	0Dh	CR	(retorno de carro)	45	2Dh	-	77	4Dh	M	109	6Dh	m	141	8Dh	ï	173	ADh	ı	205	CDh	ł	237	EDh	ÿ
14	0Eh	SO	(shift Out)	46	2Eh	.	78	4Eh	N	110	6Eh	n	142	8Eh	Ë	174	A Eh	«	206	CEh	ł	238	EEh	ÿ
15	0Fh	SI	(shift In)	47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	À	175	AFh	»	207	CFh	ł	239	EFh	ÿ
16	10h	DLE	(data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	É	176	B0h	⋮	208	D0h	ł	240	F0h	ÿ
17	11h	DC1	(device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h	⋮	209	D1h	ł	241	F1h	±
18	12h	DC2	(device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h	⋮	210	D2h	ł	242	F2h	ˆ
19	13h	DC3	(device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h	ø	179	B3h	⋮	211	D3h	ł	243	F3h	¾
20	14h	DC4	(device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h	ò	180	B4h	ł	212	D4h	ł	244	F4h	¸
21	15h	NAK	(negative acknowle.)	53	35h	5	85	55h	U	117	75h	u	149	95h	ó	181	B5h	ł	213	D5h	ł	245	F5h	¸
22	16h	SYN	(synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h	ù	182	B6h	ł	214	D6h	ł	246	F6h	¸
23	17h	ETB	(end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h	ú	183	B7h	ł	215	D7h	ł	247	F7h	¸
24	18h	CAN	(cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h	ł	216	D8h	ł	248	F8h	¸
25	19h	EM	(end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h	Û	185	B9h	ł	217	D9h	ł	249	F9h	¸
26	1Ah	SUB	(substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	Ü	186	BAh	ł	218	DAh	ł	250	FAh	¸
27	1Bh	ESC	(escape)	59	3Bh	;	91	5Bh	[123	7Bh	{	155	9Bh	ø	187	BBh	ł	219	DBh	ł	251	FBh	¸
28	1Ch	FS	(file separator)	60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch	£	188	BCh	ł	220	DCh	ł	252	FCh	¸
29	1Dh	GS	(group separator)	61	3Dh	=	93	5Dh]	125	7Dh	}	157	9Dh	Ø	189	BDh	ł	221	DDh	ł	253	FDh	¸
30	1Eh	RS	(record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	x	190	BEh	ł	222	DEh	ł	254	FEh	¸
31	1Fh	US	(unit separator)	63	3Fh	?	95	5Fh	_				159	9Fh	f	191	BFh	ł	223	DFh	ł	255	FFh	¸

Representación interna de la información:

codificación alfanumérica

Unicode

El **Estándar Unicode** es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de múltiples lenguajes y disciplinas técnicas además de textos clásicos de lenguas muertas. El término Unicode proviene de los tres objetivos perseguidos: universalidad, uniformidad y unicidad.

Este estándar es mantenido por el *Unicode Technical Committee* (UTC), integrado en el Unicode Consortium, del que forman parte con distinto grado de implicación empresas como: Microsoft, Apple, Adobe, IBM, Oracle, SAP, Google, instituciones como la Universidad de Berkeley, y profesionales y académicos a título individual. El Unicode Consortium mantiene estrecha relación con ISO/IEC (International Organization for Standardization/Comisión Electrotécnica Internacional), con la que mantiene un acuerdo desde 1991 con el objetivo de mantener la sincronización entre sus estándares que contienen los mismos caracteres y puntos de código.

El establecimiento de Unicode ha sido un ambicioso proyecto para reemplazar los esquemas de codificación de caracteres existentes, muchos de los cuales están muy limitados en tamaño y son incompatibles con entornos plurilingües.

Cuestiones y Ejercicios

1 - Pasar a base 10 los siguientes números, de las bases indicadas:

1101_2		$0,101_2$		$101,11_2$		$1,0111_2$		753_8
$0,63_8$		$17,134_8$		$3A_{16}$		$0,FF_{16}$		$A5,3B_{16}$

2.- Pasar los siguientes números de base 10 a la base indicada:

$39 \Rightarrow_2$		$0,525 \Rightarrow_2$		$23,945 \Rightarrow_2$		$123 \Rightarrow_8$
$3,1 \Rightarrow_8$		$0,14 \Rightarrow_8$		$1068 \Rightarrow_{16}$		$61,6 \Rightarrow_{16}$

3.- Completa las siguientes tablas con las transformaciones correspondientes:

Binario	Octal	Hexadecimal
100101011		
	7071	
		A2FF

Cuestiones y Ejercicios

4 - Pasar el siguiente decimal a la base indicada:

Número	Base
0,267	2
52,38	2
129,64	2
163,97	8
954,62	16

5 - Pasar a las bases indicadas:

$32_8 \Rightarrow$	2	$F1_{16} \Rightarrow$	8	$F1_{16} \Rightarrow$	2
$73_8 \Rightarrow$	16	$1010_2 \Rightarrow$	16	$10,10_2 \Rightarrow$	8

Cuestiones y Ejercicios

6. - Pasar a las bases indicadas:

$32_8 \Rightarrow$	2	$F1_{16} \Rightarrow$	8	$F1_{16} \Rightarrow$	2
$73_8 \Rightarrow$	16	$1010_2 \Rightarrow$	16	$10,10_2 \Rightarrow$	8

7. - Pasar a Unicode y ASCII:

Lenguaje	España
Juan López	Hardware
Cigüeña	juanlop@um.es

Cuestiones y Ejercicios

Prueba Teórico/Práctica:

<input type="checkbox"/>		Cuestión 1.-				
M=0	$R_{\downarrow}=0,25$	R=0,5	B=0,75	MB=1		

Describe **Compilación y Traducción**

<input type="checkbox"/>		Cuestión 2.-				
M=0	$R_{\downarrow}=0,25$	R=0,5	B=0,75	MB=1		

Define **Lenguaje de Programación, programa y compilador**

<input type="checkbox"/>		Cuestión 3.-				
M=0	$R_{\downarrow}=0,25$	R=0,5	B=0,75	MB=1		

Describe **El Teorema Fundamental de la Numeración (TFN)**

<input type="checkbox"/>		Cuestión 4.-				
M=0	$R_{\downarrow}=0,25$	R=0,75	B=1	MB=1,5		

$A5,3B_{16} = X_8, Y_2, Z_{10}$

Cuestiones y Ejercicios

<input type="checkbox"/>		Cuestión 5.-				
M=0	R↓=0,25	R=0,75	B=1	MB=1,5		

$13,17_{10} = X_{(8)}, Y_{(2)}, Z_{(16)}$

<input type="checkbox"/>		Cuestión 6.-				
M=0	R↓=0,5	R=1	B=1,50	MB=2		

¿Qué son los Estándares Web?. Define HTML, XHTML y CSS.

<input type="checkbox"/>		Cuestión 7.-				
M=0	R↓=0,5	R=1	B=1,50	MB=2		

Codificación alfanumérica. **Codificación ASCII-UNICODE**