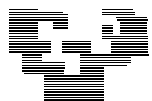


5. Gaia

Programen garapena

Programa baten egitura
Fluxu diagramak
Egoera makinak
Gertaera makinak
Egoera-gertaera makinak
Moduluzko programazioa
Parametroak pasatu
Pila erabili



Egituraketa



Prozesagailuek sekuentzialki lan egingo balute aintzinako ehungailuak bezala, lan talde batek mantendu eta sortu ezingo lituzkeen programa lista luzeak izango genituzke.



$$3xa+3xb+3xc$$



a
+
a
+
a
+
b
+
b
+
b
+
c
+
c
+
c

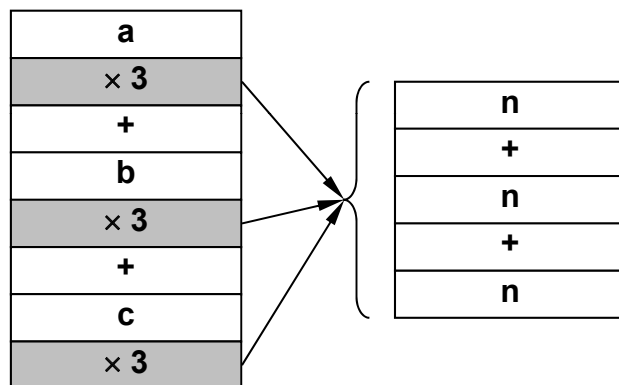


Egituraketa

Baina “dei” aginduei esker (CALL-RETURN), errutinak sortu ditzazkegu:
berriz erabiliak izan daitezkeen programa zatiak.

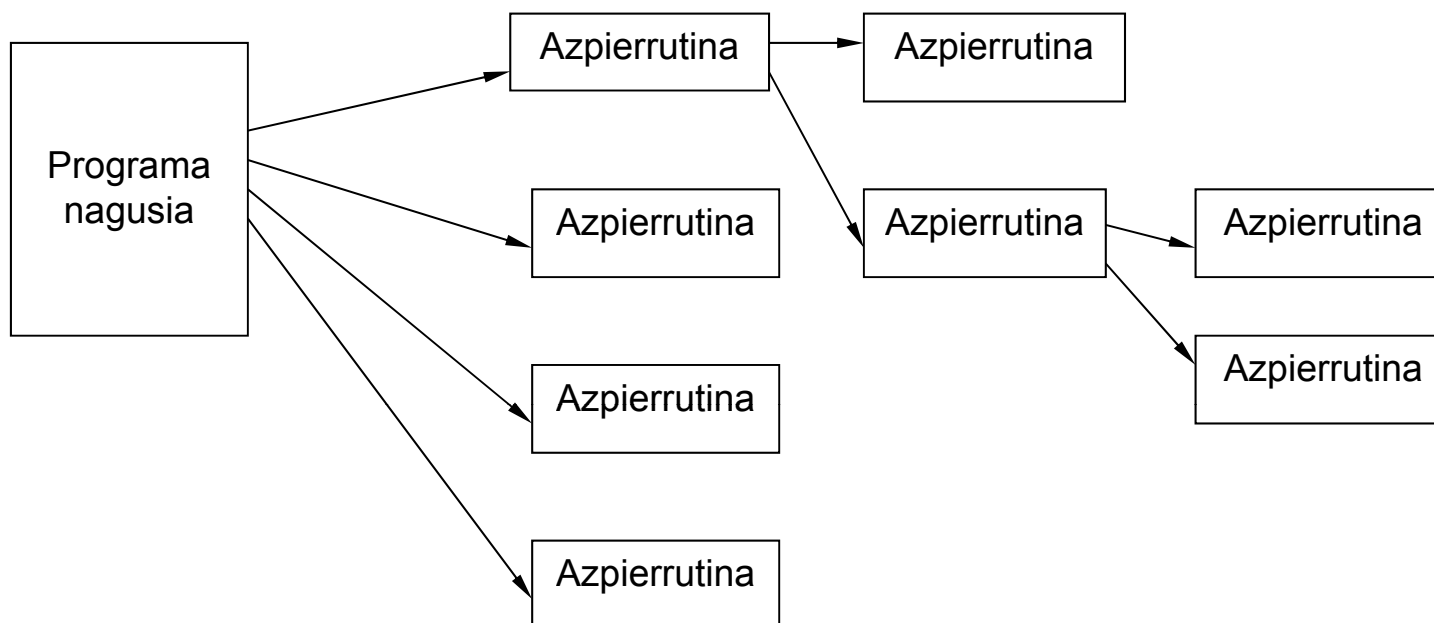
Programa zati hauei “errutina” edo “azpierrutina” deituko diegu.

Errutina bakoitzaren diseinua pertsona ezberdinei esleitu diezaiekegu:
talde lana



Egituraketa

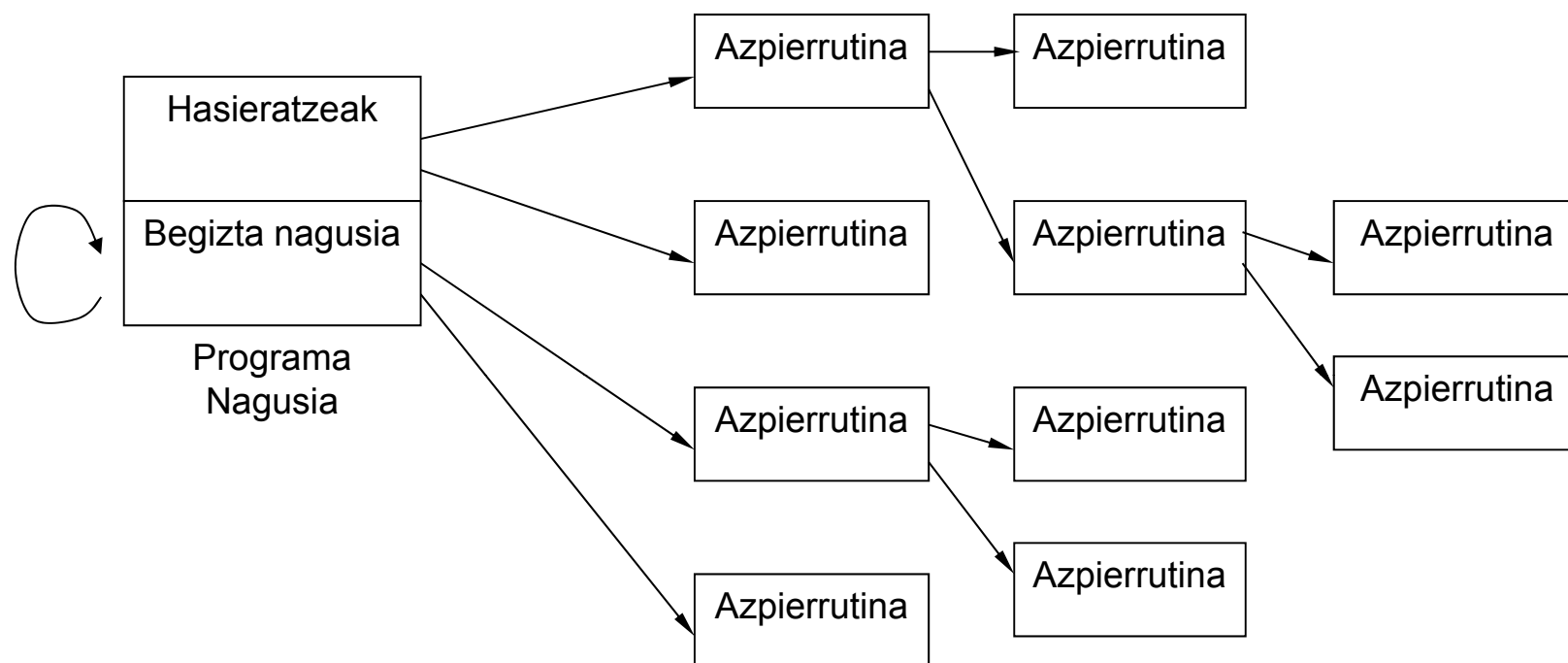
Prozesadore bat kontrolatzen duen edozein programa nagusi azpierrutinei dei egiten dien segida ordenatu bat izango da.



Printzipioz, edozein programa nagusi bi etapetan zatitzen da:

Hasieratze etapa

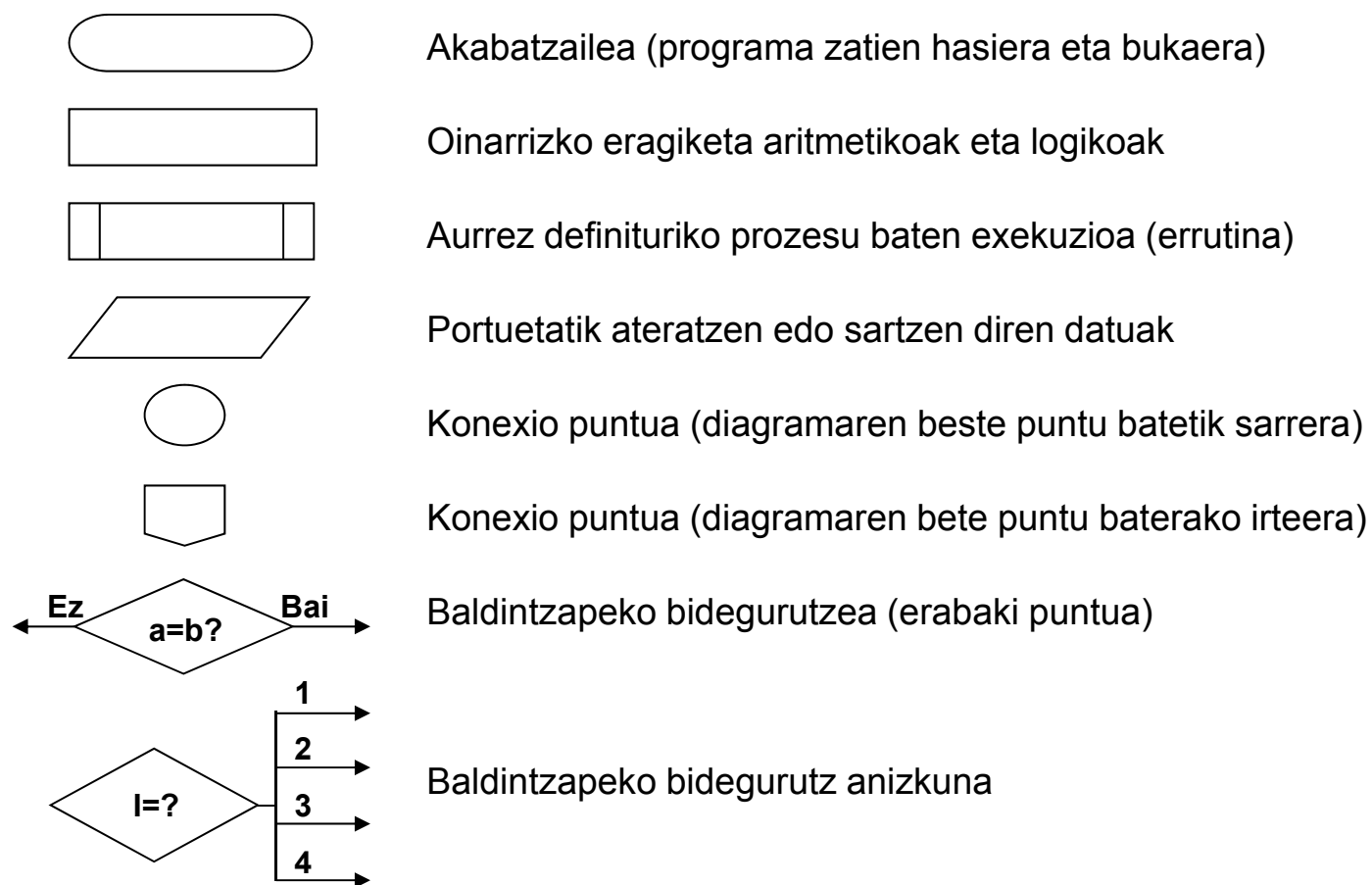
Begizta nagusia



Fluxu diagramak

Fluxu diagramek ataza bat burutzeko makina batek egin behar dituen operazioak era ulergarri baten antolatzea ahalbidetzen dute.

Ikasi behar diren oinarritzko ikur batzuk daude



Fluxu diagramak

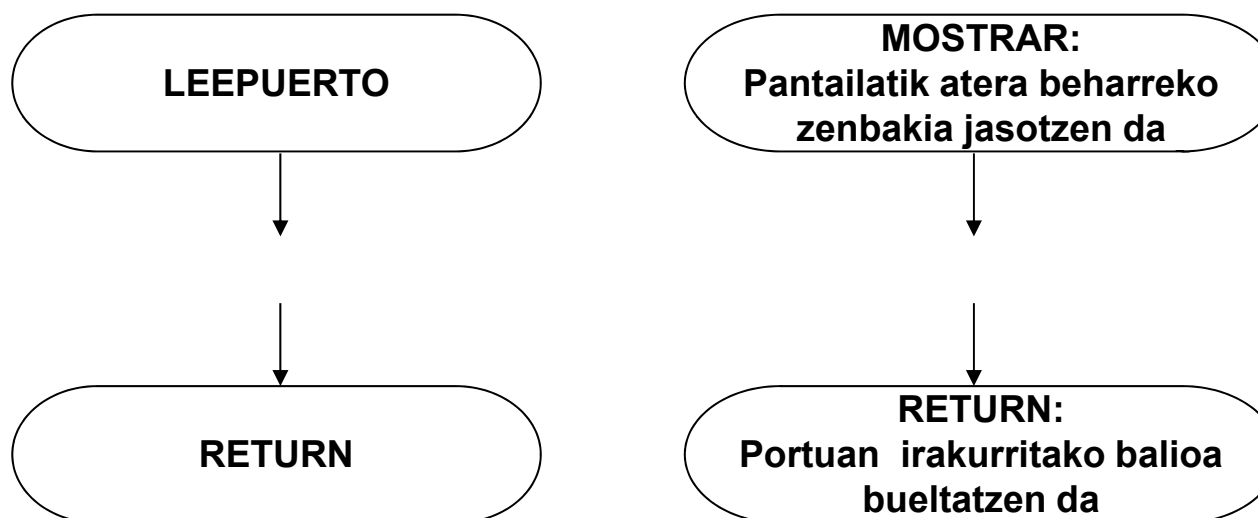
“Akabatzaileak” programa zati (errutina) baten hasiera eta bukaera seinalatzeko erabiltzen dira.

“Hasierako akabatzaileak” errutina baten hasiera seinalatzen dute eta errutinaren izenarekin identifikatzen dira.

“Bukaerako akabatzaileak” errutina baten bukaera puntuak seinalatzen dituzte eta “RETURN” hitzarekin identifikatzen dira.

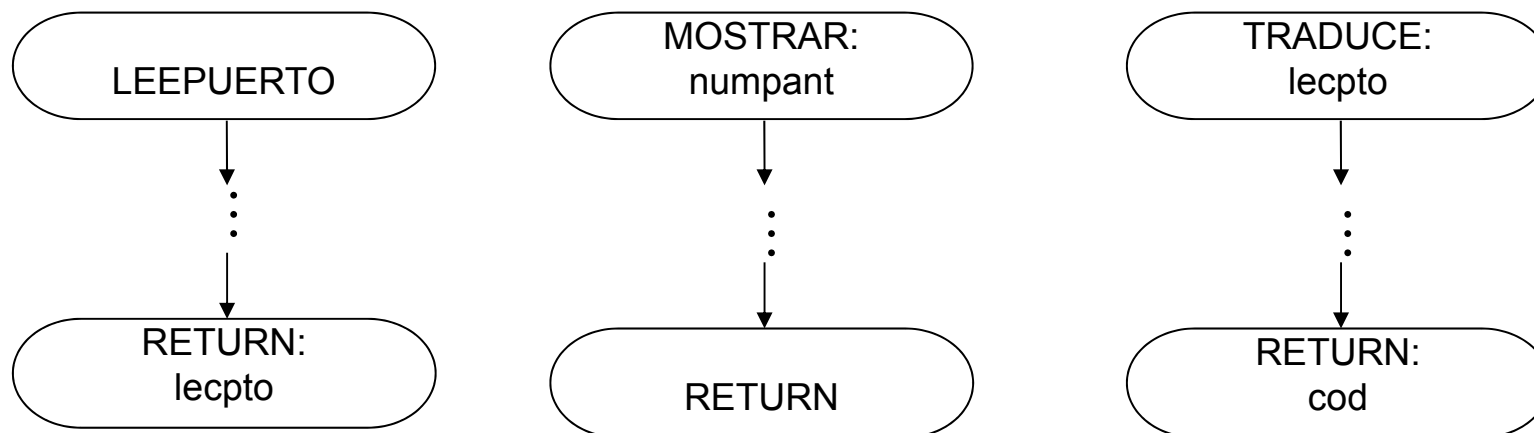
Zenbait kasutan errutinek datuak jasotzea behar izaten dute. Kasu hauetan izenaren atzetik bi puntu eta jaso behar duen datuaren izena idatzi behar dira.

Zenbait kasutan errutinek datuak bueltatzen dituzte. Kasu hauetan “RETURN” hitzaren atzetik bi puntu eta bueltatu behar den datuaren izena idatzi behar dira.



Fluxu diagramak

Adibideak:



Kontuz!

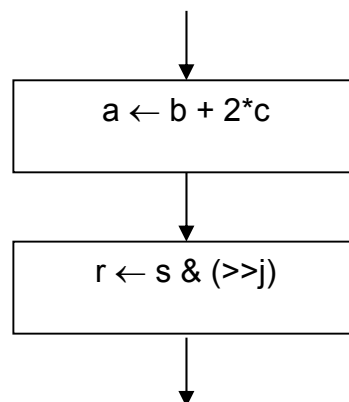
Bueltako agindurako zenbait prozesadorek izen ezberdinak erabiltzen dituzten arren, fluxu diagrametan beti erabiliko dugu “RETURN” hitza.

Fluxu diagramak ez dira prozesadore jakin batendako egiten



Fluxu diagramak

Oinarrizko operazio aritmetikoak edo logikoak (“prozesadoreak egin ohi dituenak”) laukiluze baten idazten dira.



Eragiketak hurrengoak izaten dira:

Funtzioa

Esleipena
 Batuketa
 Kenketa
 Biderketa
 And
 Or
 Exor
 Osagarria (Not)
 Ezk. desplazamendua
 Esk. desplazamendua

Eragilea

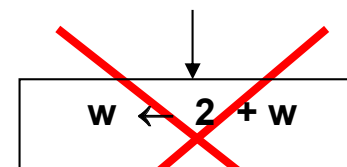
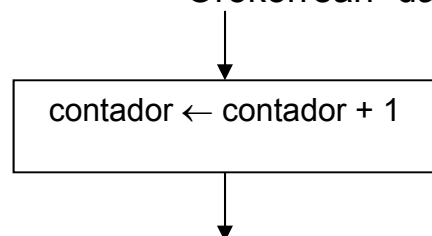
←
 +
 -
 *
 &
 ||
 ⊕
 ~
 <<
 >>

Kontuz!
 Berdintasun ikurra
 (=) ez da erabiltzen



Fluxu diagramak

Fluxu diagrametan ageri diren aldagaiak diagramak hobeto ulertzeko laguntzen duten izen sinbolikoak (aldagaiak) dira. Orokorrean “datuen oroimeneko byte-ak” dira”.



Kontuz!
Inoiz ez dira erregistroen izenak erabili behar.
 Fluxu diagramak ez dira prozesadore jakin batendako egiten

Byte bat ez den zerbaitekin nola lan egin...

Byte baten barneko bit bat - kortxeteak
 Byte *array* baten barneko byte bat - parentesiak

varest[2]
 contadores(3)

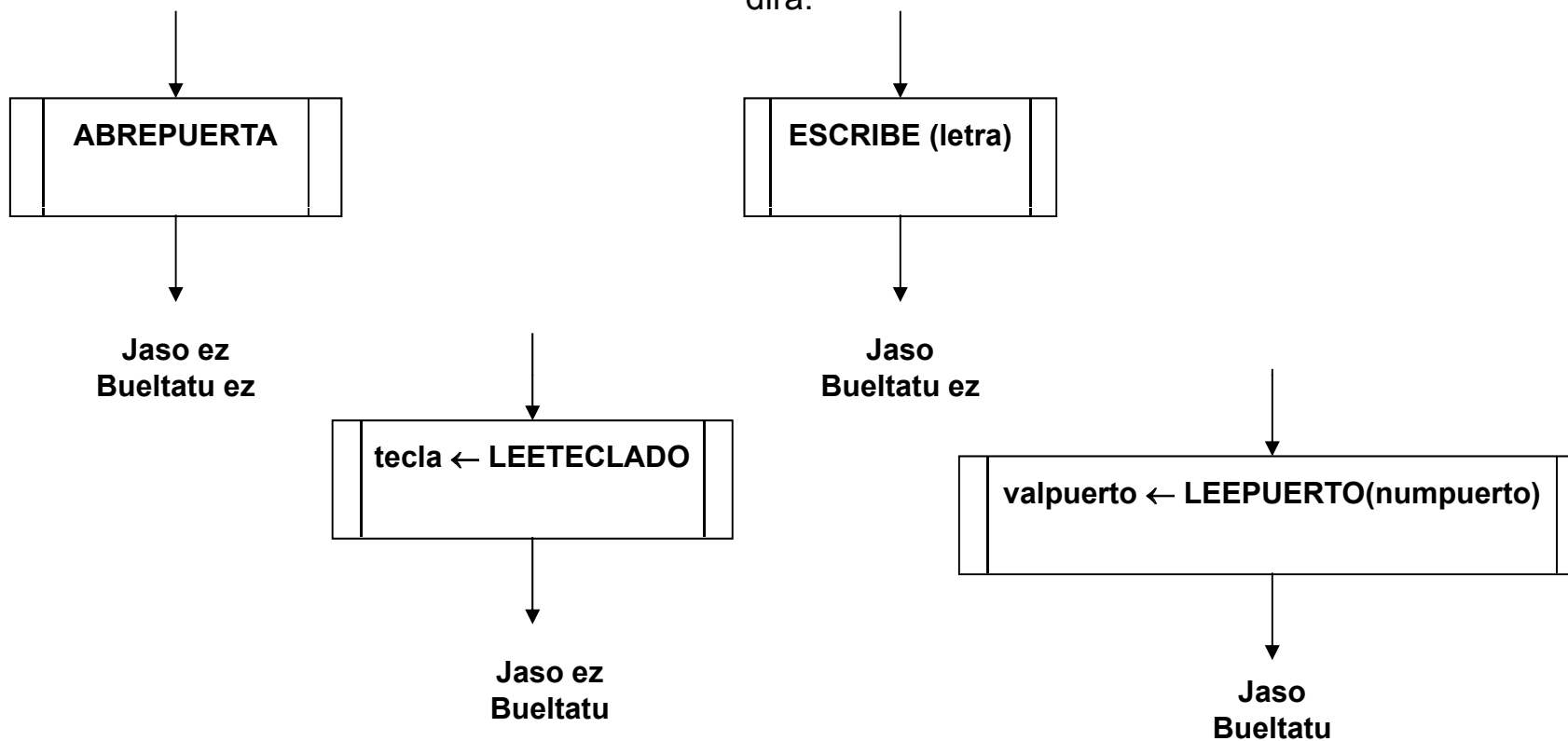


Fluxu diagramak

Batzutan beste programa zatiak (errutinak) erabiltzen dira.
Horretarako deiak erabiltzen dira (CALL, GOSUB).

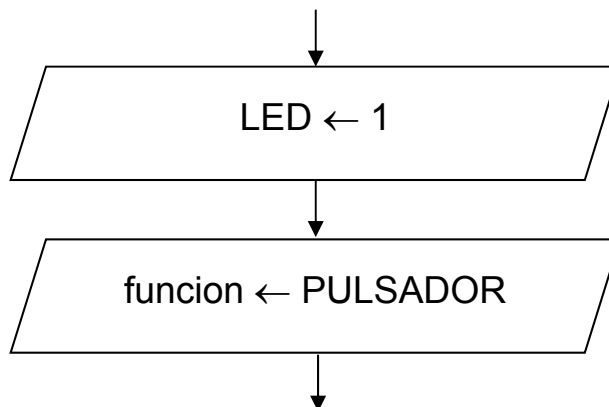
Beste errutinei egiten zaizkien deiak aldeetan barrak dituzten
laukiluze batzuen bitartez adierazten dira.

Daturik jasotzen edo bueltatzen duen arabera 4 tipo ezberdintzen
dira:



Fluxu diagramak

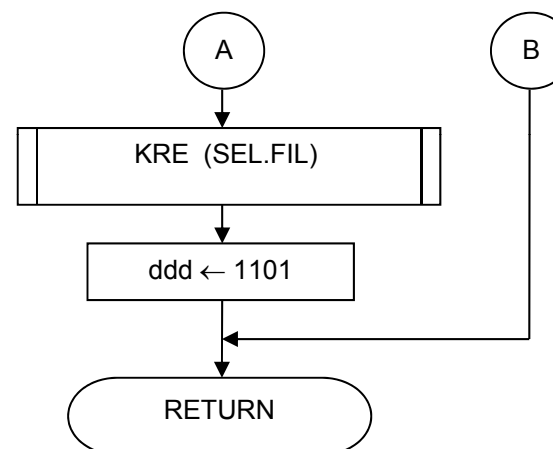
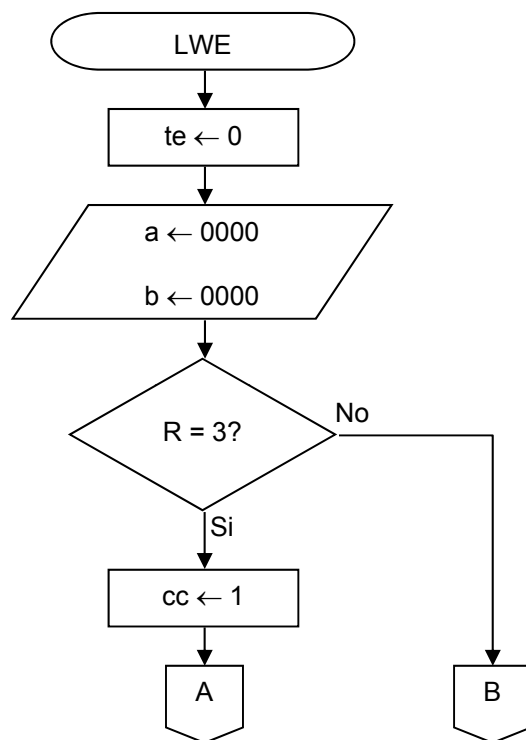
Portuetatik sartzen edo ateratzen diren datuak paralelogramo baten bidez adierazten dira



Fluxu diagramak

Konexio puntuen erabilpena:

“Hurrengo orrian jarraitzen du” eta “aurreko orritik dator” mezuen balokideak dira



Fluxu diagramak

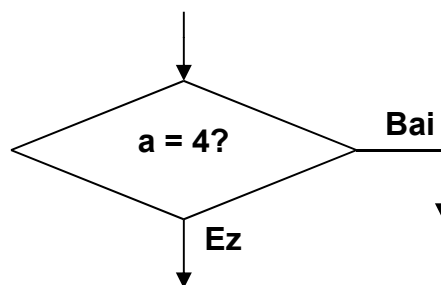
Baldintzapeko bidegurutzea
 Galderek programaren fluxua aldatzea ahalbidetzen dute.
 Ronboaren barnean bai edo ez erantzuntzat duen galdera.
 Konparazio eragileak erabiltzen dira

Funtzioa

Berdintasuna
 Handiagoa
 Txikiagoa
 Handiagoa edo berdina
 Txikiagoa edo berdina

Eragilea

=
 >
 <
 ≥
 ≤

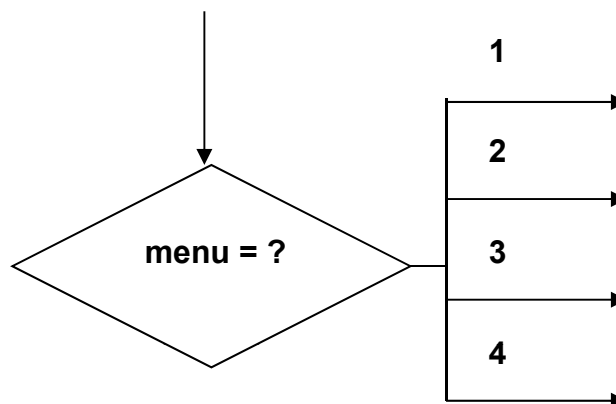


Hemen berdintasun
 ikurra (=) erabili
 daiteke



Fluxu diagramak

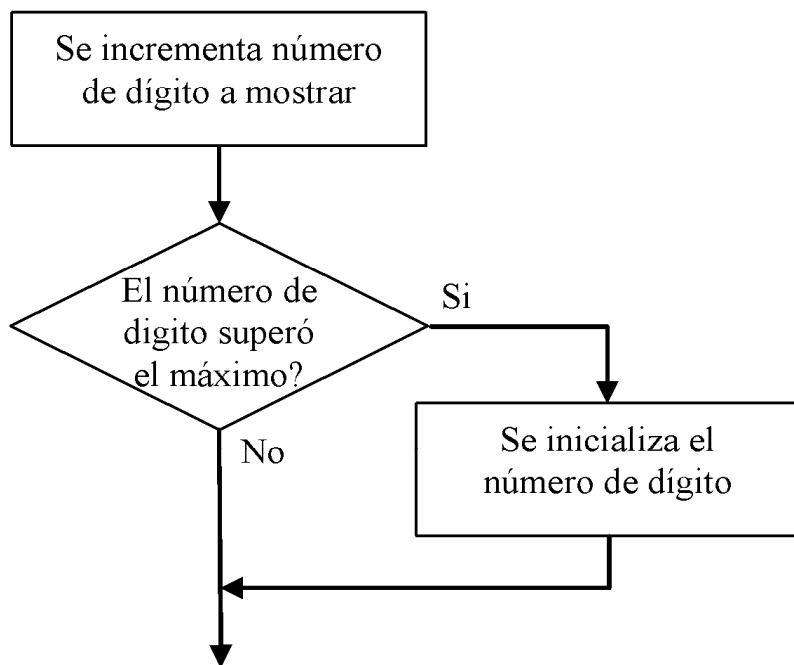
Baldintzapeko bidegurutz anizkuna
Programaren fluxuak aldagai baten balioaren arabera adarra jarraitzen du.



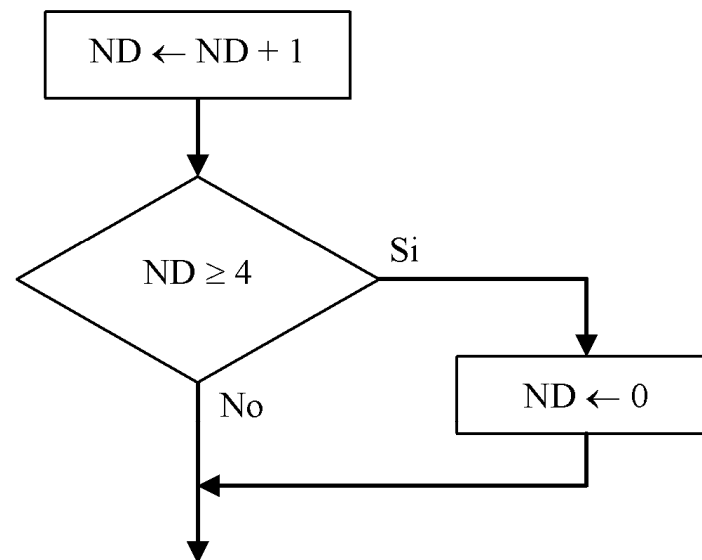
Fluxu diagramak

Fluxu diagrama bat “idazteko” bi era daude:

“Hitzezko” eran



Era “algoritmikoa”



Biak dira baliagarriak. Gehienetan era algoritmikoa erabiltzen bada ere, batzutan hitzezko era erabiltzea komeni da (bereziki prozesadorearen motaren arabera zerbait deskribatu behar den kasuetan)

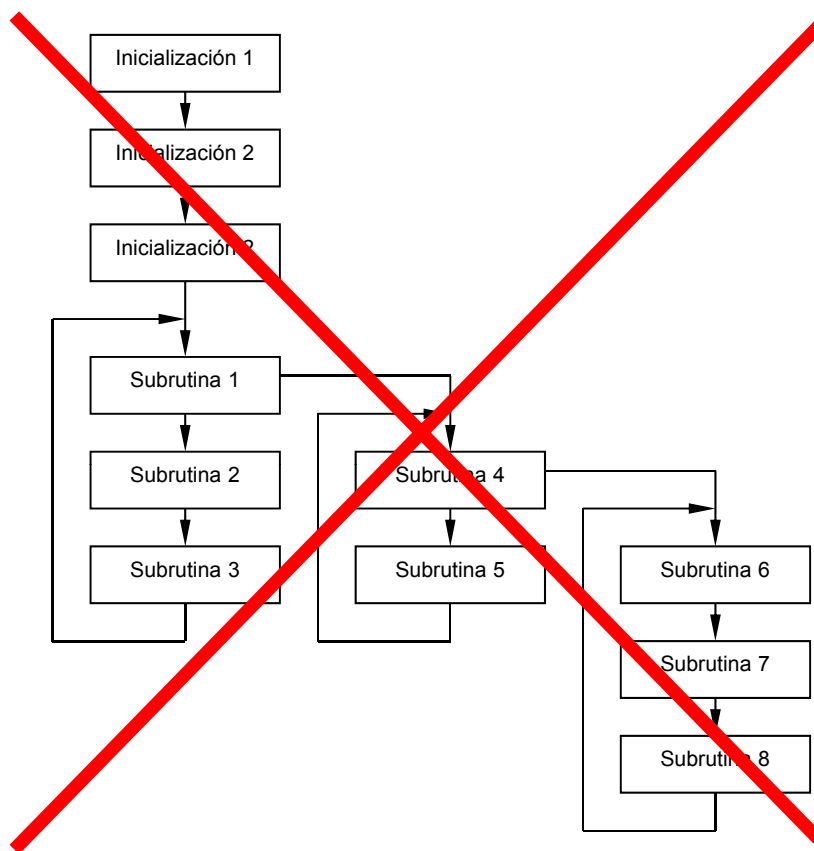
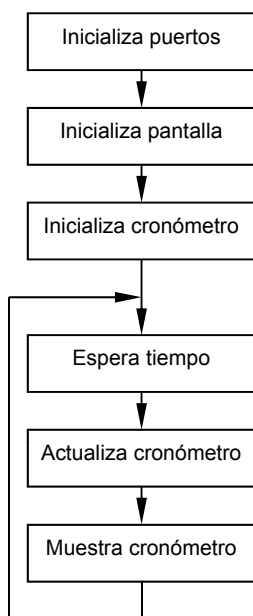


Fluxu diagramak. 1. adibidea

Programa nagusia

Programa begizta nagusian bueltaka egon behar da beti.

Ez dira INOIZ bigarren mailako begiztarik erabili behar.

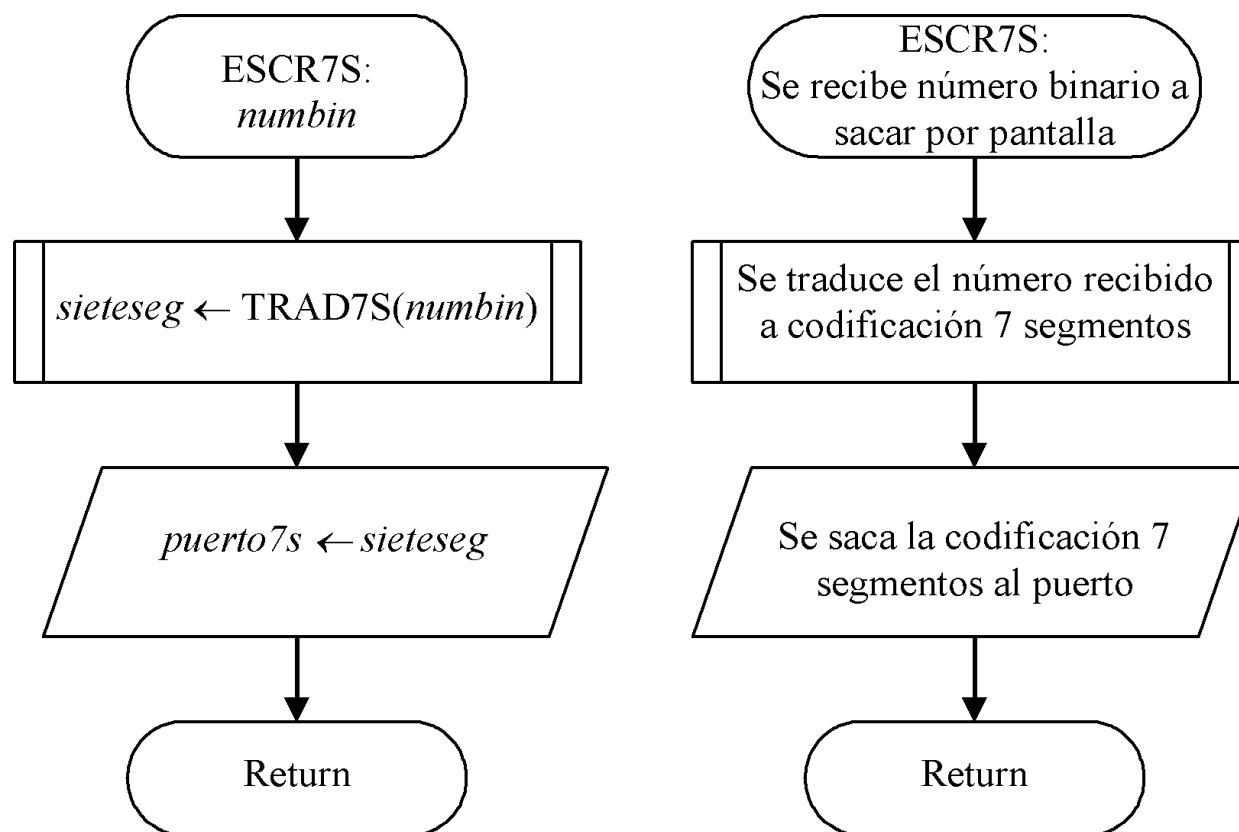
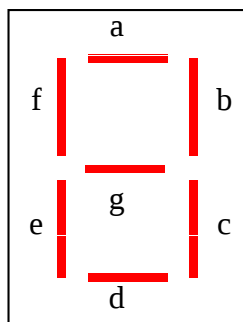


Fluxu diagramak. 2. adibidea



7 segmentuko pantaila baten idazketa

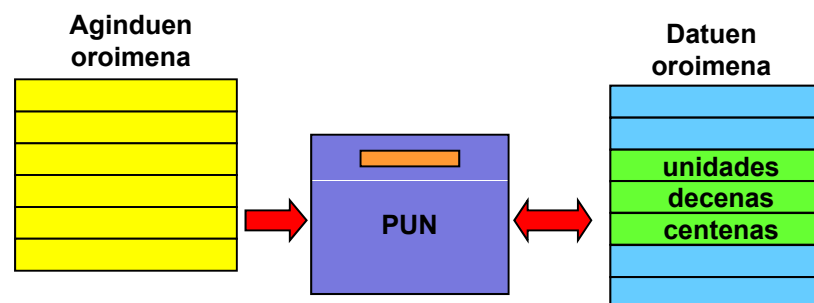
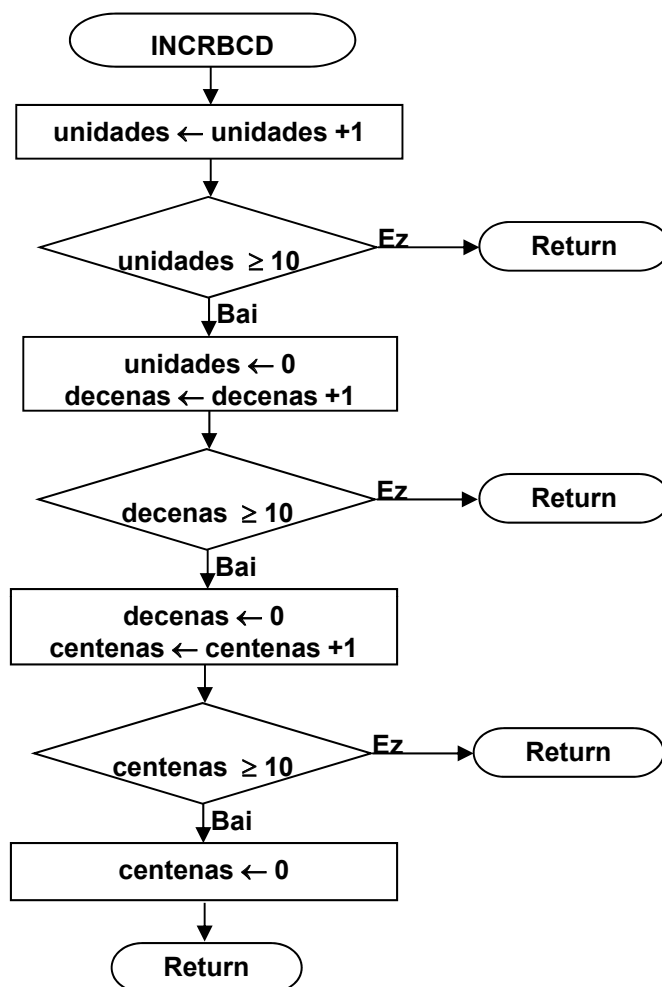
Diagrama idazteko bi modu: algoritmikoa
eta hitzezkoa



Fluxu diagramak. 5. adibidea



BCD zenbaki bat unitate baten gehitu



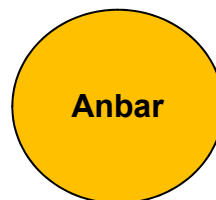
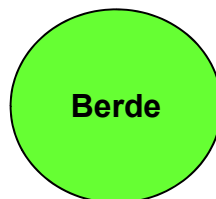
Egoera makinak

Egoera:

Sistema jakin bat egon ahal daitekeen kondizio bat (balio egonkorra).

Adibidez: semaforo bat gorri, anbar edo berde egon daiteke.

Egoerak “pilotak” bezala adierazten ditugu.



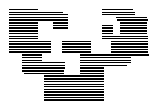
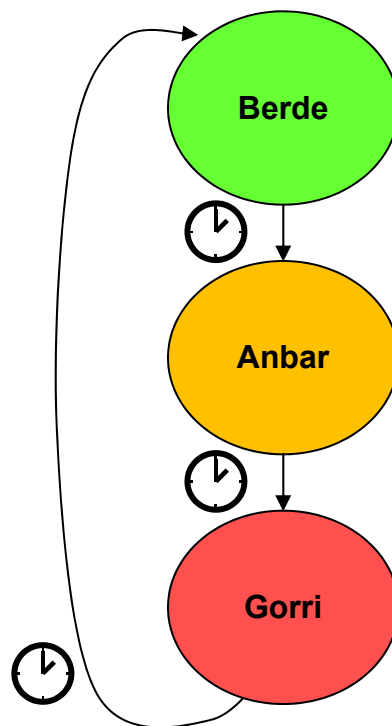
Egoera makinak

Gertaera:

Sisteman gertatzen den zerbait. Batzuetan egoera aldaketa eragiten du.

Adibidez: semaforo baten gertaera denbora da.

Gertaerak “gezien” bitartez adierazten dira.



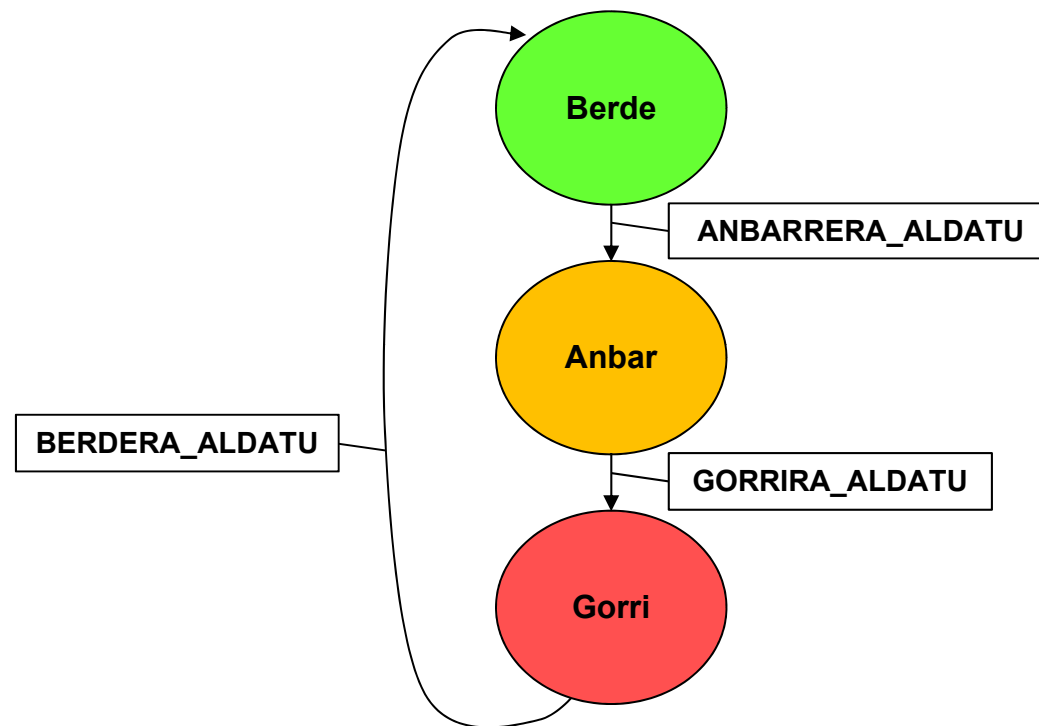
Egoera makinak

Ekintza:

Gertaera bat ematen denean egin behar den jarduera.

Adibidez: semaforo baten ekintzak bonbillen piztea eta amatatzea dira.

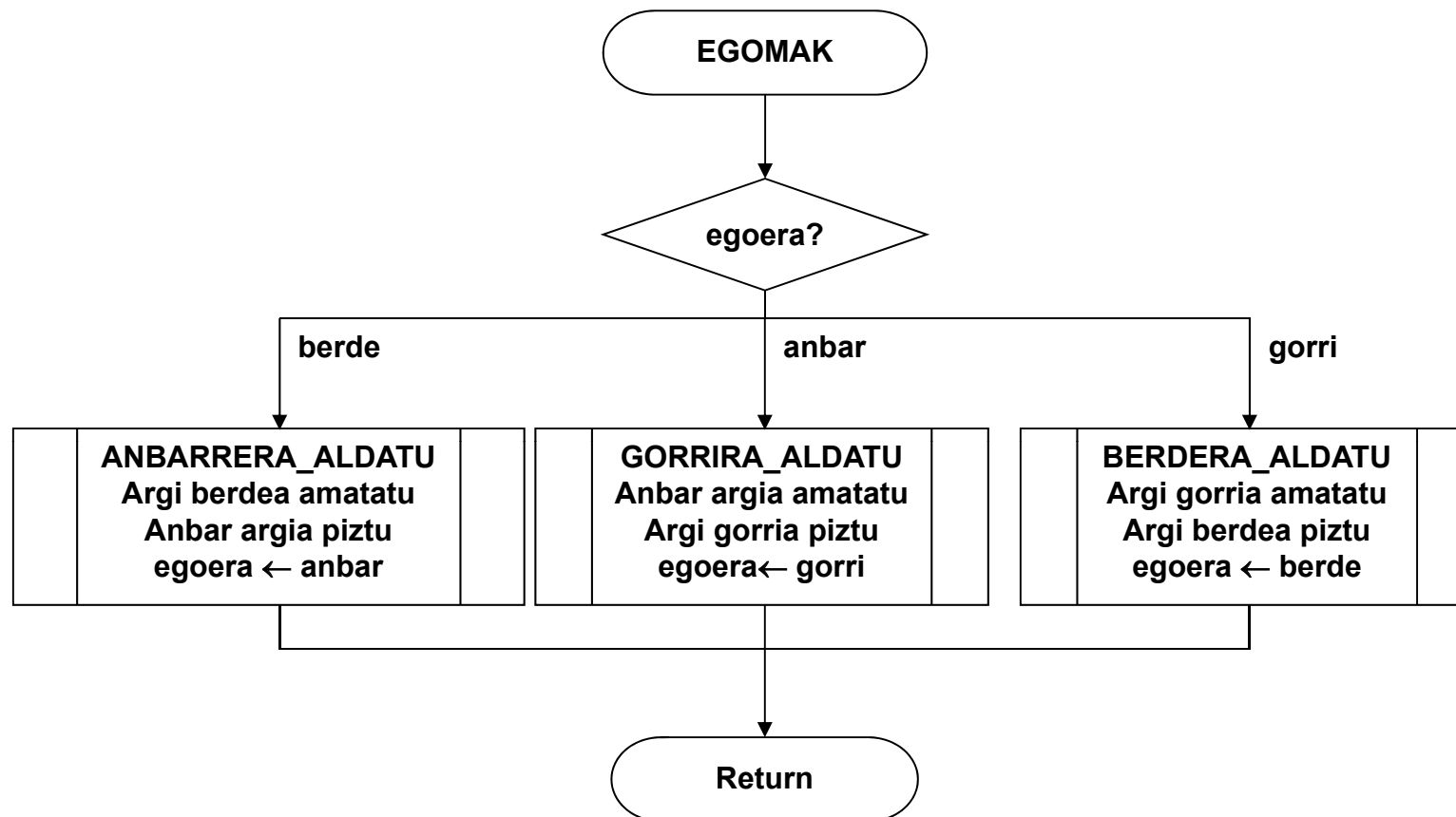
Ekintzak “laukizuzen” batzuen bitartez adierazten dira.”.



Egoera makinak

Egoera makina baten implementazioa software bitartez:

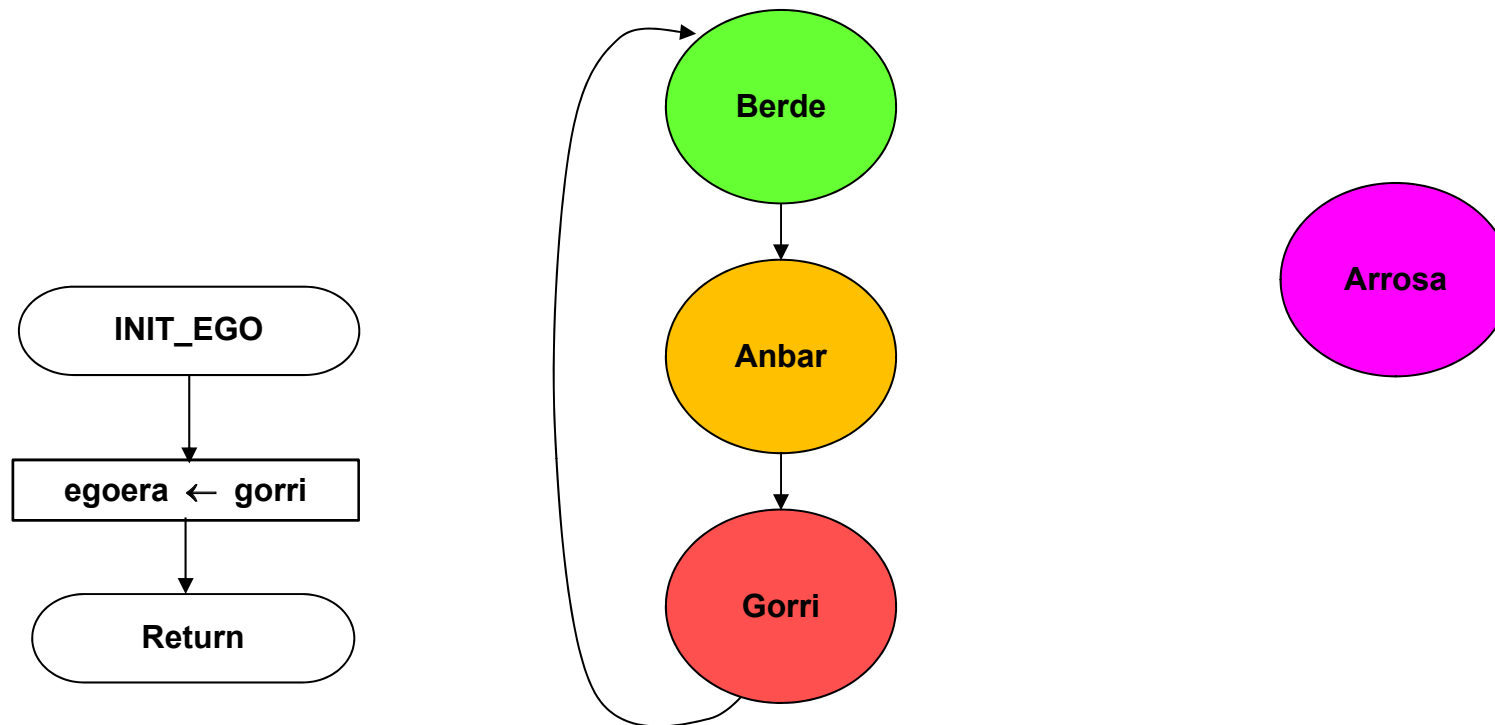
Software baten egoera bat balio numeriko abstraktu bat da
(Adibidez: berde=0, anbar=1, gorri=2).



Egoera makinak

Egoera makinak hasieratu egin behar dira!

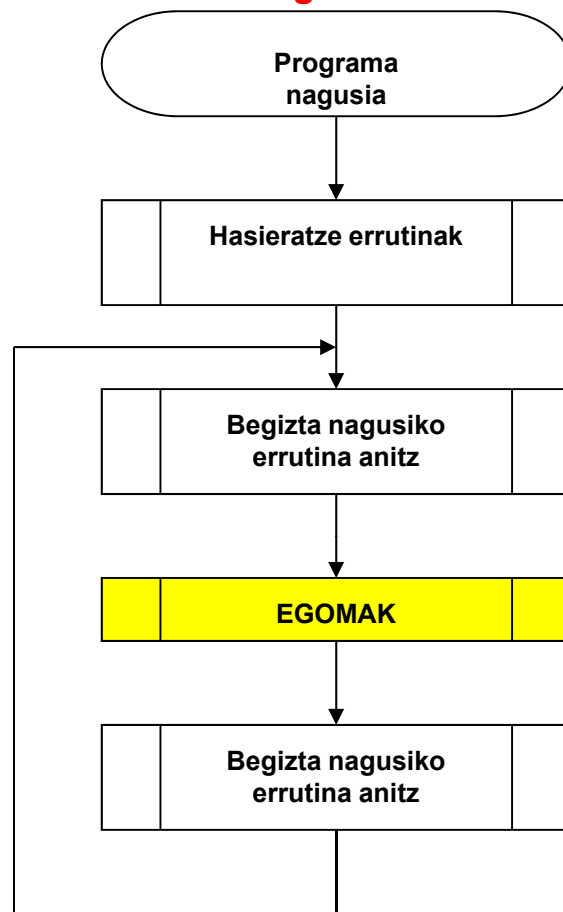
Egoera ezezagunak ekidin behar dira



Egoera makinak

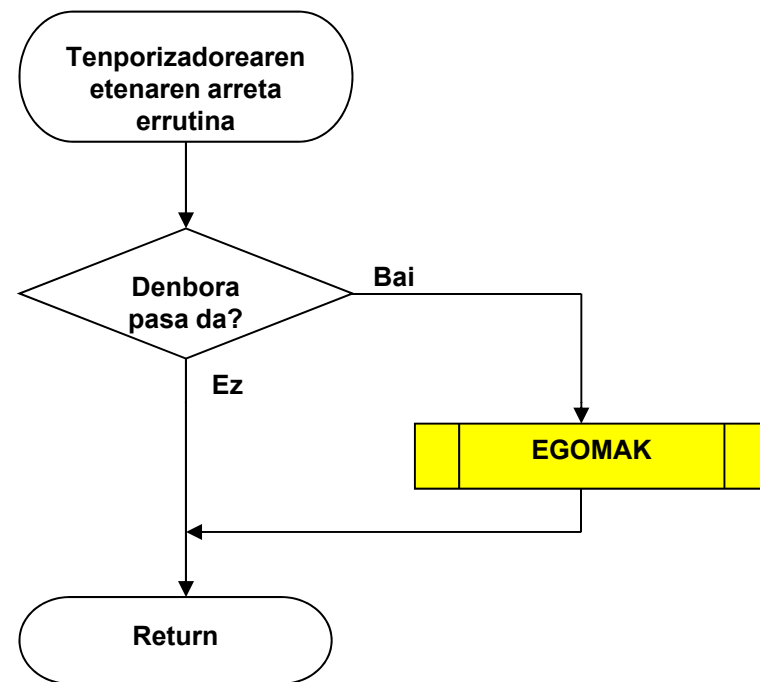
Non kokatzen dira egoera makinak:

Orokorrean programa nagusia



Batzutan denbora etenetan...

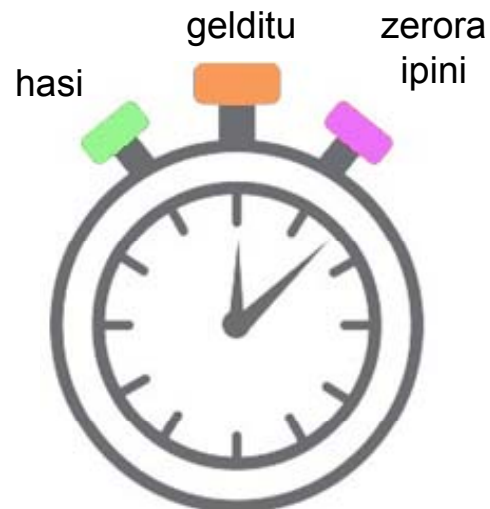
Aurrerago ikusiko ditugu...



Gertaera makinak

Semaforoaren kasuan egoera makinari deitzen dion gertaera bakar bat dago (denbora). Sistema gehienetan mota ezberdinetako gertaerak ditugu

Adibidez: kronometro baten 3 teklak (hasi, gelditu eta zerora ipini)

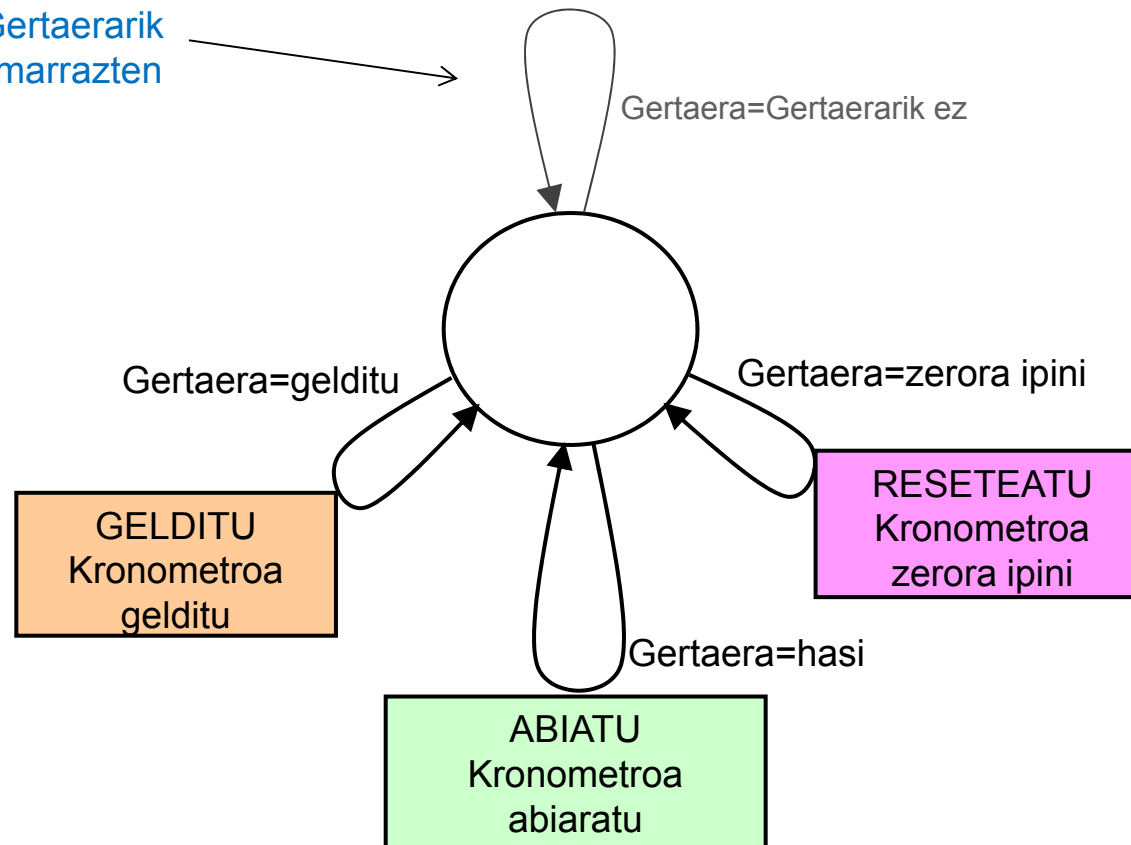


Gertaera makinak

Kasu hauetan gertaera makina bat daukagu.

Gertaera une baten gertatzen den zerbait da, ez da egoera bezala mantentzen den zerbait.

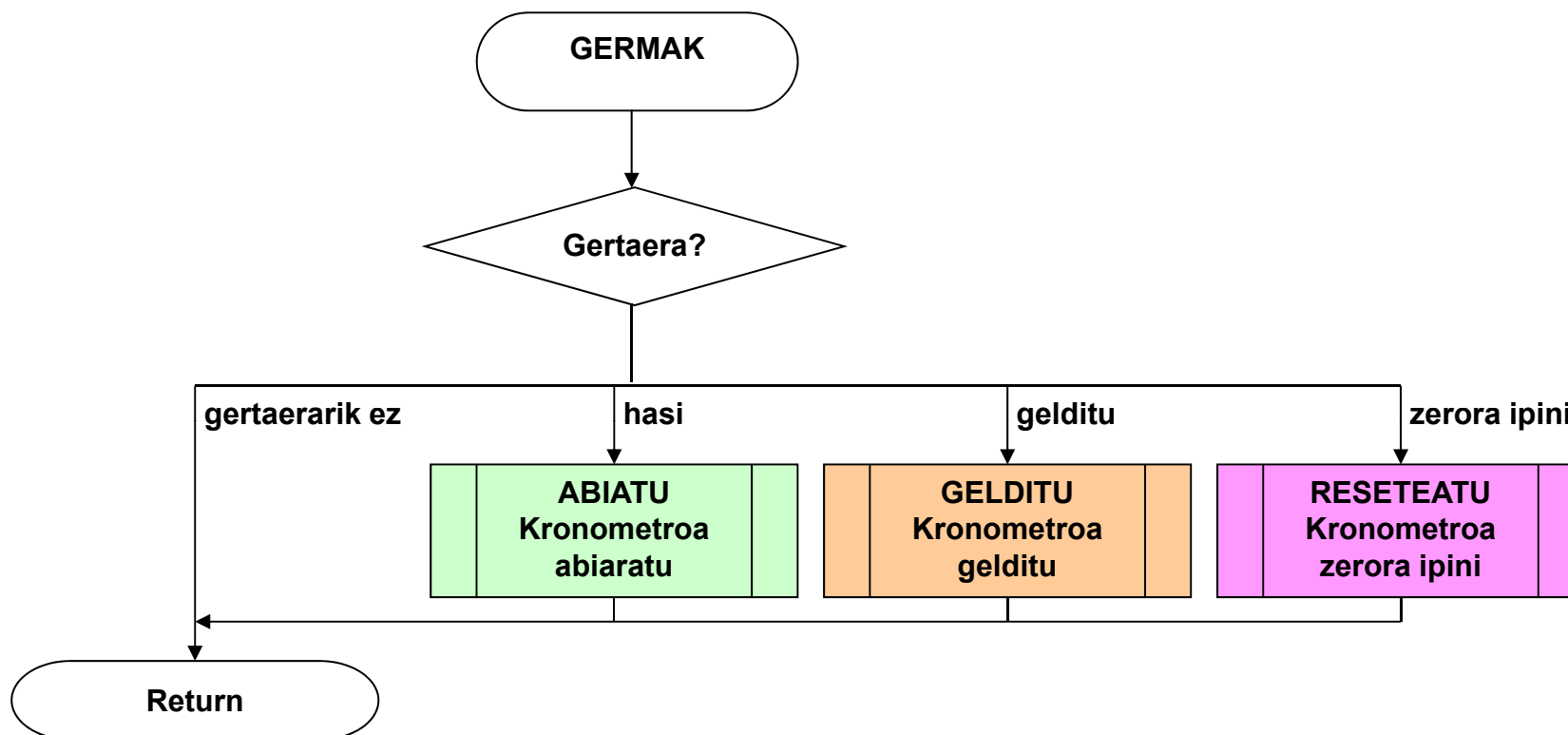
Oharra: "Gertaerarik ez" ez da marrazten



Gertaera makinak

Gertaera makina baten inplementazioa software bitartez :

Software baten egoera bat balio numeriko abstraktu bat da
(Adibidez: Geraterarik ez=0, hasi=1, gelditu=2, zerora ipini=3).

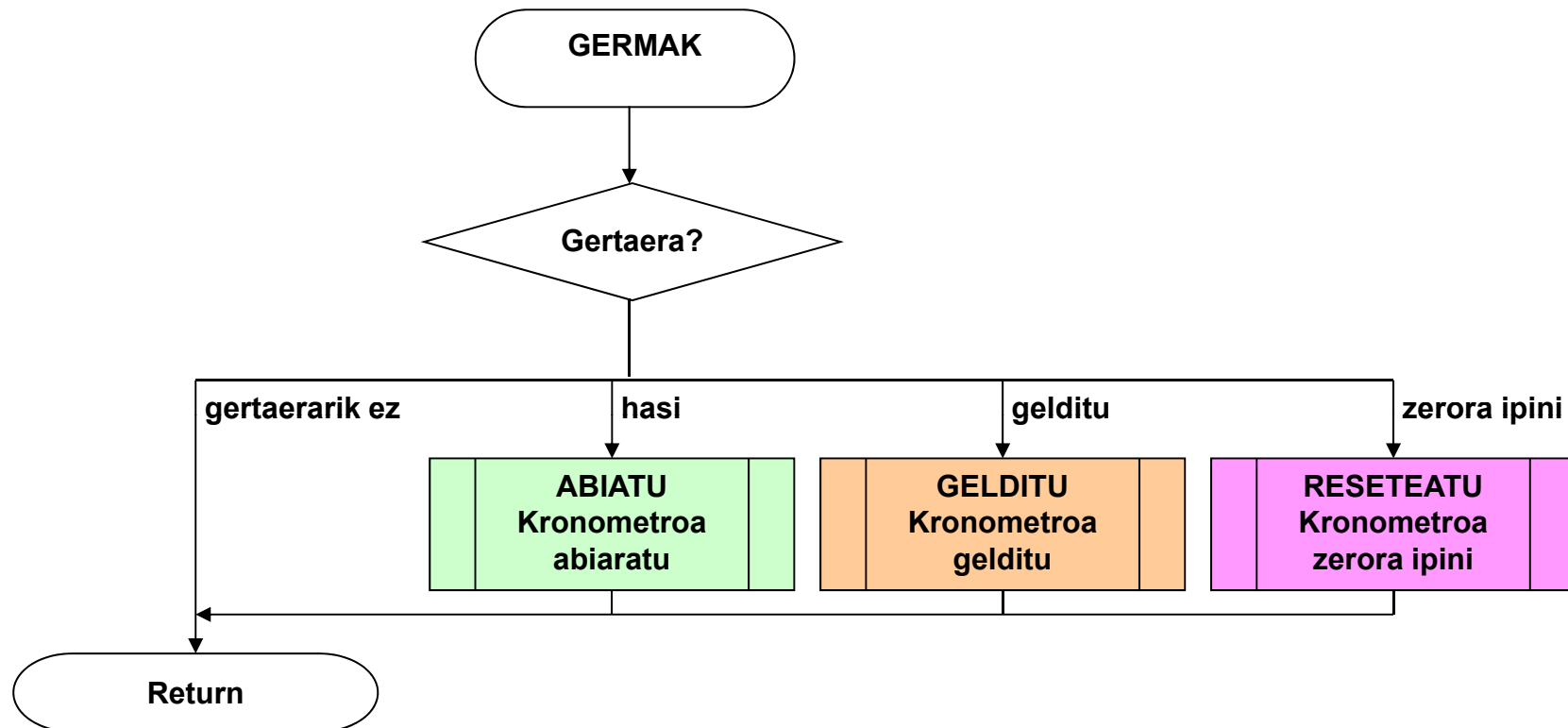


Gertaera makinak

Gertaera makina baten eta egoera makina baten arteko funtzezko aldaketak honakoak dira:

Gertaera=0 gertaerarik ez dagoela esatea nahi du, hortaz adar horrek inoiz ez du ekintzarik.

Gertaera makina hasieratzea ez da beharrezkoa



Egoera - gertaera makinak



Demagun bi operazio “modu” (egoerak)
dituen erloju bat dugula:

Ordua bistaratzen duen modua

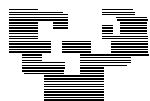
Ordua doitzeko modua

Erlojuak gertaerak sortzen dituen hiru
tekla ditu:

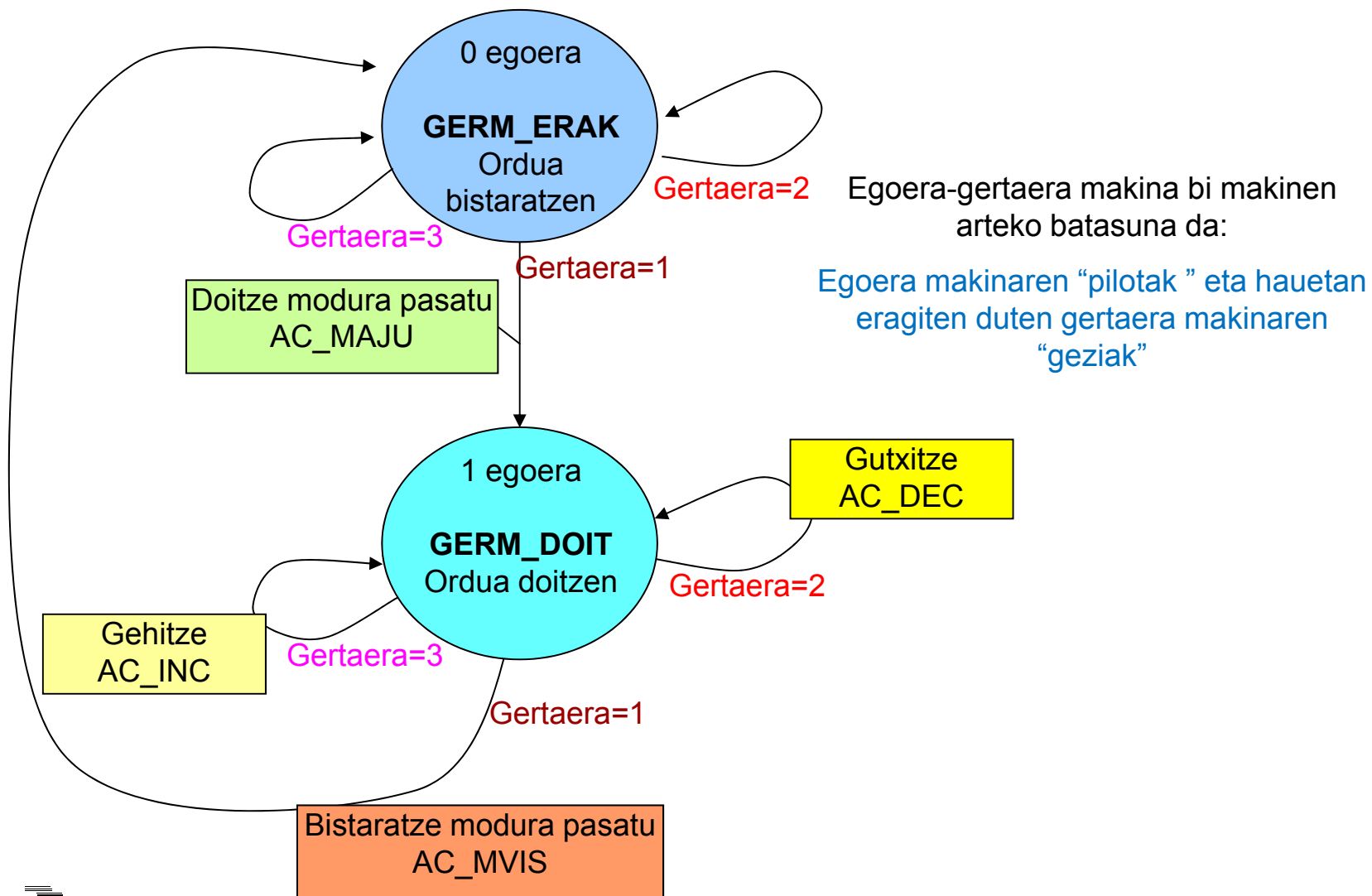
“MODO” tekla, gertaera 1

“<” tekla, gertaera 2

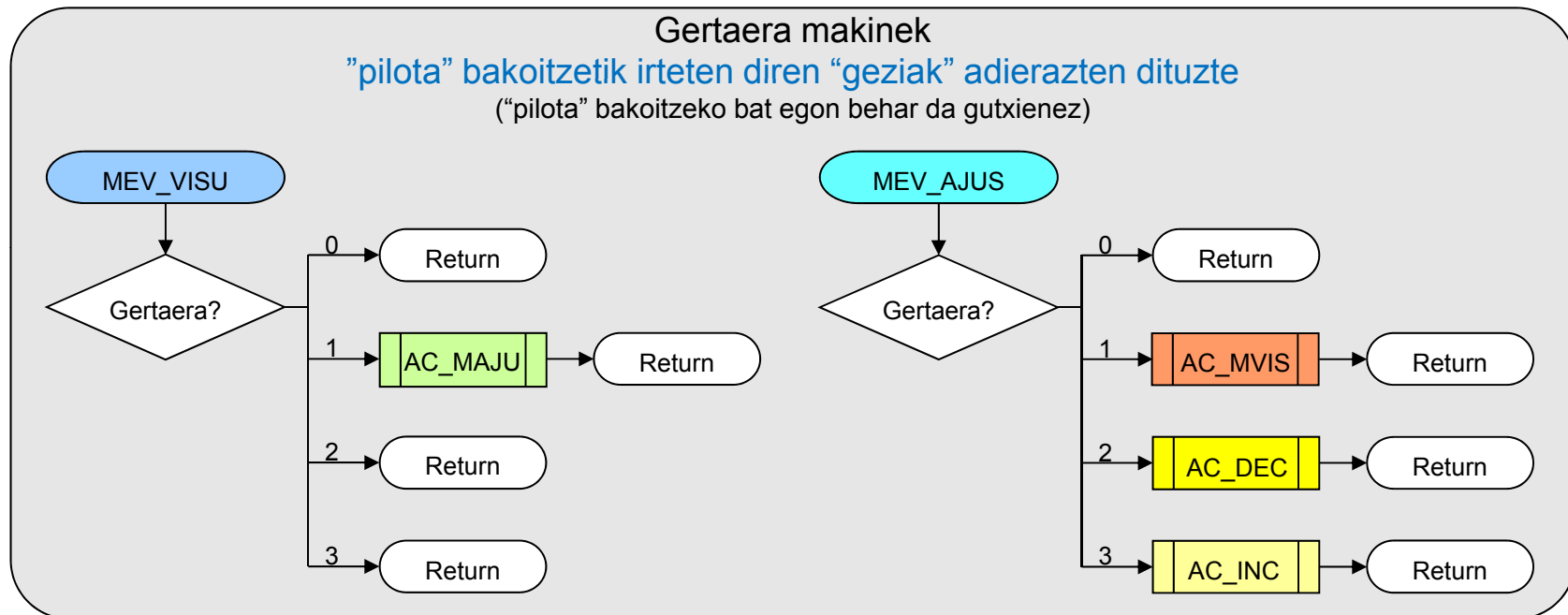
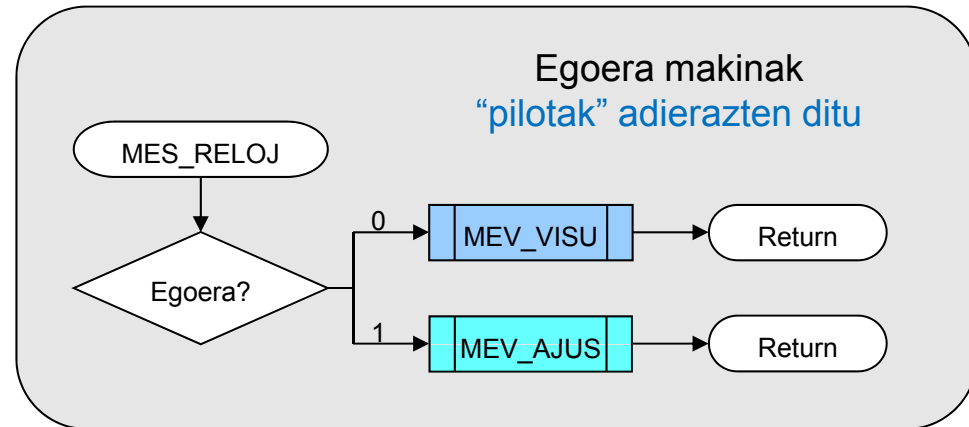
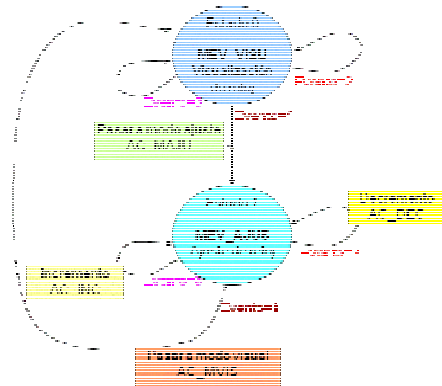
“>” tekla, gertaera 3



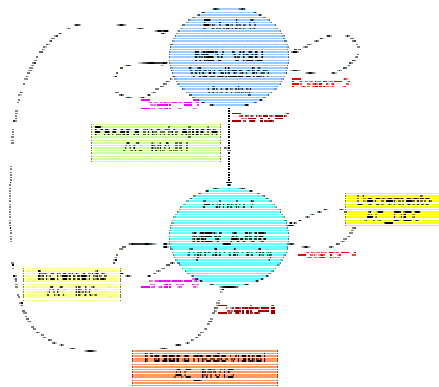
Egoera - gertaera makinak



Egoera - gertaera makinak



Egoera - gertaera makinak



Egoera aldaketa dakarten gertaera guztiek ekintza bat izan behar dute.

Egoera aldatzen ez duten gertaerek ez dute zertan ekintzik izan behar.

