

Práctica CICS LAB 1

1) ¿Cómo nos conectamos al CICS?

Hay que escribir en la pantalla inicial del terminal 3270 del z/OS la siguiente instrucción: L CICSTS61.

```
z/OS V2R5 LVLI PUT2303/RSU2303 IP Address = 10.1.3.8
VTAM Terminal = TCP00012

Application Developer System

      // 0000000 SSSS
      // 00 00 SS
zzzzzz // 00 00 SS
      zz // 00 00 SSSS
      zz // 00 00 SS
      zz // 00 00 SS
zzzzzz // 0000000 SSSS

System Customization - ADCD.Z25D.*

===> Enter "LOGON" followed by the TSO userid. Example "LOGON IBMUSER" or
===> Enter L followed by the APPLID
===> Examples: "L TSO", "L CICSTS56", "L CICSTS61", "L IMS15"

L CICSTS61
```

2) Una vez entrado, ¿qué tenemos que rellenar a continuación?

A continuación entramos en el signon de CICS y ponemos nuestro usuario de TSO + la contraseña. Mi usuario es TSOPSS1.

```
Signon to CICS APPLID CICSTS61

WELCOME TO CICS TS 6.1

Type your userid and password, then press ENTER:

  Userid . . . . tsopss1      Groupid . . . . _____
  Password . . . . _
  Language . . . . _
  New Password . . . .
```

La pantalla que aparece al meter el usuario, la contraseña y darle enter es:



DFHCE3549 Sign-on is complete (Language ENU).

Escribir la transacción CEMT y empezar a jugar con esa en concreto.

cemt i => es una transacción maestra de administración de CICS para consultar o modificar recursos, como archivos o programas.

3) Una vez realizado el sign on, identificar qué transacción se tiene que utilizar para saber a qué terminal se está conectado.

cemt i terminal => es una transacción para listar todos los terminales conocidos por el sistema.

```
CEMT I TERMINAL
STATUS: RESULTS - OVERTYPE TO MODIFY
Ter(CBRF) Pri( 000 ) Pag Ins Ati Tti Rte
Net(CBRF ) Rem(CBR )
Ter(CERR) Pri( 000 ) Pag Ins Tti Loc
Con(*ERRCON* )
Ter(0007) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00007) Acq Tas(0000256) Nqn(NETD.TCP00007 )
Ter(0011) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00011) Acq Nqn(NETD.TCP00011 )
Ter(0012) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00012) Acq Tas(0000189) Nqn(NETD.TCP00012 )
Ter(0013) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00013) Acq Nqn(NETD.TCP00013 )
Ter(0014) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00014) Acq Tas(0000265) Nqn(NETD.TCP00014 )
Ter(0015) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00015) Acq Tas(0000261) Nqn(NETD.TCP00015 )
Ter(0016) Tra(CECI) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00016) Acq Tas(0000247) Nqn(NETD.TCP00016 )

SYSID=S740 APPLID=CICSTS61
RESPONSE: NORMAL TIME: 05.14.59 DATE: 04/16/25
```

Éste es mi terminal:

cemt i terminal userid(tsopss1)

```
CEMT I TERMINAL USERID(TSOPSS1)
STATUS: RESULTS - OVERTYPE TO MODIFY
Ter(0012) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00012) Acq Tas(0000189) Nqn(NETD.TCP00012 )
```

Si me pongo encima y doy enter, puedo ver lo que contiene:

```
Terminal(0012)
Transaction(CEMT)
Priority( 000 )
Pagestatus( Pageable )
Servstatus( Inservice )
Atistatus( Ati )
Ttistatus( Tti )
Nature(Local)
Purgetype( )
Netname(TCP00012)
Console()
Termstatus( Acquired )
Createsess( Ncreate )
Task(0000189)
Remotesystem()
Nqname(NETD.TCP00012)
Rname()
Rnetname()
```

4) Utilizar la transacción adecuada y las opciones necesarias para mostrar los ficheros utilizados por el CICS. ¿Cuál es su DSN?

cemt i file => es una transacción para mostrar todos los ficheros definidos y utilizados por el sistema CICS.

```

Fil(DFHCSD ) Vsa Ope Ena Rea Upd Add Bro Del Sha
      Dsn( DFH610.CICS.DFHCSD )
Fil(DFHDBFK ) Vsa Clo Ena Rea Upd Add Bro Del Sha
Fil(DFHLRQ ) Vsa Ope Ena Rea Upd Add Bro Del Sha
      Dsn( DFH610.CICS.DFHLRQ )
Fil(EQADPFMB) Vsa Ope Ena Rea Upd Add Bro Del Sha
      Dsn( DFH610.CICS.EQADPFMB )
Fil(EZACACHE) Vsa Clo Une Rea Upd Add Bro Del Old
      Dsn( DFH610.USER.CACHE ) Use
Fil(EZACONFG) Vsa Clo Dis Rea Bro Sha
      Dsn( DFH610.TCP.CONFIG )

```

Campos importantes:

- Dentro de fil() tenemos el nombre del fichero.
- A continuación de VSA tenemos el estado del fichero.
- Luego viene la ubicación: remota o local.
- Y por último el tipo de acceso: secuencial o indexado.

```

CEMT I FILE(EZACACHE)
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(EZACACHE) Vsa Clo Une Rea Upd Add Bro Del Old
      Dsn( DFH610.USER.CACHE ) Use

```

El fichero es el EZACACHE, su estado es cerrado e inhabilitado, su ubicación es local y su tipo de acceso es secuencial. Su DSN es DFH610.USER.CACHE

Si le pongo mi userid, me sale esto:

```

CEMT I FILE USERID(TSOPSS1)
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(DFHCSD ) Vsa Ope Ena Rea Upd Add Bro Del Sha
      Dsn( DFH610.CICS.DFHCSD )
Fil(DFHDBFK ) Vsa Clo Ena Rea Upd Add Bro Del Sha
Fil(DFHLRQ ) Vsa Ope Ena Rea Upd Add Bro Del Sha
      Dsn( DFH610.CICS.DFHLRQ )
Fil(EQADPFMB) Vsa Ope Ena Rea Upd Add Bro Del Sha
      Dsn( DFH610.CICS.EQADPFMB )
Fil(EZACACHE) Vsa Clo Une Rea Upd Add Bro Del Old
      Dsn( DFH610.USER.CACHE ) Use
Fil(EZACONFG) Vsa Clo Dis Rea Bro Sha
      Dsn( DFH610.TCP.CONFIG )
Fil(FILEA ) Vsa Ope Ena Rea Upd Add Bro Del Sha
      Dsn( DFH610.CICS.FILEA )

```

- Utilizar la transacción adecuada y las opciones necesarias para modificar las opciones y mostrar sólo los ficheros que empiecen por F. ¿Cuál es su DSN?

cemt i file(f*) => es una transacción para mostrar los ficheros que empiecen por F y están definidos y utilizados por el sistema CICS.

```
CEMT I FILE(F*)
STATUS:  RESULTS - OVERTYPE TO MODIFY
  Fil(FILEA   ) Vsa Ope Ena Rea Upd Add Bro Del      Sha
          Dsn( DFH610.CICS.FILEA                      )
```

Hay 1 fichero que empieza por F, llamado Filea, cuyo estado es abierto y habilitado, la ubicación es local y su tipo de acceso es secuencial. Su DSN es DFH610.CICS.FILEA

- Utilizar la transacción adecuada y las opciones necesarias para mostrar todas las transacciones que empiezan por A.

cemt i tra(A*)

```
I TRA(A*)
STATUS:  RESULTS - OVERTYPE TO MODIFY
Tra(AADD) Pri( 001 ) Pro(DFH$AALL) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
Tra(ABRW) Pri( 001 ) Pro(DFH$ABRW) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
Tra(AINQ) Pri( 001 ) Pro(DFH$AALL) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
Tra(AMNU) Pri( 001 ) Pro(DFH$AMNU) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
Tra(AORD) Pri( 001 ) Pro(DFH$AREN) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
Tra(AORQ) Pri( 001 ) Pro(DFH$ACOM) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
Tra(AREP) Pri( 001 ) Pro(DFH$AREP) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
Tra(AUPD) Pri( 001 ) Pro(DFH$AALL) Tc1( DFHTCL00 ) Ena
  Sta      Pur Prf(DFHCICST) Uda Any Iso          Bac Wai
```

- Utilizar la transacción adecuada y las opciones necesarias para mostrar todos los terminales conectados.

cemt i ter acq

```

I TER ACQ
STATUS: RESULTS - OVERTYPE TO MODIFY
Ter(0002) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00002) Acq Nqn(NETD.TCP00002 )
Ter(0004) Tra(CEDA) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00004) Acq Tas(0001675) Nqn(NETD.TCP00004 )
Ter(0006) Tra(CEDA) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00006) Acq Tas(0001417) Nqn(NETD.TCP00006 )
Ter(0007) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00007) Acq Nqn(NETD.TCP00007 )
Ter(0012) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00012) Acq Tas(0001311) Nqn(NETD.TCP00012 )
Ter(0013) Tra(CECI) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00013) Acq Tas(0001468) Nqn(NETD.TCP00013 )
Ter(0014) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00014) Acq Tas(0001782) Nqn(NETD.TCP00014 )
Ter(0015) Tra(CEDA) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00015) Acq Tas(0001783) Nqn(NETD.TCP00015 )
+ Ter(0016) Tra(CEDA) Pri( 000 ) Pag Ins Ati Tti Loc
Net(TCP00016) Acq Tas(0001779) Nqn(NETD.TCP00016 )

```

- Utilizar la transacción adecuada y las opciones necesarias para mostrar las TDs que empiecen por E.

cemt i td(e*)

```

I TD(E*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Tdq(EQAD) Ext Ena Clo
Shr Inp Dat(001) Ddn(EQADB GDS)
Tdq(EQAL) Ind Nam(CSSL)
Tdq(EQAM) Ext Ena Clo
Shr Inp Dat(001) Ddn(EQADBGM )

```

- Utilizar la transacción adecuada y las opciones necesarias para mostrar las opciones con las que está ejecutando el CICS. ¿Está activada la opción de Autoinstall?

cemt i sys

```

I SYS _
STATUS: RESULTS - OVERTYPE TO MODIFY
System
Aging( 00500 )
Aidcount(00000)
Akp( 04000 )
Cicstslevel(060100)
Cmdprotect(Cmdprot)
Db2conn()
Debugtool( Nodebug )
Dfltuser(CICSUSER)
Dsalimit( 07340032 )
Dsrtprogram( NONE )
Dtrprogram( DFHDYP )
Dumping( Sysdump )
Edsalimit( 0838860800 )
Forceqr( Noforce )
Healthcheck(Check)
Lastcoldtime()
+ Lastemertime()

```

cemt i auto

```

I AUTO _
STATUS: RESULTS - OVERTYPE TO MODIFY
Aut Max( 100 ) Progr( DFHZATDX ) Cur(000) Ful Ena Aut

```

Sí, está activada, ya que pone Ena.

5) Utilizar la transacción adecuada y las opciones necesarias para:

5.1) Mostrar por pantalla el mapa DFH\$AGA, sabiendo que el comando a utilizar es SEND MAP (DFH\$AGA) MAPONLY.

Para ello, hay que entrar en la transacción de “ceci” y ahí ejecutar la instrucción de: send map (dfh\$aga) maponly

```

SEND MAP (DFH$AGA) MAPONLY
STATUS: ABOUT TO EXECUTE COMMAND NAME=
EXEC CICS SEND Map( 'DFH$AGA' )
  << FRom() > < LEngth() > < DAtaonly > | MAPOnly >
  < MAPSet() >
  < FMhparm() >
  < Reqid() >
  < LDc() | < ACTpartn() > < Outpartn() > >
  < MSr() >
  < Cursor() >
  < Set() < MAPPINGdev() > | PAging | Terminal < Wait > < LAst > >
  < PRint >
  < FREekb >
  < ALArm >
  < L40 | L64 | L80 | Honeom >
  < NLeom >
  < ERASE < DEfault | ALTernate > | ERASEAup >
  < ACCum >
  < FRSet >
+ < NOflush >

```

5.2) Cargar un registro en la TS PSSX, sabiendo que el comando a utilizar es WRITEQ TS QU (PSSX) FROM ('<mensaje>').

Para ello, hay que entrar en la transacción de "ceci" y ahí ejecutar la instrucción de: writeq ts qu ('pss1') from ('Hola soy PSS1')

```

WRITEQ TS QU ('PSS1') FROM ('HOLA SOY PSS1')
STATUS: COMMAND EXECUTION COMPLETE
EXEC CICS WRITEQ TS
  ( Queue( 'PSS1' ) | QName() )
  < Sysid() >
  From( 'HOLA SOY PSS1' )
  < Length( +00013 ) >
  < NUmitems( +00003 ) | Item() < Rewrite > >
  < Main | Auxiliary >
  < NOSuspend >

```

6) Utilizar la transacción adecuada y las opciones necesarias para mostrar el contenido de la TS PSSX:

Para ello, hay que entrar en la transacción de "cebr pss1" y ahí ejecutar la instrucción de "get pss1" y muestra:


```

CEBR TSQ PSS1          SYSID S740 REC      4 OF      3      COL      1 OF
ENTER COMMAND ==> GET PSS1
00003 HOLA SOY PSS1
***** BOTTOM OF QUEUE *****

UNABLE TO READ TRANSIENT DATA QUEUE.      EIBRCODE = '020000000000'X
SYSTEM HAS NO ISC CONNECTIONS
PF1 : HELP          PF2 : SWITCH HEX/CHAR      PF3 : TERMINATE BROWSE
PF4 : VIEW TOP       PF5 : VIEW BOTTOM          PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL PF11: SCROLL FORWARD FULL PF12: UNDEFINED

```

7) Verificar que la transacción AMNU está definida y en uso.

cemt i tra(amnu*) => es una transacción para comprobar/verificar que la transacción que empieza por amnu está definida y en uso.

```

CEMT I TRA(AMNU*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Tra(AMNU) Pri( 001 ) Pro(DFH$AMNU) Tc1( DFHTCL00 ) Ena
Sta      Pur Prf(DFHCICST) Uda Any Iso      Bac Wai

```

Como se puede observar, la transacción que empieza por AMNU está definida y en uso, por medio de la instrucción Ena (habilitada).

7.1) Ejecutar la transacción AMNU y probar sus diferentes opciones.

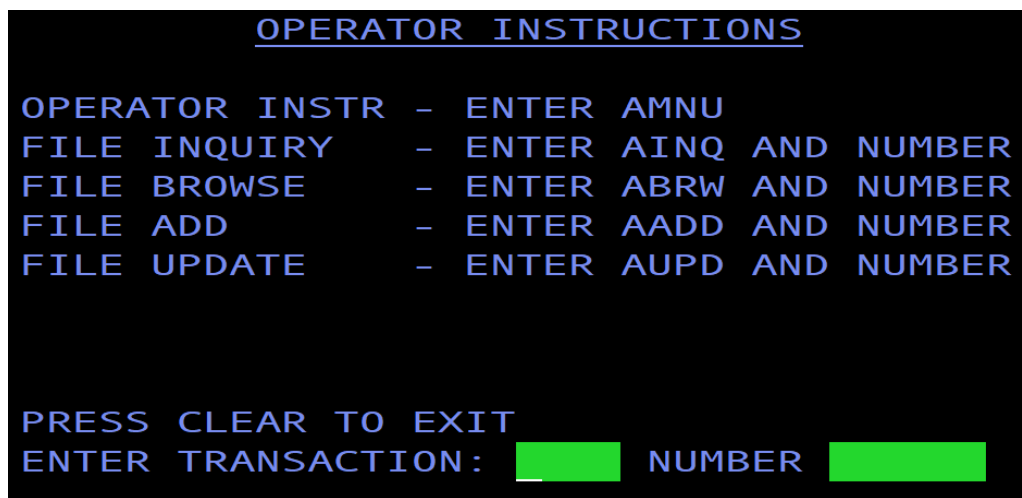
Salir del CEMT a través de F3 y cuando llegues aquí:

```

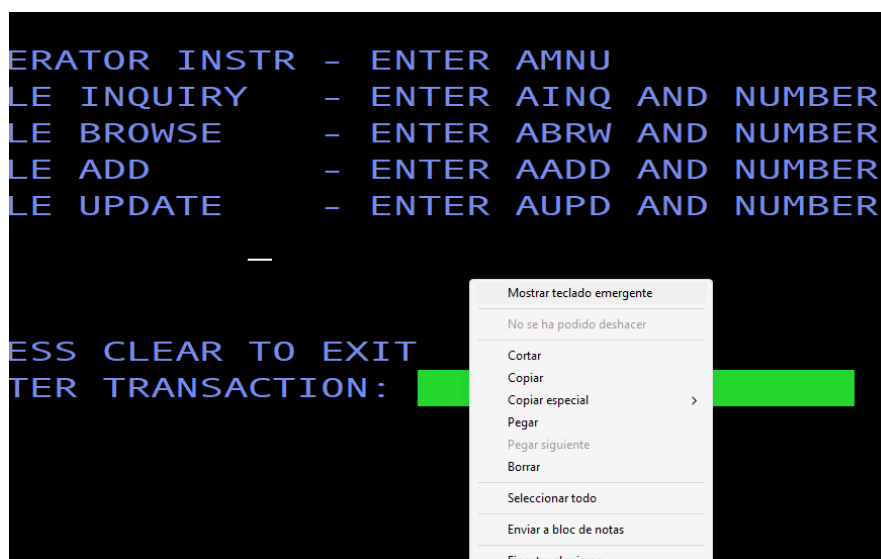
STATUS:  SESSION ENDED

```

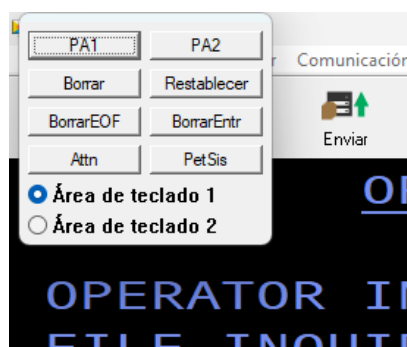
Poner AMNU para ejecutar esa transacción para mostrar:



Para salir, botón derecho > Mostrar teclado emergente:



Y a continuación, borrar:



8) Definir la transacción PSSx (x es el identificador de usuario) para que ejecute el programa DFH\$AMNU. La transacción PSSx debe estar asociada al grupo GRPSSx.

Para ello, escribimos CEDA DEF Trans:

```

DEF trans
OVERTYPE TO MODIFY                                     CICS
CEDA  DEFine TRANSAction(                               )
  TRANSAction  ==> PSS1
  Group        ==> GRPSS1
  DEscription  ==> My First Trans
  PROgram      ==> DFH$AMNU
  TWasize      ==> 00000                                0-32767
  PROfile      ==> DFHCICST
  PARTitionset ==>
  STATus       ==> Enabled                             Enabled | Disabled
  PRIMedsize   : 00000                                0-65520
  TASKDATAloc  ==> Any                                 Below | Any
  TASKDATAkey  ==> User                                User | Cics
  STOrageclear ==> No                                  No | Yes
  RUNaway      ==> System                               System | 0 | 250-2700000
  SHutdown     ==> Disabled                             Disabled | Enabled
  ISolate      ==> Yes                                  Yes | No
  Brexit       ==>
+ REMOTE ATTRIBUTES
MESSAGES: 2 SEVERE

```

Al darle enter:

```

OVERTYPE TO MODIFY                                     CICS
CEDA  DEFine TRANSAction( PSS1 )
  TRANSAction  : PSS1
  Group        : GRPSS1
  DEscription  ==> MY FIRST TRANS
  PROgram      ==> DFH$AMNU
  TWasize      ==> 00000                                0-32767
  PROfile      ==> DFHCICST
  PARTitionset ==>
  STATus       ==> Enabled                             Enabled | Disabled
  PRIMedsize   : 00000                                0-65520
  TASKDATAloc  ==> Any                                 Below | Any
  TASKDATAkey  ==> User                                User | Cics
  STOrageclear ==> No                                  No | Yes
  RUNaway      ==> System                               System | 0 | 250-2700000
  SHutdown     ==> Disabled                             Disabled | Enabled
  ISolate      ==> Yes                                  Yes | No
  Brexit       ==>
+ REMOTE ATTRIBUTES
MESSAGES: 1 WARNING 1 INFORMATORY
                                           SYSID=S740
DEFINE SUCCESSFUL                                DSN=DFH610.CICS.DFHCS

```

Y ahora tienes que ir a atrás con F3 y escribir install. Bajar y escribir en Group: GRPSS1 y, en Trans: PSS1. Por último darle a enter.

9) Verificar que la transacción PSS1 está definida y en uso:

Para ello, hay que introducir: cemt i tra(pss1*)

```

I TRANS(PSS1*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Tra(PSS1) Pri( 001 ) Pro(DFH$AMNU) Tc1( DFHTCL00 ) Ena
Sta      Pur Prf(DFHCICST) Uda Any Iso                      Bac Wai

```

9.1) Ejecutar la transacción PSS1 y probar sus diferentes opciones.

Salir del CEMT a través de F3 y cuando llegues aquí:

```
STATUS:  SESSION ENDED
```

Poner PSS1 para ejecutar esa transacción y mostrar:

OPERATOR INSTRUCTIONS

```
OPERATOR INSTR - ENTER AMNU
FILE INQUIRY    - ENTER AINQ AND NUMBER
FILE BROWSE     - ENTER ABRW AND NUMBER
FILE ADD        - ENTER AADD AND NUMBER
FILE UPDATE     - ENTER AUPD AND NUMBER
```

PRESS CLEAR TO EXIT

ENTER TRANSACTION: NUMBER

10) Definir la TDQ PSQx en el grupo GRPSSx.

Para ello, hay que introducir la sentencia o instrucción de "ceda def td" y muestra un panel a rellenar:

```
def td
OVERTYPE TO MODIFY
CEDA DEFINE TDqueue(      )
  TDqueue    ==> psq1
  Group      ==>
  Description ==>
  TYPE       ==>
EXTRA PARTITION PARAMETERS
  Databuffers ==>
  DDname      ==>
  DSname      ==>
  Sysoutclass ==>
  Erroroption ==>
  Opentime    ==>
  REWind      ==>
  TYPEFile    ==>
  RECORDSize  ==>
  BLOCKSize   ==>
  RECORDFormat ==>
+ BLOCKFormat ==>
```

En TDqueue le ponemos “psq1”, en Group “grpss1”, en la descripción “my first queue” y en type “extra”, para que la cola se comparta con otros jobs. También, podrías poner en vez de extra, intra en type, para que la cola sólo exista dentro del cics. Mejor ponemos “intra”. Luego, le damos a enter y muestra:

```
OVERTYPE TO MODIFY                                CICS RELE
CEDA  DEfine TDqueue( PSQ1 )
  TDqueue      : PSQ1
  Group        : GRPSS1
  DEscription  ==> MY FIRST QUEUE
  TYPE         ==> INtra                          Extra | INtra | INDirect
EXTRA PARTITION PARAMETERS
  Databuffers  : 1-255
  DDname       :
  DSname       :
  Sysoutclass  :
  Erroroption  : Ignore | Skip
  Opentime     : Initial | Deferred
  REWind       : Leave | Reread
  TYPEFile     : Input | Output | Rdback
  RECORDSize   : 0-32767
  BLOCKSize    : 0-32767
  RECORDFormat : Fixed | Variable
+ BLOCKFormat  : Blocked | Unblocked

                                           SYSID=S740 APPL
DEFINE SUCCESSFUL                        DSN=DFH610.CICS.DFHCS
```

11) Verificar que el grupo GRPSSx contiene la transacción PSSx y la cola PSQx.

Para ello, hay que introducir la sentencia o instrucción de “ceda di” y muestra una serie de campos a rellenar:

```
d1
OVERTYPE TO MODIFY
CEDA  Display
  Group      ==> -
  LIST       ==> -
  ALL        ==> *
  ATomservice ==>
  Bundle     ==>
  CONnection ==>
  CORbaserver ==>
  DB2Conn    ==>
  DB2Entry   ==>
  DB2Tran    ==>
  DJar       ==>
  DOctemplate ==>
  DUmpcode   ==>
  Enqmodel   ==>
  File       ==>
  Ipconn     ==>
+ JOURNALmodel ==>
```

Rellenamos sólo donde pone Group con “grpss1”, damos a enter y muestra lo siguiente:

ENTER COMMANDS			LAST CHANGE
NAME	TYPE	GROUP	
PSS1	TRANSACTION	GRPSS1	04/16/25 06:21:01
PSQ1	TDQUEUE	GRPSS1	04/21/25 05:44:06

12) Grabar un registro en la cola PSQx.

Para ello, hay que entrar en la transacción de “ceci” y ahí ejecutar la instrucción de: writeq td qu ('psq1') from ('Hola soy TDQ1') y muestra lo siguiente:

```
WRITEQ TD QU ('PSQ1') FROM ('HOLA SOY TDQ1') _
STATUS: ABOUT TO EXECUTE COMMAND
EXEC CICS WRITEQ TD
  Queue( 'PSQ1' )
  < Sysid() >
  From( 'HOLA SOY TDQ1' )
  < Length( +00013 ) >
```

13) Acceder a la TS PSSx (como en el ejercicio 6) y recuperar el registro grabado en la TD PSQx.

Para ello, hay que entrar en la transacción de “cebr pss1” y ahí ejecutar la instrucción de “get psq1”. Después muestra lo siguiente:

```
CEBR TSQ PSS1          SYSID S740 REC      1 OF      2      COL      1 OF     13
ENTER COMMAND ==>_
***** TOP OF QUEUE *****
00001 HOLA SOY TDQ1
00002 HOLA SOY TDQ1
***** BOTTOM OF QUEUE *****

SYSTEM HAS NO ISC CONNECTIONS
PF1 : HELP          PF2 : SWITCH HEX/CHAR      PF3 : TERMINATE BROWSE
PF4 : VIEW TOP      PF5 : VIEW BOTTOM          PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

No hay conexiones ISC porque no lo hemos establecido en la definición. Es un warning, no hay ningún problema.

14) Ejecutar la transacción PSSx bajo control de la transacción de “debuging” proporcionada por CICS.

Para ello, hay que salir de la transacción anterior, borrar la pantalla y entrar en la transacción de “cedf”, dando a enter y mostrando:

```
THIS TERMINAL: EDF MODE ON
```

Borramos y ponemos “pss1”. Le damos a enter y muestra:

```
TRANSACTION: PSS1 PROGRAM: DFH$AMNU TASK: 0004155 APPLID: CICSTS61 DISPLAY: 00
STATUS: PROGRAM INITIATION

EIBTIME      = 70413
EIBDATE      = 0125111
EIBTRNID     = 'PSS1'
EIBTASKN     = 4155
EIBTRMID     = '0013'

EIBCPOSN     = 4
EIBCALEN     = 0
EIBAID       = X'7D' AT X'21D1011A'
EIBFN        = X'0000' AT X'21D1011B'
EIBRCODE     = X'00000000000000' AT X'21D1011D'
EIBDS        = '.....'
+ EIBREQID    = '.....'

ENTER: CONTINUE
PF1 : UNDEFINED PF2 : SWITCH HEX/CHAR PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS PF5 : WORKING STORAGE PF6 : USER DISPLAY
PF7 : SCROLL BACK PF8 : SCROLL FORWARD PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY PF11: EIB DISPLAY PF12: UNDEFINED
```

Enter de nuevo:

```
TRANSACTION: PSS1 PROGRAM: DFH$AMNU TASK: 0004155 APPLID: CICSTS61 DISPLAY: 00
STATUS: ABOUT TO EXECUTE COMMAND
EXEC CICS SEND MAP
MAP ('DFH$AGA')
FROM ('.....J.....'.....)
MAPONLY
TERMINAL
ERASE
NOHANDLE

OFFSET:X'000176' LINE: EIBFN=X'1804'

ENTER: CONTINUE
PF1 : UNDEFINED PF2 : SWITCH HEX/CHAR PF3 : UNDEFINED
PF4 : SUPPRESS DISPLAYS PF5 : WORKING STORAGE PF6 : USER DISPLAY
PF7 : SCROLL BACK PF8 : SCROLL FORWARD PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY PF11: EIB DISPLAY PF12: ABEND USER TASK
```

NOTA) Para finalizar la sesión de CICS, escribir “cesf” y enter, yendo de nuevo al inicio:

Application Developer System

```
          // 0000000 SSSSS
        // 00    00 SS
zzzzzz // 00    00 SS
      zz // 00    00 SSSS
    zz  // 00    00    SS
  zz   // 00    00    SS
zzzzzz // 0000000 SSSS
```

System Customization - ADCD.Z25D.*

==> Enter "LOGON" followed by the TS0 userid. Example "LOGON IBMUSER" or

==> Enter L followed by the APPLID

==> Examples: "L TS0", "L CICSTS56", "L CICSTS61", "L IMS15"

Práctica CICS LAB 2

En esta práctica vamos a modificar un programa (pgm) de COBOL CICS y a realizar las definiciones necesarias en el entorno CICS para la ejecución del programa, que mostrará un “saludo” por pantalla. (Nota: x → identificador de usuario).

A través de mi usuario de TSO:

Creo un Data Set particionado de tipo Librería con el nombre de **TSOPSS1.CICS.SRC** => para almacenar fuentes de programas.

Creo un Data Set particionado de tipo Librería con el nombre de **TSOPSS1.CICS.OBJ** => para almacenar el objeto del programa.

Creo un Data Set particionado de tipo Librería con el nombre de **TSOPSS1.CICS.LOAD** => para almacenar el ejecutable del programa.

Al compilar el pgm COBOL utilizando el JCL ajustado me da:

```
17.25.58 JOB00094 $HASP165 TP1CICS ENDED AT S0W1 MAXCC=0000 CN(INTERNAL)
*** _
```

RC = 00