

596
1995
C

**INFORME FINAL
FONTEC - CORFO**

PROYECTO

**“ DESARROLLO DEL SISTEMA DE
ADMINISTRACIÓN DE FILAS Y CAJAS
BANQUEUE VERSIÓN 2 “**

005.1
E 96
1995

**EXCELSYS
CÓDIGO 93-0193**

FONTEC

**SISTEMA DE ADMINISTRACION DE
FILAS Y CAJAS BANQUEUE II**

INFORME FINAL

D-90-0001-57-033-02

Diciembre, 1995

TABLA DE CONTENIDO

título	página
1. RESUMEN	1-1
2. INTRODUCCIÓN	2-1
2.1 OBJETIVOS GENERALES	2-1
2.2 OBJETIVOS ESPECÍFICOS	2-2
3. DESARROLLO	3-1
3.1 HARDWARE.....	3-1
3.2 SOFTWARE	3-1
4. EVALUACION DEL PROTOTIPO	4-1
5. CONCLUSIONES	5-1
6. ANEXOS.....	6-1

1. RESUMEN

El presente documento es el último y final, y en él se da cuenta del desarrollo del Sistema de Filas y Cajas BANQUEUE II.

En la introducción se describe de manera general el sistema desarrollado.

Al ofrecer el producto, tanto en Chile como en el extranjero, hemos podido darnos cuenta que la solución diseñada preliminarmente no era del todo adecuada. Si bien la funcionalidad del producto resultaba atractiva a los clientes, el mayor precio constituía una fuerte limitación para su comercialización.

Gracias a la información obtenida mediante un estudio de mercado se pudo identificar que el interés de los clientes apuntaba a un sistema que tuviera las siguientes características:

1. Bajo Costo
2. Expansible
3. Simple de esperar

A la luz de esta información, se decidió poner mayor énfasis en el desarrollo del producto más económico de los de línea BANQUEUE II, privilegiando la modularidad de su diseño, a objeto de permitir un crecimiento hacia configuraciones más poderosas a partir de él. Era importante también, poner énfasis en desarrollar un sistema que poseyera una interfaz simple que facilitara su operación.

Desarrollar el sistema de acuerdo a este nuevo enfoque tuvo asociado un mayor esfuerzo de desarrollo que el contemplado al comienzo, básicamente por los siguientes conceptos:

1. El esfuerzo invertido en el primer enfoque en su mayor parte no era reutilizable.
2. El aumento de modularidad del sistema implicó un aumento en las horas hombre para diseño y pruebas: En particular se incrementó tanto la cantidad como la complejidad de las tarjetas electrónicas.
3. Para bajar los costos fue necesario investigar sensores de paso más económicos, investigación que tuvo magros resultados. De hecho, se decidió no incluir esta opción en los BANQUEUE II, por cuanto su costo era demasiado alto.
4. Fue necesario trasladar el desarrollo del Software del Sistema de Administración desde una plataforma de alto nivel (PC compatible) a un sistema empotrado de bajo nivel (familia MPU-51 de Intel). Esto también tuvo asociado un costo más alto en codificación, pruebas y mantención del software.

Sin embargo, se espera que este mayor costo de desarrollo sea recuperado con un incremento en el volumen de ventas que se percibirá producto de la disminución en el precio final y de su versatilidad. El sistema desarrollado es de menor funcionalidad que la inicial pero puede ser expandido en la medida que los requerimientos del cliente se modifiquen.

El sistema desarrollado bajo este nuevo enfoque está siendo ofrecido en Chile y Méjico con resultados bastante auspiciosos. A pesar de que para esta etapa no se pudo conseguir un cliente que aceptara probar una instalación de este tipo, al momento de redactar este informe se tiene una orden de compra por dos unidades para DIMACOFI , una orden de compra por dos unidades para BANCO OSORNO y una oferta por 1000 unidades a un banco de Méjico. Adicionalmente existen numerosas proposiciones en curso con interesantes posibilidades.

2. INTRODUCCIÓN

El Sistema BANQUEUE corresponde a un administrador de filas de espera que direcciona a los clientes hacia las cajas disponibles, mejorando la eficiencia de atención. Adicionalmente, puede entregar información importante, por un lado al cliente sobre el tiempo de espera probable y, por otro, permite al administrador de la sucursal disponer de información en tiempo real de la situación tanto de las cajas como de las filas. Al cliente, cuando sea su turno, el sistema le indica, mediante señales luminosa y acústica, la caja que está disponible para su atención.

A la Institución le permite evaluar objetivamente el rendimiento del personal, determinar la planta de personal necesaria en cada Sucursal (permanente y flotante para periodos punta), predecir el comportamiento del público, ajustando el personal requerido para la mejor atención del público a costos mínimos.

2.1 OBJETIVOS GENERALES

1. Disminuir Costos de Instalación del Sistema.
2. Rediseñar el hardware para bajar costos de materiales y fabricación.
3. Rediseñar el hardware para lograr un Sistema Modular que permita configurar desde sistemas mínimos hasta el sistema con funcionalidad completa, y de esta manera ofrecer a los clientes distintas alternativas que se ajusten a sus necesidades, a la vez que tengan la opción de ampliar su configuración según sus requerimientos.
4. Diseñar y desarrollar un producto que cumpla con normas de calidad de exportación
5. Validar el sistema en base a operación de prototipo en instalación piloto.

6. Diseñar estrategias de comercialización nacional e internacional e implementar normas de producción compatibles con este objetivo.

2.2 OBJETIVOS ESPECÍFICOS

1. Desarrollar el hardware necesario para la versión 2.
2. Rediseñar el software para que se adapte a los nuevos requerimientos surgidos por la modificación de hardware.
3. Incorporar mejoras en la entrega de datos estadísticos al usuario.
4. Integración hardware-software para la nueva versión.
5. Instalar y modificar el prototipo en oficinas de un cliente.
6. Implementar normas de producción.
7. Investigar medios de promoción y distribución de este tipo de productos.

3. DESARROLLO

3.1 HARDWARE

El hardware definido en las etapas anteriores fue completamente diseñado, construido y probado, con la sola excepción de los sensores de paso, los que debido a su alto costo en recursos financieros como humanos, fueron postergados para más adelante. En verdad no se visualizan soluciones a priori, por lo que se deberá invertir aún más en investigaciones al respecto.

3.2 SOFTWARE

El software definido en las etapas anteriores fue diseñado, construido y probado. Es de hacer notar que el software nunca se da por terminado pues siempre puede ser perfeccionado, a la vez que se le debe dar mantención. Así, en el presente caso, el software realizado que fue escrito en lenguaje Assembler, será traducido a lenguaje C, debido a las facilidades que esto arroja en cuanto a modularidad y mantenibilidad, lo que se traducirá en un menor costo de realización para nuevos proyectos.

4. EVALUACION DEL PROTOTIPO

El prototipo fue instalado en nuestros laboratorios pues no hubo ninguna empresa o institución que nos permitiera hacerlo en sus instalaciones para efectos de realizar una evaluación.

Aquí, se le sometió a un riguroso funcionamiento: encendido el 100 % del tiempo, condiciones de temperaturas más o menos elevadas (sala cerrada con estufa encendida), condiciones de humedad relativamente altas, activaciones de dos cajeros, a cualquier hora del día, largos períodos sin actividad, periodos de intensa actividad, etc.

Producto de este funcionamiento del prototipo en nuestros laboratorios como instalación piloto, se determinó realizar algunos cambios relativos a la presentación del producto, y otros de mejoramiento del hardware, como por ejemplo la fuente de poder.

Así, se desecha el uso dígitos de 2" en favor del uso de dígitos de 7". Esto requiere de modificaciones en la tarjeta de display.

Además, se desecha el uso de matrices de puntos de 1" en la tarjeta semáforo, en favor de matrices de 2". Junto con ello, se define el uso de 12 matrices por cada display. Esto requerirá cambios en la tarjeta de semáforo.

Por último, se decide incluir, en el software, un driver para manejar una impresora térmica, serial, de 80 mm., con autocorte. Con esto, se unificarán los productos BANQUEUE y VIQ, usando ambos el mismo hardware y gran parte del software, consiguiéndose así una baja importante en el costo.

Todos estos cambios se realizarán a la brevedad con el objeto de incluirlos en las órdenes de compra recibidas.

5. CONCLUSIONES

Como conclusión de este proyecto puede decirse que ha resultado bastante exitoso por cuanto se han cumplido prácticamente todos los objetivos planteados inicialmente, habiéndose logrado soluciones técnicas apropiadas y de bajo costo.

Se tiene ahora un hardware modular, reutilizable, que cumple las funciones especificadas para el BANQUEUE II, además de cumplir con las funciones establecidas para otros productos como los sistemas VIQ.

Paralelamente a los anterior, y dado que el hardware se ha estandarizado, también se ha estandarizado ciertas funciones de software específicas, como aquellos programas escritos para el manejo de dispositivos, por lo que se espera una reducción importante de costos en el desarrollo de software para todos los productos involucrados.

Desde el punto de vista comercial, el sistema diseñado cumple con las expectativas de casi todos los potenciales clientes, puesto que la flexibilidad lograda permite satisfacer las necesidades básicas, e ir incrementando la funcionalidad en base a agregar tarjetas electrónicas y paquetes de software. Esto, bajo un esquema de costos bastante razonable, que se ve ratificado con la venta de cuatro unidades en un periodo de tiempo bastante menor que en épocas anteriores.

Para ayudar a la gestión comercial, se han diseñado nuevos folletos haciendo hincapié en las nuevas características del producto, y en especial respecto de sus ventajas con otros equivalentes.

En la estrategia de ventas se ha orientado al nicho de mercado formado por bancos, Isapres, AFP y grandes empresas de servicios, usando como medios para llegar a ese mercado folletos, cartas, diskettes con demos, reuniones explicativas, etc.

También se asigna gran importancia a la decisión de usar métodos de producción ISO 9000 para la fabricación del hardware, aprovechando la experiencia lograda en el proceso de certificación ISO 9001/3 para la producción de software (proceso actualmente en marcha en conjunto con el grupo de empresas de CEES, Comité de Empresas Exportadoras de Software).

6. ANEXOS

En las páginas siguientes se anexa listados de software en assembler:

control1.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta control - puerta p1-

```
mov a,#00h
ini: mov r1,#0ffh
djnz r1,$
cpl a
mov 90h,a
ljmp ini
```

control2.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta control - watch_dog -

```
ini: mov dptr,#9000h
setb acc.7
movx @dptr,a
mov r1,#0ffh
djnz r1,$
clr acc.7
mov dptr,#9000h
movx @dptr,a
mov r1,#0ffh
djnz r1,$
mov a,#00h
ljmp ini
```

control4.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta control - lectura u2 ; escritura u10 -

```
mov r1,#07h
dly1: mov r2,0ffh
dly2: mov r3,#0ffh
      djnz r3,$
      djnz r2,dly2
      djnz r1,dly1
ini:  mov dptr,#8000h
      movx a,@dptr
      mov dptr,#9000h
      movx @dptr,a
      ljmp ini
```

control5.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta control - watch_dog - con int.

```
sjmp start
org 0bh
call ini
reti
ini:  mov dptr,#8000h
      movx a,@dptr
      jb acc.0,x1
      mov dptr,#9000h
      setb acc.7
      movx @dptr,a
      mov r1,#0ffh
      djnz r1,$
      clr acc.7
      mov dptr,#9000h
      movx @dptr,a
      mov r1,#0ffh
      djnz r1,$
      mov a,#00h
```

```
x1:  ret
start: mov  r1,#07h
dly1:  mov  r2,#0ffh
dly2:  mov  r3,#0ffh
      djnz r3,$
      djnz r2,dly2
      djnz r1,dly1
      mov  ien0,#82h
      mov  tmod,#02h
      mov  th1,#00h
      mov  tcon,#10h
      sjmp $
```

control6.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta control - ram -

```
ini:  mov  dptr,#0e000h
      mov  a,#99h
      movx @dptr,a
      mov  dptr,#0e000h
      movx a,@dptr
      mov  p1,a
      sjmp ini
```

control7.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta control - puerta serial -

```
mov  dptr,#9000h
mov  a,#0ffh
movx @dptr,a
mov  tmod,#21h
mov  th1,#0f3h
mov  tcon,#0f0h
mov  scon,#52h
```

dat: mov a,#41h

```
call tx
sjmp dat
```

```
tx:  jnb  ti,$
      clr  ti
      mov  sbuf,a
      ret
```

control8.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta display

```
call dely
```

main: mov a,#0f5h

```
call out
mov  a,#14h
call out
mov  a,#0cdh
call out
mov  a,#5dh
call out
mov  a,#3ch
call out
mov  a,#79h
call out
mov  a,#0fah
call out
```

```
    mov  a,#15h
    call out
    mov  a,#0fdh
    call out
    mov  a,#3fh
    call out
    sjmp main
out:  mov  dptr,#0a200h
    movx @dptr,a
    call dely
    mov  dptr,#0a201h
    movx @dptr,a
    call dely
    ret
dely: mov  r1,#05h
dly1: mov  r2,0ffh
dly2: mov  r3,#0ffh
    djnz r3,$
    djnz r2,dly2
    djnz r1,dly1
    ret
```

```
control9.asm
    defseg gino, absolute
    seg gino
    org 0000h

;test tarjeta display
    call dely
x1:  call both
    call dig1
    call dig2
    sjmp x1
both: mov  a,#0f5h
    call out
    mov  a,#14h
    call out
    mov  a,#0cdh
    call out
    mov  a,#5dh
    call out
    mov  a,#3ch
    call out
    mov  a,#79h
    call out
    mov  a,#0fbh
    call out
    mov  a,#15h
    call out
    mov  a,#0fdh
    call out
    mov  a,#3fh
    call out
    ret
out:  mov  dptr,#0a200h
    movx @dptr,a
    mov  dptr,#0a201h
    movx @dptr,a
    call dely
    ret
dig1: mov  dptr,#0a201h
    mov  a,#00h
    movx @dptr,a
    mov  a,#01h
```

```
    call o1
    mov  a,#05h
    call o1
    mov  a,#15h
    call o1
    mov  a,#55h
    call o1
    mov  a,#0d5h
    call o1
    mov  a,#0f5h
    call o1
    mov  a,#0fdh
    call o1
    mov  a,#0ffh
    call o1
    ret
o1:  mov  dptr,#0a200h
     movx @dptr,a
     call dely
     ret
dig2: mov  dptr,#0a200h
      mov  a,#00h
      movx @dptr,a
      mov  a,#01h
      call o2
      mov  a,#05h
      call o2
      mov  a,#15h
      call o2
      mov  a,#55h
      call o2
      mov  a,#0d5h
      call o2
      mov  a,#0f5h
      call o2
      mov  a,#0fdh
      call o2
      mov  a,#0ffh
      call o2
      ret
o2:  mov  dptr,#0a201h
     movx @dptr,a
```

```
    call dely
    ret
dely: mov  r1,#04h
dly1: mov  r2,0ffh
dly2: mov  r3,#0ffh
      djnz r3,$
      djnz r2,dly2
      djnz r1,dly1
      ret

io.asm
    defseg gino, absolute
    seg gino
    org 0000h

;test tarjeta input/output
    call dely
    mov  dptr,#0a203h
    mov  a,#90h
    movx @dptr,a
    mov  dptr,#0a201h
    mov  a,#99h
    movx @dptr,a
    mov  dptr,#0a202h
    mov  a,#66h
    movx @dptr,a
x1:  mov  dptr,#0a200h
    movx a,@dptr
    mov  p1,a
    sjmp x1
dely: mov  r1,#04h
dly1: mov  r2,0ffh
dly2: mov  r3,#0ffh
      djnz r3,$
      djnz r2,dly2
      djnz r1,dly1
      ret
```

semaf.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta semaforo

```
call dely
mov sp,#60h
ini: mov 10h,#07h
x1:  mov a,#00h
     mov r1,#00h
     mov r2,#00h
     mov r3,#00h
     mov r4,#00h
     mov dptr,#0a200h
     movx @dptr,a
     mov dptr,#0a201h
     movx @dptr,a
     mov dptr,#0a202h
     movx @dptr,a
     mov dptr,#0a203h
     movx @dptr,a
     orl 01h,#01h
     call u1
     orl 01h,#02h
     call u1
     orl 01h,#04h
     call u1
     orl 01h,#08h
     call u1
     orl 01h,#10h
     call u1
     orl 02h,#08h
     call u2
     orl 02h,#10h
     call u2
     orl 01h,#20h
     call u1
     orl 01h,#40h
     call u1
     orl 01h,#80h
     call u1
```

```
orl 02h,#40h
call u2
orl 02h,#80h
call u2
orl 02h,#01h
call u2
orl 02h,#02h
call u2
orl 02h,#04h
call u2
orl 03h,#02h
call u3
orl 03h,#04h
call u3
orl 03h,#08h
call u3
orl 03h,#10h
call u3
orl 02h,#20h
call u2
orl 04h,#10h
call u4
orl 03h,#20h
call u3
orl 03h,#40h
call u3
orl 03h,#80h
call u3
orl 03h,#01h
call u3
orl 04h,#80h
call u4
orl 04h,#01h
call u4
orl 04h,#02h
call u4
orl 04h,#04h
call u4
orl 04h,#08h
call u4
djnz 10h,x2
orl 04h,#40h
```

```
    call u4
    ljmp ini
x2:  orl  04h,#20h
    call u4
    ljmp x1
u1:  mov  dptr,#0a200h
    mov  a,r1
    movx @dptr,a
    call dely
    ret
u2:  mov  dptr,#0a201h
    mov  a,r2
    movx @dptr,a
    call dely
    ret
u3:  mov  dptr,#0a202h
    mov  a,r3
    movx @dptr,a
    call dely
    ret
u4:  mov  dptr,#0a203h
    mov  a,r4
    movx @dptr,a
    call dely
    ret
dely: mov  r5,#01h
dly1: mov  r6,#0ffh
dly2: mov  r7,#0ffh
    djnz r7,$
    djnz r6,dly2
    djnz r5,dly1
    ret
```

soni.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta sonido

```
call dely
mov  dptr,#0a200h
mov  a,#00h
movx @dptr,a
x1:  mov  dptr,#8000h
     movx a,@dptr
     mov  r1,a
     jb  acc.0,z1
     sjmp x1
z1:  mov  dptr,#0a200h
     mov  a,#02h
     movx @dptr,a
     mov  dptr,#0a201h
     mov  a,r1
     anl a,#0fah
     movx @dptr,a
     sjmp x1
dely: mov  r5,#01h
dly1: mov  r6,#0ffh
dly2: mov  r7,#0ffh
     djnz r7,$
     djnz r6,dly2
     djnz r5,dly1
     ret
```

soni1.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta sonido -programado en memoria-

```
call dely
```

```
z1: mov  dptr,#0a200h
```

```
mov  a,#02h
```

```
movx @dptr,a
```

```
mov  a,#00h
```

```
s1: mov  a,#0f0h
```

```
call out
```

```
mov  a,#10h
```

```
call out
```

```
mov  a,#0a0h
```

```
call out
```

```
mov  a,#50h
```

```
call out
```

```
mov  a,#0b0h
```

```
call out
```

```
mov  a,#20h
```

```
call out
```

```
mov  a,#0c0h
```

```
call out
```

```
mov  a,#30h
```

```
call out
```

```
mov  a,#0d0h
```

```
call out
```

```
mov  a,#60h
```

```
call out
```

```
mov  a,#90h
```

```
call out
```

```
mov  a,#70h
```

```
call out
```

```
mov  a,#80h
```

```
call out
```

```
ljmp s1
```

```
sjmp s1
```

```
out: mov  dptr,#0a201h
```

```
movx @dptr,a
```

```
call tem
```

```
ret
tem: mov r5,#01h
dlz1: mov r6,#0ffh
dlz2: mov r7,#0ffh
      djnz r7,$
      djnz r6,dlz2
      djnz r5,dlz1
      ret
dely: mov r5,#01h
dly1: mov r6,#0ffh
dly2: mov r7,#0ffh
      djnz r7,$
      djnz r6,dly2
      djnz r5,dly1
      ret
```

soni2.asm

```
defseg gino, absolute
seg gino
org 0000h
```

;test tarjeta sonido -programado en memoria-

```
call dely
z1:  mov dptr,#0a200h
     mov a,#02h
     movx @dptr,a
s1:  mov a,#00h
     call out
     mov a,#10h
     call out
     mov a,#20h
     call out
     mov a,#30h
     call out
     mov a,#40h
     call out
     mov a,#50h
     call out
     mov a,#60h
     call out
     mov a,#70h
     call out
```

```
mov a,#80h
call out
mov a,#90h
call out
mov a,#0a0h
call out
mov a,#0b0h
call out
mov a,#0c0h
call out
mov a,#0d0h
call out
mov a,#0e0h
call out
mov a,#0f0h
call out
ljmp sl
sjmp sl
out: mov dptr,#0a201h
movx @dptr,a
call tem
ret
tem: mov r5,#01h
dlz1: mov r6,#0ffh
dlz2: mov r7,#0ffh
djnz r7,$
djnz r6,dlz2
djnz r5,dlz1
ret
dely: mov r5,#01h
dly1: mov r6,#0ffh
dly2: mov r7,#0ffh
djnz r7,$
djnz r6,dly2
djnz r5,dly1
ret
```

test.asm

```
defseg gino, absolute
seg gino
org 0000h
```

```
mov dptr,#0e003h
mov a,#80h
movx@dptr,a
mov dptr,#0e000h
```

```
ini mov r1,#0ffh
```

```
inl mov a,r1
movx@dptr,a
djnz r1,inl
ljmp ini
end
```