

Primera prueba: senales generadas con kill

```
main enviando senal 38
El hilo 0 va a esperar senales 38
El hilo 0 ha recibido la senal prevista 38
El hilo 1 va a esperar senales 39
El hilo 2 va a esperar senales 40
El hilo 3 va a esperar senales 41
main enviando senal 39
El hilo 1 ha recibido la senal prevista 39
main enviando senal 40
El hilo 2 ha recibido la senal prevista 40
main enviando senal 41
El hilo 3 ha recibido la senal prevista 41
main cancelando al hilo 0
main cancelando al hilo 1
main cancelando al hilo 2
main cancelando al hilo 3
```

Segunda prueba: senales generadas con pthread_kill

```
El hilo 0 va a esperar senales 38
El hilo 1 va a esperar senales 39
El hilo 2 va a esperar senales 40
El hilo 3 va a esperar senales 41
Main enviando senal 38 a los 4 threads
Main enviando senal 39 a los 4 threads
El hilo 0 ha recibido la senal prevista 38
Main enviando senal 40 a los 4 threads
El hilo 1 ha recibido la senal prevista 39
Main enviando senal 41 a los 4 threads
El hilo 3 ha recibido la senal prevista 41
cancelando el hilo 0
El hilo 2 ha recibido la senal prevista 40
cancelando el hilo 1
cancelando el hilo 2
cancelando el hilo 3
```

Tercera prueba: Senales capturadas con manejador

```
El hilo 0 va a admitir senales 38
El hilo 1 va a admitir senales 39
El hilo 2 va a admitir senales 40
El hilo 3 va a admitir senales 41
Main enviando senal 38
El manejador recibe la senal 38
nanosleep de 0 interrumpido: Interrupted system call
Main enviando senal 39
El manejador recibe la senal 39
nanosleep de 1 interrumpido: Interrupted system call
Main enviando senal 40
El manejador recibe la senal 40
nanosleep de 2 interrumpido: Interrupted system call
Main enviando senal 41
El manejador recibe la senal 41
nanosleep de 3 interrumpido: Interrupted system call
```

Cuarta prueba: Error de segmentacion

```
Detectado error de segmentacion!
El thread erroneo es el culpable.
```