

Alumne 1:	Nota:
Alumne 2:	

Fecha de Entrega

La fecha límite de entrega del problema es el **martes 3 de noviembre de 2020 a las 9h** (acordaros de hacer una copia de los problemas para la corrección).

Enunciado Problema

PROBLEMA 1 [10 puntos]

Hacienda nos ha reclamado nuestros servicios para que mejoremos su programa de control de las declaraciones. Debido a la antigüedad de sus equipos informáticos y el reciente aumento de las declaraciones, su antiguo programa ya no es capaz de revisar todas las declaraciones (sobre todo las positivas) antes de la fecha límite.

El gabinete de crisis organizado para solventar el problema, ha oído rumores de que no es necesario actualizar el equipo informático (un 80386 en muy buen estado) si se utiliza una aplicación concurrente en vez de la aplicación secuencial original.

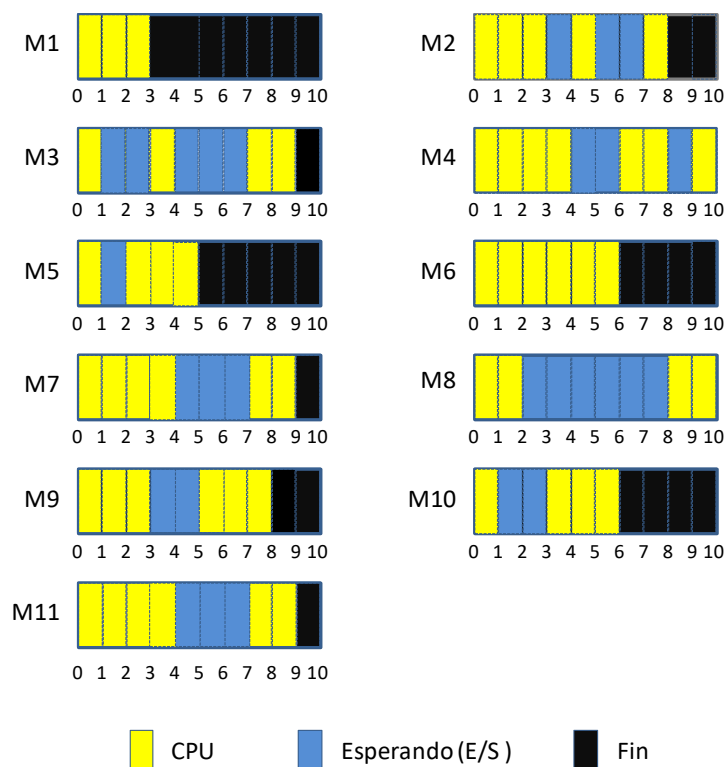
Se nos propone que diseñemos la aplicación concurrente a partir de la siguiente especificación de cada uno de los módulos del programa original:

- **Módulo 1 (M1):** Lee los datos del contribuyente desde el disco y los carga en memoria para su posterior procesamiento.
- **Módulo 2 (M2):** A partir de los datos leídos por el módulo 1 (datos personales contribuyente), se valida la identificación del contribuyente para verificar que no exista una declaración repetida.
- **Módulo 3 (M3):** Se preprocesa y clasifica la información de contabilidad leída por el módulo 1 en tres categorías: ingresos, reducciones y pagos. Esta información será procesada por los *módulos 6, 7 y 8 respectivamente*.
- **Módulo 4 (M4):** A partir de los datos personales de los contribuyentes validados proporcionados por el módulo 2, se comprueba si dicho contribuyente aparece en las bases de datos de defraudadores en paraísos fiscales proporcionadas por Falciani. Esta información será procesada por el *módulo 10*.
- **Módulo 5 (M5):** A partir de la validación realizada por el módulo 2 y el módulo 3, se buscarán todas las posibles cuentas ocultas del contribuyente.
- **Módulo 6 (M6):** A partir de los datos personales del módulo 2 y los datos procesados por el módulo 3 relativos a los ingresos del contribuyente, se verifica su cuantía y se calcula la cantidad total ingresada. La información relativa a las cuentas involucradas en estos ingresos se pasan al módulo 11 para su verificación.
- **Módulo 7 (M7):** Una vez validado el contribuyente (módulo 2) y a partir de los datos procesados por el módulo 3 relativos a las deducciones, se comprueba su validez y se calcula la cantidad total a reducir.
- **Módulo 8 (M8):** A partir de los datos procesados por el módulo 3 relativos a los pagos, se valida la existencia de estos pagos y se calcula la cantidad total pagada.

- **Módulo 9 (M8):** A partir de las cantidades totales calculadas por los módulos 7 y 8, este módulo calculará la cantidad que debe pagar ó ingresar el contribuyente. Esta información se pasará al módulo 11.
- **Módulo 10 (M10):** Si el contribuyente tiene una cuenta en un paraíso fiscal (módulo 4) o bien a partir de las posibles cuentas ocultas encontradas (módulo 5), se verifica si existen ingresos adicionales a los declarados por el contribuyente. Esta información se pasa al módulo 11.
- **Módulo 11 (M11):** El módulo 11 recibe la información procesada por el módulo 6, módulo 9 y el módulo 10, generando la notificación de declaración positiva / negativa para el contribuyente ó bien recomendando una inspección en el caso que alguno de los datos proporcionados sean incorrectos.

Se pide:

- Analizar las dependencias entre los módulos de la aplicación anterior y realizar la tabla de concurrencia y el grafo de precedencias de dicha aplicación. [2.5 puntos]
- Implementar el grafo de precedencia anterior mediante los constructores CorBegin/CorEnd (si es posible, sino indicar porque no se puede realizar). [1.5 puntos].
- Implementar el grafo de precedencia anterior mediante los constructores Fork/Join (si es posible, sino indicar porque no se puede realizar). [1.5 puntos].
- Implementar el grafo de precedencia anterior mediante Unix+C (si es posible, sino indicar porque no se puede realizar). [1.5 puntos].
- A partir de la monitorización de la ejecución de la aplicación secuencial anterior se han obtenido las siguientes trazas de Gant de la ejecución secuencial de cada uno de sus módulos:



Calcular el tiempo de ejecución de la aplicación anterior en función del tipo de implementación i de las siguientes características del sistema:

- i. Aplicación secuencial y un procesador con un único núcleo. [0.60 puntos].
- ii. Aplicación concurrente y un procesador con un único núcleo. [0.60 puntos].
- iii. Aplicación secuencial y un procesador con dos núcleos. [0.60 puntos].
- iv. Aplicación concurrente y un procesador con dos núcleos. [0.60 puntos].
- v. Aplicación concurrente y un procesador con cuatro núcleos. [0.60 puntos].