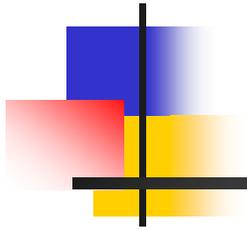
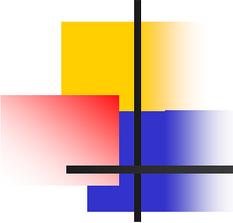


INTRODUCCION

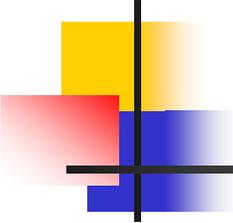
PASCAL-FC





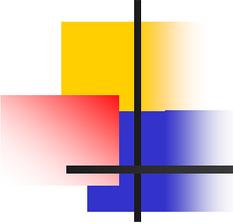
Bibliografía

- Pascal-FC (Version 5): User Guide
- Pascal-FC (Version 5): Language Reference Manual
 - G.L. Davies
 - <http://www-users.cs.york.ac.uk/~burns/pf.html>
- Programción Concurrente
 - J. Tomás, M. Garrido, F. Sánchez, A. Quesada, 2003
 - Capítulo 1 y 2
- Concurrent Programming
 - A. Burns, G Davis. Addison-Wesley, 1993
 - Capítulo 1



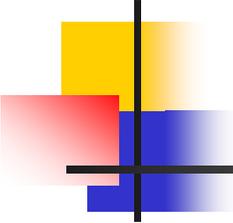
Contenidos

- ¿Por qué Pascal-FC?
- Introducción
- Estructura de un programa
- Estados de un proceso en Pascal-FC
- Ejecución de un programa en Pascal-FC



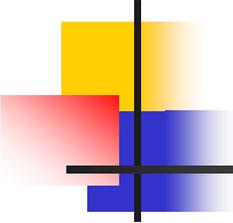
¿Por qué Pascal-FC?

- Lenguaje diseñado para ser empleado como herramienta en la enseñanza de la Programación Concurrente
 - Proporciona una amplia variedad de herramientas de comunicación y sincronización entre procesos (no existentes todas ellas en ningún lenguaje “comercial”)
 - Elimina características de otros lenguajes como Punteros, Ficheros, etc... de forma que el alumno pueda centrarse en el aprendizaje del manejo de las herramientas de comunicación y sincronización entre procesos



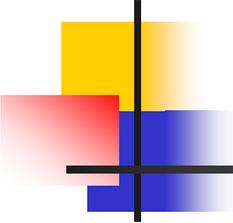
¿Por qué Pascal-FC?

- Todo ello hace que el alumno pueda empezar a escribir sus primeros programas casi inmediatamente después de la instalación
- Y que por ello nuestro esfuerzo se centre mas en el aprendizaje de las herramientas propias de la Programación Concurrente que en el lenguaje



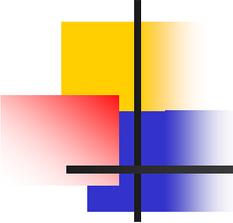
Pascal-FC

- Burns and Davies, 1993
- Está basado en un subconjunto del Lenguaje Pascal: el Pascal-S (Wirth, 1979)
- No trae algunas características que proporciona Pascal:
 - Ficheros
 - Punteros
 - Registros Variantes
 - No se puede usar el tipo "string" salvo en la instrucción "writeln" como una constante
 - No puede usarse "With"
 - No existe el tipo set
 - No existe el tipo rango



Pascal-FC

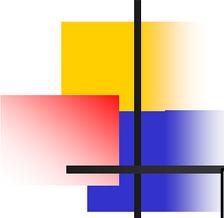
- Añadidos respecto al Pascal
 - bucle "repeat ...forever"
 - Sentencia "null"
 - La función "random"
 - $i := \text{random}(n)$: Devuelve un número aleatorio entre 0 y $\text{abs}(n)$



Estructura de un programa

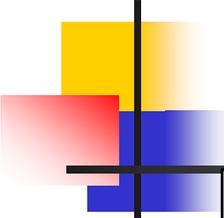
- **program** `identificador ;`
- (* Declaraciones globales: constantes, tipos, *)
- (* variables, procedimientos, funciones, *)
- (* tipos de procesos, procesos, monitores *)

- **begin**
- (* sentencias *)
- **end.**



Ejemplo 1

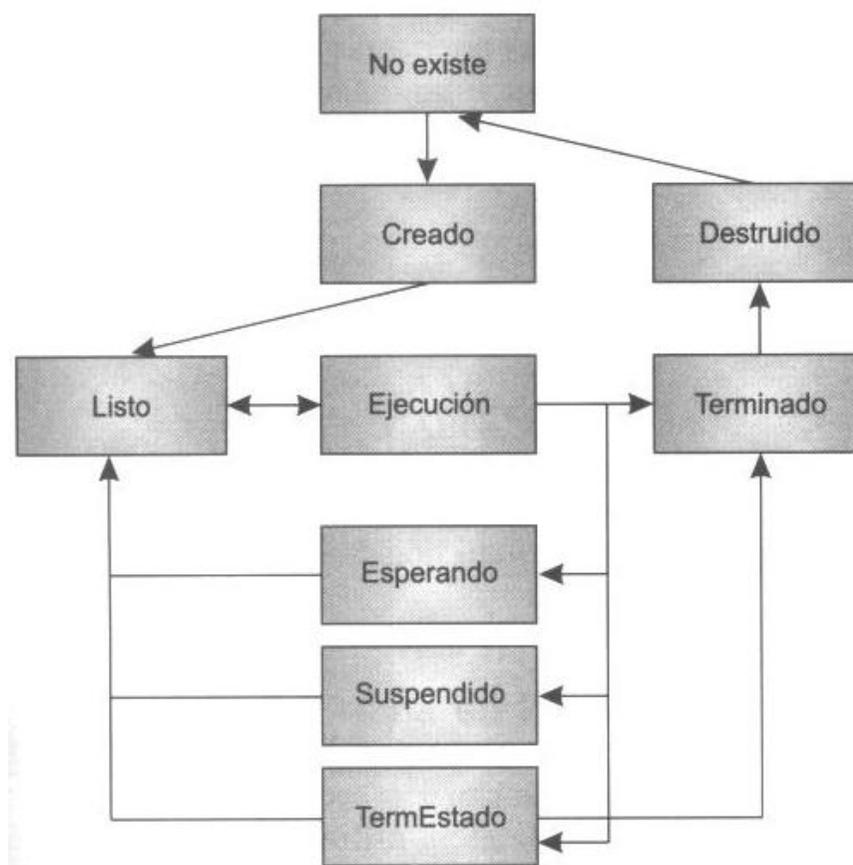
```
•program EJEMPLO;  
•  (* Programa de ejemplo que declara dos procesos que realizan una tarea trivial *)  
•process Primero;  
•var  
•  i: integer;  
•begin  
•  for i:=1 to 5 do  
•    writeln (1);  
•end;  
•process Segundo;  
•var  
•  i: integer;  
•begin  
•  for i:=1 to 5 do  
•    writeln (2);  
•end;  
•begin  
•  writeln ('Esto se ejecuta de forma secuencial');  
•  writeln ('y a continuación se ejecutarán los procesos de forma concurrente');  
•  cobegin  
•    Primero;  
•    Segundo;  
•  coend;  
•  writeln ('Esto se ejecutaría de forma secuencial después de terminar ambos procesos');  
•end.
```



Ejemplo II

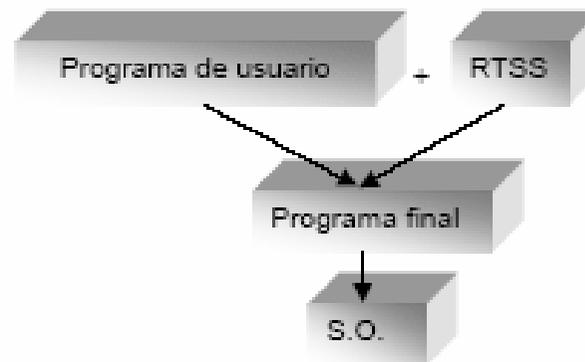
```
•program EJEMPLO;  
• (* Programa de ejemplo que declara dos procesos que realizan una tarea trivial *)  
•Process type Proceso (ident: integer)  
•var  
• i: integer;  
•begin  
• for i:=1 to 5 do  
•     writeln (ident);  
•end;  
•Var  
• Primero, Segundo: Proceso;  
•begin  
• writeln ('Esto se ejecuta de forma secuencial');  
• writeln ('y a continuación se ejecutarán los procesos de forma concurrente');  
• cobegin  
•     Primero(1);  
•     Segundo(2);  
• coend;  
• writeln ('Esto se ejecutaría de forma secuencial después de terminar ambos procesos');  
•end.
```

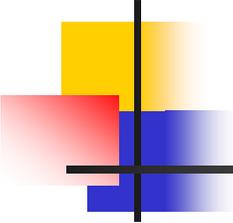
Estados de un proceso en Pascal-FC



Gestión de procesos en Pascal-FC

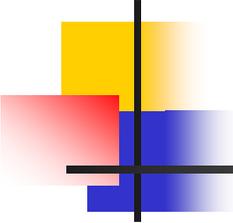
- Pascal-FC lenguaje didáctico pensado incluso para SO sin soporte de concurrencia
- Por ello el compilador al generar el ejecutable empaqueta código de usuario + planificador





Planificación de procesos

- Política injusta: ejecutar un proceso hasta que termine, luego otro, etc...
- Política justa: Compartir el tiempo del procesador dando rodajas de tiempo a los procesos



Ejecución de un programa en Pascal-FC

- Pascal-FC consiste en dos programas separados: un compilador (*pfccomp*) y un intérprete (*pint*, *ufpint*)
 - *pfccomp* FichFuente FichListado FichObjeto
 - *pint* FichObjeto FichProblemas
 - *ufpint* FichObjeto FichProblemas