

Bloque 1. Paradigmas de la Programación

Tarea 2: Establecer los paradigmas que cumplen y el nivel de cumplimiento para los siguientes lenguajes de programación:

- **Pascal:** Estructurado (soportado), Imperativo (obligatorio), Modular (obligatorio), Orientado a procedimientos (soportado).
- **Java:** Orientado a objetos (obligatorio), Imperativo (obligatorio), Estructurado (permitido), Concurrente (soportado), Modular (soportado).
- **Fortran:** Imperativo (obligatorio), Concurrente (soportado), Estructurado (permitido), Modular (soportado). **Orientado a objetos según versión.
- **C:** Imperativo (obligatorio), Estructurado (permitido), Concurrente (soportado), Modular (soportado).
- **Modula-2:** Estructurado (obligatorio), Imperativo (obligatorio), Modular (obligatorio), Orientado a procedimientos (soportado).
- **Basic:** Imperativo (obligatorio).
- **C++:** Orientada a objetos (soportado), Imperativo (obligatorio), Concurrente (soportado), Estructurado (permitido), Modular (soportado).
- **Ada:** Imperativo (obligatorio), Concurrente (permitido), Estructurado (permitido), Orientado a procedimientos (obligatorio), Modular (obligatorio), Orientado a objetos (permitido).
- **Caml:** Imperativo (obligatorio).
- **Cobol:** Imperativo (obligatorio). Orientado a negocios (obligatorio).
- **Lisp:** Declarativo - Funcional (obligatorio).
- **Prolog:** Declarativo - Lógico (obligatorio).
- **Parlog:** Declarativo - Lógico (obligatorio), Concurrente (obligatorio).
- **OCaml:** Imperativo (obligatorio), Orientado a objetos (obligatorio).
- **SmallTalk:** Orientada a objetos (obligatorio), Imperativo (obligatorio), Concurrente (permitido), Modular (soportado).
- **Algol:** Imperativo (obligatorio). Estructurado (obligatorio), Orientado a algoritmos (soportado).
- **Ensamblador:** Estructurado (obligatorio), Imperativo (obligatorio), Secuencial (obligatorio).
- **CLOS:** Orientado a objetos (obligatorio), Imperativo (obligatorio).
- **Python:** Orientada a objetos (soportado), Imperativo (obligatorio), Concurrente (permitido, importando el módulo módulo processing), Estructurado (permitido), Modular (soportado), Orientado a procedimientos (soportado).

Basado en el análisis de algunos de los siguientes códigos:

Pascal

http://sandbox.mc.edu/~bennet/cs404/doc/dayio_pas.html

http://sandbox.mc.edu/~bennet/cs404/doc/sched_pas.html

Fortran

<https://sites.esm.psu.edu/~ajm138/fortranexamples.html>

<https://www.absoft.com/resources/absoft-coding-examples/basic-fortran-overview-examples-information/>

Modula-2

<https://www.modula2.org/tutor/chapter15.php>

Ada

http://sandbox.mc.edu/~bennet/ada/examples/enumio_adb.html

http://sandbox.mc.edu/~bennet/ada/examples/compcount_ads.html

Cobol

<https://medium.com/@yvanscher/7-cobol-examples-with-explanations-ae1784b4d576>