

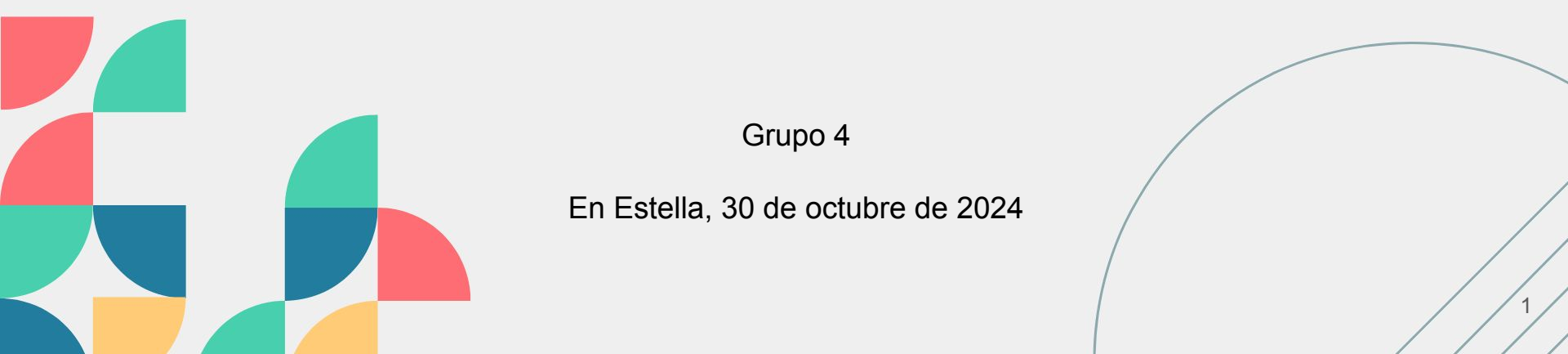
CI Politécnico Estella
Desarrollo de Aplicaciones Web y Multiplataforma
Entornos de desarrollo
Unidad de Trabajo 1

Desarrollo de software

Tarea grupal

Grupo 4

En Estella, 30 de octubre de 2024



Índice

Lenguajes de alto nivel

<u>Java.....</u>	Javier Rodrigo
<u>C#.....</u>	Mikel Viñuales Lasheras
<u>Ruby.....</u>	Jessica Trinidad
<u>PHP.....</u>	Miguel Ángel Ferri
<u>Haskell.....</u>	Adur Marques Herrero
<u>Kotlin.....</u>	Mikel Ramirez
<u>Python.....</u>	Eugen Moga

Lenguajes de menor distribucion

<u>F#.....</u>	Javier Rodrigo
<u>Crystal...</u>	Mikel Viñuales Lasheras
<u>Rust.....</u>	Jessica Trinidad
<u>Racket...</u>	Miguel Ángel Ferri
<u>C++.....</u>	Adur Marques Herrero
<u>Erlang...</u>	Mikel Ramirez
<u>OCaml...</u>	Eugen Moga

Javier Rodrigo Java

Nombre del Lenguaje	Java
Año de salida	1995
Inventores	James Gosling
Evolución de un lenguaje anterior	Java no evoluciona directamente de ningún lenguaje pero sí que tiene muchas características de otros lenguajes como c++
Última versión	La última versión es la 23 lanzada en septiembre de 2023
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto,- Uso: Backend- Compilado, interpretado- Paradigma: (POO(programación orientada a objetos))
Uso habitual	Propósitos principales (principalmente diseñado para crear aplicaciones, diseño web con spring boot, desarrollo móvil.)

Javier Rodrigo Java

Extensión de ficheros	<ul style="list-style-type: none">- Fuente (.java)- Compilado (.class,)- Ejecutable (.jar)
Nomenclatura	Convención PascalCase. Palabras comienzan por mayúscula y no se utilizan separadores.
Enlaces a manuales y documentación	https://docs.oracle.com/en/java/ https://www.oracle.com/java/technologies/javase-documentation.html
Editores IDE compatibles	Apache NetBeans,Eclipse,Vscode,intelliJ,idea,BlueJ....
Ventajas	Es un lenguaje de alto nivel, por lo que es más fácil de aprender que otros. Es multiplataforma,es decir permite ejecutar código en cualquier sistema operativo. Gran cantidad de bibliotecas y documentación.
Desventajas	Uso elevado de memoria No es adecuado para el uso de interfaces gráficas Es de lenguaje interpretado por lo que disminuye el rendimiento de ejecución de los programas.
Opcional	Ideal para crear software empresarial debido a su robustez, seguridad y capacidad para manejar grandes volúmenes de datos.

Nombre del Lenguaje	F#
Año de salida	2005
Inventores	Don Syme
Evolución de un lenguaje anterior	Es un lenguaje de programación funcional que deriva de varios lenguajes como OCaml y ML.
Última versión	La versión más reciente de F# es F# 8.0, lanzada en noviembre de 2023
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto,- Uso: ,Backend- Compilado,- Paradigma: (Funcional,imperativo y orientado a objetos)
Uso habitual	Propósitos principales (Desarrollo de servicios de backend,API,s y procesamiento de datos)

Extensión de ficheros	<ul style="list-style-type: none"> - Fuente (.fs, .fsi,) - Compilado (.dll,) - Ejecutable (.exe,)
Nomenclatura	Las clases y módulos suelen escribirse en PascalCase y las funciones y variables en camelCase.
Enlaces a manuales y documentación	https://learn.microsoft.com/en-us/dotnet/fsharp/
Editores IDE compatibles	Visual Studio: IDE recomendado para F# con soporte completo de .NET y depuración.
Ventajas	<ul style="list-style-type: none"> - sintaxis clara y concisa - alto rendimiento, es adecuado para cálculos complejos - Perfecto para la programación funcional.
Desventajas	<ul style="list-style-type: none"> - Menor popularidad, esto es negativo porque disminuye la comunidad de soporte - Pocos recursos de aprendizaje
Opcional	# es una buena opción para combinar paradigmas funcionales con la plataforma .NET, en especial para cálculos intensivos.

C# Mikel Viñuales Lasheras



Nombre del Lenguaje	C# (“Cii Sharp”)
Año de salida	2000
Inventores	Anders Hejlsberg (junto con un equipo)
Evolución de un lenguaje anterior	Sintaxis básica y conceptos derivan de C/C++ El modelo de orientación a objetos inspirado en JAVA
Última versión	C# 13 (lanzada en año 2024)
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto- Uso: Backend, con excepciones se usa como Frontend- Compilado- Paradigma: Multiparadigma (estructurada, POO, funcional, imperativa, declarativa...)
Uso habitual	Desarrollo de aplicaciones web, aplicaciones windows, desarrollo de videojuegos, desarrollo de aplicaciones móviles para iOS, Android y Windows (Xamarin), desarrollo de aplicaciones empresariales (integración con .NET).



Extensión de ficheros	- Fuente .cs Si se compilan los proyectos de aplicaciones ejecutables .exe
Nomenclatura	- Por convención PascalCase . - Variables locales y parámetros de los métodos camelCase - “_” para campos privados
Enlaces a manuales y documentación	https://learn.microsoft.com/es-es/dotnet/csharp/ (oficial) https://docs.godotengine.org/en/stable/tutorials/scripting/c_sharp/index.html
Editores IDE compatibles	Visual Studio es el IDE por excelencia. Visual Studio Code, MonoDevelop, Rider (JetBrains), sonarlint, CodeMaid
Ventajas	Fácil curva de aprendizaje, seguridad (seguridad integradas, desarrollo rápido (reducción errores y bugs), soporte multiplataforma, comunidad sólida.
Desventajas	Limitado a plataformas de Microsoft, rendimiento inferior a lenguajes nativos, costo adicional en el tiempo de ejecución, menos adecuado en sistemas de baja potencia.
Opcional	C# es la nota Do sostenido. Así, se escenifica que es superior a C/C++. Además, # se forma por 4+, escenificando también esa superioridad (C++++).

Nombre del Lenguaje	Crystal
Año de salida	2014 (primera versión oficial)
Inventores	Ary Borenszweig, Juan Wajnerman, Brian Cardiff y más de 300 colaboradores
Evolución de un lenguaje anterior	Su sintaxis está inspirada en Ruby pero no busca ser compatible.
Última versión	1.14.0 (9 de octubre de 2024)
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto- Uso: Backend- Compilado- Paradigma: Multiparadigma (POO, imperativo, funcional)
Uso habitual	Mayoritariamente desarrollo de aplicaciones web. Desarrollo videojuegos, desarrollo de software.

Extensión de ficheros	- Fuente .cr
Nomenclatura	-Clases y módulos PascalCase - Nombres variables y métodos snake_case . Constantes en mayúsculas.
Enlaces a manuales y documentación	https://crystal-lang.org/ (oficial) https://crystal-lang.org/api/1.14.0/ (API) https://github.com/crystal-lang/crystal/ (codigo fuente)
Editores IDE compatibles	Visual Studio Code, Atom. Hay más, pero estos son los más utilizados por la comunidad.
Ventajas	Sintaxis sencilla, tipado estático con inferencia, fácil integración con C, generación de código nativo eficiente, rapidez, evaluación y generación de código en tiempo de compilación (vía macros).
Desventajas	No es muy utilizado; por lo que una comunidad pequeña. No es tan rápido como se esperaba (tan rápido como C).
Opcional	Crystal es un lenguaje compilado y su primer compilador fue desarrollado en Ruby si bien a posteriori en 2013 se reescribió en Crystal dotando así a Crystal de ser un lenguaje self-hosting.



Nombre del Lenguaje	Ruby
Año de salida	1995
Inventores	Yukihiro Matsumoto
Evolución de un lenguaje anterior	Ruby no es la evolución de un lenguaje anterior, sino una combinación de características de múltiples lenguajes (Perl, Smalltalk, Lisp, Eiffel y Ada).
Última versión	Ruby 3.3 (25 de diciembre de 2023).
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto- Uso: Backend- Interpretado- Paradigma: Orientado a objetos (OO) y reflexivo.
Uso habitual	Ruby está principalmente vinculado con el desarrollo web, aunque gracias a su versatilidad se utiliza en tareas como automatización, análisis de datos, desarrollo de juegos y prototipado rápido.

Ruby



Extensión de ficheros	<ul style="list-style-type: none">- Fuente: extensión .rb- Compilado: no hay archivos compilados porque Ruby es interpretado.- Ejecutable: no genera archivos ejecutables por sí mismo. Puede ser empaquetado en .exe (Windows) utilizando herramientas externas.
Nomenclatura	Clases en CamelCase, métodos y variables en snake_case y constantes en UPPER_SNAKE_CASE
Enlaces a manuales y documentación	https://www.ruby-lang.org/es/documentation/ https://www.ruby-lang.org/es/documentation/quickstart/
Editores IDE compatibles	RubyMine, VSCode, Sublime Text, Aptana Studio, IDE Komodo, Atom, AWS Cloud9, Selenium IDE, NetBeans.
Ventajas	Sintaxis elegante y fácil de entender, promueve la reutilización del código. Su metaprogramación permite modificar el comportamiento del código en tiempo de ejecución.
Desventajas	Rendimiento inferior a C++ o Java, mayor consumo de recursos de memoria y CPU y la documentación es inconsistente o desactualizada.
Opcional	Facilitar la creación de DSLs (Domain-Specific Languages) para adaptarse a necesidades específicas de proyectos.



Nombre del Lenguaje	Rust
Año de salida	2010
Inventores	Graydon Hoare
Evolución de un lenguaje anterior	No es una evolución de otro lenguaje, aunque está inspirado en C y C ++. También está influenciado por Haskell, ML y OCaml.
Última versión	1.82.0 (17 de octubre de 2024).
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Bajo nivel aunque con características de alto nivel.- Uso: Backend- Compilado.- Paradigma: multiparadigma (funcional, por procedimientos, imperativa y orientada a objetos.
Uso habitual	Desarrollo de videojuegos, desarrollo web, sistemas embebidos y aplicaciones científicas.



Extensión de ficheros	<ul style="list-style-type: none">- Fuente: extensión .rs- Compilado: compila el código directamente a ejecutables nativos.- Ejecutable: .exe (Windows).
Nomenclatura	Structs, traits, Type Aliases y enums utilizan CamelCase, las variables, funciones y módulos en snake_case. Las constantes y estáticas en UPPER_SNAKE_CASE
Enlaces a manuales y documentación	https://prev.rust-lang.org/es-ES/documentation.html https://www.rust-lang.org/es/learn
Editores IDE compatibles	VSCode, IntelliJ IDEA, CLion, Atom, Sublime Text, Eclipse, Vim, Theia, JetBrains.
Ventajas	Seguridad de memoria sin garbage collector, ecosistema en crecimiento, rendimiento comparable al de C/C++ y concurrencia eficiente.
Desventajas	El tiempo de compilación de Rust es más lento que el de otros lenguajes, no es adecuado para principiantes por su curva de aprendizaje, recursos y soportes limitados.
Opcional	Rust facilita el desarrollo de aplicaciones concurrentes y escalables.

Nombre del Lenguaje	PHP (Hypertext Preprocessor)
Año de salida	Creado en 1994
Inventores	Rasmus Lerdorf
Evolución de un lenguaje anterior	PHP evolucionó gradualmente
Última versión	8.3.12 (26 de septiembre de 2024)
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto- Uso: Abierto- Interpretado
Uso habitual	Se utiliza principalmente para generar contenido dinámico en el servidor, como páginas web, formularios y manejo de bases de datos. Es ideal para tareas que requieren acceso a bases de datos y procesamiento en el servidor antes de enviar contenido al navegador.

Extensión de ficheros	<ul style="list-style-type: none">- Fuente (.php)- Compilado (.dll)- Ejecutable (.Phar)
Nomenclatura	Se utiliza el símbolo del dólar en el primer carácter del nombre, se diferencian las mayúsculas.
Enlaces a manuales y documentación	https://www.php.net/manual/es/index.php https://desarrolloweb.com/manuales/manual-php.html
Editores IDE compatibles	Netbeans, PHPStorm, Eclipse, Sublime Text 3, Atom
Ventajas	<p>Es un lenguaje de código abierto</p> <p>Permite automatizar tareas</p> <p>Soporta gran cantidad de datos</p> <p>Se adapta al lenguaje HTML</p> <p>Es fácil para instalarse en los servidores o sistemas operativos</p>
Desventajas	<p>Seguridad baja</p> <p>Se necesita un servidor web</p> <p>Mala calidad de manejo de errores</p>

Nombre del Lenguaje	Racket
Año de salida	En 1994.
Inventores	Matthias Felleisen y PLT Inc.
Evolución de un lenguaje anterior	Evolución.
Última versión	Versión 8.14 (Agosto 2024)
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Bajo- Uso: Backend- Entorno de desarrollo integrado- Multiparadigma- Software libre
Uso habitual	Desarrollo web

Extensión de ficheros	. rkt .ss .scm .sch
Nomenclatura	PLT
Enlaces a manuales y documentación	https://docs.racket-lang.org/ https://docs.racket-lang.org/guide/
Editores IDE compatibles	Emacs
Ventajas	Potente sistema de macros, acceso a librerías, capaz de crear y extender nuevos lenguajes
Desventajas	Curva de aprendizaje muy pronunciada, velocidad lenta
Opcional	Posibilitar la creación de nuevos lenguajes o dialectos



Nombre del Lenguaje	Haskell
Año de salida	1997 fecha de salida de la versión estable
Inventores	Simon Peyton Jones, Philip Wadler y otros del comité de programadores
Evolución de un lenguaje anterior	Haskell se basa en ideas de lenguajes funcionales como Miranda y Lisp
Última versión	Haskell 2010
Tipo de lenguaje	Nivel: Alto Uso: Backend, análisis de datos, procesamiento de texto Compilado: Utiliza GHC (Glasgow Haskell Compiler) Paradigma: Funcional puro (no soporta programación imperativa u orientada a objetos de forma nativa)
Uso habitual	Desarrollo de software académico, programación de algoritmos complejos, sistemas de procesamiento de datos

Haskell



Extensión de ficheros	.hs (código fuente) .lhs (código fuente literario)
Nomenclatura	Haskell lleva el nombre de Haskell Brooks Curry, un matemático y lógico estadounidense conocido por su trabajo en la teoría de tipos y la lógica combinatoria
Enlaces a manuales y documentación	Haskell Official Website Learn You a Haskell for Great Good!
Editores IDE compatibles	Visual Studio Code (con extensiones Haskell) IntelliJ IDEA (con el plugin Haskell) Eclipse (con el plugin Haskell)
Ventajas	Soporte nativo para programación funcional o declarativa. Sistema de tipos fuerte y estático ideal para detectar errores en tiempo de compilación. Funciones de orden superior que permiten abstraerse y reutilizar código. Excelente control de concurrencia y paralelismo.
Desventajas	Curva de aprendizaje elevada para programadores que programan de forma imperativa. Menos popular en comparación con lenguajes más utilizados como Java y C#. Menor cantidad de bibliotecas y soporte en comparación con otros lenguajes más populares.
Otros	Un compañero del trabajo, amante de las matemáticas me recomendó aprender este idioma para entender LINQ de C#, al poco tiempo de empezar, dejé de aprenderlo.



Adur Marques Herrero

Nombre del Lenguaje	C++
Año de salida	1985
Inventores	Bjarne Stroustrup
Evolución de un lenguaje anterior	C++ es una extensión del lenguaje de programación C, añadiendo características de programación orientada a objetos (POO)
Última versión	C++23 (publicada en febrero del 2023)
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto pero puedes gestionar la memoria- Uso: Backend- Compilado- Paradigma: Multiparadigma y orientado a objetos
Uso habitual	Desarrollo de software de sistemas, aplicaciones de alto rendimiento, videojuegos, control de hardware, aplicaciones en tiempo real
Otros	Un compañero del trabajo me habló muy bien de este lenguaje ya que lo utilizó para desarrollar un programa para controlar los focos del teatro de Pamplona.



Extensión de ficheros	.cpp (fuentes) .h (encabezados), .hpp (encabezados de C++)
Nomenclatura	C++ se denomina así por el operador de incremento (++), que refleja la evolución del lenguaje más allá de C.
Enlaces a manuales y documentación	C++ Reference cplusplus.com ISO C++
Editores IDE compatibles	Microsoft Visual Studio (el mejor para desarrollar c++), Eclipse, Visual Studio Code con extensiones
Ventajas	Alta eficiencia y rendimiento. Soporte para programación orientada a objetos. Portabilidad en múltiples plataformas. Gran cantidad de bibliotecas y frameworks disponibles.
Desventajas	Complejidad en el aprendizaje, especialmente para principiantes. Manejo de memoria manual, lo que puede provocar fugas de memoria si no se gestiona correctamente. Compilación más lenta en comparación con algunos lenguajes interpretados.

Nombre del Lenguaje	Kotlin
Año de salida	2016
Inventores	JetBrains. - Dmitry Jemerov (Lider de la empresa JetBrains)
Evolución de un lenguaje anterior	No es una evolución, pero se inspira en: Java, Scala, Groovy, C#, Gosu
Última versión	V 2.0.21 (Octubre de 2024)
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto,- Uso: Frontend y Backend- Compilado- Paradigma: Multiparadigma: orientado a objetos, funcional, imperativo, estructurado en bloques, declarativo, genérico, reflexivo, concurrente
Uso habitual	Cualquier tipo de desarrollo, desde la web del lado del servidor y del lado del cliente, hasta Android e iOS.

Información obtenida de: [https://es.wikipedia.org/wiki/Kotlin_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Kotlin_(lenguaje_de_programaci%C3%B3n))

<https://keepcoding.io/blog/ventajas-y-desventajas-de-kotlin/>

Y google respuestas en general.

Extensión de ficheros	kt, .kts, .ktm
Nomenclatura	Sigue las convenciones de JAVA
Enlaces a manuales y documentación	https://kotlinlang.org/docs/home.html
Editores IDE compatibles	IntelliJ IDEA y JetBrains. Empresa creadora de Kotlin
Ventajas	Kotlin soluciona el error “null” . Tiene una buena curva de aprendizaje. Integrado con Android Studio. Corrutinas que optimizan la programación asíncrona. Comunidad muy comprometida con Open Source, mucha documentación y código libre.
Desventajas	No tiene buen soporte para funciones Lambda. (No lo entiendo muy bien, pero la “anulabilidad de tipos” que soluciona el problema Null, parece generar otro tipo de problemas. Se debe realizar el binding manualmente.

Información obtenida de: [https://es.wikipedia.org/wiki/Kotlin_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Kotlin_(lenguaje_de_programaci%C3%B3n))

<https://keepcoding.io/blog/ventajas-y-desventajas-de-kotlin/>

Y google respuestas en general.

Nombre del Lenguaje	Erlang
Año de salida	1986
Inventores	Joe Armstrong Robert & Virding Mike Williams
Evolución de un lenguaje anterior	No, desarrollado enteramente para programas concurrentes que se ejecuten eternamente.
Última versión	Erlang/OTP 27.1 18 de Septiembre de 2024
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Bajo- Uso: Backend- Interpretado- Paradigma: multiparadigma: funcional, concurrente
Uso habitual	Programas concurrentes que se ejecuten eternamente.

Información obtenida de: <https://es.wikipedia.org/wiki/Erlang>
<https://recluit.com/descubre-la-historia-de-erlang/>
Y google respuestas en general.

Extensión de ficheros	.erl .hrl
Nomenclatura	*El 65,9% del código de Erlang está escrito en Erlang. El 13,4% de Erlang está escrito en XML. El 13,6% de Erlang está escrito en C. El 1,4% de Erlang está escrito en C++
Enlaces a manuales y documentación	https://www.erlang.org/docs
Editores IDE compatibles	La máquina virtual BEAM y HiPE
Ventajas	Concurrente ; procesos extremadamente livianos cuyos requerimientos de memoria varían dinámicamente. Distribuido : está diseñado para correr en ambientes distribuidos. Actualización de código "en caliente" Puede actualizarse sin detenerse.
Desventajas	Algunos algoritmos numéricos no se traducen bien. El uso de tipa dinámico puede ralentizar el programa. No funciona bien con interfaces gráficas
Opcional	



Nombre del Lenguaje	Python
Año de salida	1991
Inventores	Guido van Rossum
Evolución de un lenguaje anterior	Inspirado en ABC, Modula-3, Smalltalk
Última versión	3.12.0 Octubre 2023
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto- Uso: Backend, Frontend- Interpretado- Paradigma: Multiparadigm, Orientado a objetos, Funcional, Imperativo, etc.)
Uso habitual	Desarrollo web, ciencia de datos, inteligencia artificial, scripts, aplicaciones de escritorio, automatización, API



Extensión de ficheros	<ul style="list-style-type: none">- Fuente .py- Compilado .pyc- Ejecutable: No genera directamente puede ser empaquetado .exe .app
Nomenclatura	Convenciones PEP8, Clases en mayúsculas Variables y funciones en minúsculas con guiones bajos
Enlaces a manuales y documentación	Documentación oficial Python.org
Editores IDE compatibles	VSCode, PyCharm, Sublime Text, Jupyter Notebooks, Atom, Spyder, IDLE
Ventajas	Sintaxis simple y legible, Amplia comunidad de soporte, Gran cantidad de de bibliotecas para ciencia de datos, desarrollo web
Desventajas	Es más lento que los lenguajes compilados, no es ideal para aplicaciones en tiempo real
Opcional	Lenguaje fácil de aprender, con una sintaxis limpia y clara

Nombre del Lenguaje	OCaml
Año de salida	1996
Inventores	Xavier Leroy
Evolución de un lenguaje anterior	Basado en Caml que a su vez deriva de ML(MetaLanguage)
Última versión	5.1.0 Febrero de 2023
Tipo de lenguaje	<ul style="list-style-type: none">- Nivel: Alto- Uso: Backend- Compilado- Paradigma: Funcional, Imperativo, orientado a objetos
Uso habitual	Investigación académica, herramientas de verificación formal, compiladores, análisis estático, programación de sistemas

Extensión de ficheros	<ul style="list-style-type: none">- Fuente: .ml, .mli- Compilado: .cmo (bytecode), .cmx(nativo)- Ejecutable: .exe(Windows), archivos binarios en Linux y macOS
Nomenclatura	Convenciones en “CamelCase” para tipos y módulos, y en “snake_case” para funciones y variables locales
Enlaces a manuales y documentación	OCaml.org
Editores IDE compatibles	OCaml Platform IDE, VSCode, Emacs, Vim, Sublime Text, JetBrains
Ventajas	Capacidad de compilación tanto a bytecode como a código nativo, Alto rendimiento en ejecución nativa
Desventajas	Curva de aprendizaje más pronunciada, especialmente para principiantes Comunidad más pequeña comparada con lenguajes más populares Ecosistema de librerías más limitado que lenguajes como Python o JavaScript