**INFORMACION RUBY PRIMERA PRACTICA:**

**Parrafos:**

**Ruby** es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador [japonés](https://es.wikipedia.org/wiki/Jap%C3%B3n) [Yukihiro "Matz" Matsumoto](https://es.wikipedia.org/wiki/Yukihiro_Matsumoto" \o "Yukihiro Matsumoto), quien comenzó a trabajar en Ruby en [1993](https://es.wikipedia.org/wiki/1993), y lo presentó públicamente en [1995](https://es.wikipedia.org/wiki/1995). Combina una sintaxis inspirada en [Python](https://es.wikipedia.org/wiki/Python) y [Perl](https://es.wikipedia.org/wiki/Perl) con características de programación orientada a objetos similares a [Smalltalk](https://es.wikipedia.org/wiki/Smalltalk" \o "Smalltalk). Comparte también funcionalidad con otros lenguajes de programación como [Lisp](https://es.wikipedia.org/wiki/Lisp" \o "Lisp), [Lua](https://es.wikipedia.org/wiki/Lua" \o "Lua), [Dylan](https://es.wikipedia.org/wiki/Dylan_(lenguaje_de_programaci%C3%B3n)) y [CLU](https://es.wikipedia.org/wiki/CLU). Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una [licencia](https://es.wikipedia.org/wiki/Ruby#Licencia) de [software libre](https://es.wikipedia.org/wiki/Software_libre).

El lenguaje fue creado por [Yukihiro "Matz" Matsumoto](https://es.wikipedia.org/wiki/Yukihiro_Matsumoto" \o "Yukihiro Matsumoto), quien empezó a trabajar en Ruby el [24 de febrero](https://es.wikipedia.org/wiki/24_de_febrero) de [1993](https://es.wikipedia.org/wiki/1993), y lo presentó al público en el año [1995](https://es.wikipedia.org/wiki/1995). En el círculo de amigos de Matsumoto se le puso el nombre de Ruby (en español *rubí*) como broma aludiendo al lenguaje de programación Perl (*perla* en inglés). La última versión estable de la rama 1.8 es la 1.8.7\_p248, de la rama 1.9 es la 1.9.2\_p180 . La versión en 1.9 que incorpora mejoras sustanciales en el rendimiento del lenguaje, que se espera queden reflejadas en la próxima versión estable de producción del lenguaje, Ruby 1.9.0.1 Diferencias en rendimiento entre la actual implementación de Ruby (1.8.6) y otros lenguajes de programación más arraigados han llevado al desarrollo de varias máquinas virtuales para Ruby. Entre esas se encuentra [JRuby](https://es.wikipedia.org/wiki/JRuby" \o "JRuby), un intento de llevar Ruby a la plataforma [Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java), y [Rubinius](https://es.wikipedia.org/wiki/Rubinius" \o "Rubinius), un [intérprete](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)) modelado basado en las máquinas virtuales de [Smalltalk](https://es.wikipedia.org/wiki/Smalltalk" \o "Smalltalk). Los principales desarrolladores han apoyado la máquina virtual proporcionada por el proyecto [YARV](https://es.wikipedia.org/w/index.php?title=YARV&action=edit&redlink=1), que se fusionó en el árbol de código fuente de Ruby el [31 de diciembre](https://es.wikipedia.org/wiki/31_de_diciembre) de [2006](https://es.wikipedia.org/wiki/2006), y se dio a conocer como Ruby 1.9.

Ruby es [orientado a objetos](https://es.wikipedia.org/wiki/Orientado_a_objetos): todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivas, (como [enteros](https://es.wikipedia.org/wiki/Entero), booleanos, y *nil*). Toda [función](https://es.wikipedia.org/wiki/Funci%C3%B3n_(programaci%C3%B3n)) es un [método](https://es.wikipedia.org/wiki/M%C3%A9todo_(programaci%C3%B3n_orientada_a_objetos)). Las variables siempre son referencias a objetos, no los objetos mismos. Ruby soporta [herencia](https://es.wikipedia.org/wiki/Herencia_(programaci%C3%B3n_orientada_a_objetos)) con [enlace dinámico](https://es.wikipedia.org/wiki/Enlace_din%C3%A1mico), *[mixins](https://es.wikipedia.org/wiki/Mixin" \o "Mixin)* y [métodos *singleton*](https://es.wikipedia.org/wiki/Singleton) (pertenecientes y definidos por una sola [instancia](https://es.wikipedia.org/wiki/Instancia_(programaci%C3%B3n)) más que definidos por la clase). A pesar de que Ruby no soporta [herencia múltiple](https://es.wikipedia.org/wiki/Herencia_m%C3%BAltiple), la clases pueden importar [módulos](https://es.wikipedia.org/wiki/M%C3%B3dulo_(programaci%C3%B3n)) como *mixins*. Es compatible con la sintaxis procedimental, pero todos los métodos definidos fuera del ámbito de un objeto son realmente métodos de la clase Object. Como esta clase es padre de todas las demás, los cambios son visibles para todas las clases y objetos.

Ruby ha sido descrito como un [lenguaje de programación multiparadigma](https://es.wikipedia.org/w/index.php?title=Lenguaje_de_programaci%C3%B3n_multiparadigma&action=edit&redlink=1): permite [programación procedimental](https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_procedimientos) (definiendo funciones y variables fuera de las clases haciéndolas parte del objeto raíz Object), con orientación a objetos, (todo es un objeto) o [funcionalmente](https://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional) (tiene funciones anónimas, [clausuras](https://es.wikipedia.org/wiki/Clausura_(inform%C3%A1tica)) y [continuaciones](https://es.wikipedia.org/w/index.php?title=Continuaciones&action=edit&redlink=1); todas las sentencias tiene valores, y las funciones devuelven la última evaluación). Soporta [introspección](https://es.wikipedia.org/wiki/Introspecci%C3%B3n_de_tipos), [reflexión](https://es.wikipedia.org/wiki/Reflexi%C3%B3n_(inform%C3%A1tica)) y [metaprogramación](https://es.wikipedia.org/wiki/Metaprogramaci%C3%B3n" \o "Metaprogramación), además de soporte para [hilos de ejecución](https://es.wikipedia.org/wiki/Hilo_de_ejecuci%C3%B3n) gestionados por el intérprete. Ruby tiene [tipado](https://es.wikipedia.org/wiki/Tipo_de_dato" \o "Tipo de dato) dinámico, y soporta [polimorfismo](https://es.wikipedia.org/wiki/Polimorfismo_(programaci%C3%B3n_orientada_a_objetos)) de tipos (permite tratar a subclases utilizando la interfaz de la clase padre). Ruby no requiere de [polimorfismo](https://es.wikipedia.org/wiki/Polimorfismo_(programaci%C3%B3n_orientada_a_objetos)) de funciones (sobrecarga de funciones) al ser dinámicamente tipado (los parámetros de una función pueden ser de distinto tipo en cada llamada, encargándose dicha función de determinar el tipo y actuar en consecuencia).

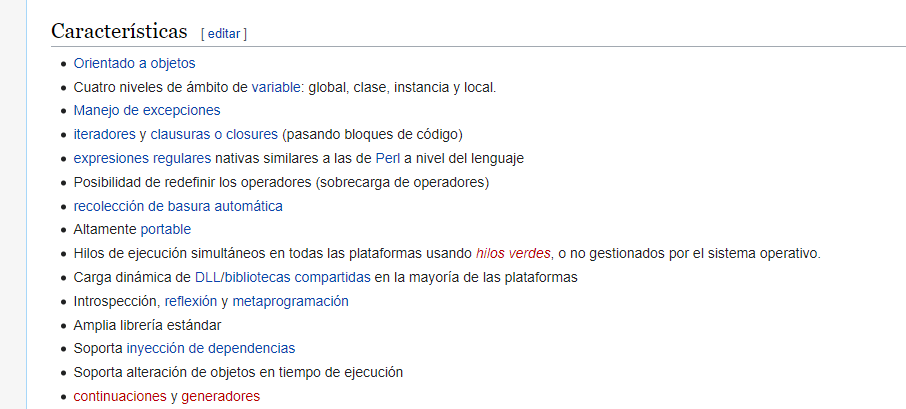
De acuerdo con las preguntas frecuentes de Ruby,[5](https://es.wikipedia.org/wiki/Ruby#cite_note-5)​ "Si te gusta [Perl](https://es.wikipedia.org/wiki/Perl), te gustará Ruby y su sintaxis. Si te gusta [Smalltalk](https://es.wikipedia.org/wiki/Smalltalk" \o "Smalltalk), te gustará Ruby y su semántica. Si te gusta [Python](https://es.wikipedia.org/wiki/Python), la enorme diferencia de diseño entre Python y Ruby/Perl puede que te convenza o puede que no".[6](https://es.wikipedia.org/wiki/Ruby#cite_note-6)​

El creador del lenguaje, [Yukihiro "Matz" Matsumoto](https://es.wikipedia.org/wiki/Yukihiro_Matsumoto" \o "Yukihiro Matsumoto), ha dicho que Ruby está diseñado para la productividad y la diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario.[2](https://es.wikipedia.org/wiki/Ruby#cite_note-2)​ Sostiene que el diseño de sistemas necesita enfatizar las necesidades humanas más que las de la máquina:[3](https://es.wikipedia.org/wiki/Ruby#cite_note-artima-3)​

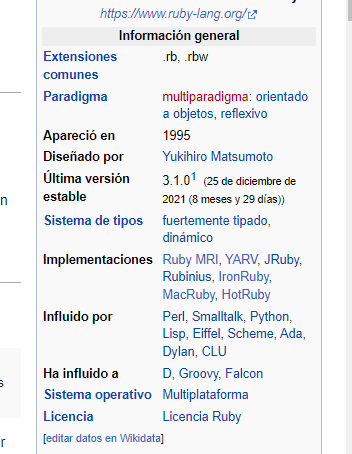
Ruby sigue el [*principio de la menor sorpresa*](https://es.wikipedia.org/wiki/Principio_de_la_m%C3%ADnima_sorpresa), lo que significa que el lenguaje debe comportarse de tal manera que minimice la confusión de los usuarios experimentados. Matz ha dicho que su principal objetivo era hacer un lenguaje que le divirtiera a él mismo, minimizando el trabajo de programación y la posible confusión. Él ha dicho que no ha aplicado el principio de menor sorpresa al diseño de Ruby,[3](https://es.wikipedia.org/wiki/Ruby#cite_note-artima-3)​ sin embargo, la frase se ha asociado al lenguaje de programación Ruby. La frase en sí misma ha sido fuente de controversia, ya que los no iniciados pueden tomarla como que las características de Ruby intentan ser similares a las características de otros lenguajes conocidos. En mayo de 2005 en una discusión en el grupo de noticias comp.lang.ruby, Matz trató de distanciar Ruby de la mencionada filosofía, explicando que cualquier elección de diseño será sorprendente para alguien, y que él usa un estándar personal de evaluación de la sorpresa. Si ese estándar personal se mantiene consistente habrá pocas sorpresas para aquellos familiarizados con el estándar.[4](https://es.wikipedia.org/wiki/Ruby#cite_note-4)​

Matz lo definió de esta manera en una entrevista:[3](https://es.wikipedia.org/wiki/Ruby#cite_note-artima-3)​

**Lista:**



**Tabla:**



**Cuestionario:**

Preguntas sobre la información que hay en los html

**AUDIO:**

Canción de chill

**Video:**

**Uno corto que explique ruby rapidamente**

**Imágenes:**

Foto del logo e información del creador