Esteban Valencia Bejarano  
Jorge Luis Ocampo Ocampo  
Santiago Vallejo Botero

**Actividad 1.**

Actividad 1.

Esteban Valencia Bejarano

Jorge Luis Ocampo Ocampo

Santiago Vallejo Botero



Jhan Carlos Martínez Ceballos

Universidad del Quindío

Facultad Ingeniería

Programa de Ingeniería en Sistemas y Computación

Programación 3

Armenia, Quindío, 2025

**1.** Elixir se ejecuta sobre Erlang. ¿Qué es Erlang y qué características tiene?

Erlang es un lenguaje de programación funcional Open Source, que fue creado a finales de los años 80, diseñado con el objetivo de soportar sistemas de telecomunicaciones.

Erlang en la actualidad es utilizado por grandes empresas como Amazon, Facebook, Whatsapp, Motorola, Yahoo! entre muchas otras.

La razón por la que se utiliza en estas aplicaciones proviene de la principal fortaleza de Erlang: la gestión de la disponibilidad de las aplicaciones.

Erlang se creó desde cero para el entorno de las telecomunicaciones, que exige que los sistemas cumplan con una disponibilidad mínima de 5x9 (99,999 % de tiempo de actividad anual). Esta cifra no deja mucho margen para el tiempo de inactividad durante un año. Por esta razón principalmente, Erlang incluye las siguientes características:

* Escalabilidad horizontal: capacidad para distribuir fácilmente los trabajos entre máquinas mediante comunicaciones fluidas dentro de una misma máquina y entre otras máquinas.
* Escalabilidad vertical: capacidad para distribuir los trabajos entre los recursos de procesamiento de la misma máquina.
* Intercambio de código en caliente: capacidad para actualizar/mejorar el código en tiempo real durante las operaciones.
* Asíncrono: el mundo real es asíncrono, por lo que Erlang se diseñó teniendo en cuenta esta característica básica. Una característica que contribuye a este requisito: los procesos “libres” de Erlang (se pueden ejecutar más de 32 000 simultáneamente).
* Supervisión: muchas estrategias diferentes para la supervisión de procesos con estrategias de reinicio, umbrales, etc. Ayuda a recuperarse más fácilmente de casos extremos/sobrecargas, al tiempo que mantiene rastros de los problemas para su posterior resolución, análisis post mortem, etc.
* Depuración en vivo: la capacidad de “iniciar sesión” en nodos en vivo a voluntad facilita las actividades de resolución de problemas. La depuración se puede realizar en vivo con acceso completo al estado de ejecución de cualquier proceso. Además, las herramientas de informe de errores integradas son muy útiles.

**2.** ¿Qué ventajas tiene usar Elixir en lugar de Erlang?



**3.** Entonces, ¿Por qué deberías aprender Elixir?

* **Concurrencia masiva.**

Este tiene la capacidad de realizar gran cantidad de procesos los cuales son unidades de ejecución ligeras y aisladas los cuales se ejecutan simultáneamente sin presentar errores que frenen el funcionamiento del programa, Elixir corre sobre una máquina virtual (Beam), creada por Erlang la cual permite realizar muchos micro procesos simultáneamente que a su vez consumen poca memoria, también, al ser un sistema de procesos aislados, si alguno presenta alguna falla este no genera errores en cadena.

* **Inmutabilidad.**

Los datos no se mutan(modifican), si no que estos se transforman en un nuevo conjunto de datos como, por ejemplo, a la variable x le asignamos un valor de 2 y después necesitamos que el valor asignado a x sea 4 y no dos lo que se hace en este lenguaje es direccionar a x a un nuevo valor que seria 4 sin modificar el 2 al que se había asignado con anterioridad. Al eliminar esta capacidad de modificar los datos el los programas funcionales suelen presentar menos errores mostrando un beneficio para el soporte de la concurrencia y estabilidad del sistema.

* **Enfoque en funciones.**

Los problemas se descomponen en funciones simples sin efectos secundarios llamadas “funciones puras”, estas producen el mismo resultado esperado para una entrada dada

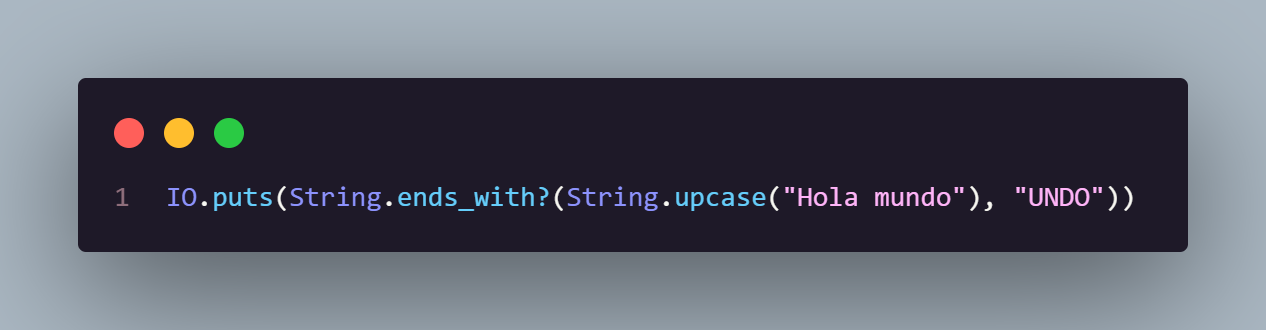
* **Fácil de testear.**

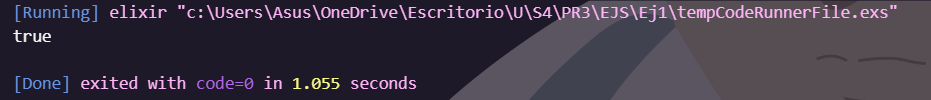
Por la misma naturaleza del lenguaje funcional como elixir que se enfatiza en pequeñas funciones sin cambios secundarios hace que testear sea más fácil.

**4**. ¿En qué tipo de proyecto es ideal usar elixir?

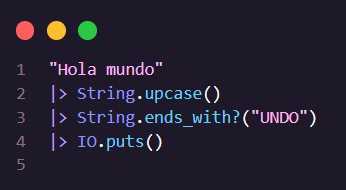
Elixir como lenguaje funcional de programación es bien conocido por su escalabilidad y confiabilidad, además de la facilidad a la hora de enfrentar a grandes volúmenes de conexiones concurrentes sin sacrificar rendimiento, se ha vuelto una de las mejores opciones a la hora de desarrollar aplicaciones web que funcionen en tiempo real, aplicaciones de mensajería, automatización de procesos empresariales y aplicaciones móviles con backend compartido.

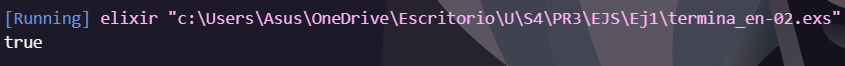
Algunos ejemplos de uso de este lenguaje serian el Phoenix framework principalmente utilizado para desarrollar apps escalables o Papercups, que es una plataforma de chat en vivo utilizada para la atención al cliente.

**5.**  



**6.**





Bibliografía

Baez, Jorge. (2024, enero 24). Elixir y el Desarrollo de Aplicaciones Actuales. Foxlabs Developers , Elixir y el Desarrollo de Aplicaciones Actuales.

Roch Moraguez, E. (s.f.). ¿Qué es el lenguaje de programación Erlang: cómo funciona y para qué sirve? LovTechnology. https://lovtechnology.com/que-es-el-lenguaje-de-programacion-erlang-como-funciona-y-para-que-sirve/

Programación en Castellano. (s.f.). Explorando Elixir: El lenguaje funcional y concurrente para aplicaciones escalables y robustas. <https://programacion.net/articulo/explorando-elixir:-el-lenguaje-funcional-y-concurrente-para-aplicaciones-escalables-y-robustas_3314>

<https://elixirschool.com/es/why>

<https://www.creolestudios.com/elixir-vs-erlang/>