**Historia**

Erlang és un llenguatge de programació funcional que va ser creat l'any 1986 per Joe Armstrong, Robert Virding i Mike Williams a Ericsson. Va ser desenvolupat inicialment com a llenguatge per a sistemes de telecomunicacions distribuïts i tolerants a fallades. El seu nom prové d'un matemàtic danès anomenat Agner Krarup Erlang. Aquest llenguatge és influenciat per alguns altres com Prolog i ha estat implementat majoritàriament en si mateix.

**Trets rellevants dels seus elements**

Variables: Són immutables, la qual cosa significa que una vegada assignat un valor a una variable, aquest no pot ser canviat.

**Condicionals:**

Erlang fa ús de l'estructura if i case per a la gestió de condicionals, on case és més utilitzat per la seva flexibilitat en el maneig de patrons.

**Funcions:**

El llenguatge permet la definició de funcions anònimes i nombrades. Les funcions són ciutadans de primera classe i es pot fer currying amb elles.

**Sintaxi i semàntica:**

La sintaxi és similar a la de Prolog i altres llenguatges funcionals. Utilitza el model de coincidència de patrons (pattern matching) extensivament.

**Sistema de tipus:**

Erlang és un llenguatge de tipatge dinàmic, però amb la possibilitat d'usar anotacions de tipus que el compilador pot utilitzar per optimitzar el codi.

Tipus de codis, compilació, i procés de transformació en codi executable

Erlang es compila a un bytecode que s'executa en una màquina virtual (BEAM - Erlang VM). El llenguatge permet la compilació dinàmica de codi, la qual cosa significa que els programes poden canviar de codi en temps d'execució. Aquesta característica és especialment útil per a sistemes que necessiten alta disponibilitat i actualitzacions en calent.

**Generació del llenguatge**

Erlang podria ser classificat com un llenguatge de quarta generació (4G) degut al seu enfocament en problemes específics de domini, en aquest cas, sistemes de telecomunicacions i aplicacions distribuïdes de gran escala. Les seves característiques de concurrencia, tolerància a falles i el maneig de múltiples tasques en temps real són altament especialitzades.

Comparació amb altres llenguatges de la mateixa generació

**Avantatges:**

**Concurrencia:** Erlang gestiona milers de processos lleugers de manera eficient.

**Tolerància a falles:** Inclou mecanismes per a la detecció i recuperació de falles.

**Actualitzacions en calent:** Permet canviar codi en execució sense aturar el sistema.

**Inconvenients**:

**Popularitat:** Menys popular que altres llenguatges, la qual cosa implica menys recursos i comunitat més petita.

**Rendiment:** Pot ser menor en comparació amb llenguatges compilats com C++ en contextos on la velocitat de processament és crítica.

**Curva d'aprenentatge:** El model funcional i les característiques úniques d'Erlang poden ser difícils de mestrejar inicialment.

**Conclusió**

Elegir el lenguaje de programación adecuado para un proyecto como la calculadora en Erlang y Elixir involucra considerar varios factores clave:

1. \*\*Características del Lenguaje\*\*: Erlang y Elixir son fuertes en concurrencia y tolerancia a fallos, lo que puede ser ventajoso si el proyecto se expande a aplicaciones críticas o en tiempo real.

2. \*\*Comunidad y Soporte\*\*: Ambos lenguajes cuentan con comunidades activas que ofrecen recursos valiosos para el desarrollo y resolución de problemas.

3. \*\*Conocimientos del Equipo\*\*: Utilizar un lenguaje con el que el equipo está familiarizado puede acelerar el desarrollo y reducir errores.

4. \*\*Requisitos del Proyecto\*\*: Se deben analizar las necesidades específicas del proyecto, como rendimiento, integración con otros sistemas, y facilidad de mantenimiento.

5. \*\*Costos de Desarrollo y Operación\*\*: Considerar los recursos que consumirá la aplicación y si el lenguaje requiere pagos de licencia.

En resumen, la elección del lenguaje se basa en un equilibrio entre las capacidades técnicas del lenguaje, la experiencia del equipo de desarrollo, y los requisitos específicos del proyecto. Para el caso de nuestra calculadora, Erlang y Elixir ofrecen una buena combinación de robustez, soporte comunitario, y características modernas que facilitan la expansión y el mantenimiento del proyecto.