

¿Porque PERL & Catalyst?

Podría empezar diciendoles que simplemente porque Perl es divertido o que Perl tiene el MOTTO, pero creo que eso no satisface a nadie así que; vemos los siguientes factores importantes a continuación:

::: Productividad :::

Existen una serie de IDE para Perl, en lo personal no he utilizado ninguno de ellos aunque he visto que la gran mayoría corren unicamente bajo plataformas Microsoft, y si bien existen unos libres no tienen el nivel necesario para considerarlos seriamente. En mi caso particular me ha resuelto la vida simplemente un editor de texto (Vim ;) y el Server de Desarrollo que trae Catalyst).

Hablando del tema de las **Librerías Disponibles**, pues el hecho de ser Perl nos abre un abanico de posibilidades enormes ya que está a nuestra entera disposición nada mas y nada menos que todo CPAN, en la cual podemos encontrar librerías para casi cualquier cosa posible.

Existen unas cuantas **opciones de Frameworks** disponibles para la programación web en Perl, entre los mas populares tenemos:

- + Catalyst
- + Gantry Framework
- + Jifty
- + Maypole
- + OpenInteract2
- + Cyclon 3
- + CGI::Builder
- + CGI::Application

El orden no indica nada, simplemente así los encuentre, en lo personal he utilizado Catalyst y ha sido una experiencia mas que agradable; Catalyst es un framework de desarrollo web el cual se centra en el patron de diseño MVC, tiene un poderoso ORM para abstraer Bases relacionales, tiene scripts helpers muy al estilo RoR y sigue la filosofía KISS (Keep it Simple, Stupid) ademas nos evita preocuparnos de reinventar la rueda para cosas como manejo de sesiones, autorización, autenticación, etc.

Lo de desarrollar un **framework** creo que se necesitaría otro proyecto para ello y desde mi punto de vista no es uno de nuestros objetivos, crear **librerías auxiliares** no es tan complejo ya que existen helpers para crear librerías dentro de Perl tales como: Module::Starter y ExtUtils::ModuleMaker por citar unos ejemplos por lo cual la dificultad de crear librerías auxiliares es baja/intermedia.

Al ser Perl un lenguaje interpretado no es "necesario" **compilarlo**, hablando del depurador Perl posee su propio **depurador** llamado Perl Debugger (perldebug) el cual nos brinda todas las comodidades que cualquier depurador decente nos brinda, ademas de que se cuenta con muchas librerías las cuales permiten crear sistemas de **pruebas automatizadas**.

Herramientas de documentación, no entiendo muy bien que se quiere decir con esto pero Perl permite crear documentación del código (llamado POD) de manera embebida, lo cual para ciclos de vida RAD es muy útil ya que p.e. hoy creo una función y en el mismo instante la voy documentando.

Dicen que Perl hace las cosas faciles "faciles" y las dificiles posibles, para alguien que viene de lenguajes como C y PHP usar Perl no sera mas que un par de Horas para adaptarse y realizar sus primeras líneas de código, ademas Perl sigue esta ley: hay que aprender el 20% del lenguaje para hacer

el 80% de las cosas que se hacen generalmente y hay que terminar de aprender el restante 80% del lenguaje para hacer el 20% de las cosas que se haran alguna vez. Por lo tanto la **curva de aprendizaje** es relativamente corta pero despues del 20% un poco empinada.

Al ser un lenguaje que durante los primeros años de su vida fue realizado para realizar reportes, desde ese instante nos da una ventaja; y así hoy por hoy, **Perl es capaz de suplir de lleno todas las necesidades que surgen a la hora de realizar un sistema transaccional** orientado a la Web como el que se pretende realizar, por lo tanto diría que Perl en sí es aplicable a este proyecto.

Ya que **RAD** busca un ciclo de desarrollo bastante agil, Perl + Catalyst se ajusta muy bien a ese ciclo de desarrollo, ya que mediante el gran conjunto de librerías y las bondades del Framework junto con un servidor de desarrollo integrado, son factores claves que ayudaran a crear prototipos agilmente, por lo tanto Perl se puede aplicar ampliamente a un ciclo de vida de desarrollo tal como RAD.

::: Popularidad del lenguaje :::

El lenguaje como tal, lastimosamente no es utilizado ampliamente por la "plebe" no se porque, pero lo que si sabemos es que Perl es como una Nava Suiza, util para todo y el ultimo recurso que tendras en una batalla (frase conocida) debe ser por lo que alguien dijo por ahí: Perl es para Geeks, pero no creo mucho eso, y segun langpop.com en su curva normalizada Perl tiene el puesto 8 en nivel de popularidad, mejor dicho de uso.

En el grupo de colaboradores actuales, por ahí he visto que Aposada, Melvin y Jotam (actualizar si alguien falta) saben o han utilizado Perl con anterioridad, con lo cual ya se tendría una base para iniciar.

Haciendo una busqueda muy escueta en Sourceforge 60/2000 ese fue el resultado 60 sistemas relacionados con contabilidad, vemos que **Perl no es muy popular en sistemas similares** al que pretendemos realizar.

El futuro de Perl es claro, se estan haciendo grandes esfuerzos y mejoras considerables para realizar Perl 6, por lo cual Perl estara en el mercado durante mucho tiempo mas.

El soporte de Perl al igual que muchos de los proyectos base el soporte principalmente en Comunidades de voluntarios, la comunidad de aportadores para Perl y CPAN es muy grande y responsable, a esto hay que añadirle el soporte que existe en ciertos países de los Perl Monks o usuarios "muy capaces" que brindan soporte siempre con el mismo espiritu de una comunidad de software libre, ademas es necesario recalcar el soporte que surge internamente en distintas distribuciones de sistemas operativos ya que estos también aportan mejoras al lenguaje.

::: Plataformas de programación :::

Una gran cantidad de **Arquitecturas de Hardware** soportan Perl, una busqueda rapida usando packages.debian.org me dio los siguientes resultados:

alpha, amd64, armel, hppa, i386, ia64, mips, mipsel, powerpc, sparc, sparc64, ...

Por lo cual el soporte que tiene Perl para las distintas arquitecturas de hardware es bastante bueno, teniendo en cuenta el contexto sobre el cual funcionaría nuestro producto aca en .sv creo que este tema esta mas que resuelto.

Este lenguaje puede ser utilizado en un gran número de **Sistemas Operativos**, entre los mas populares

tenemos Linux, FreeBSD, Mac OS X, Solaris, Windows, etc.

Al ser una arquitectura de desarrollo de software Cliente/Servidor orientado a la web creo que el tema de compatibilidad de **navegadores de internet** queda fuera de la mesa, ya que esto dependera unicamente de las herramientas y los estandares que utilizemos en la vista del usuario p.e. Javascript, CSS, etc.

Perl + Catalyst puede ser implementado en varios **Servidores Web**, entre ellos tenemos Apache, lighthttpd, IIS y Zeus