**Decision Analysis and Resolution para la Selección de Framework de Desarrollo**

1. **Establecer las guías para el análisis de decisiones**

La inadecuada elección del framework a utilizar en el desarrollo de una aplicación web genera un riesgo de impacto muy alto en cuanto a la presencia de dificultades técnicas durante el desarrollo del proyecto, teniéndose que asumir consecuencias que seguramente se convertirán en obstáculos para el cumplimiento de las actividades programadas y en las fechas establecidas. Por ello surge la necesidad de llevar a cabo un proceso formal que permita elegir la alternativa que tenga una adaptación mayor a los requerimientos del proyecto. Esto implica la obligación de disponer de un tiempo adicional en el que se ejecutará dicha elección formal, sin embargo es preferible hacerlo en comparación de los tiempos que se tendrían que emplear para la solución de inconvenientes futuros, dado que no solo se evitarían retrasos en el calendario, sino que se aumenta la posibilidad de cumplir con los objetivos trazados.

Con base en lo anterior, es entonces necesario analizar si hay razones válidas para realizar una evaluación formal a la selección de framework de desarrollo para el módulo de seguimiento que forma parte del Sistema de Información Integrado (SII) Comunidad Colciencias.

Primeramente se decidirá si la elección del framework tiene un impacto mediano o alto en el proyecto, lo cual hemos considerado que si pues hay lineamientos en aspectos de calidad y requisitos no funcionales que pueden ser más fácilmente alcanzados con unas tecnologías que con otras. Otro factor a tener en cuenta son los tiempos de desarrollo y entregas ya que dependiendo del grado de dificultad de aprendizaje y el grado de desconocimiento de una tecnología por parte del equipo de desarrollo, se podrían generar retrasos en el calendario. Así, pues se ha decidido que el análisis de framework es importante en este proyecto y los costes de su elaboración son considerables en comparación con el impacto que se podría generar a futuro.

1. **Establecer los criterios de evaluación**A continuación se enumeran los criterios que serán tenidos en cuenta para la selección del framework de desarrollo, distribuidos en 2 grupos:

|  |  |
| --- | --- |
| **Criterios Objetivos** | **Criterios Subjetivos** |
| Documentación | Dificultad de la Curva de aprendizaje |
| **Disponibilidad de librerías y aplicaciones de terceros** | Experiencia del equipo de trabajo con el framework |
| **Tamaño y grado de actividad de la comunidad** | Experiencia del equipo de trabajo con el lenguaje que usa el framework |
| **Lenguaje de programación** | Apariencia |
| Desarrollo activo |  |
| **Grado de madurez** |  |
| Testing |  |
| **Manejo de estáticos** |  |
| **Compatibilidad con el resto del ecosistema** |  |
| **Rendimiento y escalabilidad** |  |
| ORM |  |
| Seguridad |  |
| Autenticación |  |
| Reportes |  |
| **Lenguaje de plantillas** |  |

Los criterios también han sido priorizados en orden de mayor a menor importancia argumentando brevemente el porqué de su relevancia en el framework. El rango seleccionado para la calificación de cada uno de los criterios es de 1 a 10.

* 1. **Buena documentación:** Muy cierta es la frase “Puede ser el mejor framework del mundo pero si nadie sabe cómo usarlo no sirve de nada**”.** La documentación en un proyecto de software es una pieza fundamental que se debe de tener muy en cuenta pues no importa la experiencia que se tenga en un determinado framework o lenguaje de programación, siempre se va a tener que recurrir a la documentación. Por esta razón es de vital importancia que el framework tenga una documentación de calidad: fácil de ser accedida, que sea clara, concisa, completa y este actualizada a medida que el framework o el lenguaje avancen, elaborada y entregada en lo posible por los propios colaboradores del framework.
  2. **Tamaño y grado de actividad de la comunidad:** En el desarrollo siempre se van a presentar problemas, pero cuanto mayor sea el tamaño de la comunidad de un framework mayor es la probabilidad de que alguien ya haya tenido ese mismo problema y lo haya podido resolver lo cual optimiza el tiempo empleado en solucionarlo y evita que se tenga que hacer desde cero. Con una comunidad grande es fácil encontrar opiniones y ayuda en sitios web especializados como lo son StackOverflow y Google Groups.
  3. **Dificultad de la curva de aprendizaje:** El hecho de que muchos framework y lenguajes de programación sean capaces de hacer lo mismo no significa que todos sean igual de fáciles de aprender. Este hecho es importante pues una curva de aprendizaje demasiado alta podría suponer retrasos para el desarrollo del proyecto.
  4. **Lenguaje de programación:** Un framework desarrollo siempre está basado en un lenguaje de programación principal, esto implica que aunque el framework quiera implementar muchas funcionalidades siempre va a tener que estar sujeto dentro de los límites del lenguaje de programación que utiliza aprovechando sus ventajas pero también teniendo sus desventajas. Además gran parte de los problemas en el desarrollo muchas veces son referentes al lenguaje de programación por lo tanto el tamaño de la comunidad y la cantidad de documentación que gire en torno a él también se debe tener en cuenta.
  5. **Testing:** Toda construcción de software implica realizar pruebas a la aplicación para encontrar errores y verificar el correcto funcionamiento de lo planeado inicialmente. El framework elegido debe implementar un buen sistema de pruebas de software.
  6. **Manejo de estáticos:** El módulo de seguimiento en particular, requiere de un alto tráfico de archivos estáticos tanto en su carga como en su descarga por tanto es de gran importancia un framework que facilite el trato de los mismos.
  7. **Experiencia del equipo de trabajo con el framework:** una de las consideraciones que pueden facilitar en cierta medida el desarrollo del proyecto tiene que ver con el conocimiento que los encargados de la implementación tengan acerca del framework que se va a utilizar, pudiendo solucionar problemas que ya han superado con anterioridad y evadir otro tipo de dificultades, además de estar familiarizados con la sintaxis y métodos del framework.
  8. **Autenticación:** La autenticación va muy ligada al requisito de seguridad solicitado por el proyecto, además el módulo de seguimiento depende mucho de los tipos de usuarios que se encuentren realizando las operaciones así que es importante que el framework cuente con un módulo de usuarios que permita funcionalidades de creación de usuarios, asignación de permisos y manejo de sesiones.
  9. **Disponibilidad de aplicaciones y librerías de terceros:** A la hora de empezar a desarrollar en un framework generalmente sucede que, dependiendo de la complejidad del problema y del lenguaje de programación, se necesiten añadir o no librerías que incorporen funcionalidades extra. Así mismo, sucede que los módulos completos que resuelven una necesidad ya han sido creados por la comunidad y popularizados de tal manera que se han convertido en estándar. Esta característica de integrar módulos de terceros en los proyectos puede reducir enormemente el tiempo de desarrollo pues propicia la reutilización de código.
  10. **Experiencia del equipo de trabajo con el lenguaje que usa el framework:** aún si el desarrollador no tiene experiencia alguna con un determinado framework, pero si cuenta con conocimientos en el lenguaje que está implementado, no tendrá que ejercer un esfuerzo muy alto para aprender a utilizarlo y el tiempo seria mucho menor comparado con el caso en el que no se tuviera experiencia ni con el lenguaje ni con el framework.
  11. **Seguridad:** El proyecto especifica que la aplicación deberá contar con algunas características de seguridad como el uso de roles y privilegios, auditorias y tener barreras contra algunos tipos de ataques, así que el framework seleccionado deberá cumplir con la totalidad o un gran número de estos requisitos.
  12. **Rendimiento y escalabilidad:** Puesto que hay frameworks y lenguajes de programación que utilizan estructuras de datos y técnicas especiales para resolver problemas es posible que esto también se vea impactado en el rendimiento o en la escalabilidad de un proyecto de manera positiva o negativa. Es importante entonces determinar un framework que contemple un balance entre rendimiento, escalabilidad y sencillez al momento de desarrollar.
  13. **Compatibilidad:** Casi en la totalidad de los proyectos de software se tiene que una aplicación convive y se comunica con otras plataformas y tecnologías, lo que hace que la compatibilidad sea una característica interesante ya que facilitaría la escalabilidad del sistema de manera nativa o directa disminuyendo la necesidad de usar capas, protocolos, o técnicas como por ejemplo web services.
  14. **Reportes:** Para el módulo de seguimiento es importante la presencia de una herramienta para la generación de reportes, por tanto se busca que el framework seleccionado posea esta característica incluida o que como mínimo pueda ser integrada fácilmente mediante módulos de terceros.
  15. **Desarrollo activo:** Un framework con un desarrollo activo garantiza corrección de errores, mejoras de seguridad, nuevas características, simplificación. Mientras que en uno donde no haya una constante construcción es probable que pase mucho tiempo antes de encontrar una solución a un problema detectado y el proyecto este vulnerable o simplemente debas solucionarlo tú mismo en cada vez que se presente.
  16. **Grado de Madurez:** Este criterio hace referencia al tiempo de vida del framework y la presencia que tenga en el mercado, ya que un framework con más tiempo de vida que otro es probable que tenga un mayor soporte por parte de la comunidad, proyectos grandes y estables bajo su tecnología, así como documentación, aunque es de aclarar que basado solo en este criterio no hay forma de asegurar por completo lo anteriormente mencionado.
  17. **ORM (Object-Relational Mapping):** Prácticamente todos los framework de desarrollo modernos están incluyendo consigo un ORM que facilita muchísimo el manejo de la capa de persistencia en un proyecto de software pues permite realizar consultas sobre la base de datos usando el lenguaje de programación usado por el framework haciendo que sean más simples de construir y que se les pueda dar un tratamiento como si fueran objetos. Por tanto, la presencia de un ORM en un framework le da un valor agregado bastante interesante así como las funciones que permita hacer el ORM entre el modelo de clases y la base de datos.
  18. **Motor de plantillas:** Esta característica es propia de la parte de presentación en los framework de programación, pues se trata de un lenguaje de etiquetas que facilita la renderizacion de las plantillas así como el manejo de datos que vienen desde los controladores. La presencia de esta característica en el proyecto acortaría los tiempos de desarrollo y facilita el desacople junto con la escalabilidad como mínimo en la capa de presentación.
  19. **Apariencia:** es importante complementar la funcionalidad del desarrollo web netamente técnico con otro tipo de framework que permita darle una apariencia agradable y un estilo elegante a las vistas de la aplicación, estamos hablando de un framework frontend.

1. **Identificar las soluciones alternativas**

Teniendo en cuenta que el proyecto “SII – Comunidad Colciencias” es una aplicación que funcionara vía web, se han considerado frameworks especializados para esto y que hayan sido ideados con ese propósito desde su nacimiento por cuestiones de simplicidad y optimización. Así pues, basado en las experiencias del equipo de trabajo con múltiples tecnologías y en la reputación de los resultados arrojados por las mismas, se han identificado 3 posibles alternativas de configuraciones para desarrollo presentadas en el cuadro de abajo que más adelante serán puestas a evaluación con los criterios ya definidos anteriormente.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Servicio** | **Framework** | | | | |
| **Front-end** | | **Back-end** | | |
| **Bootstrap** | **Pure** | **Laravel** | **Django** | **Ruby on Rails** |
| ***Lenguaje base*** | HTML 5, CSS 3, JavaScript | CSS 3 | Php | Python | Ruby |
| ***ORM*** |  |  | Eloquent ORM | Django-ORM | Active Record Query Interface |
| ***Pre procesadores CSS*** | Less, Saas | Smacss |  |  |  |
| ***Motor de plantillas*** |  |  | Blade | Django Template Language | Erb, Builder, Coffeescript |
| ***Patrón de arquitectura*** |  |  | MVC | MTV | MVC |

1. **Seleccionar los métodos de evaluación**Ahora que se definieron los frameworks candidatos a ser elegidos, se han seleccionado algunos métodos de evaluación que serán aplicados a cada framework en el punto 5.

Un primer método es el SCORING, el cual proporciona una manera rápida y sencilla para identificar la alternativa preferible en un problema de decisión multicriterio. Las etapas del método son las siguientes:

1). Identificar la meta general del proyecto

2). Identificar las alternativas

3). Listar los criterios a emplear en la toma de decisión

4). Asignar una ponderación para cada uno de los criterios

5). Establecer en cuánto satisface cada alternativa a nivel de cada uno de los criterios

6). Calcular el Score para cada una de las alternativas

7). Ordenar las alternativas en función del score. La alternativa con el score más alto representa la alternativa a recomendar

Modelo para calcular el Score:

Donde rij=rating de la alternativa j en función del criterio I

wi=ponderación para cada criterio I

Sj=score para la alternativa j

Cada criterio tendrá un peso asignado dentro del proyecto en una escala de 1 a 5. Posteriormente cada framework será evaluado frente a este criterio obteniendo así una puntuación numérica que finalmente dirá cuál es el framework más equilibrado entre todos los criterios.

Usando sitios web como Indeed se determinara la tendencia cuanto a demanda laboral para el framework. Github nos permitirá conocer que tanta comunidad tiene el framework desarrollondolo, que tan buen puntaje tiene por los usuarios y cuantos repositorios hay usándolo. MarkLogic permite observar el comportamiento del tráfico en los últimos años con respecto al nombre de cada framework y su lenguaje base.

También se tendrán en cuenta gráficos estadísticos que permitan observar el comportamiento de cada framework en relación con la cantidad de documentación existente, la cantidad de foros creados, la actividad de la comunidad, las tendencias de empleo en determinada tecnología, la cantidad de proyectos alojados en repositorios remotos de código junto con su reputación, así como las indexaciones en los buscadores más famosos a lo largo de los últimos años hasta fechas recientes. Esta información será suministrada por sitios especializados en estadísticas de este tipo como lo son Codeeval, Indeed Job Trends, MarkLogic, Github, Google Trends, Infoq, entre otros.

Es importante aclarar el funcionamiento de Google Trends, ya que los números del gráfico reflejan la cantidad de búsquedas que se han realizado de un término específico en relación con el número de búsquedas totales realizadas en Google a lo largo del tiempo. No representan cifras de volumen de búsquedas absolutas, porque los datos están normalizados y se presentan en una escala de 0 a 100. Cada punto del gráfico se divide entre el valor del punto más alto y se multiplica por 100. Esto únicamente nos permite ver el grado de tendencia en cuanto al interés en la web por un término particular.

Adicional a esto también se realizaran pruebas de rendimiento o Benchmarks sobre un sistema operativo Linux a las diferentes combinaciones de tecnologías usadas con las herramientas proporcionadas por ***TechEmpower***.

El funcionamiento de ***TechEmpower Benchmarks*** es sencillo, la comparación de pruebas de rendimiento se basa en la realización de tareas que son comunes por los framework web, como por ejemplo: Serializacion JSON, tiempo de acceso a base de datos, tiempo de acceso al servidor. Cada framework opera en un entorno de producción (Amazon EC2 y hardware físico) con configuraciones realistas, siendo posible configurar desde el servidor web hasta el tipo de procesador a ser usado.

1. **Evaluar las soluciones alternativas**

Las evaluaciones de las alternativas se realizaran primeramente de manera individual usando los resultados arrojados por herramientas como:

* Indeed/Jobtrends
* Github
* Google Trends
* MarkLogic

Posteriormente a esto, se realizan evaluaciones comparativas entre las 3 alternativas al mismo tiempo apoyados en herramientas como:

* Codeeval
* Indeed/Jobtrends
* Google Trends
* Método Scoring
* TechEmpower Benchmarks

* 1. **Django:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Evaluador** | **Framework Django** | | | | | |
| **Fecha de lanzamiento** | **Antigüedad (# de Meses)** | | **Tamaño comprimido (Mb)** | | **Número de versiones** |
| Julio 2005 | 114 | | 6.3 | | 63 |
| **MarkLogic** | **Tendencia de crecimiento en trafico - Django**  Máximo trafico: Agosto 2008 – 9806 consultasC:\Users\Yiyo\Desktop\Django_MarkLogic.png  **Tendencia de crecimiento en trafico - Python**  Máximo trafico: Octubre 2009 – 28124 consultas  C:\Users\Yiyo\Desktop\Python_MarkLogic.png | | | | | |
| **Indeed/Jobtrends** | **Tendencia en demanda laboral**  Tendencia Absoluta **C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Django_Indeed_Absolute.png**  Tendencia Relativa**C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Django_Indeed_Relative.png** | | | | | |
| **Github** | **# Estrellas** | | **# Repositorios** | | **#Forks** | |
| 11.036 | | 36.676 | | 4.282 | |
| **Google Trends** | **Tendencia de Interes**C:\Users\Yiyo\Desktop\Django_GTrends.pngEl punto de interes más alto ha sido registrado en Enero del 2013, hace aproximadamente 1 año y 8 meses. | | | | | |

* 1. **Ruby on Rails:**

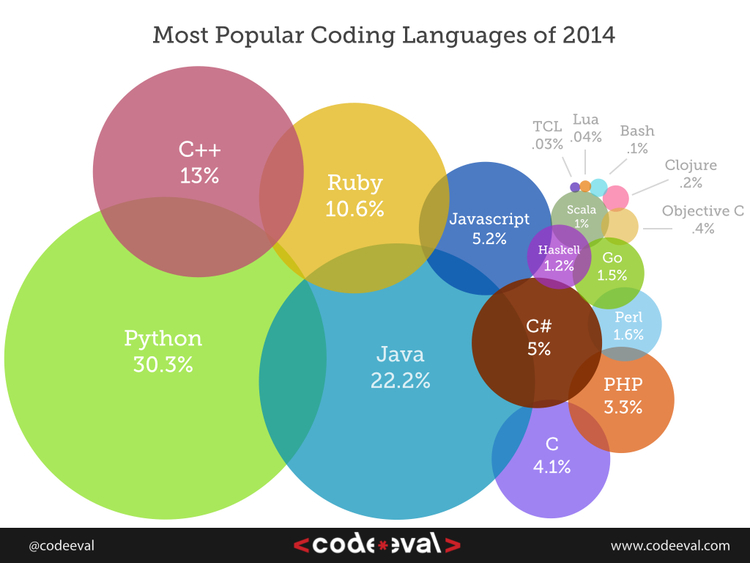
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Evaluador** | **Framework Ruby on Rails** | | | | | |
| **Fecha de lanzamiento** | **Antigüedad (# de Meses)** | | **Tamaño comprimido (Mb)** | | **Número de versiones** |
| Diciembre 2005 | 109 | | 16.2 | | 120+ |
| **MarkLogic** | **Tendencia de crecimiento en trafico – Ruby on Rails**  Máximo trafico: Marzo 2006 – 2658 consultas C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Rails_MarkLogic.png  **Tendencia de crecimiento en trafico - Ruby**  Máximo trafico: Marzo 2006 – 10266 consultas  C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Ruby_MarkLogic.png | | | | | |
| **Indeed/Jobtrends** | **Tendencia en demanda laboral**  Tendencia AbsolutaC:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Rails_Indeed_Absolute.png  Tendencia Relativa C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Rails_Indeed_Relative.png | | | | | |
| **Github** | **# Estrellas** | | **# Repositorios** | | **#Forks** | |
| 23.077 | | 135.603 | | 8.711 | |
| **Google Trends** | **Tendencia de Interes** **C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Rails_GTrends.png**El punto de interes más alto ha sido registrado en Junio del 2007, hace aproximadamente 7 años y 2 meses. | | | | | |

* 1. **Laravel**

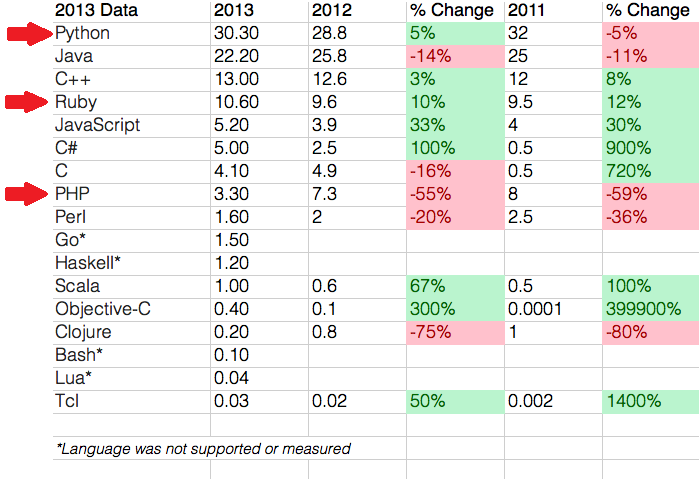
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Evaluador** | **Framework Laravel** | | | | | |
| **Fecha de lanzamiento** | **Antigüedad (# de Meses)** | | **Tamaño comprimido (Mb)** | | **Número de versiones** |
| 2011 | 40 aprox. | | 5.1 | | 10 aprox |
| **MarkLogic** | **Tendencia de crecimiento en trafico – Laravel**  Máximo trafico: Septiembre 2013 – 62 consultas C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Laravel_MarkLogic.png  **Tendencia de crecimiento en trafico - Php**  Máximo trafico: Marzo 2009 – 33208 consultas  C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Php_MarkLogic.png | | | | | |
| **Indeed/Jobtrends** | **Tendencia en demanda laboral**  Tendencia AbsolutaC:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Laravel_Indeed_Absolute.png  Tendencia Relativa C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Laravel_Indeed_Relative.png | | | | | |
| **Github** | **# Estrellas** | | **# Repositorios** | | **#Forks** | |
| 11.626 | | 8.885 | | 3.759 | |
| **Google Trends** | **Tendencia de Interes**  C:\Users\Yiyo\Google Drive\Universidad\8 Semestre\Ingenieria de Software III\Tema 5 - Componentes y Seleccion de Frameworks\Django Analytics\Laravel_GTrends.pngEl punto de interes más alto ha sido registrado en Agosto del 2014, hace aproximadamente 1 mes. | | | | | |

* 1. **Comparativa de Frameworks y Lenguajes:**

***Codeeval*** es una sitio web cuyo objetivo es ayudar a los desarrolladores, especialmente los recién graduados, a estar informados de cuáles son las últimas tendencias en lenguajes de programación por parte de las compañías, realizando de esta manera un reporte anual como el que se observa a continuación para estar al día en las tecnologías más usadas y prepararse para las futuras.

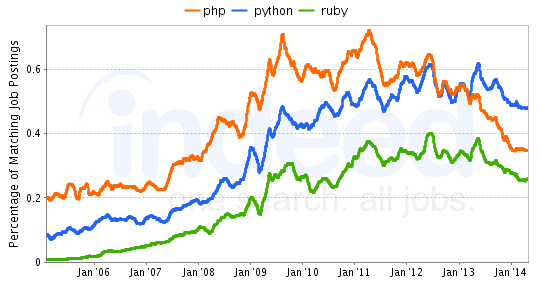
****

Además es posible observar también en la siguiente tabla la relación del comportamiento de un lenguaje con respecto al tenido en los 3 años anteriores. Para el caso de Python es posible ver como tuvo inicialmente una caída fuerte pero año a año se ha ido recuperando de a poco, mientras que en el caso de Ruby es posible observar una tendencia de crecimiento positiva en los últimos 3 años escalando a posiciones altas en la tabla y superando a otros lenguajes como por ejemplo Php, quien ha tenido un descenso demasiado fuerte en los últimos años.

****

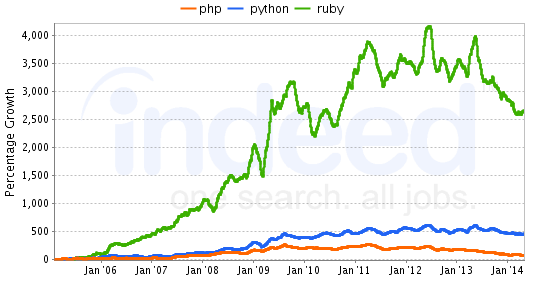
***Indeed*** por su parte nos permitio comparar al mismo tiempo la tendencia de puestos de trabajo ofertados para los 3 framework tratados en cuestión y estos fueron los resultados obtenidos hasta la fecha:

*Tendencia Absoluta Lenguajes de Programación*



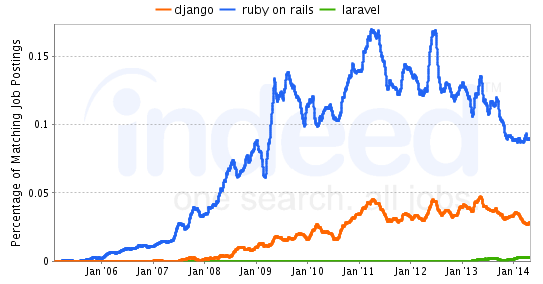
Aquí es claro como Php a nuestros días ya no es lo que fue hace algunos años, siendo superado por Python, y aunque aún no ha superado del todo a Ruby si se puede apreciar que el comportamiento de sus tendencias es completamente opuesto pues la de Ruby es positiva. Además una gran razón por la que Php aun ocupa un gran lugar es debido a que hay demasiados sitios y frameworks que lo usan a diferencia de sus competidores.

*Tendencia Relativa Lenguajes de Programación*

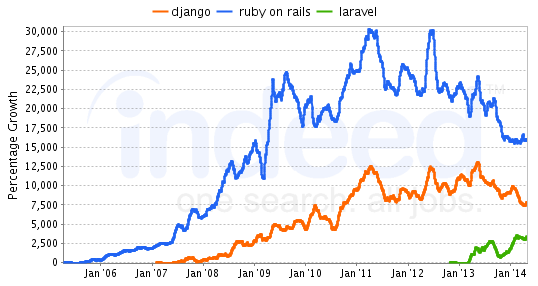


En esta grafica se puede apreciar como Ruby toma el liderazgo en términos de crecimiento pues han sido publicados muchos puestos de trabajo con este lenguaje en muy poco tiempo lo cual lo posiciona como uno de los lenguajes que estará muy presente en el futuro de la web.

*Tendencia Absoluta Frameworks*

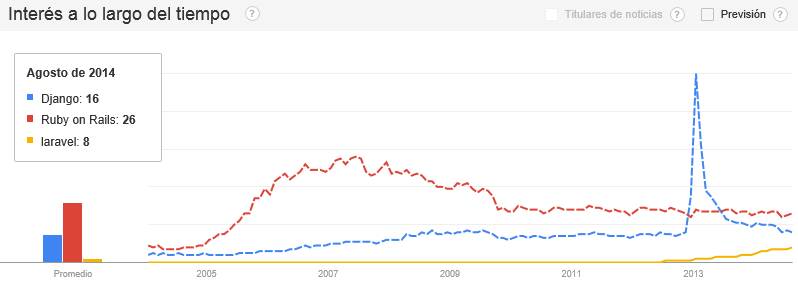


*Tendencia Relativa Frameworks*



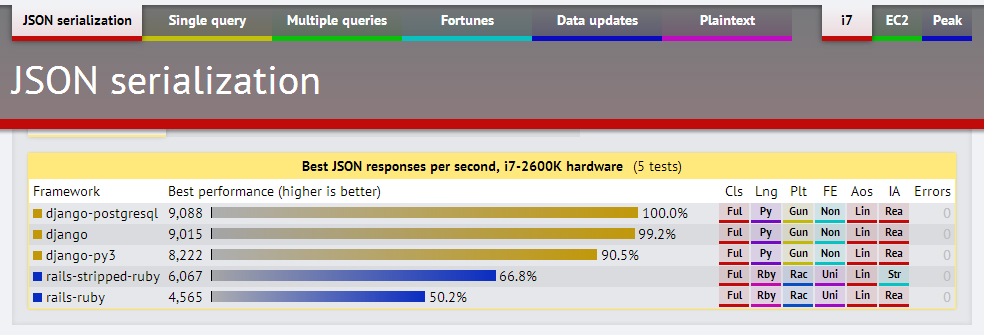
Los resultados de ***Indeed*** en cuanto a frameworks cambia un poco, pues en este lugar es Ruby on Rails quien se pone en la delantera en cuanto a cantidad de puestos de trabajo ofertados, siendo seguido por Django que se trata de mantener estable y finalmente por Laravel con una muy poca cuota actual pero con tendencia creciente.

***Google Trends*** también fue usada para comparar los 3 frameworks en cuestión al mismo tiempo, y estos fueron los índices de interés obtenidos:

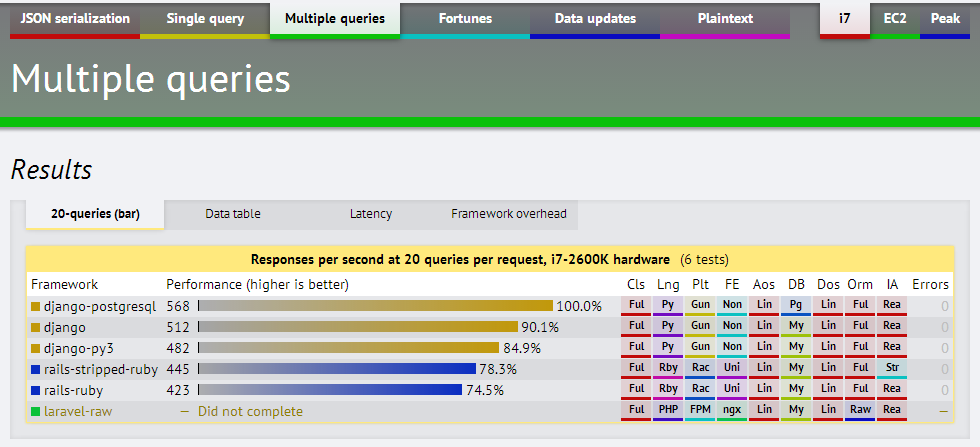


Es importante notar como aunque el interes de Laravel va aumentando aun no es lo suficientemente grande como para ser un framework maduro y soportado por una gran comunidad. Por otra parte, Django y Ruby on Rails batallan por el primer lugar teniendo ambos tendencias medianamente estables aunque hasta el momento la delantera la tiene Ruby on Rails seguido muy de cerca por Django.

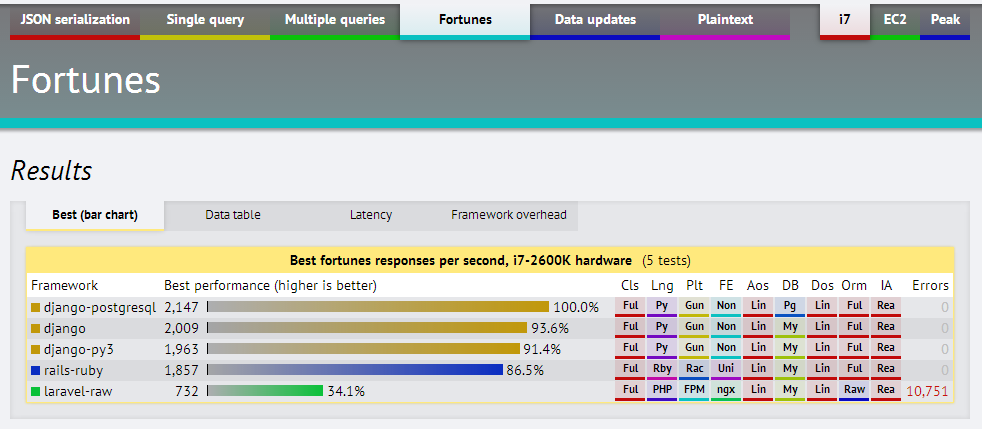
***TechEmpower Benchmarks*** es una poderosa herramienta online que permite medir el rendimiento de varios frameworks en entornos de producción poniéndolos a competir al mismo tiempo para poder observar su comportamiento bajo determinadas características. Los resultados de esta prueba bajo un procesador i7 fueron los siguientes:



*Prueba de Serializacion de JSON*



*Prueba de Múltiples Consultas a la BD*



*Prueba de Ordenamiento de Listas*

Para ver el Benchmark completo junto con todas las configuraciones especificadas e información adicional para cada una de las pruebas, puede seguir el siguiente enlace:

[http://www.techempower.com/benchmarks/#section=data-r9&hw=i7&test=json&b=1&s=2&l=5j4&p=2affw3-1º&w=1jcn&c=1&d=6&o=5&f=zik0zh-zi8qcf-v2nppb-7](http://www.techempower.com/benchmarks/#section=data-r9&hw=i7&test=json&b=1&s=2&l=5j4&p=2affw3-1o&w=1jcn&c=1&d=6&o=5&f=zik0zh-zi8qcf-v2nppb-7)

**Método Scoring:**

Aplicando el método Scoring, tenemos los siguientes pasos

1. Determinar el framework web adecuado para el desarrollo del módulo de Seguimiento perteneciente al proyecto Sistema de Información Integrado-Comunidad Colciencias.
2. Django, Ruby on Rails, Laravel
3. Criterios: ver la tabla del punto número 2, donde se establecen los criterios de evaluación.
4. Asignación de una ponderación mediante una escala de 1 a 5 puntos:

1 = muy poco importante

2 = poco importante

3 = importancia media

4 = algo importante

5 = muy importante

|  |  |  |
| --- | --- | --- |
| ID | CRITERIOS | Ponderación |
| Wi |
| 1 | Documentacion | 5 |
| 2 | Disponibilidad de librerias y aplicaciones de terceros | 3 |
| 3 | Tamaño y grado de actividad de la comunidad | 5 |
| 4 | Lenguaje de programacion | 4 |
| 5 | Desarrollo activo | 3 |
| 6 | Grado de madurez | 2 |
| 7 | Testing | 4 |
| 8 | Manejo de estáticos | 4 |
| 9 | Compatibilidad con el resto del ecosistema | 3 |
| 10 | Rendimiento y escalabilidad | 3 |
| 11 | ORM | 2 |
| 12 | Seguridad | 3 |
| 14 | Autenticación | 4 |
| 15 | Reportes | 3 |
| 16 | Lenguaje de plantillas | 2 |
| 17 | Dificultad de la Curva de aprendizaje | 4 |
| 18 | Experiencia del equipo de trabajo con el framework | 4 |
| 19 | Experiencia del equipo de trabajo con el lenguaje que usa el framework | 3 |
| 20 | Apariencia | 2 |

1. Se establece el rating de satisfacción para cada alternativa empleando una escala de 1 a 9 puntos:

1 = extra bajo 5 = medio

2 = muy bajo 6 = poco alto

3 = bajo 7 = alto

4 = poco 8 = muy alto

9 = extra alto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | CRITERIOS | Django | Ruby on Rails | Laravel |
|  | ri1 | ri2 | ri3 |
| 1 | Documentacion | 9 | 8 | 6 |
| 2 | Disponibilidad de librerias y aplicaciones de terceros | 8 | 9 | 5 |
| 3 | Tamaño y grado de actividad de la comunidad | 7 | 9 | 4 |
| 4 | Lenguaje de programacion | 8 | 8 | 8 |
| 5 | Desarrollo activo | 7 | 8 | 5 |
| 6 | Grado de madurez | 8 | 8 | 3 |
| 7 | Testing | 7 | 8 | 5 |
| 8 | Manejo de estáticos | 8 | 8 | 5 |
| 9 | Compatibilidad con el resto del ecosistema | 8 | 8 | 8 |
| 10 | Rendimiento y escalabilidad | 8 | 7 | 4 |
| 11 | ORM | 8 | 8 | 7 |
| 12 | Seguridad | 7 | 7 | 6 |
| 14 | Autenticación | 8 | 7 | 8 |
| 15 | Reportes | 6 | 6 | 4 |
| 16 | Lenguaje de plantillas | 8 | 8 | 8 |
| 17 | Dificultad de la Curva de aprendizaje | 8 | 8 | 8 |
| 18 | Experiencia del equipo de trabajo con el framework | 5 | 6 | 2 |
| 19 | Experiencia del equipo de trabajo con el lenguaje que usa el framework | 6 | 6 | 2 |
| 20 | Apariencia | 6 | 7 | 6 |

1. Se calcula la ponderación para cada alternativa.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | CRITERIOS | Ponderación | Django | Ruby on Rails | Laravel |
|  | Wi | ri1 | ri2 | ri3 |
| 1 | Documentacion | 5 | 9 | 8 | 6 |
| 2 | Disponibilidad de librerias y aplicaciones de terceros | 3 | 8 | 9 | 5 |
| 3 | Tamaño y grado de actividad de la comunidad | 5 | 7 | 9 | 4 |
| 4 | Lenguaje de programacion | 4 | 8 | 8 | 8 |
| 5 | Desarrollo activo | 3 | 7 | 8 | 5 |
| 6 | Grado de madurez | 2 | 8 | 8 | 3 |
| 7 | Testing | 4 | 7 | 8 | 5 |
| 8 | Manejo de estáticos | 4 | 8 | 8 | 5 |
| 9 | Compatibilidad con el resto del ecosistema | 3 | 8 | 8 | 8 |
| 10 | Rendimiento y escalabilidad | 3 | 8 | 7 | 4 |
| 11 | ORM | 2 | 8 | 8 | 7 |
| 12 | Seguridad | 3 | 7 | 7 | 6 |
| 14 | Autenticación | 4 | 8 | 7 | 8 |
| 15 | Reportes | 3 | 6 | 6 | 4 |
| 16 | Lenguaje de plantillas | 2 | 8 | 8 | 8 |
| 17 | Dificultad de la Curva de aprendizaje | 4 | 8 | 8 | 8 |
| 18 | Experiencia del equipo de trabajo con el framework | 4 | 5 | 6 | 2 |
| 19 | Experiencia del equipo de trabajo con el lenguaje que usa el framework | 3 | 6 | 6 | 2 |
| 20 | Apariencia | 2 | 6 | 7 | 6 |
|  | **SCORE Sj** |  | **466** | **480** | **344** |

1. La alternativa del framework Ruby on Rails obtiene la mayor ponderación, con 480 puntos, por lo tanto es la opción recomendada de acuerdo al método Scoring.
2. **Seleccionar las soluciones**

Para la elección del framework frontend, entre las dos opciones candidatas, se ha optado por Bootstrap puesto que a pesar de que Pure permite incorporar diseños y plantillas muy atractivos con CSS 3, Bootstrap incorpora además de CSS 3, los lenguajes base HTML5 y Javascript, generando así un alcance mayor para la programación de vistas de las aplicaciones.

Una vez realizada la evaluación de cada una de las soluciones alternativas, se sintetizan los siguientes resultados:

A nivel individual, se analizaron algunas características relevantes como tendencia de crecimiento en tráfico de los frameworks y sus respectivos lenguajes de programación, tendencia en la demanda laboral y tendencia de interés a lo largo del tiempo. En este aspecto, todas las alternativas presentan resultados crecientes, destacándose especialmente el notorio crecimiento de Django y Ruby on Rails en la demanda laboral que tienen hasta la fecha según la herramienta Indeed/Jobtrends. Cabe destacar la cantidad de estrellas en Github del proyecto Ruby on Rails (23.077) doblando numéricamente a los otras dos alternativas, lo cual es un indicador de que existe una comunidad muy amplia que le da respaldo a este framework y que pueden aportar soluciones a inconvenientes presentados con el manejo del mismo.

De acuerdo a las interpretaciones dadas en el punto anterior en lo referente a las comparativas entre los tres frameworks, algunos de los factores que más se destacan son la tendencia relativa de los lenguajes de programación y de los frameworks web, donde Ruby y por ende ruby on rails, respectivamente, lideran fuertemente el mercado sacándole una diferencia demasiado amplia a sus contendores.

Lo anterior, sumado al primer lugar de **Ruby on Rails** en cantidad de puntos ponderados empleando el método Scoring, permite concluir que es la alternativa que presenta mayores beneficios para el diseño e implementación web del módulo de Seguimiento del proyecto “Sistema de información integrado-Comunidad Colciencias” siendo seleccionado como el framework backendganador.