|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Angular.Js | React.Js | Vue.Js | Ember.Js | Meteor.Js | BlackBone.Js |
| Plataforma | Multiplataforma | Web | Web | Web | Multiplataforma | Web |
| Patrón de diseño | Model View Controller (MVC) | responsable solo de la capa de visualización. | Model View Controller (MVC) | Model View Controller (MVC) |  | Model View Controller (MVC) |
| Licencia | Open Source | Open Source | Open Source | Open Source | Open Source | Open Source |
| Data Binding | Si | Usa en equivalente conocido como DOM Virtual. | Si | Si | Si | Si |
| Backend | Java, Python, Ruby, c#, entre otros |  | Python, PHP, Ruby, entre otros | Java, Ruby, Python, entre otros. | PHP, Ruby,.Net, Python, entre otros. | Java, Python, Ruby, c#, entre otros |
| Comunicación al servidor | Peticiones Http | Peticiones Http | Peticiones Http | Peticiones Http | Peticiones Http y DDF | Peticiones Http desde función interna Backbone.sync. |
| Características | Ligero, buena gestión de dependencias, desacoplamiento del DOM, entre otras. | Utiliza JSX, posee isomorfismo, que permite renderizar mismo código para cliente y servidor | Gran velocidad, fácil de integrar | La curva de aprendizaje es muy pronunciada, actualiza el DOM asíncronamente. | Mantiene conexión de bajo nivel con el servidor por medio de DDF. Esta soportado sobre Node.js. Mismo lenguaje para el cliente y el servidor. | Es un framework mínimo con menos de 800 líneas de código. |
| Documentación | Posee gran cantidad de documentación | Posee gran cantidad de documentación | La documentación es abundante. | No posee mucha documentación desde el cambio de la versión 2.1.0 | Posee una gran cantidad de documentación y recursos. | Posee una gran cantidad de documentación. |
| Complejidad | Media | Media | Baja | Alta |  | Alta |
| Aplicaciones | Aplicaciones Web basadas en Javascript del lado del cliente. | Aplicaciones web completas del lado del cliente. | Puede desarrollar aplicaciones web y móviles, dependiendo de su versión. | Aplicaciones web ambisiosas. | Aplicaciones móviles y aplicaciones web. | Aplicaciones SPA (Single Page Application) |

Con ayuda del cuadro X se llevo a cabo un análisis de los principales lenguajes de programación web, dando como resultado la elección de JavaScript como el lenguaje más adecuado para el desarrollo del proyecto, tomando en cuenta todos sus beneficios. De la misma forma se llevo a cabo el desarrollo del cuadro X en donde se hace un análisis de los principales frameworks de programación de JavaScript que se encuentran en el mercado. Dando como resultado la elección del framework React.Js como el más adecuado para el desarrollo del aplicativo, debido a que las aplicaciones objetivo de este framework son aquellas que funcionan en su totalidad del lado del cliente, ocasionando así que el consumo de los recursos del servidor sea mínimo, optimizando así su uso. Por otra parte utiliza una sintaxis conocida como JSX, la cual trae grandes beneficios a la hora de optimizar el código, debido a que permite renderizar el mismo código para el cliente y el servidor [ 1 ]. Por otra parte, también posee una gran cantidad de documentación y comunidad muy activa [2], gracias a que su licencia es Open Source, ocasionando así que se encuentre diversos recursos que faciliten el desarrollo del aplicativo.

[1] <https://es.reactjs.org/docs/introducing-jsx.html>

[2] https://es.reactjs.org/community/support.html