

	CodeIgniter	Ruby On Rails
MVC	Respeto el patrón correctamente. Es bastante flexible porque no requiere la utilización del modelo (aunque es recomendable).	Perfecto. Representa el paradigma MVC completamente, ya que absolutamente todo es un objeto.
ActiveRecord	Sí, aunque utiliza una versión ligeramente modificada.	Integra perfectamente esta solución. Todas las transacciones de datos giran en torno a ella.
Scaffolding	Sí.	Trae su propio Scaffold Generator . Incluso podemos instalar un plugin que hace que funcione con AJAX.
Caché	Sí. Existe también una clase independiente llamada Sparks que lo mejora significativamente.	Sí. Sobre todo desde RoR 2.0, versión en la que se ha mejorado muchísimo esta característica (con caché de queries, páginas, etc).
Sistema de plantillas	Sí. Incorpora un sistema de plantillas integrado bastante sencillo y potente, por lo que no es necesario tirar de Smarty o similares.	En RoR el código se renderiza mediante archivos RHTML, que contienen código Ruby que genera HTML o lo que corresponda. Pese a que no es un sistema de plantillas como tal, el resultado es bastante limpio y claro.
Helpers	De todo tipo. Desde los más esenciales, hasta los de lenguaje o inserción de elementos (smilies, calendarios...).	Al igual que CodeIgniter, tiene gran variedad y calidad de ayudantes. Se podría decir que incluso le supera.
AJAX	No tiene, aunque se le puede instalar una librería que pone a disposición varios helpers para trabajar con Prototype y Scriptaculous (sólo para la versión 1.5.1).	Integra Prototype y un helper bastante extenso que ayuda a simplificar el trabajo con AJAX.
Debug	Sí. Tiene una clase que muestra datos de rendimiento y demás al final de la página si así lo deseamos.	Con Ruby-Debug podemos ver qué parte falla exactamente, ejecutar una sola línea de código, realizar cambios en caliente... Aparte de esto, cuenta también con un helper oficial.

Comparativa de frameworks

TDD	Dispone de una clase de pruebas unitarias que evalúa si el resultado esperado es el que se está obteniendo (tipos de datos, literales, booleanos...).	Completamente soportado. Tiene varios métodos integrados que lo facilitan.
Compartición de archivos	Totalmente soportada. Basta con cambiar un parámetro en el <i>config</i> e ir creando carpetas en <i>/application</i> .	El mismo caso que con CI. Sólo es necesario crear las carpetas correspondientes y tocar la configuración.
Rendimiento	Es uno de los frameworks PHP más rápidos. Parte de ello se debe a que tiene poca carga inicial, por lo que si necesitamos utilizar muchas características se vuelve algo más lento.	Muy bueno, aunque cuando se afrontan aplicaciones grandes disminuye bastante su velocidad.
Documentación	Muy buena y abundante. Pese a que hay algunas comunidades dedicadas a él, el soporte oficial es más que suficiente. Es el punto fuerte del proyecto.	También es buena y abundante, aunque sobre todo en inglés. Mientras que la de CI es sobre todo oficial, RoR dispone de más alternativas (con más libros, blogs y tutoriales dedicados a su aprendizaje).
Requisitos	PHP 4.3.2 o superior.	Ruby (con todo lo que conlleva).
Curva de aprendizaje	Bastante corta si tienes una buena base de PHP y programación orientada a objetos.	Media. Lógicamente, hace falta entender la sintaxis de Ruby, pero una vez adquiridos unos conceptos básicos, el resto está hecho para facilitar la vida del desarrollador.
Otras consideraciones		Utilizarlo supone un elemento de diferenciación , demostrando que se apuesta por las nuevas tecnologías.