Conciencia Digital
www.concienciadigital.org

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Análisis comparativo de los frameworks Grails y Spring para el desarrollo de un sistema inteligente

Comparative analysis of the frameworks grails and spring for the development of an intelligent system

Edwin Fernando Mejía Peñafiel. [1]

Recibido: 12-07-2021 / Revisado: 26-07-2021 /Aceptado: 09-08-2021/ Publicado: 06-09-2021

Abstract

DOI: https://doi.org/10.33262/concienciadigital.v4i3.2.1912

Introduction. The use of frameworks is addressed through the creation of an intelligent system in the field of medicine, implementing metrics that allow determining which framework presents the best benefits, the production rules of an expert system will allow us to determine the type of disease that suffers a patient based on their symptoms. **Objective**. Verify the best features that each of the types of frameworks has in the development of applications of this type. **Methodology**. This research is descriptive and comparative, it uses the SCRUM methodology for the development of the software and the IDEAL methodology for the development of the expert system. Tests were carried out to determine usability with 60% importance and productivity with 40% importance to frameworks, in this way to be able to establish which framework will be used within the development of the intelligent system. These percentages are granted according to what is described in the ISO / EIC 25000 standard, which proposes that the requirements and metrics of the software focused on these two parameters be evaluated. **Results**. The Grails framework in the measurement with the quality parameters gives us 57.5% instead that the Spring framework has 50%, while in productivity the values for Grails is 40% and for Spring it is 25%, from here it is obtained that the framework with which the software is going to be built is Grails. **Conclusion**. It is concluded that under the quality

[1] Escuela Superior Politécnica de Chimborazo, Facultad de Ciencias, Carrera de Ingeniería Estadística, Chimborazo – Riobamba, efmejia@espoch.edu.ec, msmejiaedwinf@yahoo.com.

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

parameters within usability and productivity, the percentages favor Grails, this result based on the principles described above, mainly the classes proposed in the framework.

Keywords: framework, agile methodologies, expert systems, intelligent systems, inference engine, knowledge base.

Resumen

Introducción. Se aborda el uso de frameworks a través de la creación de un sistema inteligente en el ámbito de la medicina, implementando métricas que permitan determinar que framework presenta mejores prestaciones, las reglas de producción de un sistema experto nos permitirán determinar el tipo de enfermedad que sufre un paciente basado en sus síntomas. **Objetivo**. Verificar las prestaciones mejores que tiene cada uno de los tipos de frameworks en el desarrollo de aplicaciones de este tipo. **Metodología**. Esta investigación está como descriptiva comparativa, usa la metodología SCRUM para el desarrollo del software y la metodología IDEAL para el desarrollo del sistema experto. Se realizaron pruebas para determinar la usabilidad con un 60% de importancia y la productividad con 40% de importancia a los frameworks, de esta manera poder establecer que framework será utilizado dentro del desarrollo del sistema inteligente. Estos porcentajes se otorgan de acuerdo a lo descrito en el estándar ISO/EIC 25000 el cual propone que se evalúen los requisitos y métricas del software enfocados en estos dos parámetros. **Resultados**. El framework Grails en la medición con los parámetros de calidad nos da un 57,5% en cambio que el framework Spring tiene el 50%, mientras que en productividad los valores para Grails es de 40% y para Spring es 25%, de aquí se obtiene que el framework con el que se va a construir el software es Grails. **Conclusión**. Se concluye que bajo los parámetros de calidad dentro de usabilidad y productividad los porcentajes favorecen a Grails, este resultado basado en los principios antes descritos, principalmente a las clases planteadas del framework.

Palabras claves: framework, metodologías ágiles, sistemas expertos, sistemas inteligentes, motor de inferencia, base de conocimiento.

Introducción

Los frameworks son una ayuda para desarrollar proyectos informáticos, se puede hablar de varios tipos de frameworks como los de desarrollo, que se pueden subdividir en desarrollo de aplicaciones genéricas, web y mobile. Este tipo de frameworks encontramos en la metodología o también denominada patrón de desarrollo como la modelo vista controlador o MVC, mediante el cual podemos separar los componentes de una aplicación en la parte lógica, de eventos y de interfaz. (Carman, 2005).

Hacia los años 80, comienza a surgir la industria de los Sistemas Expertos (Waltz, 1997). Se realizan inversiones en países de Europa, Asia y América, para generar un sistema capaz de reproducir la actividad humana con la experticia que tiene en tópicos específicos.

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

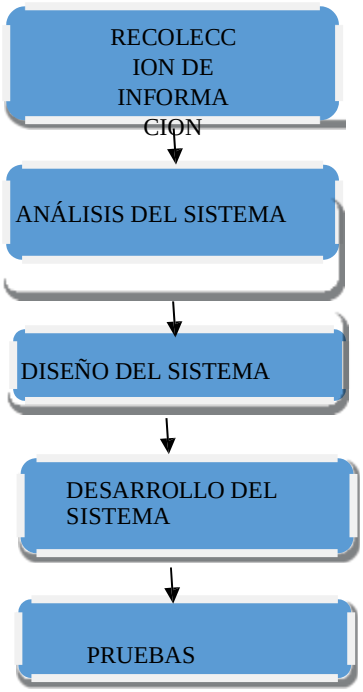
El objetivo de este trabajo de investigación consiste en realizar un análisis de los frameworks Grails y Springs bajo parámetros de calidad en el desarrollo de sistemas inteligentes que ayuden a mejorar la atención de los usuarios o pacientes en un ambiente clínico de nuestro país. La investigación realiza comparación de las métricas de Proceso y métricas de Producto con cada tipo de framework.

Metodología

Se va usar la metodología que se denomina avances por fases que se observa en la figura 1:

Figura 1

Metodología de avances por fases



Fuente: Mejía (2021)

Proyecto propuesto

Se pretende dar una solución para realizar un sistema inteligente basado en un framework. El uso de un framework para este tipo de aplicaciones puede dar mejores resultados en el momento de desarrollarlo de esta forma.

Aplicaciones Web

Las aplicaciones web reciben este nombre porque se ejecutan en internet. Es decir que los datos o los archivos en los que trabajas son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en tu computador.

El concepto de aplicaciones web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y nos envían a nuestros dispositivos o equipos los datos que requerimos en ese momento, quedando una copia temporal dentro de nuestro equipo. (GCF, 2021)

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Framework

Un Framework es una estructura previa que se puede aprovechar para desarrollar un proyecto. El Framework es una especie de plantilla, un esquema conceptual, que simplifica la elaboración de una tarea, ya que solo es necesario complementarlo de acuerdo a lo que se quiere realizar. A pesar de que su uso más común es en la informática, este concepto es también es utilizado hoy en día dentro de sistemas inteligentes (Muente G, 2020).

Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes (Oracle, 2021).

Java Enterprise Edition

Java Enterprise Edition, Java EE en adelante, es un conjunto de estándares de tecnologías dedicadas al desarrollo de Java del lado del servidor. La plataforma Java EE consta de un conjunto de servicios, API y protocolos que proporcionan la funcionalidad necesaria para desarrollar aplicaciones basadas en web de varios niveles. Es decir, desarrollaremos aplicaciones empresariales distribuidas, con arquitecturas multicapa, escritas en Java y que se ejecutan en un servidor de aplicaciones (Fontanet B., 2016).

Spring Framework

Un framework, es un conjunto de herramientas y librerías que facilitan el desarrollo de una aplicación, permitiendo la reutilización de código previamente preparado para la ejecución de cierta tarea. En otras palabras, si deseas desarrollar una aplicación, puedes optar por escribir manualmente el código correspondiente para cada proceso que necesites, como validación, formulario, interacción con la base de datos, etc., o utilizar el conocimiento de muchos otros programadores que ya lo hicieron y lo empaquetaron en un framework para que lo utilices (Unipython, 2021).

Arquitectura Spring Framework

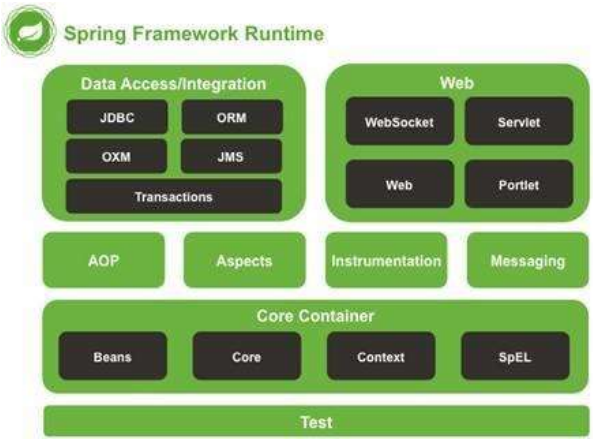
SpringFramework (creado por Rod Johnson en 2004) es una herramienta que nace con la intención de simplificar y facilitar la construcción de aplicaciones JEE. El problema de rendimiento al utilizar los EJB en JEE hizo buscar una solución para mejorar y agilizar el desarrollo de aplicaciones Java.

En la siguiente figura se pueden ver todos los módulos que integra Spring:

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Figura 2

Arquitectura de Spring Framework



Fuente: Cuervas (2021)

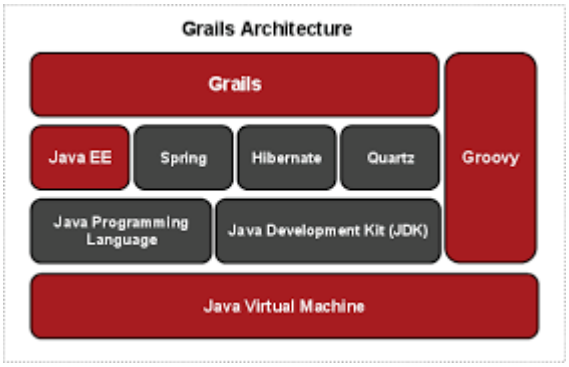
Grails Framework

Grails es un framework para el desarrollo de aplicaciones web basado en el lenguaje de programación Groovy, que a su vez se basa en la Plataforma Java. Grails está basado en los paradigmas convención sobre configuración y DRY (don't repite yourself) o no te repitas, los cuales permiten al programador olvidarse en gran parte de los detalles de configuración de más bajo nivel.

Como la mayoría de los framework de desarrollo web, Grails está basado en el patrón Modelo Vista Controlador (MVC). En Grails los modelos se conocen como clases de dominio que permiten a la aplicación mostrar los datos utilizando la vista. A diferencia de otros frameworks, en Grails las clases de dominio son automáticamente persistidas y es incluso posible generar el esquema de la base de datos. (Universidad de Alicante, 2013). En la figura 3 se muestra la arquitectura Grails y sus componentes.

Figura 3

Arquitectura Grails y sus componentes



Fuente: Smith & Ledbrook (2009)

Un sistema inteligente es una aplicación informática que posee en una Base de Conocimiento (BC) toda la información de varios expertos para resolver un problema dado. Los componentes principales de un SE son: Experto Humano, BC que está

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

compuesta de Hechos y Reglas, el Motor de Inferencia que ejecuta las reglas basada en los hechos contestados por el usuario, la Interfaz de Usuario y el Usuario quien utiliza el sistema.

MYCIN es un SE para diagnósticos, iniciado por Ed Feigenbaum y posteriormente desarrollado por E. Shortliffe. Su función es la de aconsejar a los médicos en la investigación y determinación de diagnósticos en el campo de las enfermedades infecciosas de la sangre (Badaro et al., 2013). En la figura 7 se muestran los componentes de un SE.

Condiciones para verificar las reglas de producción para un sistema experto

En la tabla 1 se muestran los hechos o antecedentes de una regla tanto en el "si" como en el "y" y "o", además en el "entonces" tenemos la conclusión de la regla o consecuente.

Tabla 1

Reglas de Producción con sus antecedentes y consecuentes

Si	Y	O	ENTONCES
Tiene fiebre	- Tiene estornudo - Arde la nariz	- Moquera	- Paciente sufre de gripe
Tiene gripe	- Dificultad para deglutir - Dolor del oído - Escalofríos - Dolor de cabeza - Dolor de garganta por más de 48 horas - Sensibilidad a la mandíbula y la		- Paciente sufre de amigdalitis aguda

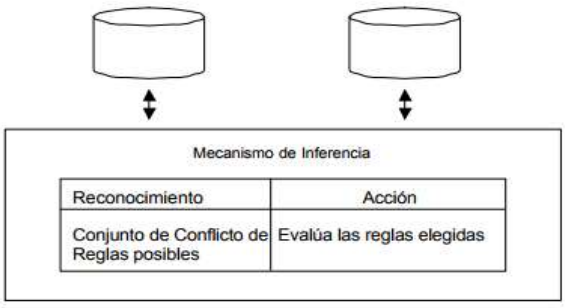
Fuente: Mejía et al. (2018)

Resultados

Esquema a dar solución como prototipo – En la figura 4 encontramos el nivel de complejidad que puede tener un motor de inferencia cuando se tiene n enfermedades con sus correspondientes síntomas, además se tiene que preveer que el usuario por si no sabe nada de diagnóstico médico.

Figura 4

Motor de inferencia y las reglas de producción

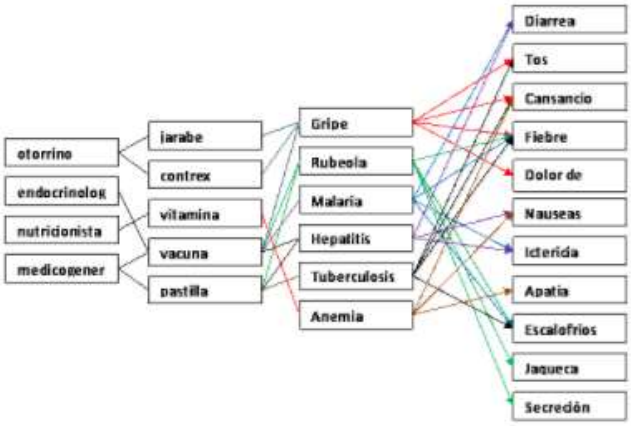


Fuente: Mejía et al. (2018)

En la figura 5 se encuentra un diagrama general de las enfermedades con sus respectivos síntomas obtenidos de la tabla 1.

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Figura 5
Diagrama general de enfermedades y sus síntomas



Fuente: Mejía et al. (2018)

Interfaz hombre - máquina – Este sistema experto que se va a realizar está compuesto por un motor de inferencia el cual infiere las reglas y las encadena de acuerdo al algoritmo propuesto desde la BC hacia la interfaz de usuario y desde la interfaz de usuario hacia la BC, en su conjunto tiene los hechos o preguntas que el sistema realiza al usuario y que se muestra en una etiqueta dentro de Java con dos botones Si y No, los cuales se almacenan en la misma con 1 si contesta Si y con -1 si contesta No, algunos usuarios no pueden contestar la pregunta entonces existe el botón siguiente. Además el usuario que sabe un poco más puede contestar con FC (Factor de Certeza de un síntoma) entre [-1 y 1] siendo -1 totalmente en desacuerdo, 0 no sé y 1 totalmente de acuerdo. La metodología de construcción del motor de inferencia está bajo programación estructurada, se tiene alrededor de 100 enfermedades con sus respectivos síntomas.

Motor de inferencia – Para poder detectar las enfermedades se usará el siguiente código, que se resuelve con un diagnóstico final o definitivo y cuando existen síntomas insuficientes para procesar las reglas (Mejía et al., 2018).

En la figura 6 se tiene el código para el motor de inferencia:

Figura 6
Código para el motor de inferencia

```
    If result = 7 Then
        frmDiagnostico1.txtcodpac.Text = txtcodpac.Text
        frmDiagnostico1.txtcodmc.Text = txtidconsulta.Text
        frmDiagnostico1.codhclinica.Text =
        txthistclinica.Text
        frmDiagnostico1.txtmotivoconsulta.Text =
        motivo_consulta.Text
        frmDiagnostico1.txtpaciente.Text = txtpaciente.Text
        frmDiagnostico1.Show()
        Me.Close()
    End If
Else
    MsgBox("Síntomas insuficientes para determinar la
    enfermedad. Por favor, seleccione nuevamente:",
    MsgBoxStyle.Critical, " ")
    frmDiagnostico1.txtcodpac.Text = txtcodpac.Text
    frmDiagnostico1.txtcodmc.Text = txtidconsulta.Text
    frmDiagnostico1.codhclinica.Text = txthistclinica.Text
    frmDiagnostico1.txtmotivoconsulta.Text =
    motivo_consulta.Text
    frmDiagnostico1.txtpaciente.Text = txtpaciente.Text
    frmDiagnostico1.Show()
    Me.Close()
End If
```

Fuente: Mejía et al. (2018)

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

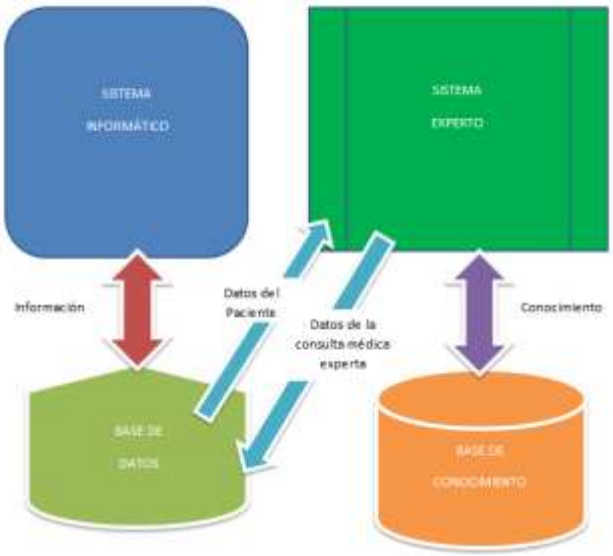
La utilidad de los encadenamientos de reglas basados en reglas de producción o reglas de inferencia, puede darse buscando el hecho de esa regla con sus correspondientes otros hechos que se unen para dar solución a una sola regla, y así sucesivamente para todas las reglas que tiene el sistema experto en sí, dentro de su BC.

En ambientes de prueba realizados nos han dado buenos resultados con un prototipo diseñado. (Mejía et al., 2018).

Infraestructura del proyecto – Para este proyecto utilizaremos la figura 7 en la cual se muestra como estaría hecho el sistema experto. La idea principal es dar solución a través de las reglas de producción y el modelo MYCIN, dando mayor fluidez al encadenamiento de reglas y dependiendo siempre del conocimiento que se ingrese. El sistema como se observa está compuesto de 2 partes principales: el sistema informático en si por donde se va ingresar la información y el sistema experto con las interfaces debidas para cada uno como son el médico y el experto, y la base de conocimiento centralizada de donde se tomarán los hechos y las reglas para poder procesar la información y así poder obtener el resultado deseado que es visualizar el diagnóstico del paciente basado en los síntomas del mismo.

Figura 7

Sistemas interconectados del proyecto



Fuente: Mejía et al. (2018)

El proceso de la toma de decisiones por parte del motor de inferencia depende de los hechos y las reglas de producción que se encuentren bien conectadas en la base de conocimiento para que el sistema experto pueda obtener una respuesta adecuada (Mejía et al., 2018).

Características del proyecto

El sistema inteligente tiene las siguientes características que se muestran en la Tabla 2:

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Tabla 2
Sistema para medicina

DESEMPEÑO	CARACTERISTICAS
Tipo de inteligencia	El reconocimiento de enfermedades lo hace como si estuviera un médico
Capacidad inferencial	Recibe antecedentes del entorno mediante la interfaz de usuario es decir a través del enfermo o cliente.
Objetivo	Obtener una enfermedad a través de los síntomas
Reglas de actuación	Interpreta los síntomas ingresados o antecedentes dentro del sistema y las compara con los que tiene en la BC.
Sistematización	Correlación entre los subsistemas que tiene el proyecto para permitir el mejor desempeño
Robustez	Reglas de producción bien definidas

Fuente: Mejía (2021)

Programación en Grails

El lenguaje de programación que emplea Grails es Groovy, Groovy fue implementado en java, he ahí el motivo por el que todas las clases, librerías java son totalmente compatibles con Grails y por ende con Groovy, la sintaxis de Groovy es totalmente descriptiva, es así que cualquier archivo que dentro de su nombre tenga las siglas "controller" sera un controlador, si dentro de su nombre tiene las siglas "service" será un servicio además de que los controladores se encuentran dentro del directorio Controllers, las clases dentro del directorio Domain Classes, etc.

El siguiente ejemplo de la clase estudiante nos dará una mejor visión del lenguaje Groovy y de la programación en Grails (Sevilla et al., 2015).

```
//1.- Definicion del paquete dentro del que se encuentra la clase de dominio
package prototipo02sesa

//2.- Defincicion de la clase de dominio Paciente
class Paciente {

//3.- Definicion de atributos para la clase paciente
String ci_Pac String nombres String apellidos Date fec_Nac
String prob_Salud
String discapacidad
String grado_Discapacidad
String telefono
String dirección

//4.- Cardinalidad entre Paciente y la historia clinica ademas de Cardinalidad entre
```

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

```
Paciente y Medico
static belongsTo =[historia:Historia, medico:Medico]
//4.1.-Cardinalidad entre Paciente y Habitos
static hasMany =[habitos:Habitos
//5.-Validaciones para los atributos de la clase Paciente
static constraints =
{
ci_Pac(blank:false, matches:'[1-9]{10}', unique:true) nombres(blank:false, maxSize:80)
apellidos(blank:false, maxSize:80) fec_Nac(validator:{return(it<new
Date())},blank:false) prob_Salud(blank:false, maxSize:80)
discapacidad(inList:["Fisica","Mental","Auditiva","Visual"])
grado_Discapacidad(inList:['25%','50%','75%','100%'])
telefono(matches:'2[1-9]{6}')
direccion(blank:false)
```

Como se puede ver el framework Grails es una herramienta tan completa que dentro del mismo archivo en el que definimos la clase son sus atributos, es posible definir las restricciones, validaciones y hasta las cardinalidades o relaciones que una clase de dominio tiene con otra clase de dominio (Sevilla et al., 2015), pero es necesario tener en claro que las cardinalidades se mapean directamente a la base de datos para de esta manera generar automáticamente el modelo de datos sin necesidad de generar la base de datos con el modelo y luego pasar a configurar la aplicación, he aquí aplicado el precepto de Grails Convención sobre Configuración (Brito, 2009).

Procesamiento de la información

La realización del análisis comparativo entre los Frameworks Grails y Spring en lo referente a Usabilidad y Productividad permitirá implementar el sistema inteligente para medicina con el framework más óptimo lo que reducirá tiempo de desarrollo, mejorará la calidad, disponibilidad y administración de la información que el sistema manejará y sin lugar a dudas mejorará la calidad del software (Sevilla et al., 2015).

Los indicadores empleados se los resume en dos tablas una para Usabilidad y otra para Productividad, se estableció una escala de valoración cuantitativa de 0 al 4 que evaluará las cualidades para lo referente a Usabilidad y a las características que se propone dentro de esta temática, para lo referente a productividad se tiene parámetros establecidos a una escala de valoración cuantitativa del 0 al 4, será necesario realizar una interpretación y posiblemente una conversión de datos empleando regla de tres para adaptar la información obtenida a la información requerida en la investigación, y finalmente se implementará una tabla resumen en la que se muestre los resultados obtenidos para cada uno de los Frameworks (Sevilla et al., 2015).

La información obtenida como resultados será recogida de los ambientes de prueba que son:

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Tabla 3
Ambientes de prueba

AMBIENTES DE PRUEBA	Determinación del mejor framework en lo referente a Usabilidad	Escenario 1: Prototipo implementado con Grails
		Escenario 2: Prototipo implementado con Spring
	Determinación del mejor framework en lo referente a Productividad	Escenario 1: Prototipo implementado con Grails
		Escenario 2: Prototipo implementado con Spring

Fuente: Sevilla et al. (2015)

Población y muestra

Una vez comprendida la problemática planteada, definidos los objetivos de investigación y las variables que forman parte de la misma es importante el determinar la población y la muestra con la que se realizaran las pruebas que demanda el análisis comparativo, constituyéndose en la población los elementos a analizar que para este caso son Frameworks para desarrollo de aplicaciones web dentro del entorno Java, existen un gran número de Frameworks que funcionan con JVM (Sevilla et al., 2015), dentro de la plataforma java entre los más populares y con mayor uso en la actualidad se encuentran PrimeFaces, IceFaces, JSF, Google web Toolkit, Spring, Struts, Grails, de los cuales se toma como muestra los Frameworks Grails y Spring dadas las características que poseen, es importante destacar que según la página de javaHispano el Framework más empleado es Spring en tanto que Grails se ubica en la quinta posición, parecería una contradicción pues uno de los pilares de Grails es Spring, por lo que Grails a parte de las funcionalidades que Spring posee implementa otras más propias de este Framework que se constituye en una fusión de varias tecnologías y hasta cierto punto varios Frameworks en Uno solo "GRAILS", por este motivo se decidió tomar como variable independiente a estos dos Frameworks Spring y Grails (Sevilla et al., 2015).

Por lo que se ve la necesidad de evaluar estos dos Frameworks para de esta manera determinar cuál es Framework que posee las mejores prestaciones en lo referente a usabilidad y productividad, elementos indispensables y decisivos al momento de elegir una plataforma para el desarrollo de aplicaciones software.

Determinación de parámetros y comparación

La necesidad de determinar que Framework da las mejores prestaciones en lo referente a usabilidad y productividad, para la implementación del sistema de evaluación y seguimiento de apraxias ha dado como resultado el realizar un análisis comparativo entre los Frameworks Grails y Spring.

Para tomar las mediciones en cuanto a usabilidad y productividad de cada uno de los Frameworks se implementarán dos prototipos uno con Grails y otro con Spring, mismos que permitirán tener una noción más real de las prestaciones que cada uno de estos Frameworks ofrece y en base a ellas determinar cuál es el mejor (Sevilla et al., 2015).

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Sevilla et al. (2015), establece que para obtener las métricas adecuadas y que se ajusten al desarrollo de software hemos empleado el estándar de la familia ISO 25000 (Square, System and Software Quality Requirements and Evaluation), estándar en el que se citan parámetros en la Calidad del producto Software, teniendo en cuenta que un producto de calidad se obtendrá únicamente cuando el proceso para conseguirlo también es de calidad, parámetros que ha saber son:

- Funcionalidad
- Rendimiento
- Compatibilidad
- Usabilidad
- Fiabilidad
- Seguridad
- Mantenibilidad
- Portabilidad

Por la naturaleza cualitativa de las características a comparar se requiere establecer un modelo que permita determinar cuantitativamente el framework con mejores prestaciones.

Los parámetros que se tendrán en cuenta para la presente investigación se han tomado de trabajos similares existentes en la unidad documental de la Escuela Superior Politécnica de Chimborazo así como en base a las necesidades que la presente investigación demanda, el estándar ISO 25000 para la calidad del producto de software, tomando únicamente la parte que hace referencia a la Usabilidad y como parte del análisis de productividad se tomó la mantenibilidad, líneas de código, componentes reutilizables, parámetros que se detallan a continuación:

Tabla 4
Parámetros a comparar

Parámetros	Parámetro 1	Parámetro 2	Parámetro 3	Parámetro 4	Parámetro 5	Parámetro 6
Usabilidad	Inteligibilidad	Aprendizaje	Operabilidad	Protección a errores de usuario	Atractividad	Accesibilidad
Productividad	Líneas de código	Mantenibilidad	Componentes reutilizables			

Fuente: Sevilla et al. (2015)

Cada uno de estos parámetros ayudaran a determinar cuál es el mejor framework para lo cual a cada una de las cualidades daremos un valor cuantitativo que determine el cumplimiento en una escala de valores como la que se muestra a continuación:

Tabla 5
Criterios y valor

Criterio	Valor
No cumple	0
Cumple en un mínimo grado	1

Fuente: Sevilla et al. (2015)

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Dada la naturaleza de la investigación y tomando en cuenta la importancia que tiene la usabilidad en los productos software, se dio un 60% de importancia a la usabilidad y un 40% de importancia a la productividad, porcentajes que se tomaron por lo descrito en el estándar ISO/EIC 25000 mismo que propone una evaluación de los requisitos y métricas del producto software enfocados en su mayoría a la usabilidad, tal cual como se muestra en la Tabla6.

Tabla 6
Tabla de valores usabilidad

Usabilidad	Puntos Posibles	Porcentaje signado
Inteligibilidad	0-4	10%
Aprendizaje	0-4	10%
Operabilidad	0-4	10%
Protección a errores de usuario	0-4	10%
Atractividad	0-4	10%
Accesibilidad	0-4	10%
TOTAL	**24**	**60%**

Fuente: Sevilla et al. (2015)

Sin quitar valor a la productividad tal cual como se muestra en la Tabla7, pero teniendo en cuenta que si bien es importante para quienes desarrollamos software la usabilidad se constituye en un factor decisivo al momento de elegir la plataforma de desarrollo, por lo que en el estándar la Usabilidad se presenta como una Categoría para la evaluación de la calidad del producto software en tanto que la Productividad únicamente se la menciona como una sub Categoría y se la toma de lo descrito en el estándar anterior a este que es el ISO/EIC 9126 y el estándar ISO/EIC 14598. (Sevilla et al., 2015).

Con base a lo mencionado en el párrafo anterior se tiene la siguiente tabla con los porcentajes y ponderaciones para cada variable analizar (Usabilidad, Productividad).

Tabla 7
Tabla de valores productividad

Productividad		Valores posibles	Valor del factor	Porcentaje Asignado
Líneas de código	Línea de código en clases	0-4	16	13,30%
	Línea de código en controladores	0-4		
	Línea de código en servicios	0-4		
	Línea de código en vistas	0-4		
Mantenibilidad	Mantenibilidad de clases	0-4	16	13,30%

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Tabla 7
Tabla de valores productividad (continuación)

Productividad		Valores posibles	Valor del factor	Porcentaje Asignado
Mantenibilidad	Mantenibilidad de controladores	0-4	16	13,30%
	Mantenibilidad de vistas	0-4		
	Mantenibilidad de servicios	0-4		
Componentes Reutilizables	Reutilización de clases	0-4	16	13,30%
	Reutilización de controladores	0-4		
	Reutilización de vistas	0-4		
	Reutilización de servicios	0-4		
TOTAL		**48**	**48**	**40%**

Fuente: Sevilla et al. (2015)

Como se puede observar en la Tabla7 el valor de productividad es de 40% referente a los valores que se van analizar.

En la Tabla8 se presenta un resumen con los porcentajes de usabilidad y productividad para el estudio de esta investigación.

Tabla 8
Tabla de resumen

Variable	Total puntuación Máxima posible	Total Porcentaje Máximo posible
USABILIDAD	24	60%
PRODUCTIVIDAD	48	40%
TOTAL	72	100%

Fuente: Sevilla et al. (2015)

Parámetros para medir la Usabilidad de los Frameworks. Inteligibilidad.

La inteligibilidad según el diccionario se refiere a Qué puede ser comprendido o entendido (Farlex, 2013), para nuestro caso la inteligibilidad se ve asociada a la capacidad del framework que permite al usuario entender si el framework es el adecuado y la manera de emplearlo para tareas y condiciones particulares (Sevilla et al., 2015).

Inteligibilidad para Grails

Teniendo en cuenta la definición de Inteligibilidad se dio una calificación a cada uno de los frameworks para lo cual se tomó en cuenta la experiencia con cada uno de ellos así como la información encontrada en estudios previos publicados en la web como

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

Decidiendo entre Java, Groovy/Grails, dando como resultado la siguiente tabla (Sevilla et al., 2015).

Inteligibilidad para Spring

La experiencia así como la información existente sobre el Framework Spring han sido quienes demuestren que Spring se constituye en una herramienta muy potente y con las prestaciones necesarias para resolver cada uno de los requisitos que el usuario pueda tener, empleando ciertos plugins y librerías para requerimientos específicos como los de seguridad, control de accesos entre otros (Sevilla et al., 2015).

Tabla 9

Tabla de Usabilidad entre Grails y Spring

Usabilidad	Puntuación GRAILS	Porcentaje GRAILS	Puntuación SPRING	Porcentaje SPRING
Inteligibilidad	4	10%	4	10%
Aprendizaje	4	10%	2	5%
Operabilidad	4	10%	2	5%
Protección a errores de usuario	3	7,50%	4	10%
Atractividad	4	10%	4	10%
Accesibilidad	4	10%	4	10%
TOTAL	**23**	**57,50%**	**20**	**50%**

Fuente: Sevilla et al. (2015)

Como se mencionó anteriormente la usabilidad tiene un valor porcentual del 60% por lo que los resultados obtenidos son teniendo en cuenta esta base, para la determinación de porcentajes se empleó la siguiente formula:

Valor Porcentual = (totalPuntuacionObtenida*60%)/(Puntuacion Maxima a Alcanzar)

Porcentaje_Grails = (23*60%)/24 = 57,50%

Porcentaje_Spring = (20*60%)/24 = 50%

Eficiencia del sistema planteado

Para la productividad se tomaron parámetros que relacionan el número de líneas de código generadas para cumplir con el o los requerimientos de un determinado sprint, es importante anotar que para medir productividad o establecer las métricas que la determinen es indispensable saber que los criterios para establecer métricas de productividad son capacidad de ser medido y objetividad, generalidad, significancia, e independencia.

(Van Laer et al., 2015).

- Líneas de código por sprint
- Líneas de código en clases

El primer Sprint es el de Administración de Pacientes, médicos y la historia clínica por lo que se implementaron las clases correspondientes y al contabilizar las líneas de código

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

los resultados arrojados en cada framework se muestra en la tabla 10.

Tabla 10
Líneas de código en clases

Framework	Líneas de Código en Clases
Grails	80
Spring	1184

Fuente: Sevilla et al. (2015)

En la Tabla11 se muestra una comparativa entre el framework Grails y Spring referente a su productividad.

Tabla 11
Tabla comparativa de Productividad entre Grails y Spring

PRODUCTIVIDAD	Puntuación GRAILS	Porcentaje GRAILS	Puntuación SPRING	Porcentaje SPRING
Línea de código en clases	4	3,33	0	0
Línea de código en controladores	4	3,33	0	0
Línea de código en servicios	4	3,33	0	0
Línea de código en vistas	4	3,33	0	0
Mantenibilidad de clases	4	3,33	3	2,5
Mantenibilidad de controladores	4	3,33	3	2,5
Mantenibilidad de vistas	4	3,33	4	3,33
Mantenibilidad de servicios	4	3,33	4	3,33
Reutilización de clases	4	3,33	4	3,33
Reutilización de controladores	4	3,33	4	3,33
Reutilización de vistas	4	3,33	4	3,33
Reutilización de servicios	4	3,33	4	3,33
TOTAL	**48**	**40%**	**30**	**25%**

Fuente: Sevilla et al. (2015)

En el caso del prototipo resultó bastante eficiente con las clases planteadas, ya que como vemos en la tabla 10 y 11 donde se muestra los resultados de la investigación tenemos que el framework Grails es con el que se va a trabajar en el desarrollo del sistema inteligente para medicina.

El emplear estándares de calidad como los empleados en la presente investigación permite tener una mayor visión y comprensión del objeto de investigación, además de brindar un respaldo y aval científico a los resultados obtenidos en una investigación.

Discusión

La toma de decisiones respecto al framework a escoger se basa en los parámetros de calidad con la usabilidad y la productividad para el desarrollo del sistema inteligente de medicina, para este caso se escogerá el framework Grails. La dimensión del tiempo en cuanto a respuesta del sistema esta medido en segundos.

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

La metodología SCRUM nos ayuda a determinar parámetros de calidad en cada fase del desarrollo de software, y en el momento que se comience a desarrollar por completo el sistema se verán los resultados óptimos del mismo, ya que todavía toca hacer unos ajustes, dado que solamente se hizo para un prototipo. A pesar de esto el prototipo está dando los resultados que se espera en cuanto a nivel de confianza y tiempos de respuesta del sistema.

Ahora también hay que decir que se presentó un problema en el momento de ejecutar el prototipo, es el de las diferentes clases que se tiene que implementar. Hay que prever que cada clase creada cumpla con sus términos deseados; por eso se tomó la situación de realizar clases solas como se dijo anteriormente.

La metodología IDEAL es un punto muy importante al momento de comenzar a realizar nuestra base de conocimiento y el motor de inferencia para el Sistema Experto de medicina.

Como trabajos futuros se propone realizar estudios más profundos con otros tipos de frameworks y en ambientes de trabajo.

Conclusiones

- En este artículo se ha realizado un análisis de los frameworks Grails y Spring para el desarrollo de sistemas inteligentes basado en reglas de producción para solucionar el problema de atención a pacientes en la ciudad de Riobamba, provincia de Chimborazo, país el Ecuador.
- El uso de frameworks es de gran ayuda para este tipo de software, pero hay que saber escoger el mejor, al momento de desarrollarlo.
- Los problemas por donde hemos ido en este campo de la creación del sistema inteligente fueron algunos, pero no complejos ya que en el camino se fue encontrando las soluciones adecuadas.
- El framework Grails en la medición con los parámetros de calidad nos da un 57,5% en cambio que el framework Spring tiene el 50%, mientras que en productividad los valores para Grails es de 40% y para Spring es 25%, de aquí se obtiene que el framework con el que se va a construir el software es Grails.
- La importancia radica que basado en el desarrollo de este sistema inteligente, el sistema experto procesa reglas de producción que según sus antecedentes o síntomas ingresados del paciente nos dan la enfermedad que el mismo padece, con esto se puede obtener un diagnóstico con hasta un 90% de efectividad, en lo referente a usabilidad y productividad. Considero que es un software muy útil para que el problema de saber usar un framework u otro dentro de sistemas inteligentes y podernos dar una solución

Referencias bibliográficas

Badaro S, Ibañez L and Agüero M. (2013). Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones. Consultado el 15 de junio de

2000http://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf.

Brito, P. (2009). Ingeniería de sistemas expertos. Editorial Nueva Librería. ISBN: 987-1104-15-4

Carman, J.M. (2005). Blended Learning Design: Five Key Ingredients. Learning Technical Report. Agilant.

Cuervas, J. (2021). Qué es Spring Framework – Características. Consultado el 20 de abril de 2021. https://www.atsistemas.com/blog/qu-es-spring-framework-caractersticas-i

Farlex. (2013), "Fundamentals of expert systems", en Ann. Rev. Comput. Science, 3.

Fontanet, B. (2016). Java EE y el desarrollo web: Un enfoque de aprendizaje. Consultado el 18 de abril del 2020. https://www.fundesem.es/bt/publicacion-java-ee-y-el-desarrollo-web-un-enfoque-de-aprendizaje

Mejía, F., Vaca, B. y Menes, I. (2018). Metodología de construcción de un sistema experto utilizando reglas de inducción con programación estructurada.

Mejía, F. (2021). Algoritmo de programación estructurada enfocado a la detección y conteo vehicular de manera inteligente en una intersección.

Muente, G. (2020). Guía completa del Framework: qué es, cuáles tipos existen y por qué es importante en Internet. Consultado el 20 de noviembre del 2020. https://rockcontent.com/es/blog/framework/

Oracle. (2021). Que es la tecnología JAVA y para que la necesito. Consultado el 18 de febrero de 2021. https://www.java.com/es/download/help/whatis_java.html

Sevilla, M., Hidalgo, M., Mejía, F. y Santillán, J. (2015). Análisis Comparativo entre los Frameworks Grails y Spring para el Desarrollo del Sistema de Evaluación y Seguimiento de Apraxias. Escuela Superior Politécnica de Chimborazo. Facultad de Informática y Electrónica. Carrera de Ingeniería en sistemas informáticos. Consultado el 16 de enero de 2018. http://dspace.espoch.edu.ec/handle/123456789/4588

Smith, G., Ledbrook, P. (2009). Grails in Action. Manning Publications Co.

Unipython. (2021). Qué es el framework spring y las ventajas de utilizarlo. Consultado el 10 de marzo de 2021. https://unipython.com/que-es-el-framework-spring-y-las-ventajas-de-utilizarlo/

Universidad de Alicante. (2013). Introducción a Grails. Departamento de Ciencia de la Computación e Inteligencia Artificial. Consultado el 16 de octubre de 2020. http://www.jtech.ua.es/j2ee/restringido/grails/sesion03-apuntes.pdf

Van Laer, T., Ruyter, K., Visconti, M. y Wetzels, M. (2014). The extended

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

transportation-imagery model: A meta-analysis of the antecedents and consequences of consumers' narrative transportation. Journal of Consumer research. Vol.40. Issue.5. Pages.797-817. University of Chicago Press. Consultado el 16 de abril de 2020. https://scholar.google.com/citations?view_op=view_citation&hl=en&user=6FPBTn8AAAAJ&citation_for_view=6FPBTn8AAAAJ:d1gkVwhDpl0C

Waltz, D. L.(1997). Artificial Intelligence: Realizing the Ultimate Promises of Computing. AI Magazine, Volume 18, Number 3. (pp 49-52)

ISSN: 2600-5859
Vol. 4, N°3.2, p. 118-137, septiembre, 2021

PARA CITAR EL ARTÍCULO INDEXADO.

Edwin Fernando. (2021). Análisis comparativo de los frameworks Grails y Spring para el desarrollo de un sistema inteligente. ConcienciaDigital, 4(3.2), 118-137. https://doi.org/10.33262/concienciadigital.v4i3.2.1912