



**UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN**

**Sistema de Gestión de Instrumentos de Cooperación para la  
Coordinación de Relaciones Nacionales e Internacionales Unach  
usando tecnología .Net Core**

**Trabajo de Titulación para optar al título de Ingeniero en  
Tecnologías de la Información**

**Autores:**

**Reyes Solís Sebastián Alejandro  
Pinto Matos Ricardo Francel**

**Tutor:**

**Ing. Pamela Alexandra Buñay Guisñan, MsC.**

**Riobamba, Ecuador. 2024**

## DECLARATORIA DE AUTORÍA

Nosotros, Sebastián Alejandro Reyes Solís, con cédula de ciudadanía 0604844100 y Ricardo Francel Pinto Matos, con cédula de ciudadanía 2300852627, autor (a) (s) del trabajo de investigación titulado: Sistema de Gestión de Instrumentos de Cooperación para la Coordinación de Relaciones Nacionales e Internacionales Unach usando tecnología .Net Core 8, certificamos que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de nuestra exclusiva responsabilidad.

Asimismo, cedemos a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de nuestra entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

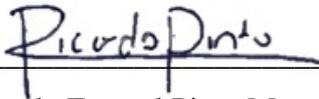
En Riobamba, 22 de mayo de 2024.



---

Sebastián Alejandro Reyes Solís

C.I: 0604844100



---

Ricardo Francel Pinto Matos

C.I: 2300852627

## DICTAMEN FAVORABLE DEL PROFESOR TUTOR



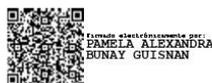
Dirección  
Académica  
VICERRECTORADO ACADÉMICO



UNACH-RGF-01-04-08.11  
VERSIÓN 01: 06-09-2021

### ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN

En la Ciudad de Riobamba, a los 23 días del mes de abril de 2024, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por los estudiantes **RICARDO FRANCEL PINTO MATOS** con CC: **2300852627** y **SEBASTIÁN ALEJANDRO REYES SOLÍS** con CC: **0604844100**, de la carrera de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado "**SISTEMA DE GESTIÓN DE INSTRUMENTOS DE COOPERACIÓN PARA LA COORDINACIÓN DE RELACIONES NACIONALES E INTERNACIONALES UNACH USANDO TECNOLOGÍA .NET CORE 8**", por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



MSc. Pamela Alexandra Buñay Guisñan  
**TUTORA**



## CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Sistema de Gestión de Instrumentos de Cooperación para la Coordinación de Relaciones Nacionales e Internacionales UNACH usando tecnología .Net Core, presentado por **Sebastián Alejandro Reyes Solís**, con cédula de identidad número **0604844100** y **Ricardo Francel Pinto Matos**, con cédula de identidad número **2300852627**, bajo la tutoría de **Msc. Pamela Alexandra Buñay Guisñan**; certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba a 22 mayo de 2024.

Mgs. Ana Congacha  
**Presidente del Tribunal de Grado**



Firma

Mgs. Diego Reina  
**Miembro del Tribunal de Grado**



Firma

Mgs. Johanna Moyano  
**Miembro del Tribunal de Grado**



Firma

# CERTIFICADO ANTIPLAGIO

## Original



Dirección  
Académica  
VICERRECTORADO ACADÉMICO



## CERTIFICACIÓN

Que, **PINTO MATOS RICARDO FRANCEL** con CC: **2300852627** y **REYES SOLÍS SEBASTIÁN ALEJANDRO** con CC: **0604844100**, estudiantes de la Carrera **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, Facultad de **INGENIERÍA**; han trabajado bajo mi tutoría el trabajo de investigación titulado "**SISTEMA DE GESTIÓN DE INSTRUMENTOS DE COOPERACIÓN PARA LA COORDINACIÓN DE RELACIONES NACIONALES E INTERNACIONALES UNACH USANDO TECNOLOGÍA .NET CORE**", cumple con el 10%, de acuerdo al reporte del sistema Anti plagio **TURNITIN**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 14 de mayo de 2024



Firmado electrónicamente por:  
PAMELA ALEXANDRA  
BUÑAY GUIÑAN

MsC. Pamela Alexandra Buñay Guisñan  
**TUTORA**

## **DEDICATORIA**

Querida Familia Reyes Solís y Pinto Matos,

Hoy, con inmenso agradecimiento y profunda emoción, dedicamos esta tesis a ustedes. En cada página escrita, en cada desafío superado, en cada descubrimiento realizado, su apoyo inquebrantable ha sido nuestro faro de inspiración.

Durante esta travesía académica, ustedes han sido nuestra ancla en las tormentas y nuestro viento a favor en los momentos de calma. Sus palabras de ánimo, paciencia infinita y amor incondicional han sido los cimientos de nuestro éxito. Sin su constante aliento, este logro no sería posible.

Esta tesis es el fruto de no solo nuestro esfuerzo, sino también de su inquebrantable fe en nosotros. Con profundo cariño, les dedicamos este trabajo como un testimonio de gratitud por cada sacrificio que han hecho y por cada sueño que han compartido con nosotros.

Nuestro amor por ustedes no tiene límites, al igual que el conocimiento que hemos adquirido a lo largo de este viaje. Esta tesis es un tributo a nuestro amor, a nuestra unidad, y a la confianza que siempre han tenido en nosotros.

Con amor y gratitud eternos,

Sebastián Alejandro Reyes Solís y Ricardo Francel Pinto Matos

## **AGRADECIMIENTO**

En el camino de la tesis, hemos contado con el apoyo invaluable de personas que han sido fundamentales en nuestra travesía académica. Hoy, queremos expresar nuestra sincera gratitud a quienes hicieron posible este logro.

En primer lugar, deseamos agradecer a nuestra tutora de tesis, la Msc. Pamela Alexandra Buñay Guisñan. Su orientación, experiencia y dedicación fueron cruciales para el desarrollo de este trabajo. Sus consejos y perspicacia nos ayudaron a enfocarnos y a alcanzar nuestros objetivos académicos. Estamos profundamente agradecidos por su paciencia y compromiso.

También queremos extender nuestro agradecimiento a nuestros amigos que han sido compañeros leales a lo largo de nuestra carrera. Sus palabras de aliento, su apoyo mutuo y su amistad inquebrantable han enriquecido nuestra experiencia universitaria. Sin su compañía, estos años no habrían sido los mismos.

Por último, pero no menos importante, agradecemos a nuestros seres queridos. Sus constantes ánimos, comprensión y amor incondicional han sido el fundamento de nuestro éxito. Este logro no habría sido posible sin su apoyo constante y su fe en nosotros.

Gracias a todos ustedes, nuestra tesis se ha convertido en una realidad. Cada uno de ustedes ha dejado una huella indeleble en nuestro viaje académico, y les estamos agradecidos más allá de las palabras. Este logro es también suyo.

Con gratitud eterna,

Sebastian Alejandro Reyes Solis y Ricardo Francel Pinto Matos

# ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA	
DICTAMEN FAVORABLE DEL PROFESOR TUTOR	
CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL	
CERTIFICADO ANTIPLAGIO	
DEDICATORIA	
AGRADECIMIENTO	
ÍNDICE GENERAL	
RESUMEN	
ABSTRACT	
CAPÍTULO I. INTRODUCCION.....	15
1.1. Antecedentes.....	15
1.2. Planteamiento del Problema.....	16
1.2.1. Problema y Justificación.....	16
1.2.2. Formulación del Problema.....	16
1.3. Objetivos.....	16
CAPÍTULO II. MARCO TEÓRICO.....	17
2.1. Aplicación Web.....	17
2.2. Tecnologías de Desarrollo Web.....	17
2.2.1. .Net Core 8.....	17
2.2.2. ASP.NET Core MVC.....	20
2.2.3. Net Core y el desarrollo de aplicaciones web.....	20
2.3. Metodologías de Desarrollo Ágil.....	22
2.3.1. XP.....	23
2.4. Arquitectura de Software.....	23
2.4.1. Microservicios.....	23
2.5. Modelo FURPS.....	24
CAPÍTULO III. METODOLOGIA.....	26
3.1. Tipo de Investigación.....	26
3.2. Técnicas de recolección de Datos.....	26
3.3. Población de estudio y tamaño de muestra.....	26
3.4. Métodos de análisis y Procesamiento de datos.....	26

3.5. Identificación de variables.....	26
3.5.1. Variable Dependiente .....	26
3.5.2. Variable Independiente.....	26
3.6. Operacionalización de Variables .....	27
3.7. Desarrollo .....	29
3.7.1. Fase de Planificación .....	29
3.7.2. Iteraciones.....	32
3.7.3. Fase de Diseño.....	34
3.7.4. Fase de Codificación .....	37
3.7.5. Fase de Pruebas .....	40
3.7.6. Fase de Lanzamiento .....	40
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN .....	41
4.1. Resumen de las pruebas de carga .....	41
4.2. Eficacia .....	42
4.3. Tiempo de Respuesta.....	42
4.4. Uso de recursos.....	42
4.5. Comparación con el modelo FURPS .....	43
4.6. Discusión .....	43
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES .....	44

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Comparación del tamaño de una aplicación en .NET 7 y .NET 8 .....	19
<b>Tabla 2:</b> Comparativa entre .Net Core y Node.Js .....	21
<b>Tabla 3:</b> Operacionalización de Variables.....	27
<b>Tabla 4:</b> Requisitos funcionales.....	30
<b>Tabla 5:</b> Requisitos no funcionales.....	30
<b>Tabla 6:</b> Historia de usuario implementada en la aplicación.....	31
<b>Tabla 7:</b> Tareas de la primera iteración .....	32
<b>Tabla 8:</b> Tareas de la segunda iteración .....	32
<b>Tabla 9:</b> Tareas de la tercera iteración.....	33
<b>Tabla 10:</b> Tareas de la cuarta iteración.....	33
<b>Tabla 11:</b> Tareas de la quinta iteración.....	33
<b>Tabla 12:</b> Lista de módulos integrados al producto final .....	37
<b>Tabla 13:</b> Prueba de carga en los módulos principales.....	41
<b>Tabla 14:</b> Requerimientos solicitados al aplicativo.....	42
<b>Tabla 15:</b> Uso de recursos .....	42
<b>Tabla 16:</b> Comparación de valores del modelo FURPS vs. valores del estudio .....	43
<b>Tabla 17:</b> Historia de Usuario Diseño de la Arquitectura .....	49
<b>Tabla 18:</b> Historia de Usuario Diseño y creación de la base de datos.....	49
<b>Tabla 19:</b> Historia de Usuario Módulo de gestión instrumentos de cooperación.....	49
<b>Tabla 20:</b> Historia de Usuario Módulo de seguimiento de actividades del convenio .....	50
<b>Tabla 21:</b> Historia de Usuario Notificaciones por correo.....	50
<b>Tabla 22:</b> Historia de Usuario Módulo de reporte de convenios.....	50
<b>Tabla 23:</b> Historia de Usuario Módulo de configuración de roles de usuario.....	50
<b>Tabla 24:</b> CRC Convenio .....	53
<b>Tabla 25:</b> CRC EmpresaConvenio .....	53
<b>Tabla 26:</b> CRC Usuario .....	53
<b>Tabla 27:</b> CRC Rol .....	53
<b>Tabla 28:</b> CRC Transacción .....	54
<b>Tabla 29:</b> CRC RolTransacción.....	54
<b>Tabla 30:</b> CRC RolUsuario .....	54

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Comparativa de Rendimiento .....	18
<b>Figura 2:</b> Flujo de los componentes de la Arquitectura MVC (Alan, 2023).....	20
<b>Figura 3:</b> Fases de la metodología XP (SINNAPS, 2020) .....	23
<b>Figura 4:</b> Arquitectura con microservicios.....	24
<b>Figura 5:</b> Plan de iteraciones .....	32
<b>Figura 6:</b> Arquitectura del sistema .....	34
<b>Figura 7:</b> Diseño de la base de datos .....	36
<b>Figura 8:</b> Controlador de Módulo de Convenio .....	37
<b>Figura 9:</b> Modulo de Convenio – Vista Principal .....	38
<b>Figura 10:</b> Diseño de Reportes - DevExpress .....	38
<b>Figura 11:</b> Modulo de Reportes – Reporte Convenios.....	39
<b>Figura 12:</b> Index del Módulo de Configuración.....	39
<b>Figura 13:</b> Modulo de Convenio – Vista Principal .....	40
<b>Figura 14:</b> Tabla de resumen de la prueba de carga.....	41
<b>Figura 15:</b> Grafica de tiempos de respuesta .....	42

## RESUMEN

La investigación se basa en la implementación y evaluación del rendimiento del sistema de gestión de instrumentos de cooperación para la Coordinación de Relaciones Nacionales e Internacionales de la Universidad Nacional de Chimborazo utilizando la tecnología .NET Core 8. Se identificó que el problema radicaba en la gestión ineficiente de los instrumentos de cooperación, que se realizaba en hojas contables, dificultando el seguimiento y análisis de datos. La solución propuesta fue desarrollar un sistema con .NET Core 8 debido a su portabilidad, reutilización de código y capacidad de integración con otras tecnologías. Se empleó la metodología XP para un desarrollo ágil y eficiente. La investigación tiene un enfoque cuantitativo que permitió analizar el rendimiento de la plataforma .Net Core una vez desarrollado el sistema según el modelo de calidad FURPS, se evaluó la eficacia, el tiempo de respuesta y el uso del CPU. Los resultados de la evaluación, proporcionados por la herramienta JMeter, mostraron un rendimiento excepcional del sistema web, superando los estándares de calidad establecidos por el modelo. Específicamente, el sistema alcanzó una eficacia del 100%, superando el 95% de satisfacción propuesto por el modelo FURPS. En cuanto al tiempo de respuesta, el promedio del sistema fue de 74,557 milisegundos, por debajo de los 5 segundos establecidos por el modelo. Además, el promedio de utilización de recursos del sistema se ubicó en un 31,66%, ligeramente por encima del 25% indicado por el modelo de calidad. Estos resultados destacan la calidad de rendimiento ofrecida por el sistema, atribuible en gran medida a la plataforma tecnológica utilizada, .NET Core 8, la adopción de esta tecnología fue fundamental para lograr este rendimiento.

**Palabras claves:** Aplicación Web, Gestión de Instrumentos de Cooperación, Metodología XP, Modelo FURPS, .Net Core 8, Rendimiento.

## ABSTRACT

The research is based on the implementation and performance evaluation of the cooperation instrument management system for the Coordination of National and International Relations of the Universidad Nacional de Chimborazo using .NET Core 8 technology. It was identified that the problem lay in the inefficient management of cooperation instruments, which was done in accounting sheets, making it difficult to track and analyze data. The proposed solution was to develop a system with .NET Core 8 due to its portability, code reuse, and ability to integrate with other technologies. The XP methodology was used for agile and efficient development. The research used a quantitative approach that allowed for the analysis of the performance of the .Net Core platform once the system was developed according to the FURPS quality model, and its effectiveness, response time, and CPU usage were evaluated. The evaluation results, provided by the JMeter tool, showed exceptional performance of the web system, surpassing the quality standards established by the model. Specifically, the system achieved an effectiveness of 100%, exceeding the 95% satisfaction proposed by the FURPS model. Regarding response time, the system's average was 74.557 milliseconds, below the 5 seconds established by the model. Additionally, the average system resource utilization was 31.66%, slightly above the 25% indicated by the quality model. These results highlight the performance quality offered by the system, primarily attributable to the technological platform used, .NET Core 8; the adoption of this technology was fundamental to achieving this performance.

**Keywords:** Web Application, Cooperation Instrument Management, XP Methodology, FURPS Model, .Net Core 8, Performance.



Reviewed by:  
Mg. Dario Javier Cutiopala Leon  
**ENGLISH PROFESSOR**  
c.c. 0604581066

## **CAPÍTULO I. INTRODUCCION**

En el presente, resulta evidente el impacto que tienen las aplicaciones web en la sociedad. Las tecnologías para su desarrollo han experimentado una notable evolución, proporcionando a los desarrolladores un amplio abanico de beneficios y herramientas. Una de estas tecnologías destacadas es .NET Core, un entorno de desarrollo de software de código abierto y multiplataforma, diseñada para lograr la máxima portabilidad entre distintos sistemas, facilitando la reutilización del código. (Nabor, 2008)

El actual problema que enfrenta la Coordinación de Relaciones Nacionales e Internacionales se centra en el sistema de gestión de instrumentos de cooperación. Los registros y la gestión de los instrumentos de cooperación se hacen con hojas de Excel, causando dificultades en el monitoreo y la recolección de información en la capacidad de análisis y generación de informes precisos y oportunos.

Tras un análisis exhaustivo de la Coordinación de Desarrollo de Sistemas Informáticos (CODESI), se decidió desarrollar un sistema para la gestión de instrumentos de cooperación con la tecnología de .NET Core. Esta elección se fundamenta en su capacidad para desarrollar aplicaciones ejecutadas en diversas plataformas, resulta muy beneficioso para el desarrollo de aplicaciones web modernas. Además, .NET Core ha sido diseñado con el objetivo de optimizar el rendimiento y el tiempo de ejecución, ofreciendo una mayor capacidad de integración con otras tecnologías y sistemas de gestión de información. Esta decisión permitirá aprovechar al máximo las ventajas que ofrece esta plataforma tecnológica avanzada.

Por tal motivo, el proyecto se enfocó en la creación del sistema de gestión de Instrumentos de Cooperación con .Net Core. Durante el desarrollo se empleó la metodología de desarrollo XP, permitió una gestión ágil y eficiente. Por último, para asegurar la calidad y rendimiento de la aplicación, se llevó a cabo una evaluación utilizando el modelo de calidad FURPS.

### **1.1. Antecedentes**

En la investigación llevada a cabo en la Universidad Técnica de Ambato en 2020 por Jessica Daniela Moreta Romero se presentó: “Sistema De Control Y Seguimiento De Los Procesos De Recursos Humanos Para Los Empleados De La Empresa Eléctrica Ambato Regional Centro Norte S. A. Mediante Una Aplicación Web Progresiva (PWA)”.

- A través del desarrollo, se pudo determinar que una PWA simplificó los procesos de control y seguimiento llevados a cabo por el departamento de Recursos Humanos. Esto se debe a que los empleados tienen acceso directo a estos procesos de manera interactiva a través de la aplicación, con ejecución instantánea.

En el año 2021, Alexis Fabián Enríquez Valle, presentó su tesis realizada en la Universidad Técnica de Ambato: “Sistema Web De Flujo De Trabajo Documental, Para La Organización En La Entrega Y Recepción De Documentos De Prácticas Preprofesionales En La Facultad De Ingeniería En Sistemas, Electrónica E Industrial”.

- Durante el desarrollo, se observó que la tecnología ASP.NET, incorporada en Visual Studio, simplifica la creación de páginas web y formularios dinámicos con menos

líneas de código. Su compatibilidad con herramientas como Ajax Control Toolkit y motores de flujo de trabajo es un aspecto destacado. Estas ventajas en conjunto favorecen la creación de proyectos sólidos y funcionales.

## **1.2. Planteamiento del Problema**

### **1.2.1. Problema y Justificación**

El problema que se planteó en este contexto se relaciona con la falta de eficiencia y precisión en la gestión de instrumentos de cooperación por parte de la Coordinación de Relaciones Nacionales e Internacionales de la Universidad Nacional de Chimborazo. Dicha coordinación trabajaba con un sistema tradicional que utilizaba hojas contables, que demostró ser ineficaz y propenso a errores. Ese enfoque dificultaba la realización de actividades esenciales, limitaba el rendimiento y disminuía la exactitud de los registros y seguimientos de los instrumentos de cooperación.

La justificación para abordar este problema radicó en la necesidad de mejorar la eficiencia y precisión en la gestión de instrumentos de cooperación. La coordinación es responsable de supervisar y gestionar acuerdos de cooperación tanto a nivel nacional como internacional, y un sistema ineficiente comprometía la calidad y la confiabilidad de esta gestión. Por lo tanto, era crucial actualizar la infraestructura tecnológica y los métodos utilizados por esta unidad para garantizar un rendimiento óptimo y la minimización de errores.

### **1.2.2. Formulación del Problema**

El problema principal que motivó este estudio se formula de la siguiente manera:

¿El uso de la tecnología .Net Core incidirá en el rendimiento del sistema de Gestión de Instrumentos de Cooperación de la Universidad Nacional de Chimborazo?

## **1.3. Objetivos**

### **Objetivo General**

Implementar el sistema de gestión de instrumentos de cooperación para el departamento de Relaciones Nacionales e Internacionales UNACH usando la tecnología .Net Core.

### **Objetivos Específicos**

- Analizar la tecnología .Net Core enfocado al desarrollo de aplicaciones web.
- Aplicar la tecnología .Net Core en el desarrollo del sistema de gestión de instrumentos de cooperación de la Universidad Nacional de Chimborazo.
- Evaluar el rendimiento del sistema mediante el modelo de calidad FURPS.

## **CAPÍTULO II. MARCO TEÓRICO**

### **2.1. Aplicación Web**

Una aplicación web es un software diseñado para funcionar en un navegador de internet. Las compañías necesitan compartir datos y ofrecer servicios de manera remota, por lo que recurren a aplicaciones web para interactuar con sus clientes de manera segura y en el momento necesario. Estas aplicaciones les permiten acceder a funciones avanzadas sin requerir la instalación o configuración de software adicional. (Amazon AWS, 2023) Las aplicaciones web son utilidades que posibilitan la realización de tareas desde un ordenador mediante el uso de internet, lo que conlleva a una disminución en el tiempo requerido para cada actividad. Este factor positivo ha contribuido a la aprobación y popularidad de este tipo de software entre los usuarios. (Ríos, 2017)

Un sistema de información se define como una serie de elementos conectados entre sí que recopilan, procesan, almacenan y distribuyen datos e información, y facilitan un mecanismo de retroalimentación para alcanzar un propósito específico. Esta retroalimentación es fundamental para que las organizaciones logren sus metas, como aumentar sus ingresos o mejorar su atención al cliente. Mediante el uso de sistemas de información, las empresas pueden potenciar sus ganancias y disminuir sus gastos operativos. (Stair, 2000)

### **2.2. Tecnologías de Desarrollo Web**

El proceso de creación de aplicaciones web se divide en dos componentes principales: el FrontEnd y el BackEnd. El FrontEnd se encarga de la parte visual de la aplicación, permitiendo la interacción del usuario con el sistema. En contraste, el BackEnd se encarga de gestionar los datos y asegurar que la información enviada desde el FrontEnd se almacene correctamente en una base de datos. Además, el BackEnd desarrolla interfaces de programación de aplicaciones (APIs) para facilitar la eficiente transferencia de datos y mejorar la experiencia del usuario. (Pérez, 2021).

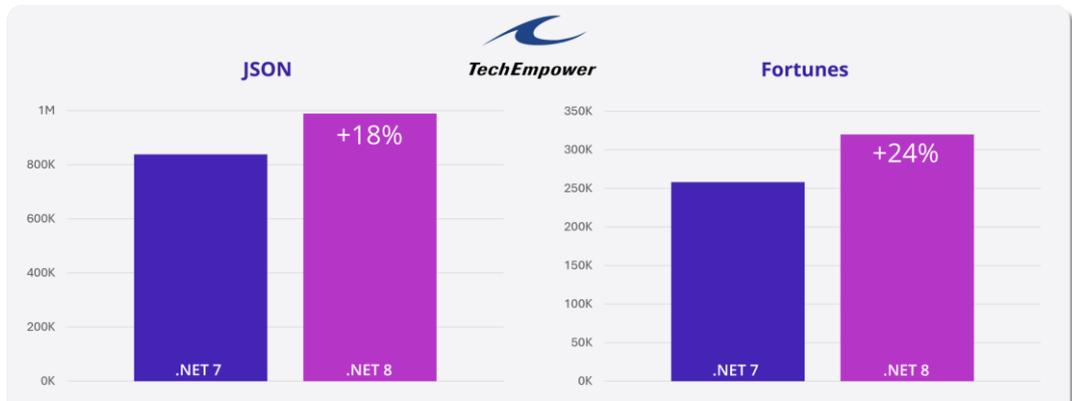
#### **2.2.1. .Net Core 8**

NET es una plataforma de desarrollo gratuita, multiplataforma y de código abierto diseñada para desarrolladores. Tiene la capacidad de compilar una amplia gama de aplicaciones y está basada en un entorno de ejecución de alto rendimiento. Es utilizada en producción por numerosas aplicaciones a gran escala. (Microsoft, 2023).

.NET 8 es la versión sucesora de .NET 7. Se proporcionará soporte técnico a largo plazo (LTS) durante un período de tres años. Introduce nuevas funcionalidades tanto en las bibliotecas principales de .NET Core como en las extensiones, además de mejorar el rendimiento y la reflexión. Integra .NET Aspire, una pila de tecnología con enfoque en la nube diseñada para desarrollar aplicaciones observables, listas para producción y distribuidas. .NET Aspire se ofrece a través de una serie de paquetes NuGet que abordan desafíos específicos relacionados con la nube, y está disponible en una versión preliminar para .NET 8. (Microsoft, 2023)

#### **Ventajas de .Net 8**

*Rendimiento:* ASP.NET Core en .NET 8 es la versión más rápida hasta la fecha. En la Figura 1 se compara .NET 7 y .NET 8. ASP.NET Core en .NET 8 es un 18 % más rápido en la prueba comparativa de JSON de Techempower y un 24 % más rápido en la prueba comparativa de Fortunes. (Daniel, 2023)



**Figura 1:** Comparativa de Rendimiento

Fuente: (Daniel, 2023)

*Serialización:* En el marco de .NET 8, se han integrado múltiples optimizaciones en la funcionalidad de serialización y deserialización de System.Text.Json. Se ha habilitado la capacidad de adaptar el tratamiento de miembros ausentes en la carga JSON, brindando así mayor flexibilidad en la personalización del proceso. Además, se han introducido diversas mejoras adicionales en el flujo de la serialización para potenciar su eficiencia y versatilidad. (Powel, 2023):

- Compatibilidad integrada con tipos adicionales
- Generador de origen
- Nuevos métodos de API de JsonNode
- Miembros no públicos
- API de deserialización de streaming
- Método de extensión WithAddedModifier
- Nuevas sobrecargas JsonContent.Create
- Inmovilización de una instancia de JsonSerializerOptions

*Compatibilidad con el proxy HTTPS:* Hasta ahora, todos los tipos de proxy que HttpClient admitía permitían un "man in the middle" para ver a qué sitio se conecta el cliente, incluso para los URI HTTPS. HttpClient ahora admite el proxy HTTPS, que establece una conexión segura entre el cliente y el proxy para que todas las solicitudes se puedan controlar con plena privacidad. Para habilitar el proxy HTTPS, establezca la variable de entorno all\_proxy o use la clase WebProxy para controlar el proxy mediante programación (Powel, 2023).

*Recolección de elementos no utilizados:* Permite configurar dinámicamente la memoria durante la ejecución. Esto resulta especialmente útil en entornos de "cloud services", donde la demanda puede variar considerablemente. Para garantizar la rentabilidad, los

servicios deben ser capaces de adaptarse a estos cambios, escalando verticalmente su consumo de recursos según sea necesario.

Anteriormente, ajustar el límite de memoria podía causar errores porque el “garbage collector” no estaba al tanto de estos cambios y podía asignar más memoria de la permitida. (Powel, 2023)

*La compatibilidad con AOT (Ahead-of-Time) nativo:* Fue incorporado en la versión inicial de .NET 7. Al desplegar un proyecto con native AOT, se crea una iteración totalmente independiente que prescinde de un entorno de ejecución suplementario; todos los componentes se fusionan en un único archivo. (Powel, 2023).

La Tabla 1 muestra el tamaño de una aplicación "Hola mundo" publicada con AOT nativa, incluyendo todo el entorno de ejecución de .NET, comparando los tamaños en .NET 7 y .NET 8.

**Tabla 1:** Comparación del tamaño de una aplicación en .NET 7 y .NET 8

Sistema operativo	.NET 7	.NET 8
Linux x64 (con -p:StripSymbols=true)	3,76 MB	1,84 MB
Windows x64	2,85 MB	1,77 MB

Fuente: (Microsoft, 2023)

*Introduce la capacidad de especificar una preferencia de optimización:* Por defecto, el compilador elige generar código rápido, considerando simultáneamente el tamaño de la aplicación. No obstante, puedes utilizar la propiedad <OptimizationPreference> de MSBuild para optimizar específicamente para una de estas opciones. Para obtener más detalles, consulta la sección de Optimización de implementaciones de AOT.

*Blazor WebAssembly:* Blazor es un framework web frontend de .NET que permite la renderización tanto en el servidor como en el cliente, todo dentro de un único modelo de programación. Permite crear interfaces de usuario interactivas y ricas en funcionalidades utilizando C#. Con Blazor, es posible compartir la lógica de la aplicación escrita en .NET entre el lado cliente y servidor. La interfaz de usuario se representa utilizando HTML y CSS, lo que garantiza la compatibilidad con todos los navegadores, incluidos los móviles. Las aplicaciones de Blazor se desarrollan mediante componentes (Microsoft, 2023).

*Razor Page:* Simplifica la programación de escenarios centrados en páginas, ofreciendo una mayor productividad en comparación con controladores y vistas. Está diseñado para facilitar la implementación de patrones comunes utilizados en aplicaciones web al compilar una aplicación. Los enlaces de modelos, las etiquetas de ayuda y los asistentes de HTML colaboran con las propiedades definidas en una clase de Razor Pages (Microsoft, 2023).

*Dev Express:* DevExpress es una entidad perteneciente a Developer Express Inc. En sus inicios, su principal labor era la creación de componentes de interfaz de usuario para Borland Delphi / C++ Builder, así como controles ActiveX para Microsoft Visual Studio. Con el transcurso del tiempo, su ámbito de acción se ha ampliado para abarcar productos orientados a desarrolladores que emplean diversas tecnologías, como Visual Studio y otras tecnologías.

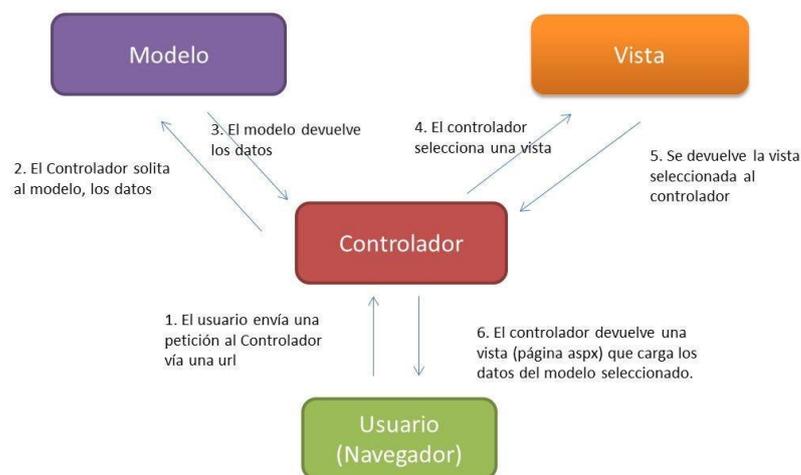
DevExpress ofrece una amplia gama de herramientas para el desarrollo de aplicaciones, algunas de las cuales incluyen:

- Las aplicaciones web, funcionan en cualquier navegador.
- Reduce los tiempos de desarrollo de las aplicaciones al proporcionar componentes que eliminan la necesidad de escribir código fuente.
- Incluye plantillas definidas para los proyectos.
- Facilita el relleno de controles mediante conexiones con almacenes externos de datos de manera rápida y sencilla.
- Proporciona una herramienta denominada Theme Builder que facilita la edición del estilo de los controles y la generación automática de la hoja de estilos.

### 2.2.2. ASP.NET Core MVC

ASP.NET Core MVC es un robusto framework para desarrollar aplicaciones web y API, empleando el patrón Modelo-Vista-Controlador (MVC), que promueve una clara separación de preocupaciones. En este enfoque, las solicitudes del usuario son dirigidas a un controlador correspondiente, el cual se comunica con el modelo para ejecutar las acciones necesarias o recuperar información. Posteriormente, el controlador elige la vista apropiada para presentar al usuario, suministrando los datos del modelo requeridos en el proceso (Microsoft, 2023).

En la Figura 2 se observan los tres componentes principales de MVC, esta división de responsabilidades facilita la escalabilidad de la aplicación, ya que resulta más sencillo codificar, depurar y probar componentes que tienen una única función (Microsoft, 2023).



**Figura 2:** Flujo de los componentes de la Arquitectura MVC (Alan, 2023)

### 2.2.3. Net Core y el desarrollo de aplicaciones web

.NET Core 8 es una solución completa para el desarrollo web moderno que maneja todas las necesidades de desarrollo web, desde el frontend hasta el backend. Permite construir experiencias web bellas y ricas en interactividad con Blazor, y APIs de backend de alto

rendimiento y servicios que son confiables y seguros. Muchos desarrolladores crean aplicaciones web ASP.NET Core que incluye cambios en la arquitectura. Sin embargo, en el mundo del desarrollo web, dos tecnologías destacan: Node.js y ASP.NET. Ambas son pilares en el desarrollo del lado del servidor, brindando soluciones de vanguardia para la industria. Con habilidades notables en el desarrollo de software, han dejado una marca significativa en el panorama tecnológico.

Node.js, con su enfoque basado en JavaScript, ofrece flexibilidad y escalabilidad, atrayendo a desarrolladores por su capacidad para manejar aplicaciones en tiempo real de manera eficiente. Por otro lado, ASP.NET Core, respaldado por Microsoft, ofrece una estructura robusta y un ecosistema integrado para construir aplicaciones web y servicios. Su sólido soporte comunitario facilita a los desarrolladores de .NET Core acelerar, optimizar y ampliar sus proyectos.

Al evaluar entre Node.js y ASP.NET Core, es crucial considerar las características específicas de cada tecnología. Node.js destaca en la velocidad de desarrollo y la interoperabilidad, mientras que ASP.NET Core sobresale en la integración con el ecosistema Microsoft y la seguridad. La elección entre ambas depende de las necesidades y requisitos particulares de cada proyecto. Al decidir qué tecnología es mejor para el desarrollo de aplicaciones web, se debe comparar y contrastar Node.js con NET Core utilizando los siguientes factores que se detallan en la Tabla 2 para ver en qué se diferencian.

**Tabla 2:** Comparativa entre .Net Core y Node.js

<b>Factores</b>	<b>.Net Core</b>	<b>Node.js</b>
<b>Rendimiento</b>	ASP.NET Core destaca en proyectos específicos que requieren resistencia, contrastando con la agilidad de Node.js en tareas menos demandantes. Aunque Node.js es adecuado para actividades menos exigentes, el continuo avance de .NET Core, un 15% más rápido con el tiempo, lo posiciona como una opción superior para los desarrolladores en general.	Node.js, basado en el motor JavaScript V8, es preferido por muchos desarrolladores debido a su capacidad para manejar eficientemente múltiples tareas y un alto tráfico de servidor, superando a otras tecnologías en rendimiento y escalabilidad.
<b>Escalabilidad</b>	ASP.NET Core destaca por su escalabilidad y es idóneo para el diseño de microservicios, simplificando el desarrollo de aplicaciones móviles multiproceso con menos carga de servidor en comparación con Node.js.	Node.js es un entorno de servidor para sistemas distribuidos que permite desarrollar software escalable basado en microservicios.  Utiliza un solo subproceso por instancia, requiriendo a los desarrolladores cargar

	Utiliza el SDK de Azure Service Fabric para escalar y ofrecer servicios complementarios eficientemente, haciendo que contratar desarrolladores de ASP.NET Core sea una opción sólida para proyectos de desarrollo de aplicaciones móviles.	los núcleos del servidor para optimizar el rendimiento mediante el módulo Cluster o un sistema de control externo.
<b>Fiabilidad</b>	ASP.NET Core destaca por su alto rendimiento y optimización de código, lo que se traduce en mejores resultados. Requiere menos codificación, lo que facilita la optimización del código para los desarrolladores y reduce el tiempo de desarrollo, contribuyendo así al ahorro presupuestario del equipo de desarrollo.	Aprovecha el motor JavaScript V8 de Google para interpretar el código JavaScript. La integración directa del código JS en el código de máquina agiliza la implementación del programa, permitiendo una ejecución más rápida y eficiente, lo que se ve aún más mejorado por el entorno de ejecución de JavaScript.
<b>Velocidad de Desarrollo</b>	Aunque ASP.NET Core simplifica y moderniza el código para los desarrolladores, la redirección de arreglos aún puede ser un proceso desafiante y requiere la creación manual de aplicaciones, lo que consume más tiempo.	Al seleccionar la tecnología para una aplicación web, la velocidad de desarrollo es crucial. Node.js destaca por su capacidad asincrónica, lo que permite manejar devoluciones de llamada de manera eficiente. Además, ofrece la flexibilidad de trabajar con piezas más pequeñas en lugar de estructuras más grandes, lo que facilita la agilidad en el desarrollo.

Fuente: (Nimap, 2023)

La comparación entre Node.js y ASP.NET Core muestra que ambas tecnologías tienen sus propias fortalezas y debilidades. Node.js destaca por su capacidad asincrónica, velocidad de desarrollo y flexibilidad para trabajar con piezas más pequeñas, lo que lo hace conveniente en el desarrollo de aplicaciones que requieren alta escalabilidad y manejo eficiente de devoluciones de llamada. Por otro lado, ASP.NET Core ofrece un alto rendimiento, optimización de código y simplificación del desarrollo, convirtiéndola en una gran opción para proyectos que priorizan la modernización y la eficiencia del código.

### 2.3. Metodologías de Desarrollo Ágil

Se emplean Metodologías Ágiles en proyectos con la meta de lograr la funcionalidad temprana del software. Este enfoque busca proporcionar al cliente versiones iniciales para que pueda evaluar y contribuir a la idea de negocio. Para cumplir con este propósito,

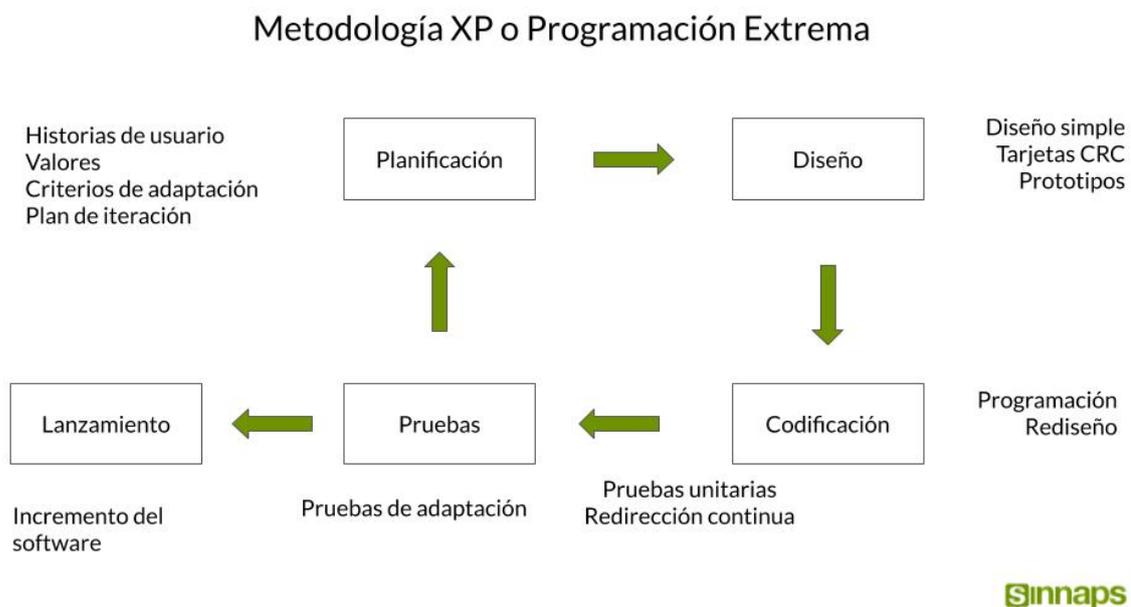
se trabaja en iteraciones sucesivas, mejorando cada interacción con base en la anterior (García, 2015).

La Metodología Ágil plantea un método de trabajo flexible con una planificación que se actualiza constantemente, a diferencia de las metodologías tradicionales que siguen una planificación detallada desde el inicio (García, 2015).

### 2.3.1. XP

Extreme Programming es un enfoque ágil para el desarrollo de software en equipo que se enfoca en un código limpio y buena comunicación. Recomendada para pequeños y medianos proyectos, especialmente aquellos en los que se cambien requerimientos frecuentemente. (Ramírez-Bedoya, Branch-Bedoya, & Jiménez-Builes, 2019).

En la Figura 3 se observan las fases de la metodología XP que se realizan en iteraciones dando oportunidad a completar y revisar el marco al final de cada iteración, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la eficiencia máxima.



**Figura 3:** Fases de la metodología XP (SINNAPS, 2020)

## 2.4. Arquitectura de Software

La arquitectura del software representa la configuración intrínseca de los módulos y su interacción dentro de un sistema informático, definiendo tanto su disposición física como lógica. Incluye la especificación de los componentes del sistema, las relaciones entre ellos y las pautas que rigen su desarrollo y mantenimiento a largo plazo. En esencia, es la infraestructura conceptual que sustenta la funcionalidad del software, determinando cómo se ensamblan y conectan los elementos para cumplir con los requisitos funcionales y no funcionales del sistema. (Garlan, 1995)

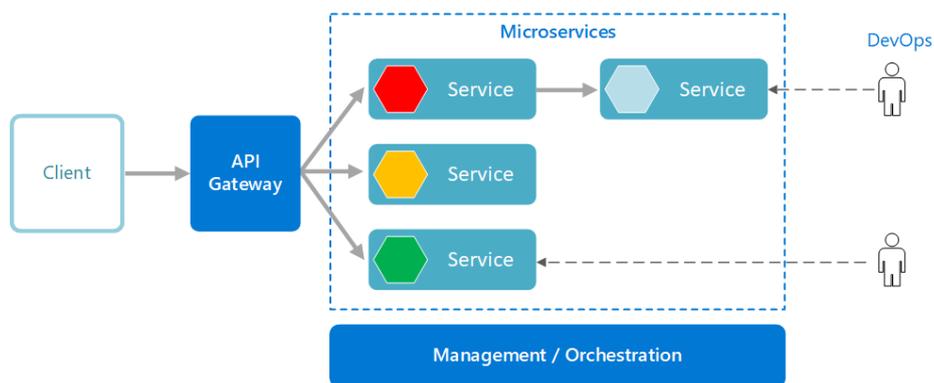
### 2.4.1. Microservicios

La arquitectura de microservicios consiste en desarrollar aplicaciones como un conjunto de pequeños microservicios, donde cada uno funciona de manera independiente en su

propio proceso y se comunica con los demás a través de mecanismos ligeros, como las API's de HTTP. (Lewis, 2014).

En este tipo de aplicaciones, la lógica se divide en varios procesos independientes que pueden utilizar diferentes tecnologías, lo que muestra un acoplamiento débil entre las diversas partes de la aplicación. En el contexto de los microservicios, realizar cambios implica reconstruir y desplegar solo el microservicio afectado. Estos microservicios cuentan con una puerta de enlace que actúa como interfaz para que los clientes se comuniquen con ellos, proporcionando una capa adicional de seguridad y funciones como autenticación o registro.

Como se observa en la Figura 4, está la arquitectura de microservicios, donde microservicios individuales se ejecutan en procesos propios y se comunican mediante API's ligeras, como HTTP. La descentralización permite un débil acoplamiento, facilitando cambios específicos al reconstruir y desplegar solo el microservicio afectado. Destaca una puerta de enlace que actúa como interfaz segura para la comunicación de clientes con los microservicios, ofreciendo funciones adicionales como autenticación o registro



**Figura 4:** Arquitectura con microservicios

## 2.5. Modelo FURPS

A nivel mundial, se han creado diversos modelos para gestionar los procesos de software con el fin de mejorar la productividad y la calidad en este campo. Uno de los modelos más reconocidos es el modelo FURPS, que se basa en Funcionalidad, Usabilidad, Confiabilidad, Rendimiento y Soporte (FURPS). Desarrollado por Hewlett-Packard en 1987, este modelo establece un conjunto de factores para evaluar la calidad del software. (Constanzo, 2014).

En proyectos de investigación, el modelo FURPS se ha empleado para evaluar el rendimiento web, el cual consiste en medir el tiempo que transcurre desde que se accede a un sitio web hasta que se muestra completamente a través del HTTP. En esta solicitud, los recursos utilizados por el servidor de aplicaciones pueden resultar afectados (Solvetic, 2015). Los parámetros del rendimiento son:

- **Eficacia:** La eficacia en el rendimiento de una aplicación web se refiere a qué tan bien la aplicación logra su propósito o resultado deseado. En el contexto del

desarrollo web, la eficacia puede referirse a qué tan bien una aplicación satisface las necesidades del usuario, proporciona una experiencia de usuario positiva y logra los objetivos comerciales. La eficacia puede verse afectada por varios factores, incluyendo la funcionalidad de la aplicación, la facilidad de uso, la fiabilidad y la capacidad de respuesta. (Kanjilal, 2022)

- **Uso de Recursos:** El uso de recursos de una aplicación web se refiere a cómo la aplicación utiliza los recursos del sistema, como la CPU y la memoria. Un uso eficiente de los recursos puede mejorar el rendimiento de la aplicación al reducir la latencia, minimizar el tiempo de carga y maximizar la capacidad de respuesta. Sin embargo, un uso ineficiente de los recursos puede resultar en un rendimiento deficiente, como tiempos de carga lentos y una mala experiencia de usuario. (Kanjilal, 2022)
- **Tiempo de Respuesta:** El tiempo de respuesta en el rendimiento de una aplicación web se refiere al lapso de tiempo que la aplicación demora en responder a las acciones ejecutadas por el usuario. Un tiempo de respuesta rápido puede mejorar la experiencia del usuario al hacer que la aplicación se sienta más receptiva y eficiente. Por el contrario, un tiempo de respuesta lento puede llevar a una experiencia de usuario negativa, ya que la aplicación podría percibirse como lenta e ineficiente. (Microsoft, 2024)

Estos parámetros pueden afectar directamente la experiencia del usuario y la eficacia de la aplicación. Una aplicación que es eficaz utiliza eficientemente los recursos y tiene un tiempo de respuesta rápido puede proporcionar una experiencia de usuario positiva y lograr sus objetivos comerciales de manera más efectiva.

## **CAPÍTULO III. METODOLOGIA**

### **3.1. Tipo de Investigación**

El proyecto de investigación se basó en un enfoque cuantitativo, que permitió analizar el rendimiento de la plataforma .Net Core una vez desarrollado el sistema de Gestión de Instrumentos de Cooperación para el departamento de Relaciones Nacionales e Internacionales de la Universidad Nacional de Chimborazo.

### **3.2. Técnicas de recolección de Datos**

La recopilación de datos para los resultados se realizó con la herramienta JMeter.

- JMeter: Esta herramienta evaluó el rendimiento del sistema y obtuvo datos numéricos de la eficacia, utilización de recursos y tiempo de respuesta.

### **3.3. Población de estudio y tamaño de muestra**

El rendimiento del sistema se evaluó con el software JMeter que brinda la posibilidad de crear escenarios con un número ilimitado de usuarios simulados, por tal motivo su población es infinita, pero está limitada a las especificaciones de la máquina donde se ejecuta, de manera que se configuró el escenario para realizar 5 bucles de pruebas de carga con 100 solicitudes al sistema en 10 segundos en los 4 módulos principales, dando un total de 2000 solicitudes.

### **3.4. Métodos de análisis y Procesamiento de datos**

Las mediciones de rendimiento se las realizó a través de JMeter, y esta aplicación produjo datos numéricos que están vinculados a la eficacia, utilización de recursos y los tiempos de respuesta.

### **3.5. Identificación de variables**

#### **3.5.1. Variable Dependiente**

Sistema de Gestión de Instrumentos de Cooperación.

#### **3.5.2. Variable Independiente**

Tecnología .NET CORE.

### 3.6. Operacionalización de Variables

En la Tabla 3 se describen el problema, objetivos y variables del tema, además de la conceptualización, dimensiones e indicadores para cada variable.

**Tabla 3:** Operacionalización de Variables

PROBLEMA	TEMA	OBJETIVOS	VARIABLES	CONCEPTUALIZACIÓN	DIMENSION	INDICADORES
¿El uso de la tecnología .Net Core incidirá en el rendimiento del sistema de gestión de instrumentos de cooperación de Relaciones Nacionales e Internacionales con tecnología .Net Core de la Universidad Nacional de Chimborazo?	Sistema de Gestión de Instrumentos de Cooperación para la Coordinación de Relaciones Nacionales e Internacionales con tecnología .Net Core	<b>General</b> <ul style="list-style-type: none"> <li>Desarrollar el Sistema de Gestión de Instrumentos de Cooperación para la Coordinación de Relaciones Nacionales e Internacionales con la tecnología .Net Core.</li> </ul>	<b>Independiente</b> Tecnología .NET CORE	Tecnología de desarrollo multiplataforma y de código abierto diseñada para que sea portátil entre plataformas a fin de permitir la reutilización del código al máximo y su uso compartido. (Nabor, 2008)	Desarrollo del sistema de Gestión de Instrumentos de Cooperación para la Coordinación de Relaciones Nacionales e Internacionales	- Número de funcionalidades - Número de módulos desarrollados - Framework
		<b>Específicos</b> <ul style="list-style-type: none"> <li>Analizar la tecnología .Net Core enfocado al desarrollo de aplicaciones web.</li> <li>Aplicar la tecnología .Net Core en el desarrollo del sistema de</li> </ul>	<b>Dependiente</b> El rendimiento del sistema de gestión de instrumentos de cooperación.	Conjunto de procesos y procedimientos diseñados para gestionar los instrumentos de cooperación.	Rendimiento	- % Eficacia. - % Tiempo de respuesta. % Utilización de recursos.

---

gestión de instrumentos  
de cooperación de la  
Universidad Nacional  
de Chimborazo.

- Evaluar el rendimiento  
del sistema utilizando  
softwares de valoración  
del rendimiento.
-

### 3.7. Desarrollo

El Sistema de Gestión de Instrumentos de Cooperación del departamento de Relaciones Nacionales e Internacionales de la Universidad Nacional de Chimborazo se desarrolló usando la tecnología .Net Core y basándose en la metodología de desarrollo ágil XP, finalmente se analizaron los datos obtenidos de la evaluación del rendimiento.

El proyecto tuvo en cuenta las fases de la metodología XP que fueron:

- **Planificación:** Se identificaron y exploraron los requisitos del proyecto mediante la creación de las historias de usuario. XP es iterativa y se ajusta a medida que avanza el proyecto.
- **Diseño:** Se definieron los diseños del sistema, además se identificaron las clases creando tarjetas CRC (Class – Responsibility - Collaboration) y se detallaron los riesgos en la codificación.
- **Codificación:** En esta fase, el equipo desarrolló el programa de una manera organizada y planificada.
- **Pruebas:** Se realizaron pruebas continuamente, al tratarse de fases iterativas es necesario que las pruebas sean constantes y automatizadas.
- **Lanzamiento:** Finalmente, se implementó el sistema siguiendo el protocolo del CODESI.

#### 3.7.1. Fase de Planificación

Durante la fase de planificación, se elaboró la explicación del sistema, se especificaron los requisitos tanto funcionales como no funcionales, se redactaron las historias de usuario, se delinearon las iteraciones y se estableció la velocidad del proyecto.

**Descripción del sistema:** La investigación se basó en la implementación de un sistema que gestione los instrumentos de cooperación para la coordinación de Relaciones Nacionales e Internacionales de la Universidad Nacional de Chimborazo, que permite registrar los convenios y dar seguimiento a sus actividades.

El sistema tiene un módulo de convenios que para registrarse debe contener parámetros fundamentales como: país, administradores, instrumento, ámbito; donde cada uno contiene su funcionalidad de CRUD, y también contiene el módulo de seguimiento de convenios porque está vinculado a un módulo de notificaciones por correo, permite notificar a las dos partes del convenio las actividades de seguimiento. El sistema también contiene un módulo de reportes, tanto de convenios como de seguimientos, que pueden ser filtrados dependiendo la información que se desee mostrar, en el caso de convenios por un intervalo de tiempo y los seguimientos por la etapa en la que se encuentren, asegurando la confiabilidad y seguridad de los datos, además, existe un módulo de configuración de roles de usuarios que limita el acceso de los datos del sistema.

El sistema se desarrolló utilizando la tecnología .Net Core 8 con el lenguaje de programación C#, en el proceso de desarrollo para almacenar la información se utilizó el motor de base de datos de Microsoft SQL Server 2019.

**Requerimientos:** Para el desarrollo del sistema de gestión de instrumentos de cooperación, se necesitó cumplir con los siguientes requerimientos:

**Requerimientos funcionales:** Los requisitos funcionales constituyen una explicación de las acciones que un sistema debe llevar a cabo. Estos requisitos indican las capacidades que el sistema final debe tener la capacidad de ejecutar. En la Tabla 4 se lista los requerimientos funcionales especificados por el cliente.

**Tabla 4:** Requisitos funcionales

<b>Código</b>	<b>Descripción de Requerimientos</b>
<b>REQ1</b>	El sistema debe contener un módulo de gestión instrumentos de cooperación.
<b>REQ2</b>	El sistema debe contener un módulo de seguimiento de actividades del convenio.
<b>REQ3</b>	El sistema debe permitir enviar notificaciones por correo para el seguimiento del convenio.
<b>REQ4</b>	El sistema debe contener un módulo de reporte de convenios.
<b>REQ5</b>	El sistema debe limitar el acceso a los datos mediante el módulo de configuración de roles de usuario.

**Requerimientos no funcionales:** Un atributo de calidad o requisito no funcional se refiere a criterios que permiten evaluar el desempeño de un sistema en lugar de analizar sus comportamientos específicos, que están asociados a los requisitos funcionales. En la Tabla 5 se lista los requerimientos no funcionales especificados por el cliente.

**Tabla 5:** Requisitos no funcionales

<b>Requerimiento</b>	<b>Descripción de Requerimientos</b>
Seguridad Confidencialidad	El módulo deberá tener un sistema jerárquico de acuerdo con el cargo que ocupa el usuario.
Eficiencia Rendimiento	El módulo deberá recopilar información de los registros de instrumentos de cooperación.
Estabilidad	Los módulos deben diseñarse con una interfaz amigable para el usuario.
Mantenibilidad	La aplicación debe permitir modificar las funcionalidades existentes y agregar nuevas funcionalidades.

**Historias de usuario:** El propósito de las historias de usuario radica en presentar un escenario de uso en dos o tres líneas, utilizando el lenguaje del cliente. En el Anexo I se describen las historias de usuario que fueron implementadas en el desarrollo del sistema

de gestión de instrumentos de cooperación. En la Tabla 6 se muestra una de las historias de usuario que se implementaron dentro del sistema.

**Tabla 6:** Historia de usuario implementada en la aplicación

<b>Historias de Usuario</b>	
<b>Numero:</b> 01	<b>Nombre:</b> Módulo de gestión instrumentos de cooperación
<b>Usuario:</b> Ricardo Pinto - Desarrollador	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada:</b> 1
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 50PE
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 25 días
<b>Descripción:</b> Como usuario necesito registrar, editar, eliminar y listar los instrumentos de cooperación para garantizar una gestión eficiente de los mismos.	
<b>Pruebas de Aceptación:</b> Que el módulo cree, edite, elimine y liste los instrumentos de cooperación.	

**Plan de Publicaciones:** Después de haber definido las historias de usuario, fue esencial elaborar un plan de lanzamientos que detalló las historias de usuario que se incluirían en cada versión del programa, junto con las fechas de publicación correspondientes. Este plan, conocido como plan de publicaciones, implicó una coordinación entre desarrolladores y clientes para establecer los momentos ideales de implementación de las historias de usuario, determinar su prioridad y especificar cuáles se incluirían en cada versión del programa. Tras la creación del plan de publicaciones, fue crucial tener claros cuatro elementos clave: los objetivos a cumplir (principalmente las historias que se desarrollarían en cada versión), la duración estimada para el desarrollo y lanzamiento de las versiones del programa, el tamaño del equipo de desarrollo y los criterios para evaluar la calidad del trabajo realizado.

En cada iteración se realizó un módulo de la aplicación dependiendo de las historias de usuario. La primera iteración correspondió al módulo de gestión de instrumentos de cooperación; este módulo abarcó la mayoría de las funcionalidades que se realizaban con los convenios (registrar, eliminar, editar, listar y asignar administradores). La segunda iteración permitió darle seguimiento a las actividades que contenían los instrumentos de cooperación, y así poder mantener un control sobre los mismos. La tercera iteración correspondió a las notificaciones por correo que se enviaron a los administradores de los convenios para darle el seguimiento correspondiente a las actividades de estos. La cuarta iteración correspondió a los reportes de los instrumentos de cooperación, donde pudieron ser filtrados por periodos de tiempo o por algún elemento en específico, y así se pudo llevar un control en el manejo de reporte de los convenios. Por último, si se deseaba limitar el acceso a los datos del sistema en la quinta iteración, se pudo realizar mediante el módulo de configuración de roles de usuario.

### 3.7.2. Iteraciones

		Name	Duration	Start	Finish
1		Diseño de la Arquitectura del Sistema	1 day	10/4/23 8:00 AM	10/4/23 5:00 PM
2		Diseño y Creación de la base de datos	1 day	10/5/23 8:00 AM	10/5/23 5:00 PM
3		Módulo de gestión instrumentos de cooperación	25 days	10/6/23 8:00 AM	11/9/23 5:00 PM
4		Módulo de seguimiento de actividades del convenio	25 days	11/10/23 8:00 AM	12/14/23 5:00 PM
5		Notificaciones por correo	20 days	12/15/23 8:00 AM	1/11/24 5:00 PM
6		Módulo de reporte de convenios	15 days	1/12/24 8:00 AM	2/1/24 5:00 PM
7		Módulo de configuración de roles de usuario	15 days	2/2/24 8:00 AM	2/22/24 5:00 PM

**Figura 5:** Plan de iteraciones

En la Figura 5 se observa el plan de iteraciones, durante este se seleccionan las historias de usuario a abordar y se definen las tareas de desarrollo correspondientes. Se eligieron seis historias de usuario para su desarrollo las mismas que tienen un tiempo de implementación de entre 50 a 30 puntos estimados. Un punto estimado tiene la duración de 4 horas En cada iteración se presenta la cantidad de tareas a realizar y la duración del desarrollo, expresada en semanas. El período de desarrollo para cada iteración depende del historial de usuario, pero se aproximó una duración entre 3 a 5 semanas.

**Primera iteración:** En la Tabla 7 se encuentran detalladas las tareas realizadas durante la primera iteración.

**Tabla 7:** Tareas de la primera iteración

Tareas de Ingeniería		
Número de Tarea	Historia de Usuario	Nombre de la Tarea
1	1	Diseño de la arquitectura del sistema.
2	2	Diseño de la base de datos.
3	2	Creación de la base de datos.
4	3	Diseño de la interfaz del módulo de convenios.
5	3	Creación de un CRUD países.
6	3	Creación de un CRUD instrumentos, clases y tipos.
7	3	Creación de un CRUD empresas.
8	3	Creación de un CRUD dependencias internas con una conexión con el API de Talento Humano.
9	3	Creación de un CRUD administradores.
10	3	Creación del módulo de convenios con asignación de empresas, dependencias y administradores de cada uno.

**Segunda iteración:** En la Tabla 8 se encuentran detalladas las tareas realizadas durante la segunda iteración.

**Tabla 8:** Tareas de la segunda iteración

<b>Tareas de Ingeniería</b>		
<b>Número de Tarea</b>	<b>Historia de Usuario</b>	<b>Nombre de la Tarea</b>
1	4	Diseño de la base de datos.
2	4	Modificación de la base de datos.
3	4	Diseño de la interfaz del módulo de seguimiento de actividades del convenio.
4	4	Creación del módulo de seguimiento de actividades.

**Tercera iteración:** En la Tabla 9 se encuentran detalladas las tareas realizadas durante la tercera iteración.

**Tabla 9:** Tareas de la tercera iteración

<b>Tareas de Ingeniería</b>		
<b>Número de Tarea</b>	<b>Historia de Usuario</b>	<b>Nombre de la Tarea</b>
1	5	Diseño del sistema de alertas por correo electrónico.
2	5	Creación del sistema de alertas por correo electrónico.
3	5	Conexión del sistema con el módulo de seguimiento de actividades.

**Cuarta iteración:** En la Tabla 10 se encuentran detalladas las tareas realizadas durante la cuarta iteración.

**Tabla 10:** Tareas de la cuarta iteración

<b>Tareas de Ingeniería</b>		
<b>Número de Tarea</b>	<b>Historia de Usuario</b>	<b>Nombre de la Tarea</b>
1	6	Diseño de la base de datos.
2	6	Modificación de la base de datos.
3	6	Diseño de la interfaz del módulo de reporte de convenios.
4	6	Diseño de plantillas de los reportes pedidos.
5	6	Creación del módulo de reportes de convenios.

**Quinta iteración:** En la Tabla 11 se encuentran detalladas las tareas realizadas durante la quinta iteración.

**Tabla 11:** Tareas de la quinta iteración

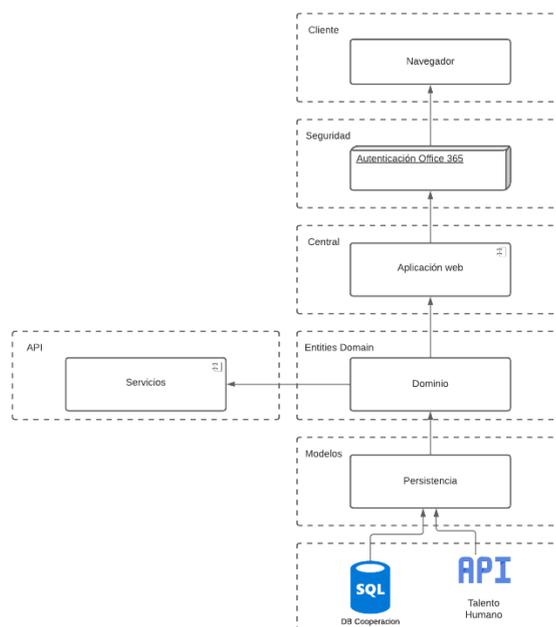
<b>Tareas de Ingeniería</b>		
<b>Número de Tarea</b>	<b>Historia de Usuario</b>	<b>Nombre de la Tarea</b>

1	7	Diseño de la base de datos.
2	7	Modificación de la base de datos.
3	7	Diseño de la interfaz del módulo de configuración de roles.
4	7	Diseño de roles de usuario y los accesos que estos garantizan.
5	7	Creación de un CRUD de personal de la universidad, utilizando el API de Talento Humano.
6	7	Creación del módulo de configuración de roles de usuarios.

### 3.7.3. Fase de Diseño

Durante la fase de diseño, se desarrollan gráficos que detallan la arquitectura del sistema, la base de datos realizada, los riesgos estimados e identificados y las tarjetas CRC necesarias para la identificación de las clases del sistema.

**Diseño de la arquitectura:** La arquitectura del Sistema de Gestión de Instrumentos de Cooperación siguió una estructura de microservicios que operó en varios servidores a medida que los recursos escalaban. La aplicación implementó una arquitectura con una capa central de tipo aplicación web, que accedió a un motor de base de datos SQL Server mediante el ORM Entity Framework Core y al API de Talento Humano de la UNACH. En la Figura 6 se observa el diseño de la arquitectura del sistema, que obtiene datos de la base de datos Cooperación y de la API de Talento Humano, además, se visualiza la capa central con la aplicación web, los datos generados en el sistema se exponen en una API para otros sistemas de la universidad.



**Figura 6:** Arquitectura del sistema

**Diseño de la base de datos:** La base de datos se creó a partir del análisis de requerimientos de los módulos del sistema, el diseño de la base de datos ayuda a gestionar adecuadamente la información que requiere la aplicación.

En la Figura 7 se observa el diseño de la base de datos, tiene las tablas de los módulos que creamos para la aplicación, también contiene los datos de los usuarios, roles y transacciones que nos ayudan para el módulo de configuración, y además se implementaron tablas para la gestión de auditorías informáticas que se realizan en la universidad.



**Tarjetas C.R.C:** Las tarjetas C.R.C se basan en el desarrollo orientado a objetos, las tarjetas son una manera efectiva para identificar y definir las clases de un sistema, se centran en las responsabilidades que deben cumplir cada clase y con que otros objetos colaboran para cumplir cada objetivo. Las tarjetas C.R.C desarrolladas para el Sistema de Gestión de Instrumentos de Cooperación se encuentran detalladas en el Anexo III.

### 3.7.4. Fase de Codificación

La codificación debe realizarse siguiendo los estándares de codificación establecidos por la Coordinación de Desarrollo de Sistemas Informáticos (CODESI), que se encuentran detallados en el Anexo IV. La adhesión a estos estándares asegura la consistencia del código y facilita su comprensión y escalabilidad. En la Tabla 12 se especifica el orden de implementación de los módulos en el sistema.

**Tabla 12:** Lista de módulos integrados al producto final

Módulos	Orden de Implementación
Módulo de Convenio	1
Módulo de Reporte	2
Módulo de Configuración	3
Módulo de Seguimiento	4

**Módulo de Convenio:** El módulo de convenio tiene varios parámetros fundamentales como: país, administradores, instrumento, tipos de convenio, clases de convenio; donde cada uno contiene su funcionalidad de CRUD.

Como se observa en la Figura 8, el controlador del módulo de convenio tiene su método (ObtenerTodosEnOtraVista) para que se pueda observar en la vista principal los convenios que están agregados con toda su información.

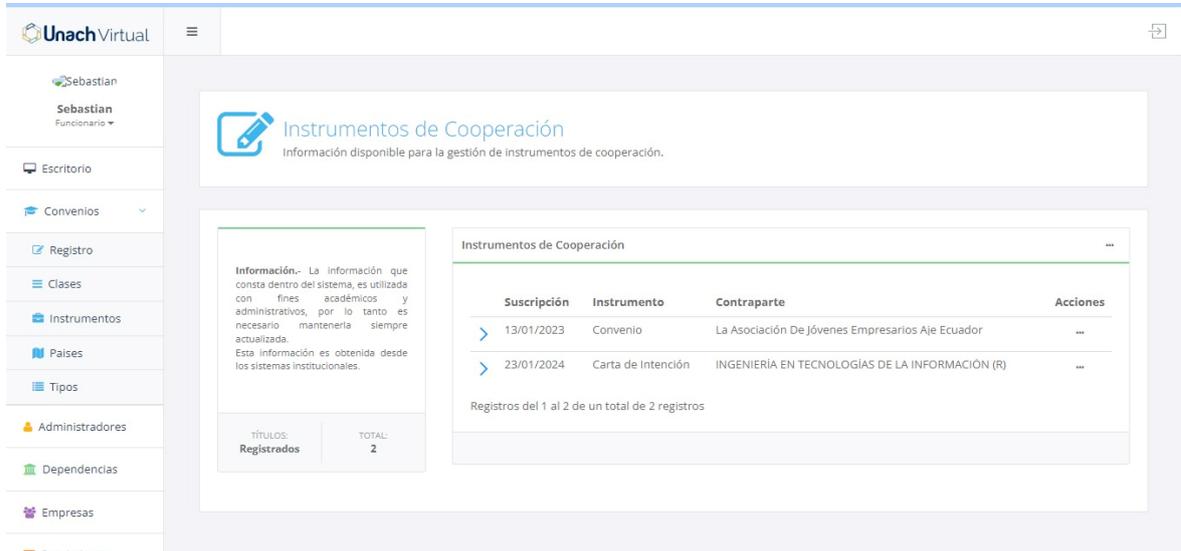
```

31 //, ApiDinarDapMap _dinarDapMap)
32 {
33     _mapper = mapper;
34     entitiesDomain = new EntitiesDomain(options);
35     logger = log.CreateLogger(typeof(ConvenioController));
36     //dinarDapMap = _dinarDapMap;
37 }
38 public IActionResult Index()
39 {
40     int IdUsuario = HttpContext.ServidorAutenticado().IdServidor;
41     RolUsuario rolusuario = entitiesDomain.RolUsuarioRepositorio.BuscarPor(x => x.IdUsuario == IdUsuario).FirstOrDefault();
42
43     var modelo = entitiesDomain.ConvenioRepositorio.ObtenerTodosEnOtraVista(
44         m => new ConvenioViewModel
45         {
46             Id = m.Id,
47             IdInstrumento = m.IdInstrumento,
48             IdPais = m.IdPais,
49             IdTipoConvenio = m.IdTipoConvenio,
50             IdAmbito = m.IdAmbito,
51             IdClaseConvenio = m.IdClaseConvenio,
52             IdRol = rolusuario.IdRol,
53             IdUsuario=IdUsuario,
54
55             Nombre = m.Nombre,
56             Objeto = m.Objeto,
57             Duracion = m.Duracion,
58             Suscripcion = m.Suscripcion,
59             Expiracion = m.Expiracion,
60             Estado = (bool)m.Estado,
61             Seguimiento = (bool)m.Seguimiento,
62
63             Instrumento = m.IdInstrumentoNavigation.Nombre,
64             Pais = m.IdPaisNavigation.Nombre,
65             TipoConvenio = m.IdTipoConvenioNavigation.Nombre,
66             Ambito = m.IdAmbitoNavigation.Nombre,
67             ClaseConvenio = m.IdClaseConvenioNavigation.Nombre,
68

```

**Figura 8:** Controlador de Módulo de Convenio

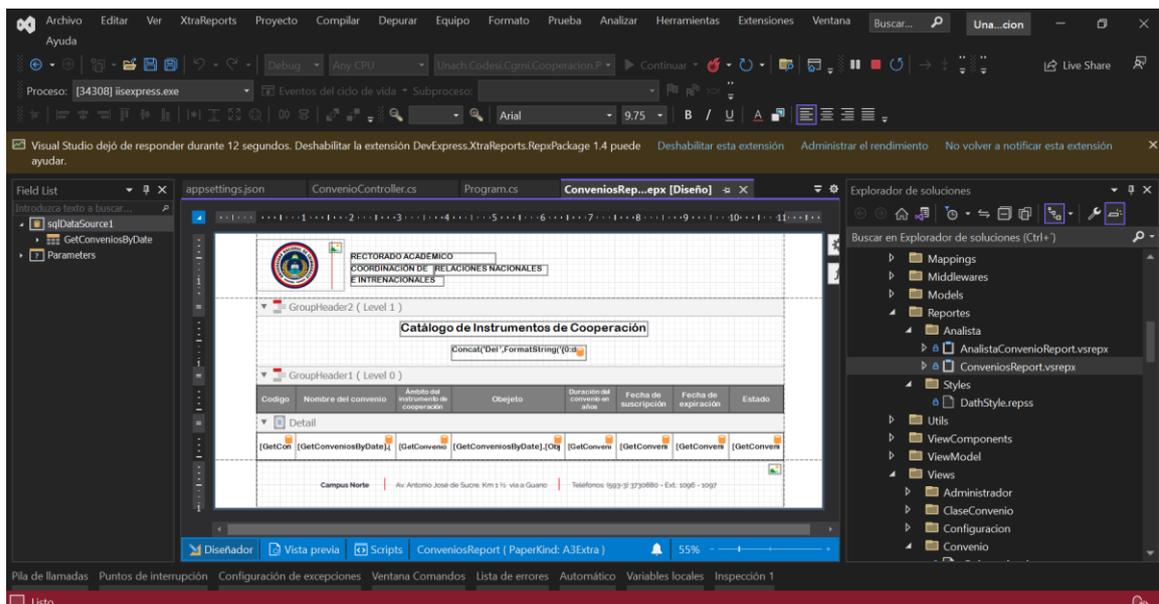
Como se observa en la Figura 9, la vista principal del módulo de convenio tiene un convenio con la información detallada y los administradores asignados al mismo. También en la parte izquierda se ven los submódulos que están ligados al módulo principal: países, instrumentos, tipos y clases.



**Figura 9:** Modulo de Convenio – Vista Principal

**Módulo de Reportes:** El módulo de reportes filtra por fecha o algún elemento en específico los convenios que están registrados en el sistema

Como se observa en la Figura 10, el diseño del reporte que se filtra por fecha mediante DevExpress.



**Figura 10:** Diseño de Reportes - DevExpress

Como se observa en la Figura 11, el reporte muestra un solo convenio cuando se filtra por fecha, esto quiere decir que, en el rango de fecha que se ha puesto, ese es el único convenio con esa fecha de suscripción que existe en el sistema.

**Figura 11:** Módulo de Reportes – Reporte Convenios

**Módulo de Configuración:** El módulo de configuración limita el acceso a los datos del sistema mediante roles de usuario, esto asegura la confiabilidad al tratar información institucional.

Como se observa en la Figura 12, el Index del módulo de configuración se escribió con una sintaxis de Razor que se utilizó en todo el Front-End del proyecto.

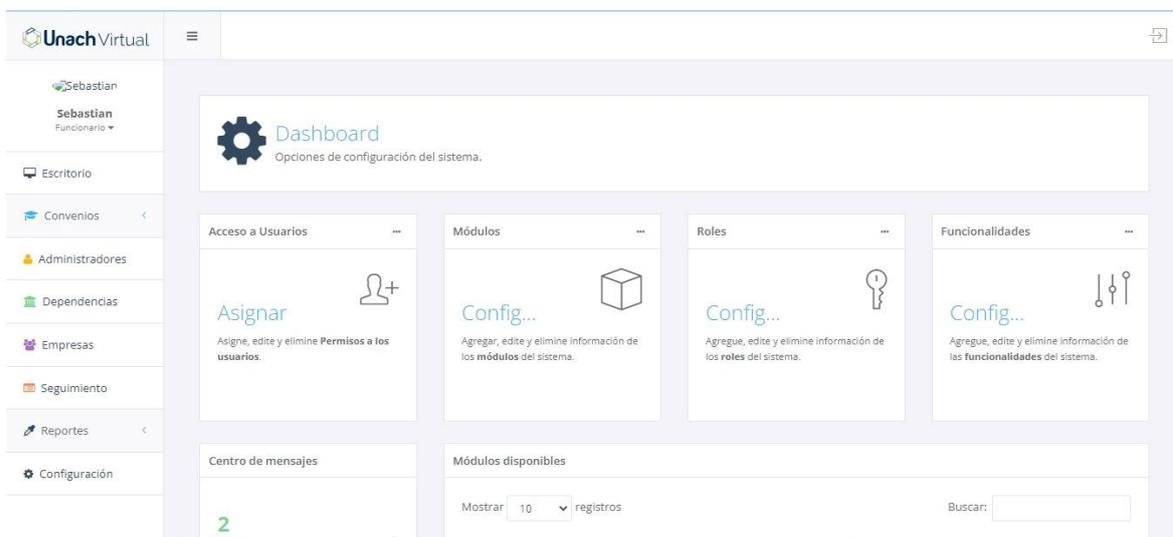
```

1  ViewData["Title"] = "Dashboard";
2
3
4
5  del DashboardConfiguracionViewModel
6
7  rtial name="_HeaderNombreVista" model="@((new HeaderViewModel() {Titulo=ViewData["Title"].ToString(), Subtitulo="Opciones de configurac
8
9  div class="content animate-panel">@
10
11
12
13  <div class="content animate-panel">
14
15
16  <div class="row">
17  <div class="col-lg-3">
18  <div class="hpanel">
19  <div class="panel-heading hbuilt">
20  Acceso a Usuarios
21  <div class="pull-right">
22  <div class="btn-group">
23  <button data-toggle="dropdown" class="btn btn-clear btn-xs dropdown-toggle" aria-expanded="false"><span class=
24  <ul class="dropdown-menu dropdown-menu-right">
25  <li><a asp-action="NuevoUsuarioActivo" asp-controller="Configuracion">Nuevo</a></li>
26  <li class="divider"></li>
27  <li><a asp-action="Administradores" asp-controller="Configuracion">Administradores</a></li>
28  <li><a asp-action="Usuarios" asp-controller="Configuracion">Usuarios</a></li>
29
30  </ul>
31  </div>
32  &nbsp;</div>
33  </div>
34  <div class="panel-body text-left h-200">
35  <div class="stats-title pull-left">
36  <h4>&nbsp;</h4>
37  </div>
38  <div class="stats-icon pull-right">
39  <i class="pe-7s-add-user fa-4x"></i>
40  </div>
41

```

**Figura 12:** Index del Módulo de Configuración

Como se observa en la Figura 13, el módulo de configuración muestra las funciones de acceso a usuarios, creación de módulos, roles y funcionalidades para limitar el acceso a los usuarios a los distintos módulos y funcionalidades de sistema.



**Figura 13:** Modulo de Convenio – Vista Principal

### 3.7.5. Fase de Pruebas

Se realizaron pruebas funcionales para verificar el funcionamiento del código desarrollado previo a ser implementado. En conjunto con personal de la Coordinación de Relaciones Nacionales e Internacionales se desarrollaron pruebas de casos de uso, planteando situaciones reales y evaluando el comportamiento del sistema.

Las pruebas se las realizaron al final de cada tarea de desarrollo, verificando el cumplimiento de los requerimientos. En las historias de usuario se describieron las “Pruebas de aceptación”, detallando las características que debe cumplir el sistema:

- Que el módulo de cooperación cree, edite, elimine y liste los instrumentos.
- El sistema debe permitir el seguimiento de las actividades de los convenios en tiempo real, para que ambas partes tengan el control sobre el cumplimiento
- El sistema debe enviar notificaciones por correo, a falta de un o dos días que se requiera confirmar la actividad
- El sistema genera reportes filtrando por fechas que el usuario desee.
- El sistema debe poder asignar roles de usuario para limitar el acceso a los datos

### 3.7.6. Fase de Lanzamiento

En la fase final del proceso, una vez se completó el desarrollo del sistema de instrumentos de cooperación, la última versión fue cargada en Team Explorer, una herramienta colaborativa de desarrollo. El director del CODESI asumió el rol clave de poner en producción el sistema, llevando a cabo las acciones necesarias para asegurar su implementación exitosa. Su participación garantizó la transición efectiva del sistema del entorno de desarrollo a un entorno operativo, asegurando su disponibilidad para su uso práctico y funcional. Este paso marcó el cierre exitoso de la fase de lanzamiento del proyecto.

## CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

### 4.1. Resumen de las pruebas de carga

Para realizar las pruebas de carga en JMeter se utilizaron peticiones HTTP con el método POST, según el módulo a evaluar se enviaron los parámetros correspondientes a su contenido en la base de datos. En la Tabla 13 se muestra la cantidad de procedimientos llevados a cabo durante la evaluación de la aplicación. En total, se realizaron 5 bucles de pruebas de carga con 100 solicitudes al sistema en 10 segundos en los 4 módulos principales (convenio, seguimiento, reportes y configuración).

**Tabla 13:** Prueba de carga en los módulos principales

Modulo	N° de solicitudes totales
Convenio	500
Configuración	500
Seguimiento	500
Reporte	500

Como se observa en la Figura 14, los datos indican que el sistema tiene buen rendimiento, el porcentaje de error es cero en todos los módulos y el rendimiento total es significativo. Sin embargo, se puede observar cierta variabilidad en los tiempos medios y máximos entre los distintos módulos, lo que podría sugerir áreas específicas que podrían optimizarse para mejorar aún más el rendimiento global del sistema.



**Figura 14:** Tabla de resumen de la prueba de carga

## 4.2. Eficacia

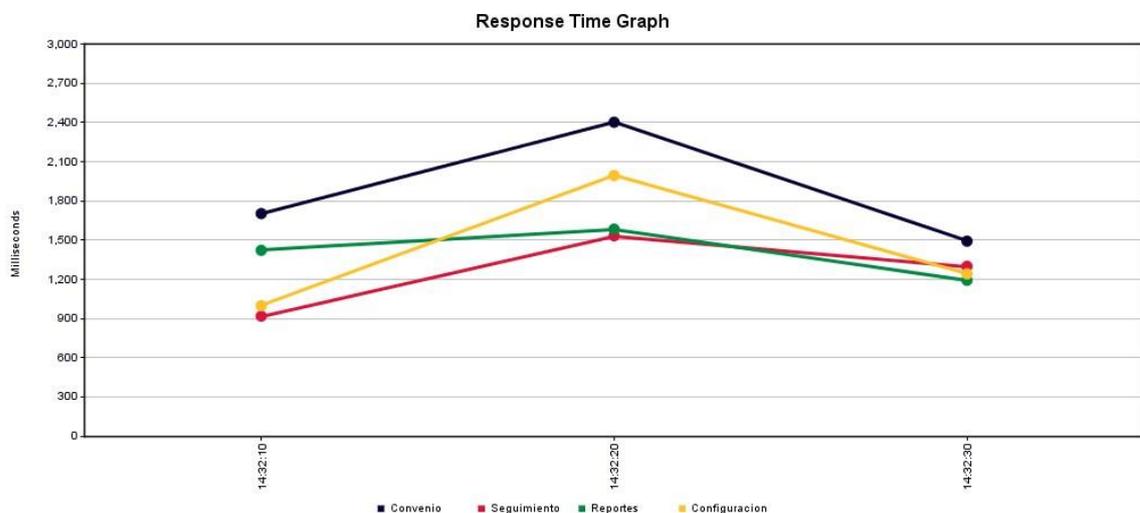
En la Tabla 14 presentaron de manera visual el grado de satisfacción alcanzado por la aplicación, tras analizar la cantidad de solicitudes exitosas. En este contexto, se observa que la eficiencia del producto alcanzó un 100% después de realizar los 5 bucles de 100 solicitudes al sistema en un periodo de tiempo de 10 segundos en cada uno de los módulos.

**Tabla 14:** Requerimientos solicitados al aplicativo

Modulo	Requerimientos Exitosos	Requerimientos Fallidos
Convenio	500	0
Configuración	500	0
Seguimiento	500	0
Reporte	500	0

## 4.3. Tiempo de Respuesta

Como se observa en la Figura 15, al recibir varias solicitudes en los diversos módulos, exhibió un tiempo promedio de respuesta que osciló entre 900 y 2400 milisegundos, tomando en cuenta que el módulo de Convenio tuvo un pico alto al inicio de prueba de carga y posteriormente se estabilizó.



**Figura 15:** Grafica de tiempos de respuesta

## 4.4. Uso de recursos

Durante las pruebas de carga se analizó el consumo de recursos del sistema, en la Tabla 15 se muestran los porcentajes de consumo de cada uno de estos indicadores. Esto significa que el sistema no ocupó un gran porcentaje de recursos al momento de realizar las pruebas.

**Tabla 15:** Uso de recursos

<b>Indicador</b>	<b>Porcentaje de consumo</b>
Uso del CPU	16%
Uso de la memoria RAM	40%
Uso de disco duro	3%

#### 4.5. Comparación con el modelo FURPS

Después de que los usuarios accedieron a la aplicación simultáneamente, en la Tabla 16 se muestra que el rendimiento en el estado inicial de la aplicación fue excelente. Esto significa que las pruebas de los módulos fueron exitosas y no se registró ninguna falla que obstaculizara el funcionamiento del sistema.

**Tabla 16:** Comparación de valores del modelo FURPS vs. valores del estudio

<b>Parámetro</b>	<b>Modelo FURPS</b>	<b>Valores del estudio</b>
Eficacia	95%	100%
Tiempo de Respuesta	5s	1408ms
Uso de recursos (CPU)	25%	16%

#### 4.6. Discusión

Se optó por la herramienta JMeter para llevar a cabo una exhaustiva evaluación de los indicadores de rendimiento en el sistema web de servicios académicos de la Universidad Nacional de Chimborazo. La eficacia del sistema se destacó al alcanzar un 100%, superando la satisfacción del 95% propuesta por el modelo FURPS. En relación con el tiempo de respuesta, el promedio del sistema se situó en 74,557 milisegundos, una cifra significativamente inferior a los 5 segundos establecidos por el modelo. Asimismo, el promedio de utilización de recursos del sistema se ubicó en un 31,66%, superando ligeramente el 25% indicado por el modelo de calidad. Estos resultados resaltan de manera clara la calidad de rendimiento ofrecida por el sistema. Este rendimiento excepcional puede atribuirse en gran medida a la plataforma tecnológica utilizada, .NET Core 8. La adopción de esta tecnología ha contribuido significativamente a la eficiencia y rendimiento del aplicativo web, permitiendo alcanzar y superar los estándares establecidos por el modelo de calidad FURPS.

## **CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES**

### **CONCLUSIONES**

- La investigación acerca de la tecnología .Net Core 8 para el desarrollo de aplicaciones web reveló las nuevas bondades que trae consigo esta actualización, destacando su idoneidad para la creación de sistemas web robustos y eficientes. Algunas razones claves por las que se eligió .NET Core 8 en aplicaciones web son: su arquitectura sencilla y modular, la unificación de API web y UI web, su diseño para la capacidad de prueba y la posibilidad de usar C# en el navegador a través de Blazor.
- La implementación de la tecnología .NET Core 8 en el desarrollo del sistema de gestión de instrumentos de cooperación de la Universidad Nacional de Chimborazo se llevó a cabo siguiendo la metodología de desarrollo XP, utilizando el gestor de base de datos SQL Server Management Studio 19, Power BI para la implementación de estadísticas, DevExtreme para la generación de reportes y Visual Studio 2022 que proporcionó un entorno de desarrollo integrado equipado con una amplia gama de herramientas y características que facilitan la creación, depuración y despliegue de la aplicación .NET.
- Los resultados obtenidos de las pruebas de rendimiento con JMeter demuestran que el sistema ha superado satisfactoriamente los criterios establecidos por el modelo de calidad FURPS, con una eficacia del 95%, un tiempo de respuesta de 1408 ms y un uso del CPU del 16%, cumpliendo los objetivos del sistema.

## **RECOMENDACIONES**

- Dada la eficacia revelada en la investigación sobre .Net Core 8, se sugiere profundizar en la exploración de sus características específicas que se alinean con los requisitos de proyectos futuros. Además, se recomienda mantenerse actualizado sobre las futuras versiones y actualizaciones de .Net Core para aprovechar las últimas mejoras y características.
- La implementación del sistema de gestión de instrumentos de cooperación ha demostrado ser positiva. Se aconseja considerar la tecnología .NET Core 8 para futuros proyectos de desarrollo de sistemas en la Universidad Nacional de Chimborazo.
- A pesar de los resultados positivos obtenidos mediante la evaluación del rendimiento a través del modelo FURPS, se insta a mantener un enfoque proactivo en la monitorización continua del sistema. Se recomienda establecer un calendario regular de evaluaciones utilizando modelos de calidad y rendimiento, identificando y abordando posibles áreas de mejora. La retroalimentación de los usuarios también puede ser valiosa para ajustar la experiencia del sistema según las necesidades cambiantes.

## BIBLIOGRAFÍA

- Alan, M. Q. (2023). *Implementación de un Sistema Web bajo el Patrón de Arquitectura de Software (MVC) para mejorar el Proceso de Gestión Académica en Instituciones Educativas Peruanas*. Lima.
- Amazon AWS. (2023). *¿Qué es una aplicación web?* Obtenido de <https://aws.amazon.com/es/what-is/web-application/>
- Casierra Cavada, J. &.-K.-I. (2019). ANÁLISIS COMPARATIVO DE METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE: UNA REVISIÓN BIBLIOGRÁFICA. .
- Chen, S.-C. (2003). *A Three-Tier System Architecture Design and Development for Hurricane Occurrence Simulation*. Miami, Estados Unidos.
- Constanzo, M. A. (2014). Comparación de modelos de calidad, factores y métricas en el ámbito de la ingeniería de software. *Dialnet*, págs. 8-9.
- dotcom-monitor. (2022). *loadview*. Obtenido de <https://www.loadview-testing.com/es/blog/5-ejemplos-de-pruebas-de-carga-de-jmeter/>
- García, I. M. (2015). Metodologías de desarrollo software. ¿tradicional o ágil? *Moleqla*, 17-18.
- Garlan, D. &. (1995). Introduction to the Special Issue on Software Architecture DRAFT. Carnegie Mellon University.
- Jimmy Rolando Molina Ríos, J. A.-S. (2021). COMPARATIVA DE METODOLOGÍAS DE DESARROLLO DE. *3C Tecnología. Glosas de innovación aplicadas a la pyme*, 73-93.
- Lewis, J. &. (2014). *Microservices*. Obtenido de <https://martinfowler.com/articles/microservices.html>
- Microsoft. (2023). *¿Qué es .NET? Introducción e información general*. Obtenido de <https://learn.microsoft.com/es-es/dotnet/core/introduction>
- Microsoft. (2023). *Blazor de ASP.NET Core*. Obtenido de <https://learn.microsoft.com/es-es/aspnet/core/blazor/?view=aspnetcore-8.0>
- Microsoft. (2023). *Información general de ASP.NET Core MVC*. Obtenido de <https://learn.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-8.0>
- Microsoft. (2023). *Tutorial: Introducción a Razor Pages en ASP.NET Core*. Obtenido de <https://learn.microsoft.com/es-es/aspnet/core/tutorials/razor-pages/razor-pages-start?view=aspnetcore-8.0&tabs=visual-studio>
- Pérez Ibarra Susana Graciela, Q. J. (2021). Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd. *XXIII Workshop de Investigadores en Ciencias de la Computación*, 347-350.
- Powel, M. (2023). *Microsoft*. Obtenido de <https://learn.microsoft.com/es-es/dotnet/core/whats-new/dotnet-8#source-generator>
- Ralph M. Stair, G. W. (2000). *Principios de sistemas de información: enfoque administrativo*. International Thomson Editores.
- Ramírez-Bedoya, D. L., Branch-Bedoya, J. W., & Jiménez-Builes, J. A. (2019). METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA PLATAFORMAS EDUCATIVAS ROBÓTICAS USANDO ROS-XP. *Revista Politécnica*, 55-69.
- Ríos, J. R., Ordóñez, M. P., Segarra, M. J., & Zerda, F. G. (2017). Metodologías de desarrollo en aplicaciones web. *3c Tecnología: glosas de innovación aplicadas a la pyme*, 54-71
- Sandra Milena Velásquez Restrepo, J. D.-M.-A.-M.-Z.-M. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Cintex*, 13-23.

SINNAPS. (2020). *METODOLOGÍA XP O PROGRAMACIÓN EXTREMA*. Obtenido de <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>

Solvetic. (2015). *Solvetic tutoriales*. Obtenido de <https://www.solvetic.com/tutoriales/article/1502-c%C3%B3mo-medir-el-rendimiento-de-una-aplicaci%C3%B3n-web/>

Universidad Nacional de Chimborazo. (2020). *Universidad Nacional de Chimborazo*. Obtenido de <https://www.unach.edu.ec/images/reglamentos/2020/6.junio/REGLAMENTO%20DE%20INSTRUMENTOS%20DE%20COOPERACION.pdf>

Universidad Nacional de Chimborazo. (2023). *Universidad Nacional de Chimborazo*. Obtenido de <https://www.unach.edu.ec/coordinacion-de-relaciones-nacionales-e-internacionales/>

Valencia, S., Villa, A., & Ocampo, A. (2009). Modelo de Calidad de Software. *Universidad Tecnológica de Pereira*, 172-173.

Wagner, B. (2023). *Microsoft*. Obtenido de <https://learn.microsoft.com/es-es/dotnet/csharp/linq/>

Yazmin Nabor, M. M. (2008). *Introducción a .net*.

# **ANEXO I**

## **HISTORIAS DE USUARIO**

Las historias de usuario reflejan los requisitos del software y se describen en una o dos frases empleando el lenguaje cotidiano del usuario. A continuación, se detallan algunas historias de usuario para el sistema de gestión de instrumentos de cooperación.

**Tabla 17:** Historia de Usuario Diseño de la Arquitectura

<b>Historias de Usuario</b>	
<b>Numero: 01</b>	<b>Nombre:</b> Diseño de la Arquitectura
<b>Usuario:</b> Ricardo Pinto - Desarrollador	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada:</b> 1
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 2 PE
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 día
<b>Descripción:</b> Se realizará el diseño de la arquitectura del sistema.	
<b>Pruebas de Aceptación:</b>	

**Tabla 18:** Historia de Usuario Diseño y creación de la base de datos

<b>Historias de Usuario</b>	
<b>Numero: 02</b>	<b>Nombre:</b> Diseño y creación de la base de datos
<b>Usuario</b>	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada:</b> 1
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 2 PE
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1 días
<b>Descripción:</b> Se realizará el diseño de la base de datos del sistema.	
<b>Pruebas de Aceptación:</b>	

**Tabla 19:** Historia de Usuario Módulo de gestión instrumentos de cooperación

<b>Historias de Usuario</b>	
<b>Numero: 03</b>	<b>Nombre:</b> Módulo de gestión instrumentos de cooperación
<b>Usuario:</b> Ricardo Pinto - Desarrollador	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada:</b> 1
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 50PE
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 25 días
<b>Descripción:</b> Como usuario necesito registrar, editar, eliminar y listar los instrumentos de cooperación para garantizar una gestión eficiente de los mismos.	
<b>Pruebas de Aceptación:</b> Que el módulo cree, edite, elimine y liste los instrumentos de cooperación.	

**Tabla 20:** Historia de Usuario Módulo de seguimiento de actividades del convenio

<b>Historias de Usuario</b>	
<b>Numero: 04</b>	<b>Nombre:</b> Módulo de seguimiento de actividades del convenio
<b>Usuario:</b> Sebastián Reyes - Desarrollador	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada: 2</b>
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 50 PE
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 25 días
<b>Descripción:</b> Como usuario quiero hacer el seguimiento de las actividades requiere el convenio, para así llevar control del cumplimiento de estos.	
<b>Pruebas de Aceptación:</b> El sistema debe permitir el seguimiento de las actividades de los convenios en tiempo real, para que ambas partes tengan el control sobre el cumplimiento.	

**Tabla 21:** Historia de Usuario Notificaciones por correo

<b>Historias de Usuario</b>	
<b>Numero: 05</b>	<b>Nombre:</b> Notificaciones por correo
<b>Usuario:</b> Sebastian Reyes - Desarrollador	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada: 3</b>
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 40 PE
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 20 días
<b>Descripción:</b> Como usuario quiero recibir notificaciones por correo para anunciar el seguimiento de las actividades del convenio.	
<b>Pruebas de Aceptación:</b> El sistema debe enviar notificaciones por correo, a falta de un o dos días que se requiera confirmar la actividad.	

**Tabla 22:** Historia de Usuario Módulo de reporte de convenios

<b>Historias de Usuario</b>	
<b>Numero: 06</b>	<b>Nombre:</b> Módulo de reporte de convenios
<b>Usuario:</b> Ricardo Pinto - Desarrollador	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada: 4</b>
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 30 PE
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 15 días
<b>Descripción:</b> Como usuario quiero tener reportes de los convenios que se han realizado a lo largo de periodo académico u otros periodos.	
<b>Pruebas de Aceptación:</b> El sistema genera reportes filtrando por fechas que el usuario desee.	

**Tabla 23:** Historia de Usuario Módulo de configuración de roles de usuario

<b>Historias de Usuario</b>	
-----------------------------	--

---

<b>Numero: 07</b>	<b>Nombre:</b> Módulo de configuración de roles de usuario
<hr/>	
<b>Usuario:</b> Sebastián Reyes - Desarrollador	
<hr/>	
<b>Modificación de Historia Números</b>	<b>Iteración Asignada:</b> 5
<hr/>	
<b>Prioridad de Negocio:</b> Alta	<b>Puntos Estimados:</b> 30 PE
<hr/>	
<b>Riesgo de Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 15 días
<hr/>	
<b>Descripción:</b> Como usuario quiero limitar el acceso de los datos mediante roles de usuario.	
<hr/>	
<b>Pruebas de Aceptación:</b> El sistema debe poder asignar roles de usuario para limitar el acceso a los datos.	
<hr/>	

**ANEXO II**  
**TARJETAS C.R.C**

**Tabla 24: CRC Convenio**

<b>Clase: Convenio</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Gestionar información del convenio. Calcular el tiempo de duración e identificar el estado de cada convenio.	Colabora con las siguientes clases para llenar la información de cada convenio: AmbitoInstrumento ClaseConvenio Pais Continente TipoConvenio Instrumento EmpresaConvenio

**Tabla 25: CRC EmpresaConvenio**

<b>Clase: EmpresaConvenio</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Asignar a cada convenio por lo menos un administrador de tipo institucional y uno de la empresa. Asignar a cada convenio por lo menos una empresa o institución y una dependencia de la universidad.	Colabora con las siguientes clases para llenar asignar información a cada convenio: Administrador TipoAdministrador Empresa TipoEmpresa Convenio

**Tabla 26: CRC Usuario**

<b>Clase: Usuario</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Gestionar la información de los usuarios permitidos en el sistema.	Esta clase no tiene colaboradores.

**Tabla 27: CRC Rol**

<b>Clase: Rol</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Gestionar la información de los roles del sistema.	Colabora con las siguientes clases para asignar información a cada rol: Sistema

**Tabla 28:** CRC Transacción

---

**Clase:** Transacción

---

<b>Responsabilidad</b>	<b>Colaborador</b>
Gestionar la información de las transacciones realizadas en el sistema, forma parte del módulo de auditoría de la universidad.	Colabora con las siguientes clases para asignar información a cada transacción: Módulo Sistema

---

**Tabla 29:** CRC RolTransacción

---

**Clase:** RolTransacción

---

<b>Responsabilidad</b>	<b>Colaborador</b>
Asignar los roles a las transacciones.	Colabora con las siguientes clases para unir la información entre roles y transacciones: Transacción Rol

---

**Tabla 30:** CRC RolUsuario

---

**Clase:** RolUsuario

---

<b>Responsabilidad</b>	<b>Colaborador</b>
Asignar los roles a los usuarios.	Colabora con las siguientes clases para unir la información entre roles y usuarios: Usuario Rol

---

**ANEXO III**  
**ACTA ENTREGA-RECEPCIÓN**

## INFORME DE ENTREGA DEL SISTEMA DE INSTRUMENTOS DE COOPERACIÓN

Riobamba, 09 de mayo del 2024

Este informe certifica que el Sistema de Gestión de Instrumentos de Cooperación para la Coordinación de Relaciones Nacionales e Internacionales UNACH ha sido implementado con éxito como parte del trabajo de investigación titulado "Sistema de Gestión de Instrumentos de Cooperación para la Coordinación de Relaciones Nacionales e Internacionales Unach usando tecnología .Net Core".

El sistema desarrollado consta de 5 módulos principales:

- **Convenios:** Permite la gestión de instrumentos de cooperación y sus componentes, incluyendo la creación, modificación y eliminación de registros.
- **Seguimiento:** Facilita el seguimiento de los instrumentos de cooperación, permitiendo visualizar el estado, las actividades y obligaciones asociadas.
- **Reportes:** Proporciona funcionalidades para la generación de informes y reportes relacionados con los instrumentos de cooperación y sus seguimientos.
- **Notificaciones:** Gestiona las notificaciones asociadas con las fechas respectivas para la realización de los seguimientos.
- **Configuración de Usuarios:** Permite la configuración de usuarios y sus roles dentro del sistema.

En el marco de esta entrega, se proporciona lo siguiente:

- **Código fuente del sistema,** desarrollado en tecnología .Net Core, disponible para su revisión y modificación.
- **Scripts de la base de datos utilizada por el sistema,** que incluyen la estructura de las tablas, procedimientos almacenados y tareas necesarias para su funcionamiento.
- **Manual de usuario detallado,** que describe las funcionalidades del sistema y cómo utilizarlas de manera efectiva.
- **Indicaciones técnicas,** que se encuentran comentadas dentro del código fuente, proporcionando orientación sobre el funcionamiento interno del sistema.

La implementación de este sistema representa un avance significativo en la gestión de instrumentos de cooperación de la Universidad Nacional de Chimborazo, facilitando la coordinación y seguimiento de relaciones tanto nacionales como internacionales.



Ricardo Francel Pinto Matos

ESTUDIANTE DE TI



Sebastián Alejandro Reyes Solís

ESTUDIANTE DE TI



Ing. Henry Javier Paca Quinaluiza

COORDINADOR CODESI



**ANEXO IV**  
**MANUAL DE USUARIO**



## **Propósito del Documento**

El propósito de este documento es ayudar al usuario con el uso adecuado del "Sistema de Gestión de Instrumentos de Cooperación", desarrollado para la Coordinación de Relaciones Nacionales e Internacionales

## Índice

Íconos usados en el Manual.....	62
Presentación de la Solución.....	64
2.1 Breve Descripción .....	64
2.2 Íconos Descriptivos .....	64
2.3 Usuarios del Sistema .....	64
Acceso a la Aplicación .....	65
3.2 Dirección .....	65
3.2 Autenticación.....	65
Estructura de la Aplicación.....	66
4.1 Interfaz.....	67
4.2 Menús .....	67
4.3 Funcionalidades .....	67
Función 01 / Creación de un Instrumento de Cooperación .....	69
5.1 Gestión de Países .....	69
5.2 Gestión de Clases de Instrumentos de Cooperación.....	69
5.3 Gestión de Instrumentos .....	70
5.4 Gestión de Tipos de Instrumentos de Cooperación .....	71
5.5 Gestión de Dependencias.....	72
5.6 Gestión de Empresas .....	73
5.7 Gestión de Administradores .....	74
5.8 Gestión de Instrumentos de Cooperación .....	75
Función 02 / Gestión de Seguimientos .....	78
6.1 Creación de Actividades, Evidencias y Obligaciones .....	78
Función 03 / Reportes.....	80
7.1 Reporte de Seguimiento Individual .....	80
7.2 Reporte General de Instrumentos de Cooperación .....	81

7.3 Reporte de Convenios por Clase .....	82
7.4 Reporte de Convenios por Tipo.....	82
7.5 Reporte de Cumplimiento.....	83
Función 04 / Configuración.....	84
8.1 Gestión de Funciones.....	84
8.2 Gestión de Módulos.....	85
8.3 Gestión de Roles.....	86
8.4 Gestión de Usuarios.....	87

## Íconos usados en el Manual

A lo largo de este documento, los iconos a continuación se utilizan para subrayar puntos o elementos importantes.



Ícono de módulo inicial “Escritorio”



Ícono de módulo de instrumentos de cooperación



Ícono de registro de instrumentos de cooperación



Ícono de registro de clases de instrumentos de cooperación



Ícono de registro de instrumentos



Ícono de registro de países de los instrumentos de cooperación



Ícono de registro de los tipos de instrumentos de cooperación



Ícono de registro de los administradores internos y contraparte



Ícono de registro de las dependencias



Ícono de registro de los tipos de instrumentos de cooperación



Ícono de registro de las empresas



Ícono de módulo de reportes



Ícono de módulo de configuración de usuarios



Ícono de edición de información de una tabla



Ícono de asignación de administradores (internos y contraparte) y empresas



Ícono de eliminación de información de una tabla



Ícono de carga de documento (pdf) del instrumento de cooperación



Ícono de generación de seguimientos de los instrumentos de cooperación



Ícono de creación de usuarios del sistema



Ícono de creación de módulos del sistema



Ícono de creación de roles del sistema



Ícono de creación de funcionalidades del sistema

## **Presentación de la Solución**

### **2.1 Breve Descripción**

UNACH Virtual es una plataforma de servicios en línea, desarrollada el objetivo de enfrentar las necesidades formativas que requiere la institución; y, dentro de ello, los procesos de evaluación y de mejora permanentes, la institución pone a su disposición esta plataforma que integra los siguientes sistemas.

- Control Académico
- Aulas Virtuales
- Investigación
- Evaluación
- Vinculación
- Bienestar Estudiantil
- Talento Humano
- Biblioteca
- Tutorías

### **2.2 Íconos Descriptivos**

Los iconos usados dentro del Sistema de Gestión de Instrumentos de Cooperación representan el acceso a las funcionalidades y resultados de las acciones ejecutadas.

Estos iconos han sido diseñados por la Unidad de Administración y Desarrollo de Software, en función de la actividad que el usuario va a realizar.

### **2.3 Usuarios del Sistema**

Los usuarios del sistema, también conocidos como usuarios clientes, son individuos que acceden al sistema para utilizar los servicios que ofrece. Dentro de estos usuarios, se pueden distinguir varios perfiles o niveles, y según el nivel que tengan, tendrán más o menos privilegios durante su interacción con el Sistema de Gestión de Instrumentos de Cooperación.

## Acceso a la Aplicación

### 3.1 Dirección

Los usuarios registrados dentro del Sistema de Gestión de Instrumentos de Cooperación deben conectarse a la aplicación mediante los siguientes pasos:

1.- Ingresar a la dirección

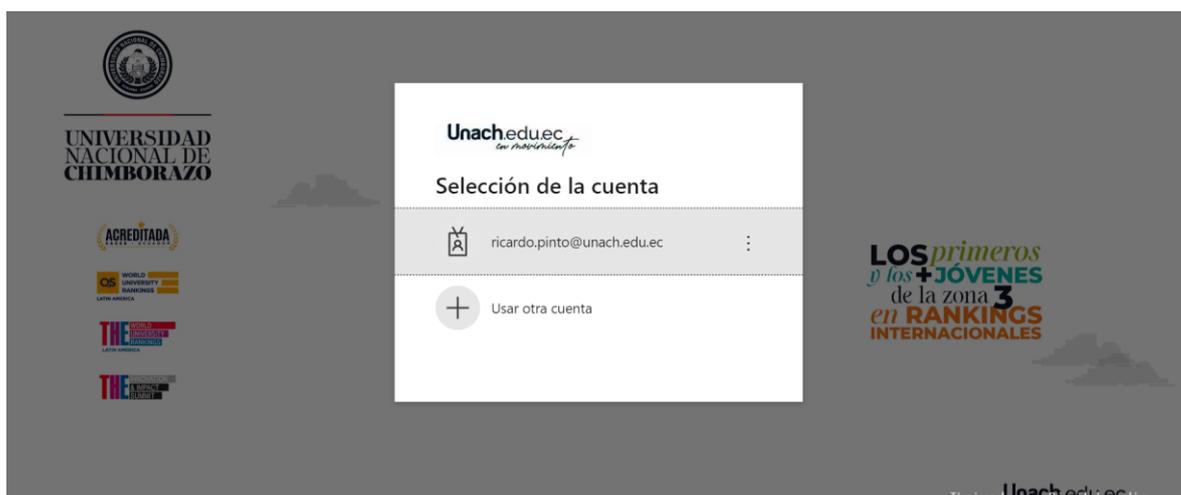
<https://uvirtual.unach.edu.ec/>

**Opción:** Inst. de Cooperación

### 3.2 Autenticación

El Sistema de Gestión de Instrumentos de Cooperación, utiliza el mecanismo de autenticación del correo electrónico institucional, esto le permitirá acceder a distintos sistemas únicamente usando su cuenta de correo institucional. Para ingresar al sistema siga los siguientes pasos:

1.- Ingresar su usuario y contraseña del correo electrónico institucional.



Los usuarios que no se encuentren registrados dentro de la base de datos de servidores institucionales o aquella cuya cuenta de correo electrónico institucional no se encuentre registrada no podrán acceder al sistema por lo que visualizarán la siguiente pantalla.



# ¡Atención!

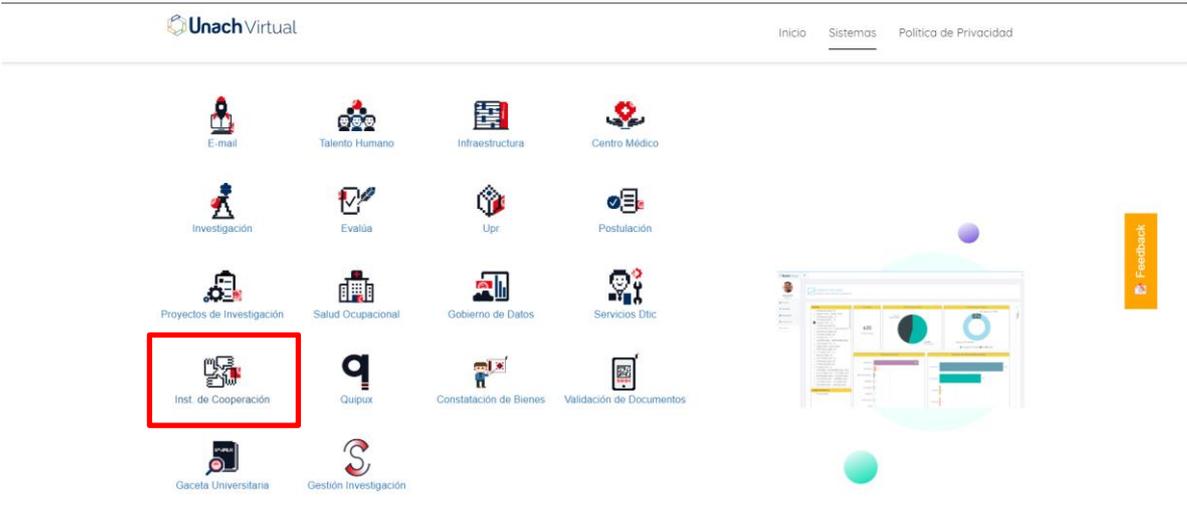
**No tiene acceso a este módulo.**

Lo sentimos, el sistema detectó que no es un usuario registrado. Verifique que su información se encuentre registrada correctamente.

Si el problema persiste, por favor comuníquese con el Departamento de Administración y Desarrollo de Software.

[Regresar](#)

También se podrá acceder desde **uvirtual.unach.edu.ec** Después Click en el icono Inst. de Cooperación

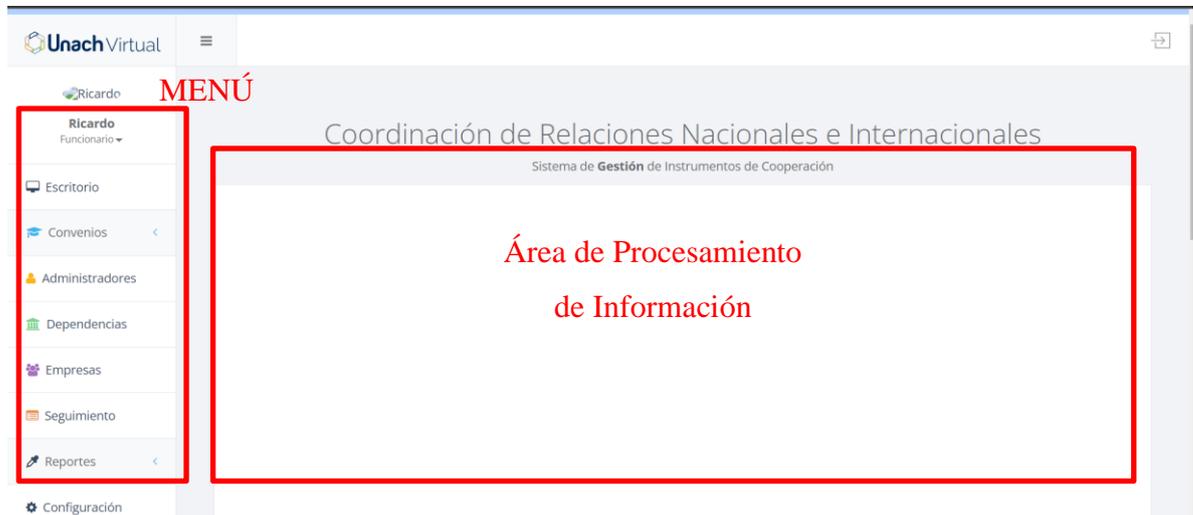


## Estructura de la Aplicación

Las siguientes funcionalidades se encuentran dentro del sistema mediante la siguiente agrupación.

## Interfaz

La interfaz de usuario que se emplea en los sistemas de la plataforma UVirtual emplean un esquema basado en tres componentes que brindar ayudan a la operatividad de este.



- **Menú.** - Acceso a distintas funcionalidades del sistema.
- **Área de procesamiento de información.** - Integra diferentes funcionalidades propias del sistema para realizar las tareas y procesos que se requiere por parte del usuario.

## Menús

El sistema presenta las siguientes opciones desde la sección Menú.

- **Escritorio.** - Presenta la pantalla inicial del sistema y los diferentes accesos al menú.
- **Convenios.** – Presenta el listado de los instrumentos de cooperación.
- **Administradores.** – Presenta el listado de los administradores internos y contraparte de los instrumentos de cooperación.
- **Dependencias.** – Presenta el listado de las dependencias de los instrumentos de cooperación.
- **Empresas.** – Presenta el listado de las empresas de los instrumentos de cooperación.
- **Seguimiento.** - Listado de instrumentos de cooperación que se encuentran en ejecución.
- **Reportes.** - Pantalla de generación de reportes generales de instrumentos de cooperación y sus seguimientos.
- **Configuración.** – Presenta el listado de los usuarios, roles, módulos y funcionalidades del sistema, así como su administración.

## Funcionalidades

*FUNCIONALIDADES PRESENTES/DESCRITAS*

Las siguientes funcionalidades se encuentran dentro del sistema mediante la siguiente agrupación.

#### Convenios

- Gestión de Empresas
- Gestión de Dependencias
- Gestión de Administradores
- Gestión de Países
- Gestión de Tipos
- Gestión de Clase
- Gestión de Instrumentos
- Gestión de Instrumentos de Cooperación
- Asignación de Empresas, Dependencias y Administradores
- Generación de Seguimientos

#### Seguimientos

- Creación de Actividades, Evidencias y Obligaciones

#### Reportes

- Reporte de Seguimiento Individual
- Reporte de Cumplimiento
- Reporte General Anual
- Reporte de Instrumentos de Cooperación por Tipos
- Reporte de Instrumentos de Cooperación por Clases

#### Configuración de Usuarios

- Administración de Funciones
- Administración de Módulos
- Administración de Roles
- Administración de Usuarios

#### *FUNCIONALIDADES NO PRESENTES/ NO DESCRITAS*

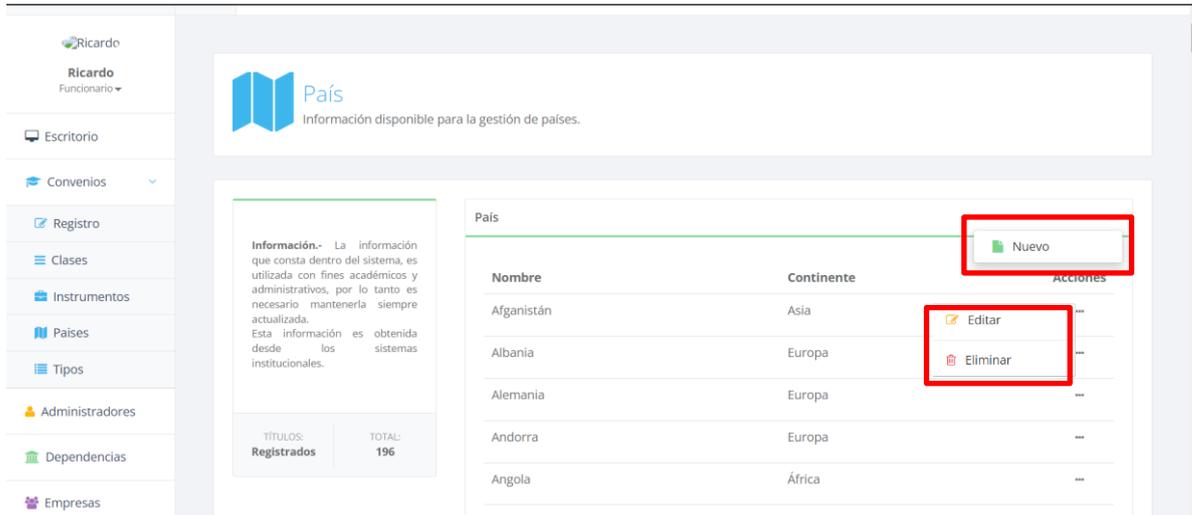
Las funcionalidades a continuación presentadas, forman parte del sistema sin embargo el usuario no puede interactuar directamente con ellas:

- Autenticación mediante el correo electrónico institucional (Office 365).
- Notificaciones de correo electrónico automáticos.
- Generación de documentos con extensión .PDF, DOCX, CSV, entre otros.

## Función 01 / Creación de un Instrumento de Cooperación

### 5.1 Gestión de Países

La sección de País contiene un listado de países también un botón para crear, editar y eliminar.

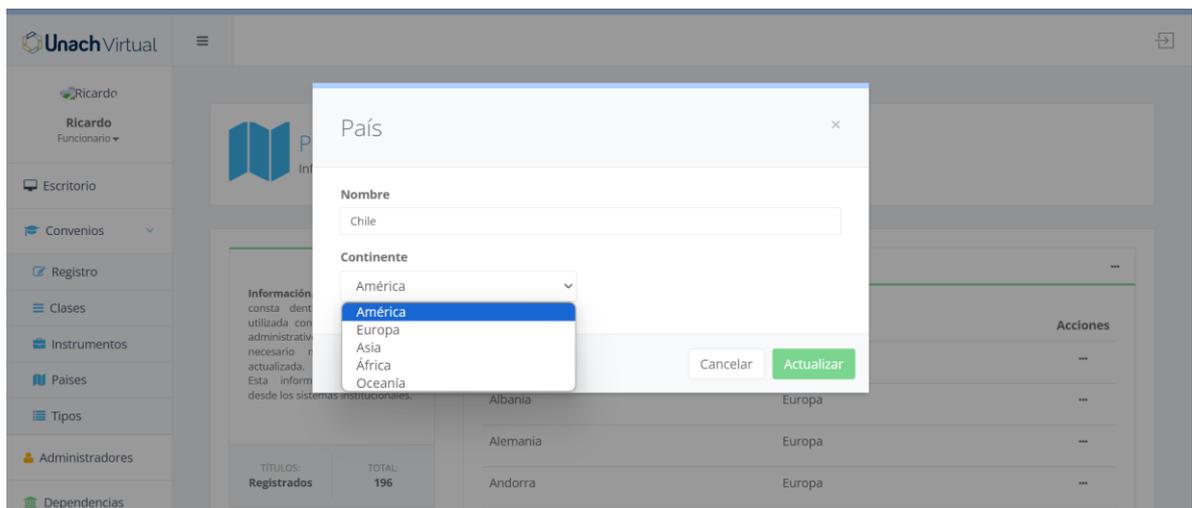


**Nuevo.** - Se puede realizar la creación de un nuevo país.

**Editar.** - Se puede realizar la modificación de un país.

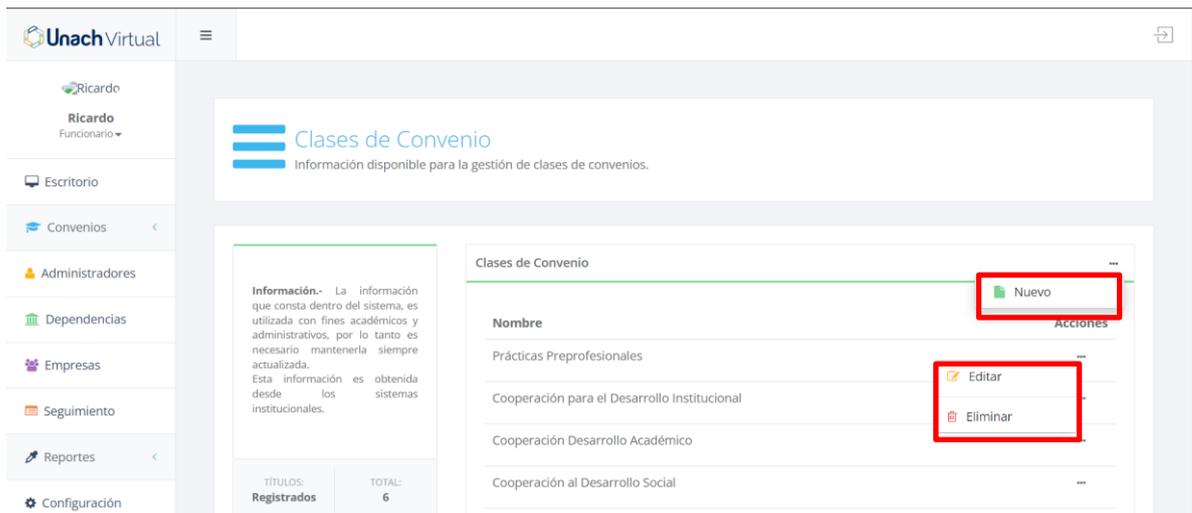
**Eliminar.** - Se puede realizar eliminación de un país.

**Paso 1.** – Se ingresa el nombre del país y se selecciona el continente



### 5.2 Gestión de Clases de Instrumentos de Cooperación

La sección de Clases contiene un listado de las clases de instrumentos de cooperación también un botón para crear, editar y eliminar.

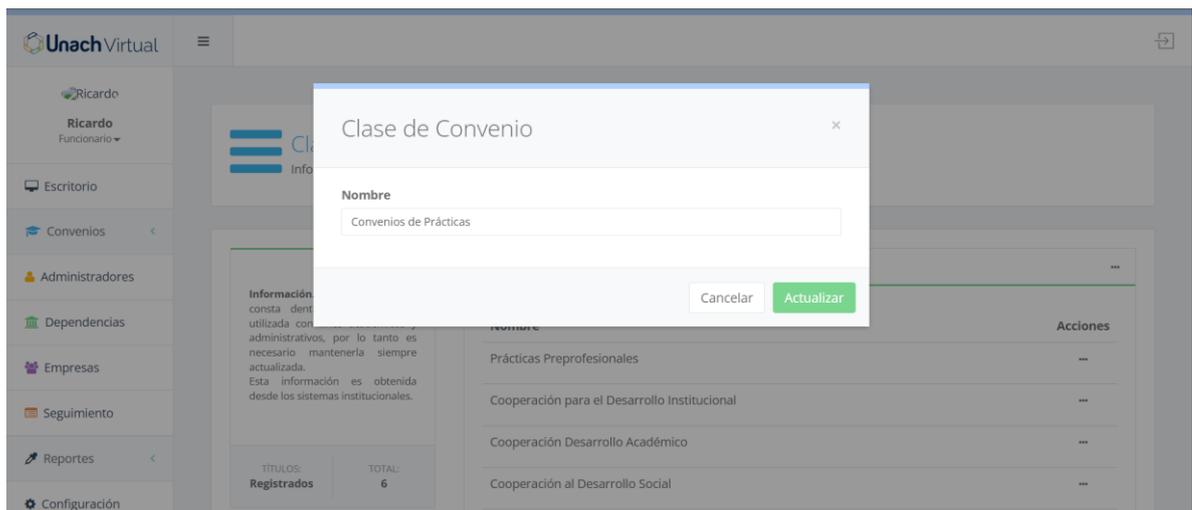


**Nuevo.** - Se puede realizar la creación de una nueva clase.

**Editar.** - Se puede realizar la modificación de una clase.

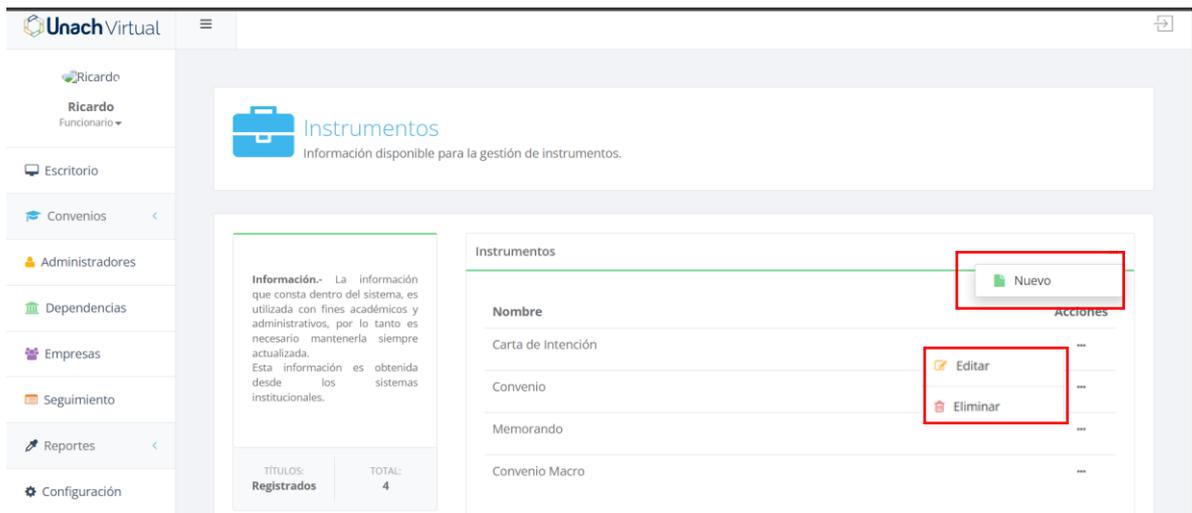
**Elimina.** - Se puede realizar eliminación de una clase.

**Paso 1.** – Se ingresa el nombre de la clase de instrumento de cooperación



### 5.3 Gestión de Instrumentos

La sección de Instrumentos contiene un listado de instrumentos también un botón para crear, editar y eliminar.

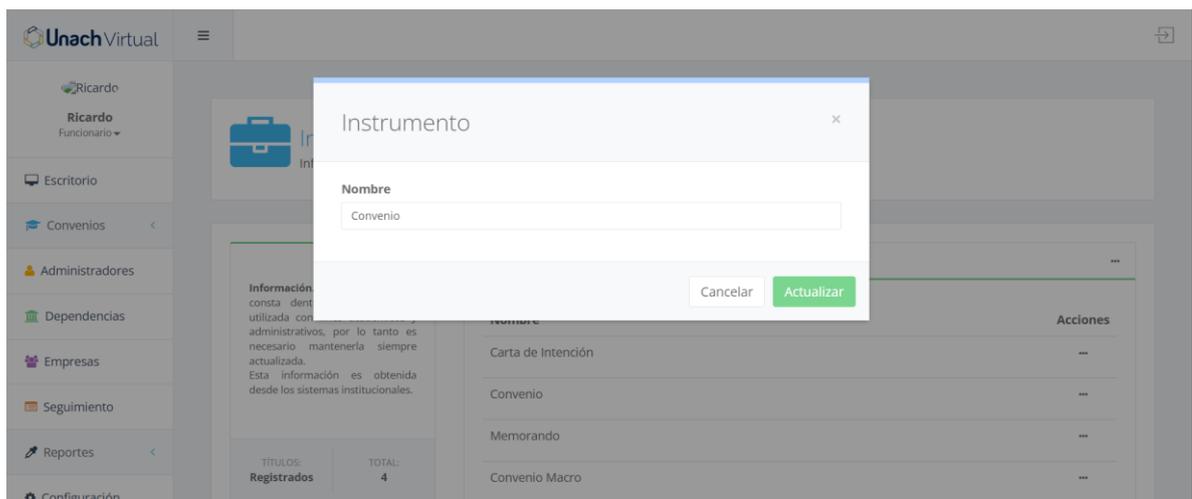


**Nuevo.** - Se puede realizar la creación de un nuevo instrumento.

**Editar.** - Se puede realizar la modificación de un instrumento.

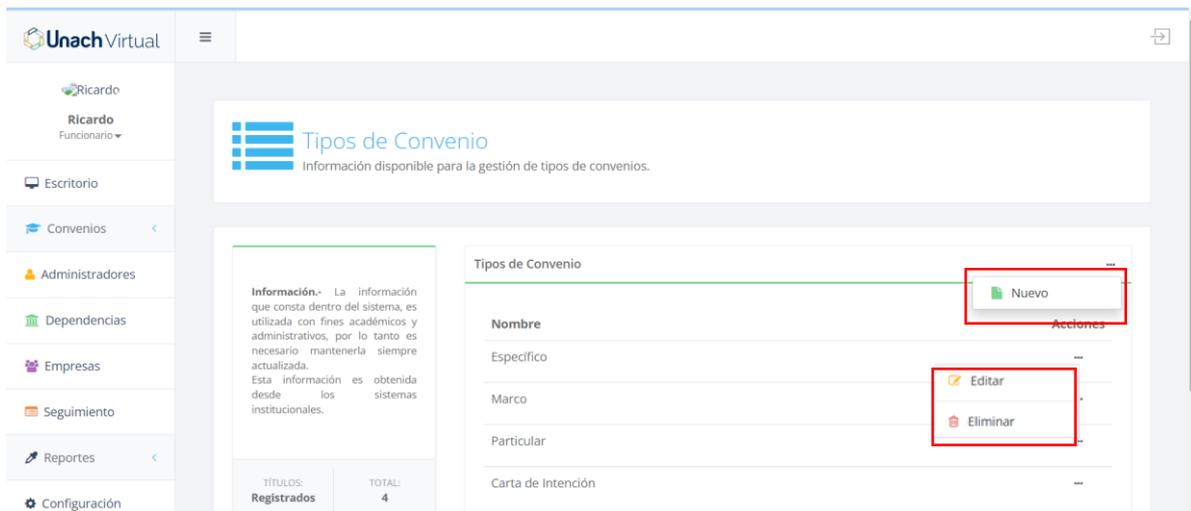
**Elimina.** - Se puede realizar eliminación de un instrumento.

**Paso 1.** – Se ingresa el nombre del instrumento



## 5.4 Gestión de Tipos de Instrumentos de Cooperación

La sección de Tipos contiene un listado de tipos de instrumentos de cooperación también un botón para crear, editar y eliminar.

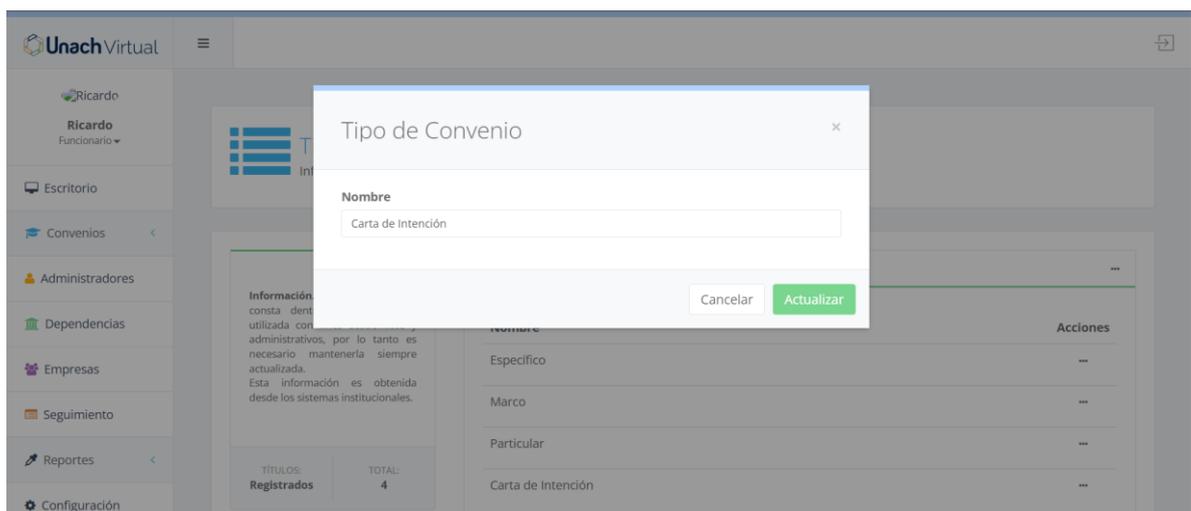


**Nuevo.** - Se puede realizar la creación de un nuevo tipo de instrumento de cooperación.

**Editar.** - Se puede realizar la modificación de un tipo de instrumento de cooperación.

**Eliminar.** - Se puede realizar eliminación de un tipo de instrumento de cooperación.

**Paso 1.** – Se ingresa el nombre del tipo de instrumento de cooperación



## 5.5 Gestión de Dependencias

La sección de Dependencias contiene un listado de países también un botón para crear, editar y eliminar.

**Dependencias**  
Información disponible para la gestión de dependencias.

**Información.-** La información que consta dentro del sistema, es utilizada con fines académicos y administrativos, por lo tanto es necesario mantenerla siempre actualizada. Esta información es obtenida desde los sistemas institucionales.

Nombre	Identificación	Acciones
INGENIERIA EN TECNOLOGIAS DE LA INFORMACION (R)	0602494205	<a href="#">Nuevo</a> <a href="#">Editar</a> <a href="#">Eliminar</a>

Registros del 1 al 1 de un total de 1 registros

TÍTULOS: Registrados TOTAL: 1

**Nuevo.** - Se puede realizar la creación de una nueva dependencia.

**Editar.** - Se puede realizar la modificación de una dependencia.

**Eliminar.** - Se puede realizar eliminación de una dependencia.

## 5.6 Gestión de Empresas

La sección de Empresas contiene un listado de las empresas también un botón para crear, editar y eliminar.

**Empresas**  
Información disponible para la gestión de empresas.

**Información.-** La información que consta dentro del sistema, es utilizada con fines académicos y administrativos, por lo tanto es necesario mantenerla siempre actualizada. Esta información es obtenida desde los sistemas institucionales.

Nombre	Identificación	Acciones
La Asociación De Jóvenes Empresarios Aje Ecuador	1-2023	<a href="#">Nuevo</a> <a href="#">Editar</a> <a href="#">Eliminar</a>
Microsoft	0604844100	

Registros del 1 al 2 de un total de 2 registros

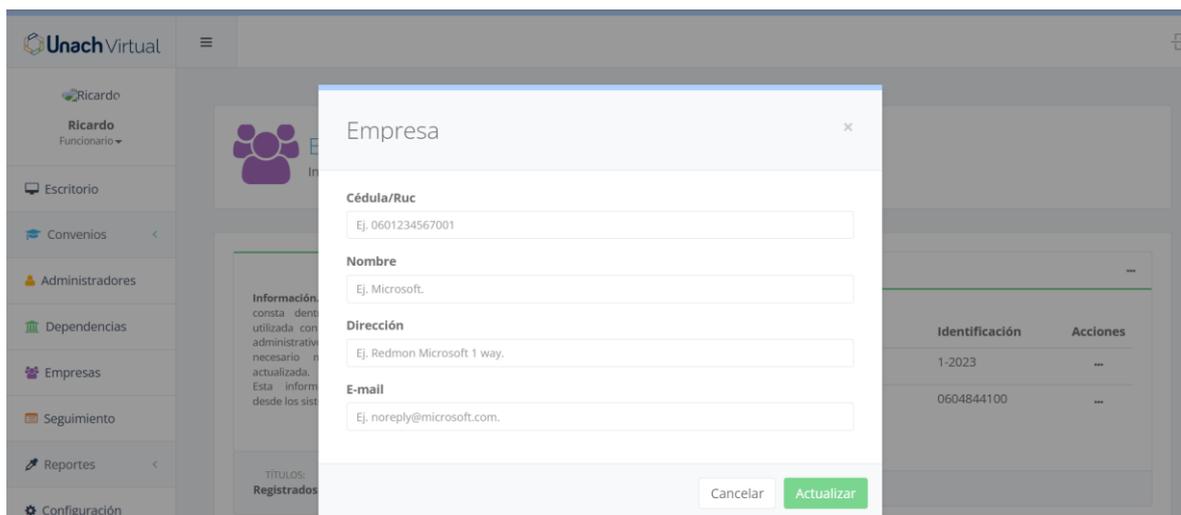
TÍTULOS: Registrados TOTAL: 2

**Nuevo.** - Se puede realizar la creación de una nueva empresa.

**Editar.** - Se puede realizar la modificación de una empresa.

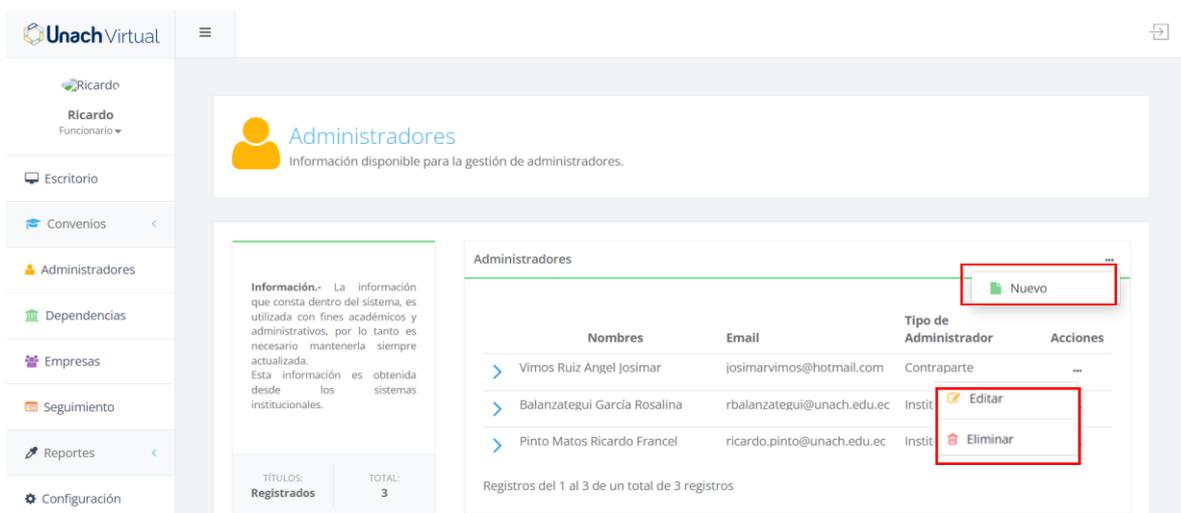
**Eliminar.** - Se puede realizar eliminación de una empresa.

**Paso 1.** – Se ingresa la Cédula/Ruc, nombre, dirección y e-mail asociado a la empresa.



## 5.7 Gestión de Administradores

La sección de Administradores contiene un listado de administradores un botón para crear, editar y eliminar.

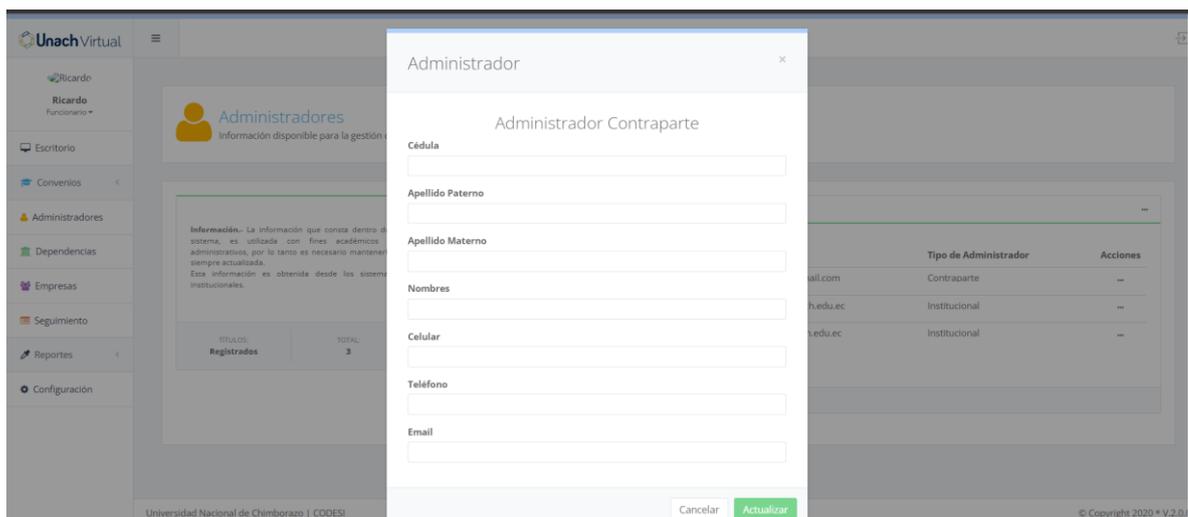


**Nuevo.** - Se puede realizar la creación de un nuevo administrador.

**Editar.** - Se puede realizar la modificación de un administrador.

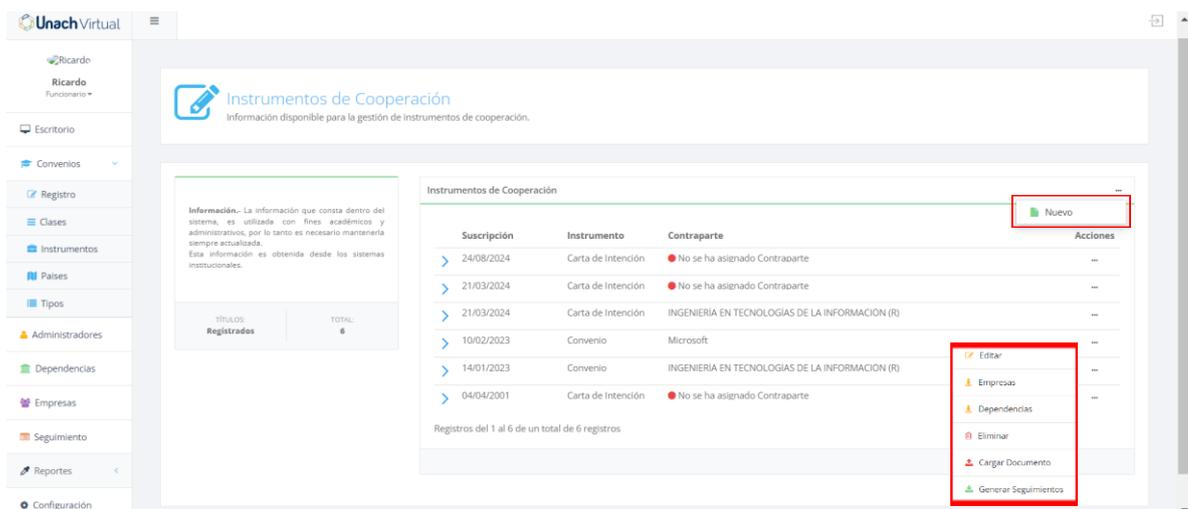
**Eliminar.** - Se puede realizar eliminación de un administrador.

**Paso 1.** – Se ingresa la cédula, apellido paterno, apellido materno, nombres, celular, teléfono y e-mail asociado a los administradores de la contraparte.



## 5.8 Gestión de Instrumentos de Cooperación

La sección de Convenios contiene un listado de los instrumentos de cooperación un botón para crear, editar y eliminar.



**Nuevo.** - Se puede realizar la creación de un nuevo instrumento de cooperación.

**Empresa.** - Se puede realizar la asignación de una empresa con su administrador al instrumento de cooperación.

**Dependencia.** - Se puede realizar la asignación de una dependencia con su administrador al instrumento de cooperación.

**Eliminar.** - Se puede realizar eliminación de un instrumento de cooperación.

**Cargar Documento.** - Se puede subir el documento PDF del instrumento de cooperación.

**Generar Seguimiento.** - Se puede genera los seguimientos del instrumento de cooperación según los años de vigencia (1 seguimiento anual).

**Paso 1.** – Se ingresa la clase, tipo, instrumento, país (el ámbito cambiará de manera automática según el país que se seleccione), estado, título, objeto, fecha de suscripción y fecha fin.

The screenshot shows the 'Instrumento de Cooperación' form in the UnachVirtual system. The form is titled 'Instrumento de Cooperación' and contains the following fields:

- Clase de Convenio:** Cooperación al Desarrollo Insti
- Tipo de Convenio:** Carta de Intención
- Instrumento:** Carta de Intención
- País:** Afganistán
- Ámbito:** Internacional
- Estado:**
- Título:** Ej. Convenio de prácticas profesionales.
- Objeto:** Ej. Instrumentar la cooperación interinstitucional para el desarrollo de actividades de prácticas preprofesionales.
- Fecha de Suscripción:** 31/03/2024
- Fecha de Expiración:** 31/03/2024

At the bottom of the form, there are 'Cancelar' and 'Actualizar' buttons. The background shows a sidebar with navigation options and a main content area with a table of registered instruments.

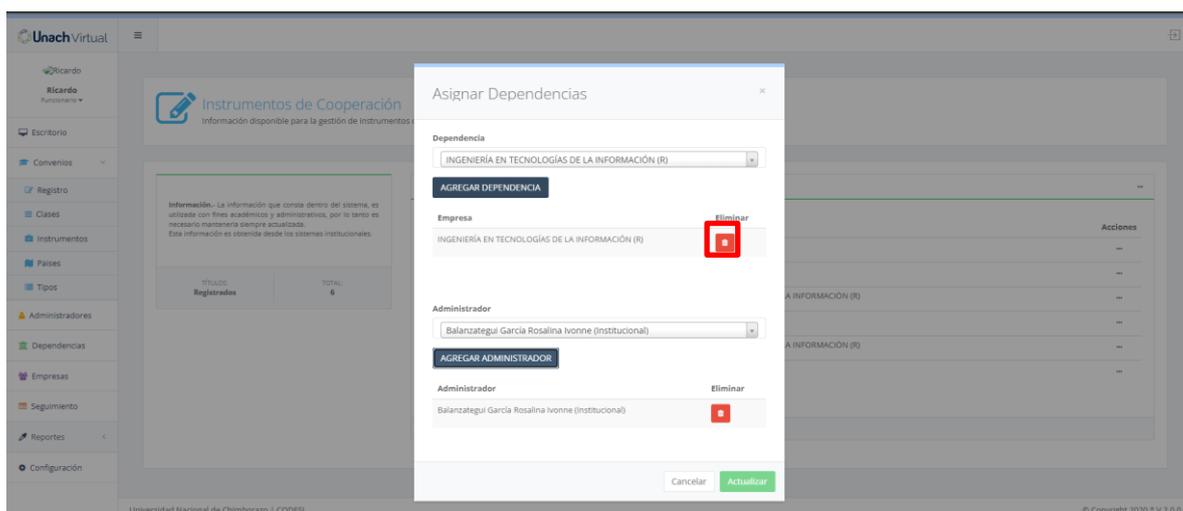
**Paso 2.** – Se asigna la empresa y el administrador que estarán asociados al instrumento de cooperación. En caso de agregar por error dando habrá un botón para eliminar

The screenshot shows the 'Asignar Empresas' form in the UnachVirtual system. The form is titled 'Asignar Empresas' and contains the following fields:

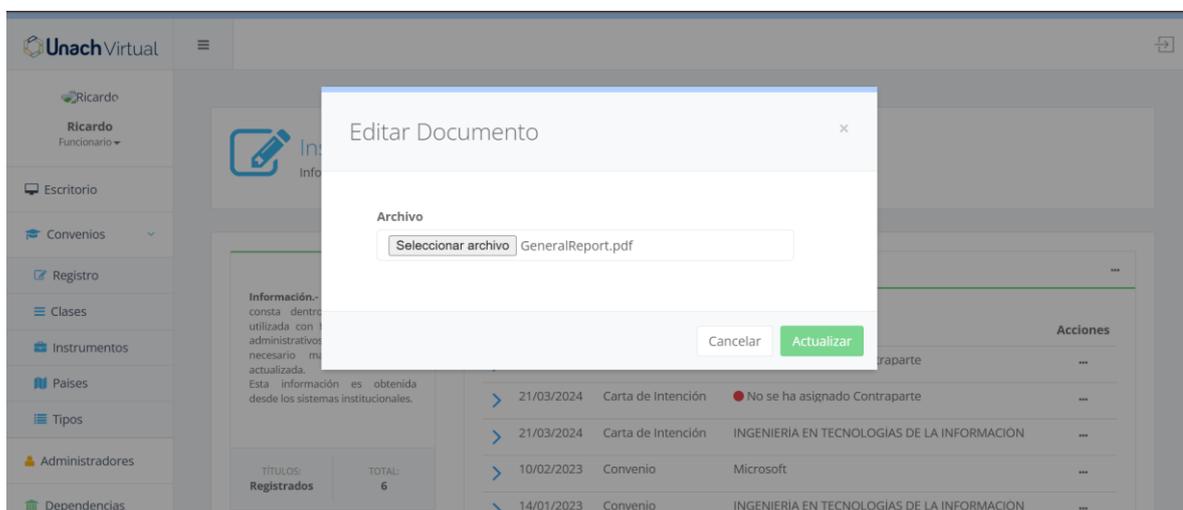
- Empresa:** La Asociación De Jóvenes Empresarios Aje Ecuador
- Administrador:** Vimos Ruiz Angel Josimar(Contraparte)

Below each field, there are 'AGREGAR EMPRESA' and 'AGREGAR ADMINISTRADOR' buttons. There are also 'Eliminar' buttons with a red trash icon next to each entry. At the bottom of the form, there are 'Cancelar' and 'Actualizar' buttons. The background shows the same sidebar and main content area as in the previous screenshot.

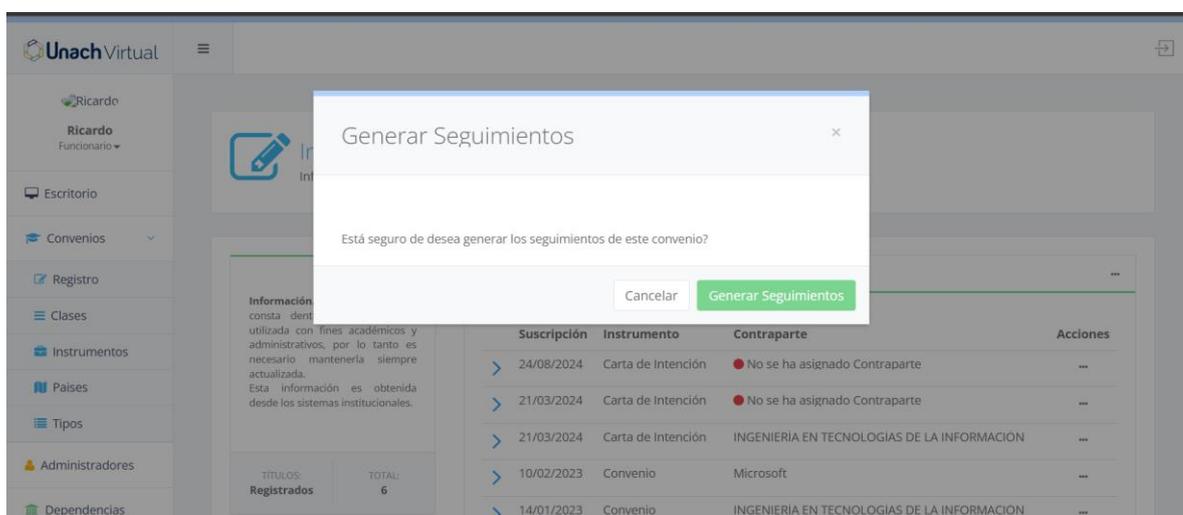
**Paso 3.** – Se asigna la dependencia y el administrador que estarán asociados al instrumento de cooperación. En caso de agregar por error dando habrá un botón para eliminar



**Paso 4.** – Se sube el documento PDF del instrumento de cooperación



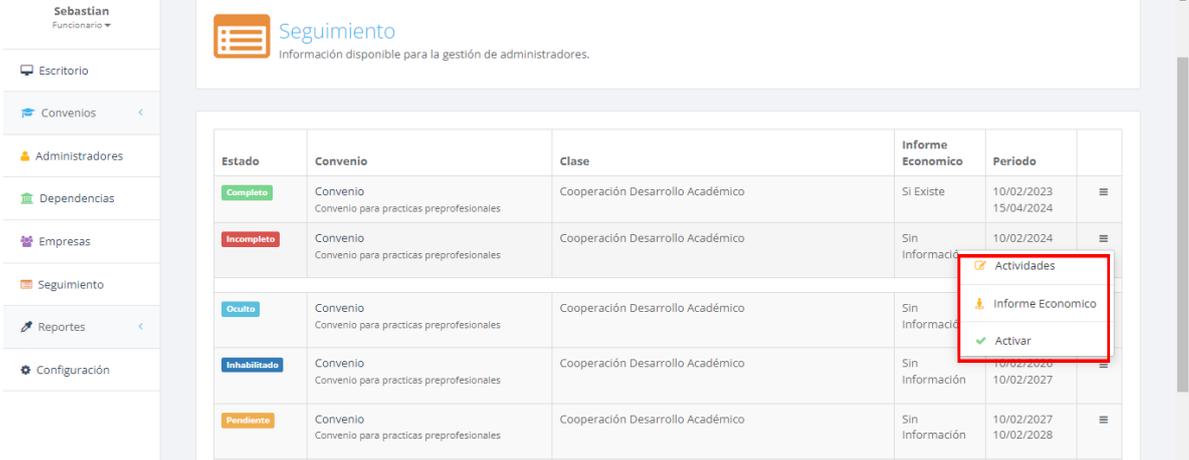
**Paso 5.** – Se genera los seguimientos según los años de vigencia del convenio los cuales se mostrarán en el módulo de seguimientos



## Función 02 / Gestión de Seguimientos

### 6.1 Creación de Actividades, Evidencias y Obligaciones

El módulo de Seguimientos cuenta con un listado de seguimientos y botones para generar Actividades e Informe Económico



The screenshot displays the 'Seguimiento' (Follow-up) module interface. On the left is a navigation sidebar with options: Escritorio, Convenios, Administradores, Dependencias, Empresas, Seguimiento, Reportes, and Configuración. The main area shows a table with columns: Estado, Convenio, Clase, Informe Economico, and Periodo. A context menu is open over the 'Incompleto' row, showing options: Actividades, Informe Economico, and Activar.

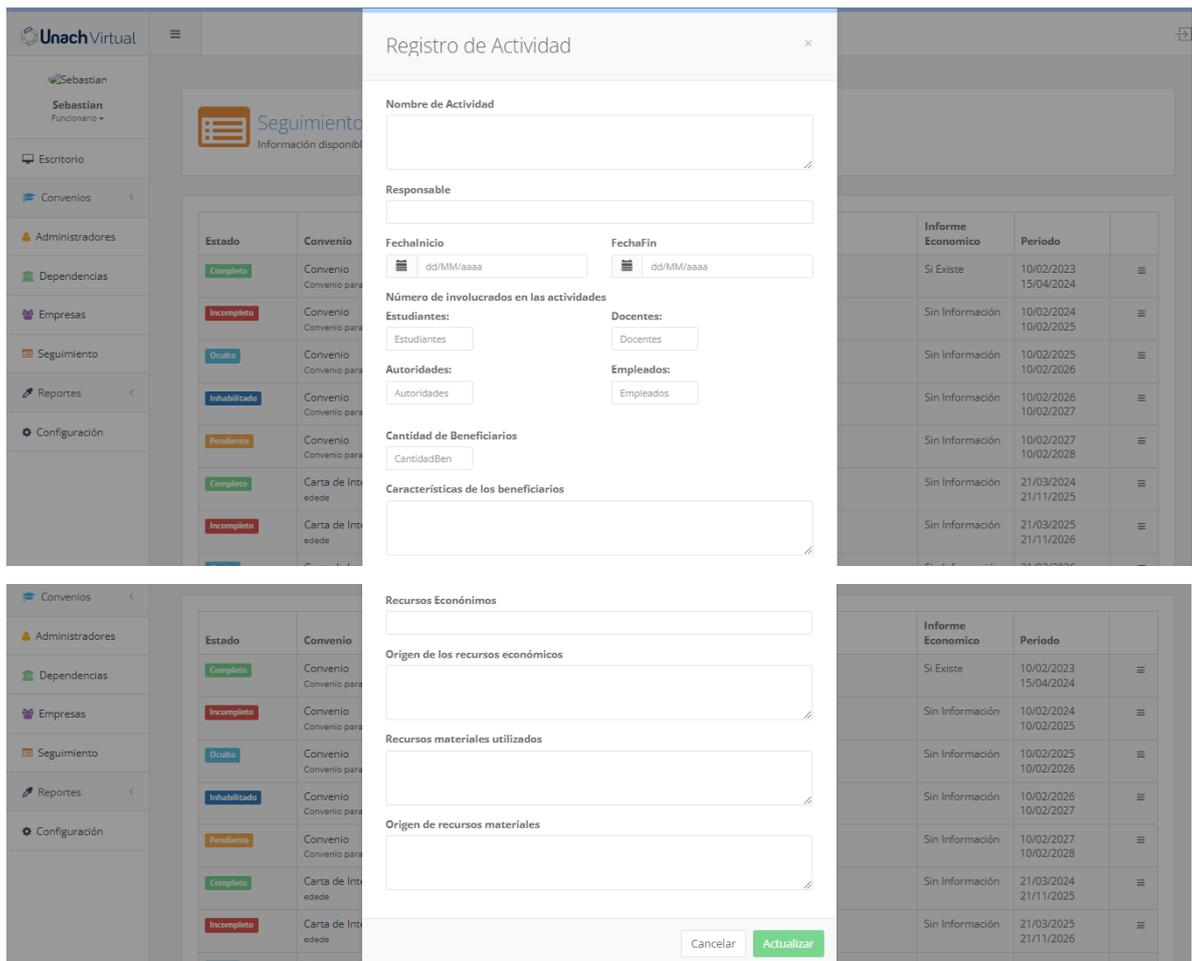
Estado	Convenio	Clase	Informe Economico	Periodo
Completo	Convenio Convenio para practicas preprofesionales	Cooperación Desarrollo Académico	Si Existe	10/02/2023 15/04/2024
Incompleto	Convenio Convenio para practicas preprofesionales	Cooperación Desarrollo Académico	Sin Información	10/02/2024
Oculto	Convenio Convenio para practicas preprofesionales	Cooperación Desarrollo Académico	Sin Información	
Inhabilitado	Convenio Convenio para practicas preprofesionales	Cooperación Desarrollo Académico	Sin Información	10/02/2020 10/02/2027
Pendiente	Convenio Convenio para practicas preprofesionales	Cooperación Desarrollo Académico	Sin Información	10/02/2027 10/02/2028

**Actividad.** - Se puede realizar la creación de una nueva actividad para el seguimiento

**Empresa.** - Se puede confirmar la existencia de un Informe Económico para ese seguimiento

**Activar.** – Esta función la realiza únicamente el Administrador del sistema. Con este botón puede activar seguimientos con estado Incompleto tras una petición formal.

**Paso 1.** – Se ingresa el nombre de la Actividad, el responsable, las fechas de Inicio y Fin, el número de involucrados, la cantidad de beneficiarios, los recursos económicos y su ubicación (en caso de que aplique), los recursos materiales y su origen (en caso que aplique). Para finalizar hacemos clic en Actualizar.



**Paso 2.** – Una vez ingresada información en la actividad podemos ingresar las evidencias, obligaciones o eliminarla.



**Paso 3.** – Para asignar Evidencias se escoge el tipo de evidencia, se indica el lugar donde reposa la evidencia y los detalles de la misma, para añadir a la lista hacer clic en Agregar Evidencia, y si se necesitan añadir más evidencias se repetirá el proceso de este paso. Para finalizar se hace clic en Actualizar.

Asignar Evidencias

Evidencia  
Informe

Lugar donde reposa la evidencia

Detalle

**AGREGAR EVIDENCIA**

Evidencias Seleccionadas:

Cancelar Actualizar

**Paso 4.** – Para asignar Obligaciones se escoge el tipo de obligación y los detalles de este, para añadir a la lista hacer clic en Agregar Obligación, y si se necesitan añadir más obligaciones se repetirá el proceso de este paso. Para finalizar se hace clic en Actualizar.

Asignar Obligaciones

Tipo de Obligacion  
Unach

Obligaciones

**AGREGAR OBLIGACIÓN**

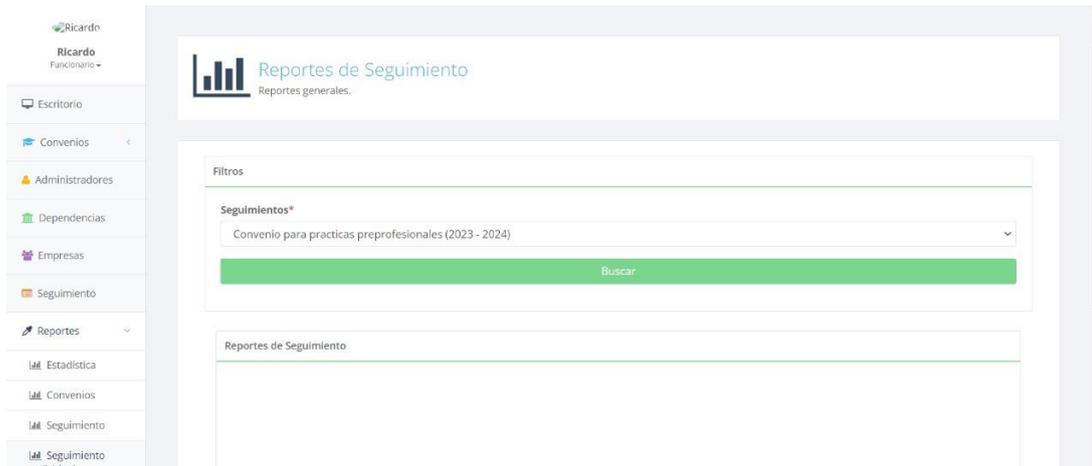
Obligaciones Seleccionadas:

Cancelar Actualizar

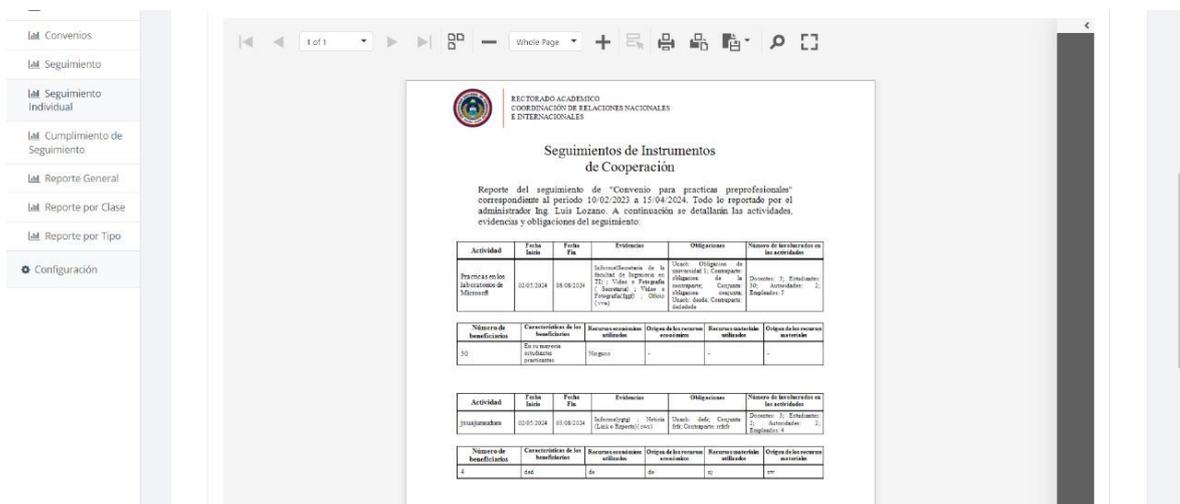
## **Función 03 / Reportes**

### **7.1 Reporte de Seguimiento Individual**

**Paso 1.** – Se selecciona el seguimiento del se desea obtener el reporte

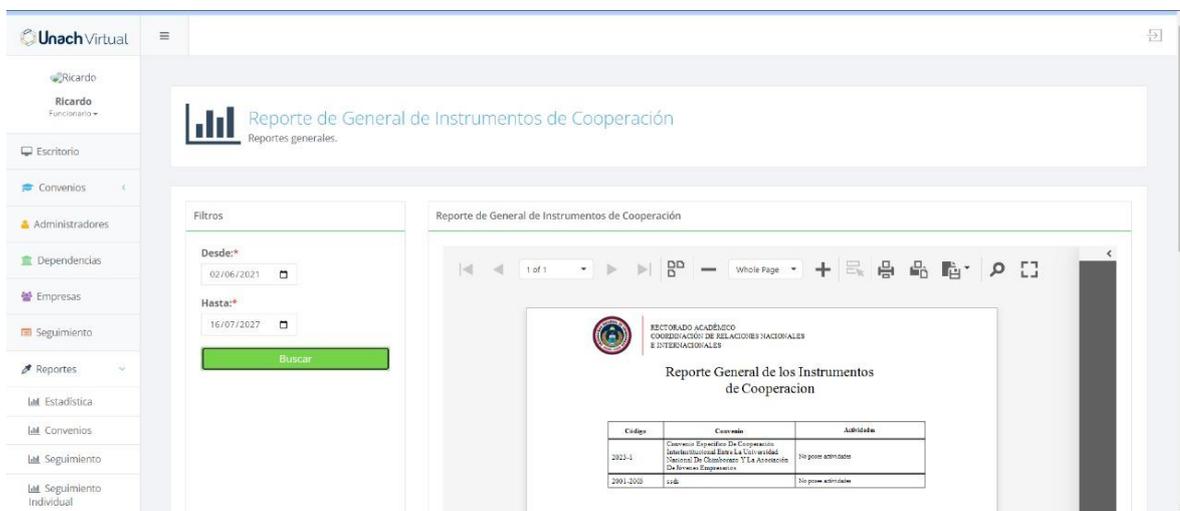


**Paso 2.** – Se genera el reporte y se imprime o se guarda como PDF



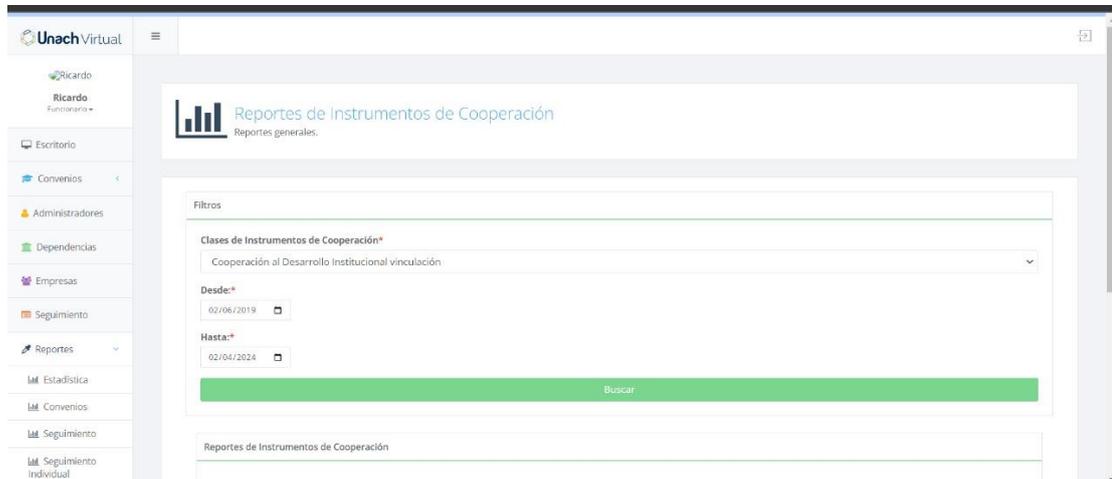
**7.2 Reporte General de Instrumentos de Cooperación**

**Paso 1.** – Se seleccionan las fechas por las que se desea filtrar el reporte y se genera.

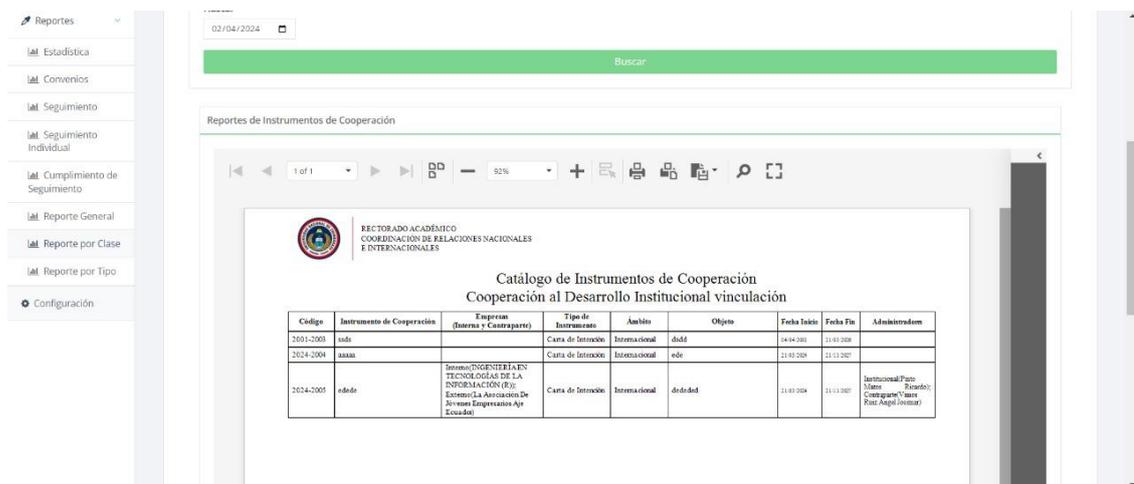


## 7.3 Reporte de Convenios por Clase

**Paso 1.** – Se selecciona la clase por la que se va a filtrar y las fechas en las que se encuentran activos los convenios buscados.

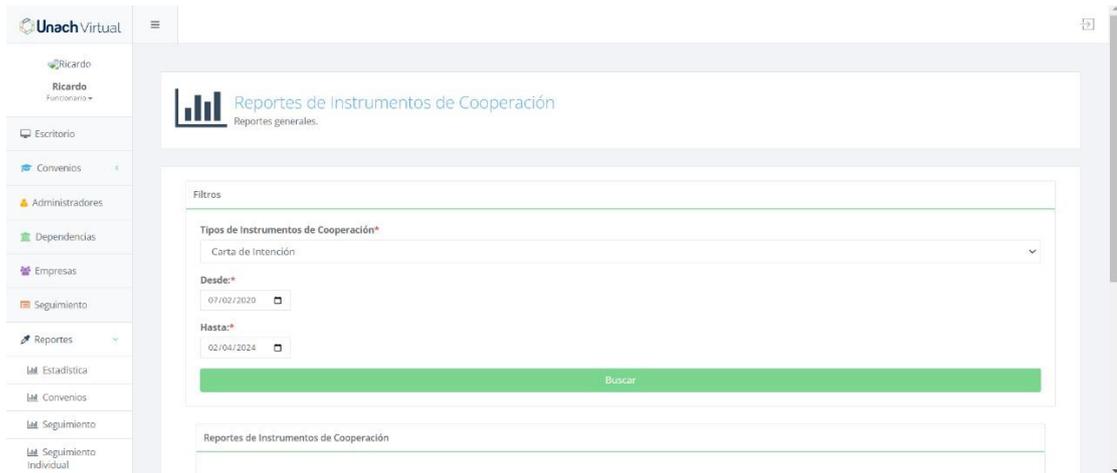


**Paso 2.** – Se genera el reporte y se imprime o se guarda como PDF

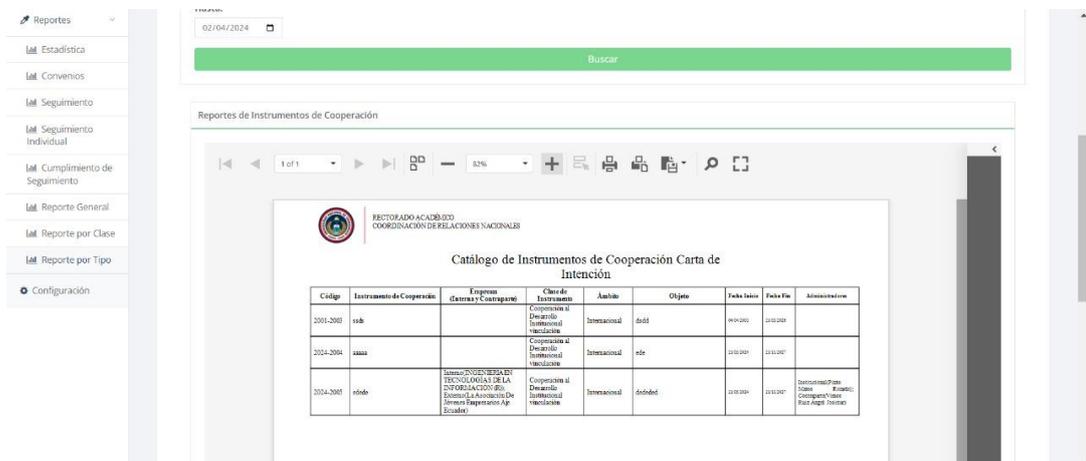


## 7.4 Reporte de Convenios por Tipo

**Paso 1.** – Se selecciona el tipo por el que se va a filtrar y las fechas en las que se encuentran activos los convenios buscados.



**Paso 2.** – Se genera el reporte y se imprime o se guarda como PDF

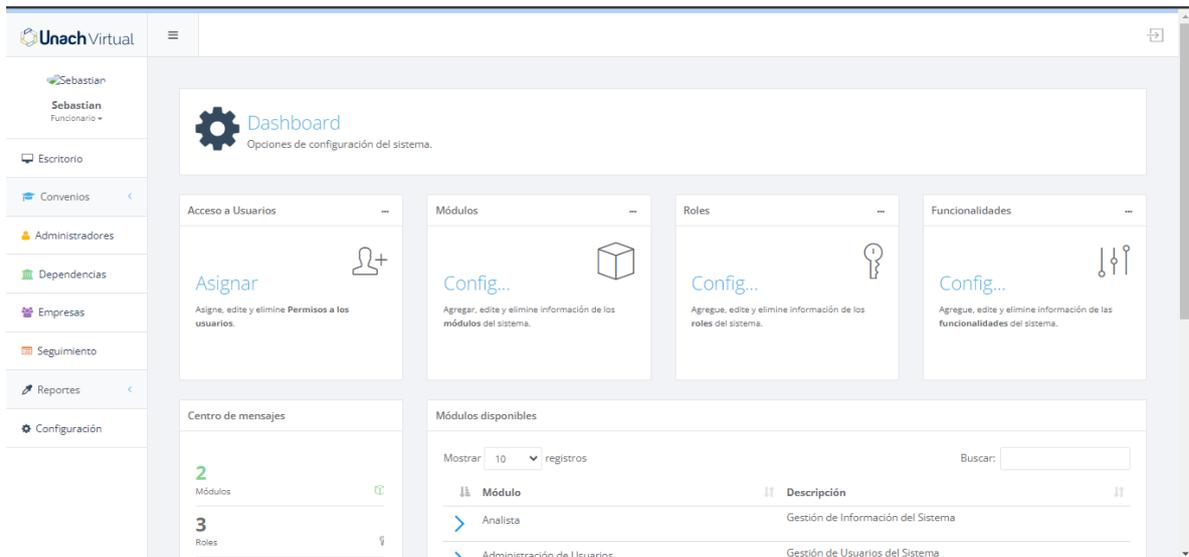


## 7.5 Reporte de Cumplimiento

**Paso 1.** – Se seleccionan las fechas por las que se filtrarán los seguimientos para saber en qué estado se encuentran



## Función 04 / Configuración



El módulo de Configuración cuenta con un dashboard para la configuración de Usuarios, Módulos, Roles y Funcionalidades.

**Usuarios.** – Aquellas personas que tengan acceso al sistema.

**Módulos.** – Módulos existentes en el sistema.

**Roles.** – Asignan funcionalidades a los usuarios del sistema, filtrando el acceso a ciertas partes de este.

**Funcionalidades.** – Describe las funcionalidades del sistema, todo lo que se puede realizar en el mismo.

### 8.1 Gestión de Funciones

En gestión de Funciones se listan las funciones que existen y se puede añadir una nueva función

Sebastian  
Funcionario

Funcionalidades  
Opciones de configuración del sistema.

Escritorio

Convenios

Administradores

Dependencias

Empresas

Seguimiento

Reportes

Configuración

Información.- La información que consta dentro del sistema, es utilizada con fines académicos y administrativos, por lo tanto es necesario mantenerla siempre actualizada.

Filtros  
Los filtros activados para la búsqueda son:

Funcionalidades

Show 10 entries

Search: Nuevo

Estado	Funcionalidad	Link	Orden	Módulo
●	Configuración	Configuracion/Index	8	Administración de Usuarios
●	Registro	Convenio/Index	1	Analista
●	Estadística	Reportes/Index	1	Analista
●	Escritorio	Home/Index	1	Analista
●	Seguimiento	Reportes/ReporteSeguimiento	1	Analista
●	Seguimiento Individual	Reportes/ReporteSeguimientoIndividual	1	Analista
●	Cumplimiento de Seguimiento	Reportes/ReporteCumplimientoSeguimiento	1	Analista
●	Reporte General	Reportes/ReporteGeneral	1	Analista
●	Reporte por Clase	Reportes/ReporteConvenioPorClase	1	Analista
●	Reporte por Tipo	Reportes/ReporteConvenioPorTipo	1	Analista

**Paso 1.** – Se selecciona el módulo al que se le va a asignar, las transacciones, su título, controlador, acción, ícono, orden, descripción y si se encuentra activo o inactivo.

Nueva Funcionalidad  
Opciones de configuración del sistema.

Información.- La información que consta dentro del sistema, es utilizada con fines académicos y administrativos, por lo tanto es necesario mantenerla siempre actualizada.

Filtros  
Los filtros activados para la búsqueda son:

Nueva Funcionalidad

IdModulo  
- Seleccionar una opción -

Transacciones

Titulo  
Ej. Buscar.

Controlador  
Ej. Servidores.

Accion  
Ej. Buscar.

IconClass  
Ej. fa fa-gear.

Orden  
0

Descripción  
Ej. -.

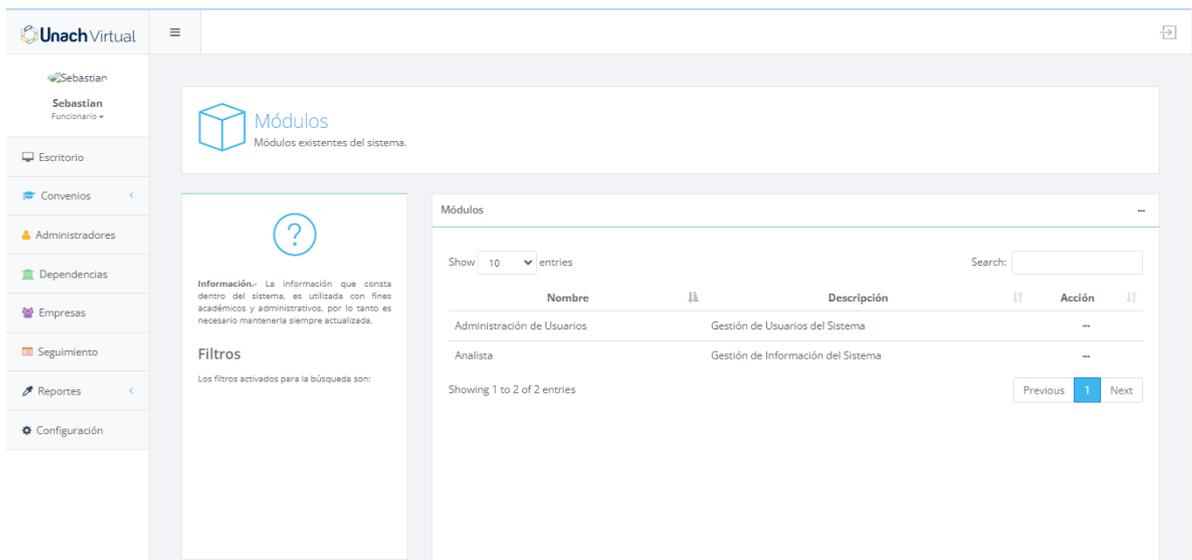
Activo

Visible

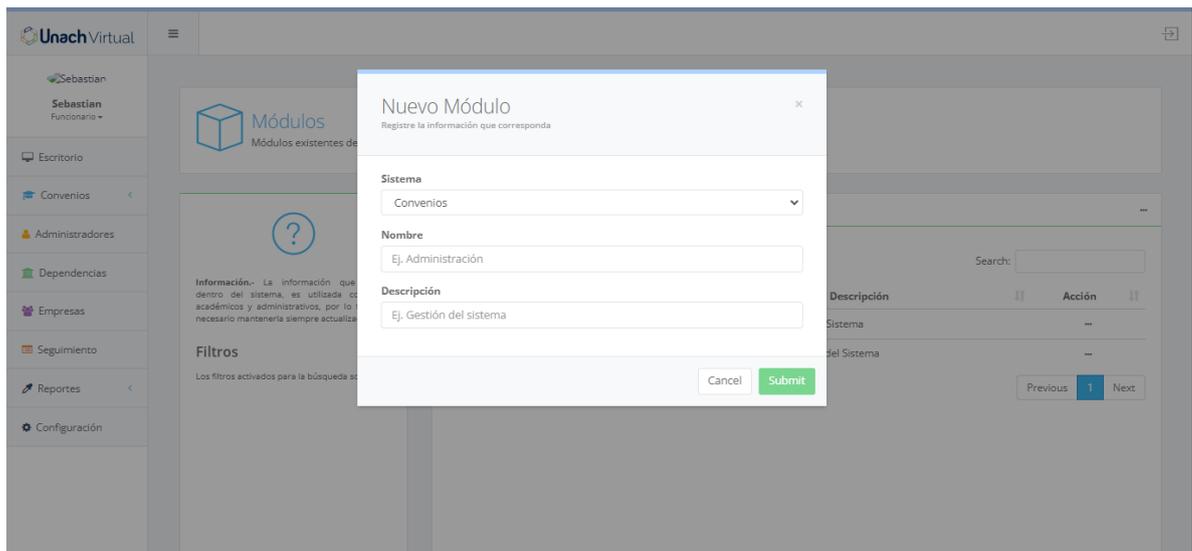
Guardar

## 8.2 Gestión de Módulos

En gestión de Módulos se listan los módulos que existen y se puede añadir un nuevo módulo.

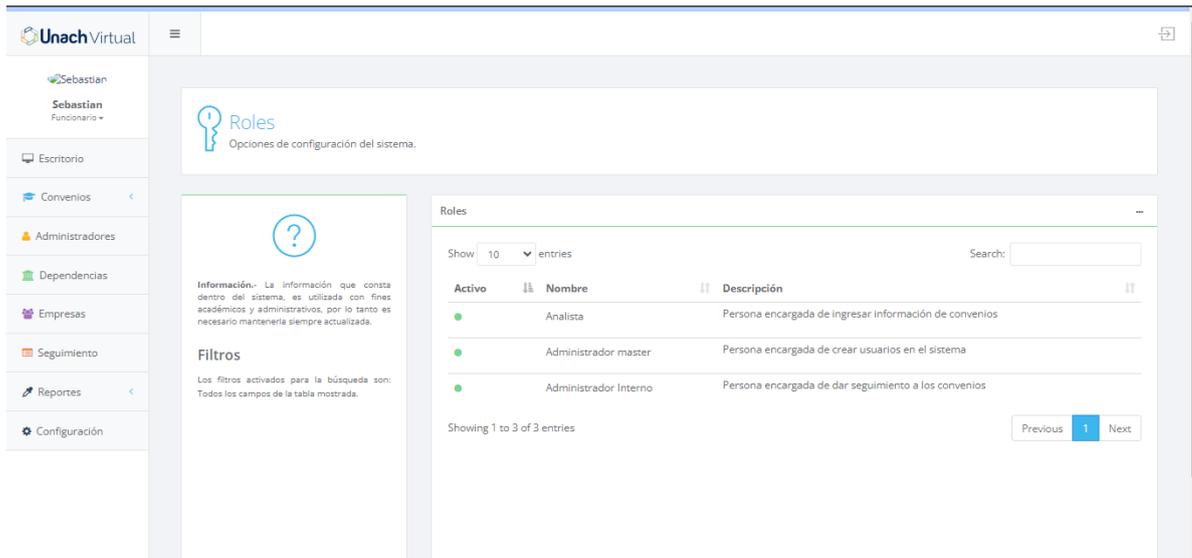


**Paso 1.** – Se selecciona el sistema para el cual se creará el módulo y añadir el nombre y la descripción.

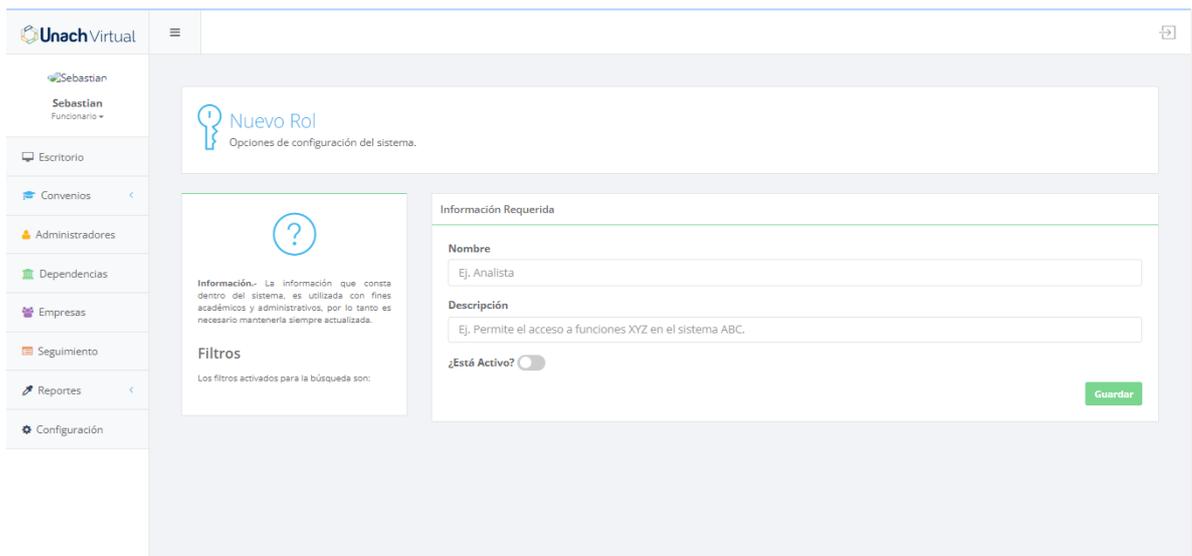


### 8.3 Gestión de Roles

En gestión de Roles se listan los roles que existen y se puede añadir un nuevo rol.



**Paso 1.** – Se añade un nombre y una descripción, además se cambia el estado entre activo e inactivo.



## 8.4 Gestión de Usuarios

En gestión de Usuarios se listan los usuarios que existen y se puede añadir nuevos usuarios.

The screenshot shows the 'Usuarios' (Users) management interface in UnachVirtual. The page title is 'Usuarios' with the subtitle 'Registro de usuarios del sistema'. On the left, there is a sidebar menu with options like 'Escritorio', 'Convenios', 'Administradores', 'Dependencias', 'Empresas', 'Seguimiento', 'Reportes', and 'Configuración'. The main content area is divided into two sections. The left section contains an 'Información' box with a question mark icon and a 'Filtros' section with three active filters: 'Documento de identidad', 'Nombres y Apellidos', and 'Expediente'. The right section is a table titled 'Usuarios' with columns for 'Nombres', 'Roles Asignados', and 'Acción'. The table lists four users: ALVAREZ CADENA DANIEL ROBERTO, CABEZAS RAMIREZ TATIANA ALEXANDRA, Pinto Matos Ricardo, and Reyes Solis Sebastian. Below the table, it indicates 'Showing 1 to 4 of 4 entries' and has 'Previous', '1', and 'Next' navigation buttons.

Nombres	Roles Asignados	Acción
ALVAREZ CADENA DANIEL ROBERTO 0603023524	1	--
CABEZAS RAMIREZ TATIANA ALEXANDRA 0603904426	1	--
Pinto Matos Ricardo 2300852627	1	--
Reyes Solis Sebastian 0604844100	1	--

**Paso 1.** – Para añadir un usuario se busca al funcionario de la universidad por medio de su correo. De la base de datos de talento humano se obtiene la información del usuario. Una vez creado se le agrega un rol.

The screenshot shows the 'Nuevo Usuario' (New User) creation page in UnachVirtual. The page title is 'Nuevo Usuario' with the subtitle 'Documento generado con información registrada en los sistemas institucionales.' The left sidebar menu is identical to the previous screenshot. The main content area features a search input field with the placeholder text 'Buscar: Cédula de Identidad / Pasaporte.' and a search button. Below the search field, there is an 'Información' box with a question mark icon and a paragraph of text: 'Información.- La información que consta dentro del sistema, es utilizada con fines académicos y administrativos, por lo tanto es necesario mantenerla siempre actualizada.'