

“Servicio Web para la integración de la Plataforma Uniautónoma del Cauca Virtual (moodle) con el Sistema de Gestión Académica PowerCampus”



FABIAN ANDRES IBARRA FLOREZ

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS INFORMÁTICOS

POPAYAN

2020

“Servicio Web para la integración de la Plataforma Uniautónoma del Cauca Virtual (moodle) con el Sistema de Gestión Académica PowerCampus”



FABIAN ANDRES IBARRA FLOREZ

Trabajo de grado para obtener el título de Ingeniero de Sistemas

Director

Ingeniero

Gabriel Ángel Osorio

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS INFORMÁTICOS

POPAYAN

2020

NOTA DE ACEPTACIÓN

Popayán, Cauca. 8 de abril de 2020

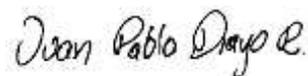
El director de trabajo de grado y los jurados de trabajo de grado en modalidad pasantía denominado “Servicio Web para la integración de la Plataforma Uniautónoma del Cauca Virtual (moodle) con el Sistema de Gestión Académica PowerCampus” realizado por el estudiante Fabian Andres Ibarra Florez, una vez revisado el informe final y aprobada la sustentación, autorizan para que se realicen los trámites concernientes para optar por el título profesional de Ingeniería de sistemas informáticos.



Firma Director



Firma Jurado



Firma Jurado

DEDICATORIA

Al ingeniero y docente Jimmy Andres Campo por transmitir esa pasión y amor nato que tiene por las ciencias de la computación, al ingeniero Gabriel Ángel Osorio por su paciencia y consejos de vida, al ingeniero Juan Pablo Diago por su amistad sincera y honesta, a mi padre Víctor Orlando Ibarra por su ejemplo de trabajo duro, perseverancia y amor por el saber, a la ingeniera María Alejandra Varona por acompañarme y apoyarme y creer en mí, en este largo proceso académico y ser parte de mi vida.

AGRADECIMIENTOS

Agradezco de manera especial y sincera al director de este trabajo Gabriel Ángel Osorio por el gran apoyo que me brindo, por transmitir su confianza y por orientarme con su gran conocimiento y experiencia en esta área.

También debo gratitud a todos los docentes y administrativos de la Corporación Universitaria Autónoma del Cauca por brindarme la colaboración en este proceso en mi crecimiento personal y profesional.

TABLA DE CONTENIDO

TABLA DE CONTENIDO	6
LISTA DE FIGURAS	8
LISTA DE TABLAS	9
LISTA DE ANEXOS	10
RESUMEN.....	11
ABSTRACT.....	12
INTRODUCCIÓN.....	13
1. CAPÍTULO I.....	14
1.1 Planteamiento del Problema.....	14
1.2 Justificación	14
1.3 Objetivos.....	15
1.4 Objetivo General.....	15
1.5 Objetivos Específicos.....	15
2 CAPÍTULO II. MARCO TEÓRICO.....	16
2.1 Estado del arte.....	16
2.1.1 Antecedentes.	16
2.1.2 Servicios Web (Web Services)	16
2.1.3 Computación en la nube.....	17
2.1.4 Arquitectura cliente-servidor.....	17
2.1.5 Hosting y dominio.....	17
2.1.6 SIA - POWER CAMPUS.....	17
2.1.7 LMS MOODLE	18
2.2 Herramientas de Apoyo.	18
2.2.1 Editor de VIM.	18
2.2.2 Moodle DEV.	18
2.2.3 PHP.....	18
2.2.4 Google Meet.....	18
2.2.5 Udemy.....	19
3 CAPÍTULO III. METODOLOGÍA.....	20
3.1 Metodología de la investigación	20
3.1.1 Procedimiento SIA (Power Campus).	21
3.2 Procedimientos UACV.	21

CAPITULO IV. INGENIERÍA DEL PROYECTO.....	24
3.3 Requerimientos funcionales.....	24
3.4 Requerimientos no funcionales.....	24
3.5 Historias de usuario.....	24
3.6 Estado inicial del proyecto.....	27
3.6.1 Servidor Uniautónoma del Cauca Virtual Moodle.....	27
3.7 Proceso de reingeniería de la arquitectura de virtualización.....	28
3.8 Scrum como marco de trabajo.....	29
3.9 Sprint 1.....	29
3.9.1 Análisis de datos que debe servir el SIA.....	29
3.9.2 Análisis de datos para la creación de cursos en UACV.....	29
3.9.3 Consulta de cursos al SIA.....	30
3.9.4 Matricula de docentes y estudiantes en cursos UACV.....	30
3.9.5 Tablas y campos necesarios para la consulta matricula docentes.....	31
3.9.6 Tablas y campos necesarios para la consulta matricula estudiantes.....	31
3.9.7 Problemas encontrados sprint 1.....	32
3.10 Sprint 2.....	33
3.10.1 Modelado de datos para el cliente UACV del servicio web.....	33
3.10.2 Diseño de arquitectura del servicio web.....	33
3.10.3 Diseño de la estructura de datos del servicio web.....	33
3.10.3.1 Análisis de la naturaleza de los campos de datos para estimar el tamaño de cada objeto JSON retornado por la consulta.....	34
3.10.4 Diseño arquitectura del backend para el SIA del servicio web.....	34
3.10.4.1 Diseño de seguridad para el servicio web.....	34
3.10.5 Diseño arquitectura del servicio web.....	34
3.10.6 Problemas encontrados sprint 2.....	35
3.11 Sprint 3.....	35
3.11.1 Instalación y configuración de la arquitectura del servicio web.....	35
3.11.2 Instalación de php 7.3.23.....	37
3.11.3 Programación del JSON.....	37
3.12 Configuración de servicios web en UACV.....	38
3.13 Activar Servicios Web en UACV.....	39
3.14 APIS.....	43
CAPÍTULO V. RESULTADOS.....	44
CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES.....	50

LISTA DE FIGURAS

Figura 1. Estadística instalaciones moodle en Colombia.....	16
Figura 2. Gestión de cursos y categorías UACV 2021	22
Figura 3. Tabla equivalencias-materias-moodle-2020	22
Figura 4. Arquitectura de virtualización	27
Figura 5. Actualización PHP 7.3.....	28
Figura 6 Puntos de montaje	28
Figura 7 Listado de Volúmenes de disco.....	29
Figura 8. Configuración libro de calificaciones UACV- Curso Administración Financiera	32
Figura 9. Diseño servicio REST-GET	33
Figura 10. Familia del sistema operativo	35
Figura 11. Sistema operativo reportado top500.org	35
Figura 12. Rendimiento sistema operativo reportado top500.org	36
Figura 13. Instalación Máquina Virtual Servidor Web Linux CentOS 7	37
Figura 14 Instalación de php 7.3.23	37
Figura 15.Herramienta POSTMAN REQUEST.....	38
Figura 16. JSON RESPONSE POSTMAN	38
Figura 17. Habilitar Servicios Web	39
Figura 18. Habilitar protocolo REST en UACV	39
Figura 19. Configuración usuario servicio web desde UACV.....	40
Figura 20. Lista de usuario administrador y servicio web UACV	40
Figura 21. Activar la documentación de servicios web en UACV	43
Figura 22. Arquitectura de servicios Uniautónoma del Cauca	44
Figura 23. Arquitectura AWS para UACV	45
Figura 24. Oferta económica aliado estratégico 1	45
Figura 25. Oferta económica aliado estratégico 2	46
Figura 26. Pagina principal de UACV moodle	46
Figura 27. Instalando Plugin wsuacv	47
Figura 28. Activando plugin wsuacv	47
Figura 29. configuración WSUACV REST.....	48
Figura 30. Método de matriculación WSUAC REST.....	48
Figura 31. Matricula masiva desde SIA Fuente.....	48
Figura 32. listado de matrícula WSUACV REST	49
Figura 33. confirmación de matrícula WSUACV REST	49

LISTA DE TABLAS

Tabla 1. Nomenclatura archivo plano matricula de cursos.	23
Tabla 2. Requerimientos funcionales	24
Tabla 3. Requerimientos no funcionales	24
Tabla 4. Historia de usuario consultar estudiantes	24
Tabla 5. Tarea Diseñar consulta estudiantes matriculados al servidor SIA	25
Tabla 6. Implementar APIREST estudiantes	25
Tabla 7. Tarea manejo de datos del APIREST estudiantes.....	25
Tabla 8. Historia de usuario consultar cursos.....	25
Tabla 9. Tarea diseñar consulta de cursos ofertados	25
Tabla 10. Tarea implementar APIREST cursos ofertados	26
Tabla 11. Tarea manejo de datos APIREST cursos ofertados	26
Tabla 12. Historia de usuario consultar docentes	26
Tabla 13. Tarea diseñar consulta de docentes al servidor SIA.....	26
Tabla 14. Tarea implementar APIREST docentes cursos asociados.....	26
Tabla 15. Tarea manejo del APIREST docentes cursos.....	27
Tabla 16. Requisitos mínimos del sistema operativo Linux CentOS 7	36
Tabla 17. Pasos activación de servicios web UACV	38
Tabla 18. APIS.....	43

LISTA DE ANEXOS

Anexo 1. Certificado de horas de pasantía TI Fabian Ibarra.

Anexo 2. Formato de evaluación previa de pasantía FORMATO No 2.

Anexo 3. Formato de evaluación de pasantía FORMATO No 3.

RESUMEN

La Corporación Universitaria Autónoma del Cauca en miras a mejorar sus servicios educativos, se ha logrado evidenciar cómo la educación cada día depende más de los avances tecnológicos, es evidente el uso del celular, las computadoras y actualmente los dispositivos vestibles o en inglés wearables en el aula de clase, siendo estos dispositivos en la mayoría de los casos los que terminan acaparando toda la atención del estudiante y tranzando el desarrollo de los planes de aula, exigiendo a los docentes la elaboración de los materiales de clase, la preparación de talleres y pruebas de aprendizaje, es así como se evidencia la necesidad de vincular los procesos académicos a los dispositivos móviles y hacerlos disponibles, atractivos visualmente, con planeación y objetivos pedagógicos medibles, claros y rastreables, para lo cual se ha implementado la plataforma LMS moodle, en la que es requerido matricular a docentes y estudiantes con sus roles correspondientes, crear las clases con los contenidos y actividades que permitan ser cuantificadas.

He aquí en este proceso de evaluación y matrícula una gran oportunidad para aplicar un proceso de optimización aplicando ingeniería y hacer que este LMS moodle pueda establecer comunicación con el sistema de gestión académica PowerCampus, y garantizar la continuidad, veracidad, fluidez y eficiencia de los procesos académicos, para lograrlo se utilizarán técnicas y documentación preestablecida llamada servicios web.

Palabras clave

Corporación Universitaria Autónoma del Cauca; LMS Learning Management System; Moodle; Servicios Web; Sistema de Gestión Académica.

ABSTRACT

The Corporación Universitaria Autónoma del Cauca in order to improve its educational services, it has been possible to show how education increasingly depends on technological advances, it is evident the use of the cell phone, computers and currently wearable devices in the classroom, it is these devices in most cases that end up hoarding all the attention of the student and transferring the development of classroom plans , requiring teachers to develop class materials, preparing workshops and learning tests, this is how the need to link academic processes to mobile devices is evident and make them available, visually appealing, with measurable, clear and traceable pedagogical planning and objectives, for which the LMS moodle platform has been implemented, in which it is required to enroll teachers and students with their corresponding roles , create classes with the contents and activities that allow to be quantified.

Here is in this evaluation and enrollment process a great opportunity to apply an optimization process applying engineering and make this LMS moodle able to establish communication with the PowerCampus academic management system, and ensure the continuity, veracity, fluidity and efficiency of academic processes, techniques and documentation called web services will be used to achieve this.

Keywords

Academic Management System; Corporación Universitaria Autónoma del Cauca; LMS Learning Management System; Moodle; Web Services.

INTRODUCCIÓN

En este documento encontrará el estudio de cómo desarrollar un Servicio Web entre los Sistemas PowerCampus (SIA) y LMS moodle (UACV), haciendo uso de la documentación de moodle para la implementación de servicio web y aplicando esta misma al sistema PowerCampus, con el propósito de lograr la interacción y comunicación entre estos dos sistemas de información.

En el documento se plantea la construcción de un plugin o extensión que habilita la interacción y comunicación entre los sistemas antes mencionados, además genera la posibilidad de servir información para otros sistemas.

Se investigó cuál sería la mejor técnica para realizar la comunicación de la información teniendo en cuenta los dos sistemas.

Se realizó el desarrollo y pruebas de integración que habilitan la comunicación entre los dos sistemas de información aplicando como referencia la metodología de trabajo en equipos SCRUM.

1. CAPÍTULO I.

1.1 Planteamiento del Problema

La Corporación Universitaria Autónoma del Cauca es una Institución de Educación Superior con más de 35 años que se encuentra establecida en la ciudad de Popayán que cuenta con 9 programas de pregrado y 4 de posgrado.

A pesar de las ventajas ofrecidas por la plataforma Uniautónoma del Cauca virtual (moodle) en el manejo de cursos virtuales en adelante denominada UACV, se ha detectado retrasos en los procesos académicos en especial durante las primeras semanas de cada semestre, los docentes quedan en cola de espera mientras se realiza la asignación de cursos en el sistema de información académico PowerCampus en adelante denominado SIA, luego se procede a realizar la solicitud de apertura de curso en UACV, el administrador de plataforma moodle crea el curso previa confirmación en SIA y por fin el docente puede acceder al curso en UACV. Para el caso de los estudiantes docentes o administrativos que deseen acceder a un recurso en plataforma en calidad de estudiantes, cuentan con dos opciones de matrícula en la primera opción y la más segura, deben ser matriculados de forma manual al curso por el docente o el administrador de plataforma, en la segunda opción es contar con un código de auto matriculación que será pre compartido. Como se evidencia en las dos opciones se incurre en tiempos de espera y dificulta el desarrollo natural del proceso académico además es posible que un estudiante se matricule por su cuenta con la clave pre compartida en un curso para el cual no está debidamente matriculado. Es así como para la creación de cursos existe la necesidad de contar con un servicio que establezca la integración de información con el SIA, donde toda la comunidad educativa administrativos, docentes y estudiantes, mejoren sus actividades en los procesos de creación de cursos virtuales, matriculación de estudiantes a los cursos virtuales y la asignación de docentes por curso, proporcionando normalidad académica, coherencia y seguridad de la información. Es así como la Corporación Universitaria Autónoma del Cauca la Vicerrectoría Académica y su plataforma virtual de aprendizaje UACV, requiere un servicio que permita automatizar los procesos administrativos antes mencionados.

1.2 Justificación

El Servicio web es una propuesta técnicamente viable y genérica que permitirá y facilitará la consulta de matrícula académica de estudiantes, labor académica de docentes y administrativos, registro y reporte de notas, registro y reporte de asistencias. Se podrán comunicar con el SIA y UACV independientemente del lenguaje de programación o de la plataforma sobre la cual se ha implementado cada

uno de estos sistemas, definiendo solo una técnica de acceso a la información de forma eficiente y veraz manteniendo las políticas de seguridad pertinentes.

Este desarrollo permitirá apoyar la toma de decisiones académicas gracias al flujo de información actualizada.

Es importante aclarar que los sistemas servirán y consumirán la información proporcionando una comunicación transparente con protocolos bien definidos debido a que PowerCampus es el sistema de información académica y personal para estudiantes y docentes su información no podrá ser modificada por moodle y será PowerCampus quien decida cuándo consumir la información.

SOA: Arquitectura Orientada a Servicios.

REST: Transferencia de Estado Representacional.

1.3 Objetivos

1.4 Objetivo General

Implementar un Servicio Web para la integración de la Plataforma Uniautónoma del Cauca Virtual (moodle) con el Sistema de Gestión Académica PowerCampus.

1.5 Objetivos Específicos

- Modelar un Servicio Web que permita la integración Moodle - PowerCampus.
- Diseñar la arquitectura que permita la implementación del Servicio Web.
- Realizar las pruebas de consulta y registro de los datos por medio del Servicio Web.
- Publicar el Servicio Web en la página web de los servidores de la Corporación Universitaria Autónoma del Cauca.

2 CAPÍTULO II. MARCO TEÓRICO

2.1 Estado del arte.

2.1.1 Antecedentes.

En la actualidad moodle cuenta con 107,384 sitios reportados 19,137,213 de cursos y más de 16'251.002 usuarios registrados a nivel mundial, las cifras para Suramérica posicionan a Colombia como el segundo país con 2646 (MOODLE, 2021) instalaciones reportadas a moodle.org, estas cifras posicionan a moodle como el LMS más utilizado a nivel mundial, por tratarse de un desarrollo de código libre con origen en Australia la comunidad alrededor de este sistema crece cada día, cuenta con portal de complementos y su equipo de desarrollo se preocupa por incluir nuevas funcionalidades al sistema base y para ayudar su expansión han desarrollado un conjunto de funciones que permiten desarrollar servicios web en miras a comunicar moodle con otros sistemas de información, hay pocos casos de éxito documentados de forma oficial.

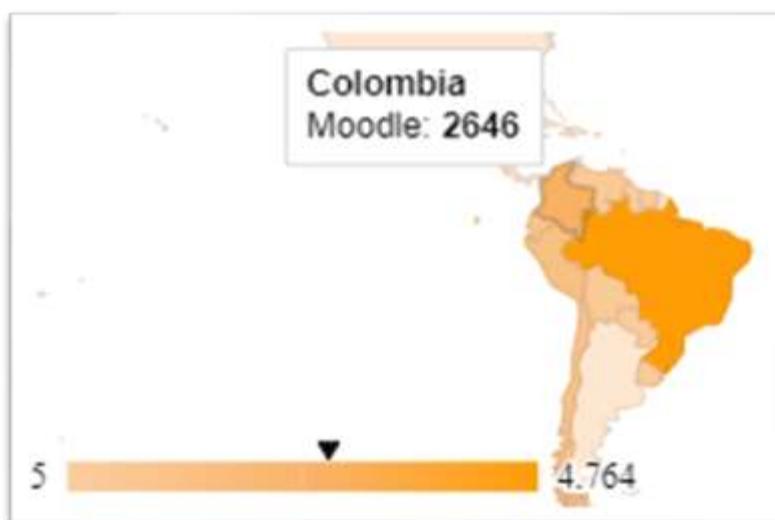


Figura 1. Estadística instalaciones moodle en Colombia
Fuente. moodle.org (MOODLE, 2021)

2.1.2 Servicios Web (Web Services)

Un servicio web (en inglés, web service o web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las

organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web (W3, 2020).

Un servicio web es un sistema software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable. Cuenta con una interfaz descrita en un formato procesable por un equipo informático (específicamente en WSDL), a través de la que es posible interactuar con el mismo mediante el intercambio de mensajes SOAP, típicamente transmitidos usando serialización XML sobre HTTP juntamente con otros estándares web (W3, 2020).

2.1.3 Computación en la nube.

La computación en la nube consiste en usar recursos de cómputo (almacenamiento de datos, servidores, aplicaciones, servicios de red). Con el fin de utilizarlos a través de Internet y bajo un modelo de consumo en demanda, es decir, como se contrata la energía eléctrica, el agua, conexión a internet y TV. La corporación Universitaria Uniautónoma del Cauca migrara a la computación en la nube para prestar un servicio a la comunidad universitaria estudiantes, docentes, administrativos y al público en general.

2.1.4 Arquitectura cliente-servidor.

Actualmente el sistema de UACV está basado en una arquitectura cliente servidor, ésta, se caracteriza por seguir un modelo de aplicación distribuida en el que las tareas se reparten entre los servidores y los clientes, es decir, los clientes hacen peticiones a los servidores y estos dan respuesta logrando así su funcionalidad. Para lograr hacer una petición desde un cliente al servidor es necesario que el cliente tenga una conexión a Internet. En su concepción tradicional, la arquitectura cliente-servidor es un modelo de sistemas distribuidos en donde múltiples clientes realizan múltiples peticiones a múltiples servidores, que procesan la información y devuelven los resultados a quien corresponda (Bircher, 2017).

2.1.5 Hosting y dominio.

Hosting es el servicio que se contrata para poder almacenar toda la información que se desea servir a los usuarios de internet (imágenes, vídeos, o cualquier contenido accesible vía web), actualmente UACV se encuentra virtualizado en sus propios servidores, arquitectura conocida como on-premise.

2.1.6 SIA - POWER CAMPUS

Ellucian PowerCampus™ apoya a las universidades a enlazar personas, procesos y tecnologías y mejorar el desempeño institucional, creando un ambiente flexible para los alumnos, personal docente y administrativo.

PowerCampus permite a las instituciones de educación superior—en particular aquellas que se encuentran en proceso de crecimiento—automatizar los procesos

críticos, tanto académicos como administrativos: reclutamiento, admisión, registro académico, egreso, cobranza y emisión de recibos (ellucian, 2021).

2.1.7 LMS MOODLE

Moodle es un sistema gratuito para el Manejo del Aprendizaje en línea, que les permite a los educadores la creación de sus propios sitios web privados, llenos de cursos dinámicos que extienden el aprendizaje, en cualquier momento, en cualquier sitio (MOODLE, 2020).

2.2 Herramientas de Apoyo.

2.2.1 Editor de VIM.

Vim es un editor de texto que es compatible con Vi. Se puede utilizar para editar todo tipo de texto sin formato. Es especialmente útil para editar programas, consume pocos recursos y puede ser ejecutado desde una terminal de comandos.

2.2.2 Moodle DEV.

Cada desarrollador crea herramientas sencillas para evitar repetir tareas engorrosas y/o aburridas, y es precisamente por eso que se ha creado MDK: empaquetar todas esas herramientas útiles de forma portátil a través de sistemas. Inicialmente desarrollado en Bash, el proyecto se trasladó a Python para evitar lidiar con inconsistencias entre plataformas Unix y, finalmente, para admitir Windows (vim, 2021).

2.2.3 PHP.

PHP es un popular lenguaje de scripting de propósito general que es especialmente adecuado para el desarrollo web.

Rápido, flexible y pragmático, PHP impulsa todo, desde su blog hasta los sitios web más populares del mundo (PHP, 2021)

2.2.4 Google Meet

Hangouts Meet es una nueva experiencia de reunión de video con un objetivo: hacer que unirse a las reuniones sea sin esfuerzo para que las personas puedan ser tan productivas como lo son cuando están cara a cara. Hemos escuchado constantemente de los clientes acerca de maneras en que podemos mejorar Hangouts, como facilitar el trabajo con clientes externos o reducir el "tiempo para comenzar" una reunión a cero. Es por eso que hemos construido Hangouts Meet para tener una interfaz ligera, rápida y una gestión inteligente de los participantes (Google, 2021).

2.2.5 Udemy.

La tienda virtual de aprendizaje y enseñanza global líder, curso vim.

BASES TEÓRICAS

Interoperabilidad: La interoperabilidad, que se define como la capacidad que exhiben los sistemas para compartir datos y procesos (Zhang, Jiang, Yue, & Gong, 2020) esta permite a los sistemas intercambiar datos y ejecutar tareas en conjunto.

Arquitectura Orientada a Servicios (SOA): SOA es un modelo de componentes que interrelaciona las diferentes unidades funcionales de una aplicación, llamadas servicios, a través de interfaces bien definidas entre dichos servicios. Las interfaces se defiende una manera neutral, independiente de la plataforma de hardware, sistema operativo, o lenguaje de programación en el que el servicio se implementa. Esto permite que los servicios, construidos sobre una gran variedad de tecnologías, puedan interactuar unos con otros de una manera uniforme y universal (Bolo, 2006).

JSON (JavaScript Object Notation- Notación de Objetos de JavaScript): Es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros (json, 2020).

Servicios Web (Web Service): Los servicios web son sistemas de software que permiten el intercambio de datos y funcionalidad entre aplicaciones sobre una red (Morales, 2010).

Metodología SCRUM: Scrum es un marco de trabajo para desarrollar, entregar y mantener productos complejos. Esta definición consiste en los roles, eventos y artefactos de Scrum y las reglas que los relacionan (Schwaber & Sutherland, 2016).

3 CAPÍTULO III. METODOLOGÍA

3.1 Metodología de la investigación

La metodología de investigación que se empleó es aplicada ya que se implementa una tecnología conocida, probada y aceptada en la industria para realizar el intercambio de información entre aplicativos de software.

Con el paso del tiempo los lenguajes de programación han brindado infinidad de herramientas que permiten la creación de un desarrollo web de calidad, En nuestro proyecto emplearemos las cuatro fases para la creación de software: Análisis, Diseño, implementación y pruebas.

En la primera fase se hará la toma de requerimientos y datos útiles para la toma de decisiones. En la segunda fase se procederá a diseñar la estructura del sistema de información con su respectiva plataforma web. En la tercera fase se dará paso a implementar el sistema de información y plataforma web. Por último, se harán validaciones y pruebas del producto.

Para la elaboración del servicio web, se optó por el uso de las metodologías ágiles, específicamente las que brinden apoyo en la ingeniería de requerimientos. Aquí se rescatan tres metodologías (XP, Scrum, RUP) como las que abarcan esta etapa de las cuales se decidió emplear SCRUM como marco de trabajo y como referente la metodología XP, (programación extrema) agrupando un conjunto de reglas, que se desarrollan en cuatro actividades: planeación, diseño, codificación y pruebas XP se caracteriza por la flexibilidad y facilidad para incorporar en las prácticas de desarrollo de software de equipos pequeños y medianos, que trabajen en proyectos donde los requisitos cambien rápidamente.

Scrum, es un marco de trabajo basado en los métodos ágiles, que tiene como objetivo el control continuo sobre el estado actual del software, en el cual el cliente establece las prioridades y el equipo Scrum se autoorganiza para determinar la mejor forma de entregar resultados.

El marco de trabajo SCRUM usa tres roles y unos elementos que implementaremos en nuestro desarrollo:

Roles

- **Product Owner:** Representa a todos los interesados en el producto final que en este caso sería la comunidad en general.
- **Scrum Master:** Ayuda al grupo del producto a aprender y aplicar scrum para conseguir valor de negocio. El scrum master hace todo lo que sea necesario para ayudar a que el equipo tenga éxito.

- **Team:** Equipo de trabajo. Responsable de transformar el Backlog de la iteración en un incremento de la funcionalidad del software, es decir, de convertir el product backlog en un software entregable.

En primear instancia se realizó un análisis del proceso de registro y consulta de matrícula en el SIA, también se analizó el proceso de alistamiento de plataforma UACV, para la creación de cursos, matrícula de docentes y estudiantes.

3.1.1 Procedimiento SIA (Power Campus).

Estudiantes: Existen dos casos, estudiantes regulares y estudiantes nuevos.

Cursos: se conoce como oferta académica y se establece desde registro y control académico con la información de labor académica aprobada por la Rectoría y Vicerrectoría Académica.

Docentes: el profesorado es asignado a los cursos desde la oficina de registro y control académico con la información de labor académica aprobada por la Rectoría y Vicerrectoría Académica.

El reporte de Cursos ofertados se genera directamente mediante consulta a la base de datos del SIA, teniendo en cuenta filtrar el año y periodo académico en curso, luego es enviado por correo electrónico en formato hoja de cálculo al administrador de UACV.

3.2 Procedimientos UACV.

Creación de cursos y categorías.

El administrador debe realiza las siguientes actividades:

- Crea las categorías de nivel superior una para el año en curso.
- Crea la categoría para el periodo académico.
- Crea una subcategoría para pregrados y una subcategoría para posgrados.
- Para pregrado crea una subcategoría por facultad y una subcategoría para los cursos de campo común.
- Crea una subcategoría por posgrado.
- Cada categoría creada en UACV cuenta con un numero identificador o id el cual permite relacionar los cursos pertenecientes a cada facultad.

actividades y recursos, durante el proceso de creación de la nueva oferta académica.

- Una vez se consolida la información se genera un archivo plano en formato CSV (texto separado por comas), el archivo se nombra según la nomenclatura F-UACV-MC-1P21-M1, el separador de listas debe ser punto y coma (;) y la codificación del archivo UTF8 sin BOM.

Tabla 1. Nomenclatura archivo plano matricula de cursos.

F	UACV	MC	1P21	M1	18 FEB
FORMATO	UNIAUTONOMA DEL CAUCA VIRTUAL	MATRICULA CURSOS	PRIMER PERIODO DEL AÑO 2021	MODULO 1	FECHA DE CONSULTA AL SIA

Fuente. Elaboración propia.

Este procedimiento se debe repetir cuantas veces sea modificada la labora académica.

Matricula de docentes en UACV.

El administrador de UACV valida la información de los docentes con los cursos equivalente o integrados según la tabla de equivalencias (este procedimiento se realiza a de forma manual utilizando hojas de cálculo, no existe información actualizada en los sistemas de información), se genera un archivo plano en formato csv (texto separado por comas), el archivo se nombra según la nomenclatura F-UACV-MD-1P21-M1, variando solo la 9 letra la cual indica si se trata de curso (C), docente (D), estudiante (E).

Matricula de estudiantes en UACV.

El administrador de UACV valida la información de los estudiantes con los cursos equivalente o integrados según el formato de matrícula de cursos F-UACV-MC (este procedimiento se realiza a de forma manual utilizando hojas de cálculo, no existe información actualizada en los sistemas de información), se genera un archivo plano en formato csv (texto separado por comas), cambiando el nombre del archivo en este caso F-UACV-ME-1P21-M1, variando solo la 9 letra la cual indica si se trata de curso (C), docente (D), estudiante (E).

Estos procedimientos varían entre 2 y 6 horas, dependiendo de las modificaciones a la labor académica.

CAPITULO IV. INGENIERÍA DEL PROYECTO

3.3 Requerimientos funcionales

Tabla 2. Requerimientos funcionales

LISTA DE REQUERIMIENTOS FUNCIONALES		
CÓDIGO	NOMBRE DEL REQUERIMIENTO	DESCRIPCIÓN
RF1	Consultar estudiantes	Consultar estudiantes en el SIA y en UACV
RF2	Consultar cursos	Consultar cursos en el SIA y en UACV
RF3	Consultar docentes	Consultar docentes en SIA y UACV
RF4	Matricular estudiantes	Registrar estudiantes según el curso en UACV.
RF5	Crear cursos	Crear cursos en UACV.
RF6	Matricular docentes	Registrar docentes según el curso en UACV.

Fuente. Elaboración propia

3.4 Requerimientos no funcionales

Tabla 3. Requerimientos no funcionales

LISTA DE REQUERIMIENTOS NO FUNCIONALES		
CÓDIGO	NOMBRE DEL REQUERIMIENTO	DESCRIPCIÓN
RNF1	Requisitos de rendimiento	Pruebas de los datos solicitados al SIA utilizando APIREST para consumir en UACV.
RNF2	Seguridad	Comunicación de datos entre SIA y UACV de forma segura.
RNF3	Disponibilidad	Disponibilidad depende de la infraestructura establecida.
RNF4	Usabilidad	Se basa en la API de moodle para Front-end.

Fuente. Elaboración propia

3.5 Historias de usuario.

Las historias de usuario contienen información útil sobre las funcionalidades que debe atender el servicio web, describiendo en cada historia de usuario una tarea solida que aporta valor al estudiante de pasantía permitiendo evidenciar lo que se quiere desarrollar en el sistema.

Tabla 4. Historia de usuario consultar estudiantes

Historia de usuario: HU1	
Programador responsable: Fabian Andres Ibarra Florez	
Nombre de la historia de usuario: Consultar estudiantes	
Nivel de riesgo de desarrollo: Alto	
Peso de la historia de usuario: 20 puntos de la historia	
Tiempo estimado: 50 horas	
Perspectiva del producto	
Como: docente, administrador o gestor en el contexto del curso	
Se requiere: consultar la lista de estudiantes en el SIA	
Para: actualizar la matricula en el contexto del curso en UACV.	
Criterios de aceptación	
Colección de objetos en formato JSON que representa la matricula académica del estudiante.	
Actividades	Horas de implementación
• Diseñar consulta estudiantes matriculados al servidor SIA	30
• Implementar APIREST	10
• Manejo de datos del APIREST	10

Fuente. Elaboración propia

Tabla 5. Tarea Diseñar consulta estudiantes matriculados al servidor SIA

Tarea de ingeniería	
Número de tarea: 1	Historia de usuario: HU1
Nombre de tarea: Diseñar consulta estudiantes matriculados al servidor SIA	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 30 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: Crear la consulta en lenguaje SQL que permitirá retornar los usuarios matriculados según el año, periodo y curso.	

Fuente. Elaboración propia

Tabla 6. Implementar APIREST estudiantes

Tarea de ingeniería	
Número de tarea: 2	Historia de usuario: HU1
Nombre de tarea: Implementar APIREST consulta estudiantes	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 10 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: generar un método que reciba un identificador de curso y permita consultar los estudiantes matriculados y retorne un objeto de tipo JSON con los usuarios matriculados.	

Fuente. Elaboración propia

Tabla 7. Tarea manejo de datos del APIREST estudiantes

Tarea de ingeniería	
Número de tarea: 3	Historia de usuario: HU1
Nombre de tarea: Manejo de datos del APIREST	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 10 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: solicitar al APIREST la lista de usuarios matriculados, pasando como parámetro el identificador de curso, se espera como valor de retorno una colección de objetos de tipo JSON con los usuarios matriculados.	

Fuente. Elaboración propia

Tabla 8. Historia de usuario consultar cursos

Historia de usuario: HU2	
Programador responsable: Fabian Andres Ibarra Florez	
Nombre de la historia de usuario: Consultar cursos	
Nivel de riesgo de desarrollo: Alto	
Peso de la historia de usuario: 20 puntos de la historia	
Tiempo estimado: 50 horas	
Perspectiva del producto	
Como: administrador en el contexto del sistema	
Se requiere: consultar la lista de cursos y grupos en el SIA	
Para: crear los cursos en el contexto del sistema en UACV.	
Criterios de aceptación	
Colección de objetos en formato JSON con cursos y sus respectivos grupos listo para consumir por UACV	
Actividades	Horas de implementación
• Diseñar consulta de cursos ofertados al servidor SIA	30
• Implementar APIREST	10
• Manejo de datos del APIREST	10

Fuente. Elaboración propia

Tabla 9. Tarea diseñar consulta de cursos ofertados

Tarea de ingeniería	
Número de tarea: 1	Historia de usuario: HU2
Nombre de tarea: Diseñar consulta cursos ofertados al servidor SIA	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 30 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: Crear la consulta en lenguaje SQL que permitirá retornar los cursos ofertados según el año, periodo activo.	

Fuente. Elaboración propia

Tabla 10. Tarea implementar APIREST cursos ofertados

Tarea de ingeniería	
Número de tarea: 2	Historia de usuario: HU2
Nombre de tarea: Implementar APIREST	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 10 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: generar un método que recibe el año y periodo activo y retorne un objeto de tipo JSON con los cursos activos.	

Fuente. Elaboración propia

Tabla 11. Tarea manejo de datos APIREST cursos ofertados

Tarea de ingeniería	
Número de tarea: 3	Historia de usuario: HU2
Nombre de tarea: Manejo de datos del APIREST	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 10 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: solicitar al APIREST la lista de cursos ofertados, pasando como parámetro el año y periodo, se espera como valor de retorno una colección de objetos de tipo JSON.	

Fuente. Elaboración propia

Tabla 12. Historia de usuario consultar docentes

Historia de usuario: HU3	
Programador responsable: Fabian Andres Ibarra Florez	
Nombre de la historia de usuario: Consultar docentes	
Nivel de riesgo de desarrollo: Alto	
Peso de la historia de usuario: 20 puntos de la historia	
Tiempo estimado: 50 horas	
Perspectiva del producto	
Como: administrador o gestor en el contexto del curso	
Se requiere: consultar la lista de docentes en el SIA	
Para: actualizar la matricula docente en el contexto del curso en UACV.	
Criterios de aceptación	
Generar un listado de docentes con sus respectivos grupos listo para consumir por UACV.	
Actividades	Horas de implementación
<ul style="list-style-type: none"> • Diseñar consulta de docentes al servidor SIA. • Implementar APIREST • Manejo de datos del APIREST 	<p>30</p> <p>10</p> <p>10</p>

Fuente. Elaboración propia

Tabla 13. Tarea diseñar consulta de docentes al servidor SIA

Tarea de ingeniería	
Número de tarea: 1	Historia de usuario: HU3
Nombre de tarea: Diseñar consulta de docentes al servidor SIA	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 30 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: Crear la consulta en lenguaje SQL que permitirá retornar los docentes con los cursos asociados según el año y periodo activo.	

Fuente. Elaboración propia

Tabla 14. Tarea implementar APIREST docentes cursos asociados

Tarea de ingeniería	
Número de tarea: 2	Historia de usuario: HU3
Nombre de tarea: Diseñar consulta de docentes al servidor SIA	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 30 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: Crear la consulta en lenguaje SQL que permitirá retornar los docentes con los cursos asociados según el año y periodo activo.	

Fuente. Elaboración propia

Tabla 15. Tarea manejo del APIREST docentes cursos

Tarea de ingeniería	
Número de tarea: 3	Historia de usuario: HU3
Nombre de tarea: Manejo de datos del APIREST	
Tipo de área: Desarrollo	Puntos estimados: 20
Tiempo de implementación estimado: 10 horas	
Programador responsable: Fabian Andres Ibarra Florez	
Descripción: solicitar al APIREST la lista de docentes con los cursos asociados pasando como parámetro el año y periodo, se espera como valor de retorno una colección de objetos de tipo JSON.	

Fuente. Elaboración propia

3.6 Estado inicial del proyecto

Para iniciar con el proyecto se realiza un diagnóstico de la arquitectura de información para UACV identificando que el proyecto se encuentra virtualizado en las instalaciones de Corporación Universitaria Autónoma del Cauca, a continuación, los hallazgos.

3.6.1 Servidor Uniautónoma del Cauca Virtual Moodle

El servidor moodle se encuentra virtualizado con las siguientes características:

El servidor anfitrión corresponde a un servidor físico DELL PowerEdge R310 el cual cuenta con un procesador Intel Xeon X3430 doble núcleo y 4 hilos a 2.4 GHz, 24GB RAM, 1TB de almacenamiento en RAID 5, 4 interfaces de Red Intel Gigabit ET Quad Port NIC PCIe-4, fuente de poder redundante de 400W.

Software: Sistema Operativo Windows server 2008 R2 64bits, ORACLE VirtualBox, IIS (servidor web internet information services) con ROL DNS.

Máquina virtual moodle corresponde a una instancia de tipo Linux arquitectura de 64 bits, procesadores lógicos asignados 2 al 100%, 4GB RAM, almacenamiento 1 VDI de 100GB, interfaz de red en modo puente asignada a "Intel Gigabit ET Quad Port Server Adapter 3".

Software: Sistema Operativo GNU/Linux CentOS 6.10 64 bits, ejecuta servidor NGINX, php5.4, servidor de correo DOVECOT y POSTFIX.



Figura 4. Arquitectura de virtualización

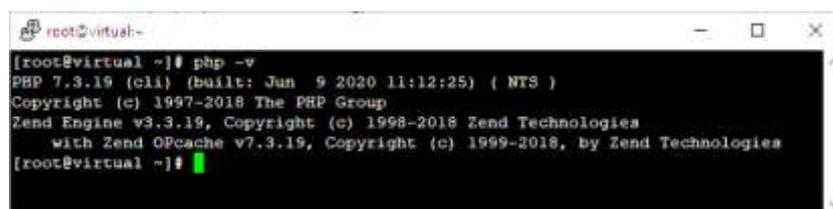
Fuente: Elaboración propia

3.7 Proceso de reingeniería de la arquitectura de virtualización

Se requiere un proceso de actualización de la máquina virtual dedicada a ejecutar moodle lo cual obliga a realizar la configuración del servidor y la máquina virtual para atender un mayor volumen de usuarios durante el periodo intersemestral aumentando la capacidad actual de 500 a 1000 usuarios.

Características máquina virtual moodle UACV: Instancia de tipo Linux, arquitectura de 64 bits, incrementa procesadores lógicos 4 al 100%, incrementa memoria 8GB RAM, almacenamiento 1, incremento en 50GB para alcanzar un total VDI de 150GB y 1 disco VHD de 50G, interfaz de red en modo puente asignada a “Intel Gigabit ET Quad Port Server Adapter 3”.

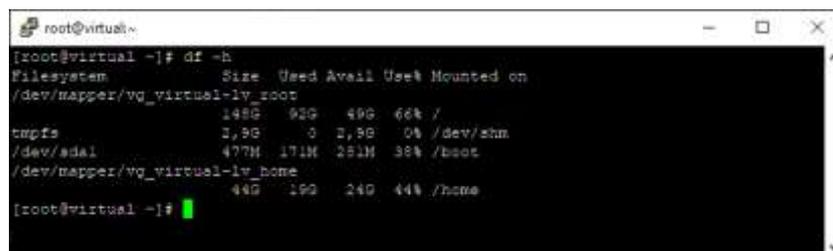
Software máquina virtual moodle UACV: Sistema Operativo GNU/Linux CentOS 6.10 64 bits, ejecuta servidor NGINX, se actualizó php7.3.19 utilizando repositorios oficiales de la comunidad de Fedora, servidor de correo DOVECOT y POSTFIX se mantienen sin cambios.



```
root@virtual:~# php -v
PHP 7.3.19 (cli) (built: Jun  9 2020 11:12:25) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.19, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.19, Copyright (c) 1999-2018, by Zend Technologies
root@virtual:~#
```

Figura 5. Actualización PHP 7.3
Fuente Elaboración propia

Reasigna el almacenamiento en el servidor para soportar las nuevas cargas de trabajo visualización de los puntos de montaje, se evidencia el uso de volúmenes de disco.



```
root@virtual:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg_virtual-lv_root
                148G  92G  49G   64% /
tmpfs           1.9G   0  1.9G   0% /dev/shm
/dev/sdal       477M  171M  251M   36% /boot
/dev/mapper/vg_virtual-lv_home
                44G   19G  24G   44% /home
root@virtual:~#
```

Figura 6 Puntos de montaje
Fuente Elaboración propia

El sistema se ha implementado utilizando volúmenes de disco lo cual permite agregar nuevos discos a los volúmenes existentes aumentando la capacidad de almacenamiento de forma dinámica sin alterar la configuración de los servicios o el sistema operativo.

```

root@virtual:~# lsblk
Logical volumes ---
LV Path                /dev/mapper/centos-root
LV Name                 lv_root
VG Name                 vg_virtual
LV UUID                 40000000-0000-0000-0000-000000000000
LV Write Access         read/write
LV Creation Host, time virtual.machine.com, 2014-08-12 18:28:08 -0500
LV Status               available
# open                  1
LV Size                 100.00 MiB
Current LE              2554
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device            253:0

Logical volumes ---
LV Path                /dev/mapper/centos_home
LV Name                 lv_home
VG Name                 vg_virtual
LV UUID                 40000000-0000-0000-0000-000000000000
LV Write Access         read/write
LV Creation Host, time virtual.machine.com, 2014-08-12 18:28:12 -0500
LV Status               available
# open                  1
LV Size                 49.51 MiB
Current LE              11594
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device            253:2

Logical volumes ---
LV Path                /dev/mapper/centos_swap
LV Name                 lv_swap
VG Name                 vg_virtual
LV UUID                 40000000-0000-0000-0000-000000000000
LV Write Access         read/write
LV Creation Host, time virtual.machine.com, 2014-08-12 18:28:13 -0500
LV Status               available
# open                  1
LV Size                 4.00 MiB
Current LE              1280
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device            253:3
root@virtual:~#

```

Figura 7 Listado de Volúmenes de disco
Fuente Elaboración propia.

3.8 Scrum como marco de trabajo

Scrum (n): Un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente (Schwaber & Sutherland, 2016).

Scrum es adoptado en los equipos de desarrollo de software como un marco de trabajo aplicando los métodos ágiles, que tiene como objetivo el control continuo sobre el estado actual del software, en el cual el cliente establece las prioridades y el equipo Scrum se autoorganiza para determinar la mejor forma de entregar resultados.

3.9 Sprint 1

Se obtiene el diagrama de entidad relación de moodle.

3.9.1 Análisis de datos que debe servir el SIA.

El sistema de información académico debe proporcionar a moodle la información necesaria para la creación de cursos ofertados en el periodo, los docentes a matricular por curso y los estudiantes a matricular por curso

3.9.2 Análisis de datos para la creación de cursos en UACV

Según la documentación de moodle para la creación de cursos por volumen se requieren los siguientes campos:

Tabla 3. Campos creación cursos en moodle

CAMPO	DESCRIPCION	CONDICION
shortname	El nombre corto del curso	obligatorio
fullname	El nombre completo del curso (obligatorio).	obligatorio
category	Esta es el identificador en la base de datos de la categoría (encontrada en la URL de la categoría) en donde acomodar al curso (obligatorio).	obligatorio
idnumber	El número ID (identificación) del curso	Opcional
visible	1 si el curso es visible, 0 si está oculto	Opcional
startdate	La hora en que inicia el curso. Por favor, tome nota de que este valor es pasado a la función PHP strtotime para generar el sello-de-tiempo (timestamp).	Opcional
summary	El resumen del curso.	Opcional
format	El formato de curso a emplear, esto debe de ser un nombre válido de plugin de formato de curso; por ejemplo: weeks, topics.	Opcional
lang	El idioma para usar.	Opcional
newsitems	El número de ítems de noticias	Opcional
showgrades	1 para mostrarles el Libro de calificaciones a los estudiantes y 0 para ocultarlo.	Opcional
showreports	1 para mostrar los reportes de actividades y 0 para ocultarlos.	Opcional
legacyfiles	1 para habilitar los Archivos de curso heredados y 0 para no habilitarlos.	Opcional
maxbytes	El tamaño en bytes máximo de archivo a subir en el curso. Use 0 para emplear el límite del sitio.	Opcional
groupmode	0 para Sin grupos, 1 para Grupos separados y 2 para Grupos visibles.	Opcional
groupmodeforce	1 para forzar el modo de grupo; de otra forma usar 0.	Opcional
enablecompletion	1 para habilitar Finalización de actividad y 0 para no hacerlo.	Opcional

Fuente. Elaboración propia basada en documentación moodle

Para dar cumplimiento a este requisito se consultar a la base de datos del SIA (sistema de información académico PowerCampus) y se establecen los parámetros para cada uno de los campos en concordancia con los parámetros institucionales con los que iniciara cada curso, a continuación, se mencionan la tabla y los campos involucrada en la consulta al SIA por motivos de seguridad la consulta no será publicada en este texto.

3.9.3 Consulta de cursos al SIA

Tabla: SECTIONS

Campos SECTIONS: ACADEMIC_YEAR, ACADEMIC_TERM, ACADEMIC_SESSION, CLASS_LEVEL, SECTION, EVENT_ID, EVENT_LONG_NAME, CURRICULUM, DEPARTMENT, START_DATE, END_DATE

Se crea el procedimiento almacenado UACV_MCURSOS.

3.9.4 Matricula de docentes y estudiantes en cursos UACV

Según la documentación de moodle para la matrícula de usuarios en cursos se asigna mediante roles por lo tanto los campos mínimos para realizar este procedimiento de forma masiva se requieren los siguientes campos:

Tabla 4. Campos matricula de usuarios en moodle

CAMPO	DESCRIPCION	CONDICION
username	Este campo de username (nombre_de_usuario) solamente puede contener letras alfabéticas minúsculas, números, guión '-', guión_bajo '_', punto '.', o signo de arroba '@'.	Obligatorio
firstname	puede contener letras alfabéticas, máximo 100 caracteres.	Obligatorio
lastname	puede contener letras alfabéticas, máximo 100 caracteres	Obligatorio
email	Debe ser de la forma nombre@dominio_de_correo	Obligatorio
auth	El campo auth (autenticación) debe usarse si el sitio emplea un método de Autenticación alternativo, como por ejemplo LDAP, porque de lo contrario el método de autenticación por defecto será manual y los usuarios que empleen un método de autenticación diferente no podrán ingresar al sitio. Use los códigos de nombre_corto definidos en Plugins > Autenticación para los diferentes tipos, como por ejemplo, Cuentas manuales, Sin ingreso, LDAP, CAS, MNet, Autenticación por BasedeDatos externa, Sin Autenticación. Si Usted no incluye una columna de autenticación auth, entonces los nuevos usuarios recién creados serán creados con el tipo de cuenta manual. Usted puede configurar "auth" a "nologin" (Sin ingreso) en su archivo CSV, lo que significará que los usuarios creados nuevos no podrán ingresar al sitio.	Opcional
idnumber	Máximo 255 caracteres.	Opcional

Elaboración. Fuente propia basada en documentación moodle.

Para dar cumplimiento a este requisito se consultará a la base de datos del SIA (sistema de información académico PowerCampus) y se establecen los parámetros para cada uno de los campos en concordancia con los parámetros institucionales, a continuación, se mencionan la tabla y los campos involucrada en la consulta al SIA, por motivos de seguridad la consulta no será publicada en este texto.

3.9.5 Tablas y campos necesarios para la consulta matricula docentes

Tablas: SECTIONPER, PEOPLE, ADDRESS”.

Campos SECTIONPER: ACADEMIC_YEAR, ACADEMIC_TERM, ACADEMIC_SESSION, SECTION, EVENT_ID, PERSON_CODE_ID.

Campos PEOPLE: PEOPLE_CODE_ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, Last_Name_Prefix.

Campos ADDRESS: PEOPLE_CODE_ID, EMAIL_ADDRESS.

3.9.6 Tablas y campos necesarios para la consulta matricula estudiantes

Tabla: TRANSCRIPTDETAIL

Campos: ACADEMIC_YEAR, ACADEMIC_TERM, ACADEMIC_SESSION, SECTION, EVENT_ID, PERSON_CODE_ID.

Tabla: ACADEMIC

Campos: ACADEMIC_YEAR, ACADEMIC_TERM, ACADEMIC_SESSION, SECTION, PERSON_CODE_ID, CURRICULUM, DEPARTMENT.

Tabla: PEOPLE

Campos: PEOPLE_CODE_ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, Last_Name_Prefix.

Tabla: ADDRESS:

Campos: PEOPLE_CODE_ID, EMAIL_ADDRESS.

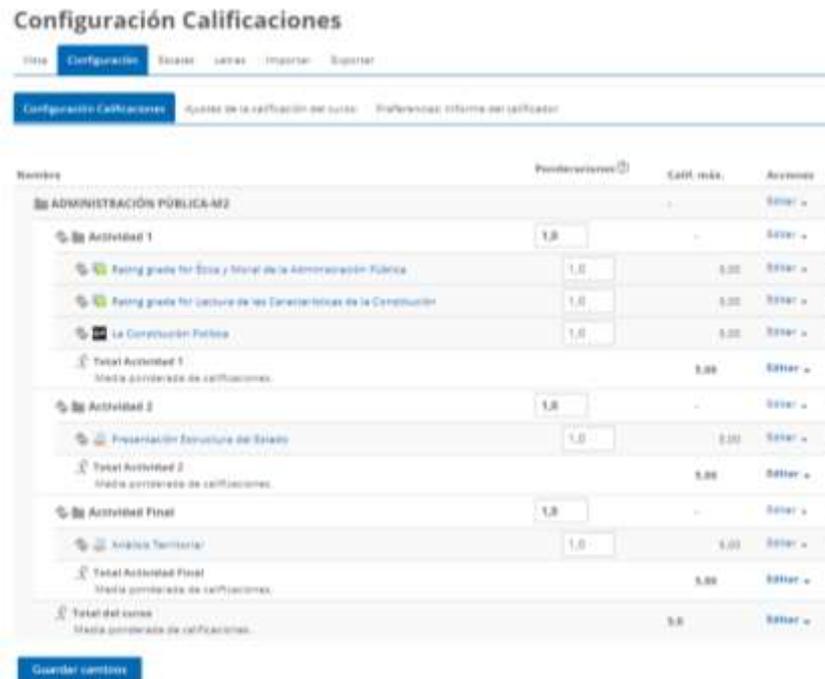


Figura 8. Configuración libro de calificaciones UACV- Curso Administración Financiera Fuente. UACV

3.9.7 Problemas encontrados sprint 1

- No existe diagrama de entidad relación del SIA, con la colaboración de los administradores de bases de datos se obtienen las tablas y los campos que se utilizaran para realizar las consultas.
- Moodle está en la capacidad de entregar al SIA las calificaciones, con la implementación de la metodología modular se establece como reporte al SIA solo la nota definitiva obtenida por el estudiante en plataforma UACV.
- Para que el sistema UACV reporte la nota definitiva al SIA el profesor del curso debe configurar correctamente el libro de calificaciones creando las categorías de calificación y asignando ponderaciones de cada una de las actividades y categorías en concordancia con la consigna elaborada para cada curso, en caso de una mala configuración o ausencia de esta, el reporte de nota del estudiante será errado y el docente deberá registrar las notas manualmente en el SIA.

3.10 Sprint 2

3.10.1 Modelado de datos para el cliente UACV del servicio web.

Como formato de archivo se decide utilizar JSON (son las siglas de JavaScript Object Notation) o su traducción como Notación de Objeto JavaScript, se decide utilizar este formato por su facilidad de uso ya que se encuentra muy bien soportado de forma nativa por los lenguajes de programación como PHP, JavaScript lenguajes en los cuales ha sido desarrollado moodle, es popular en otros sistemas de información lo cual deja un marco de referencia para futuros proyectos de intercambio de información, es legible por humanos, requiere menos detalles de etiquetado, el tamaño de los archivos a transmitir es reducido.

3.10.2 Diseño de arquitectura del servicio web.

La arquitectura por implementar para los servicios web se basa en el modelo REST, transferencia de estado representacional, el cual establece solicitudes REQUEST y respuestas RESPONSE, utilizando un protocolo de comunicación como http o https, cuenta con operaciones POST, GET, PUT y DELETE estas operaciones tienen similitud al CRUD (create, read, update, delete) de las bases de datos en español crear, leer, actualizar, eliminar. Para esta implementación se utilizará https el cual permite el encriptado de la información ente el cliente y el servidor utilizando certificado SSL incrementando la seguridad de la información.



Figura 9. Diseño servicio REST-GET
Fuente. Elaboración propia.

3.10.3 Diseño de la estructura de datos del servicio web.

Según el resultado del análisis de datos que debe servir el SIA a UACV se propone la siguiente estructura para el proceso de creación de cursos en UACV.

Tabla 5. Modelo JSON F-UACV-MC

```
{
  "fullname": "ACTIVIDAD FISICA PARA LA PROMOCION DE LA",
  "shortname": "24090533-2P20-P001",
  "faculty": 58,
  "group1": "DA-24090533-2P20-M1",
  "startdate": "30/01/2017",
  "enddate": "26/05/2017",
  "idnumber": "24090533-2P20-P001"
}
```

3.10.3.1 Análisis de la naturaleza de los campos de datos para estimar el tamaño de cada objeto JSON retornado por la consulta.

Tabla 6. Tamaño estructura de datos JSON F-UACV-MC

CAMPO_REQUERIDO	TIPO DE DATO	Tamaño en bytes
fullname	nvarchar 40	80 bytes
shortname	nvarchar 15 nvarchar 10 varchar 6	62 bytes
faculty	varchar 2	4 bytes
group1	char 1 char 1 nvarchar 4 varchar 8	16 bytes
startdate	Datetime 8 bytes	8 bytes
enddate	Datetime 8 bytes	8 bytes
idnumber	Varchar 100 bytes	100 bytes
Total, tamaño estimado por objeto		278 bytes.

Fuente. Elaboración propia.

3.10.4 Diseño arquitectura del backend para el SIA del servicio web.

3.10.4.1 Diseño de seguridad para el servicio web.

La seguridad del servicio web se debe establecer en dos niveles, basándonos en el modelo TCP/IP se opta por asegurar la capa de transporte y la capa de aplicación, para ello todas las peticiones entre SIA y UACV contarán con una clave de encriptación, que a su vez servirá como token de sesión, la capa de transporte será asegurada con la implementación del certificado SSL y el estricto uso de https para publicación de los servicios.

3.10.5 Diseño arquitectura del servicio web.

El servicio web estará disponible en <https://wsuacv.uniautonomo.edu.co> y solo atenderá peticiones desde el dominio moodle.uniautonomo.edu.co, el servidor web dispuesto para este fin será NGINX y el lenguaje de programación PHP.

Tabla 7. Estructura JSON F-UACV-ME

<pre>{ "username": "miguel.zapata.c@uniautonomo.edu.co", "email": "miguel.zapata.c@uniautonomo.edu.co", "firstname": "MIGUEL ANGEL", "lastname": "ZAPATA CASTRO", "idnumber": "P000009537", "profile_field_faculty": "FACED", "profile_field_pregrado": "DEPORTE Y ACTIVIDAD FISICA", "auth": "oauth2", "course1": "28170328-2P20-M3",</pre>
--

```
"role1": "student",
"group1": "DA-28170328-2P20"
}
```

Fuente. Elaboración propia basada en documentación moodle

3.10.6 Problemas encontrados sprint 2

El servidor moodle no cuenta con certificado SSL por lo cual se realizó la reconfiguración del servidor web NGINX para implementar el certificado SSL

3.11 Sprint 3

3.11.1 Instalación y configuración de la arquitectura del servicio web.

- Sistema operativo Linux: se elige este sistema operativo por su alto rendimiento como servidor estando presente en el 100% las supercomputadoras del mundo según el reporte publicado por el proyecto TOP500 (TOP500, 2021).

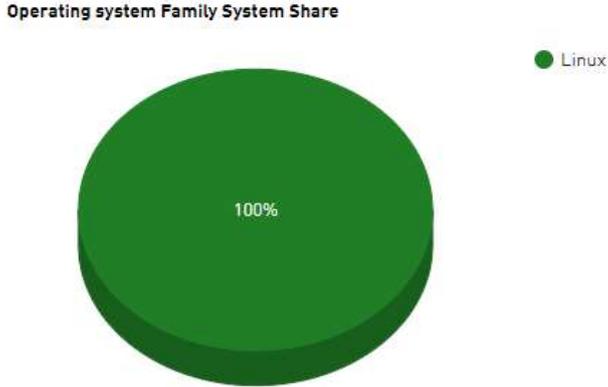


Figura 10. Familia del sistema operativo Fuente. (top500.org, 2021)

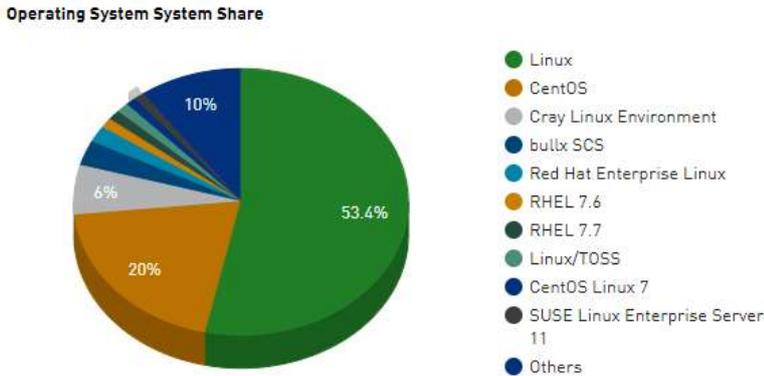


Figura 11. Sistema operativo reportado top500.org Fuente: (top500.org, 2021)

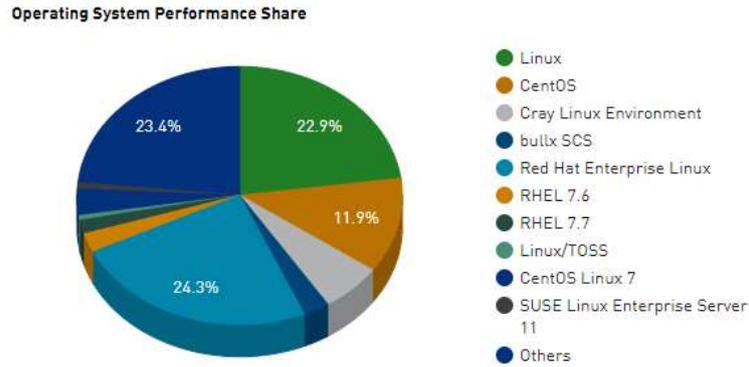


Figura 12. Rendimiento sistema operativo reportado top500.org
Fuente. (top500.org, 2021)

- La distribución seleccionada es Community Enterprise Operative System más conocida como CentOS, en su versión 7, por ser orientada a servidores, contar con licencia GPL, bajo consumo de recursos hardware, gran estabilidad y seguridad, cuenta con soporte a largo termino o por sus siglas en ingles Long Term Support lo cual garantiza actualizaciones de seguridad hasta el 30 de junio de 2024.

Tabla 16. Requisitos mínimos del sistema operativo Linux CentOS 7

Procesador	x86_64, POWER, System z
Memoria	1Gb por cada CPU lógica
Espacio en disco	10GB mínimo, Recomendado 20GB

Fuente elaboración propia (redhat, 2021)

La instalación del sistema operativo se realiza en una máquina virtual sobre arquitectura de virtualización Microsoft Hyper-v, según las prácticas de virtualización de servidores de la división de tecnologías de la información y medios educativos.



Figura 13. Instalación Máquina Virtual Servidor Web Linux CentOS 7
Fuente. Elaboración propia.

3.11.2 Instalación de php 7.3.23

La instalación de php se realiza utilizando el manejador de paquetes de linux CentOS:
`dnf install php7.3-common php7.3-cli php7.3-bcmath php7.3-bz2 php7.3-curl php7.3-gd php7.3-intl php7.3-json php7.3-mbstring php7.3-readline php7.3-xml php7.3-xmlrpc php7.3-soap`

```

Instalado:
php-7.3.23-1.el7.x86_64          php-gd-7.3.23-1.el7.x86_64    php-intl-7.3.23-1.el7.x86_64
php-soap-7.3.23-1.el7.x86_64  php-xml-7.3.23-1.el7.x86_64  php-xmlrpc-7.3.23-1.el7.x86_64
php-pdo-7.3.23-1.el7.x86_64  gd-2.3.0-2.el7.x86_64        libcups-2.2.5-1.el7.x86_64
apr-1.4.8-5.el7.x86_64        apr-util-1.5.2-6.el7.x86_64  freibidi-1.0.2-1.el7.x86_64
httpd-tools-2.4.6-93.el7.centos.x86_64  jbigkit-libs-2.0-11.el7.x86_64  libciff-4.0.3-32.el7.x86_64

php-mbstring-7.3.23-1.el7.x86_64  php-mysqlnd-7.3.23-1.el7.x86_64  php-pecl-zip-1.19.1-1.el7.x86_64
php-cli-7.3.23-1.el7.x86_64        php-common-7.3.23-1.el7.x86_64  php-json-7.3.23-1.el7.x86_64
libwebp-1.0.3-1.el7.x86_64         libzip-1.7.3-1.el7.x86_64        oniguruma6php-6.9.5+rev1-4.el7.x86_64
graphite2-1.3.10-1.el7.x86_64      harfbuzz-1.7.5-2.el7.x86_64      httpd-2.4.6-93.el7.centos.x86_64
mailcap-2.1.41-2.el7.noarch        libargon2-20161029-3.el7.x86_64  librpm-0.7.0-4.el7.x86_64

;Listo!
[root@w ~]#

```

Figura 14 Instalación de php 7.3.23
Fuente. Elaboración propia.

- Servicios requeridos: servidor web nginx o apache interpretando php >= 7.0 con extensión sqldr.
- Registro DNS tipo A para resolver wsuacv.uniautonomo.edu.co.
- Credenciales de acceso a la base de datos del SIA con permisos de consulta.

3.11.3 Programación del JSON

Para realizar las pruebas se emplea la herramienta POSTMAN la cual permite realizar las peticiones REQUEST de tipo GET o POST y visualizar la respuesta RESPONSE del servicio en este caso en formato JSON.

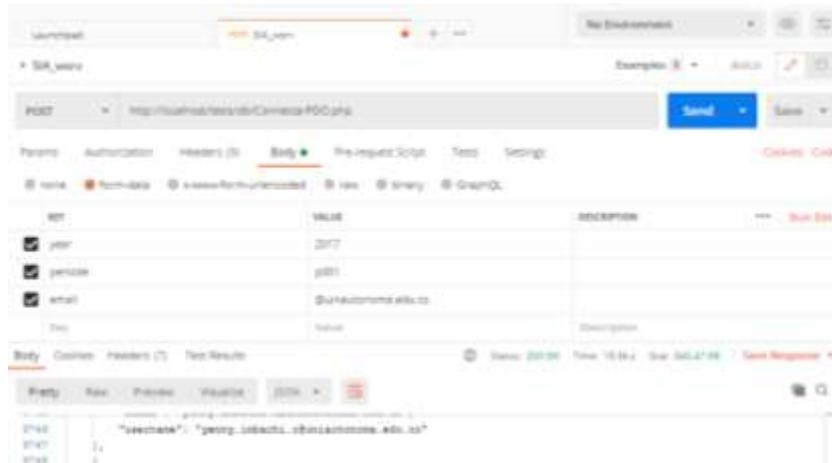


Figura 15. Herramienta POSTMAN REQUEST
Fuente elaboración propia

```

9759 | {
9760 |   "firstname": "YUDY COLOMBIA",
9761 |   "secondname": "DE LA CRUZ ACOSTA",
9762 |   "shortname": "161701072P20-M1",
9763 |   "fullname": "CATEDRA AUTONOMA",
9764 |   "Grupo": "B",
9765 |   "Jornada": "DIURNO",
9766 |   "Programa": "INGENIERIA AMBIENTAL Y SANITARIA",
9767 |   "email": "yudy.delacruz.a@uniiautonoma.edu.co",
9768 |   "username": "yudy.delacruz.a@uniiautonoma.edu.co"
9769 | },
9770 | {
9771 |   "firstname": "YUDY COLOMBIA",
9772 |   "secondname": "DE LA CRUZ ACOSTA",
9773 |   "shortname": "261202132P20-M1",
9774 |   "fullname": "AYUDAS Y MEDIOS EDUCATIVOS",
9775 |   "Grupo": "A",
9776 |   "Jornada": "DIURNO",
9777 |   "Programa": "LICENCIATURA EN EDUCACION PARA LA PRIMER",
9778 |   "email": "yudy.delacruz.a@uniiautonoma.edu.co",
9779 |   "username": "yudy.delacruz.a@uniiautonoma.edu.co"
9780 | },
9781 | {
9782 |   "firstname": "YUDY COLOMBIA",
9783 |   "secondname": "DE LA CRUZ ACOSTA",
9784 |   "shortname": "261203212P20-M1",

```

Figura 16. JSON RESPONSE POSTMAN
Fuente elaboración propia

3.12 Configuración de servicios web en UACV

El procedimiento para habilitar los servicios web en el sistema UACV consta de 10 pasos, según la documentación oficial de moodle, esta configuración debe ser realizada por el administrador de UACV en plataforma moodle.

Tabla 17. Pasos activación de servicios web UACV

ACTIVAR SERVICIOS WEB MOODLE UACV
1. Habilitar Servicios Web: Sí.
2. Habilitar los protocolos: REST. Al menos un protocolo debe estar habilitado. Por razones de seguridad, los protocolos que se van a utilizar deben estar habilitados.
3. Seleccione un servicio: Un servicio es un conjunto de funciones de 'servicios web'. Permitirá acceder al usuario a un nuevo servicio. En la página "Añadir servicio" marque la opción 'Habilitado' y desmarque 'Usuario autorizado'. Seleccione 'No se requiere permiso'.

4. Agregar funciones: Seleccione las funciones requeridas para el servicio de nueva creación.
5. Comprobar permisos de los usuarios: Los usuarios deben tener dos capacidades (permisos) - webservice: createtoken y una capacidad que concuerde con los protocolos utilizados, por ejemplo, webservice/rest:use, webservice/soap:use. Para lograr esto, cree un rol de servicios web con las capacidades apropiadas habilitadas y asígnelo a usuario de servicios web como un rol del sistema.
6. Comprobar el servicio: Simular el acceso externo al servicio empleando el servicio web de prueba del cliente. Antes de hacerlo, acceda como usuario con el permiso moodle/webservice:createtoken y obtenga la clave de seguridad (token) vía las configuraciones de Mi perfil. Usted empleará esta ficha (token) en el cliente de prueba. En el cliente de prueba, seleccione también un protocolo habilitado con autenticación por ficha (token).
ADVERTENCIA: ¡Las funciones que usted pruebe SERÁN EJECUTADAS para este usuario, por lo que debe tener cuidado con lo que decida probar!

Fuente. Elaboración propia basada en documentación moodle

3.13 Activar Servicios Web en UACV

Para activar los servicios web se debe autenticar con el rol de administrador luego debe ir al bloque de administración del sitio → Extensiones → Servicios Web

Paso 1: Se habilitan los Servicios Web.



Figura 17. Habilitar Servicios Web
Fuente. Elaboración propia UACV

Paso 2: Se habilita el protocolo en este caso REST



Figura 18. Habilitar protocolo REST en UACV
Fuente. Elaboración propia UACV

Paso 3: Crear el usuario webs_user_uacv

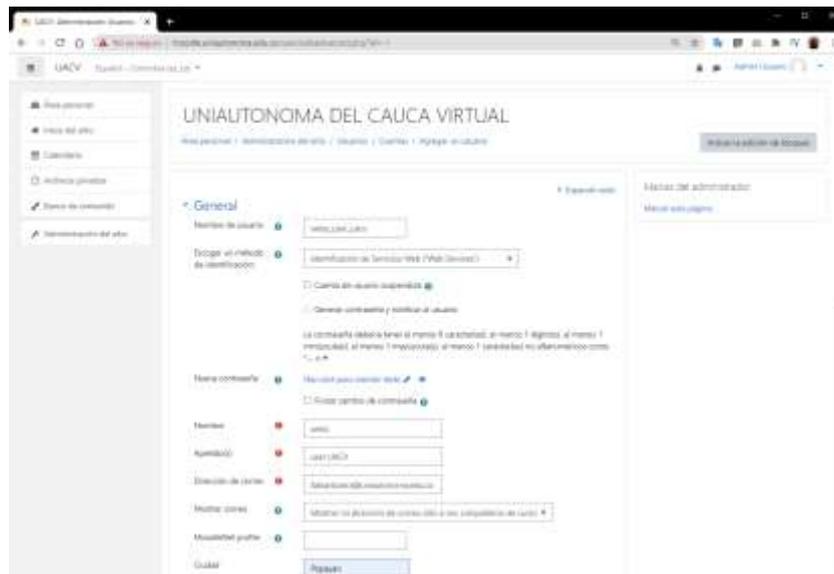


Figura 19. Configuración usuario servicio web desde UACV Fuente. Elaboración propia UACV

Paso 4: Se Comprueban los permisos del usuario en el contexto del sistema.

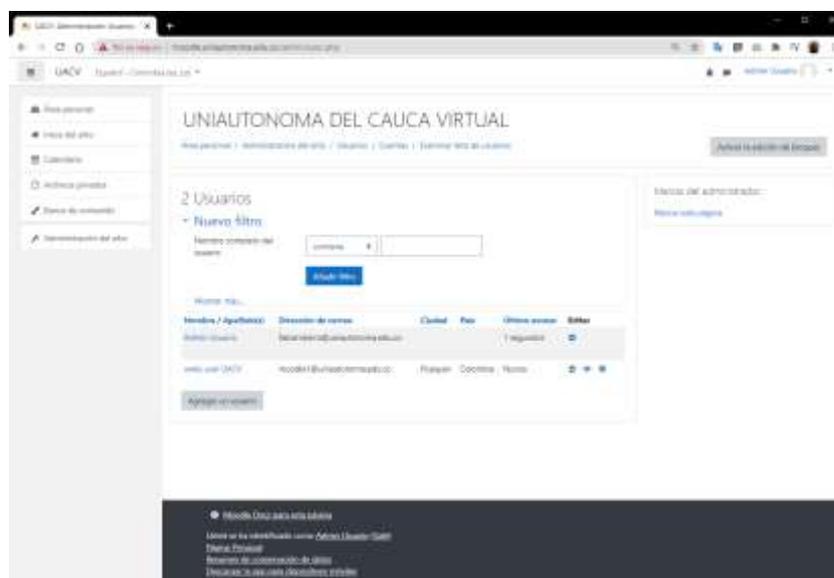
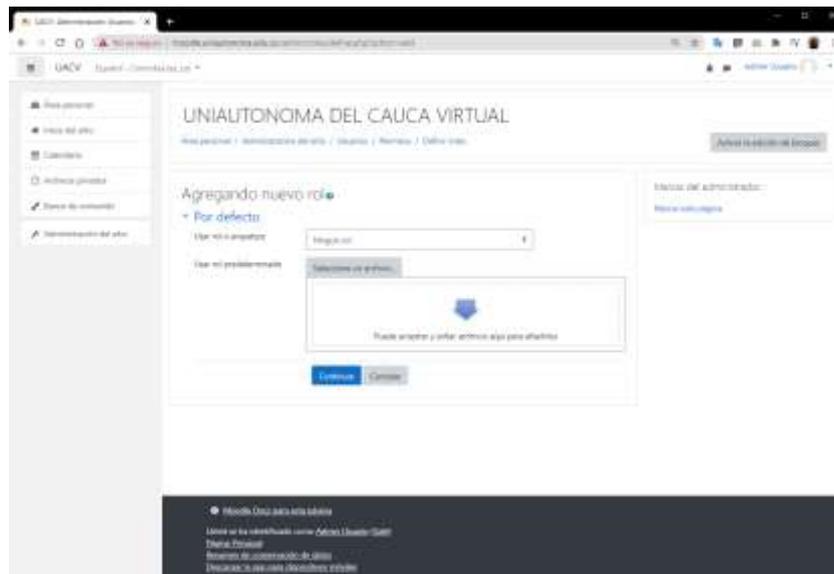


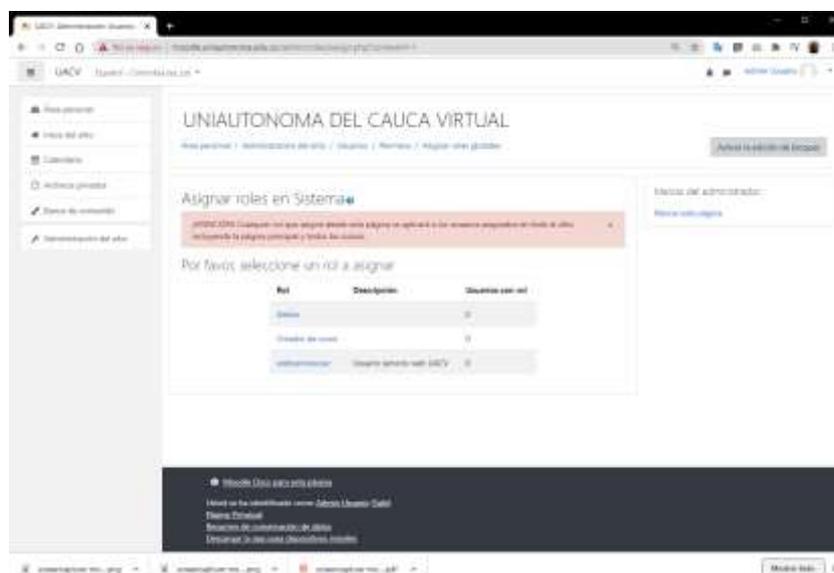
Figura 20. Lista de usuario administrador y servicio web UACV Fuente. Elaboración propia UACV

Paso 5: Se Crea un nuevo rol para el usuario del servicio web.



Creación nuevo ROL
Fuente. Elaboración propia UACV

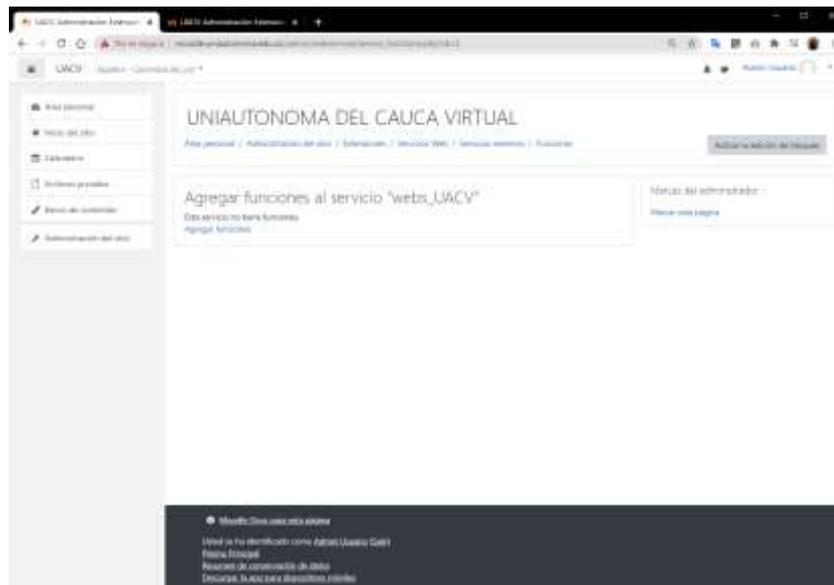
Se Selecciona el contexto de permisos para el usuario en este caso a nivel de sistema.



Configuración de ROL a nivel de sistema
Fuente. Elaboración propia UACV

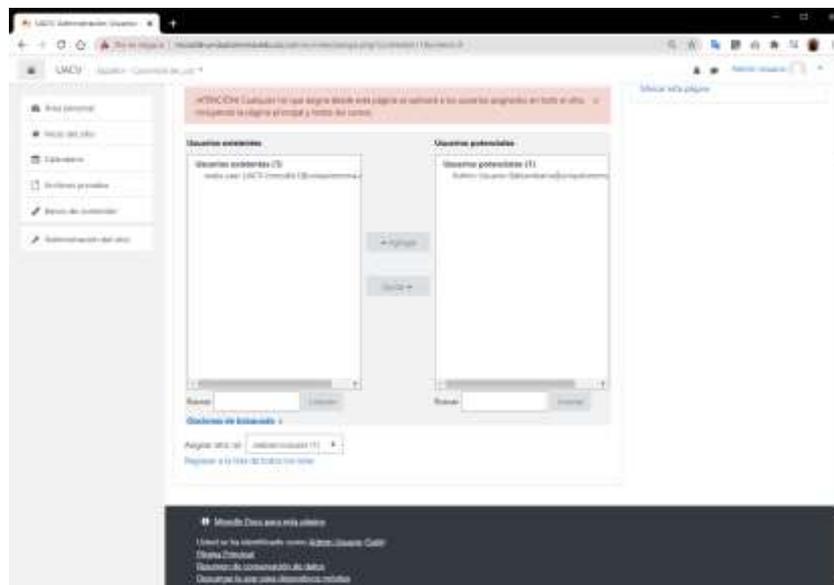
Paso 6: Se asignan las funciones del sistema a las cuales el usuario webs_user_uacv puede instanciar, para ese caso:

- core_course_get_contents
- core_enrol_get_enrolled_users
- core_grades_update_grades
- enrol_manual_enrol_users
- enrol_manual_unenrol_users



Asignación de funciones según el ROL en el sistema
Fuente. Elaboración propia UACV

Paso 7: En la siguiente imagen se evidencia el usuario `webs_user_uacv` en como usuario de sistema.



Asignación de rol en el contexto de sistema
Fuente. Elaboración propia UACV

Paso 8: Creamos el token que será compartido entre los dos servidores como un medio de autenticación.

Paso 9: Activamos la documentación de los servicios web que incluye moodle directamente en plataforma.



Figura 21. Activar la documentación de servicios web en UACV
Fuente. Elaboración propia UACV

Paso 10: Se realizan las pruebas (Ver Capítulo V).

3.14 APIS

El api fue creado con el fin de poder servir los datos entre el SIA y UACV, de esta forma permite establecer los datos a obtener, para realizar la configuración de UACV.

Tabla 18. APIS

search_enrolled_users	Obtiene la colección de objetos usuarios matriculados en el contexto del curso.
validtokens	Comprueba si la llave de acceso es válida.
get_courses	Obtiene la colección de objetos con los cursos ofertados
get_docentes	Obtiene la colección de objetos con los docentes y sus cursos.

Fuente. Elaboración propia

CAPÍTULO V. RESULTADOS

En este Capítulo se muestran los resultados del trabajo, en la siguiente figura se representa la Arquitectura y contexto el servidor UACV en las instalaciones de la Corporación Universitaria Autónoma del Cauca.

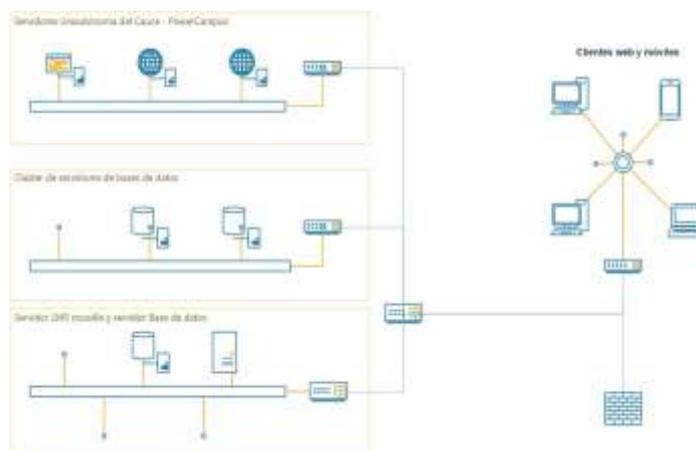


Figura 22. Arquitectura de servicios Uniautónoma del Cauca
Fuente. Elaboración propia UACV

Durante el desarrollo de esa práctica se origina en china la infección por virus COVID 19, infección que derivó rápidamente en pandemia mundial alcanzando a Colombia en tan solo tres meses desde su aparición en China. Con la llegada del virus el gobierno nacional ordena cuarentena absoluta y con ello las empresas se ven obligadas a implementar estrategias de teletrabajo y las instituciones educativas a ofertar sus programas académicos en una modalidad denominada presencialidad mediada por TIC, para dar alcance a esta nueva forma de estudio se diseñó la migración del servidor UACV a la nube para poder soportar la concurrencia de usuarios y la expansión de datos, ya que en las instalaciones propias de la institución no se contaba con recursos suficientes que permitieran atender la demanda del servicio.

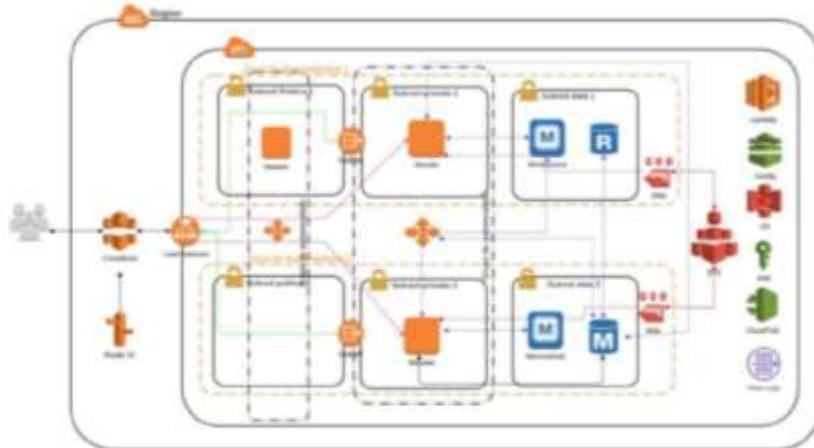


Figura 23. Arquitectura AWS para UACV
Fuente elaboración propia

La implementación del servicio se contrata con una empresa local luego de múltiples negociaciones con empresas extranjeras propuestas como socios estratégicos para el despliegue del servicio, el acuerdo con el aliado estratégico local implica que ellos implementaran la infraestructura en un plazo no mayor a 6 días y pasante deberá implementar la plataforma y migrar la información en un plazo de 3 días.

En la siguiente figura se evidencia un fragmento de la oferta económica del primer aliado estratégico, se contempla la propuesta inicial por 12 meses de servicio la cual genera un costo total, 31884 dólares equivalen a 120,192,158.28 Pesos Colombianos según la tasa de cambio del dólar para el mes de agosto de 2020.

4.2 Oferta económica

Recurrente mensual del Servicio	12 Meses	24 Meses	36 Meses
Total recurrente (USD)	USD 2.657	USD 2.400	USD 2.250

Forma de Pago: Recurrente Mensual según termino de contratación.
Moneda: Oferta en Dólares Americanos (USD), se factura en pesos colombianos (COP) según la TRM vigente al momento de la facturación.

4.3 Condiciones comerciales específicas

- Tarifas no incluyen impuestos, estampillas ni gastos de legalización
- Moneda: Para el servicio de Moodle la Oferta va en Dólares Americanos (USD), se factura en pesos colombianos (COP) según la TRM vigente al momento de la facturación. Los servicios opcionales se ofertan y se facturan en pesos colombianos.
- Tarifas no incluyen ningún tipo de adecuación en las salas, ni equipos de proyección (Video Beams, Televisores, entre otros). Sólo se incluyen los equipos expresamente descritos en la oferta
- Tarifas no incluyen ningún tipo de conectividad, salvo que así se mencione de manera expresa y clara.
- Si el cliente se retira antes de la vigencia contractual deberá cancelar el valor total de las mensualidades restantes.

Figura 24. Oferta económica aliado estratégico 1

Fuente elaboración proveedor de servicios de telecomunicaciones.

La oferta económica del segundo aliado por 12 meses de servicio genera un costo total, 25,500,000 además incluye 100 horas de soporte al año.

4.2. Servicios

DESCRIPCIÓN	VALOR
Servicios de Configuración de Arquitectura en AWS según alcance	COPS 12.600.000
Bolsa de horas de soporte (100 horas)	COPS 12.500.000

- Valores antes de IVA

5. Facturación y forma de pago

La facturación y pagos se realizarán de la siguiente manera:

- La facturación de la3 se realiza mes vencido para pago inmediato
- La facturación de los servicios de "Configuración de Arquitectura en AWS según alcance" se realizará 50% al inicio del proyecto, 30% a la entrega de la infraestructura a pruebas y 20% a la puesta en producción; para pago a 30 días
- La facturación de la bolsa de horas es anticipada por el 100% del valor para pago a 30 días.

6. Validez de la oferta

La presente propuesta tiene validez de treinta (30) días a partir de la fecha de entrega de la propuesta 14/07/2020. En caso de no ser aprobada durante este lapso y UNIAUTONOMA siga interesada en la ejecución de esta, deberá ser revisada y presentada nuevamente con las modificaciones pertinentes.

Figura 25. Oferta económica aliado estratégico 2
Fuente elaboración proveedor de servicios de consultoría AWS.

Gracias a la experiencia en instalación, administración de moodle y a la experiencia en el despliegue de servicios con arquitectura tipo Linux se recomienda la propuesta económica del segundo aliado.

Acceso principal a moodle.

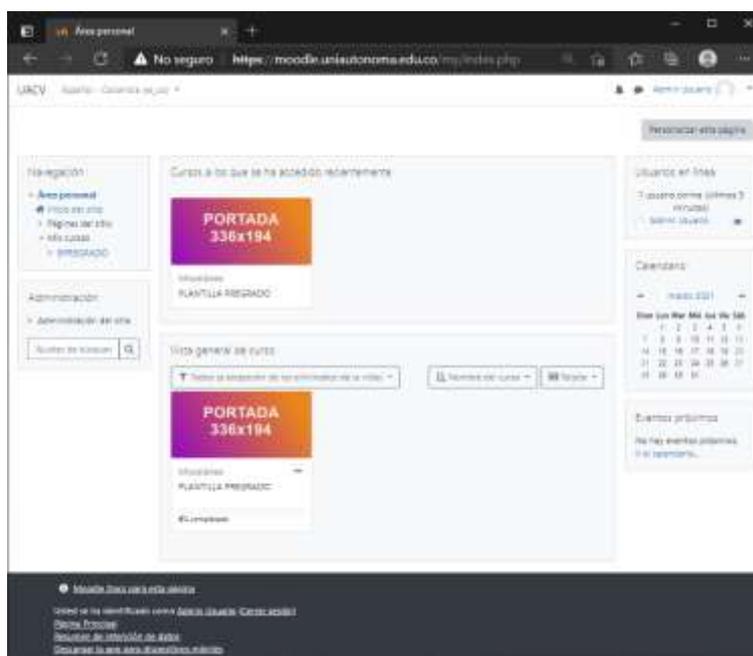


Figura 26. Pagina principal de UACV moodle
Fuente. Elaboración propia UACV

Instalación del plugin: la carpeta wsuacv con los archivos fuente debe ser cargada en el subdirectorio *enrol* el cual está en la raíz de UACV.

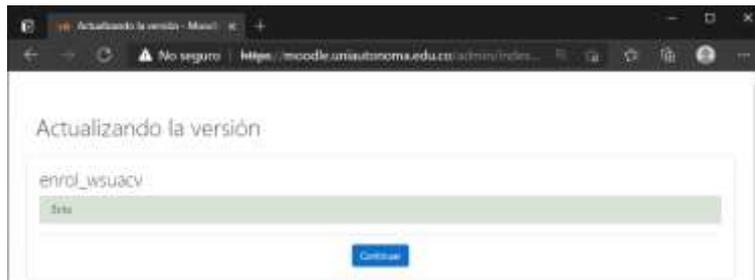


Figura 27. Instalando Plugin wsuacv
Fuente. Elaboración propia UACV

Activando el plugin: desde la página de administración de extensiones en la categoría de matriculaciones.

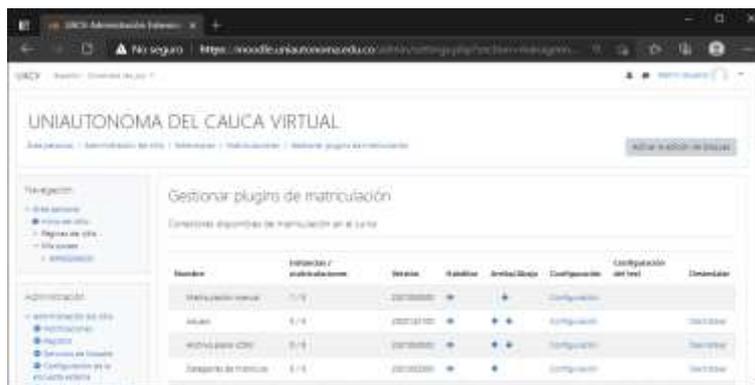


Figura 28. Activando plugin wsuacv
Fuente. Elaboración propia UACV

Configurando el plugin: el primer parámetro a configurar es identificador de recurso uniforme URI, para este caso es <https://wsuacv.uniautonomo.edu.co> dirección del servicio web que realiza la consulta al SIA, el segundo parámetro, establecer el método de comunicación, por seguridad se debe utilizar POST, para efectos de verificación y pruebas se puede seleccionar GET, ya que este permite observar los parámetros transmitidos al servicio web desde la URL en el navegador, como tercer parámetro se crea el token o cadena de seguridad la cual garantiza que solo los usuarios autorizados puedan consumir el servicio web, el cuarto parámetro es el rol por defecto que tendrán los usuarios a matricular en el contexto del curso, está seleccionado Estudiante, pero permite seleccionar cualquier rol disponible en el sitio, el quinto parámetro es una opción que permite agregar de forma automática el método de matriculación a todos los cursos nuevos.

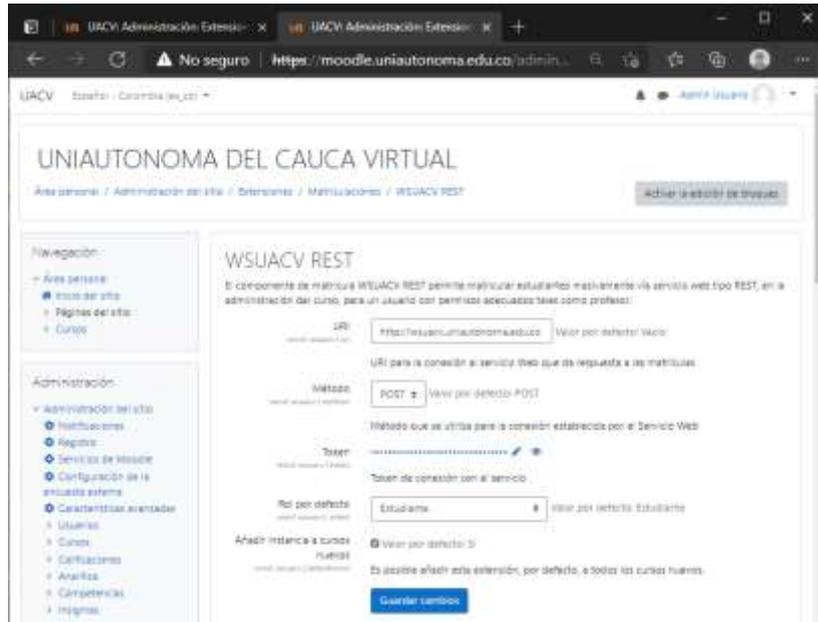


Figura 29. configuración WSUACV REST
Fuente. Elaboración propia UACV

Prueba del plugin desde el curso PLANTILLA PREGRADO

En el bloque de administración del curso, métodos de matriculación y seleccionamos el icono agregar usuarios.



Figura 30. Método de matriculación WSUAC REST
Fuente. Elaboración propia UACV

Matricula masiva desde SIA.

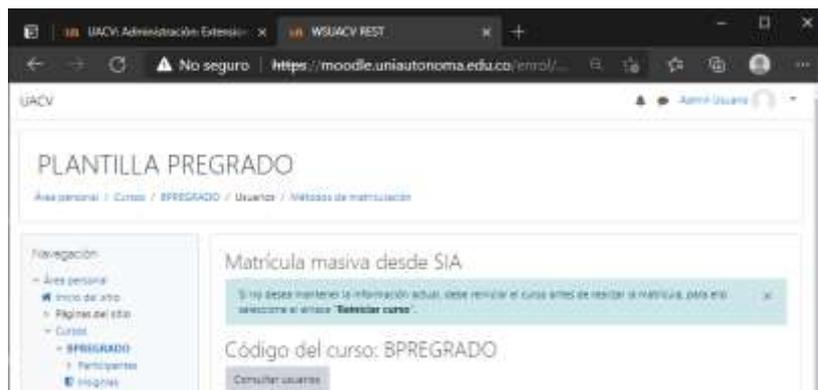


Figura 31. Matricula masiva desde SIA Fuente.

Listado de los usuarios por matricular y los nuevos usuarios, muy útil para el manejo de novedades de matrícula, el administrador del curso o docente puede actualizar la matrícula en cualquier momento.

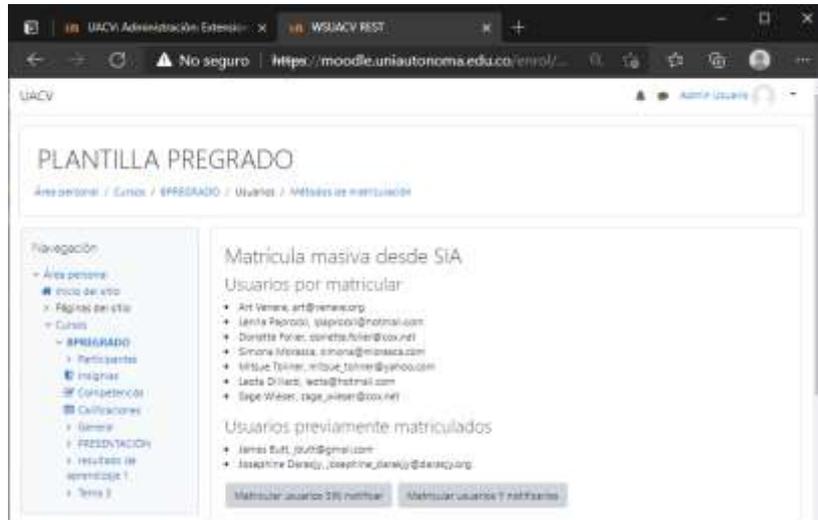


Figura 32. listado de matrícula WSUACV REST
Fuente elaboración propia UACV

Finalmente, en la siguiente figura se representa la Confirmación del proceso de matrícula y creación de usuarios nuevos.



Figura 33. confirmación de matrícula WSUACV REST
Fuente elaboración propia UACV

CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES

Esta práctica sirve como insumo de gran utilidad para el proceso de matrículas en sistema externo en caso basado en plataforma LMS moodle para la Corporación Universitaria Autónoma del Cauca, además de haber sido un gran aporte en la enseñanza del año 2020 que trabajo con la modalidad asistida por TIC.

Al contar con un sistema de recursos AVA en una Institución de Educación Superior fortalece sus actividades académicas dentro de entornos virtuales de aprendizaje

Gracias a la experiencia en instalación, administración de moodle y a la experiencia en el despliegue de servicios con arquitectura tipo Linux se recomienda la propuesta económica del segundo aliado estratégico ahorrando costos de servicio y administración por más de cien millones de pesos, procede a la ejecución de este proyecto.

El uso de Moodle como LSM presento diferentes ventajas en el desarrollo del proyecto, gracias a sus registros permite realizar un seguimiento histórico, de las interacciones de sus usuarios: docentes, estudiantes, administradores, profesional de apoyo a la permanencia, personal de TI, además por ser una plataforma software libre Open Source permite la revisión del código fuente y es extensible para adaptarse a cualquier institución o proceso.

Es importante estandarizar los planes académicos, la mayor dificultad encontrada durante la ejecución de esta practica radicó en la inexistencia de cursos equivalentes o integraciones en el SIA, los cambios de metodología obligaron a cambiar el comportamiento del servicio, porque durante todo el periodo académico se tienen habilitadas las novedades a la matricula.

Se recomienda a la Corporación Universitaria Autónoma del Cauca seguir en el fortalecimiento de herramientas educativas basadas en Plataformas LMS.

REFERENCIAS

- Bircher, C. (2017). *Arquitectura Cliente-Servidor/Cliente-Servidor en Aplicaciones de Misión Crítica*.
- Bolo, M. (2006). Arquitectura de integración orientada a servicios. *Revista Interfases*, 19-46.
- ellucian. (2021). From <https://www.ellucian.com/assets/es/folleto/powercampus.pdf>
- Google. (2021). *Meet the new Hangouts*. From <https://www.blog.google/products/g-suite/meet-the-new-enterprise-focused-hangouts/>
- json. (2020). *json.org*. From <https://www.json.org/json-es.html>
- MOODLE. (2020). From https://docs.moodle.org/all/es/Caracter%C3%ADsticas_de_Moodle_3.10
- MOODLE. (2021). From <https://stats.moodle.org/sites/>
- Morales, C. A. (2010). *Estado del Arte: Servicios Web*. Universidad Nacional de Colombia.
- PHP. (2021). From <https://www.php.net>
- redhat. (2021). From <https://access.redhat.com/articles/rhel-limits>
- Schwaber, K., & Sutherland, J. (2016). *La Guía de Scrum*. From <https://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf>
- TOP500. (2021). From <https://www.top500.org/project/>
- top500.org. (2021). From <https://www.top500.org/statistics/list/>
- vim. (2021). From <https://www.vim.org/>
- W3. (2020). From <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>
- Zhang, M., Jiang, L., Yue, P., & Gong, J. (2020). Interoperable web sharing of environmental models using OGC web processing service and Open Modeling Interface (OpenMI).