



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO/A EN SISTEMAS INFORMÁTICOS

TEMA:

INTEGRACIÓN SIGE – EVA(Moodle)

AUTOR/ A:

RUBÉN EFRAÍN VILLALBA ESPINOSA
JOSÉ LUIS FABRE CAÑIZARES
DIEGO ANÍBAL ATIAJA JIMÉNEZ

TUTOR/ A:

MG. RENATO TOASA

QUITO- ECUADOR

AÑO: 2021

APROBACIÓN DEL TUTOR



Yo, Mg. Renato Mauricio Toasa Guachi con C.I: 1804724167 en mi calidad de Tutor del proyecto de investigación titulado: Integración SIGE – Moodle

Elaborado por: Rubén Villalba, José Fabre, Diego Atiaja (Grupo Silver), de C.I: 1714623673, 1721831962, 1719153106, estudiantes de la Carrera: Ingeniero En Sistemas Informáticos, de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Ingeniero, me permito declarar que luego de haber orientado, analizado y revisado el proyecto de titulación, lo apruebo en todas sus partes.

Quito D.M., 5 de marzo de 2021



Firma

DEDICATORIA

Este trabajo está dedicado de todo corazón a nuestro Dios, familiares, amigos y profesores, que han sido motivo de inspiración para no desmayar en el esfuerzo para culminar nuestra carrera, que nos han permitido alcanzar y superar los conocimientos que la especialidad requiere.

“Porque tú eres mi roca y mi fortaleza,
Por amor de tu nombre me guiarás
Y me encaminarás”
Salmos 31:3

AGRADECIMIENTO

Gracias:

A la Universidad Israel por habernos abierto sus puertas para que podamos ser parte de esta hermosa institución y sus diferentes autoridades y docentes que compartieron sus conocimientos y el apoyo durante todo el tiempo de nuestros estudios.

Agradecemos a Msc. Renato Toasa tutor de nuestro proyecto como a Msc. Paúl Baldeón por habernos permitido formar parte del Grupo Silver y poder sumarnos como integrantes para dar un granito de nuestro conocimiento, y el habernos guiado con el conocimiento científico para la culminación del proceso de investigación.

Tabla de contenidos

APROBACIÓN DEL TUTOR	II
RESUMEN	X
ABSTRACT	XI
INFORMACIÓN GENERAL	XII
Contextualización del tema	XII
Pregunta Problémica	XII
Objetivo general	XII
Objetivos específicos	XII
Beneficiarios directos:	XII
CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO	1
1.1. Contextualización de fundamentos teóricos	1
1.2. Problema a resolver	5
1.3. Proceso de investigación	5
1.4. Vinculación con la sociedad	8
1.5. Indicadores de resultados	8
CAPÍTULO II: PROPUESTA	9
2.1. Fundamentos teóricos aplicados	9
2.2. Descripción de la propuesta	10
2.3. Explicación del aporte	14
2.3.1. Funcionamiento del proceso	15
2.3.2. Estrategias y/o técnicas	16
2.4. Matriz de articulación	25
CAPITULO III: IMPLEMENTACIÓN	27
3.1 Plan de implementación	27
3.2 Requerimientos	28
3.2.1. Requerimientos de software	28
3.2.2. Requerimientos de Hardware	28
3.3 Manual Técnico	29
2.3.3. Manual de instalación	29
CONCLUSIONES	44
RECOMENDACIONES	45
BIBLIOGRAFÍA	46

Índice de tablas

Tabla 1. Cronograma de Capacitaciones SIGE-EVA(Moodle).....	17
Tabla 2. Casos de prueba	21
Tabla 3. Resultados cado de Prueba 01	22
Tabla 4.. Resultados cado de Prueba 02	22
Tabla 5. Resultados cado de Prueba 03	23
Tabla 6.. Resultados cado de Prueba 04	23
Tabla 7.. Resultados cado de Prueba 05	24
Tabla 8.. Resultados cado de Prueba 06	24
Tabla 9. Cronograma de Implementación.....	27

Índice de figuras

Figura 1. Entorno Virtual de Aprendizaje “EVA” de la Universidad Israel, donde los alumnos ingresan a las materias del ciclo correspondiente de matriculación.....	2
Figura 2. Moodle, plataforma utilizada para aprendizaje educativo.....	3
Figura 3. Sistemas de Información.....	4
Figura 4. Sistemas de Información para la Gestión (MIS).....	4
Figura 5. Ciclo de metodología ágil vs la metodología tradicional.....	7
Figura 6. Integración SIGE-EVA (Moodle) Diagrama de Secuencia donde se muestra los diferentes pasos para la interacción.....	11
Figura 7. Representación grafica Proceso general de Integracion SIGE-EVA(Moodle), diagrama macro de los procesos más representativos de la implementación.....	11
Figura 8. Representación gráfica Proceso Registro de Matriculación de Integración SIGE-EVA (Moodle).....	12
Figura 9. Representación gráfica Proceso Cambio de Paralelo de Matriculación de Integración SIGE-EVA (Moodle).....	13
Figura 10. Representación gráfica Proceso Dar de Basa materia de Integración SIGE-EVA (Moodle).....	13
Figura 11. Ciclo de Integración SIGE-Moodle.....	14
Figura 12. Casos de Uso del estudiante en el proceso de matriculación y la interacción con las plataformas del SIGE-EVA(Moodle).....	15
Figura 13. Diagrama de componentes Integración SIGE-EVA(Moodle).....	15
Figura 14. Diagrama de interpretación del Linked Server.....	19
Figura 15. Scripts de validación Integración matriculación.....	20
Figura 16. Cronograma de Implementación.....	28
Figura 17. Versión de archivo de odbc a descargarse del sitio oficial de PostgreSQL.....	30
Figura 18. Ruta donde se almaceno la descarga del odbc del sitio oficial de PostgreSQL... ..	30
Figura 19. Pantalla de bienvenida del instalador de ODBC para Windows.....	31
Figura 20. Pantalla de licenciamiento del instalador de ODBC para Windows.....	31
Figura 21. Pantalla de productos que se pueden seleccionar para la instalación del ODBC para Windows.....	32
Figura 22. Pantalla de advertencia previo a la instalación final del ODBC para Windows..	32
Figura 23. Pantalla de finalización de instalación de ODBC para Windows.....	33
Figura 24. Pantalla de despliegue de la versión del OBDC Data Source.....	33
Figura 25. Pantalla de Systema Data source para agregar un nuevo DSN.....	34
Figura 26. Pantalla de selección de Proveedor del DSN.....	34
Figura 27. Pantalla de configuración DSN.....	35
Figura 28. Pantalla de test conexión después de la configuración DSN.....	36
Figura 29. Pantalla de Visual Studio para la configuración del Linked Server en SSMS.....	36
Figura 30. Pantalla de menú para agregar un nuevo servidor.....	37
Figura 31. Pantalla de propiedades de Linked Server para la conexión de Moodle - General.....	38
Figura 32. Pantalla de propiedades de Linked Server para la conexión de Moodle - Seguridad.....	38
Figura 33. Pantalla de propiedades de Linked Server para la conexión de Moodle – Opciones servidor.....	39
Figura 34. Pantalla de Prueba de la conexión de Moodle usando el Linked Server.....	40

<i>Figura 35.</i> Pantalla de resultado de la conexión de Moodle.....	40
<i>Figura 36.</i> Pantalla de ejecución del store procedure matricula estudiante moodle.	41
<i>Figura 37.</i> Pantalla de ejecución del store procedure eliminación matricula estudiante moodle.	41
<i>Figura 38.</i> Pantalla de ejecución del store procedure actualizar estudiante moodle.....	42
<i>Figura 39.</i> Pantalla de ejecución del store procedure salvar matricula del estudiante siges ...	42
<i>Figura 40.</i> Pantalla de ejecución del store procedure de cambio de paralelo estudiante.	43

RESUMEN

El objetivo de la siguiente investigación fue desarrollar una comunicación entre las plataformas SIGE (Sistema Integrado de Gestión Estratégica y EVA (Entorno Virtual de Aprendizaje) de la Universidad Tecnológica Israel bajo en el consumo de Linked Server como medio de integración.

Dentro del proceso de herramientas de la investigación que se usó se encuentra las capacitaciones y pruebas de diferentes alternativas, Luego se desarrolló la integración utilizando el método XP (programación extrema) que es el más usado dentro de la programación. Las fases o procesos para la implementación fueron: análisis, diseño, implementación y verificación de la comunicación de Linked Server entre los servidores de las plataformas mencionadas, permitiendo registrar la matriculación de estudiantes, así como el registro de usuarios si no lo hubiera en la plataforma EVA.

Se realizó algunas recomendaciones de la creación de procedimientos que se manejaran como buenas prácticas, se recomienda el mantener estable la comunicación entre el servidor de base de datos administrados por CEDIA y el servidor de base de datos de la Universidad Israel.

Palabras claves, Implementación, Linked Server, SIGE, EVA, Moodle.

ABSTRACT

The objective of this research was to develop a communication between the SIGE (Integrated Strategic Management System) and EVA (Virtual Learning Environment) platforms of the Israel Technological University under the use of Linked Server as a means of integration.

Among the process of research tools that were used are the trainings and tests of different alternatives, then the integration was developed using the XP method (extreme programming) which is the most used in programming. The faces or processes for the implementation were: analysis, design, implementation, and verification of the Linked Server communication between the servers of the mentioned platforms, allowing to register the enrollment of students, as well as the registration of users if there were not in the EVA platform.

Some recommendations were made for the creation of procedures to be managed as good practices, it is recommended to maintain stable communication between the database server managed by CEDIA and the database server of Israel University.

Keywords, Implementation, Linked Server, SIGE, EVA, Moodle.

INFORMACIÓN GENERAL

Contextualización del tema

Pregunta Problemática

P.P1. ¿De qué manera la integración de la plataforma virtual SIGE se relaciona con la plataforma EVA (Moodle) en el proceso de matriculación?

P.P2. ¿La creación de un proceso para la automatización del registro de matrícula permitirá mejorar el proceso de matriculación en la Universidad Israel?

Objetivo general

Diseñar una solución para la automatización del registro de matrícula mediante la integración de las plataformas SIGE y EVA para facilitar el proceso de matriculación de los estudiantes en la Universidad Tecnológica Israel.

Objetivos específicos

- Identificar las tablas intervinientes dentro del proceso de matriculación que utiliza la plataforma SIGE y plataforma EVA
- Definir el proceso de integración entre plataformas SIGE y EVA para el proceso de matriculación.
- Realizar las pruebas de la integración SIGE –EVA.
- Implementar la integración entre plataformas SIGE y EVA.

Beneficiarios directos:

Existen características técnicas de los programas que se convierten en herramientas útiles para el ingreso, procesamiento, almacenamiento y recuperación de información; las Nuevas Tecnologías de la Información y la Comunicación (NTIC) han impactado la vida cotidiana del hombre, y ésta tendrá la tendencia a socializarse cada día más. Así han creado amplias posibilidades de capacitación, razón por la que el rumbo de la educación debe ser transformado de un sistema clásico y conservador a un ambiente dinámico y creativo.

De esta manera se consideran como beneficiarios directos de este proyecto a toda la comunidad educativa de la Universidad Tecnológica Israel, es decir, el área administrativa en donde se involucran las secretarías y de esta manera sus actividades serán eficaces y eficientes a más de ser más prácticas y rápidas, además los estudiantes de las diferentes carreras, los docentes que la usarán como herramienta para resolver problemas en la enseñanza práctica y por último los coordinadores de carrera, ya que con este nuevo proceso, se reducirá la duplicidad de registro en el proceso de matriculación en las plataformas SIGE y EVA(Moodle).

CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

1.1. Contextualización de fundamentos teóricos

El presente proyecto está fundamentado en el “Pragmatismo” que se caracteriza por la utilidad y practicidad como componentes esenciales de la verdad cuya causa y efecto es la puesta en práctica en el área informática y administrativa de la Universidad Israel, pues esta perspectiva teórica que se plantea para implementar una integración innovadora entre los sistemas SIGE y EVA(Moodle) que ayude a facilitar y agilizar proceso de matriculación.

1.1.1. **El Sistema de Matrículas** es el que permite, en primera instancia, a un estudiante, vincularse formalmente como integrante de la comunidad educativa. Esta vinculación se establece mediante un contrato de servicios educativos, en el cual se enuncian los derechos, deberes y obligaciones económicas de los contratantes.

1.1.2. Un **Sistema de información** Según Peralta (2008):” Define sistema de información como: conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema”.

1.1.3. **“El Entorno virtual de Aprendizaje (EVA)**, es un espacio de aprendizaje mediado por la tecnología que facilita la comunicación, procesamiento y distribución de la información, permitiendo nuevas posibilidades para el aprendizaje e interacciones entre los actores que intervienen en el proceso de enseñanza a aprendizaje permitiendo la creación y mantenimiento de comunidades virtuales”, es presentado por (Carmona & Rodríguez 2009)



Figura 1. Entorno Virtual de Aprendizaje “EVA” de la Universidad Israel, donde los alumnos ingresan a las materias del ciclo correspondiente de matriculación.

Para Boneu (2007). “hay cuatro **características básicas, e imprescindibles**, que cualquier plataforma de e-learning debería tener:

- **Interactividad:** conseguir que la persona que está usando la plataforma tenga conciencia de que es el protagonista de su formación.
- **Flexibilidad:** conjunto de funcionalidades que permiten que el sistema de e-learning tenga una adaptación fácil en la organización donde se quiere implantar, en relación a la estructura institucional, los planes de estudio de la institución y, por último, a los contenidos y estilos pedagógicos de la organización.
- **Escalabilidad:** capacidad de la plataforma de e-learning de funcionar igualmente con un número pequeño o grande de usuarios.
- **Estandarización:** Posibilidad de importar y exportar cursos en formatos estándar como SCORM”

1.1.4. **Moodle** es una de aprendizaje desarrollado para crear y gestionar entornos de formación online, es decir, se trata de una herramienta de gestión de aprendizaje o LMS (Learning Management System). Siendo un gestor de contenidos escrito en PHP y open source, bajo licencia pública GNU, esto significa que Moodle es de código abierto y que cualquier usuario puede utilizarlo y modificarlo de forma gratuita, así como colaborar en su desarrollo.



Figura 2. Moodle, plataforma utilizada para aprendizaje educativo.

Estas son algunas de sus principales características:

- **Siempre a la última:** gracias a la comunidad que existe alrededor de Moodle, frecuentemente se lanzan nuevas versiones que incorporan mejoras y novedades para adaptarse a todas las necesidades.
- **Flexible y personalizable:** existen varias plantillas predefinidas para darle el aspecto que quieras a tu plataforma Moodle. También puedes desarrollar tus propios temas.
- **Interfaz intuitiva y fácil de utilizar:** Tiene un panel de control muy intuitivo que te hará la gestión de tus cursos mucho más sencilla.
- **Tablero personalizado:** en la plataforma Moodle puedes organizar tus cursos por semanas, temas, tipo de contenido, etc.
- **Actividades y herramientas colaborativas:** wikis, foros, diarios, cuestionarios, recursos, consultas, encuestas, tareas, chats y talleres para facilitar el aprendizaje de los alumnos.
- **Estadísticas:** acceso a los informes de actividad de los alumnos, rendimiento por cada módulo e historial de participación en el curso.
- **Calendario:** Moodle incluye una aplicación todo en uno que te permite incluir el calendario académico, fecha de entregas de trabajos, exámenes, reuniones o incluso añadir notas personales.
- **Editor de texto simple e intuitivo:** dispondrás de un editor de texto con el que podrás darles formato a tus textos y que es compatible en todos los navegadores y dispositivos.
- **Multiidioma:** tendrás la posibilidad de configurar tu sitio para usuarios u organizaciones multilingües.

Para Debons **sistema de información** es "un conjunto de personas, maquinaria y procedimientos que integrados hacen posible a los individuos trabajar con inputs y demandas que aparecen en el trabajo cotidiano".



Figura 3. Sistemas de Información

Sistemas de Información para la Gestión (MIS). “Los Sistemas de Información para la Gestión son un conjunto de herramientas que combinan las tecnologías de la información (hardware + software) con procedimientos que permitan suministrar información a los gestores de una organización para la toma de decisiones.

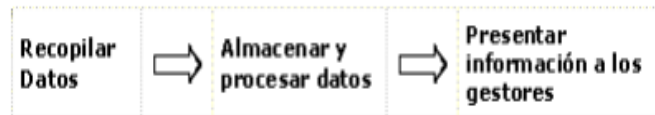


Figura 4. Sistemas de Información para la Gestión (MIS)

Sistema Operativo de Red. Un sistema operativo de red (Network Operating System) es un componente software de una computadora que tiene como objetivo coordinar y manejar las actividades de los recursos del ordenador en una red de equipos. Consiste en un software que posibilita la comunicación de un sistema informático con otros equipos en el ámbito de una red. Algunos Sistemas Operativos de Red.

El software del sistema operativo de red se integra en un número importante de sistemas operativos conocidos:

- Windows 2000 Server/Professional
- Windows NT Server/Workstation
- Apple Talk.
- GNU/Linux
- BSD
- NetBIOS

1.2. Problema a resolver

La Universidad Israel brinda formación para líderes actuales con una alta responsabilidad y pensamiento positivo, impactando a través de proyectos de investigación básica estimulando y entregando herramientas teóricas y prácticas para la visión integral y así sus estudiantes puedan impulsar iniciativas empresariales.

La Universidad Israel, presenta la necesidad de integrar la plataforma de SIGE (Sistema Integrado de Gestión Estratégica), con el EVA (Entorno Virtual de Aprendizaje) (Moodle), para la transparencia en la matriculación de estudiantes a materias, ya que año a año aumenta su población estudiantil lo que causa problemas en su proceso de matriculación en dos plataformas diferentes, por este motivo se hace urgente y necesaria la integración entre el sistema SIGE y su entorno de aprendizaje EVA.

Con esta integración de sistemas en el proceso de matriculación, la universidad podrá tener a su comunidad educativa registrada en ambas plataformas sin necesidad de acceder al sistema EVA y realizar los procesos adicionales que esto conllevaría.

1.3. Proceso de investigación

Con la utilización de un sistema informático que será diseñado con herramientas fáciles de aplicar y manipular, se facilitará a la comunidad educativa de la Universidad Israel el proceso de matriculación de los estudiantes de manera sencilla y eficaz dando así soluciones rápidas y efectivas.

Método Deductivo

Se habla del método deductivo para referirse a una forma específica de pensamiento o razonamiento, que extrae conclusiones lógicas y válidas a partir de un conjunto dado de premisas o proposiciones. Es, dicho de otra forma, un modo de pensamiento que va de lo más general (como leyes y principios) a lo más específico (hechos concretos).

Tabla 1. Método Deductivo procesos de Integración SIGE-Moodle

Determinación de la situación	Búsqueda de información	Recopilación de datos y análisis	Definición y formulación	Planteamiento de soluciones
La Universidad Israel, posee	Al tener conocimiento	Una vez que se obtuvo la	Cuando se ha analizado que es	Con la integración de

<p>dos plataformas para la gestión estratégica (SIGE), y el entorno virtual de aprendizaje (EVA), las cuales cumplen sus funciones de manera independiente sin embargo existe la necesidad de compartir la información entre las dos plataformas que se encuentran separadas. Es necesaria la intervención de la comunidad educativa para creación de cursos, matriculación y registro, de esta manera se realiza un doble trabajo al duplicar la información que ya está definida en la plataforma SIGE efectuado en el proceso de matriculación, volviendo a cargar en la plataforma EVA.</p>	<p>del problema que presentaba la IE, se procede a buscar y recolectar información necesaria, para indagar el por qué y cuáles son los motivos que aún se lleve un sistema lento al momento de realizar el proceso de matriculación.</p>	<p>información necesaria, se realizaron las respectivas entrevistas, resultados que fueron utilizados para analizar la propuesta que se tiene para dar solución al problema de investigación.</p>	<p>necesario una integración entre los sistemas SIGE y EVA en el módulo de matriculación de los estudiantes que permita ahorrar tiempo, realizando sus respectivos trámites, y de esta manera mejorar la calidad de atención que se brinda al usuario.</p>	<p>las plataformas SIGE y EVA en el módulo de matriculación, los estudiantes en el proceso correspondiente a la matriculación tendrán facilidad para ingresar y procesar la información desde el sistema SIGE, eliminado la duplicidad del proceso en el sistema EVA, permitiendo que la comunidad educativa se sienta a gusto al conocer que acudirá a la Institución con la seguridad que se encuentra registrada su matrícula en las dos plataformas de la institución.</p>
---	--	---	--	--

Este trabajo investigativo se aplicó a estudiantes, secretarías y docentes para conocer de qué manera se realiza en la actualidad el proceso de matriculación y cómo mejorará el servicio con la integración entre las plataformas SIGE – EVA (Moodle).

Metodologías Ágiles

Para el desarrollo ágil de software como término sólo ha existido desde principios de la década de 2000. Los primeros aspectos comparables de este método existían a principios de los años noventa. Con su libro "Extreme Programming" de 1999, el informático Kent Beck creó una especie de base teórica para futuros procesos ágiles. En

2001, el "Manifiesto Ágil" fue formulado por un grupo liderado por Kent Beck, que contiene los principios del desarrollo ágil de software. (Ryte, 2020, p.1)



Figura 5. Ciclo de metodología ágil vs la metodología tradicional

Metodología XP

La Metodología XP "Extreme Programming" o "Programación Extrema" es una de las llamadas metodologías Ágiles de desarrollo de software más exitosas. Es habitual relacionarla con scrum, y la combinación de ambas asegura un mayor control sobre el proyecto, y una implementación más efectiva y eficiente. (Vila, 2016, p.1)

La programación extrema XP está enfocada al desarrollo en equipo, es por esto que define un conjunto de valores que deben tener, además incluye al cliente como parte fundamental ya que sin él no se tendrían los requerimientos del producto. Scrum al igual que XP tiene un equipo de trabajo, la única diferencia es que divide el equipo en scrum master (líder), DBA (administrador de la base de datos), Programadores, diseñadores y el product owner (el cliente). (Cevallos, 2015, p.1).

Metodología XP o Programación Extrema

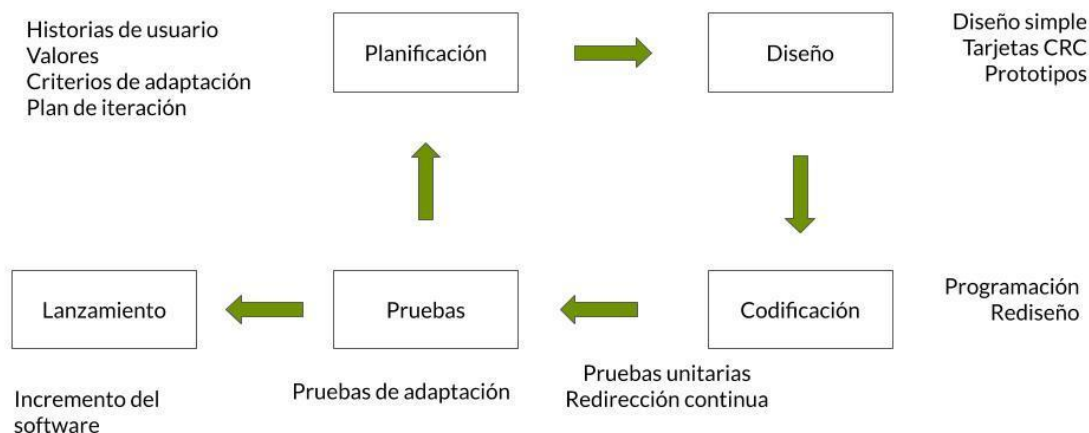


Figura 4. Metodología XP o Programación Extrema

1.4. Vinculación con la sociedad

El proyecto de integración entre las plataformas SIGE y EVA se vinculará directamente con toda la comunidad educativa.

El proceso de matriculación es efectuado en la herramienta SIGE por la secretaría, docentes y alumnos, por lo que es necesario pasar esta información a la plataforma EVA de manera automática manteniendo la integridad en la información.

Con esta integración de sistemas en el proceso de matriculación, la universidad podrá tener a su comunidad educativa registrada en ambas plataformas sin necesidad de acceder al sistema EVA(Moodle) y realizar los procesos adicionales que esto conllevaría, beneficiando así a todos los componentes humanitarios que laboran en la Universidad.

1.5. Indicadores de resultados

Este proyecto será desarrollado y debidamente probado en el ambiente de desarrollo al término de su creación, para así poder asegurar la máxima eficacia y rendimiento de este, dar garantía de que, con la integración SIGE-EVA(Moodle), se agilizará la manera en cómo se realiza el proceso de matriculación de los estudiantes.

CAPÍTULO II: PROPUESTA

2.1. Fundamentos teóricos aplicados

Moodle: “Acrónimo de Module Object-Oriented Dynamic Learning Environment (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos), desarrollado por el pedagogo e informático australiano Martin Dougiamas en el año de 2002, es una plataforma de aprendizaje diseñada para proporcionarle a educadores, administradores y estudiantes en un entorno de enseñanza virtual. Un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados, basado en tecnología PHP y bases de datos.” **Moodle. (2020).**

Sistema de Gestión: “Es una herramienta que establece el protocolo a seguir en el día a día de una actividad productiva. Su implantación permitirá optimizar los recursos disponibles, mejorar la organización, una reducción de costes y mejorar el rendimiento de la empresa. Adoptar un sistema de gestión en una compañía es una garantía de que el proceso cumple los estándares estipulados por normas internacionales. Cada disciplina de la empresa puede tener su propio sistema de gestión.” **CTMA Consultores 15 (2018).**

Base de Datos: “Es un conjunto de información que se organiza de manera que se pueda acceder a ella, administrarla y de fácil actualización. Las bases de datos suelen contener registros o archivos de datos, que mantienen información sensible y perteneciente al negocio”. **María Estela Raffino. (2020).**

Microsoft SQL Server “Es un sistema relacional de administración de bases de datos, o RDBMS, desarrollado y comercializado por Microsoft, se basa en SQL, un lenguaje de programación estándar para interactuar con las bases de datos relacionales. SQL ServerSQL Server está vinculado a Transact-SQLTransact-SQL o T-SQL, la implementación de SQL de Microsoft que agrega un conjunto de construcciones de programación propietarias”. **SQLServerTutorial.Net. (2020)**

Linked Server: “Los servidores enlazados permiten enviar una sentencia T-SQL a una instancia de SQL Server, que retorna datos desde otras instancias. Un servidor enlazado permite combinar datos desde muchas instancias usando una sola sentencia T-SQL cuando los datos existen en múltiples bases de datos en diferentes instancias SQL.” **Marko Zivkovic. (2018).**

Postgres: “Sistema de administración de base de datos relacionales de objetos (ORBDMS), es compatible con una gran parte del estándar SQL. posee licencia gratuita, esto permite su utilización, modificación y distribución por cualquier usuario de manera gratuita sin importar el propósito del uso pudiendo ser, privado, comercial o académico.” **Postgres. (2020).**

ODBC: “Open Data Base Connectivity estándar de acceso a bases de datos utilizado en los sistemas Microsoft. Por medio de esta herramienta los sistemas Windows se puede conectar con cualquier base de datos. Los desarrolladores de las diferentes bases de datos son los encargados de crear los drivers ODBC para que su base de datos se pueda conectar desde un sistema Microsoft”. **Microsoft. (2017).**

Visual Studio: “Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos los mismos entornos de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML, fue lanzado en 1997, cuenta con versiones gratis y de venta”. **Microsoft. (2019).**

PuTTY: “Es programa de terminal versátil para Windows, es el cliente SSH gratuito más popular del mundo que permite conexiones SSH, telnet y socket sin procesar con buena emulación de terminal. Es compatible con la autenticación de clave pública y el inicio de sesión único de Kerberos”. **PuTTY.(2017).**

2.2. Descripción de la propuesta

Estructura general

Descripción de la propuesta a través de diagrama de secuencia.

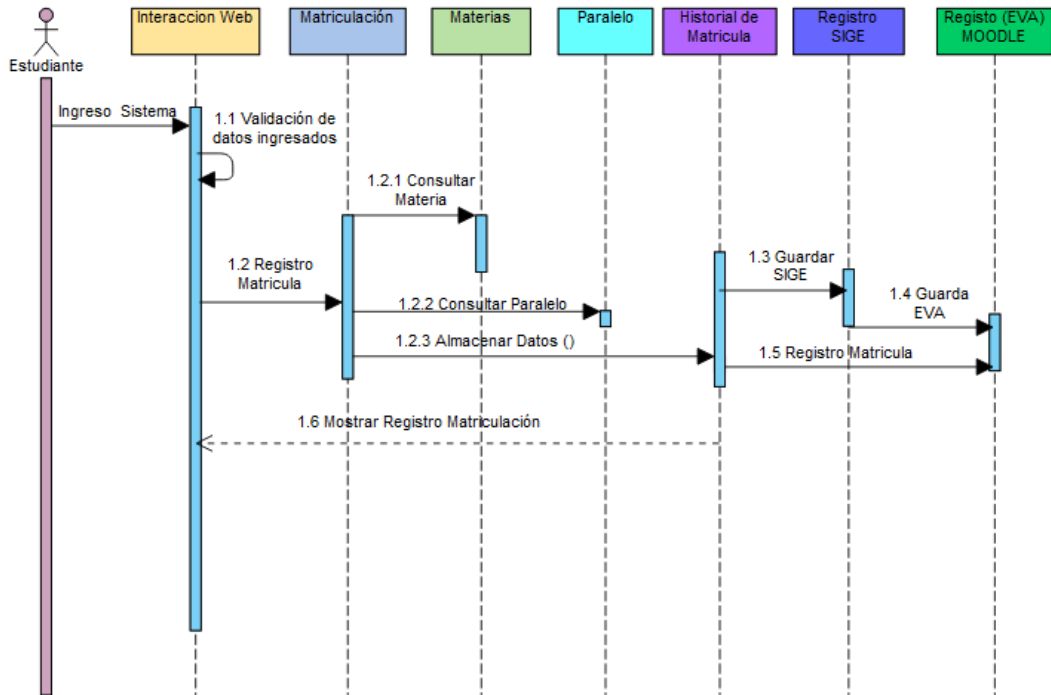


Figura 6. Integración SIGE-EVA (Moodle) Diagrama de Secuencia donde se muestra los diferentes pasos para la interacción

En la siguiente figura se muestra el proceso en general de integración entre los sistemas SIGE y EVA a nivel de base de datos por medio de un Linked Server y la ejecución de procedimientos almacenados.

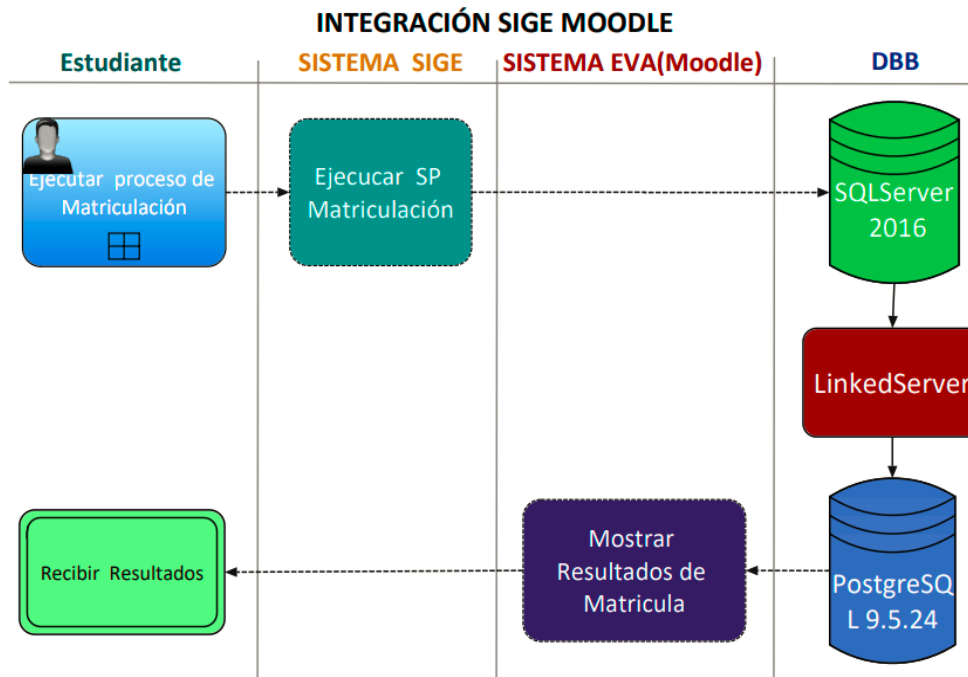


Figura 7. Representación grafica Proceso general de Integracion SIGE-EVA(Moodle), diagrama macro de los procesos más representativos de la implementación.

En la siguiente figura se muestra el proceso de registro de matrícula de un estudiante mediante la integración entre los sistemas SIGE y EVA a nivel de base de datos, por medio de un Linked Server y la ejecución de procedimientos almacenados.

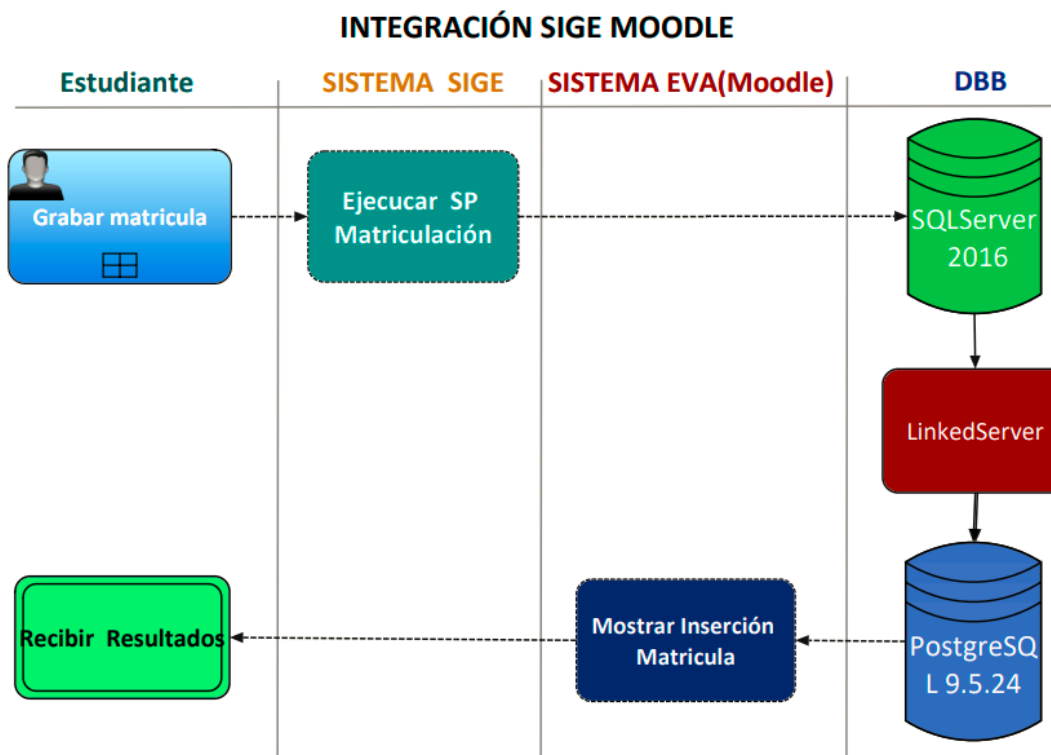


Figura 8. Representación gráfica Proceso Registro de Matriculación de Integración SIGE-EVA (Moodle)

En la siguiente figura se muestra el proceso de cambio de paralelo de un estudiante mediante la integración entre los sistemas SIGE y EVA a nivel de base de datos por medio de un Linked Server y la ejecución de procedimientos almacenados.

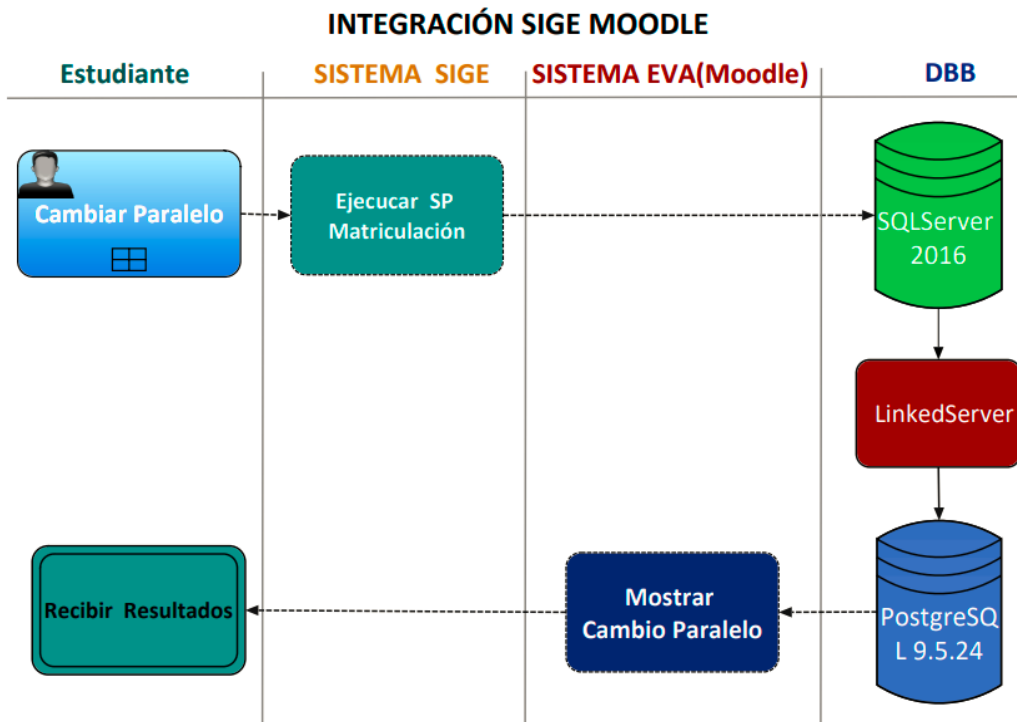


Figura 9. Representación gráfica Proceso Cambio de Paralelo de Matriculación de Integración SIGE-EVA (Moodle)

En la siguiente figura se muestra el proceso de eliminar materia de un estudiante que ejecuta el coordinador de carrera mediante la integración entre los sistemas SIGE y EVA a nivel de base de datos por medio de un Linked Server y la ejecución de procedimientos almacenados.

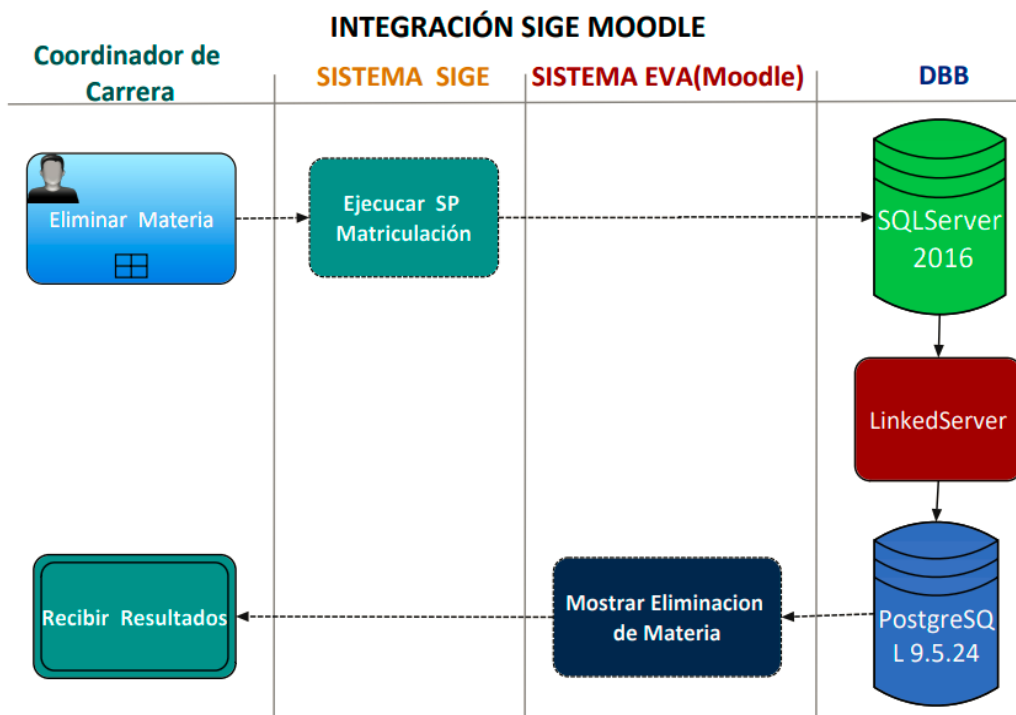


Figura 10. Representación gráfica Proceso Dar de Basa materia de Integración SIGE-EVA (Moodle)

2.3. Explicación del aporte

Actualmente la Universidad Israel año a año aumenta su población estudiantil, lo que causa problemas en su proceso de matriculación en dos plataformas diferentes, por este motivo se hace urgente y necesaria la integración entre el sistema SIGE y su entorno de aprendizaje EVA.

Con esta integración de sistemas en el proceso de matriculación, la universidad podrá tener a su comunidad educativa registrada en ambas plataformas sin necesidad de acceder al sistema EVA y realizar los procesos adicionales que esto conlleva.

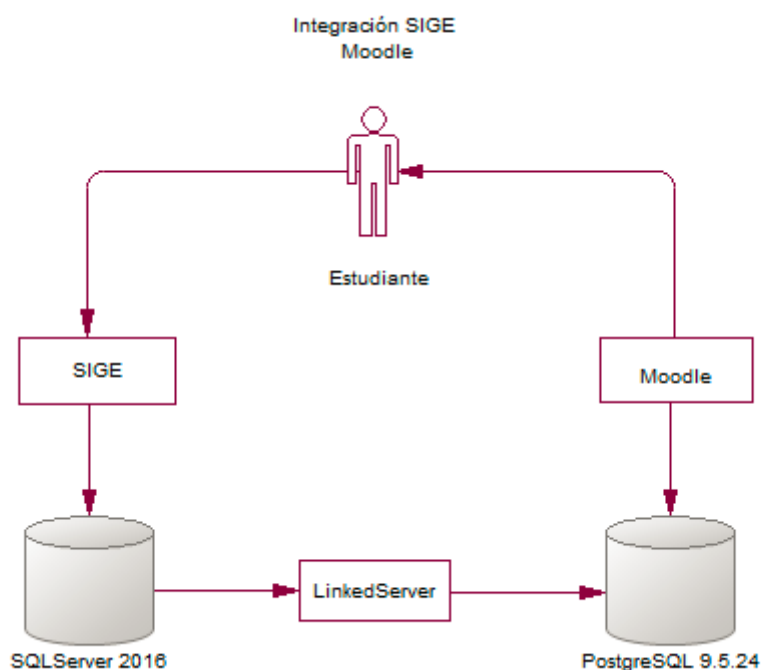


Figura 11. Ciclo de Integración SIGE-Moodle

La integración entre los sistemas SIGE y EVA se genera a nivel de base de datos, mediante un enlace de servidor y el uso de procedimientos que utilizan los siguientes componentes.

Stored procedure: Procedimiento almacenado que ejecuta el script donde valida la información entre base de datos SQL Server y Postgres para poder insertar, actualizar o eliminar registros en las tablas de matriculación del sistema Moodle.

Linked Server: Conexión que vincula los servidores SQL Server y Postgres, facilitando la comunicación y permitiendo generar los scripts entre tablas de las distintas bases de datos.

ODBC: Permite el acceso a la base de datos Postgres desde cualquier sistema Windows.

2.3.1. Funcionamiento del proceso

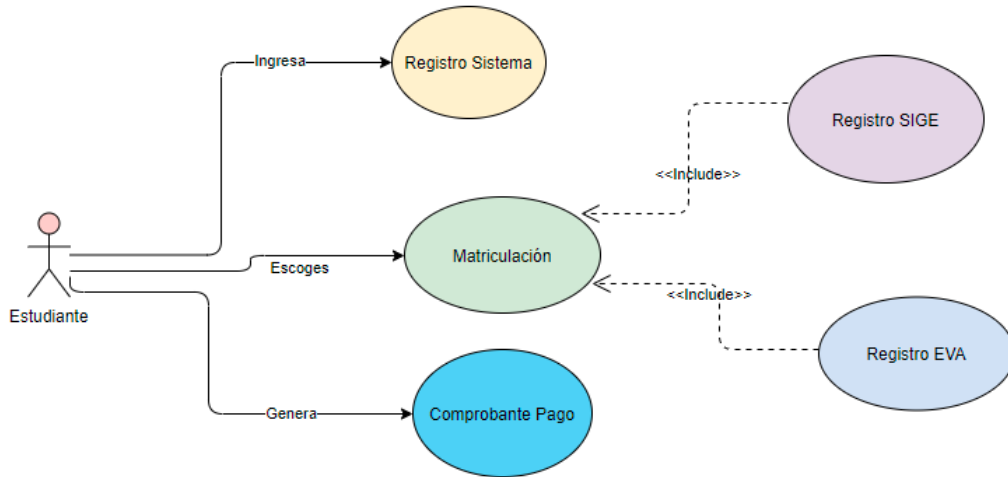


Figura 12. Casos de Uso del estudiante en el proceso de matriculación y la interacción con las plataformas del SIGE-EVA(Moodle)

El sistema SIGE en su proceso de matriculación llama el procedimiento almacenado que registra la información en la base de datos SQL Server, ejecutando SPs de integración y registro de datos directos desde el SIGE hacia la base de datos Postgres del sistema Moodle, eliminando la duplicidad del trabajo del alumno de volver a registrar su matrícula en el sistema Moodle.

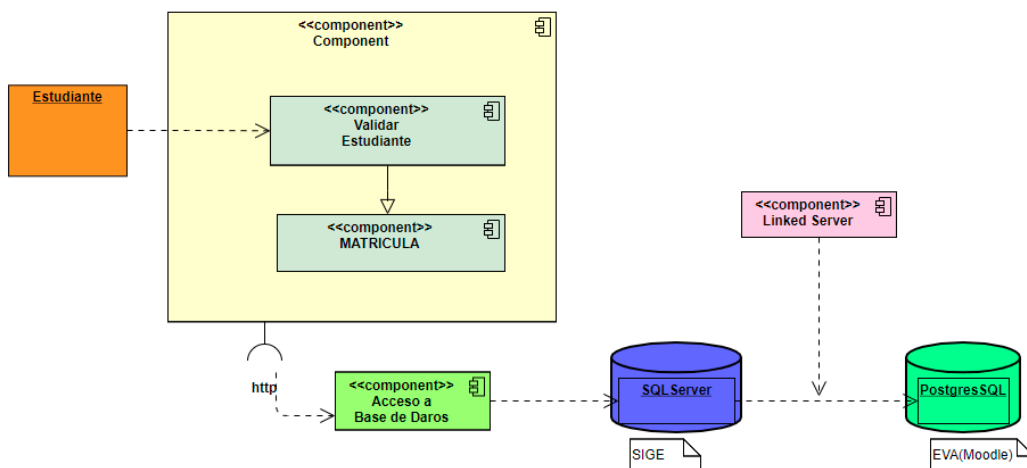


Figura 13. Diagrama de componentes Integración SIGE-EVA(Moodle)

2.3.2. Estrategias y/o técnicas

1) HISTORIAS DE USUARIO

A continuación se presentan las historias de usuarios más relevantes para la estructuración y creación del proceso de integración SIGE-Moodle en el módulo de matriculación de los estudiantes.

Tabla 2. Historia de Usuario 01

Historia de Usuario			
Número:	1	Usuario:	Estudiante
Nombre historia:	Matricular en materia – Moodle		
Prioridad de negocio:	Alta	Riesgo en desarrollo:	Media
Puntos estimados:	1	Interacción asignada:	1
Responsable:	José Fabre		
Descripción:	Quiero que las materias habilitadas en el proceso de matriculación que se realiza en la plataforma el SIGE, se agreguen las que apliquen en la plataforma EVA.		

Tabla 3. Historia de usuario 02

Historia de Usuario			
Número:	2	Usuario:	Estudiante
Nombre historia:	Cambio de paralelo – Moodle		
Prioridad de negocio:	Alta	Riesgo en desarrollo:	Media
Puntos estimados:	1	Interacción asignada:	1
Responsable:	Diego Atiaja		
Descripción:	Quiero que las materias habilitadas en el proceso de matriculación que se realiza en la plataforma el SIGE, si se realiza el cambio de paralelo, el cambio se aplique también en la plataforma EVA.		

Tabla 4. Historia de usuario 03

Historia de Usuario			
Número:	3	Usuario:	Tutor
Nombre historia:	Des matricular materia – Moodle		
Prioridad de negocio:	Alta	Riesgo en desarrollo:	Media
Puntos estimados:	1	Interacción asignada:	1
Responsable:	Rubén Villalba		
Descripción:			

Quiero tener la opción de eliminar la matrícula de un estudiante por materia. el cambio debe de igual manera reflejarse en la eliminación del curso en la plataforma EVA para el estudiante.

2) Inducciones

Debido a que el trabajo de la integración se realiza entre el sistema SIGE (Sistema Integrado de Gestión Estratégica), de autoría de la Universidad Israel encabezado por el MSc. Paúl Baldeón desarrollador, y el sistema EVA(Moodle) que es una plataforma web que permite gestionar entornos de enseñanza virtual, se recibe capacitaciones y reuniones de trabajo por parte del personal técnico encargado de los diferentes sistemas.

Tabla 1. Cronograma de Capacitaciones SIGE-EVA(Moodle)

No.	Fecha	Modulo	Responsable
1	21/11/2020	SIGE(SQLServer)	Esteban Silva
2	28/11/2020	SIGE(SQLServer)	Esteban Silva
3	5/12/2020	SIGE(SQLServer)	Esteban Silva
4	12/12/2020	SIGE(SQLServer)	Esteban Silva
5	25/11/2020	SIGE(SQLServer)	Esteban Silva
6	9/1/2021	SIGE(SQLServer)	Esteban Silva
10	11/12/2020	Moodle(Administración)	Ángelo Mendoza Varela

Gracias a las reuniones con el personal técnico encargado, el siguiente trabajo se pudo encaminar de una manera objetiva, permitiendo tener resultados favorables en el proyecto de integración entre las plataformas que intervienen en el proceso de matriculación de los estudiantes de la Universidad Israel.

3) Análisis de Esquemas de Bases de Datos SIGE, Moodle.

Considerando que la integración de los sistemas SIGE y Moodle en el proceso de matriculación es a nivel de bases de datos, se realiza el respectivo análisis de las diferentes bases de datos, obteniendo como resultados las tablas que intervienen en el proceso antes mencionado

SIGE (Base de datos SQL Server)

- TMGA_PERSONAS
- TMGA_SEDES

- TMGA_PERSONAS
- TMGA_ASIGNATURAS
- TMGA_INSCRIPCIONES
- RMGA_CARRERAS_MODALIDADES
- RMGA_SEDES_AREAS
- TMGA_CARRERAS
- TMGA_MODALIDADES
- TMGA_TIPOS_INSCRIPCIONES
- TMGA_TIPOS_MATRICULAS
- TMGA_MATRICULAS
- TMGA_PERIODOS_ACADEMICOS
- TMGA_NIVELES
- TMGA_PROFESORES
- TMGA_MALLAS_DETALLES
- TMGA_PLANIFICACIONES
- TMGA_MATRICULAS_DETALLES
- TMGA_PLANIFICACIONES_MODULAR
- TMGA_MOMENTOS
- RMGA_PLANIFICACION_HORARIOS
- TMGA_HORARIOS
- TMGA_DIAS_SEMANAS

Modelo entidad relación se puede observar en el **ANEXO 1**

Store Procedures (Base de datos SQL Server)

- **SP_SAVE_MATRICULA_EST_POR_ASIGNATURA_2** - Store procedure utilizado en el proceso de matriculación del SIGE que matricula al estudiante por materia.
- **SP_CAMBIO_PARALELO_ESTUDIANTE** - Store procedure utilizado en el proceso de matriculación del SIGE que permite al estudiante cambiar de paralelo en las materias disponibles.

El Moodle trabaja con el BDMS de PostgreSQL con su versión para la plataforma 9.5.24 que en su modelo se encuentra en su 3FN con el proceso de Des normalización para optimizar el funcionamiento de la plataforma.

MOODLE (Base de datos PostgreSQL)

- mdl_course
- mdl_context
- mdl_user
- mdl_role
- mdl_user_enrolments
- mdl_role_assignments

- mdl_course_categories

Modelo entidad relación se puede observar en el **ANEXO 2**

4) Configure linked Server SqlServer a PostgreSQL

Como parte fundamental dentro de la integración de los sistemas SIGE y Moodle, se encuentra la comunicación entre los motores de bases de datos SQL Server y PostgreSQL, la herramienta Linked Server de SQL Server permite la combinación de datos entre las distintas instancias.

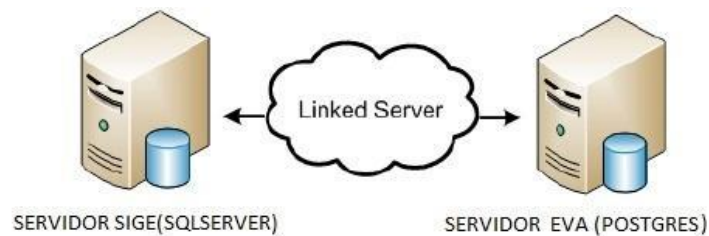


Figura 14. Diagrama de interpretación del Linked Server

La conectividad del Linked Server se genera vía ODBC que apunta hacia el motor de base de datos PostgreSQL, para ello es primordial la instalación y configuración del ODBC de Postgres.

5) Desarrollo de Store Procedure

Identificadas las diferentes tablas involucradas y las relaciones entre ellas, en el proceso de matriculación tanto en el sistema SIGE como EVA (Moodle), y el enlace entre bases de datos mediante el Linked Server, se diseñan los diferentes scripts para los procedimientos almacenados que permitirán la integración a nivel de base de datos en el proceso de matriculación.

Lógica que de los Stores Procedures de integración.

- Parámetros de entrada Código de inscripción, Código de planificación, estos parámetros permitirán la validación.

- La validación se realiza mediante una verificación de existencia de la misma la planificación tanto en Las bases de datos del SIGE (SQL Server) y Moodle (PostgreSQL), en base los parámetros de entrada
- Si el resultado de la validación es positivo se procede a la Inserción(INSERT), modificación(UPDATE) o Eliminación(DELETE) del estudiante a la materia o materias

```

SELECT
    ASI.ASI_DESCRIPCION
FROM
    TMGA_PERSONAS TP
INNER JOIN
    TMGA_SEDES TS ON 1=1
INNER JOIN
    TMGA_PERSONAS TPI ON 1=1
INNER JOIN
    TMGA_ASIGNATURAS ASI ON UPPER(SUBSTRING(ASI.ASI_DESCRIPCION, 1, 10)) not IN (SELECT UPPER(SUBSTRING(mdl_course.fullname, 1, 10))
FROM MOODLE.[posgrado].[public].mdl_course
INNER JOIN MOODLE.[posgrado].[public].mdl_context ON mdl_context.instanceid = mdl
INNER JOIN MOODLE.[posgrado].[public].mdl_role_assignments ON mdl_context.id = mdl
and mdl_role_assignments.userid =(select A.ID
from MOODLE.[posgrado].[public].mdl_user
where UPPER(A.FIRSTNAME) = UPPER('TORRES F
and UPPER(A.LASTNAME) = UPPER('HENRRY PAÚL
OR SUBSTRING(A.USERNAME, 2, 10)='17148997E
INNER JOIN MOODLE.[posgrado].[public].mdl_role
INNER JOIN MOODLE.[posgrado].[public].mdl_user
WHERE mdl_role.shortname = 'student' --Paramet
and mdl_course.visible =1
and mdl_course.enablecompletion=1)

INNER JOIN
    TMGA_INSCRIPCIONES TI ON TP.PER_CODIGO = TI.PER_CODIGO AND TI.INS_ESTADO = 1
INNER JOIN
    RMGA_CARRERAS_MODALIDADES RCM ON RCM.RCM_CODIGO = TI.RCM_CODIGO
INNER JOIN
    RMGA_SEDES_AREAS RSA ON RSA.SED_CODIGO = TS.SED_CODIGO
INNER JOIN
    TMGA_CARRERAS TC ON TC.CAR_CODIGO = RCM.CAR_CODIGO AND TC.SDA_CODIGO = RSA.SDA_CODIGO
INNER JOIN
    TMGA_MODALIDADES TM ON TM.MOD_CODIGO = RCM.MOD_CODIGO
INNER JOIN
    TMGA_TIPOS_INSCRIPCIONES TPI ON TPI.TPI_CODIGO = TI.TPI_CODIGO
INNER JOIN
    TMGA_TIPOS_MATRICULAS TTM ON TTM.TPM_CODIGO IN (1,2)
INNER JOIN
    TMGA_MATRICULAS M ON TTM.TPM_CODIGO = M.TPM_CODIGO AND M.MTR_ESTADO = 1 AND M.INS_CODIGO = TI.INS_CODIGO AND M.PRA_CODIGO IN ( SELECT PRA_CODIGO FROM TMGA
INNER JOIN
    TMGA_PERIODOS_ACADEMICOS PRA ON PRA.PRA_CODIGO = M.PRA_CODIGO AND PRA.PRA_ESTADO=1 AND PRA.PRA_HABIL_MATRICULA=1
INNER JOIN
    TMGA_NIVELES NI ON NI.NIV_CODIGO = M.NIV_CODIGO
INNER JOIN
    TMGA_PROFESORES PRF ON PRF.PER_CODIGO = TPI.PER_CODIGO
INNER JOIN
    TMGA_MALLAS_DETALLES MAD ON MAD.ASI_CODIGO = ASI.ASI_CODIGO
INNER JOIN
    TMGA_PLANIFICACIONES PLA ON PLA.PRF_CODIGO = PRF.PRF_CODIGO AND PLA.DTN_CODIGO = MAD.DTN_CODIGO

```

Figura 15. Scripts de validación Integración matriculación

SPI_MATRICULA_ESTUDIANTE_MOODLE:

Store procedure que realiza la inserción(INSERT) del estudiante a la materia en la base de datos postgresql del Moodle, recibe como parámetros el código de inscripción y el código de planificación

SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE:

Store procedure que realiza la eliminación (DELETE) del estudiante en la materia sobre la base de datos postgresql del Moodle, recibe como parámetros el código de inscripción y el código de planificación

SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE:

Store procedure que realiza la eliminación e inserción del estudiante a una nueva planificación correspondiente al cambio de paralelo, recibe como parámetros el código de inscripción, código de planificación anterior y código de planificación nuevo.

6) Diseño de casos de pruebas

El éxito de un desarrollo se demuestra por medio de los resultados que emiten las pruebas realizadas al proceso.

Considerando los diferentes escenarios y las historias de usuario, se diseñan los casos de prueba que permiten comprobar el éxito o fracaso de la integración de las plataformas en el proceso de matriculación de estudiantes de la Universidad Israel.

Tabla 2. Casos de prueba

Módulo / Funcionalidad	Id. CPI	Escenario	Objetivo de la prueba	Paso a Paso /Pre condiciones	Resultado o Comportamiento Esperado	Nivel de Criticidad
MATRICULACION SIGE	SIGE-MOO D-001	Ejecutar SPI_MATRICULA_ESTUDIANTE con datos de un estudiante para registro de matrícula BDD SIGE a Moodle	Inserción del estudiante a la materia por medio del SP en las bases de datos SQL Server (SIGE) y los cambios se visualicen automáticamente en la base de datos Postgres (EVA)	Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	Ejecutar el SP de registro de matrículas en el SIGE, automáticamente se registre la matrícula en el sistema EVA	Alto
MATRICULACION SIGE	SIGE-MOO D-002	Ejecutar SPI_ELIMINAR_MATRICULA_ESTUDIANTE con datos de un estudiante para registro de matrícula BDD SIGE a Moodle	Eliminación del estudiante a la materia por medio del SP en las bases de datos SQL Server (SIGE) y los cambios se visualicen automáticamente en la base de datos Postgres (EVA)	Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	Ejecutar el SP de registro de matrículas en el SIGE, automáticamente se registre la matrícula en el sistema EVA	Alto
MATRICULACION SIGE	SIGE-MOO D-003	Ejecutar SPI_ACTUALIZAR_MATRICULA_ESTUDIANTE con datos de un estudiante para registro de matrícula BDD SIGE a Moodle	Eliminación e inserción del estudiante a una nueva planificación correspondiente al cambio de paralelo por medio del SP en las bases de datos SQL Server (SIGE) y los cambios se visualicen automáticamente en la base de datos Postgres (EVA)	Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	Ejecutar el SP de registro de matrículas en el SIGE, automáticamente se registre la matrícula en el sistema EVA	Alto
MATRICULACION SIGE	SIGE-MOO D-004	Estudiante registre su matrícula desde el sistema SIGE	Ejecutar un solo proceso de matriculación desde sistema SIGE y la inserción también se ejecute en el EVA de forma automática	Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	Al registrar el estudiante su matrícula en el SIGE, automáticamente se registre la matrícula en el sistema EVA	Alto
MATRICULACION SIGE	SIGE-MOO D-005	Estudiante realiza el cambio de paralelo de las materias	Cambiar de paralelo las materias seleccionadas desde el sistema SIGE y los cambios también se ejecuten en el EVA de forma automática	Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	En el momento de realizar el cambio de paralelo del estudiante su matriculado en el SIGE, automáticamente se registre los cambios en el sistema EVA	Alto

MATRICULACION SIGE	SIGE-MOO D-006	Coordinador de carrera elimina materia de un estudiante en el sistema SIGE, ejecutar SP Matriculación , inserte la materias en el sistema SIGE Y EVA (Moodle)	Dar de baja una materia de un estudiante y volver a registrar mediante el SP de matriculación, mostrando los resultados en el SIGE y EVA	Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	En el momento de realizar el registro del estudiante su matriculado en el SIGE, automáticamente se registre los cambios en el sistema EVA	Alto
---------------------------	-------------------	--	--	---	---	------

7) Ejecución y resultados de casos de pruebas

La ejecución de los diferentes casos de prueba tubo resultados satisfactorios, demostrando la correcta ejecución de la integración de los sistemas SIGE y Moodle en el proceso de matriculación y sus diferentes acciones.

Tabla 3. Resultados cado de Prueba 01

Nombre Caso de Prueba	Código
SPI_MATRICULA_ESTUDIANTE_MOODLE con datos de un estudiante para registro de matrícula BDD SIGE a Moodle	SIGE-MOOD-001
Descripción	
Inserción del estudiante a la materia por medio del SP en las bases de datos SQL Server (SIGE) y los cambios se visualicen automáticamente en la base de datos Postgres (EVA)	
Prerrequisitos	
Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	
Resultados esperados	
Ejecutar el SP de registro de matrículas en el SIGE, automáticamente se registre la matricula en el sistema EVA	
Resultados Obtenidos	
Los resultados fueron satisfactorios ya que se produjo la integración de forma adecuada, permitiendo reflejar los resultados en las dos plataformas	

Tabla 4.. Resultados cado de Prueba 02

Nombre Caso de Prueba	Código
SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE con datos de un estudiante para registro de matrícula BDD SIGE a Moodle	SIGE-MOOD-002
Descripción	
Eliminación del estudiante a la materia por medio del SP en las bases de datos SQL Server (SIGE) y los cambios se visualicen automáticamente en la base de datos Postgres (EVA)	
Prerrequisitos	
Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	
Resultados esperados	

Ejecutar el SP de registro de matrículas en el SIGE, automáticamente se reflejara el cambio en el sistema EVA

Resultados Obtenidos

Los resultados fueron satisfactorios ya que se produjo la integración de forma adecuada, permitiendo reflejar los resultados en las dos plataformas

Tabla 5. Resultados cado de Prueba 03

Nombre Caso de Prueba	Código
SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE con datos de un estudiante para registro de matrícula BDD SIGE a Moodle	SIGE-MOOD-003
Descripción	
Eliminación e inserción del estudiante a una nueva planificación correspondiente al cambio de paralelo por medio del SP en las bases de datos SQL Server (SIGE) y los cambios se visualicen automáticamente en la base de datos Postgres (EVA)	
Prerrequisitos	
Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	
Resultados esperados	
Ejecutar el SP de registro de matrículas en el SIGE, automáticamente se registre la matrícula en el sistema EVA	
Resultados Obtenidos	
Los resultados fueron satisfactorios ya que se produjo la integración de forma adecuada, permitiendo reflejar los resultados en las dos plataformas	

Tabla 6.. Resultados cado de Prueba 04

Nombre Caso de Prueba	Código
Estudiante registre su matrícula desde el sistema SIGE	SIGE-MOOD-004
Descripción	
Ejecutar un solo proceso de matriculación desde sistema SIGE y la inserción también se ejecute en el EVA de forma automática	
Prerrequisitos	
Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	
Resultados esperados	
Al registrar el estudiante su matrícula en el SIGE, automáticamente se registre la matricula en el sistema EVA	
Resultados Obtenidos	
Los resultados fueron satisfactorios ya que se produjo la integración de forma adecuada, permitiendo reflejar los resultados en las dos plataformas	

Tabla 7.. Resultados cado de Prueba 05

Nombre Caso de Prueba	Código
Estudiante realiza el cambio de paralelo de las materias	SIGE-MOOD-005
Descripción	
Cambiar de paralelo las materias seleccionadas desde el sistema SIGE y los cambios también se ejecuten en el EVA de forma automática	
Prerrequisitos	
Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	
Resultados esperados	
En el momento de realizar el cambio de paralelo del estudiante su matriculado en el SIGE, automáticamente se registre los cambios en el sistema EVA	
Resultados Obtenidos	
Los resultados fueron satisfactorios ya que se produjo la integración de forma adecuada, permitiendo reflejar los resultados en las dos plataformas	

Tabla 8.. Resultados cado de Prueba 06

Nombre Caso de Prueba	Código
Coordinador de carrera elimina materia de un estudiante en el sistema SIGE, ejecutar SP Matriculación , inserte la materias en el sistema SIGE Y EVA (Moodle)	SIGE-MOOD-006
Descripción	
Dar de baja una materia de un estudiante y volver a registrar mediante el SP de matriculación, mostrando los resultados en el SIGE y EVA	
Prerrequisitos	
Registro de datos del Estudiante, materias, cursos y planificación en los sistemas SIGE y EVA	
Resultados esperados	
En el momento de realizar el registro del estudiante su matriculado en el SIGE, automáticamente se registre los cambios en el sistema EVA	
Resultados Obtenidos	
Los resultados fueron satisfactorios ya que se produjo la integración de forma adecuada, permitiendo reflejar los resultados en las dos plataformas	

2.4. Matriz de articulación

En la presente matriz se sintetiza la articulación del producto realizado con los sustentos teóricos, metodológicos, estratégicos-técnicos y tecnológicos empleados.

Tabla 5. Matriz de articulación

EJES O PARTES PRINCIPALES	SUSTENTO TEÓRICO	SUSTENTO METODOLÓGICO	ESTRATEGIAS / TÉCNICAS	DESCRIPCIÓN DE RESULTADOS	TIC
Analizar Requerimientos	Ingeniería del Software Un enfoque práctico 7th Ed. Roger S. Pressman, Capitulo 6 y 7.	Metodología XP Fase Análisis	Análisis de Historias de usuarios	La historia de usuario permite definir el con mayor claridad el enfoque del proyecto.	Word, Excel, SIGE, Moodle
Identificar Secuencia de Proceso	Reunión con los involucrados (Ángelo Geovanny Mendoza Varela 2020), Capacitación Moodle; Esteban Silva, Paul Balden capacitación SIGE.	Metodología XP Fase Análisis	Reuniones de capacitaciones y trabajo a realizar	Por medio de las diferentes reuniones y capacitaciones con las personas involucradas directa e indirectamente con el proyecto, permitiendo conocer las secuencias que realizan los diferentes sistemas en el proceso de matriculación.	SIGE, Moodle, SQL Server, PostgreSQL
Revisar Manual de Moodle y SIGE	Manuales de Moodle Manual para Administradores, Moodledev page y para SIGE reunión con Esteban	Metodología XP Fase Diseño	Análisis de Esquemas de bases de datos (M.E.R) SIGE- Moodle	La determinación de las tablas que intervienen en el proceso de matriculación tanto en el sistema SIGE como Moodle. Con esta información	SQLServer, PostgreSQL, Putty, SQLManagemenStudio,PGAdmin 4

	Silva, Paul Balden 2020, Modelo de Datos SIGE.			se viabilizo el uso un Linked Server entre SQL Server y PostgreSQL	
Generar script de SQL	Microsoft Documentación de SQL Artículo "Generar scripts (SQL Server Management Studio).	Metodología XP Fase Desarrollo	Elaboración de los Store Procedure con la lógica para realizar la Inserción, Actualización y Eliminación a través de un Linked Server en el proceso de matriculación SIGE-Moodle	La correcta estructura de los scripts permite un adecuado proceso de validación, consultas, inserción y actualización al momento de matricular un estudiante.	SQLServer, PostgreSQL, Putty, SQLManagemenStudio, PGAdmin 4
Definir Pruebas de Integración	Ingeniería del Software Un enfoque práctico 7th Ed. Roger S. Pressman, Capitulo 18, 19 y 20.	Metodología XP Fase Pruebas	Creación de casos de pruebas de los distintos escenarios	La correcta elaboración de pruebas de la integración permite un adecuado enfoque a los puntos que se desea evaluar	Excel, Word, Google Drive
Ejecutar pruebas de Integración	Ingeniería del Software Un enfoque práctico 7th Ed. Roger S. Pressman, Capitulo 18, 19 y 20, y sesiones en ambientes con Esteban Silva, Renato Toasa, Henry Recalde y Ángel Mendoza, Integración SIGE-Moodle.	Metodología XP Fase Pruebas	Ejecución de pruebas de integración del proceso de matriculación entre sistemas SIGE- Moodle	Las pruebas permiten validar el correcto funcionamiento de la integración o a su vez mejorar y optimizar el proceso	SIGE , SQL Server, Moodle, PostgreSQL

Fuente: Elaboración propia, autores Diego Atiaja, José Fabre, Rubén Villalba

CAPITULO III: IMPLEMENTACIÓN

3.1 Plan de implementación

Tabla 9. Cronograma de Implementación

NOMBRE DE TAREA	DURACIÓN	COMIENZO	FIN	PREDECESORAS	NOMBRES DE LOS RECURSOS
Validación de Prerrequisitos	1 día	26/2/2021	26/2/2021		Equipo Desarrollo / Esteban Silva
Instalación de Driver ODBC Postgres	1 día	26/2/2021	26/2/2021	1	Equipo Desarrollo / Esteban Silva
Creación de Linked Server Posgrado	1 día	26/2/2021	26/2/2021	2	Equipo Desarrollo / Esteban Silva
Creación de Linked Server Pregrado	1 día	1/3/2021	1/3/2021	3	Equipo Desarrollo / Esteban Silva
Despliegue de Store Procedures en SqlServer	1 día	1/3/2021	1/3/2021	4	Equipo Desarrollo / Esteban Silva
Ejecución de pruebas	10 días	1/3/2021	12/3/2021	5	Equipo Desarrollo / Esteban Silva - Ángelo Mendoza
Matriculación Posgrado y Pregrado	1 día	1/3/2021	1/3/2021		Equipo Desarrollo / Esteban Silva - Ángelo Mendoza
Cambio de Paralelo Posgrado y Pregrado	1 día	1/3/2021	1/3/2021	7	Equipo Desarrollo / Esteban Silva - Ángelo Mendoza
Eliminación de matrícula Posgrado y Pregrado	1 día	1/3/2021	1/3/2021	8	Equipo Desarrollo / Esteban Silva - Ángelo Mendoza
Correcciones	3 días	2/3/2021	4/3/2021	9	Equipo Desarrollo
Validaciones Finales	1 día	5/3/2021	5/3/2021	10	Equipo Desarrollo / Esteban Silva - Ángelo Mendoza
Paso a Producción	1 día	8/3/2021	8/3/2021	11	Equipo Desarrollo / Esteban Silva - Ángelo Mendoza
Cierre	4 días	9/3/2021	12/3/2021	12	Equipo Desarrollo /

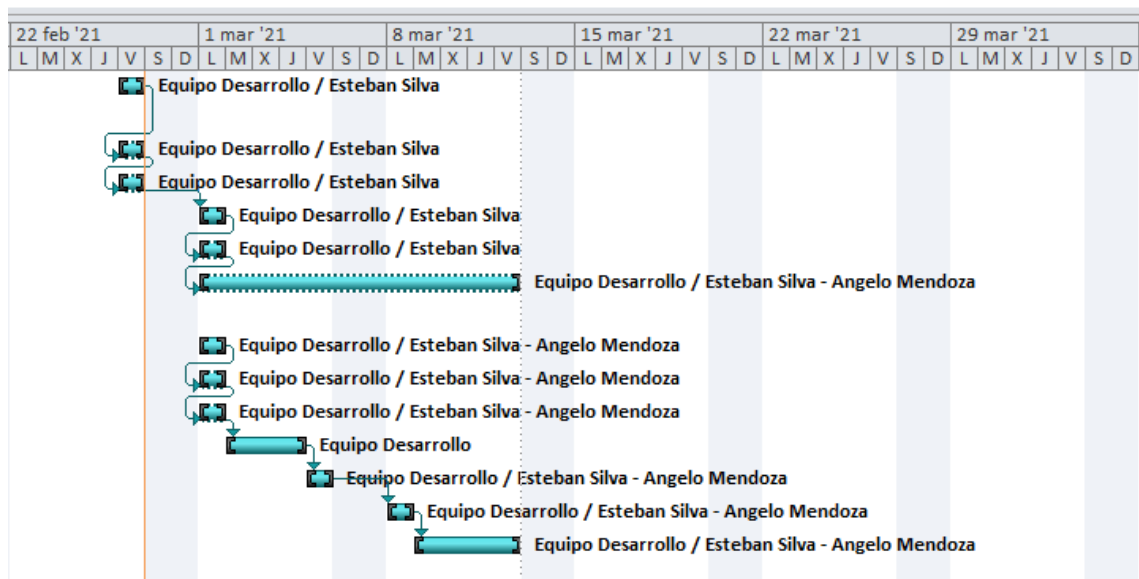


Figura 16. Cronograma de Implementación

3.2 Requerimientos

Dentro de los requerimientos mínimos para un correcto funcionamiento para el proceso de Integración SIGE-EVA

3.2.1. Requerimientos de software

- PostgreSQL 9.5.24 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609, 64-bit
- Microsoft SQLServer 2016
- ODBC Postgres psqldb_12_00_0000
- Visual Studio 2017

3.2.2. Requerimientos de Hardware

- Servidor HP Proliant DL 180
- Procesador Xeon E5-2600
- HP Dynamic Smart Array B140i estándar
- Memoria RAM DDR4, 2133 Mhz

3.3 Manual Técnico

2.3.3. Manual de instalación

Presentación

Esta guía explica cómo se debe proceder para la instalación de los componentes para una correcta configuración del plan de tesis.

Se ha estructurado este manual en secciones para una correcta explicación de los pasos que hay que seguir para lograr la comunicación entre SQL Server y PostgreSQL, así como la instalación de los Stored Procedure en ambiente de la plataforma SIGE.

El presente documento se ha dividido en las siguientes secciones:

- 1.1. **Instalación y configuración de Linkend Server:** Detalle al proceso de instalación de la comunicación entre SQL Server 2016 y PostgreSQL.
- 1.2. **Instalación de Stored Procedure en SQL Server:** Detalle al proceso de instalación de los nuevos procedimientos para el funcionamiento de la nueva funcionalidad de matriculación.

1.1. Instalación Y Configuración De Linkend Server

1.1.1. Instalación De ODBC De Postgres

Como paso previo a la instalación de ODBC de PostgreSQL es necesario descargar el driver correspondiente de la página oficial del fabricante e instalar en el Servidor de SQL Server 2016, para lo cual se procede a navegar en el explorador web de su preferencia a la siguiente ruta. <https://www.postgresql.org/ftp/odbc/versions/msi/> el archivo con el siguiente nombre `psqlodbc_12_00_0000.zip`

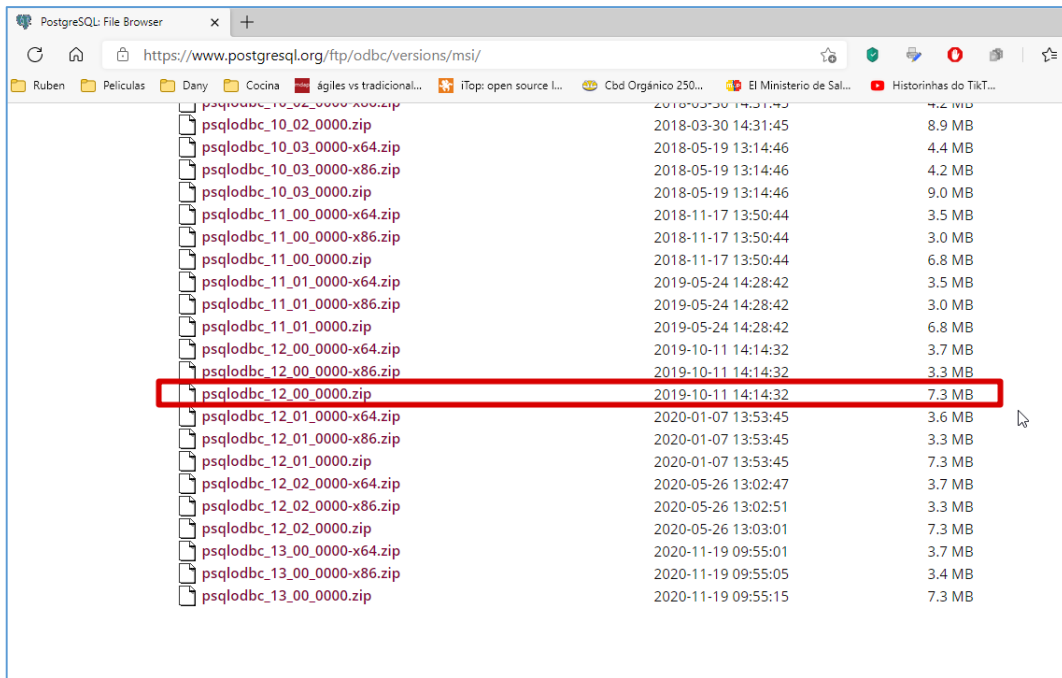


Figura 17. Versión de archivo de odbc a descargarse del sitio oficial de PostgreSQL

Cabe recalcar que, si se tiene una preferencia por algunas de las arquitecturas a utilizar, también existe los archivos correspondientes a la misma, pero el archivo que se descarga se considera las dos arquitecturas para una correcta instalación.

Procede a descomprimir el archivo Zip en la ruta de preferencia, para este caso en particular se procede a bajar en la ruta D: de nuestra máquina.

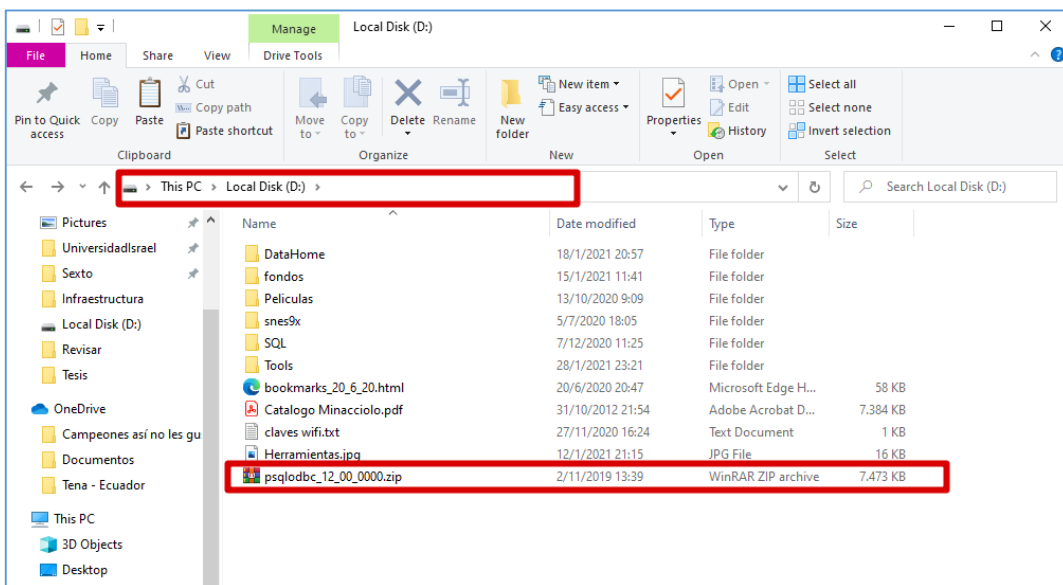


Figura 18. Ruta donde se almaceno la descarga del odbc del sitio oficial de PostgreSQL

Ejecutar el archivo para iniciar la instalación del controlador.

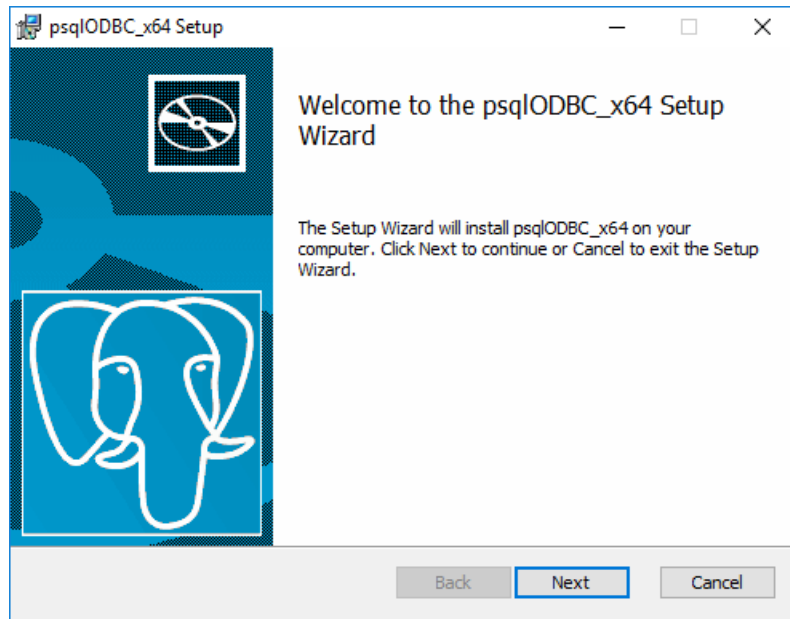


Figura 19. Pantalla de bienvenida del instalador de ODBC para Windows.

Clic en el botón siguiente “Next” para continuar con la instalación

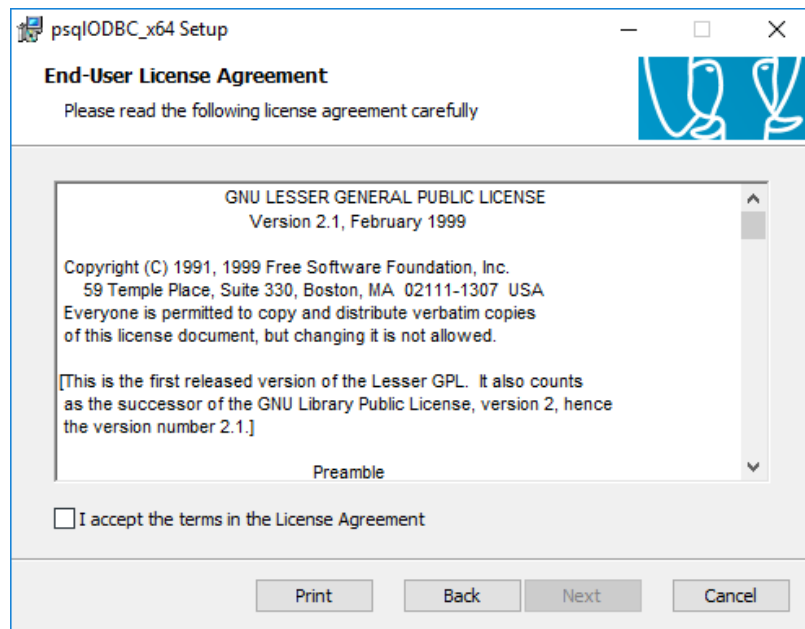


Figura 20. Pantalla de licenciamiento del instalador de ODBC para Windows.

Seleccionar aceptar los terminos y clic en el botón siguiente “Next” para continuar con la configuración.

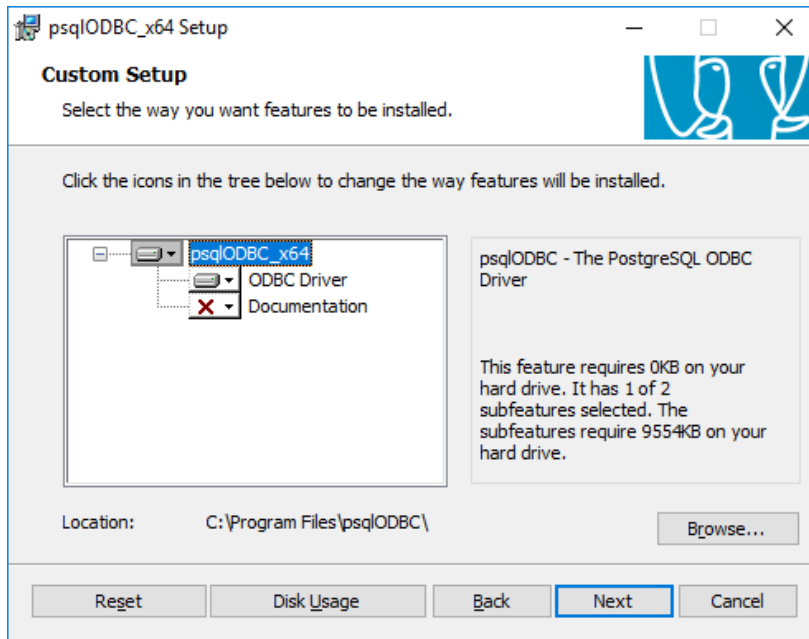


Figura 21. Pantalla de productos que se pueden seleccionar para la instalación del ODBC para Windows.

Escoger la versión de ODBC a instalar y clic en el botón siguiente “Next”

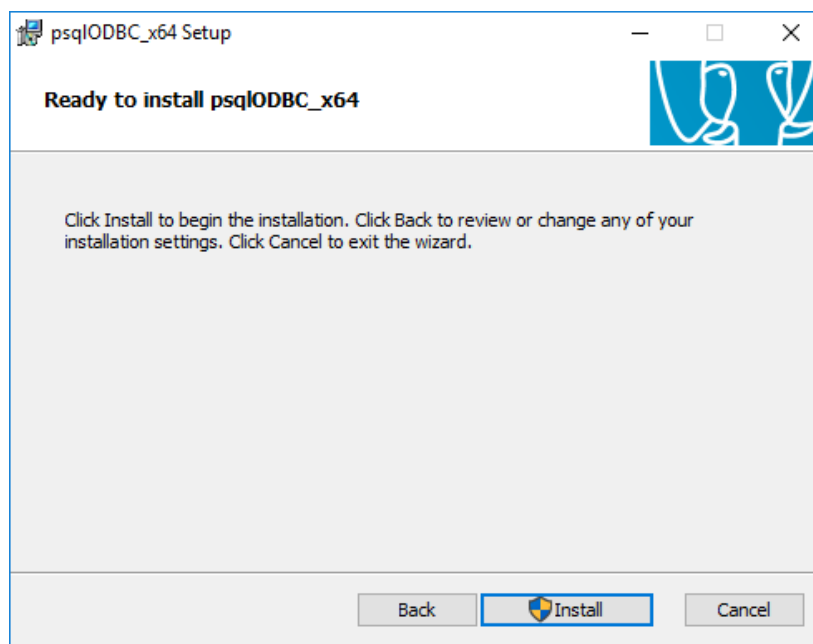


Figura 22. Pantalla de advertencia previo a la instalación final del ODBC para Windows.

Para que la instalación tenga curso es necesario dar clic en el botón instalar “Install”

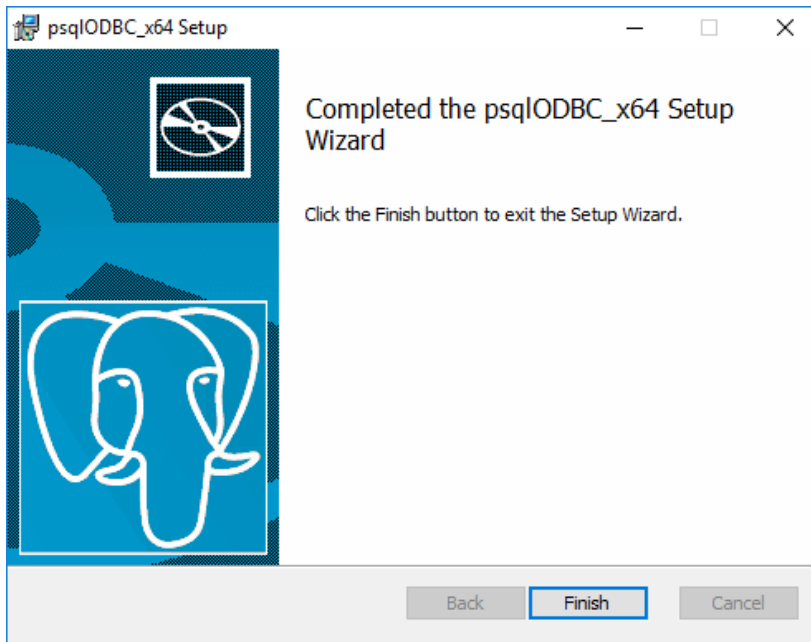


Figura 23. Pantalla de finalización de instalación de ODBC para Windows.

Clic en el botón en finalizar “Finish”.

Para confirmar la instalación del ODBC de PostgreSQL digitar el comando ODBC y seleccionar la versión dependiendo de la arquitectura del SO del Servidor, para este caso en particular seleccionar la versión de 64 bit.

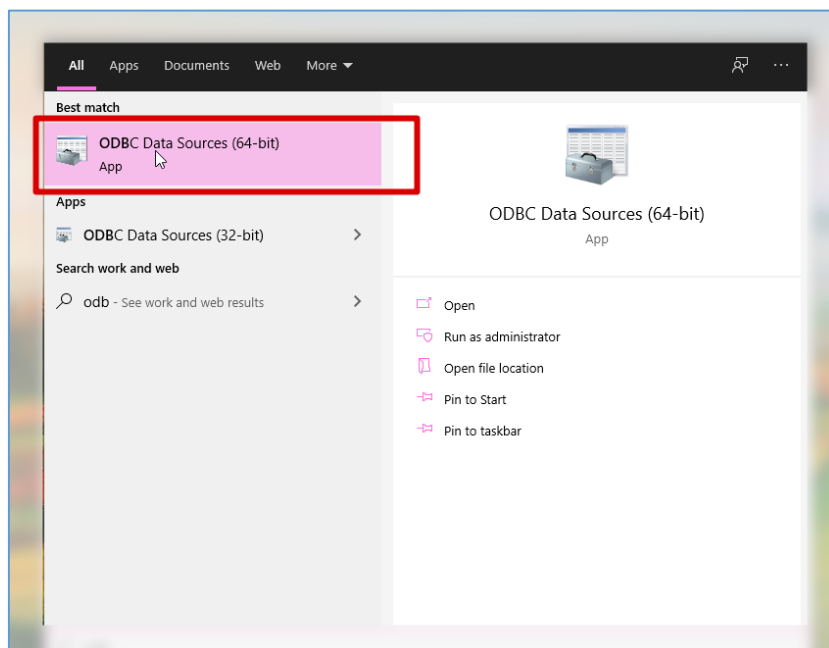


Figura 24. Pantalla de despliegue de la versión del OBDC Data Source.

Proceder a añadir un nuevo DataSource del tipo DSN para lo cual agregar “add”

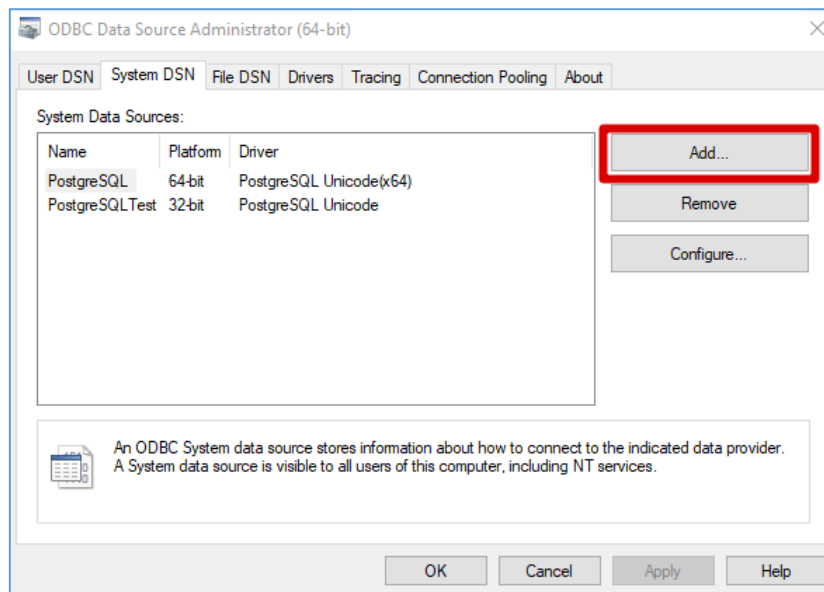


Figura 25. Pantalla de Systema Data source para agregar un nuevo DSN

Es importante escoger la opción PostgreSQL Unicode(x64) para el nombre

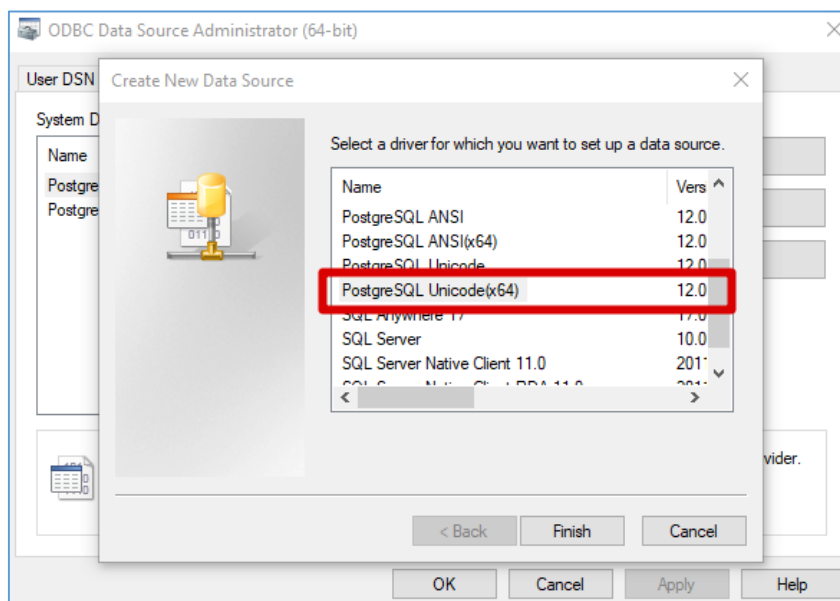


Figura 26. Pantalla de selección de Proveedor del DSN.

Crear la configuración del DSN con las siguientes opciones dependiendo del ambiente al que se está accediendo.

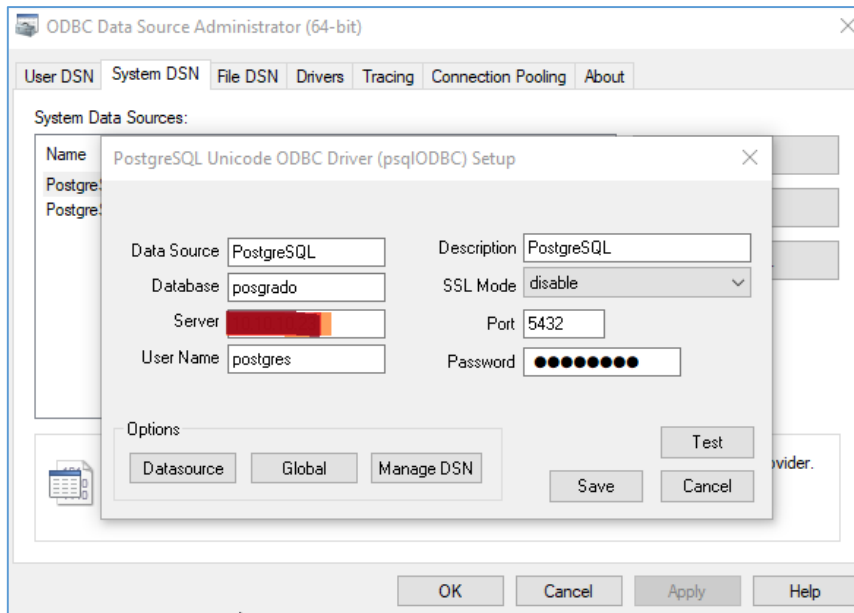


Figura 27. Pantalla de configuración DSN.

DataSource : PostgreSQL

Descripción : PostgreSQL

Database : Posgrado

SSL Mode : Disable

Server : IP del Servidor (ejm: 127.0.0.1)

Port : 5432

User Name : Postgres

Password : XXXXXXXXXXXX

Probar la conexión mediante el uso del botón “Test”

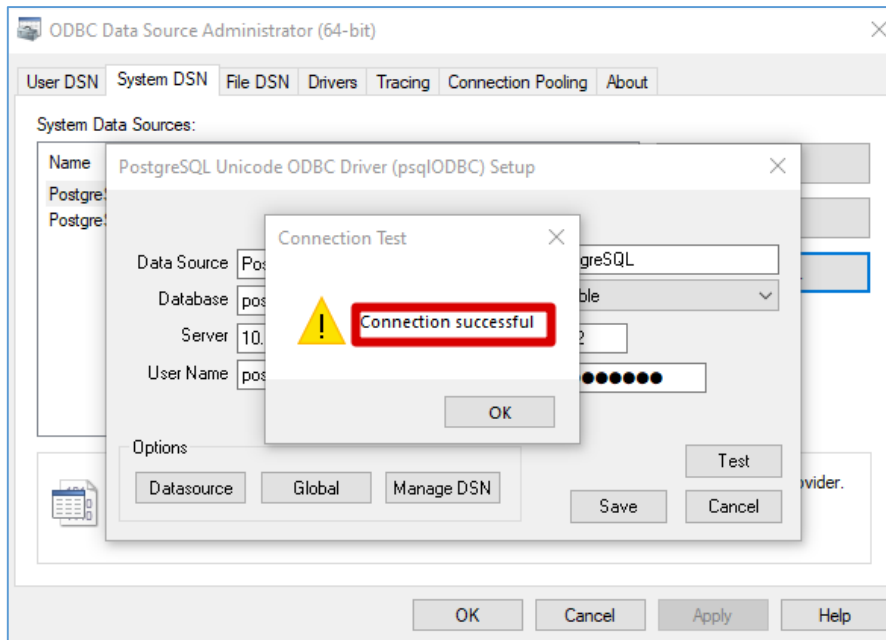


Figura 28. Pantalla de test conexión después de la configuración DSN.

1.1.2. Configuración Linked Server

Para la configuración del Linked Server es necesario la apertura del Microsoft SQL Server Management Studio con DBMS de SQL Server 2016 y conectar al servidor. Abrir la opción que se encuentra en la siguiente ruta: Server Objects -> Likend Server

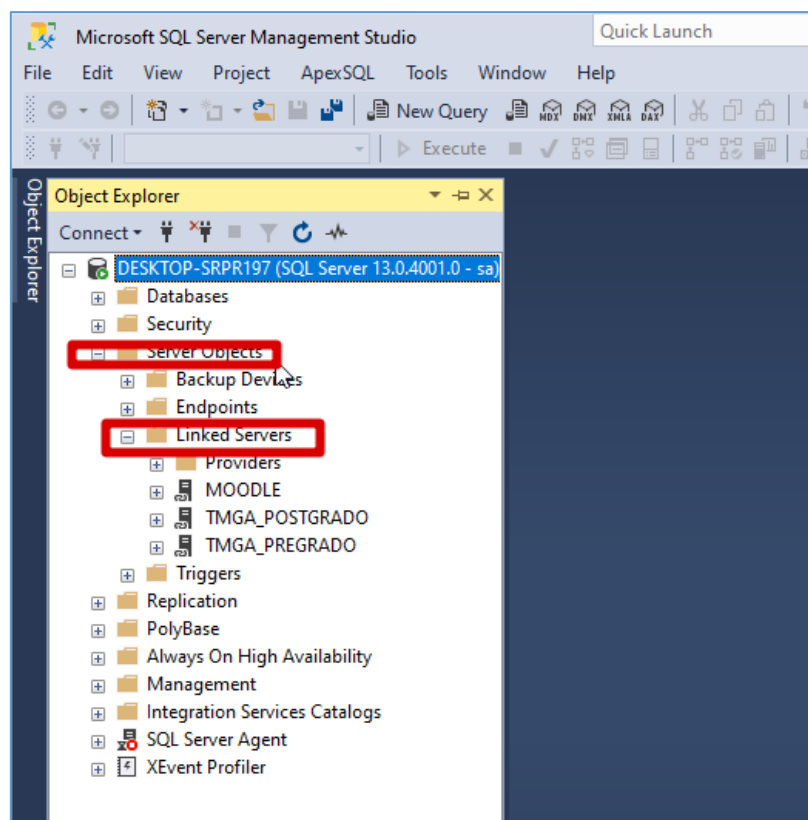


Figura 29. Pantalla de Visual Studio para la configuración del Linked Server en SSMS

Se procede con la creación de un nuevo Linked Server para lo cual se realiza clic derecho sobre el Linked Servers y al menú New Linked Server.

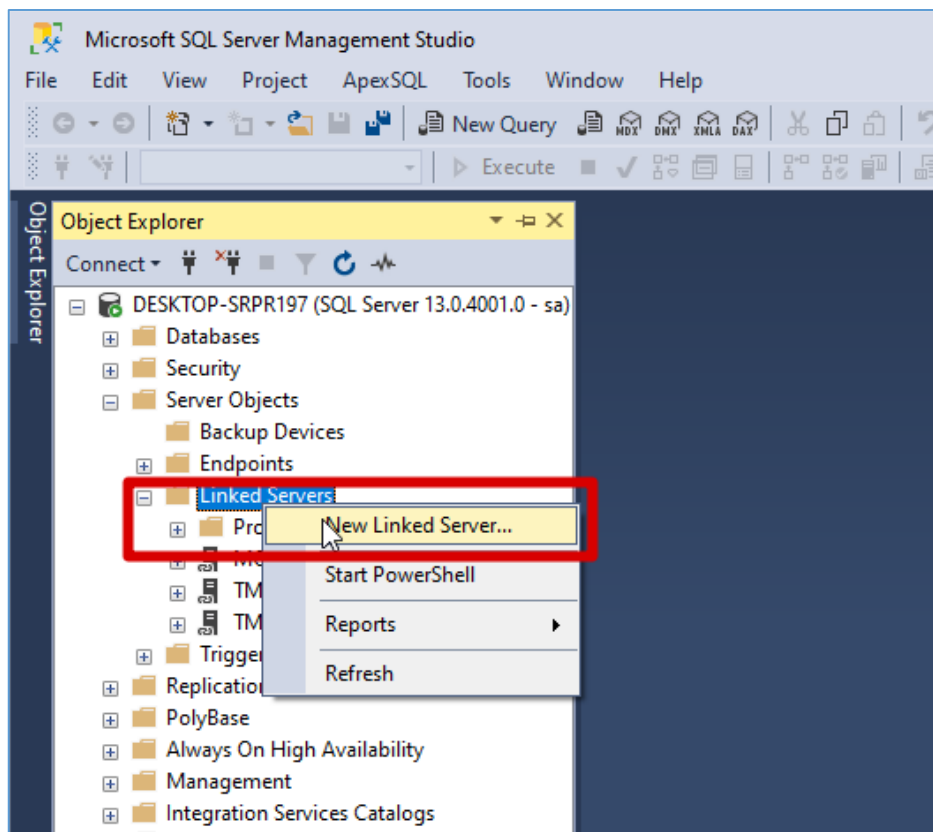


Figura 30. Pantalla de menú para agregar un nuevo servidor.

En el nuevo menú en la pantalla seleccionada se llena la siguiente información.

Linked Server :	MOODLE
------------------------	--------

Provider :	Microsoft OLE DB Provider for ODBC Drivers
-------------------	---

Product name :	PostgreSQL
-----------------------	------------

Data source :	PostgreSQL (Nombre del DataSource configurado en ODBC Data Source del sistema.
----------------------	--

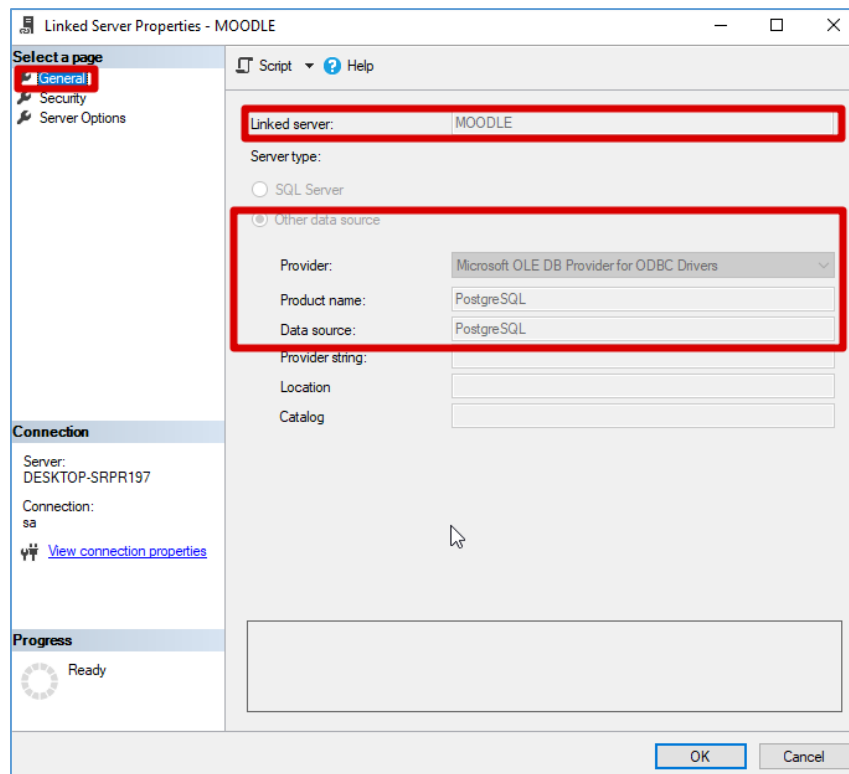


Figura 31. Pantalla de propiedades de Linked Server para la conexión de Moodle - General

En la opción de **Security** llenar la opción **“Be made using the security contex”** y digitar las credenciales que fueron usadas al configurar el Data source en el sistema.

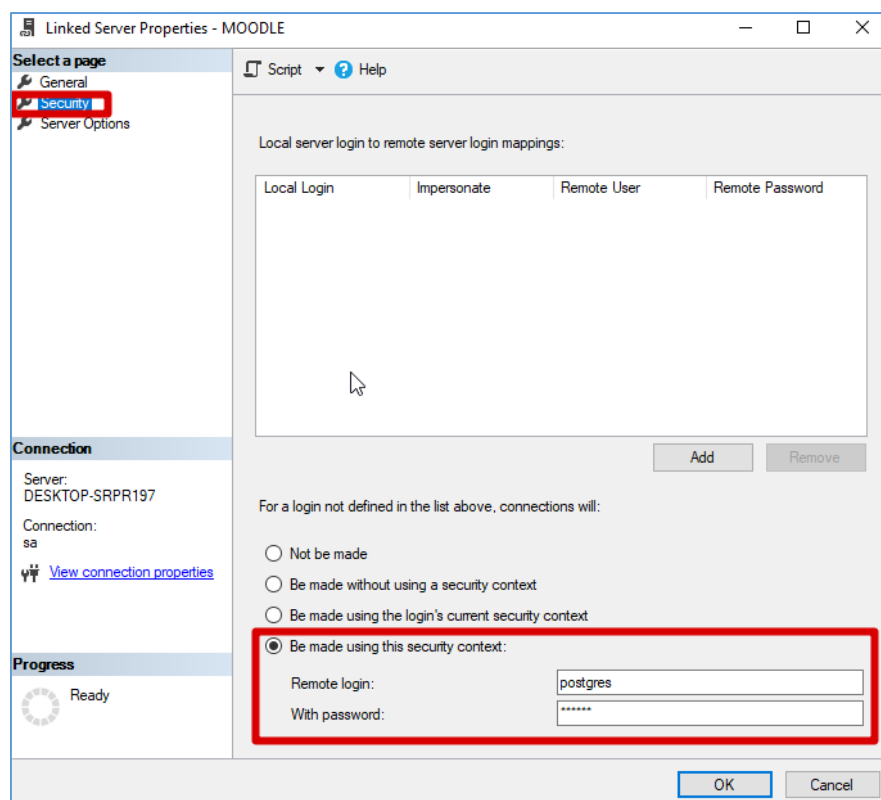


Figura 32. Pantalla de propiedades de Linked Server para la conexión de Moodle - Seguridad

Se procede a la opción de **Server Options** que es necesario para la instalación y procedemos activas las opciones de **RPC** y **RPC Out** a **true**

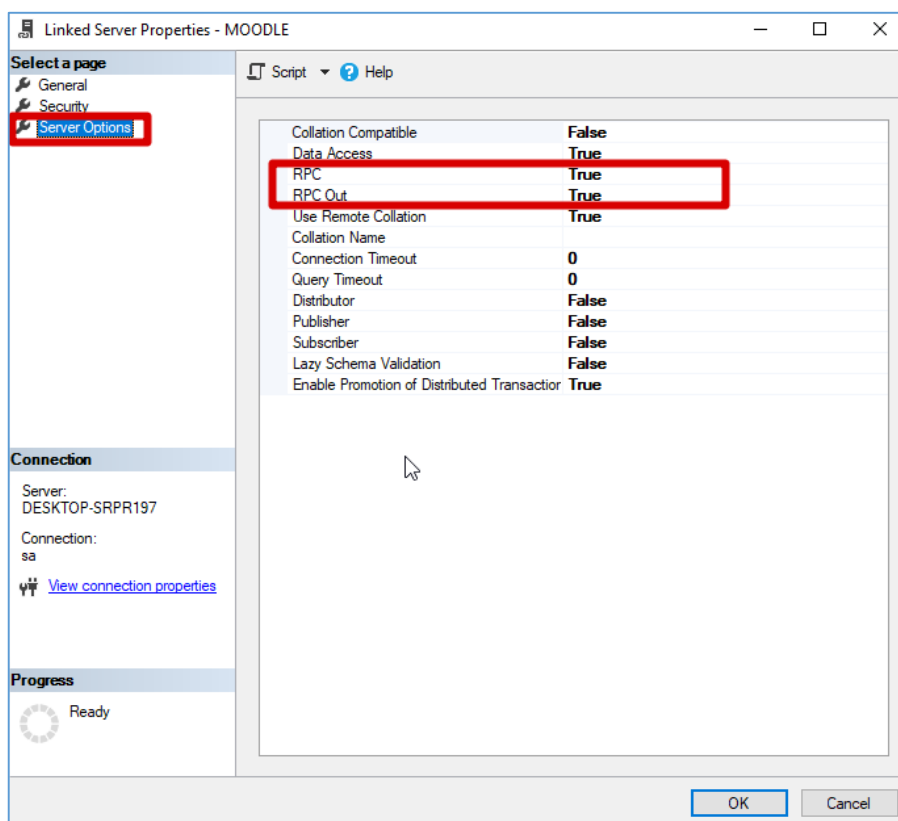


Figura 33. Pantalla de propiedades de Linked Server para la conexión de Moodle – Opciones servidor

Para confirmar la correcta configuración es necesario realizar el test correspondiente de la configuración del Linked Server creado.

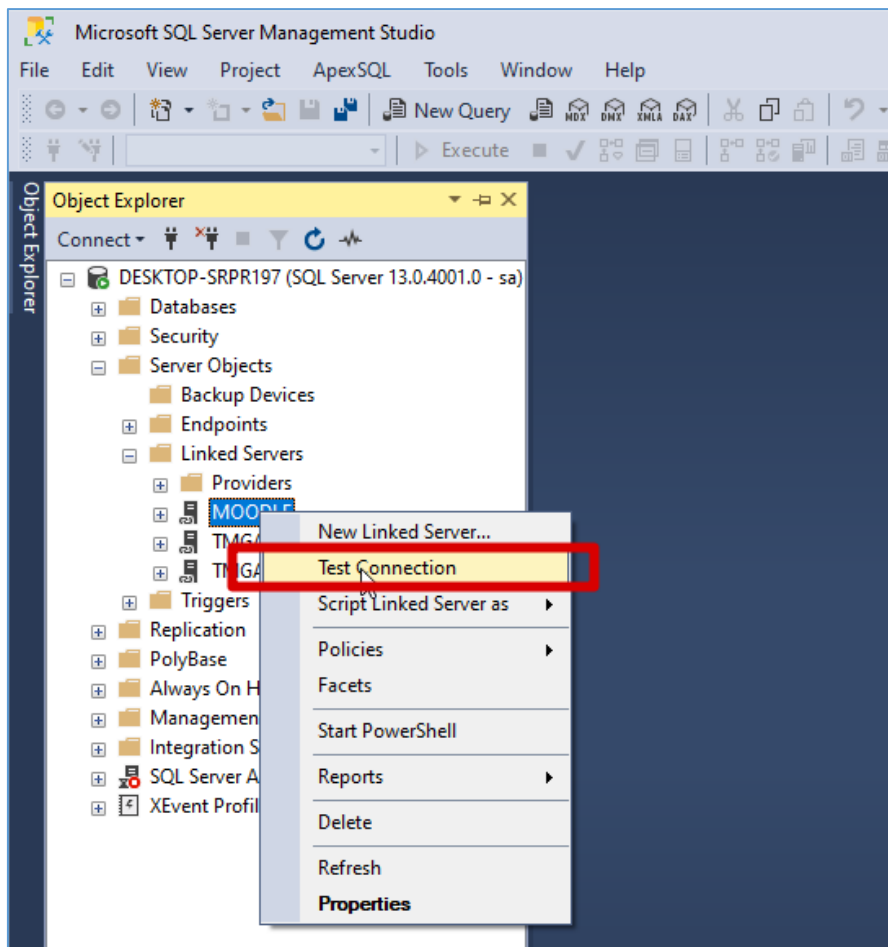


Figura 34. Pantalla de Prueba de la conexión de Moodle usando el Linked Server.

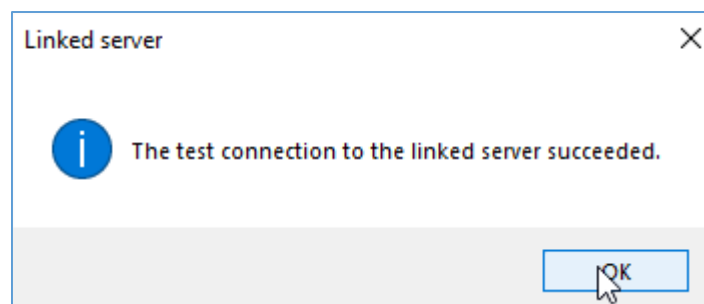


Figura 35. Pantalla de resultado de la conexión de Moodle.

1.2. Instalación De Store Procedure En Sql Server

Para continuar con la instalación de la configuración hay que compilar en ambiente de SQL Server 2016 los siguientes sps en el orden abajo descrito.

1. SPI_MATRICULA_ESTUDIANTE_MOODLE.sql

```

10 ALTER PROCEDURE [dbo].[SPI_MATRICULA_ESTUDIANTE_MOODLE](
11     @ID_CEDULA VARCHAR(13),
12     @ID_PLANIFICACION INT,
13     @TIPO INT
14 ) AS
15 DECLARE
16     @mdl_userid VARCHAR (10),
17     @mdl_context VARCHAR (10),
18     @mdl_roleid varchar(10),
19     @mdl_courseid varchar(100),
20     @mdl_enrolid varchar(10),
21     @count int;
22
23 BEGIN
24
25 IF(@TIPO = 1 )
26 BEGIN
27     ---SE OBTIENE EL ID DEL USUARIO EN EL SISTEMA MOODLE
28     select @mdl_userid= A.ID, @count= count(1)
29     from MOODLE_PRE.[posgrado].[public].mdl_user A
30     where SUBSTRING(A.USERNAME, 2, 10)=@ID_CEDULA
31     GROUP BY A.ID;
32     IF(@count > 0 )
33     BEGIN
34         SELECT
35             @count= count(1)
36             FROM MOODLE_PRE.[posgrado].[public].mdl_course
37             INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_context ON mdl_context.instanceid = mdl_course.id
38             INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role_assignments ON mdl_context.id = mdl_role_assignments.contextid
39             and mdl_role_assignments.userid =@mdl_userid
40             INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role ON mdl_role.id = mdl_role_assignments.roleid

```

Figura 36. Pantalla de ejecución del store procedure matricula estudiante moodle.

2. SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE.sql

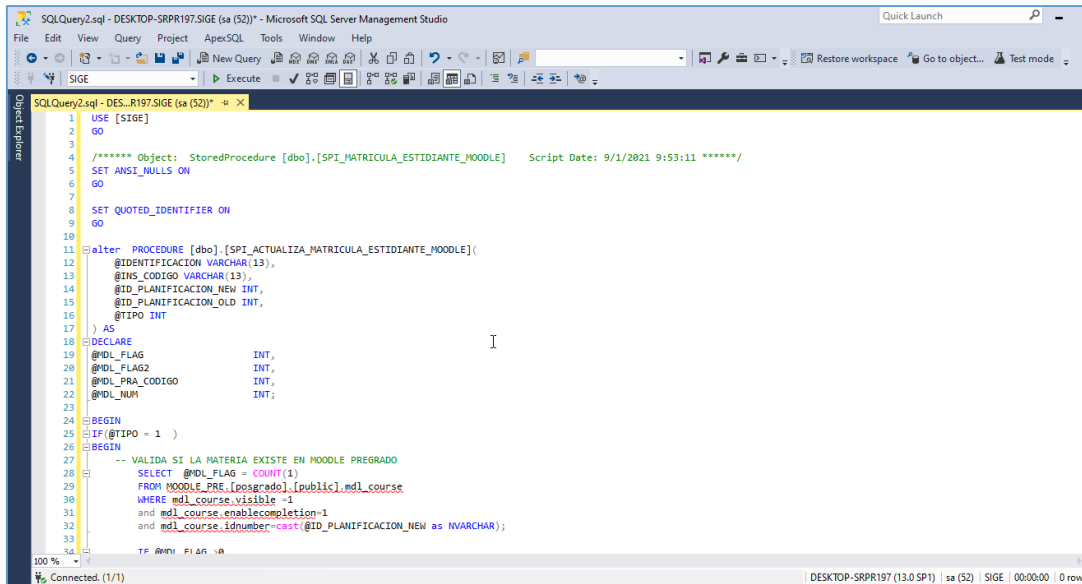
```

10
11
12 ALTER PROCEDURE [dbo].[SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE](
13     @ID_CEDULA VARCHAR(13),
14     @ID_PLANIFICACION INT,
15     @TIPO INT
16 ) AS
17 DECLARE
18     @mdl_userid VARCHAR (10),
19     @mdl_user_enrol_id varchar(10),
20     @mdl_role_asig_id varchar(10),
21     @count int;
22
23 BEGIN
24
25 IF(@TIPO = 1 )
26 BEGIN
27     ---SE OBTIENE EL ID DEL USUARIO EN EL SISTEMA MOODLE
28     select @mdl_userid= A.ID, @count= count(1)
29     from MOODLE_PRE.[posgrado].[public].mdl_user A
30     where SUBSTRING(A.USERNAME, 2, 10)=@ID_CEDULA
31     GROUP BY A.ID;
32     IF(@count > 0 )
33     BEGIN
34         ---Valida si el usuario esta matriculado en moodle
35         SELECT
36             @mdl_role_asig_id = mdl_role_assignments.id,
37             @mdl_user_enrol_id = mdl_user_enrolments.id,
38             @count= count(1)
39             FROM MOODLE_PRE.[posgrado].[public].mdl_course
40             INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_context ON mdl_context.instanceid = mdl_course.id
41             INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role_assignments ON mdl_context.id = mdl_role_assignments.contextid
42             and mdl_role_assignments.userid =@mdl_userid
43             INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role ON mdl_role.id = mdl_role_assignments.roleid

```

Figura 37. Pantalla de ejecución del store procedure eliminación matricula estudiante moodle.

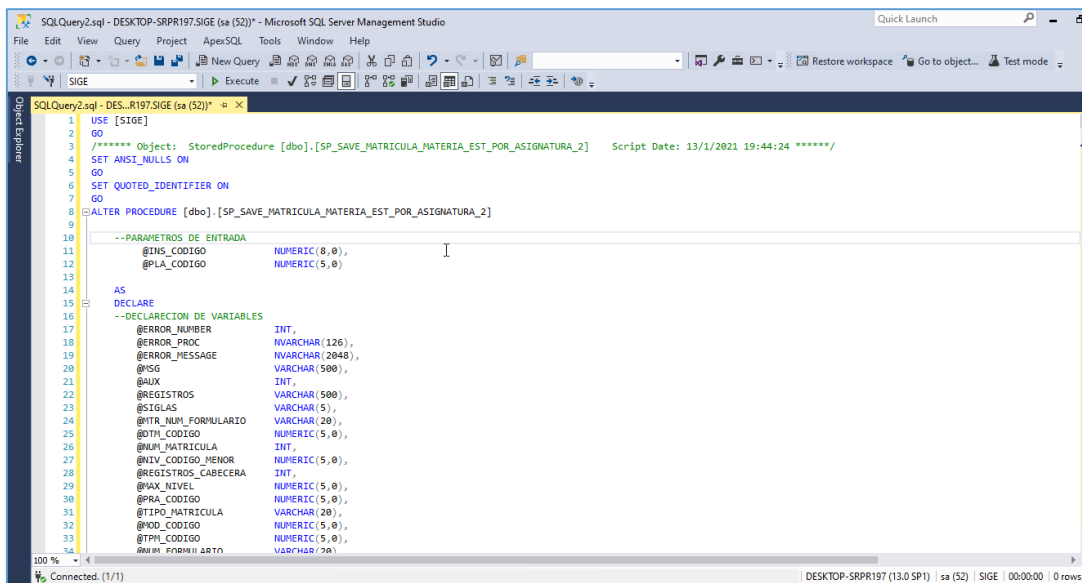
3. SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE.sql



```
1 USE [SIGE]
2 GO
3 /***** Object: StoredProcedure [dbo].[SPI_MATRICULA_ESTUDIANTE_MOODLE]    Script Date: 9/1/2021 9:53:11 *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8 ALTER PROCEDURE [dbo].[SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE]
9 AS
10
11 @IDENTIFICACION VARCHAR(13),
12 @INS_CODIGO VARCHAR(13),
13 @ID_PLANIFICACION_NEW INT,
14 @ID_PLANIFICACION_OLD INT,
15 @TIPO INT
16
17 ) AS
18 DECLARE
19 @MDL_FLAG INT,
20 @MDL_FLAG2 INT,
21 @MDL_PRA_CODIGO INT,
22 @MDL_NUM INT;
23
24 BEGIN
25 IF (@TIPO = 1 )
26 BEGIN
27 -- VALIDA SI LA MATERIA EXISTE EN MOODLE PREGRADO
28 SELECT @MDL_FLAG = COUNT(1)
29 FROM MOODLE_PREGRADO.[public].mdl_course
30 WHERE mdl_course.visible = 1
31 and mdl_course.enablecompletion = 1
32 and mdl_course.idnumber = cast(@ID_PLANIFICACION_NEW as NVARCHAR);
33
34
```

Figura 38. Pantalla de ejecución del store procedure actualizar estudiante moodle.

4. SP_SAVE_MATRICULA_MATERIA_EST_POR_ASIGNATURA_2.sql



```
1 USE [SIGE]
2 GO
3 /***** Object: StoredProcedure [dbo].[SP_SAVE_MATRICULA_MATERIA_EST_POR_ASIGNATURA_2]    Script Date: 13/1/2021 19:44:24 *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8 ALTER PROCEDURE [dbo].[SP_SAVE_MATRICULA_MATERIA_EST_POR_ASIGNATURA_2]
9
10 --PARAMETROS DE ENTRADA
11 @INS_CODIGO NUMERIC(8,0),
12 @PLA_CODIGO NUMERIC(5,0)
13
14 AS
15 DECLARE
16 --DECLARACION DE VARIABLES
17 @ERROR_NUMBER INT,
18 @ERROR_PROC NVARCHAR(126),
19 @ERROR_MESSAGE NVARCHAR(2048),
20 @MSG VARCHAR(500),
21 @AUX INT,
22 @REGISTROS VARCHAR(500),
23 @SIGLAS VARCHAR(5),
24 @MTR_NUM_FORMULARIO VARCHAR(20),
25 @OTM_CODIGO NUMERIC(5,0),
26 @NUM_MATRICULA INT,
27 @NIV_CODIGO_MENOR NUMERIC(5,0),
28 @REGISTROS_CABECERA INT,
29 @MAX_NIVEL NUMERIC(5,0),
30 @PRA_CODIGO NUMERIC(5,0),
31 @TIPO_MATRICULA VARCHAR(20),
32 @MOD_CODIGO NUMERIC(5,0),
33 @TPM_CODIGO NUMERIC(5,0),
34 @NUM_FORMULARIO VARCHAR(20)
```

Figura 39. Pantalla de ejecución del store procedure salvar matricula del estudiante sige.

5. SP_CAMBIO_PARALELO_ESTUDIANTE.sql

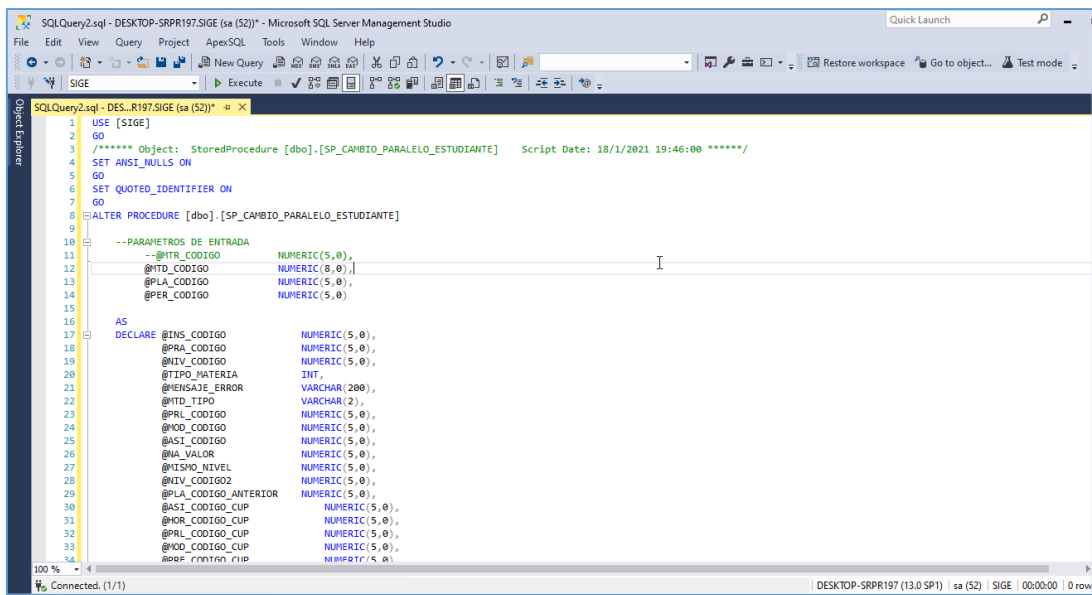


Figura 40. Pantalla de ejecución del store procedure de cambio de paralelo estudiante.

Este proceso permitirá instalar 3 stored procedure nuevos y 2 modificados que permitirán realizar el enlace de la plataforma SIGE con el EVA.

CONCLUSIONES

- La presente investigación ha permitido definir y exponer los fundamentos técnicos en el proceso de matriculación de los estudiantes en las plataformas SIGE y EVA(Moodle), donde se analizó las posibles formas de optimizar siendo como base para el posterior desarrollo e implementación de la solución.
- Al analizar el proceso actual de matriculación de los estudiantes de la Universidad Israel, se pudo observar que, si bien existen procesos establecidos para cada plataforma, no se encuentran integrados entre plataformas, por lo que el presente diseño podrá solventar este requerimiento presentado.
- Para el proceso de desarrollo de software es un factor fundamental la capacitación previa de las herramientas y giro de negocio
- Por medio del uso de la investigación y sus métodos se determinó la percepción de los usuarios referente al proceso de matriculación en las plataformas de la Universidad Israel.
- La implementación de una integración a nivel de base de datos entre los sistemas SIGE y EVA(Moodle) de la Universidad Israel, mejoró el proceso de matriculación de los estudiantes proporcionando muchas ventajas, una de ellas el ahorro de tiempo, mejora de la gestión, entre otras.

RECOMENDACIONES

- Es fundamental que tanto para la implementación de la presente propuesta, así como para procesos futuros se deba considerar todos los fundamentos técnicos, normativas estándares y normativas que manejan en el área de desarrollo de Universidad Israel.
- Los procesos académicos que contienen las plataformas SIGE y EVA(Moodle) similares, se deberían integrar a través de desarrollos que consideren siempre las normativas institucionales y los fundamentos técnicos.
- Se recomienda que para la integración entre las plataformas SIGE y EVA(Moodle), se parta de la conceptualización y necesidad de los diferentes procesos.
- Utilice distintas técnicas de investigación como la observación, encuesta y entrevista ya que permitirán aclarar el objetivo para poder dar pasos seguros en una solución informática.
- Se recomienda que para futuras integraciones entre las plataformas SIGE y EVA(Moodle), se considere el uso del Linked Server expuestos en este trabajo, ya que la comunicación entre los DBMS(SQLServer-Postgres), será mucho más fácil.
- Se recomienda que para futuros desarrollos de Store Procedures, se considere las Best Practices de Microsoft

BIBLIOGRAFÍA

Baldeón Egas, Paúl Francisco, Orozco Molina, Hernán Santiago, (2018), MÓDULO DE COEVALUACIÓN DEL PROFESOR, BASADO EN EL SIGE DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL, MEDIANTE LA METODOLOGÍA EXTREME PROGRAMMING, Quito, <http://repositorio.uisrael.edu.ec/handle/47000/1670>

Ceballos, K. (04 de 05 de 2015). *Metodología de Desarrollo Ágil: XP y Scrum, Ingeniería del Software*. Recuperado el 27 de 10 de 2020, de <https://ingsoftwarekarlacevallos.wordpress.com/2015/05/08/metodologia-de-desarrollo-agil-xp-y-scrum/>

CTMA Consultores 15 (2018). *¿Qué son los Sistemas de Gestión Integrados y por qué los necesitas?*, Recupeado el 20 de 01 de 2018, de <https://ctmaconsultores.com/sistemas-gestion-integrados/>

Grau, J. L. (08 de 06 de 2016). *La Metodología XP, PROAGILIST*. Recuperado el 26 de 10 de 2020, de <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/>

María Estela Raffino. (2020). *Base de datos*, Argentina : Concepto.de, Recuperado el 24 de 06 de 2020, de <https://concepto.de/base-de-datos/>

Marko Zivkovic. (2018). *Cómo crear y configurar un servidor enlazado en SQL Server Management Studio*, Chapel Hill, US : SQLShack, Recuperado el 16 de 04 de 2018, de <https://www.sqlshack.com/es/como-crear-y-configurar-un-servidor-enlazado-en-sql-server-management-studio/>

Martínez Mosquera, Silvia Diana, Baldeón Egas, Paúl Francisco, Oña Llumiquinga, José Pablo (2018), *Módulo de gestión en la Unidad de Idiomas extranjero basado en el SIGE-UI bajo la norma ISO 27001*, Quito, <http://repositorio.uisrael.edu.ec/handle/47000/1817>

Microsoft. (2017). "Microsoft Open Database Connectivity (ODBC)", United States: Microsoft, Recuperado el 19 de 11 de 2017, de <https://docs.microsoft.com/en-us/sql/odbc/microsoft-open-database-connectivity-odbc?view=sql-server-ver15>

Microsoft. (2019). "Welcome to the Visual Studio IDE", United Stated: Microsoft, Recuperado el 19 de 03 de 2019, de <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>

Microsoft. (20 de 01 de 2020). *Msn Notas*. Recuperado el 23 de 10 de 2020, de <https://www.msn.com/es-cl/noticias/microsoftstore/%C2%BFqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAAnLZL9>

Moodle. (7 de noviembre del 2020). *Acerca de Moodle*. Recuperado el 7 de noviembre del 2020, de https://docs.moodle.org/all/es/Acerca_de_Moodle

Moodle (2020), *Acerca de Moodle*, Moodle.Net Recuperado, el 7 de 11 de 2020, de [Acerca de Moodle - MoodleDocs](#)

Omar Stalin Altamirano Altamirano, I. R. (2018). *Desarrollo Del Módulo De Producción Para Las Empresas Dedicadas Al Proceso De Corte De Papel*. Quito: <http://repositorio.uisrael.edu.ec/handle/47000/1657>

Pressman R. (2012). *Ingeniería del Software Un Enfoque Práctico*. 7 Ed. McGraw Hill.

Postgres. (2020). "What is PostgreSQL?", The PostgreSQL Global Development Group, Recuperado el 12 de 11 de 2020, de <https://www.postgresql.org/about/>

PuTTY. (2017). "Introduction to PuTTY", United States: PuTTY.org, Recuperado el 27 de 01 de 2020, de <https://www.putty.org/>

Raffino, M. E. (2020 de 06 de 2020). *Método Deductivo*. Recuperado el 27 de 10 de 2020, de <https://concepto.de/metodo-deductivo-2/#ixzz6boy3bsml>

Sommerville I. (2011). *Ingeniería del Software*. Addison-Wesley.

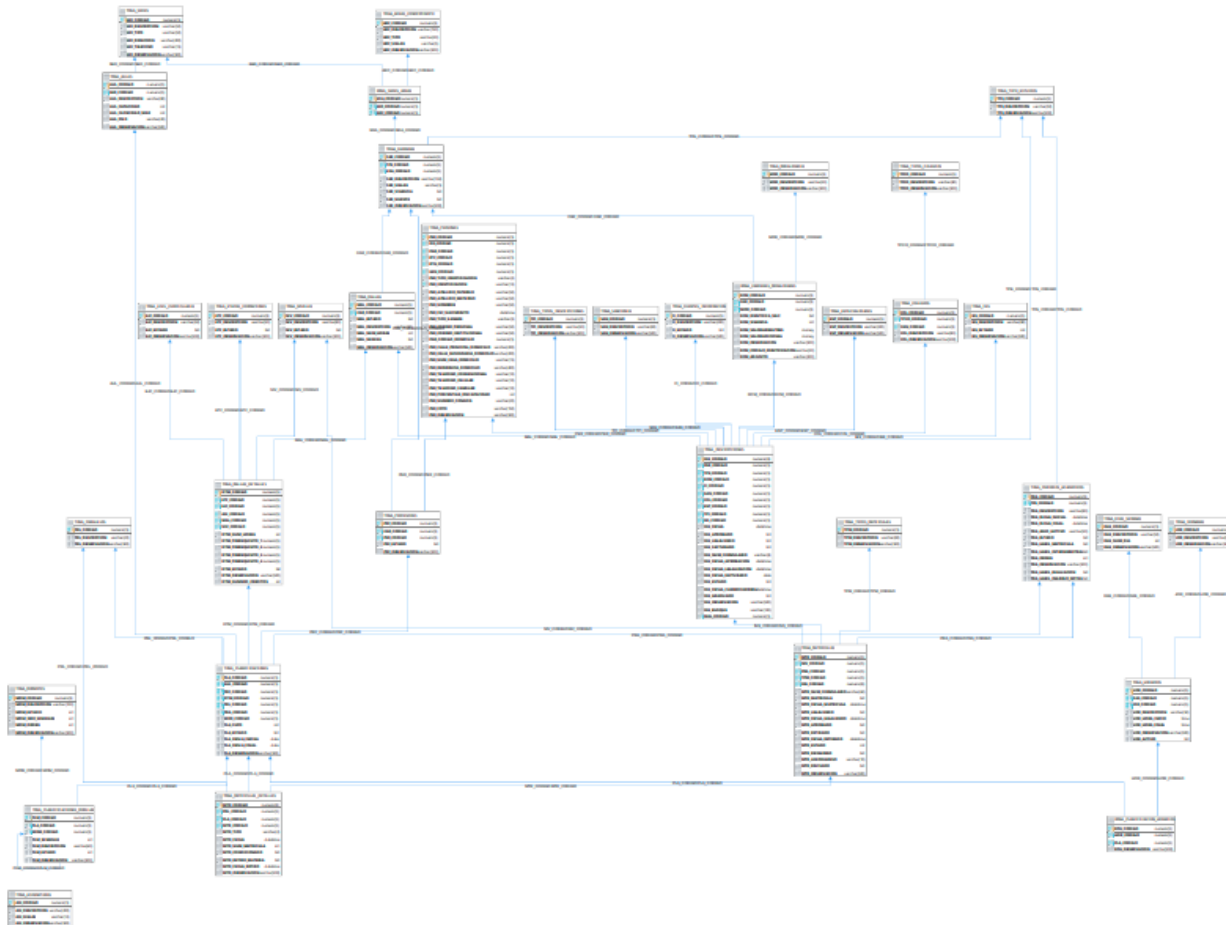
SQLServerTutorial.Net, (2020). "What is SQL Server", SQLServerTutorial.Net, Recuperado el 02 de 02 de 2020, de <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/>

Wikipedia. (10 de 2020). Microsoft SQL Server. Recuperado el 23 de 10 de 2020, de <https://es.wikipedia.org>

ANEXOS

ANEXO 1

MODELO ENTIDAD RELACION SQLSERVER(SIGE) QUE INTERVIENEN EN EL PROCESO DE MATRICULACION



ANEXO 2

MODELO ENTIDAD RELACION POSTGRESSQL(MOODLE) QUE INTERVIENEN EN EL PROCESO DE MATRICULACION

mdl_user	
id	bigint
auth	varchar(20)
confirmed	smallint
policyagreed	smallint
deleted	smallint
suspended	smallint
methodid	bigint
username	varchar(100)
password	varchar(255)
idnumber	varchar(255)
firstname	varchar(100)
lastname	varchar(100)
email	varchar(100)
emailstop	smallint
icq	varchar(15)
skype	varchar(50)
yahoo	varchar(50)
aim	varchar(50)
msn	varchar(50)
phone1	varchar(20)
phone2	varchar(20)
institution	varchar(255)
department	varchar(255)
address	varchar(255)
city	varchar(120)
country	varchar(2)
lang	varchar(30)
calendartype	varchar(30)
theme	varchar(50)
timezone	varchar(100)
firstaccess	bigint
lastaccess	bigint
lastlogin	bigint
currentlogin	bigint
lastip	varchar(45)
secret	varchar(15)
picture	bigint
url	varchar(255)
description	text
descriptionformat	smallint
mailformat	smallint
maildigest	smallint
maildisplay	smallint
autosubscribe	smallint
trackforums	smallint
timecreated	bigint
timemodified	bigint
trustbitmask	bigint
imagealt	varchar(255)
lastnamephonetic	varchar(255)

mdl_course	
id	bigint
category	bigint
sortorder	bigint
fullname	varchar(254)
shortname	varchar(255)
idnumber	varchar(100)
summary	text
summaryformat	smallint
format	varchar(21)
showgrades	smallint
newsitems	integer
startdate	bigint
enddate	bigint
relativedatesmode	smallint
marker	bigint
maxbytes	bigint
legacyfiles	smallint
showreports	smallint
visible	smallint
visibleold	smallint
groupmode	smallint
groupmodeforce	smallint
defaultgroupingid	bigint
lang	varchar(30)
calendartype	varchar(30)
theme	varchar(50)
timecreated	bigint
timemodified	bigint
requested	smallint
enablecompletion	smallint
completionnotify	smallint
cacherev	bigint

mdl_course_categories	
id	bigint
name	varchar(255)
idnumber	varchar(100)
description	text
descriptionformats	smallint
parent	bigint
sortorder	bigint
coursecount	bigint
visible	smallint
visibleold	smallint
timemodified	bigint
startdate	bigint
depth	bigint
path	varchar(255)
theme	varchar(50)

mdl_role_assignments	
id	bigint
roleid	bigint
contextid	bigint
userid	bigint
timemodified	bigint
modifier	bigint
component	varchar(100)
itemid	bigint
sortorder	bigint

mdl_user_enrolments	
id	bigint
status	bigint
enrolid	bigint
userid	bigint
timestart	bigint
timewend	bigint
modifierid	bigint
timecreated	bigint
timemodified	bigint

mdl_role	
id	bigint
name	varchar(255)
shortname	varchar(100)
description	text
sortorder	bigint
archetype	varchar(30)

mdl_context	
id	bigint
contextlevel	bigint
instanceid	bigint
path	varchar(255)
depth	smallint
locked	smallint

ANEXO 3

Store Procedure usados para el manejo de la implementación entre la plataforma SIGE - EVA

1. SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE.sql

```
USE [SIGE]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
    WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE'
)
    DROP PROCEDURE dbo.SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE
GO

CREATE PROCEDURE [dbo].[SPI_ACTUALIZA_MATRICULA_ESTUDIANTE_MOODLE] (
    @IDENTIFICACION VARCHAR(13),
    @INS_CODIGO VARCHAR(13),
    @ID_PLANIFICACION_NEW INT,
    @ID_PLANIFICACION_OLD INT,
    @TIPO INT
)
AS
DECLARE
    @MDL_FLAG INT,
    @MDL_FLAG2 INT,
    @MDL_PRA_CODIGO INT,
    @MDL_NUM INT;

BEGIN
    IF (@TIPO = 1 )
    BEGIN
        -- VALIDA SI LA MATERIA EXISTE EN MOODLE PREGRADO
        SELECT @MDL_FLAG = COUNT(1)
        FROM MOODLE_PRE.[posgrado].[public].mdl_course
        WHERE mdl_course.visible =1
        and mdl_course.enablecompletion=1
        and mdl_course.idnumber=cast(@ID_PLANIFICACION_NEW as NVARCHAR);

        IF @MDL_FLAG >0
        BEGIN

            IF @IDENTIFICACION IS NOT NULL AND @ID_PLANIFICACION_OLD IS NOT
            NULL AND @ID_PLANIFICACION_NEW IS NOT NULL
            BEGIN
                EXEC [SPI_ELIMINA_MATRICULA_ESTIDIANTE_MOODLE]
                @IDENTIFICACION, @ID_PLANIFICACION_OLD,1;
                EXEC [SPI_MATRICULA_ESTIDIANTE_MOODLE] @IDENTIFICACION,
                @ID_PLANIFICACION_NEW,1;

                PRINT 'PREGRADO: Se realizo el cambio de paralelo...';
            END
            ELSE
            BEGIN
                PRINT 'PREGRADO: No se puede realizar el cambio - Error en la
                planificación SIGE - MOODLE';
            END
        END
    END
END
```

```

        ELSE
        BEGIN
            PRINT ' PREGRADO: La planifiación nueva no existe en moodle...';
        END
    END
ELSE
    -- VALIDA SI LA MATERIA EXISTE EN MOODLE POSGRADO
    SELECT @MDL_FLAG = COUNT(1)
    FROM MOODLE_POS.[posgrado].[public].mdl_course
    WHERE mdl_course.visible =1
    and mdl_course.enablecompletion=1
    and mdl_course.idnumber=cast(@ID_PLANIFICACION_NEW as NVARCHAR);

    SELECT @MDL_FLAG2 = COUNT(1)
    FROM MOODLE_POS.[posgrado].[public].mdl_course
    WHERE mdl_course.visible =1
    and mdl_course.enablecompletion=1
    and mdl_course.idnumber=cast(@ID_PLANIFICACION_OLD as NVARCHAR);

    IF @MDL_FLAG >0 AND @MDL_FLAG2 >0
    BEGIN

        IF @IDENTIFICACION IS NOT NULL AND @ID_PLANIFICACION_OLD IS NOT
        NULL AND @ID_PLANIFICACION_NEW IS NOT NULL
        BEGIN
            EXEC [SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE]
            @IDENTIFICACION, @ID_PLANIFICACION_OLD,2;
            EXEC [SPI_MATRICULA_ESTUDIANTE_MOODLE] @IDENTIFICACION,
            @ID_PLANIFICACION_NEW,2;

            PRINT 'POSGRADO: Se realizo el cambio de paralelo...';
        END
        ELSE
        BEGIN
            PRINT 'POSGRADO: No se puede realizar el cambio - Error en la
            planificación SIGE - MOODLE';
        END

    END
    ELSE
    BEGIN
        PRINT ' POSGRADO: La materia nueva no existe en moodle...';
    END
END
GO

```

2. SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE.sql

```

USE [SIGE]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
    WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE'
)
DROP PROCEDURE dbo.SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE
GO

```

```

CREATE PROCEDURE [dbo].[SPI_ELIMINA_MATRICULA_ESTUDIANTE_MOODLE] (
    @ID_CEDULA VARCHAR(13),
    @ID_PLANIFICACION INT,
    @TIPO INT
)
AS
DECLARE
    @mdl_userid VARCHAR (10),
    @mdl_user_enrol_id varchar(10),
    @mdl_role_asig_id varchar(10),
    @count int;

BEGIN
    IF(@TIPO = 1 )
    BEGIN
        -- SE OBTINE EL ID DEL USUARIO EN EL SISTEMA MOODLE
        select @mdl_userid= A.ID, @count= count(1)
        from MOODLE_PRE.[posgrado].[public].mdl_user A
        where SUBSTRING(A.USERNAME, 2, 10)=@ID_CEDULA
        GROUP BY A.ID;
        IF(@count > 0 )
        BEGIN
            ---Valida si el usuario esta matriculado en moodle
            SELECT
                @mdl_role_asig_id = mdl_role_assignments.id,
                @mdl_user_enrol_id = mdl_user_enrolments.id,
                @count=count(1)
            FROM MOODLE_PRE.[posgrado].[public].mdl_course
            INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_context ON
mdl_context.instanceid = mdl_course.id
            INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role_assignments ON
mdl_context.id = mdl_role_assignments.contextid
            and mdl_role_assignments.userid=@mdl_userid
            INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role ON mdl_role.id
= mdl_role_assignments.roleid
            INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_user ON mdl_user.id
= mdl_role_assignments.userid
            INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_enrol ON
mdl_enrol.courseid = mdl_course.id
            inner join MOODLE_PRE.[posgrado].[public].mdl_user_enrolments on
mdl_user_enrolments.enrolid=mdl_enrol.id
            and mdl_user_enrolments.userid=@mdl_userid
            WHERE mdl_role.shortname = 'student'
            and mdl_course.visible =1
            and mdl_course.enablecompletion=1
            and mdl_course.idnumber=cast(@ID_PLANIFICACION as NVARCHAR)
            group by mdl_role_assignments.id, mdl_user_enrolments.id;

            IF(@count = 1 )
            BEGIN
                DELETE FROM
MOODLE_PRE.[posgrado].[public].mdl_role_assignments WHERE id =
@mdl_role_asig_id;
                DELETE FROM MOODLE_PRE.[posgrado].[public].mdl_user_enrolments
WHERE id = @mdl_user_enrol_id;
                PRINT 'PREGRADO: Se elimino la matricula del estudiante
correctamente...';
            END
            ELSE
            BEGIN
                PRINT 'PREGRADO: El estudiante no se encunetra matriculado en
la materia...';
            END
            END
            ELSE
            BEGIN
                PRINT 'PREGRADO: El estudiante no existe...';
            END
        END
    END

```



```

END
END
ELSE
BEGIN
----SE OBTINE EL ID DEL USUARIO EN EL SISTEMA MOODLE
select @mdl_userid= A.ID, @count= count(1)
from MOODLE_POS.[posgrado].[public].mdl_user A
where SUBSTRING(A.USERNAME, 2, 10)=@ID_CEDULA
GROUP BY A.ID;

IF(@count > 0 )
BEGIN
-- Valida si el usuario esta matriculado en moodle
SELECT
@mdl_role_asig_id = mdl_role_assignments.id,
@mdl_user_enrol_id = mdl_user_enrolments.id,
@count=count(1)
FROM MOODLE_POS.[posgrado].[public].mdl_course
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_context ON
mdl_context.instanceid = mdl_course.id
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_role_assignments ON
mdl_context.id = mdl_role_assignments.contextid
and mdl_role_assignments.userid =@mdl_userid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_role ON mdl_role.id
= mdl_role_assignments.roleid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_user ON mdl_user.id
= mdl_role_assignments.userid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_enrol ON
mdl_enrol.courseid = mdl_course.id
inner join MOODLE_POS.[posgrado].[public].mdl_user_enrolments on
mdl_user_enrolments.enrolid=mdl_enrol.id
and mdl_user_enrolments.userid=@mdl_userid
WHERE mdl_role.shortname = 'student'
and mdl_course.visible =1
and mdl_course.enablecompletion=1
and mdl_course.idnumber=cast(@ID_PLANIFICACION as NVARCHAR)
group by mdl_role_assignments.id, mdl_user_enrolments.id;

IF(@count = 1 )
BEGIN
DELETE FROM
MOODLE_POS.[posgrado].[public].mdl_role_assignments WHERE id =
@mdl_role_asig_id;
DELETE FROM MOODLE_POS.[posgrado].[public].mdl_user_enrolments
WHERE id = @mdl_user_enrol_id;
PRINT 'POSGRADO: Se elimino la matricula del estudiante
correctamente...';
END
ELSE
BEGIN
PRINT 'POSGRADO: El estudiante no se encunetra matriculado en
la materia...';
END
END
ELSE
BEGIN
PRINT 'PREGRADO: El estudiante no existe...';
END
END
END
GO

```

3. SPI_MATRICULA_ESTUDIANTE_MOODLE.sql

```

USE [SIGE]
GO

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
    WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'SPI_MATRICULA_ESTUDIANTE_MOODLE'
)
    DROP PROCEDURE dbo.SPI_MATRICULA_ESTUDIANTE_MOODLE
GO

CREATE PROCEDURE [dbo].[SPI_MATRICULA_ESTUDIANTE_MOODLE] (
    @ID_CEDULA VARCHAR(13),
    @ID_PLANIFICACION INT,
    @TIPO INT
)
AS
DECLARE
    @mdl_userid VARCHAR (10),
    @mdl_context VARCHAR (10),
    @mdl_roleid varchar(10),
    @mdl_courseid varchar(100),
    @mdl_enrolid varchar(10),
    @count int;

BEGIN
    IF(@TIPO = 1 )
        BEGIN
            -- SE OBTIENE EL ID DEL USUARIO EN EL SISTEMA MOODLE
            select @mdl_userid= A.ID, @count= count(1)
            from MOODLE_PRE.[posgrado].[public].mdl_user A
            where SUBSTRING(A.USERNAME, 2, 10)=@ID_CEDULA
            GROUP BY A.ID;

            IF(@count > 0 )
                BEGIN
                    SELECT
                        @count= count(1)
                    FROM MOODLE_PRE.[posgrado].[public].mdl_course
                    INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_context ON
mdl_context.instanceid = mdl_course.id
                    INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role_assignments ON
mdl_context.id = mdl_role_assignments.contextid
                    and mdl_role_assignments.userid=@mdl_userid
                    INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role ON mdl_role.id
= mdl_role_assignments.roleid
                    INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_user ON mdl_user.id
= mdl_role_assignments.userid
                    INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_enrol ON
mdl_enrol.courseid = mdl_course.id
                    WHERE mdl_role.shortname = 'student'
                    and mdl_course.visible =1
                    and mdl_course.enablecompletion=1
                    and mdl_course.idnumber=cast(@ID_PLANIFICACION as NVARCHAR)
                    IF(@count = 0 )
                        BEGIN
                            SELECT
                                @mdl_roleid= mdl_role.id ,
                                @mdl_context = max(mdl_context.id),
                                @mdl_enrolid= max(mdl_enrol.id)
                            FROM MOODLE_PRE.[posgrado].[public].mdl_course
                            INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_context ON
mdl_context.instanceid = mdl_course.id

```

```

INNER JOIN
MOODLE_PRE.[posgrado].[public].mdl_role_assignments ON mdl_context.id =
mdl_role_assignments.contextid
INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_role ON
mdl_role.id = mdl_role_assignments.roleid
INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_user ON
mdl_user.id = mdl_role_assignments.userid
INNER JOIN MOODLE_PRE.[posgrado].[public].mdl_enrol ON
mdl_enrol.courseid = mdl_course.id
WHERE mdl_role.shortname = 'student'
and mdl_course.visible =1
and mdl_course.enablecompletion=1
and mdl_course.idnumber=cast(@ID_PLANIFICACION as
NVARCHAR)
GROUP BY mdl_role.id;

IF @mdl_roleid IS NOT NULL AND @mdl_context IS NOT NULL
AND @mdl_enrolid IS NOT NULL AND @mdl_userid IS NOT NULL
BEGIN
INSERT INTO
MOODLE_PRE.[posgrado].[public].mdl_user_enrolments
(enrolid,status,userid,timestart,timeend,modifierid,timecreated,timemodified)
VALUES (@mdl_enrolid,'0', @mdl_userid, '1609638116',
'0', @mdl_userid, '1609638116', '1609638116');

INSERT INTO
MOODLE_PRE.[posgrado].[public].mdl_role_assignments
(roleid,contextid,userid,component,itemid,timemodified,modifierid,sortorder)
VALUES (@mdl_roleid,@mdl_context, @mdl_userid, '',
'0', '1609638116', @mdl_enrolid, '0');

PRINT 'PREGRADO: Estudiante matriculado
correctamente...';
END
ELSE
BEGIN
PRINT 'PREGRADO: No se puede realizar el insert -
Error en la planificación SIGE - MOODLE';
END
END
ELSE
BEGIN
PRINT 'PREGRADO: ya se encuentra matriculado';
END
END
ELSE
BEGIN
PRINT 'PREGRADO: EL usuario no existe en moodle ';
END
END
ELSE
BEGIN
-- SE OBTINE EL ID DEL USUARIO EN EL SISTEMA MOODLE
select @mdl_userid= A.ID, @count= count(1)
from MOODLE_POS.[posgrado].[public].mdl_user A
where SUBSTRING(A.USERNAME, 2, 10)=@ID_CEDULA
GROUP BY A.ID;

IF(@count > 0 )
BEGIN
SELECT
@count= count(1)
FROM MOODLE_POS.[posgrado].[public].mdl_course
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_context ON
mdl_context.instanceid = mdl_course.id
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_role_assignments ON
mdl_context.id = mdl_role_assignments.contextid
and mdl_role_assignments.userid =@mdl_userid

```

```

INNER JOIN MOODLE_POS.[posgrado].[public].mdl_role ON mdl_role.id
= mdl_role_assignments.roleid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_user ON mdl_user.id
= mdl_role_assignments.userid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_enrol ON
mdl_enrol.courseid = mdl_course.id
WHERE mdl_role.shortname = 'student'
and mdl_course.visible =1
and mdl_course.enablecompletion=1
and mdl_course.idnumber=cast(@ID_PLANIFICACION as NVARCHAR)
IF(@count = 0 )
BEGIN
SELECT
@mdl_roleid= mdl_role.id ,
@mdl_context = max(mdl_context.id),
@mdl_enrolid= max(mdl_enrol.id)
FROM MOODLE_POS.[posgrado].[public].mdl_course
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_context ON
mdl_context.instanceid = mdl_course.id
INNER JOIN
MOODLE_POS.[posgrado].[public].mdl_role_assignments ON mdl_context.id =
mdl_role_assignments.contextid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_role ON
mdl_role.id = mdl_role_assignments.roleid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_user ON
mdl_user.id = mdl_role_assignments.userid
INNER JOIN MOODLE_POS.[posgrado].[public].mdl_enrol ON
mdl_enrol.courseid = mdl_course.id
WHERE mdl_role.shortname = 'student'
and mdl_course.visible =1
and mdl_course.enablecompletion=1
and mdl_course.idnumber=cast(@ID_PLANIFICACION as
NVARCHAR)
GROUP BY mdl_role.id;

IF @mdl_roleid IS NOT NULL AND @mdl_context IS NOT NULL
AND @mdl_enrolid IS NOT NULL AND @mdl_userid IS NOT NULL
BEGIN
INSERT INTO
MOODLE_POS.[posgrado].[public].mdl_user_enrolments
(enrolid,status,userid,timestart,timeend,modifierid,timecreated,timemodified)
VALUES (@mdl_enrolid,'0', @mdl_userid, '1609638116',
'0', @mdl_userid, '1609638116', '1609638116');

INSERT INTO
MOODLE_POS.[posgrado].[public].mdl_role_assignments
(roleid,contextid,userid,component,itemid,timemodified,modifierid,sortorder)
VALUES(@mdl_roleid,@mdl_context, @mdl_userid, '',
'0', '1609638116', @mdl_enrolid, '0');

PRINT 'POSGRADO: Estudiante matriculado
correctamente...';
END
ELSE
BEGIN
PRINT 'POSGRADO: No se puede realizar el insert -
Error en la planificación SIGE - MOODLE';
END

END
ELSE
BEGIN
PRINT 'POSGRADO: ya se encunetra matriculado';
END

END
ELSE
BEGIN
PRINT 'POSGRADO: EL usuario no existe en moodle ';
END

```

```

END
END
GO

```

4. SP_CAMBIO_PARALELO_ESTUDIANTE.sql

```

USE [SIGE]
GO
/***** Object:  StoredProcedure [dbo].[SP_CAMBIO_PARALELO_ESTUDIANTE]
Script Date: 18/1/2021 19:46:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
    WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'SP_CAMBIO_PARALELO_ESTUDIANTE'
)
    DROP PROCEDURE dbo.SP_CAMBIO_PARALELO_ESTUDIANTE
GO

CREATE PROCEDURE [dbo].[SP_CAMBIO_PARALELO_ESTUDIANTE]
    @MTD_CODIGO          NUMERIC(8,0),
    @PLA_CODIGO          NUMERIC(5,0),
    @PER_CODIGO          NUMERIC(5,0)
AS
DECLARE @INS_CODIGO      NUMERIC(5,0),
        @PRA_CODIGO      NUMERIC(5,0),
        @NIV_CODIGO      NUMERIC(5,0),
        @TIPO_MATERIA    INT,
        @MENSAJE_ERROR   VARCHAR(200),
        @MTD_TIPO        VARCHAR(2),
        @PRL_CODIGO      NUMERIC(5,0),
        @MOD_CODIGO      NUMERIC(5,0),
        @ASI_CODIGO      NUMERIC(5,0),
        @NA_VALOR        NUMERIC(5,0),
        @MISMO_NIVEL     NUMERIC(5,0),
        @NIV_CODIGO2     NUMERIC(5,0),
        @PLA_CODIGO_ANTERIOR NUMERIC(5,0),
        @ASI_CODIGO_CUP   NUMERIC(5,0),
        @HOR_CODIGO_CUP   NUMERIC(5,0),
        @PRL_CODIGO_CUP   NUMERIC(5,0),
        @MOD_CODIGO_CUP   NUMERIC(5,0),
        @PRF_CODIGO_CUP   NUMERIC(5,0),
        @AUL_CODIGO_CUP   NUMERIC(5,0),
        @PRA_CODIGO_CUP   NUMERIC(5,0),
        @CAPACIDAD_AULA_TOTAL INT,
        @CANT_ESTUDIANTES_CAPACIDAD INT,
        @VAL_LOG          VARCHAR(200),
        @VAL_LOG2         VARCHAR(200),
        @MTR_CODIGO      NUMERIC(5,0),
        -----Variable moodle
        @INS_CODIGO_MDL   INT,
        @TIPO_E           INT,
        @IDENTIFICACION   VARCHAR(13)

SET @INS_CODIGO = 0
SET @PRA_CODIGO = 0
SET @NIV_CODIGO = 0
SET @TIPO_MATERIA = 0
SET @MENSAJE_ERROR = ''
SET @MTD_TIPO = 0
SET @PRL_CODIGO = 0

```

```

SET @MOD_CODIGO = 0
SET @ASI_CODIGO = 0

--PLANIFICACIÓN A CAMBIAR (ANTERIOR)
SELECT @PLA_CODIGO_ANTERIOR = PLA_CODIGO, @MTR_CODIGO = MTR_CODIGO
FROM TMGA_MATRICULAS_DETALLES
WHERE MTD_CODIGO = @MTD_CODIGO

/*
EXEC SP_VERIFICA_CRUCE_HORARIOS @MTR_CODIGO, @PLA_CODIGO, @MENSAJE_ERROR =
@MENSAJE_ERROR OUTPUT
IF ISNULL(@MENSAJE_ERROR, '') <> ''
BEGIN
    SELECT @MENSAJE_ERROR AS MENSAJE, 2 AS TIPO
    RETURN 0
END
*/

--INSCRIPCIÓN, PERIODO, NIVEL
SELECT @INS_CODIGO = INS_CODIGO,
@PRA_CODIGO = PRA_CODIGO,
@NIV_CODIGO = NIV_CODIGO
FROM TMGA_MATRICULAS WHERE MTR_CODIGO = @MTR_CODIGO

--ASIGNATURA
SELECT @ASI_CODIGO = TMAD.ASI_CODIGO
FROM TMGA_PLANIFICACIONES TPL
INNER JOIN TMGA_MALLAS_DETALLES TMAD
ON TPL.DTM_CODIGO = TMAD.DTM_CODIGO
WHERE PLA_CODIGO = @PLA_CODIGO

--NIVEL PLANIFICACIÓN
SELECT @NIV_CODIGO2 = TMAD.NIV_CODIGO
FROM TMGA_PLANIFICACIONES TPL
INNER JOIN TMGA_MALLAS_DETALLES TMAD
ON TPL.DTM_CODIGO = TMAD.DTM_CODIGO
AND PLA_CODIGO = @PLA_CODIGO

--VERIFICACIÓN DE SI LA ASIGNATURA ES DEL MISMO NIVEL QUE LA MATRÍCULA
IF @NIV_CODIGO = @NIV_CODIGO2
BEGIN
    SET @MISMO_NIVEL = 1
END
ELSE
BEGIN
    SET @MISMO_NIVEL = 0
END

--VERIFICACIÓN DE SI LA MATERIA YA SE APROBÓ
IF EXISTS(SELECT 1
FROM TMGA_NOTAS_ACADEMICAS TN
INNER JOIN TMGA_TIPOS_NOTAS_ACADEMICAS TTN
ON TN.TNA_CODIGO = TTN.TNA_CODIGO
INNER JOIN TMGA_MATRICULAS_DETALLES TMD
ON TN.MTD_CODIGO = TMD.MTD_CODIGO
INNER JOIN TMGA_MATRICULAS TM
ON TMD.MTR_CODIGO = TM.MTR_CODIGO
INNER JOIN TMGA_PLANIFICACIONES TPL
ON TMD.PLA_CODIGO = TPL.PLA_CODIGO
INNER JOIN TMGA_MALLAS_DETALLES TMAD
ON TPL.DTM_CODIGO = TMAD.DTM_CODIGO
WHERE TM.INS_CODIGO = @INS_CODIGO
AND TMAD.ASI_CODIGO = @ASI_CODIGO
AND TTN.TPN_CODIGO = 3
AND TN.NA_VALOR = 1)
BEGIN
    SELECT 'NO SE PUEDE INSERTAR ASIGNATURA APROBADA.' AS MENSAJE, 2 AS
TIPO

```

```

RETURN 0
END

--ASIGNATURA ES DEL MISMO NIVEL DE LA MATRÍCULA
IF @MISMO_NIVEL = 1
BEGIN
SET @TIPO_MATERIA = 0
END
ELSE --ASIGNATURA DE DIFERENTE NIVEL DE LA MATRÍCULA, SE VERIFICA SI ES DE
ARRASTRE O SECUENCIA
BEGIN
IF NOT EXISTS (SELECT 1
FROM TMGA_NOTAS_ACADEMICAS TN
INNER JOIN TMGA_TIPOS_NOTAS_ACADEMICAS TTN
ON TN.TNA_CODIGO = TTN.TNA_CODIGO
INNER JOIN TMGA_MATRICULAS_DETALLES TMD
ON TN.MTD_CODIGO = TMD.MTD_CODIGO
INNER JOIN TMGA_MATRICULAS TM
ON TMD.MTR_CODIGO = TM.MTR_CODIGO
INNER JOIN TMGA_PLANIFICACIONES TPL
ON TMD.PLA_CODIGO = TPL.PLA_CODIGO
INNER JOIN TMGA_MALLAS_DETALLES TMAD
ON TPL.DTM_CODIGO = TMAD.DTM_CODIGO
WHERE TM.INS_CODIGO = @INS_CODIGO --1082--
@INS_CODIGO_HOMOLOG
AND TMAD.ASI_CODIGO = @ASI_CODIGO--376
AND TTN.TPN_CODIGO = 3)
BEGIN
SET @TIPO_MATERIA = 2 --SECUENCIA
END
ELSE
BEGIN
SELECT @NA_VALOR = TN.NA_VALOR
FROM TMGA_NOTAS_ACADEMICAS TN
INNER JOIN TMGA_TIPOS_NOTAS_ACADEMICAS TTN
ON TN.TNA_CODIGO = TTN.TNA_CODIGO
INNER JOIN TMGA_MATRICULAS_DETALLES TMD
ON TN.MTD_CODIGO = TMD.MTD_CODIGO
INNER JOIN TMGA_MATRICULAS TM
ON TMD.MTR_CODIGO = TM.MTR_CODIGO
INNER JOIN TMGA_PLANIFICACIONES TPL
ON TMD.PLA_CODIGO = TPL.PLA_CODIGO
INNER JOIN TMGA_MALLAS_DETALLES TMAD
ON TPL.DTM_CODIGO = TMAD.DTM_CODIGO
WHERE TM.INS_CODIGO = @INS_CODIGO --1082--@INS_CODIGO_HOMOLOG
AND TMAD.ASI_CODIGO = @ASI_CODIGO--376
AND TTN.TPN_CODIGO = 3
--AND TN.NA_VALOR = 0

IF @NA_VALOR = 0
BEGIN
SET @TIPO_MATERIA = 1 --ARRASTRE
END
END
END

--MODALIDAD
SELECT @MOD_CODIGO = RCM.MOD_CODIGO
FROM TMGA_INSCRIPCIONES TI
INNER JOIN RMGA_CARRERAS_MODALIDADES RCM
ON TI.RCM_CODIGO = RCM.RCM_CODIGO
WHERE INS_CODIGO = @INS_CODIGO

--TIPO MATRÍCULA
SELECT @MTD_TIPO = RCT.TPM_CODIGO
FROM TMGA_CALENDARIOS_ACADEMICOS TCA
INNER JOIN TMGA_CALENDARIOACAD_DETALLES TCD
ON TCA.CLA_CODIGO = TCD.CLA_CODIGO

```

```

INNER JOIN RMGA_CALENDARIODETALLE_TIPOMATRICULAS RCT
ON TCD.CLD_CODIGO = RCT.CLD_CODIGO
INNER JOIN TMGA_TIPOS_MATRICULAS TM ON RCT.TPM_CODIGO = TM.TPM_CODIGO
WHERE PRA_CODIGO = @PRA_CODIGO
AND MOD_CODIGO = @MOD_CODIGO
AND GETDATE() BETWEEN CLD_FECHA_INICIO AND CLD_FECHA_FIN

--PARALELO
SELECT @PRL_CODIGO = PRL_CODIGO
FROM TMGA_PLANIFICACIONES
WHERE PLA_CODIGO = @PLA_CODIGO

/*
--INSERCIÓN EN TABLA DE ASIGNATURAS DE NUEVA PLANIFICACIÓN
INSERT INTO TMP_PARALELOS (PLA_CODIGO, PRL_CODIGO, ASI_CODIGO,
ASI_DESCRIPCION, INS_CODIGO,
NIV_CODIGO, TIPO_MATERIA, SN_MOSTRAR, SN_MATRICULADO, SN_ESTADO,
PRA_CODIGO)
SELECT @PLA_CODIGO, @PRL_CODIGO, TMAD.ASI_CODIGO, TA.ASI_DESCRIPCION,
@INS_CODIGO,
@NIV_CODIGO, @TIPO_MATERIA, 1, 0, 1, @PRA_CODIGO
FROM TMGA_PLANIFICACIONES TPL
INNER JOIN TMGA_MALLAS_DETALLES TMAD
ON TPL.DTM_CODIGO = TMAD.DTM_CODIGO
INNER JOIN TMGA_ASIGNATURAS TA
ON TMAD.ASI_CODIGO = TA.ASI_CODIGO
WHERE PLA_CODIGO = @PLA_CODIGO
*/

/*EXEC SP_VALIDA_CAPACIDAD_PRL_ASI @INS_CODIGO, @PRL_CODIGO, @NIV_CODIGO2,
@PLA_CODIGO, @MENSAJE_ERROR = @MENSAJE_ERROR OUTPUT
IF ISNULL(@MENSAJE_ERROR, '') <> ''
BEGIN
SELECT @MENSAJE_ERROR AS MENSAJE, 2 AS TIPO
RETURN 0
END*/

--SEMIPRESENCIAL
/*IF @MOD_CODIGO = 2
BEGIN
--VERIFICA CAPACIDAD DEL AULA PARA MATRICULAR
SELECT @ASI_CODIGO_CUP = MD.ASI_CODIGO, @HOR_CODIGO_CUP =
RH.HOR_CODIGO,
@MOD_CODIGO_CUP = PLA.MOD_CODIGO, @PRF_CODIGO_CUP =
PLA.PRF_CODIGO, @AUL_CODIGO_CUP = PLA.AUL_CODIGO, @PRA_CODIGO_CUP =
PLA.PRA_CODIGO
FROM TMGA_PLANIFICACIONES PLA
INNER JOIN TMGA_MALLAS_DETALLES MD
ON PLA.PLA_CODIGO = @PLA_CODIGO
INNER JOIN RMGA_PLANIFICACION_HORARIOS RH
ON MD.DTM_CODIGO = PLA.DTM_CODIGO
WHERE RH.PLA_CODIGO = PLA.PLA_CODIGO
SELECT @CAPACIDAD_AULA_TOTAL = ISNULL(CPP_CUPO, 0) FROM
[TMP_CUPOS_PLANIFICACIONES] WHERE ASI_CODIGO = @ASI_CODIGO_CUP
AND HOR_CODIGO = @HOR_CODIGO_CUP AND MOD_CODIGO = @MOD_CODIGO_CUP
AND PRF_CODIGO = @PRF_CODIGO_CUP
AND AUL_CODIGO = @AUL_CODIGO_CUP AND PRA_CODIGO = @PRA_CODIGO_CUP

SELECT @CANT_ESTUDIANTES_CAPACIDAD = COUNT(*)
FROM TMGA_MATRICULAS_DETALLES MD
INNER JOIN TMGA_PLANIFICACIONES PLA
ON MD.PLA_CODIGO = PLA.PLA_CODIGO
INNER JOIN TMGA_MATRICULAS M
ON M.MTR_CODIGO = MD.MTR_CODIGO
AND M.PRA_CODIGO = PLA.PRA_CODIGO
INNER JOIN TMGA_MODALIDADES MO
ON MO.MOD_CODIGO = PLA.MOD_CODIGO
INNER JOIN TMGA_MALLAS_DETALLES MAL

```



```

ON PLA.DTM_CODIGO = MAL.DTM_CODIGO
INNER JOIN TMGA_ASIGNATURAS ASI
ON ASI.ASI_CODIGO = MAL.ASI_CODIGO
INNER JOIN TMGA_NIVELES NI
ON NI.NIV_CODIGO = MAL.NIV_CODIGO
INNER JOIN TMGA_PARALELOS PRL
ON PRL.PRL_CODIGO = PLA.PRL_CODIGO
INNER JOIN TMGA_MALLAS MA
ON MA.MAL_CODIGO = MAL.MAL_CODIGO
INNER JOIN TMGA_CARRERAS CA
ON MA.CAR_CODIGO = CA.CAR_CODIGO
INNER JOIN RMGA_PLANIFICACION_HORARIOS RH
ON RH.PLA_CODIGO = PLA.PLA_CODIGO
INNER JOIN TMGA_INSCRIPCIONES TI
ON TI.INS_CODIGO = M.INS_CODIGO
INNER JOIN TMGA_PERSONAS TP
ON TP.PER_CODIGO = TI.PER_CODIGO
WHERE PLA.PRF_CODIGO = @PRF_CODIGO_CUP --
AND PLA.PRA_CODIGO = @PRA_CODIGO_CUP --
AND TI.INS_ESTADO = 1
AND PLA.AUL_CODIGO = @AUL_CODIGO_CUP --
AND RH.HOR_CODIGO = @HOR_CODIGO_CUP --
AND PLA.MOD_CODIGO = @MOD_CODIGO_CUP --
AND ASI.ASI_CODIGO = @ASI_CODIGO_CUP --

--SELECT @ASI_CODIGO_CUP, @HOR_CODIGO_CUP, @MOD_CODIGO_CUP ,
@PRF_CODIGO_CUP ,@AUL_CODIGO_CUP , @PRA_CODIGO_CUP

--SELECT @CAPACIDAD_AULA_TOTAL AS CAPACIDAD,
@CANT_ESTUDIANTES_CAPACIDAD AS CANT

IF (@CANT_ESTUDIANTES_CAPACIDAD >= @CAPACIDAD_AULA_TOTAL)
BEGIN
SELECT 'El cupo para este paralelo se encuentra copado, por favor
seleccione otro.' AS MENSAJE, 2 AS TIPO
--ROLLBACK TRANSACTION
RETURN 0
END
END
*/

--VERIFICA SI EXISTEN CUPOS DISPONIBLES
SELECT @CAPACIDAD_AULA_TOTAL = PLA_CUPO FROM TMGA_PLANIFICACIONES WHERE
PLA_CODIGO = @PLA_CODIGO
SELECT @CAPACIDAD_AULA_TOTAL = ISNULL(@CAPACIDAD_AULA_TOTAL,0)
SELECT @CANT_ESTUDIANTES_CAPACIDAD = COUNT(1) FROM
TMGA_MATRICULAS_DETALLES WHERE PLA_CODIGO = @PLA_CODIGO

IF @CAPACIDAD_AULA_TOTAL = 0
BEGIN
SELECT 'No existe cupo disponible en esta aula (0 cupo)' AS MENSAJE, 2
AS TIPO
RETURN 0
END
IF @CAPACIDAD_AULA_TOTAL < 0
BEGIN
SELECT 'No se pudo verificar la capacidad del aula por favor
contactese con la Universidad Israel.' AS MENSAJE, 2 AS TIPO
RETURN 0
END
IF (@CANT_ESTUDIANTES_CAPACIDAD >= @CAPACIDAD_AULA_TOTAL)
BEGIN
SELECT 'El cupo para este paralelo se encuentra copado, por favor
seleccione otro.' AS MENSAJE, 2 AS TIPO
RETURN 0
END

--ACTUALIZACIÓN TABLA DE ASIGNATURAS NUEVA PLANIFICACIÓN

```

```

UPDATE TMP_PARALELOS
SET SN_MATRICULADO = 1,
SN_MOSTRAR = 0
WHERE INS_CODIGO = @INS_CODIGO
AND PLA_CODIGO = @PLA_CODIGO

--ACTUALIZACIÓN TABLA DE ASIGNATURAS PLANIFICACIÓN ANTERIOR
UPDATE TMP_PARALELOS
SET SN_MATRICULADO = 0,
SN_MOSTRAR = 0 --1
WHERE INS_CODIGO = @INS_CODIGO
AND PLA_CODIGO = @PLA_CODIGO_ANTERIOR

--ACTUALIZACIÓN MATRÍCULAS DETALLES CON LA NUEVA PLANIFICACIÓN
UPDATE TMGA_MATRICULAS_DETALLES
SET PLA_CODIGO = @PLA_CODIGO,
PRL_CODIGO = @PRL_CODIGO
WHERE MTD_CODIGO = @MTD_CODIGO

--ACTUALIZACIÓN TMP NOTAS CON LA NUEVA PLANIFICACIÓN
UPDATE TMP_NOTAS_HOR_ESTUDIANTES
SET PLA_CODIGO = @PLA_CODIGO
WHERE MTD_CODIGO = @MTD_CODIGO

--DATOS INSERCIÓN LOG
SET @VAL_LOG = 'PLA_CODIGO ANTERIOR ' + CONVERT(VARCHAR(5),
@PLA_CODIGO_ANTERIOR) + 'PLA_CODIGO NUEVO ' +
CONVERT(VARCHAR(5), @PLA_CODIGO)

SET @VAL_LOG2 = 'MTR_CODIGO ' + CONVERT(VARCHAR(5), @MTR_CODIGO) +
'MTD_CODIGO ' + CONVERT(VARCHAR(8), @MTD_CODIGO)

--DATOS INSERCIÓN LOG
IF ISNULL(@PER_CODIGO, 0) <> 0
BEGIN
EXEC SP_GENERAR_LOG_HISTORIAL @PER_CODIGO, @PRA_CODIGO, 'CAMBIO
HORARIOS ESTUDIANTE', 'ACTUALIZACIÓN HORARIO',
@VAL_LOG, @VAL_LOG2
END

SELECT '' AS MENSAJE, 1 AS TIPO;

/*****LÓGICA PARA CAMBIAR DE PARALELO EN MOODLE
*****/
--SE OBTIENE EL CÓDIGO DE INSCRIPCIÓN DEL ESTUDIANTE
SELECT @INS_CODIGO_MDL=INS_CODIGO,@TIPO_E =TPE_CODIGO,
@IDENTIFICACION=PER IDENTIFICACION
FROM TMGA_PERSONAS TP
INNER JOIN TMGA_INSCRIPCIONES TI ON TI.PER_CODIGO = @PER_CODIGO AND
INS_ESTADO=1 AND TPE_CODIGO IN (1,2)
where TP.PER_CODIGO = @PER_CODIGO

EXEC [SPI_ACTUALIZA_MATRICULA_ESTIDIANTE_MOODLE]
@IDENTIFICACION,@INS_CODIGO_MDL, @PLA_CODIGO,@PLA_CODIGO_ANTERIOR,@TIPO_E;
GO

```

5. SP_SAVE_MATRICULA_MATERIA_EST_POR_ASIGNATURA_2.sql

```

USE [SIGE]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

IF EXISTS (

```

```

SELECT *
FROM INFORMATION_SCHEMA.ROUTINES
WHERE SPECIFIC_SCHEMA = N'dbo'
AND SPECIFIC_NAME = N'SP_SAVE_MATRICULA_MATERIA_EST_POR_ASIGNATURA_2'
)
DROP PROCEDURE dbo.SP_SAVE_MATRICULA_MATERIA_EST_POR_ASIGNATURA_2
GO

CREATE PROCEDURE [dbo].[SP_SAVE_MATRICULA_MATERIA_EST_POR_ASIGNATURA_2]

--PARAMETROS DE ENTRADA
    @INS_CODIGO          NUMERIC(8,0),
    @PLA_CODIGO          NUMERIC(5,0)

AS
DECLARE
--DECLARACION DE VARIABLES
    @ERROR_NUMBER        INT,
    @ERROR_PROC           NVARCHAR(126),
    @ERROR_MESSAGE       NVARCHAR(2048),
    @MSG                  VARCHAR(500),
    @AUX                  INT,
    @REGISTROS            VARCHAR(500),
    @SIGLAS               VARCHAR(5),
    @MTR_NUM_FORMULARIO  VARCHAR(20),
    @DTM_CODIGO          NUMERIC(5,0),
    @NUM_MATRICULA        INT,
    @NIV_CODIGO_MENOR    NUMERIC(5,0),
    @REGISTROS_CABECERA  INT,
    @MAX_NIVEL           NUMERIC(5,0),
    @PRA_CODIGO          NUMERIC(5,0),
    @TIPO_MATRICULA      VARCHAR(20),
    @MOD_CODIGO          NUMERIC(5,0),
    @TPM_CODIGO          NUMERIC(5,0),
    @NUM_FORMULARIO      VARCHAR(20),
    @SIGLAS_CARRERA      VARCHAR(10),
    @MTR_CODIGO          NUMERIC(5,0),
    @ASI_CODIGO          NUMERIC(5,0),

    @DUBLICADOS           INT,
    @AUL_CODIGO          NUMERIC(5,0),
    @CAPACIDAD_AULA_TOTAL INT,
    @HOR_CODIGO          NUMERIC(5,0),
    @CONDICIONADO        bit,
    @TIPO_MATERIA        INT,
    @MAT QUIERE          INT,
    @MAT TIENE           INT,
    @MENSAJE_ERROR       VARCHAR(500),
    @TPE_CODIGO          NUMERIC(5,0),
    @PRL_CODIGO          numeric(5,0),
    @HOR_CODIGO_INGRESA  NUMERIC(5,0),
    @MTD_CODIGO          NUMERIC(8,0),
    @CANT_ESTUDIANTES_CAPACIDAD INT,

    @ASI_CODIGO_CUP      NUMERIC(5,0),
    @HOR_CODIGO_CUP      NUMERIC(5,0),
    @PRL_CODIGO_CUP      NUMERIC(5,0),
    @MOD_CODIGO_CUP      NUMERIC(5,0),
    @PRF_CODIGO_CUP      NUMERIC(5,0),
    @AUL_CODIGO_CUP      NUMERIC(5,0),
    @PRA_CODIGO_CUP      NUMERIC(5,0),
    @MIN_NIVEL           NUMERIC(5,0),
    @NIVEL_MATERIA       NUMERIC(5,0),
    @ASI_CODIGO_NUM      NUMERIC(5,0),
    @PRL_CODIGO_A_MATRICULAR NUMERIC(5,0),
    @PRL_CODIGO_MATRICULADO NUMERIC(5,0),
    @PRL_DESCRIPCION     VARCHAR(10),
    @ASI_CODIGO_VER      NUMERIC(5,0),

```

```

@NIV_CODIGO                NUMERIC (5,0),
@MTR_LEGALIZADO            INT,
@FEC_LEGALIZADO            DATETIME,
@PRL_CODIGO_BASE           NUMERIC (5,0),
@ASI_CODIGO_BASE           NUMERIC (5,0),
@ASI_DESCRIPCION_BASE      VARCHAR (200),
@NIV_MATERIA_BASE          NUMERIC (5,0),
@ASI_DESCRIPCION_CRUCE     VARCHAR (120),
@MENSAJE_BECA              VARCHAR (50),
@ASI_DESCRIPCION           VARCHAR (150),
@ENVIAR_CORREO             INT,
@PER_CODIGO                NUMERIC (5,0),
@MOM_CODIGO                NUMERIC (5,0),
@MOM_CODIGO_MAT            NUMERIC (5,0),
@PLA_CODIGO_MAT            NUMERIC (5,0),
@SEMANAS_MAT               INT,

@PCC_CODIGO                NUMERIC (5,0),
@TMP_CODIGO                NUMERIC (8,0),
@ERROR                     INT,
@FECHA_PAGO                DATE,
@SEMANAS                   INT,
@MAX_NIVEL_EST             NUMERIC (5,0),
@DESCRIPCION_CERTIFICACION VARCHAR (200),
--VARIABLES MOODLE
@IDENTIFICACION           VARCHAR (13),
@MDL_FLAG                  INT,
@MDL_PRE_POS               INT,
@MDL_NUM                   INT;

BEGIN TRY

    SET @MENSAJE_BECA = ''
    SET @ENVIAR_CORREO = 0

    -- OBTIENE EL NOMBRE DE LA ASIGNATURA
    SELECT @ASI_DESCRIPCION = ASI.ASI_DESCRIPCION
    FROM TMGA_PLANIFICACIONES PLA
    INNER JOIN TMGA_MALLAS_DETALLES MAD ON MAD.DTM_CODIGO =
PLA.DTM_CODIGO
    INNER JOIN TMGA_ASIGNATURAS ASI ON ASI.ASI_CODIGO = MAD.ASI_CODIGO
    WHERE PLA.PLA_CODIGO = @PLA_CODIGO

    SELECT @TPE_CODIGO = TPE_CODIGO, @PER_CODIGO = PER_CODIGO FROM
TMGA_INSCRIPCIONES WHERE INS_CODIGO = @INS_CODIGO

    --OBTIENE EL NIVEL DE LA ASIGNATURA
    SELECT @NIV_CODIGO = NIV_CODIGO
    FROM TMP_PARALELOS
    WHERE INS_CODIGO = @INS_CODIGO
    AND PLA_CODIGO = @PLA_CODIGO

    --VERIFICA EL PERIODO MAXIMO HABILITADO PARA MATRICULAS
    SELECT @PRA_CODIGO = MAX(PRA_CODIGO)
    FROM TMGA_PERIODOS_ACADEMICOS
    WHERE TPE_CODIGO = (SELECT TPE_CODIGO FROM TMGA_INSCRIPCIONES
WHERE INS_CODIGO = @INS_CODIGO AND TPE_CODIGO IN (1,2))
    AND PRA_HABIL_MATRICULA = 1 -- PERIODO HABILITADO PARA MATRICULAS
    AND PRA_ESTADO = 1 --TIENE QUE ESTAR EL PERIODO HABILITADO PARA
MATRICULARSE

    --SET @PRA_CODIGO = 38

    --VALIDA SI EXISTE EN LA TMP PARALELOS

```

```

SELECT @AUX = COUNT(1) FROM TMP_PARALELOS
WHERE PLA_CODIGO = @PLA_CODIGO
      AND INS_CODIGO = @INS_CODIGO
      AND PRA_CODIGO = @PRA_CODIGO
IF @AUX = 0
BEGIN
      SELECT @PRL_CODIGO_BASE = PRL_CODIGO, @ASI_CODIGO_BASE =
MD.ASI_CODIGO , @ASI_DESCRIPCION_BASE = ASI.ASI_DESCRIPCION, @NIV_MATERIA_BASE
= MD.NIV_CODIGO
      FROM TMGA_PLANIFICACIONES PLA
      INNER JOIN TMGA_MALLAS_DETALLES MD ON MD.DTM_CODIGO =
PLA.DTM_CODIGO
      INNER JOIN TMGA_ASIGNATURAS ASI ON ASI.ASI_CODIGO =
MD.ASI_CODIGO
      WHERE PLA_CODIGO = @PLA_CODIGO

      INSERT INTO TMP_PARALELOS VALUES(@PLA_CODIGO,@PRL_CODIGO_BASE,
@ASI_CODIGO_BASE,
@ASI_DESCRIPCION_BASE,@INS_CODIGO,@NIV_MATERIA_BASE,0,0,1,1,@PRA_CODIGO,'INGRE
SADO POR BASE',NULL)

      END

      SELECT @MOM_CODIGO = MOM_CODIGO FROM TMGA_PLANIFICACIONES_MODULAR
WHERE PLA_CODIGO = @PLA_CODIGO

      UPDATE TMP_PARALELOS SET OBSERVACIONES = ''
      WHERE INS_CODIGO = @INS_CODIGO
      AND PLA_CODIGO = @PLA_CODIGO
      AND PRA_CODIGO = @PRA_CODIGO

      BEGIN TRANSACTION

      ----VALIDA SI YA TIENE UNA ASIGNATURA IGUAL AGREGADA A LA
MATRICULA
      SELECT @ASI_CODIGO_VER = ASI.ASI_CODIGO
      FROM TMGA_PLANIFICACIONES PLA
      INNER JOIN TMGA_MALLAS_DETALLES MD ON MD.DTM_CODIGO =
PLA.DTM_CODIGO
      INNER JOIN TMGA_ASIGNATURAS ASI ON ASI.ASI_CODIGO = MD.ASI_CODIGO
      WHERE PLA.PLA_CODIGO = @PLA_CODIGO
      AND PLA.PRA_CODIGO = @PRA_CODIGO

      SELECT @AUX = COUNT(1)
      FROM TMGA_MATRICULAS_DETALLES MD
      INNER JOIN TMGA_MATRICULAS M ON M.MTR_CODIGO = MD.MTR_CODIGO AND
M.PRA_CODIGO = @PRA_CODIGO
      INNER JOIN TMGA_INSCRIPCIONES TI ON TI.INS_CODIGO = M.INS_CODIGO
      AND TI.INS_CODIGO = @INS_CODIGO
      INNER JOIN TMGA_PLANIFICACIONES PLA ON PLA.PLA_CODIGO =
MD.PLA_CODIGO
      INNER JOIN TMGA_MALLAS_DETALLES MAD ON MAD.DTM_CODIGO =
PLA.DTM_CODIGO
      INNER JOIN TMGA_ASIGNATURAS ASI ON ASI.ASI_CODIGO = MAD.ASI_CODIGO
      AND ASI.ASI_CODIGO = @ASI_CODIGO_VER

      IF @AUX > 0
      BEGIN
            SELECT 'Ya se encuentra matriculado en la asignatura
seleccionada.' AS MENSAJE, 2 AS TIPO
            ROLLBACK TRANSACTION
            RETURN 0
      END

      --VALIDA QUE SE MATRICULE EN EL NIVEL INFERIOR
      SELECT @MIN_NIVEL = MIN(NIV_CODIGO)
      FROM TMP_PARALELOS TMP

```

```

INNER JOIN TMGA_INSCRIPCIONES TI ON TI.INS_CODIGO =
TMP.INS_CODIGO AND TI.INS_CODIGO = @INS_CODIGO
WHERE PRA_CODIGO IN (SELECT MAX(PRA_CODIGO) FROM
TMGA_PERIODOS_ACADEMICOS WHERE TPE_CODIGO = TI.TPE_CODIGO AND
PRA_HABIL_MATRICULA = 1)
AND TMP.SN_MOSTRAR = 1
and TIPO_MATERIA != 0
and NIV_CODIGO = 0
and ASI_CODIGO is null

SELECT @NIVEL_MATERIA = MIN(NIV_CODIGO)
FROM TMP_PARALELOS TMP
INNER JOIN TMGA_INSCRIPCIONES TI ON TI.INS_CODIGO =
TMP.INS_CODIGO AND TI.INS_CODIGO = @INS_CODIGO
WHERE TMP.PRA_CODIGO IN (SELECT MAX(PRA_CODIGO) FROM
TMGA_PERIODOS_ACADEMICOS WHERE TPE_CODIGO = TI.TPE_CODIGO AND
PRA_HABIL_MATRICULA = 1)
AND TMP.SN_MOSTRAR = 1
AND TMP.PLA_CODIGO = @PLA_CODIGO
and TIPO_MATERIA != 0
and NIV_CODIGO = 0
and ASI_CODIGO is null

IF @MIN_NIVEL != @NIVEL_MATERIA
BEGIN
SELECT 'Debe matricularse primero en las materias de nivel
inferior' AS MENSAJE, 2 AS TIPO
ROLLBACK TRANSACTION
RETURN 0
END

SELECT @PRL_CODIGO = ISNULL(PRL_CODIGO,0) FROM TMP_PARALELOS
WHERE PLA_CODIGO = @PLA_CODIGO
AND INS_CODIGO = @INS_CODIGO
IF @PRL_CODIGO = 0
BEGIN
SELECT 'No se encuentra ese paralelo disponible para
matricular.' AS MENSAJE, 2 AS TIPO
ROLLBACK TRANSACTION
RETURN 0
END

-- OBTIENE LA MODALIDAD DE LA INSCRIPCION
SELECT @MOD_CODIGO = RC.MOD_CODIGO, @SIGLAS_CARRERA =
CA.CAR_SIGLAS
FROM TMGA_INSCRIPCIONES TI
INNER JOIN RMGA_CARRERAS_MODALIDADES RC ON RC.RCM_CODIGO =
TI.RCM_CODIGO
INNER JOIN TMGA_CARRERAS CA ON CA.CAR_CODIGO = RC.CAR_CODIGO
WHERE TI.INS_CODIGO = @INS_CODIGO

--
//*****
*****\
-----SOLO PARA ESTUDIANTES DEL NAC - O DISTANCIA VERIFICA QUE LAS
ASIGNATURAS A LAS QUE SE MATRICULA SEAN DEL MISMO PARALELO
IF @NIV_CODIGO = 0 or @MOD_CODIGO = 3 OR @TPE_CODIGO = 2
BEGIN
--VERIFICA EN QUE ASIGNATURA ESTA MATRICULADO
SELECT @AUX = COUNT(1)
FROM TMGA_MATRICULAS_DETALLES MD
INNER JOIN TMGA_MATRICULAS MTR ON MTR.MTR_CODIGO =
MD.MTR_CODIGO AND MTR.INS_CODIGO = @INS_CODIGO AND MTR.NIV_CODIGO =
@NIV_CODIGO AND MTR.PRA_CODIGO = @PRA_CODIGO
INNER JOIN TMGA_PLANIFICACIONES PLA ON PLA.PLA_CODIGO =
MD.PLA_CODIGO
IF @AUX > 0

```

```

        SELECT @PRL_CODIGO_MATRICULADO = PLA.PRL_CODIGO
        FROM TMGA_MATRICULAS_DETALLES MD
        INNER JOIN TMGA_MATRICULAS MTR ON MTR.MTR_CODIGO =
MD.MTR_CODIGO AND MTR.INS_CODIGO = @INS_CODIGO AND MTR.NIV_CODIGO =
@NIV_CODIGO AND MTR.PRA_CODIGO = @PRA_CODIGO
        INNER JOIN TMGA_PLANIFICACIONES PLA ON PLA.PLA_CODIGO
= MD.PLA_CODIGO

        SELECT @PRL_CODIGO_A_MATRICULAR = PRL_CODIGO FROM
TMGA_PLANIFICACIONES PLA WHERE PLA.PLA_CODIGO = @PLA_CODIGO
        SELECT @PRL_DESCRIPCION = PRL_DESCRIPCION FROM
TMGA_PARALELOS WHERE PRL_CODIGO = @PRL_CODIGO_MATRICULADO

        IF @PRL_CODIGO_A_MATRICULAR != @PRL_CODIGO_MATRICULADO
        BEGIN
            SELECT 'Debe seleccionar todas las asignaturas del
mismo paralelo. (' + @PRL_DESCRIPCION + ')' AS MENSAJE, 2 AS TIPO
            ROLLBACK TRANSACTION
            RETURN 0
        END
    END

--
--*****\
--SOLO VALIDA CAPACIDAD DE AULA PARA TODOS
IF @TPE_CODIGO = 1 --- SOLO GRADO
BEGIN
    --IF @MOD_CODIGO = 2    AND @NIV_CODIGO > 0 -- MODALIDAD
SEMIPRESENCIAL
    --BEGIN

    -----VERIFICA CAPACIDAD DEL AULA PARA MATRICULAR
        SELECT @ASI_CODIGO_CUP = MD.ASI_CODIGO,
@HOR_CODIGO_CUP = RH.HOR_CODIGO,
        @MOD_CODIGO_CUP = PLA.MOD_CODIGO, @PRF_CODIGO_CUP =
PLA.PRF_CODIGO, @AUL_CODIGO_CUP = PLA.AUL_CODIGO, @PRA_CODIGO_CUP =
PLA.PRA_CODIGO

        FROM TMGA_PLANIFICACIONES PLA
        INNER JOIN TMGA_MALLAS_DETALLES MD ON MD.DTM_CODIGO =
PLA.DTM_CODIGO
        INNER JOIN RMGA_PLANIFICACION_HORARIOS RH ON
RH.PLA_CODIGO = PLA.PLA_CODIGO
        WHERE PLA.PLA_CODIGO = @PLA_CODIGO

        SELECT @CAPACIDAD_AULA_TOTAL = PLA_CUPO FROM
TMGA_PLANIFICACIONES WHERE PLA_CODIGO = @PLA_CODIGO
        SELECT @CAPACIDAD_AULA_TOTAL =
ISNULL(@CAPACIDAD_AULA_TOTAL,0)
        SELECT @CANT_ESTUDIANTES_CAPACIDAD = COUNT(1) FROM
TMGA_MATRICULAS_DETALLES WHERE PLA_CODIGO = @PLA_CODIGO

        IF @CAPACIDAD_AULA_TOTAL = 0
        BEGIN
            SELECT 'No existe cupo disponible en esta aula (0
cupo)' AS MENSAJE, 2 AS TIPO
            ROLLBACK TRANSACTION

            UPDATE TMP_PARALELOS SET OBSERVACIONES = 'No existe
cupo disponible en esta aula'
            WHERE INS_CODIGO = @INS_CODIGO
            AND PLA_CODIGO = @PLA_CODIGO
            AND PRA_CODIGO = @PRA_CODIGO

            RETURN 0
        END
    END

```

```

        IF @CAPACIDAD_AULA_TOTAL < 0
        BEGIN
            ROLLBACK TRANSACTION

            SELECT 'No se pudo verificar la capacidad del aula por
favor contactese con la Universidad Israel.' AS MENSAJE, 2 AS TIPO
            UPDATE TMP_PARALELOS SET OBSERVACIONES = 'No se pudo
verificar la capacidad del aula.'
            WHERE INS_CODIGO = @INS_CODIGO
            AND PLA_CODIGO = @PLA_CODIGO
            AND PRA_CODIGO = @PRA_CODIGO

            RETURN 0
        END
        IF (@CANT_ESTUDIANTES_CAPACIDAD >= @CAPACIDAD_AULA_TOTAL)
        BEGIN
            SELECT 'No existe cupo para sus planificaciones.' AS
MENSAJE, 2 AS TIPO
            ROLLBACK TRANSACTION

            UPDATE TMP_PARALELOS SET OBSERVACIONES = 'No existe
cupo disponible en esta aula'
            WHERE INS_CODIGO = @INS_CODIGO
            AND PLA_CODIGO = @PLA_CODIGO
            AND PRA_CODIGO = @PRA_CODIGO
            RETURN 0
        END
    END
    -- END
    END

    --SELECCIONA EL MAXIMO NIVEL DEL ESTUDIANTE
    SELECT @MAX_NIVEL = MAX(NIV_CODIGO)
    FROM TMP_PARALELOS
    WHERE INS_CODIGO = @INS_CODIGO --AND PRL_CODIGO = @PRL_CODIGO --
AND SN_MOSTRAR = 1
    AND PRA_CODIGO = @PRA_CODIGO
    AND SN_ESTADO = 1

    SELECT @MAT_TIENE = COUNT(*)
    FROM TMGA_MATRICULAS M
    INNER JOIN TMGA_MATRICULAS_DETALLES MD ON M.MTR_CODIGO =
MD.MTR_CODIGO
    WHERE M.INS_CODIGO = @INS_CODIGO
    AND M.PRA_CODIGO = @PRA_CODIGO

    SELECT @MAT QUIERE = COUNT(*)
    FROM TMP_PARALELOS
    WHERE INS_CODIGO = @INS_CODIGO
    AND PLA_CODIGO = @PLA_CODIGO

    --SELECT @MAT_TIENE, @MAT QUIERE, @PRL_CODIGO, @PRA_CODIGO

    IF @TPE_CODIGO = 1
    BEGIN
        IF (ISNULL(@MAT_TIENE,0) + ISNULL(@MAT QUIERE,0) > 6)
        BEGIN
            SELECT 'El número de materias excede el límite de materias
permitidas (6).' AS MENSAJE, 2 AS TIPO
            ROLLBACK TRANSACTION

            UPDATE TMP_PARALELOS SET OBSERVACIONES = 'Excede el
límite de materias permitidas (6)'
            WHERE INS_CODIGO = @INS_CODIGO
            AND PLA_CODIGO = @PLA_CODIGO
            AND PRA_CODIGO = @PRA_CODIGO

            RETURN 0
        END
    END

```



```

        END
    END
    ELSE
    BEGIN
        IF (ISNULL(@MAT_TIENE,0) + ISNULL(@MAT QUIERE,0) > 5)
        BEGIN
            SELECT 'EL NÚMERO DE MATERIAS EXCEDE EL LÍMITE DE MATERIAS
PERMITIDAS (5).' AS MENSAJE, 2 AS TIPO
            ROLLBACK TRANSACTION

            UPDATE TMP_PARALELOS SET OBSERVACIONES = 'EXCEDE EL LÍMITE DE
MATERIAS PERMITIDAS (5)'
                WHERE INS_CODIGO = @INS_CODIGO
                AND PLA_CODIGO = @PLA_CODIGO
                AND PRA_CODIGO = @PRA_CODIGO

            RETURN 0
        END
    END

    SELECT @SEMANAS = PLM_SEMANAS
    FROM TMGA_PLANIFICACIONES MODULAR
    WHERE PLA_CODIGO = @PLA_CODIGO

    IF @NIV_CODIGO > 0
    BEGIN
        --VERIFICA CRUCE DE HORARIOS
        -----
        DECLARE CruceHorarios CURSOR FOR
            SELECT PH.HOR_CODIGO, PLA.PLA_CODIGO
            FROM TMGA_PLANIFICACIONES PLA
            INNER JOIN TMGA_MATRICULAS_DETALLES MTD ON
MTD.PLA_CODIGO = PLA.PLA_CODIGO
            INNER JOIN TMGA_MATRICULAS MTR ON
MTR.MTR_CODIGO = MTD.MTR_CODIGO AND INS_CODIGO = @INS_CODIGO AND
MTR.PRA_CODIGO = @PRA_CODIGO
            INNER JOIN RMGA_PLANIFICACION_HORARIOS PH ON
PH.PLA_CODIGO = PLA.PLA_CODIGO

        OPEN CruceHorarios
        FETCH NEXT FROM CruceHorarios INTO @HOR_CODIGO,
@PLA_CODIGO_MAT

        WHILE @@fetch_status = 0
        BEGIN
            SELECT @HOR_CODIGO_INGRESA = HOR_CODIGO
            FROM RMGA_PLANIFICACION_HORARIOS WHERE PLA_CODIGO = @PLA_CODIGO

            SELECT @MOM_CODIGO_MAT = MOM_CODIGO,
            FROM
            TMGA_PLANIFICACIONES MODULAR WHERE PLA_CODIGO = @PLA_CODIGO_MAT

            IF ((@HOR_CODIGO = @HOR_CODIGO_INGRESA)
            AND (@MOM_CODIGO = @MOM_CODIGO_MAT))
            BEGIN
                SELECT @ASI_DESCRIPCION_CRUCE =

                FROM TMGA_PLANIFICACIONES PLA
                INNER JOIN TMGA_MATRICULAS_DETALLES
                MTD ON PLA.PLA_CODIGO = MTD.PLA_CODIGO
                INNER JOIN TMGA_MATRICULAS MTR ON
                MTR.MTR_CODIGO = MTD.MTR_CODIGO AND INS_CODIGO = @INS_CODIGO AND
                MTR.PRA_CODIGO = @PRA_CODIGO
                INNER JOIN RMGA_PLANIFICACION_HORARIOS
                PH ON PH.PLA_CODIGO = PLA.PLA_CODIGO AND PH.HOR_CODIGO = @HOR_CODIGO
                INNER JOIN TMGA_MALLAS_DETALLES MAD ON
                MAD.DTM_CODIGO = PLA.DTM_CODIGO
            END
        END
    END

```

```

INNER JOIN TMGA_ASIGNATURAS ASI ON
ASI.ASI_CODIGO = MAD.ASI_CODIGO

--obtiene el momento y la
planificacion de la asignatura con la que se cruza

IF @MOM_CODIGO IS NULL OR
@MOM_CODIGO = 0
BEGIN
SET @MOM_CODIGO = 0
END

IF @MOM_CODIGO > 0
BEGIN
IF @MOM_CODIGO =
@MOM_CODIGO_MAT
BEGIN
--semanas 0 - 0
--semanas 1 - 4
--semanas 2 - 8
--semanas 3 - 16

IF (@SEMANAS != 1 AND
@MOM_CODIGO = 2)
BEGIN
CLOSE CruceHorarios
DEALLOCATE
COMMIT

SELECT 'La asignatura
seleccionada, tiene cruce de horarios con lo matriculado. Asignatura: ' +
@ASI_DESCRIPCION_CRUCE AS MENSAJE, 2 AS TIPO

UPDATE TMP_PARALELOS
WHERE INS_CODIGO =
AND PLA_CODIGO =
AND PRA_CODIGO =
RETURN 0
END
END
ELSE
BEGIN
CLOSE CruceHorarios
DEALLOCATE CruceHorarios
COMMIT

SELECT 'La asignatura
seleccionada, tiene cruce de horarios con lo matriculado. Asignatura: ' +
@ASI_DESCRIPCION_CRUCE AS MENSAJE, 2 AS TIPO

UPDATE TMP_PARALELOS SET
WHERE INS_CODIGO = @INS_CODIGO
AND PLA_CODIGO = @PLA_CODIGO
AND PRA_CODIGO = @PRA_CODIGO
RETURN 0
END
END
END

```

```

        FETCH NEXT FROM CruceHorarios
        INTO @HOR_CODIGO, @PLA_CODIGO_MAT
        END
        CLOSE CruceHorarios
        DEALLOCATE CruceHorarios
    END

    SELECT @MTR_CODIGO = MTR_CODIGO
    FROM TMGA_MATRICULAS
    WHERE INS_CODIGO = @INS_CODIGO
    --AND NIV_CODIGO = @MAX_NIVEL
    AND PRA_CODIGO = @PRA_CODIGO

    SELECT @REGISTROS_CABECERA = COUNT(1)
    FROM TMGA_MATRICULAS
    WHERE INS_CODIGO = @INS_CODIGO
    -- AND NIV_CODIGO = @MAX_NIVEL
    AND PRA_CODIGO = @PRA_CODIGO

    --*****CABECERA
    *****
    IF @REGISTROS_CABECERA = 0
    BEGIN
        --*****VERIFICA SI ESTA ACTIVO EL
        PARAMETRO PARA ENVIAR CORREO POR LA EMERGENCIA
        *****
        SELECT @ENVIAR_CORREO = PRM_ESTADO FROM TMGA_PARAMETROS_DATOS
        WHERE PRM_CODIGO = 13
        --
        *****
        *****
        *

        -- VERIFICO QUE TIPO DE MATRICULA ES ORDINARIA O EXTRA
    ORDINARIA
        SELECT @AUX = COUNT(1)
        FROM TMGA_CALENDARIOACAD_DETALLES CD
        INNER JOIN RMGA_CALENDARIODETALLE_TIPOMATRICULAS RCT ON
        CD.CLD_CODIGO = RCT.CLD_CODIGO
        INNER JOIN TMGA_CALENDARIOS_ACADEMICOS CA ON CA.CLA_CODIGO =
        CD.CLA_CODIGO
        INNER JOIN TMGA_TIPOS_MATRICULAS TM ON TM.TPM_CODIGO =
        RCT.TPM_CODIGO
        WHERE MOD_CODIGO = @MOD_CODIGO
        AND PRA_CODIGO = @PRA_CODIGO
        AND CONVERT(DATE,GETDATE()) BETWEEN CD.CLD_FECHA_INICIO AND
        CD.CLD_FECHA_FIN
        IF @AUX = 0
        BEGIN
            SET @TPM_CODIGO = 1 -- PONE POR DEFECTO ORDINARIA
            SET @TIPO_MATRICULA = 'O' -- PONE POR DEFECTO
        ORDINARIA
        END
        ELSE
        BEGIN
            SELECT DISTINCT @TPM_CODIGO = ISNULL(TM.TPM_CODIGO,2),
            @TIPO_MATRICULA = SUBSTRING(TM.TPM_DESCRIPCION,1,1)
            FROM TMGA_CALENDARIOACAD_DETALLES CD
            INNER JOIN RMGA_CALENDARIODETALLE_TIPOMATRICULAS RCT ON
            CD.CLD_CODIGO = RCT.CLD_CODIGO
            INNER JOIN TMGA_CALENDARIOS_ACADEMICOS CA ON CA.CLA_CODIGO
            = CD.CLA_CODIGO
            INNER JOIN TMGA_TIPOS_MATRICULAS TM ON TM.TPM_CODIGO =
            RCT.TPM_CODIGO
            WHERE MOD_CODIGO = @MOD_CODIGO
            AND PRA_CODIGO = @PRA_CODIGO

```

```

AND CONVERT (DATE, GETDATE ()) BETWEEN CD.CLD_FECHA_INICIO
AND CD.CLD_FECHA_FIN
END

if @TPE_CODIGO = 2
begin
set @TPM_CODIGO = 1
end
else
begin
--//COMENTAR EN CASO DE ERROR
IF @NIV_CODIGO = 1 OR @NIV_CODIGO = 0
BEGIN
-- IF (GETDATE() <= '21-04-2019 11:59:00')-- SE APLAZA
DOS SEMANAS PARA INSCRIPCIONES
-- BEGIN
ORDINARIA SET @TPM_CODIGO = 1 -- PONE POR DEFECTO
ORDINARIA SET @TIPO_MATRICULA = 'O' -- PONE POR DEFECTO
END
ELSE
BEGIN
IF @TPM_CODIGO = 0 OR @TPM_CODIGO IS NULL
BEGIN
SET @TPM_CODIGO = 1
SET @TIPO_MATRICULA = 'O'
END
END
end
-- ELSE
-- BEGIN
-- SET @TPM_CODIGO = 2 -- PONE POR DEFECTO
EXTRAORDINARIA SET @TIPO_MATRICULA = 'E' -- PONE POR DEFECTO
EXTRAORDINARIA
-- END
--END
--select @TPM_CODIGO, @TPE_CODIGO

--NUMERO DE REGISTROS
SELECT @REGISTROS = COUNT(*) FROM TMGA_MATRICULAS_DETALLES

-- GENERA LA SECUENCIA DEL NUMERO DE FORMULARIO
SELECT @NUM_FORMULARIO = 'UI - ' + ISNULL(@SIGLAS_CARRERA, 'SN
- SIGLA') + ' - ' + CONVERT(VARCHAR(20),@REGISTROS) + '-' +
ISNULL(@TIPO_MATRICULA, 'SN - SIGLA')

--VERIFICA QUE LA CABECERA TENGA REGISTROS

--*****--INSERTA EL DETALLE DE LA
CABECERA*****

--VERIFICA LA MAYOR CANTIDAD DE MATERIAS SEGUN EL
NIVEL PARA ASIGNAR

SELECT @MAX_NIVEL = MAX(NIV_CODIGO)
FROM TMP_PARALELOS
WHERE INS_CODIGO = @INS_CODIGO
AND PRA_CODIGO = @PRA_CODIGO
AND SN_MOSTRAR = 1

SELECT @MAX_NIVEL_EST = MAX(NIV_CODIGO)
FROM TMGA_MATRICULAS
WHERE INS_CODIGO = @INS_CODIGO
AND PRA_CODIGO = @PRA_CODIGO

```

```

= 0 end

if @MAX_NIVEL_EST is null begin set @MAX_NIVEL_EST

NULL)

IF (@MAX_NIVEL_EST > @MAX_NIVEL) OR (@MAX_NIVEL IS

BEGIN
    SET @MAX_NIVEL = @MAX_NIVEL_EST
END

IF (@NIV_CODIGO IS NULL) OR (@MAX_NIVEL_EST IS NULL )
BEGIN

    SELECT @MAX_NIVEL_EST = MAX(NIV_CODIGO)
    FROM TMGA_MATRICULAS
    WHERE INS_CODIGO = @INS_CODIGO
    AND PRA_CODIGO = @PRA_CODIGO

    IF @MAX_NIVEL_EST IS NULL
    BEGIN
        ROLLBACK TRANSACTION

        UPDATE TMP_PARALELOS SET OBSERVACIONES = 'No
se pudo verificar el nivel que le corresponde.'
        WHERE INS_CODIGO = @INS_CODIGO
        AND PLA_CODIGO = @PLA_CODIGO
        AND PRA_CODIGO = @PRA_CODIGO
        RETURN 0
    END
END

--POSGRADO SALE APROBADO PARA QUE APARESCA EN LISTADOS
IF @TPE_CODIGO = 2
BEGIN
    IF @MAX_NIVEL > 1
    BEGIN
        SET @MTR_LEGALIZADO = 1
        SET @FEC_LEGALIZADO = GETDATE()
    END
    ELSE
    BEGIN
        SET @MTR_LEGALIZADO = 0
        SET @FEC_LEGALIZADO = NULL
    END

    INSERT INTO TMGA_MATRICULAS
    VALUES (@MAX_NIVEL, @PRA_CODIGO, @TPM_CODIGO,
@INS_CODIGO, @NUM_FORMULARIO, 1, GETDATE(), @MTR_LEGALIZADO, @FEC_LEGALIZADO
,1, 0, NULL, 1, NULL, NULL, NULL, '')
    END
    ELSE
    BEGIN
        INSERT INTO TMGA_MATRICULAS
        VALUES (@MAX_NIVEL, @PRA_CODIGO, @TPM_CODIGO,
@INS_CODIGO, @NUM_FORMULARIO, 1, GETDATE(), 0, NULL, 0, 0, NULL, 1
, NULL, NULL, NULL, '')
    END

    SET @MTR_CODIGO = scope_identity()

END -- END DE CABECERA

-- SELECCIONA EL CODIGO DE LA CABECERA DE ESA MATRICULA
SELECT @MTR_CODIGO = MTR_CODIGO
FROM TMGA_MATRICULAS
WHERE INS_CODIGO = @INS_CODIGO
AND NIV_CODIGO = @MAX_NIVEL AND PRA_CODIGO = @PRA_CODIGO

```

```

--
*****
*****
--*****DETALLE
MATRICULA*****
***

--          SELECT 'MATRICULA DETALLE'
--          -- GRABA EL DETALLE DE LA ASIGNATURA SELECCIONADA
                SELECT @TIPO_MATERIA = TIPO_MATERIA
                FROM TMP_PARALELOS
                WHERE PLA_CODIGO = @PLA_CODIGO AND

INS_CODIGO = @INS_CODIGO

                IF @TIPO_MATERIA = 0
                BEGIN
                        SET @CONDICIONADO = 0
                END
                ELSE
                BEGIN
                        SET @CONDICIONADO = 1
                END

                --SELECCIONA LA ASIGNATURA PARA CONTAR EL
NUMERO DE MATRICULA

                SELECT @ASI_CODIGO_NUM = ASI.ASI_CODIGO
                FROM TMGA_PLANIFICACIONES PLA
                INNER JOIN TMGA_MALLAS_DETALLES MD ON

MD.DTM_CODIGO = PLA.DTM_CODIGO
                INNER JOIN TMGA_ASIGNATURAS ASI ON

ASI.ASI_CODIGO = MD.ASI_CODIGO
                INNER JOIN RMGA_PLANIFICACION_HORARIOS RH ON

RH.PLA_CODIGO = PLA.PLA_CODIGO
                WHERE PLA.PLA_CODIGO = @PLA_CODIGO

                ----NUMERO DE MATRICULA EN ESA MATERIA
                --SELECT @NUM_MATRICULA = COUNT(*)
                --FROM TMGA_MATRICULAS_DETALLES MD
                --INNER JOIN TMGA_MATRICULAS M ON
MD.MTR_CODIGO = M.MTR_CODIGO AND INS_CODIGO = @INS_CODIGO
                --INNER JOIN TMGA_PLANIFICACIONES PLA ON

PLA.PLA_CODIGO = MD.PLA_CODIGO
                --INNER JOIN TMGA_MALLAS_DETALLES MAD ON

MAD.DTM_CODIGO = PLA.DTM_CODIGO
                --INNER JOIN TMGA_ASIGNATURAS ASI ON

MAD.ASI_CODIGO = ASI.ASI_CODIGO AND ASI.ASI_CODIGO = @ASI_CODIGO_NUM

                SELECT @NUM_MATRICULA = COUNT(ASI.ASI_CODIGO)
                FROM TMGA_MATRICULAS_DETALLES MD,
TMGA_MATRICULAS M, TMGA_PLANIFICACIONES PLA, TMGA_ASIGNATURAS ASI,
TMGA_MALLAS_DETALLES MAD,

                TMGA_INSCRIPCIONES TI
                WHERE MD.MTR_CODIGO = M.MTR_CODIGO
                AND PLA.PLA_CODIGO = MD.PLA_CODIGO
                AND MAD.DTM_CODIGO = PLA.DTM_CODIGO
                AND MAD.ASI_CODIGO = ASI.ASI_CODIGO
                AND ASI.ASI_CODIGO = @ASI_CODIGO
                AND TI.INS_CODIGO = M.INS_CODIGO
                AND TI.PER_CODIGO = @PER_CODIGO
                AND MTR_RETIRADO = 0

                IF @NUM_MATRICULA = 0
                BEGIN
                        SET @NUM_MATRICULA = 1
                END

```

```

--INSERTA LA MATERIAS POR DETALLE
INSERT INTO TMGA_MATRICULAS_DETALLES
VALUES (@PRL_CODIGO,
@PLA_CODIGO,@MTR_CODIGO,'N',GETDATE(),@NUM_MATRICULA,@CONDICIONADO,0,NULL,')

--OBTIENE EL CODIGO DE MATRICULA DETALLE
SET @MTD_CODIGO = scope_identity()
-- SELECT @MTD_CODIGO
-- SELECT * FROM TMGA_MATRICULAS_DETALLES
WHERE MTD_CODIGO = @MTD_CODIGO
-- SELECT @PRL_CODIGO,
@PLA_CODIGO,@MTR_CODIGO,'N',GETDATE(),@NUM_MATRICULA,@CONDICIONADO,0,NULL,')

UPDATE TMP_PARALELOS SET SN_MOSTRAR = 0
,SN_MATRICULADO = 1, SN_ESTADO = 0
WHERE INS_CODIGO = @INS_CODIGO
AND PLA_CODIGO = @PLA_CODIGO

SELECT @ASI_CODIGO = ASI_CODIGO
FROM TMGA_PLANIFICACIONES PLA
INNER JOIN TMGA_MALLAS_DETALLES MD ON
MD.DTM_CODIGO = PLA.DTM_CODIGO
WHERE PLA_CODIGO = @PLA_CODIGO

UPDATE TMP_PARALELOS SET SN_MOSTRAR = 0
WHERE ASI_CODIGO = @ASI_CODIGO
AND INS_CODIGO = @INS_CODIGO

IF @TPE_CODIGO = 1 --GRADO
BEGIN
EXEC SP_GENERAR_NOTAS_MATRICULADOS_POR_ASIGNATURA
@PRA_CODIGO,@INS_CODIGO,@MTD_CODIGO,@PLA_CODIGO
END
IF @TPE_CODIGO = 2 --POSGRADO
BEGIN
EXEC SP_GENERAR_NOTAS_MATRICULADOS_POR_ASIGNATURA_POSGRADO
@PRA_CODIGO,@INS_CODIGO,@MTD_CODIGO,@PLA_CODIGO
END

--*****GENERA
VALORES*****
*****

SELECT @MTR_CODIGO = MTR_CODIGO,
@NIV_CODIGO = NIV_CODIGO
FROM TMGA_MATRICULAS
WHERE INS_CODIGO = @INS_CODIGO
AND PRA_CODIGO = @PRA_CODIGO

EXEC SP_SAVE_DETALLE_PENSIONES_BLOQUES_AUT_ASI
@INS_CODIGO = @INS_CODIGO, @NIV_CODIGO = @NIV_CODIGO,
@TIPO_MATERIA = 0, @MTR_CODIGO = @MTR_CODIGO

--
*****
*****

--*****VERIFICA
BECA*****

EXEC SP_VERIFICA_BECA_AUTOMATICA_ESTUDIANTES @INS_CODIGO,@PRA_CODIGO,0,
@MENSAJE = @MENSAJE_BECA OUTPUT

UPDATE TMGA_DETALLE_PENSIONES_CONFIGURACIONES SET DPC_BECA_ESTADO = 1
FROM TMGA_DETALLE_PENSIONES_CONFIGURACIONES DPC
INNER JOIN TMGA_PENSIONES_CONFIGURACIONES PC ON DPC.PNC_CODIGO =
PC.PNC_CODIGO
WHERE DPC.DPC_PORCENTAJE_BECAS > 0

```

```

AND DPC_BECA_ESTADO = 0
AND PC.PRA_CODIGO = @PRA_CODIGO

--*****VERIFICA CURSOS IBEC Y CISCO
PARA ASIGNAR VALORES AUTOMATICO*****

IF EXISTS (SELECT 1
FROM TMGA_PENSIONES_CONFIGURACIONES_CERTIFICACIONES PC
WHERE ASI_CODIGO = @ASI_CODIGO
AND PRA_CODIGO = @PRA_CODIGO
)
BEGIN
-- VERIFICA SI NO TOMO ANTES ESA CERTIFICACION

SELECT @AUX = COUNT(1)
FROM TMGA_CERTIFICACIONES_CURSOS CC
INNER JOIN TMGA_MATRICULAS_DETALLES MD ON CC.MTD_CODIGO =
MD.MTD_CODIGO
INNER JOIN TMGA_MATRICULAS MTR ON MTR.MTR_CODIGO =
MD.MTR_CODIGO
INNER JOIN TMGA_PLANIFICACIONES PLA ON PLA.PLA_CODIGO =
MD.PLA_CODIGO
INNER JOIN TMGA_MALLAS_DETALLES MAD ON MAD.DTM_CODIGO =
PLA.DTM_CODIGO
WHERE MTR.INS_CODIGO = @INS_CODIGO
AND MTR.PRA_CODIGO < @PRA_CODIGO
AND ASI_CODIGO = @ASI_CODIGO
AND CC.CCS_ESTADO IN (1,2) -- 1 YA TIENE APROBADA LA
CERTIFICACION - 2 NO TIENEN IGRESADAD LA NOTA
IF @AUX > 0
BEGIN
COMMIT TRANSACTION
RETURN 0
END

SELECT @AUX = COUNT(1)
FROM TMGA_CERTIFICACIONES_CURSOS CC
INNER JOIN TMGA_MATRICULAS_DETALLES MD ON CC.MTD_CODIGO =
MD.MTD_CODIGO
INNER JOIN TMGA_MATRICULAS MTR ON MTR.MTR_CODIGO =
MD.MTR_CODIGO
INNER JOIN TMGA_PLANIFICACIONES PLA ON PLA.PLA_CODIGO =
MD.PLA_CODIGO
INNER JOIN TMGA_MALLAS_DETALLES MAD ON MAD.DTM_CODIGO =
PLA.DTM_CODIGO
WHERE MTR.INS_CODIGO = @INS_CODIGO
AND MTR.PRA_CODIGO < @PRA_CODIGO
AND ASI_CODIGO = @ASI_CODIGO
AND CC.CCS_ESTADO IN (1,2) -- 1 YA TIENE APROBADA LA
CERTIFICACION - 2 NO TIENEN IGRESADAD LA NOTA

IF @AUX > 0
BEGIN
COMMIT TRANSACTION
RETURN 0
END

--SACA EL NOMBRE DE LA ASIGNATURA
SELECT @DESCRIPCION_CERTIFICACION = CASE WHEN TCE_CODIGO = 1
THEN REPLACE(PCC_DESCRIPCION, 'IBEC-', '')
ELSE REPLACE(PCC_DESCRIPCION, 'CISCO-', '') END
FROM TMGA_PENSIONES_CONFIGURACIONES_CERTIFICACIONES PC
INNER JOIN TMGA_TIPOS_CERTIFICACIONES TTC ON TTC.TFC_CODIGO =
PC.TFC_CODIGO
WHERE ASI_CODIGO = @ASI_CODIGO
AND PRA_CODIGO = @PRA_CODIGO

```



```

-- select @DESCRIPCION_CERTIFICACION, @PLA_CODIGO

select @AUX = count(1)
FROM TMGA_PENSIONES_CONFIGURACIONES_CERTIFICACIONES PCC
INNER JOIN TMGA_PENSIONES_CONFIGURACIONES PC ON 'A' +
CONVERT(VARCHAR(20),PCC.PCC_CODIGO) = PC.PNC_GRUPO
INNER JOIN TMGA_DETALLE_PENSIONES_CONFIGURACIONES DPC ON
DPC.PNC_CODIGO = PC.PNC_CODIGO
INNER JOIN TMGA_INSCRIPCIONES TI ON TI.INS_CODIGO =
DPC.INS_CODIGO
AND PC.PRA_CODIGO <= @PRA_CODIGO
AND TI.PER_CODIGO = @PER_CODIGO
AND PCC.PCC_DESCRIPCION LIKE '%' + @DESCRIPCION_CERTIFICACION
+ '%'

if @aux > 0
BEGIN
    select 1

    COMMIT TRANSACTION
    RETURN 0
END

--VERIFICA SI NO TIENE CARGADO ESE CURSO Y ASIGNADO EL VALOR
IF NOT EXISTS ( SELECT 1
FROM TMGA_PENSIONES_CONFIGURACIONES_CERTIFICACIONES PCC
INNER JOIN TMGA_PENSIONES_CONFIGURACIONES PC ON 'A' +
CONVERT(VARCHAR(20),PCC.PCC_CODIGO) = PC.PNC_GRUPO
INNER JOIN TMGA_DETALLE_PENSIONES_CONFIGURACIONES DPC ON
DPC.PNC_CODIGO = PC.PNC_CODIGO
AND PC.PRA_CODIGO = @PRA_CODIGO --EN EL PERIODO ACTUAL
AND INS_CODIGO = @INS_CODIGO
)
BEGIN

    SELECT @PCC_CODIGO = ISNULL(PCC_CODIGO,0)
    FROM TMGA_PENSIONES_CONFIGURACIONES_CERTIFICACIONES PCC
    WHERE ASI_CODIGO = @ASI_CODIGO
    AND PRA_CODIGO = @PRA_CODIGO

    IF @PCC_CODIGO IS NOT NULL OR @PCC_CODIGO != 0
    BEGIN
        EXEC
        SP_CONSULTA_ESTUDIANTES_CERTIFICACIONES_INDIVIDUAL @PRA_CODIGO, @ASI_CODIGO,
        @PCC_CODIGO
    END

    SELECT @TMP_CODIGO = ISNULL(TMP_CODIGO, '')
    FROM TMP_ESTUDIANTES_CERTIFICACIONES_INDIVIDUAL
    WHERE PCC_CODIGO = @PCC_CODIGO
    AND PRA_CODIGO = @PRA_CODIGO
    AND ASI_CODIGO = @ASI_CODIGO
    AND INS_CODIGO = @INS_CODIGO
    IF @TMP_CODIGO IS NOT NULL OR @TMP_CODIGO != 0
    BEGIN

        select @MOM_CODIGO = MOM_CODIGO from
        TMGA_PLANIFICACIONES_MODULAR where PLA_CODIGO = @PLA_CODIGO
        IF @MOM_CODIGO = 1
        BEGIN
            SET @FECHA_PAGO = '20201105'
        END
        IF @MOM_CODIGO = 2
        BEGIN
            SET @FECHA_PAGO = '20201205'
        END
        IF @MOM_CODIGO = 3
        BEGIN

```

```

        SET @FECHA_PAGO = '20210105'
    END
    IF @MOM_CODIGO = 0
    BEGIN
        SET @FECHA_PAGO = '20201105'
    END

    EXEC
    SP_GENERAR_VALORES_CERTIFICACIONES_INDIVIDUAL_MATRICULAS
    @TMP_CODIGO,1,@PER_CODIGO, @FECHA_PAGO

    --ACTUALIZA EL CURSO CON 100% DE BECA

    UPDATE TMGA_DETALLE_PENSIONES_CONFIGURACIONES SET
    DPC_PORCENTAJE_BECAS = 100, DPC_BECA_ESTADO = 1
    FROM TMGA_PENSIONES_CONFIGURACIONES_CERTIFICACIONES
    PCC
    INNER JOIN TMGA_PENSIONES_CONFIGURACIONES PC ON 'A' +
    CONVERT(VARCHAR(20),PCC.PCC_CODIGO) = PC.PNC_GRUPO
    INNER JOIN TMGA_DETALLE_PENSIONES_CONFIGURACIONES DPC
    ON DPC.PNC_CODIGO = PC.PNC_CODIGO
    AND PC.PRA_CODIGO = @PRA_CODIGO --EN EL PERIODO ACTUAL
    AND INS_CODIGO = @INS_CODIGO
    AND PC.PNC_GRUPO = 'A122'

    END
END
END

--
*****
*****

    SELECT 'Se ha matriculado correctamente en la asignatura: ' +
    @ASI_DESCRIPCION AS MENSAJE, 1 AS TIPO, @MENSAJE_BECA AS 'MENSAJE_BECA',
    @ENVIAR_CORREO AS 'ENVIAR_CORREO';

    COMMIT TRANSACTION;

    -- LOGICA PARA MATRICULAR EN MOODLE
    -- SE OBTIENE LA IDENTIFICACION

    SELECT
    @IDENTIFICACION = TP.PER_IDENTIFICACION,
    @MDL_PRE_POS = TI.TPE_CODIGO
    FROM TMGA_PERSONAS TP
    INNER JOIN TMGA_INSCRIPCIONES TI ON TP.PER_CODIGO =
    TI.PER_CODIGO AND TI.INS_CODIGO= @INS_CODIGO;

    IF @MDL_PRE_POS = 1 --MATRICULA EN MOODLE PREGRADO
    BEGIN
        -- VALIDA SI LA MATERIA EXISTE EN MOODLE PREGRADO
        SELECT @MDL_FLAG = COUNT(1)
        FROM MOODLE_PRE.[posgrado].[public].mdl_course
        WHERE mdl_course.visible =1
        and mdl_course.enablecompletion=1
        and mdl_course.idnumber=cast(@PLA_CODIGO as NVARCHAR);

        IF @MDL_FLAG >0
        BEGIN
            --MATRICULA EN MOODLE GRADO
            EXEC [dbo].[SPI_MATRICULA_ESTUDIANTE_MOODLE]
            @IDENTIFICACION, @PLA_CODIGO , 1;
        END
        ELSE
        BEGIN
            PRINT ' PREGRADO: La materia no existe en moodle...';
        END
    END

```

```

END
ELSE --MATRICULA EN MOODLE POSGRADO
BEGIN
-- VALIDA SI LA MATERIA EXISTE EN MOODLE POSGRADO
SELECT @MDL_FLAG = COUNT(1)
FROM MOODLE_POS.[posgrado].[public].mdl_course
WHERE mdl_course.visible =1
and mdl_course.enablecompletion=1
and mdl_course.idnumber=cast(@PLA_CODIGO as NVARCHAR);

IF @MDL_FLAG >0
BEGIN
--MATRICULA EN MOODLE GRADO
PRINT 'POSGRADO: inicio...';
EXEC [dbo].[SPI_MATRICULA_ESTUDIANTE_MOODLE]
@IDENTIFICACION, @PLA_CODIGO , 2;
PRINT 'POSGRADO: fin...';
END
ELSE
BEGIN
PRINT 'POSGRADO: La materia no existe en moodle...';
END
END
RETURN 0

END TRY
BEGIN CATCH
SELECT @ERROR_NUMBER = ERROR_NUMBER(),
@ERROR_PROC = ERROR_PROCEDURE(),
@ERROR_MESSAGE = ERROR_MESSAGE()
RAISERROR( 'ERROR: ( %D ) EN %S. %S ', 16, 1, @ERROR_NUMBER,
@ERROR_PROC, @ERROR_MESSAGE )
IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION
END CATCH
GO

```