



FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas

“APLICATIVO INTEGRADOR EN EL PROCESO DE
MATRÍCULA PARA LA UNIVERSIDAD PRIVADA DEL
NORTE”

Trabajo de suficiencia profesional para optar el título
profesional de:

Ingeniero de Sistemas

Autor:

Bach. Carlos Fernando Villacorta Padilla

Asesor:

Ing. Pedro Gilmer Castillo Domínguez

Trujillo - Perú

2021

DEDICATORIA

A Dios, toda la gloria sea a Él.

A mis padres por su apoyo incondicional
e impulso para seguir adelante.

AGRADECIMIENTO

A Dios, gracias a Él por la vida y sustento que me permitieron mantenerme durante la presente pandemia de COVID-19 pudiendo realizar el presente trabajo.

A mis padres, por el sacrificio, cuidado en cada enseñanza que me pudieron dar.

A mis hermanos, por ser ejemplos para nunca rendirme frente a las adversidades.

A mi asesor Ing. Pedro Castillo Domínguez, por haberme brindado la oportunidad de recurrir a su experiencia para guiarme en el presente trabajo.

A todos mis docentes, sin su esfuerzo y desapego mi carrera hubiera sido en vano.

Tabla de contenidos

DEDICATORIA.....	2
AGRADECIMIENTO	3
ÍNDICE DE TABLAS	6
ÍNDICE DE FIGURAS	7
ANEXOS	8
RESUMEN EJECUTIVO	9
CAPÍTULO I. INTRODUCCIÓN	10
1. EXPERIENCIA PROFESIONAL	10
2. DESCRIPCIÓN DE LA EMPRESA.....	10
2.1. Razón social	10
2.2. Nombre de la Empresa	10
2.3. Giro de la Empresa.....	10
2.4. RUC.....	10
2.5. Inicio de Operaciones.....	11
2.6. Representantes Legales	11
2.7. Sobre nosotros.....	11
2.8. Misión	11
2.9. Visión	11
2.10. Nuestros valores	12
2.11. Ubicación geográfica.....	12
2.12. Organigrama del área de TI:.....	13
3. INFRAESTRUCTURA TECNOLÓGICA	14
3.1. Arquitectura de CRM Dynamics 2011 UPN.....	14
3.2. Arquitectura de Integración Banner con Aplicaciones UPN locales.....	15
CAPÍTULO II. MARCO TEÓRICO	16
➤ PROCESO DE MATRÍCULA	16
➤ FUNDAMENTOS DE PROGRAMACIÓN.....	16
➤ NET FRAMEWORK.....	17
➤ APLICACIÓN WEB	17
➤ SERVICIO DE WINDOWS	17
➤ NODE.JS	18
➤ SOA	18
➤ CRM	19
➤ CRM DYNAMICS 2011	19
➤ MESSAGE BROKER.....	21
➤ PROTOCOLO AMQP	21
➤ RABBITMQ.....	22
➤ SIS	22
➤ BANNER	22
➤ SCRUM.....	23
CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA.....	24
1. DESCRIPCIÓN DE LA PROBLEMÁTICA.....	24
2. JUSTIFICACIÓN	26
2.1. Justificación social:	26
2.2. Justificación económica:	26
2.3. Justificación tecnológica:.....	26
2.4. Justificación práctica:	27
3. LIMITACIONES.....	27

4.	RESTRICCIONES	27
4.1.	<i>Tiempo de entrega:</i>	27
4.2.	<i>Pruebas funcionales:</i>	27
5.	REQUISITOS FUNCIONALES Y NO FUNCIONALES	27
5.1.	<i>RabbitMQ</i>	27
5.2.	<i>Servicio Web</i>	28
5.3.	<i>Servicio de consumidor</i>	29
6.	ANÁLISIS Y DISEÑO	29
6.1.	<i>Servicio Web</i>	29
6.2.	<i>Servicio consumidor</i>	30
7.	DIAGRAMAS DE FLUJO ORIENTADO AL PROCESO, DISEÑO DE INTERFACES	30
8.	MODELO DE DATOS.....	32
9.	IMPLEMENTACIÓN	33
9.1.	<i>Servicio Web</i>	33
9.2.	<i>Servicio consumidor</i>	34
10.	DIAGRAMA DE FLUJO ALGORITMO ORIENTADO A LA IMPLEMENTACIÓN	37
10.1.	<i>Flujo de formulario de CRM</i>	37
10.2.	<i>Flujo de Servicio Web</i>	38
10.3.	<i>Servicio consumidor</i>	40
11.	DIAGRAMA DE COMPONENTES	40
12.	DIAGRAMA DE DESPLIEGUE	41
13.	PRUEBAS	41
13.1.	<i>Pruebas de performance</i>	41
13.2.	<i>Pruebas funcionales y de integración</i>	42
14.	OBJETIVO GENERAL	42
15.	OBJETIVOS ESPECÍFICOS	42
16.	IMPLANTACIÓN	42
CAPÍTULO IV. RESULTADOS		46
1.	CANTIDAD DE INCIDENTES	46
1.1.	<i>Incidentes CRM-FOX</i>	47
1.2.	<i>Incidentes CRM-Banner</i>	48
1.3.	<i>Disminución de incidentes</i>	49
2.	TIEMPOS DE ATENCIÓN	50
2.1.	<i>Tiempos de resolución tickets CRM-FOX</i>	50
2.2.	<i>Tiempos de resolución tickets CRM-Banner</i>	51
2.3.	<i>Tiempos de resolución tickets CRM-FOX</i>	52
3.	ACUMULADO DE HORAS INVERTIDAS EN ATENCIÓN DE INCIDENTES.....	53
3.1.	<i>Acumulado de horas invertidas en atención de incidentes en integración CRM-FOX</i>	53
3.2.	<i>Acumulado de horas invertidas en atención de incidentes en integración CRM-Banner</i>	54
3.3.	<i>Disminución porcentual de horas invertidas para la atención de incidentes</i>	55
CAPÍTULO V. CONCLUSIONES.....		56
REFERENCIAS		58
ANEXOS		59

ÍNDICE DE TABLAS

Tabla 1 - Representantes legales Universidad Privada del Norte S.A.C.	11
Tabla 2 - Descripción de Servidores de Arquitectura CRM Dynamics 2011 UPN	15

ÍNDICE DE FIGURAS

Ilustración 1 - Organigrama del área de Sistemas	13
Ilustración 2 - Arquitectura de CRM Dynamics 2011 UPN	14
Ilustración 3 - Arquitectura de Integración Banner con Aplicaciones UPN locales	15
Ilustración 5 - Captura de CRM Dynamics 2011	20
Ilustración 6 - Ventana previa al generar un archivo CSV	24
Ilustración 7 - Servicio Web entregando de archivo CSV de exportación con FOX.....	25
Ilustración 8 - Contenido del archivo CSV.....	25
Ilustración 9 - Interfaz de CRM con botón de integración	29
Ilustración 10 - Captura del componente de Servicio Web	30
Ilustración 11 - Proceso de envío de alumno a Banner.....	30
Ilustración 12 - Proceso de levantamiento de retenciones por documentos en Banner	31
Ilustración 13 - Diagrama de datos de entidades usadas de CRM.....	32
Ilustración 14 - Componentes de Integración SIS-CRM.....	33
Ilustración 15 - Interfaz desarrollada con .NET WebForms.....	34
Ilustración 16 - Captura de código de tarea consumidora.....	34
Ilustración 17 - Captura de código tarea consumidora	35
Ilustración 18 - Captura de historial de cambios guardando el ID de estudiante en CRM	35
Ilustración 19 - Captura de correo de notificación satisfactoria	36
Ilustración 20 - Captura de correo de notificación de error en el proceso	36
Ilustración 21 - Flujo de comportamiento nuevo botón en formulario CRM	37
Ilustración 22 - Flujo botón 1: Nuevo Alumno	38
Ilustración 23 - Flujo botón 2: Actualización Documentos	39
Ilustración 24 - Flujo botón 3: Actualización Resultado Exámenes.....	39
Ilustración 25 - Flujo servicio consumidor.....	40
Ilustración 26 - Diagrama de Componentes de la solución	40
Ilustración 27 - Diagrama de despliegue de solución	41
Ilustración 28 - Captura de pruebas de performance	42
Ilustración 32 - Captura configuración colas RabbitMQ.....	43
Ilustración 33 - Captura configuración colas RabbitMQ.....	43
Ilustración 34 - Configuración Shovel en RabbitMQ.....	43
Ilustración 35 - Despliegue de aplicación web en IIS de CRM.....	44
Ilustración 36 - Servicio consumidor como Servicio de Windows.....	45
Ilustración 37 - Cantidad de incidentes CRM-FOX por semestre calendario.....	47
Ilustración 38 - Cantidad incidentes CRM-Banner por semestre calendario	48
Ilustración 39 - Disminución de incidentes en % por semestres calendario espejo.....	49
Ilustración 40 - Tiempo de atención promedio de incidentes en horas por semestre Integración CRM-FOX	50
Ilustración 41 - Tiempo de atención promedio de incidentes en horas por semestre Integración CRM-Banner	51
Ilustración 42 - Disminución en tiempos de atención en % por semestres calendario espejo	52
Ilustración 43 - Acumulado de horas para resolución de incidentes por semestre Integración CRM-FOX	53
Ilustración 44 - Acumulado de horas para resolución de incidentes por semestre Integración CRM-Banner.	54
Ilustración 45 - Disminución de horas en resolución de incidentes en % por semestres calendario espejo Integraciones CRM-Banner / CRM-FOX.....	55

ANEXOS

Anexo 1 - Diagrama de flujo de proceso de matrícula UPN versión 7.....	59
Anexo 2 - Diagrama de componentes de CRM Dynamics 2011	60
Anexo 3 - Caso de pruebas funcionales.....	62

RESUMEN EJECUTIVO

El presente trabajo se centra en la implementación de un servicio Message Broker y el desarrollo de las interfaces para la integración entre el sistema CRM Dynamics y el Sistema académico Banner de la empresa Universidad Privada del Norte S.A.C.

La finalidad del trabajo radica en la disminución de errores, mejora en los tiempos de atención en el proceso de matrícula y modernización tecnológica del proceso de matrícula de la universidad.

La solución de manera integral fue desarrollada, probada y puesta en producción. Hoy en día se sigue utilizando y fue mejorada para ser integrada con otros sistemas de la universidad.

Este informe presenta el desarrollo inicial de esta solución que responde a la demanda por el crecimiento de la universidad, cubriendo la deuda tecnológica que se tenía con el antiguo sistema académico FOX.

CAPÍTULO I. INTRODUCCIÓN

1. Experiencia Profesional

Mi experiencia en la empresa educativa Universidad Privada del Norte inicia en agosto de 2015 con actividades de mantenimiento a sistemas web desarrollados en lenguajes VB .NET y C# .NET. Luego fue puesto a mi cargo el mantenimiento del sistema de ventas por excelencia de Microsoft llamado Microsoft CRM Dynamics 2011 que había sido implementado en el 2013 en la institución. Habiendo llegado a tener un conocimiento avanzado del sistema, desarrollaba módulos en lenguaje C# y Javascript que modificaban el comportamiento habitual del sistema para adaptarnos a las exigencias cambiantes del negocio. Dada la experiencia adquirida en este sistema también fui involucrado en la integración de este sistema CRM junto con el nuevo sistema académico Banner. Para esto fui retado por mi jefatura a desarrollar módulos de integración que se comuniquen con Banner por medio de un Message Broker (Sistema de mensajes por colas). Esto trae ventajas adicionales para poder integrar la información de muchos otros sistemas del ecosistema de aplicaciones de la Universidad Privada del Norte. Para cumplir el reto presentado, pude aplicar muchos de los conocimientos obtenidos durante mis estudios universitarios sobre desarrollo de aplicaciones web con el ecosistema de herramientas de Microsoft lo que me permitió adaptarme rápidamente con uno de sus productos líderes como CRM Dynamics.

2. Descripción de la empresa

2.1. Razón social

Universidad Privada del Norte S.A.C.

2.2. Nombre de la Empresa

Universidad Privada del Norte

2.3. Giro de la Empresa

Enseñanza superior

2.4. RUC

20215276024

2.5. Inicio de Operaciones

5 de enero de 1994

2.6. Representantes Legales

Nombre	Cargo
Ramos Sanguinetti, Alvaro	Director
Flores Garcés, Oscar Hernán	Gerente general
Velarde Talleri, Andrés René José	Apoderado
Anamaría Sánchez, Kevin	Director

Tabla 1 - Representantes legales Universidad Privada del Norte S.A.C.

Fuente: www.sunat.gob.pe

2.7. Sobre nosotros

“COMPROMETIDOS CON EL DESARROLLO DEL PAÍS”

Formamos la generación que ya está cambiando el mundo

Con una idea clara desde 1994, queremos seguir formando profesionales dueños de su destino cumpliendo nuestro compromiso con la educación, el desarrollo social y la formación integral. Nuestros valores se ven reflejados en las bases de nuestra institución, así como en sus principales autoridades y nuestro modelo educativo.

Somos parte de Laureate International Universities. Una red de instituciones de educación superior con importante presencia en Latinoamérica y reconocida globalmente por ser una entidad experta en gestión educacional.”

2.8. Misión

“Ser reconocidos como una universidad de excelente calidad académica y accesible, que forma profesionales dueños de su destino que se desempeñan exitosamente en un entorno global”.

2.9. Visión

“Transformar la vida de nuestros estudiantes mediante procesos educativos innovadores que privilegian el aprendizaje, el pensamiento crítico y el espíritu emprendedor, permitiéndoles contribuir al desarrollo sostenible de nuestra sociedad y al logro de sus objetivos de vida”.

2.10. Nuestros valores

Nuestra institución tiene el propósito de constituir una comunidad educativa basada en valores, tales como:

- **Trascendencia:** enfocamos nuestros esfuerzos y procesos en transformar positivamente la vida de nuestros estudiantes, colaboradores y su entorno, buscando formar egresados sobresalientes y exitosos para nuestra sociedad.
- **Compromiso:** ponemos pasión en cada cosa que hacemos, viviendo nuestros valores día a día y enfocando nuestros esfuerzos hacia el logro de nuestra misión.
- **Integridad:** la coherencia que existe entre lo que decimos y hacemos y el respeto hacia los demás ante cualquier circunstancia son la base de la confianza en todos nuestros niveles.
- **Trabajo en equipo:** construimos equipos de alto desempeño porque sabemos que, escuchando, valorando opiniones, sumando talento y compromiso lograremos resultados extraordinarios.
- **Innovación:** constantemente buscamos soluciones nuevas y prácticas para contribuir al éxito de nuestra institución.
- **Orientación a resultados:** nuestros resultados se basan en establecer objetivos desafiantes, superar los obstáculos y motivar el logro de estos.

2.11. Ubicación geográfica

2.11.1. Campus Trujillo

- El Molino: Av. El Ejercito 920 – Urb. El Molino
- San Isidro: Calle 31 - Urb. San Isidro 2da Etapa

2.11.2. Campus Lima

- Breña: Av. Tingo María 1122, Cercado de Lima
- Chorrillos: Av. Guardia Peruana 890, Chorrillos
- Comas: Av. Andrés Belaunde cdra. 10 s/n, Comas
- Los Olivos: Av. Alfredo Mendiola 6062, Los Olivos
- San Juan de Lurigancho: Av. El Sol 461 San Juan de Lurigancho

2.11.3. Cajamarca

- Cajamarca: Vía de Evitamiento s/n cuadra 15

2.12. Organigrama del área de TI:



Ilustración 1 - Organigrama del área de Sistemas

Fuente: Elaboración propia

3. Infraestructura tecnológica

3.1. Arquitectura de CRM Dynamics 2011 UPN

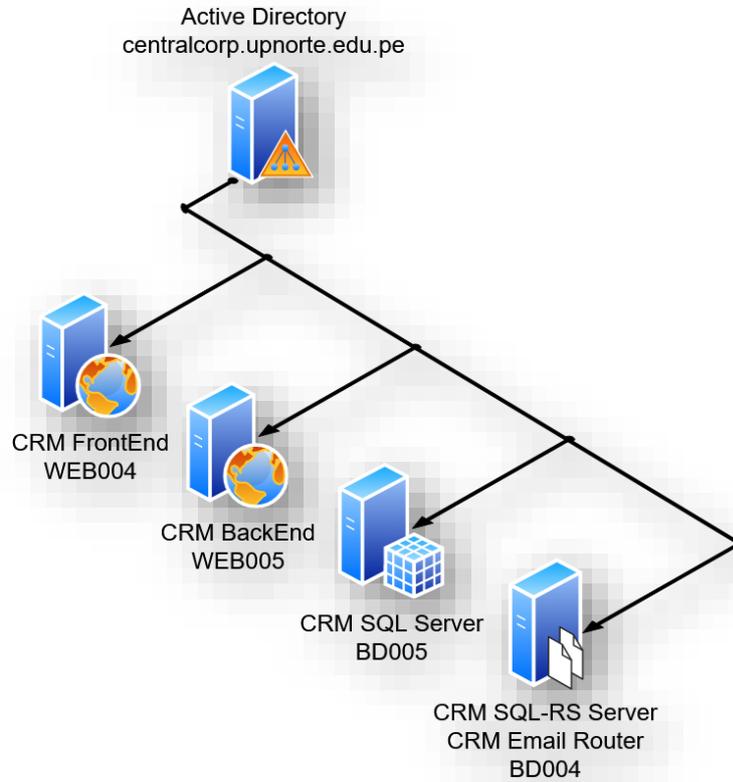


Ilustración 2 - Arquitectura de CRM Dynamics 2011 UPN

Fuente: Elaboración propia

Servidor	Rol
DC UPNORTE	Controladores de dominio y Active Directory CENTRALCORP.UPNORTE.EDU.PE. brindan servicios de autenticación y seguridad para CRM.
WEB004	Servidor FrontEnd de CRM, otorga servicios Web y Aplicación para CRM, Interactúa directamente con Usuarios
WEB005	Servidor BackEnd de CRM, otorga servicios de Aplicación y Hostea Servicios nativos de CRM, Interactúa directamente con Capa de Aplicación y de Datos.
BD005	Servidor de Base de datos, a través de los servicios de SQL Server Engine, otorga soporte y hosting para las BD's de Organización y configuración de CRM.

BD004	<p>Servidor de Reportes de CRM y CRM Email Router, a través de los servicios de SQL Server Reporting Services otorga capacidades de visualización de Reportes para CRM.</p> <p>Asimismo, a través del conector de CRM Email Router entrega una interfaz de conexión con el Servidor de Correo de UPN, para permitir a CRM la salida de mensajería electrónica.</p>
--------------	--

Tabla 2 - Descripción de Servidores de Arquitectura CRM Dynamics 2011 UPN

Fuente: Documento de arquitectura de solución CRM Dynamics 2011 UPN

3.2. Arquitectura de Integración Banner con Aplicaciones UPN locales

La arquitectura que soporta Banner propone implementar un Message Broker (Sistema de Colas) en nuestra red de servidores para poder comunicar Banner con otros sistemas de UPN.

El presente trabajo se centra en la implementación del servicio de Message Broker y el desarrollo de las interfaces para la integración entre el sistema CRM Dynamics y un nuevo sistema académico llamado Banner para UPN. Para esto, desarrollé módulos integradores que permiten comunicar estos dos sistemas junto con las configuraciones al servicio de Message Broker RabbitMQ.

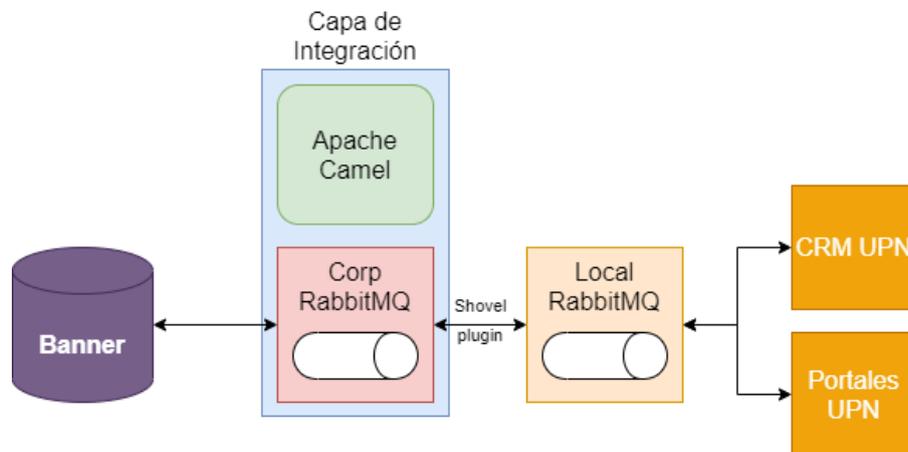


Ilustración 3 - Arquitectura de Integración Banner con Aplicaciones UPN locales

Fuente: Elaboración propia

CAPÍTULO II. MARCO TEÓRICO

➤ **Proceso de matrícula**

Según la Universidad Privada del Norte (Universidad Privada del Norte S.A.C., 2019) el proceso consiste en “asegurar que los postulantes a la universidad sean admitidos como estudiantes y se cumplan con los criterios de selección previamente establecidos por la Universidad.”

Dentro de este proceso, se involucra la participación del sistema CRM Dynamics como sistema de partida de la admisión del postulante.

Parte de la información de los postulantes almacenada en CRM Dynamics, es migrada al sistema académico de la institución llamado Banner.

Habiendo recibido esta información en Banner, es aquí donde se realiza la admisión del estudiante bajo la modalidad de ingreso, carrera universitaria y periodo académico deseado.

Más detalles sobre el flujo del proceso se puede encontrar en el Anexo 1 - Diagrama de flujo de proceso de matrícula UPN versión 7.

➤ **Fundamentos de programación**

El programar es definir un conjunto de instrucciones ordenadas para poder controlar un dispositivo. Estas instrucciones son ejecutadas por el centro de procesamiento del dispositivo y este a su vez puede estar conectado a otros dispositivos a ser controlados como: motores, pantallas gráficas, interfaces humanas (ratón, teclado), etc.

Según (Aguilar, 2008), la creación de un programa o software es esencialmente un proceso creativo que consiste en seguir una serie de fases o pasos comunes, que generalmente deben seguir todos los programadores.

○ **Análisis**

El problema se analiza teniendo presente la especificación de los requisitos dados por el cliente de la empresa o por la persona que encarga el programa.

○ **Diseño**

Una vez analizado el problema, se diseña una solución que conducirá a un algoritmo que resuelva el problema.

○ **Codificación (implementación)**

La solución se escribe en la sintaxis del lenguaje de alto nivel (por ejemplo, Pascal) y se obtiene un programa fuente que se compila a continuación.

- **Ejecución, verificación y depuración**

El programa se ejecuta, se comprueba rigurosamente y se eliminan todos los errores (denominados “bugs”, en inglés) que puedan aparecer.

- **Mantenimiento**

El programa se actualiza y modifica, cada vez que sea necesario, de modo que se cumplan todas las necesidades de cambio de sus usuarios.

- **Documentación**

Escritura de las diferentes fases del ciclo de vida del software, esencialmente el análisis, diseño y codificación, unidos a manuales de usuario y de referencia, así como normas para el mantenimiento.

- **NET Framework**

Es una serie de librerías que conforman el paquete de desarrollo principal de Microsoft. Actualmente se divide en .NET Framework, .NET Core y Xamarin. (Microsoft, 2021)

La primera es la implementación original de .NET, con la que fue construida CRM Dynamics 2011 y la que usaremos en este proyecto.

La segunda es la implementación de .NET orientada a funcionar en múltiples plataformas (Windows, Linux y macOS). Su código fuente es libre y se encuentra en Github.

Y, por último, Xamarin es la implementación de .NET orientada para dispositivos móviles iOS y Android.

- **Aplicación Web**

Una aplicación web es una página web que no tiende a ser estática (aunque podría serlo si su contenido es generado en una memoria caché de manera periódica) y que puede ser compuesto por N-capas. En la mayoría de los desarrollos de aplicaciones web se opta por distribuirla en 3 capas. (Microsoft, 2014)

- **Capa de presentación**

- **Capa de negocio**

- **Capa de datos**

- **Servicio de Windows**

Un servicio de Windows es una aplicación especial del sistema operativo Windows. No

son aplicaciones comunes con las cuales un usuario puede interactuar por alguna interfaz gráfica. Estas aplicaciones funcionan en segundo plano. Inician junto con el sistema operativo y son controladas por el Panel de control de Servicios o por otra aplicación que pueda utilizar las funciones de los servicios. (Microsoft, 2018). En los servicios de Windows suelen alojarse las aplicaciones más críticas del sistema operativo para su correcto funcionamiento y para poder enriquecer el funcionamiento del sistema operativo.

Cualquier individual también puede desarrollar nuevas aplicaciones como servicios de Windows. Esto ocurre cuando nos encontramos en la necesidad de ejecutar un proceso desde el inicio del sistema operativo. En este proyecto se mostrará el desarrollo de un servicio de Windows como parte de la solución de integración entre CRM y Banner.

➤ **Node.js**

Es un conjunto de librerías que interpretan código JavaScript para poder desarrollar programas orientados a eventos asíncronos. Node.js es software libre. (Node.js, 2020)

Node.js se orienta a eventos asíncronos para poder trabajar en 1 solo hilo y de este modo no preocuparnos por manejar distintos hilos lo cual puede llegar a ser complejo en proyectos grandes.

Todos los inputs y outputs (lectura de archivos en disco o una petición HTTP) no bloquean la ejecución del proceso lo cual permite escalar fácilmente ya que, si se inician nuevas instancias del proceso, cada uno de estos es independiente.

Dada su gran adopción y

por los grandes aportes de librerías libres desarrolladas por la comunidad de usuarios, es posible utilizar este framework para desarrollar un servicio en este proyecto.

➤ **SOA**

La definición de una Service Oriented Architecture o Arquitectura Orientada a Servicios según (Bianco, 2007) no es universalmente aceptada. Lo que es central a todo, sin embargo, es la noción de un servicio. Para SOA, un servicio es:

- Es independiente.
- Está distribuido a través de la red.
- Tiene una interfaz publicada y se puede conocer cómo comunicarse con el servicio.
- Enfatiza la interoperabilidad, permite su comunicación con múltiples lenguajes y plataformas.

- Es detectable, por medio de un directorio de servicios para poder ser usado por los usuarios.
- Está acoplado dinámicamente, la definición del servicio puede existir y su implementación puede ser cargada en tiempo de ejecución.

En la realidad, los servicios que componente una SOA pueden obviar o relajar algunas de estas características como su posibilidad de ser detectada o su acoplamiento dinámico.

Vamos a definir SOA como un estilo de arquitectura donde los sistemas consisten en servicios de usuario y servicios de proveedores.

En el presente trabajo, los componentes y servicios desarrollados adoptan las características de una arquitectura SOA.

➤ CRM

Customer Relationship Management (Gestión de relación de clientes) o CRM, es un acrónimo utilizado para sistemas que ayudan a una empresa a manejar su relación con los clientes haciendo su seguimiento hasta la venta y post-venta. La definición de una aplicación “CRM” suele diluirse y se confunde con aplicaciones de empresas que están en el rango de automatización de fuerza de ventas, sistemas para manejo de centros de contacto y hasta tiendas de comercio virtual.

Pero ha emergido una nueva era de CRM, una que está enfocada en la gestión de relaciones. Algunas personas en la industria lo llaman “XRM” porque el factor de X puede significar cualquier relación importante para el negocio de la que quiera manejar información. El cambio de paradigma de CRM a XRM quita el enfoque del enfoque que le puede dar un fabricante a su solución hacia las necesidades de las empresas. Esto permite a las empresas definir sus reglas, procesos y gestionar las relaciones en el software. Los sistemas CRM más modernos son modulares y se les pueden adicionar funcionalidades que se adecúen a la necesidad del negocio. (CRM Software Blog, 2011)

Incluso se pueden integrar con otros sistemas para la construcción de un ecosistema empresarial en línea.

➤ CRM Dynamics 2011

Es el sistema CRM con el que trabajamos en el presente informe. Cualquier empresa queda pequeña ante esta aplicación ya que muy personalizable.

El sistema tiene una arquitectura multi-nivel que provee una gran escalabilidad, flexibilidad y extensibilidad que una aplicación de 2 capas (cliente-servidor). (Packt, 2013)

El nivel de presentación provee una interfaz a través de un cliente web, CRM para

Outlook o CRM para móviles. Esta interfaz puede ser personalizable desarrollando integraciones a nivel de cliente web para modificar el comportamiento de esta.

A nivel de aplicación ejecuta el servidor de CRM que incluye servidores web, lógica de negocio, seguridad y componentes de acceso a datos. Puede ser personalizada usando integraciones a nivel de servidor, flujos de trabajo, diálogos, plugins y actividades de flujos de trabajo personalizables.

En el Anexo 2 - Diagrama de componentes de CRM Dynamics 2011 se puede apreciar un diagrama de los componentes que tiene el producto CRM Dynamics 2011.

El nivel de datos almacena la información de los clientes y la metadata del sistema. Se pueden realizar personalizaciones a través de cambios a la metadata, pero no es soportada una modificación del esquema de metadatos directamente a la base de datos.

Cada aplicación que interactúa con CRM lo hace a través de sus servicios web nativos. Alternativamente el sistema provee de vistas filtradas en base a los niveles de acceso de cada usuario para poder hacer consultas SQL a la base de datos directamente para poder extraer información del sistema.

El sistema actualmente se encuentra desplegado en modalidad on-premise siendo alojado en el datacenter de la empresa.

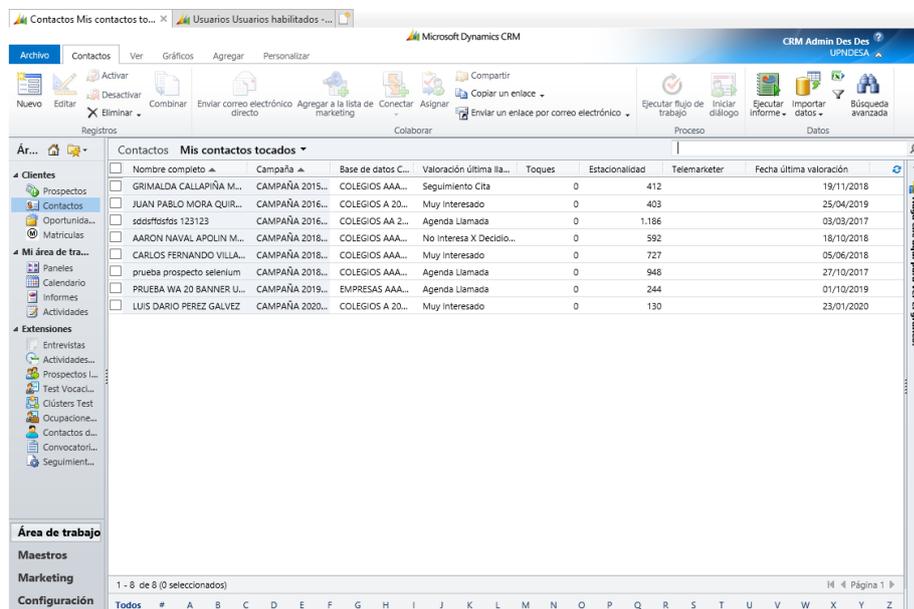


Ilustración 4 - Captura de CRM Dynamics 2011

Fuente: Captura de ambiente de desarrollo CRM Dynamics 2011 UPN

- **Plugins**

Los plugins de CRM tienen un concepto muy similar a los triggers o gatilladores de

una base de datos moderna.

Los plugins son funciones personalizadas que pueden ser invocadas cuando se efectúa una operación específica sobre un registro en CRM.

Algunos de las operaciones más comunes pueden ser:

- **Lectura de registros**
 - **Creación de nuevos registros**
 - **Modificación de registros**
 - **Eliminación de registros**
 - **Asignación de registros**
- **SDK de CRM Dynamics 2011**

El SDK o Software Development Kit es un conjunto de librerías y herramientas para desarrollar componentes sobre un sistema destino y específico. En este caso usaremos el SDK para CRM Dynamics 2011 que provee de todas las interfaces y librerías para conectarnos al sistema y poder desarrollar nuevos componentes como plugins, o actividades de workflow personalizadas.

Al igual que CRM Dynamics 2011, son librerías desarrolladas con .NET Framework.

➤ **Message Broker**

Un Message Broker o Sistema de colas, permite establecer un ESB. Este sirve como un middleware para la comunicación entre diferentes sistemas, plataformas y protocolos en los que distintos servicios pueden estar funcionando.

Un protocolo con el que nos comunicamos con un software de Message Broker es AMQP.

Existen distintos softwares de Message Broker, pero en el presente trabajo nos centraremos en el uso de RabbitMQ.

➤ **Protocolo AMQP**

Advanced Message Queuing Protocol es un protocolo de estándar abierto a nivel de capa de aplicación. (Vinoski, 2006) El protocolo se centra en definir la orientación de mensajes en un sistema de colas. Su encolamiento, enrutamiento, exactitud y seguridad. La especificación es tan amplia que los softwares de Message Broker actuales no llegan a implementar la totalidad del protocolo, solo lo necesario para un funcionamiento

básico. Este no es el único protocolo de estándar abierto para manejar un sistema de colas.

➤ **RabbitMQ**

Es un software de Message Broker Open Source que implementa el protocolo AMQP. Este brinda librerías en distintos lenguajes (C#, Javascript, Java, Python, Ruby, etc.) para poder comunicarnos con el servicio.

Este servicio se encuentra entre los más utilizados para sistemas de funciones críticas siendo empleado desde empresas de bolsas de valores hasta salas de chat.

➤ **SIS**

Un Student Information System o Sistema de Información Estudiantil, es un software diseñado para apoyar los procesos de escuelas y universidades en el manejo de toda la información de sus estudiantes. (Paget, 2018)

Desde notas, asistencia, comportamientos y más.

Un SIS sirve como la base de datos central para todas las operaciones de una institución educativa. Estos han ido incorporando las siguientes funcionalidades y módulos para tareas administrativas como:

○ **Admisiones y matrículas**

Suelen manejar el proceso entero desde el contacto inicial hasta finalizar la matrícula.

○ **Facturación y pagos**

Pagar y ser pagado puede ser administrador por un SIS ya que tiene la información de periodos, padres y empleados.

○ **Reportes**

Los reportes ayudan a las instituciones educativas a identificar tendencias en el comportamiento de estudiantes y analizar los resultados para desarrollar estrategias para su mejorar el rendimiento académico de sus estudiantes.

➤ **Banner**

Es el software SIS utilizado en la actualidad en la empresa. Este sistema es desarrollado por la empresa Ellucian, líder internacional en software para educación superior. Banner

es sistema de renombre internacional implementado en más de 1400 instituciones.
(Ellucian, 2021)

➤ **Scrum**

Scrum es un proceso de desarrollo de software donde se definen equipos de trabajo fuertemente ligados y colaborativos que fabrican entregables de productos de software en lapsos de tiempo llamados Sprints.

La meta de un Sprint es producir un producto visible, usable y entregable que implementa una o más interacciones de usuario con el sistema.

En la actualidad (2021), en la Universidad Privada del Norte utilizamos esta metodología ágil para los desarrollos de nuestros productos.

CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA

1. Descripción de la problemática

La empresa educativa Universidad Privada del Norte, tuvo la necesidad de integrar dos sistemas importantes, el sistema CRM y un nuevo sistema académico llamado Banner.

Antes de la implementación de la solución expuesta en el presente trabajo, el sistema CRM se comunicaba con el antiguo sistema académico FOX desarrollado en Visual FoxPro.

Un servicio web que se conectaba al CRM extraía la información que se quería llevar a FOX generando un archivo plano o archivo separado por comas que descargaban a la PC. Este contenía información como datos personales y datos académicos (carrera, modalidad de ingreso) y algunos identificadores internos.

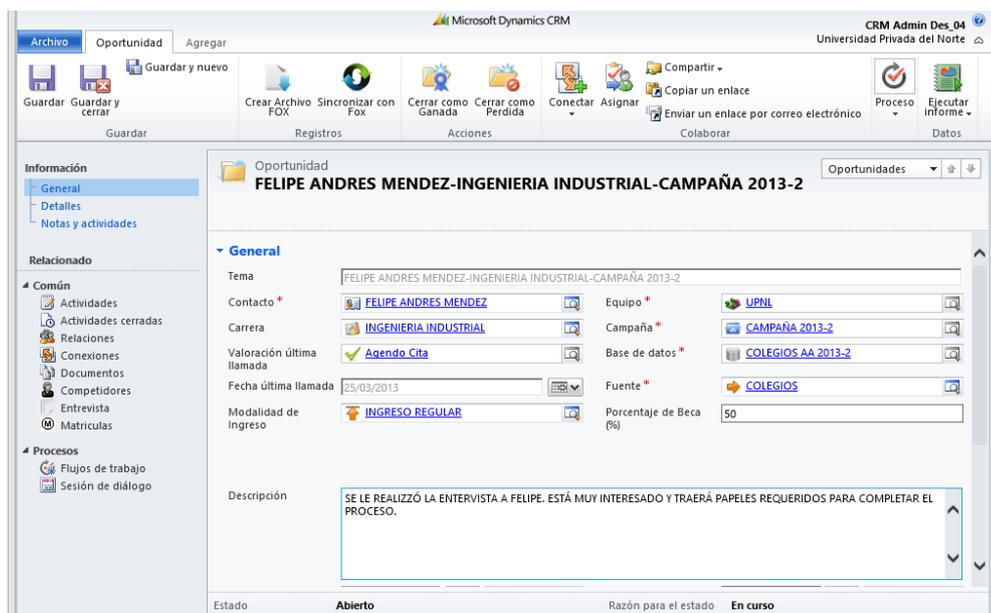


Ilustración 5 - Ventana previa al generar un archivo CSV

Fuente: Captura de sistema CRM de UPN

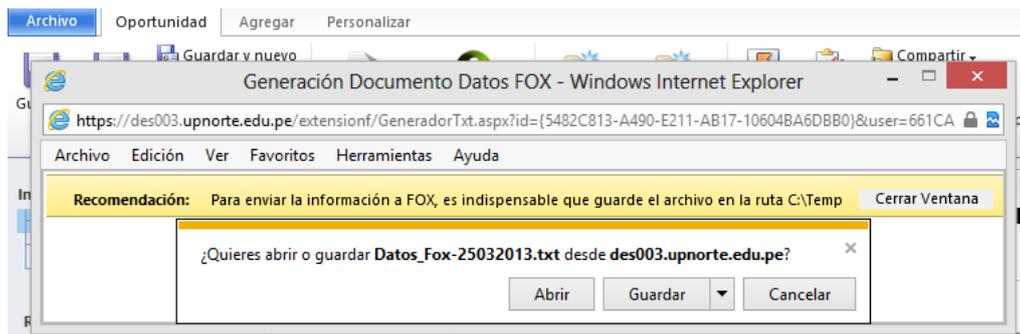


Ilustración 6 - Servicio Web entregando de archivo CSV de exportación con FOX

Fuente: Captura de sistema CRM de UPN

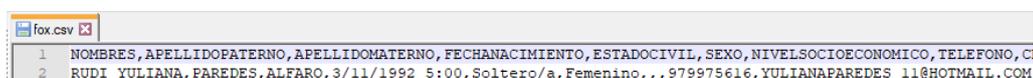


Ilustración 7 - Contenido del archivo CSV

Fuente: Archivo CSV de integración obsoleta

La carga manual de este archivo hacia FOX podría causar los siguientes problemas:

- Los valores que se exportaban llevaban una coma o un símbolo extraño que causaba un problema al momento de leer el archivo en el sistema FOX.
- Cuando el estudiante ya existía en los archivos de bases de datos de FOX, se seguían una serie de pasos especiales dentro del sistema FOX. De no realizarlos, podrían estar creando un estudiante duplicado en el sistema.
- El sistema FOX no respondía y había que abrir la consola del sistema nuevamente.
- En el peor escenario la carga del archivo corrupto había dejado un estudiante insertado con datos incompletos o corruptos.

Estos casos llevaban a una gran carga para el área de sistemas ya que los especialistas en el sistema FOX debían brindar soporte a los trabajadores del área de Admisión, ya sea para corregir caso por caso manualmente o reforzando los pasos que debían realizar en FOX de manera cautelosa.

Con la llegada de Banner finalmente adoptaríamos un sistema que permitiría integraciones que ya no requerirían una carga manual de archivos expuesta a errores humanos.

2. Justificación

En los últimos años, las empresas deben acoplarse al creciente ritmo agitado del estilo de vida de las personas. No basta con tener aplicaciones web donde una empresa pueda cargar información y que sus clientes la consuman. Los clientes esperan interactuar y ver reflejadas sus acciones en el ecosistema de aplicaciones de una empresa.

La Universidad Privada del Norte optó por modernizar su sistema académico y con ello, abrir un abanico de nuevas tecnologías para integrarse a nuevos sistemas.

El presente trabajo es clave al ser la primera integración entre el nuevo sistema Académico y CRM.

A continuación, presentamos las siguientes justificaciones:

2.1. Justificación social:

Cada minuto perdido durante el proceso de matrícula causa malestar a los estudiantes. Cada error durante el proceso de matrícula son minutos valiosos perdidos que se multiplican por cada trabajador atendiendo en cada sede a cada estudiante que se está matriculando. Con la solución puesta en marcha se busca disminuir el tiempo de espera del estudiante durante su matrícula, disminuyendo cualquier malestar causado por tiempos muertos.

2.2. Justificación económica:

La presente solución busca disminuir los tiempos perdidos con los incidentes que ocurrían con el sistema FOX en el proceso de matrícula. Esto se refleja directamente sobre un aumento de productividad de cada trabajador y sobre la cantidad de personas que pueden atender en su jornada laboral. Como resultado, el flujo de ingreso económico de la Universidad aumenta.

2.3. Justificación tecnológica:

La solución presentada intenta cubrir una comunicación en línea entre Banner y CRM la cual no era posible con FOX y CRM.

El sistema académico FOX causaba una gran deuda técnica a la universidad, ya que FOX utilizaba archivos como bases de datos y el lenguaje de programación era antiguo y no tenía soporte (Visual FoxPro). Todo esto no dificultaba tener un proceso de matrícula en línea por medio de un sitio web, ya que la tecnología nativamente no era compatible con .NET Framework.

Ahora con un sistema moderno como Banner, nos permite tener una integración uniforme a futuro incluso con otros sistemas.

2.4. Justificación práctica:

Permitió explorar nuevas tecnologías como un RabbitMQ, NodeJS, uso SDK de CRM, expandió mi umbral de conocimientos y también el de mis compañeros de trabajo. Hoy en día, todos hemos tenido la oportunidad de desarrollar algún componente que se comunica con RabbitMQ o con CRM. Hemos publicado o consumido mensajes para comunicar distintos sistemas desarrollados en distintas plataformas y lenguajes.

3. Limitaciones

El presente trabajo se limita a cubrir la integración entre 2 sistemas principales de la Universidad: CRM Dynamics 2011 y Banner. Abarcando 3 actividades principales: registrar al postulante en Banner, levantar retenciones por documentos en Banner y envío de resultados de exámenes de admisión a Banner.

4. Restricciones

4.1. Tiempo de entrega:

Era muy importante llegar a la fecha de entrega ya que las capacitaciones al equipo de admisión estaban programadas por parte del equipo de Banner. Estas involucraban el uso de CRM junto con Banner.

4.2. Pruebas funcionales:

Cuando necesitábamos probar todo el ciclo del flujo de la información requería que se encuentren 3 equipos en una reunión: equipo de desarrollo de UPN, equipo local de Banner y equipo de implementación de Banner que se encontraba en USA Baltimore. Esto para poder atacar cualquier incidente rápidamente. Aunque este proyecto era la prioridad número 1 en la universidad, dependíamos de la disponibilidad de cada miembro del equipo incluso en zonas geográficas y horarias distintas.

5. Requisitos funcionales y no funcionales

5.1. RabbitMQ

Código	Descripción
RF1	<p>Configurar dos exchange en el RabbitMQ local: upn2sis.outbound.fanout y sis2upn.inbound.topic</p> <p>Outbound: todos los mensajes salientes pasan a ambas colas. upn2sis.outbound y upn2sis.outbound.wiretap</p> <p>Inbound:</p> <p>Routing key: crm.studentid queue: crm.success.queue</p> <p>Routing key: crm.stuSuspense, crm.error, crm.appl.badFormat, crm.ApplNotPushed, crm.stuSuspense queue: crm.error.queue</p>

RF2	Establecer una conexión entre dos instancias de RabbitMQ. Una local y otra ya existente en el grupo de servidores de Banner. Se usará la característica Shovel de RabbitMQ para consumir mensajes de una cola de un servidor externo y también para que la instancia remota pueda consumir una cola de nuestro servidor local.
-----	--

5.2. Servicio Web

Este componente es una aplicación web que contiene 3 botones para generar 3 tipos de mensajes que Banner recibirá y procesará. Todos los mensajes estarán en formato XML. Todos los mensajes se enviarán al exchange al Exchange del RabbitMQ local upn2sis.outbound.fanout. Cada mensaje llevará un routing key distinto para que Banner pueda distinguir el tipo de mensaje que procesará.

Código	Descripción
RF3	Se requiere que la aplicación pueda abrirse desde el formulario de la Oportunidad de un estudiante. Se enviarán por la URL los parámetros del ID de la oportunidad y el ID del usuario que está realizando la operación.
RF4	El primer botón generará un mensaje que mensaje contiene los datos para iniciar una solicitud de admisión en Banner. Los datos son extraídos de la ficha de Oportunidad y Contacto del estudiante en CRM.
RF5	El segundo botón generará un mensaje conteniendo los datos como Código de estudiante, periodo y 2 campos sobre el estado de la documentación del estudiante. Estos 2 campos sobre la documentación son: Documentación mínima entregada, Documentación completa entregada. En Banner estos mensajes servirán para levantar retenciones sobre la documentación entregada por los estudiantes. El estado de la documentación del estudiante podría limitar la matrícula del estudiante.
RF6	El tercer botón generará un mensaje para enviar los estados de los resultados del examen de admisión. Las notas son recolectadas de dos campos en CRM. Las notas son de un examen de ciencias y un examen de letras. En el mensaje debe incluir un atributo según los resultados de los exámenes. Atributo DDOS = Aprobó ambos exámenes Si desaprueba uno de los exámenes se envía un atributo vacío. Esto le retirará una restricción para que pueda matricularse en el módulo de nivelación PDN.
RF7	Se tendrá un cuarto botón que solo se muestra con el usuario administrador del sistema. Solo usado para propósitos de pruebas.

	Envía el mensaje XML que construye el primer botón, a un correo configurado.
--	--

5.3. Servicio de consumidor

Código	Descripción
RF8	Consumir mensajes de la cola de Rabbit crm.success.queue. Los mensajes entrantes contienen el Id de oportunidad y código del estudiante. Con el ID de la oportunidad encontrar el contacto del estudiante y guardar el código del estudiante en el Contacto en el campo Código de estudiante.
RF9	Consumir mensaje de la cola de Rabbit crm.error.queue. Todos los mensajes de error incluyen el ID de la oportunidad en cuestión. Todos los mensajes que llegan a esta cola serán enviados por correo al consejero de la oportunidad con copia a su Jefe configurado en CRM.

6. Análisis y Diseño

6.1. Servicio Web

Según los requerimientos, se necesita hacer un nuevo botón dentro de la interfaz de CRM para invocar al componente del Servicio Web.

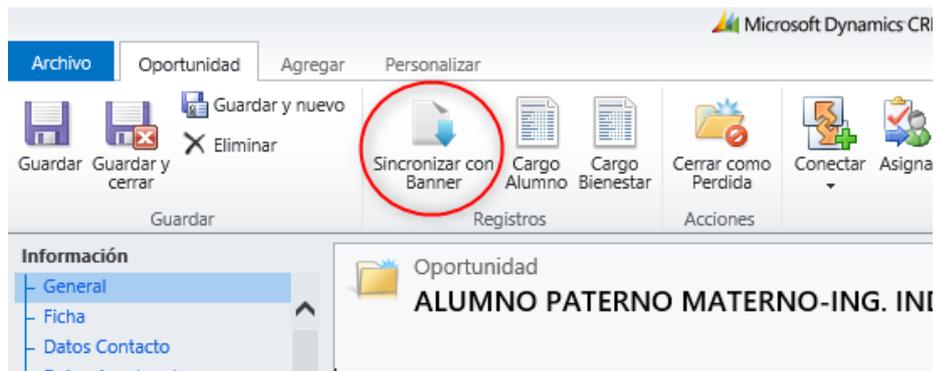


Ilustración 8 - Interfaz de CRM con botón de integración

Fuente: Captura de sistema CRM de UPN

La funcionalidad de los 3 botones era lo primordial. En versiones futuras del desarrollo mejoré el diseño con una interfaz más amigable.

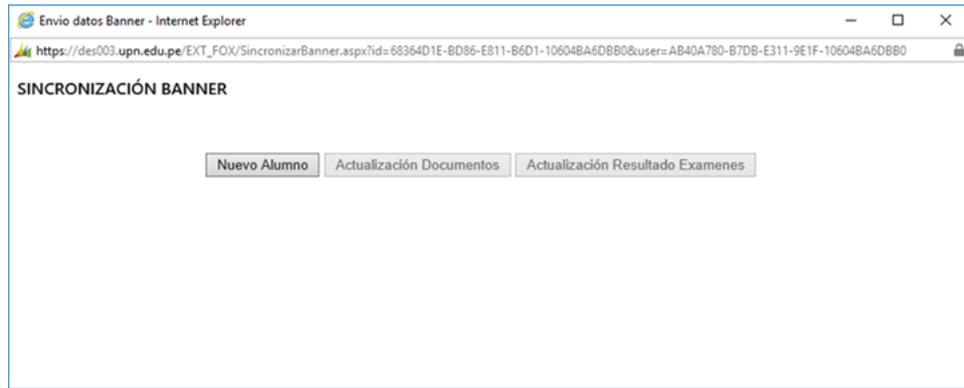


Ilustración 9 - Captura del componente de Servicio Web

Fuente: Captura de sistema CRM de UPN

6.2. Servicio consumidor

Este componente está desarrollado en lenguaje Javascript con el framework Node.js y tiene la función de consumir los mensajes de respuesta de Banner que llegan por RabbitMQ.

Estos mensajes pueden ser el resultado de la solicitud de admisión. Puede contener el ID de estudiante que debe ser guardado en CRM usando el SDK de CRM compilado para Javascript, o notifica por correo al usuario funcional que inició la gestión si es que ocurrió un problema en el proceso.

7. Diagramas de flujo orientado al proceso, diseño de interfaces

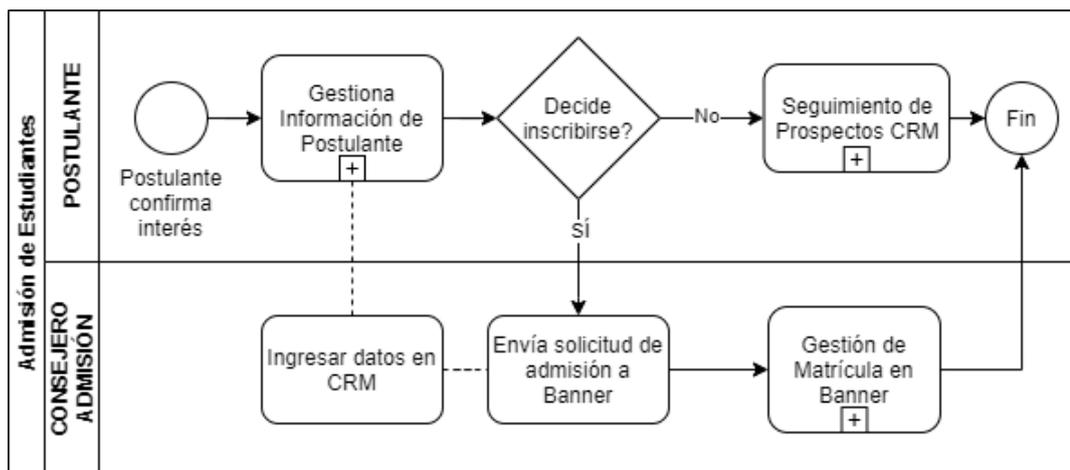


Ilustración 10 - Proceso de envío de alumno a Banner

Fuente: Elaboración propia

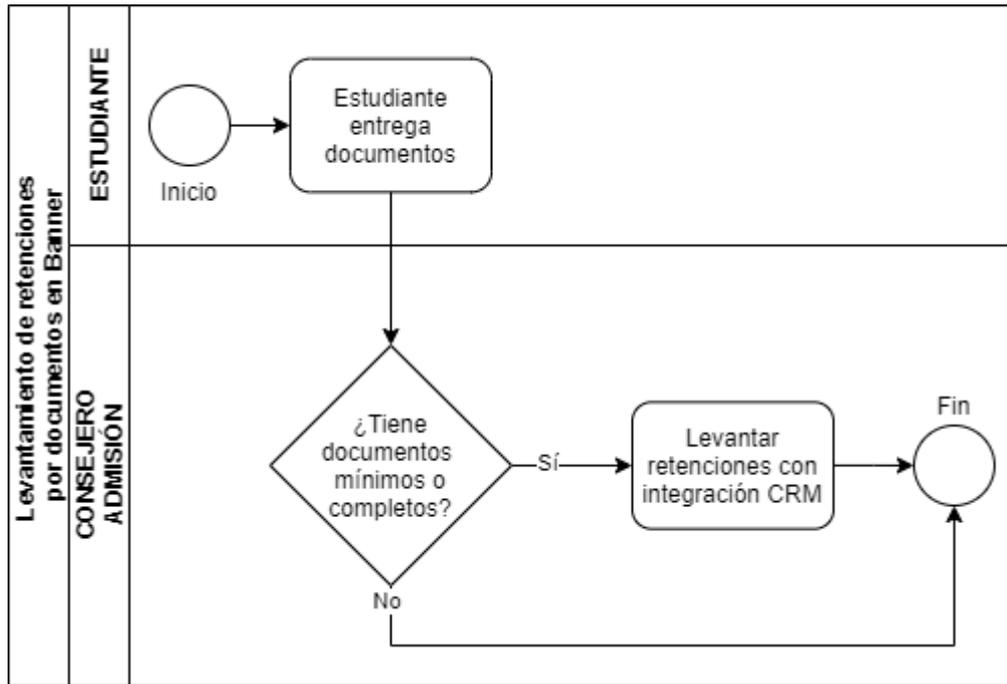


Ilustración 11 - Proceso de levantamiento de retenciones por documentos en Banner

Fuente: Elaboración propia

8. Modelo de datos

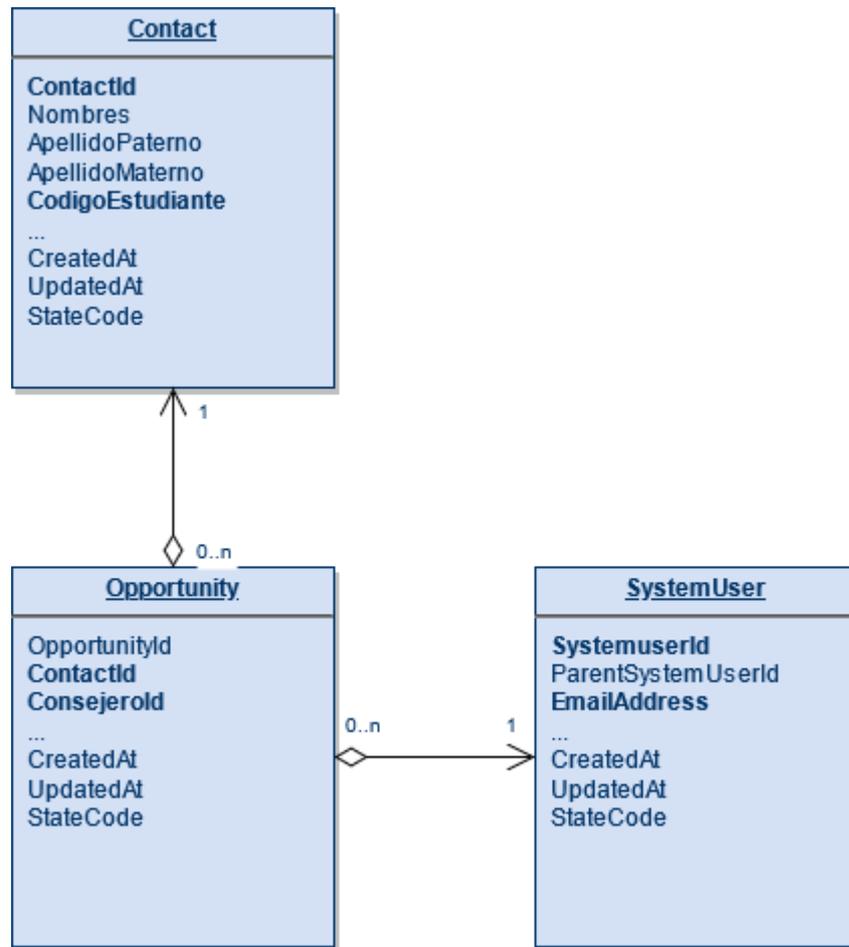


Ilustración 12 - Diagrama de datos de entidades usadas de CRM

Fuente: Elaboración propia

9. Implementación

En el área de desarrollo de software de la Universidad usamos la metodología ágil Scrum.

Se necesitaba entregar un producto mínimo viable, pudiendo enviar solicitudes de admisión al sistema académico, y luego proceder con las capacitaciones de usuario. Durante el desarrollo de este proyecto estábamos adaptándonos a esta metodología y elaboré el primer Sprint en 1 semana cumpliendo los RF1, RF2, RF3, RF4, RF8 y RF9.

En un segundo Spring cubrí las funcionalidades de RF5, RF6 y RF7.

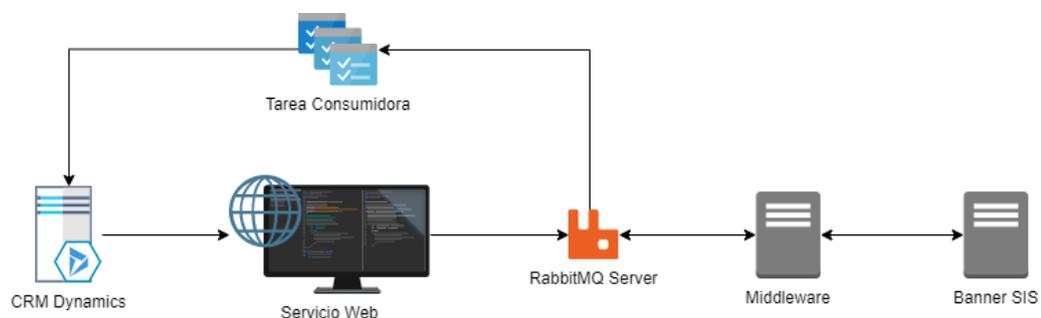


Ilustración 13 - Componentes de Integración SIS-CRM

Fuente: Elaboración propia

9.1. Servicio Web

El servicio web consiste en el formulario que muestra los 3 botones de integración, fue desarrollado en .NET Framework junto con el SDK para .NET desarrollado por Microsoft.

Utilicé el SDK de CRM para conectarnos a CRM Dynamics y extraer los valores que se requieren para construir los mensajes XML.

Microsoft siempre recomienda utilizar el SDK de CRM para consultar y manipular información del sistema CRM Dynamics. En casos muy extremos deberíamos considerar una consulta directa a la base de datos. Y en ninguna circunstancia hacer alguna consulta DML a la base de datos de CRM.

Se utilizó la tecnología WebForms de .NET Framework para elaborar la interfaz.

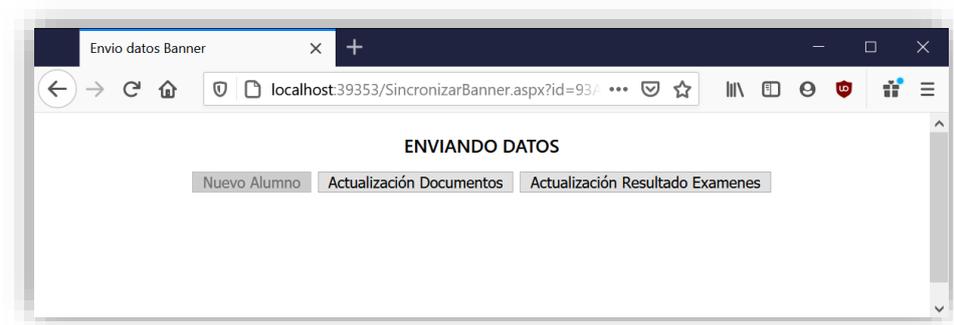


Ilustración 14 - Interfaz desarrollada con .NET WebForms

Fuente: Captura de Servicio Web integrador

9.2. Servicio consumidor

Al mensaje contener el resultado satisfactorio, procesamos el mensaje haciendo unas validaciones previas del ID del registro donde debe ser guardado el ID de estudiante:

```
export default function successQueueCallback(q) {
  q.subscribe({ ack: true }, function (message, headers, deliveryInfo, messageObject) {
    if (message && deliveryInfo.routingKey == 'crm.studentid') {
      var textChunk = decoder.write(message.data);

      xmlParser.parseString(textChunk, function (err, result) {

        var xmlResponse = result.DocumentElement.Result;

        var spridenId = xmlResponse.SPRIDEN_ID;
        var crmGuidPos = xmlResponse.CRM_GUID.lastIndexOf('*');

        if (crmGuidPos == -1) {
          badFormat(message, headers, xmlResponse)
          messageObject.acknowledge(false);

          return
        }

        crmGuidPos = crmGuidPos + 1;

        var crmGuid = xmlResponse.CRM_GUID.substr(crmGuidPos, 36);
```

Ilustración 15 - Captura de código de tarea consumidora

Fuente: Código fuente de servicio consumidor

Luego guardamos el valor SPRIDEN_ID (el cual es el ID de estudiante que tiene el formato N123456), en el Contacto del registro en CRM

```

var consejero = crmConn.retrieve('systemuser', { systemuserid: opportunity.cmtx_consejeroid });
if (consejero) {
  var body = 'Consejero: ' + consejero.fullname
  + '<br>Codigo Alumno: ' + spridenId + '<br>'
  + 'Oportunidad: ' + opportunity.name + '<br>'
  + 'Id: ' + crmGuid;

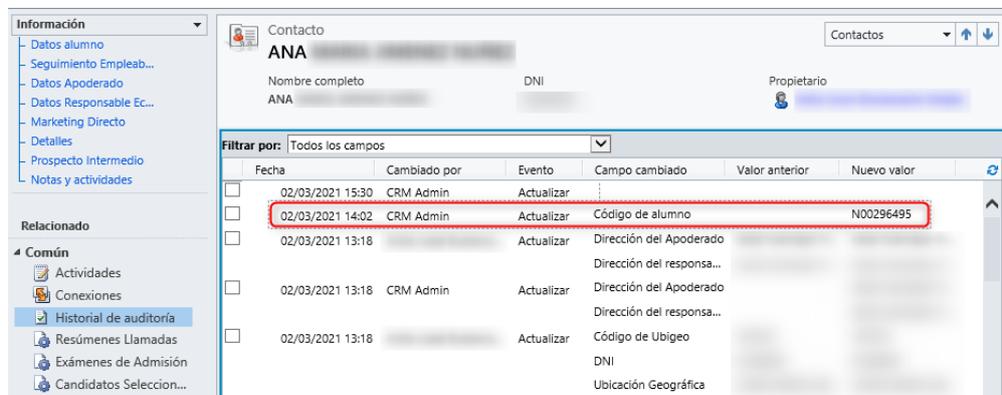
  var administrador = crmConn.retrieve('systemuser', { systemuserid: consejero.parentsystemuserid });
  if (administrador) {
    sendMail(
      [ consejero.internalemailaddress, administrador.internalemailaddress ]
      , 'Notificaciones Banner'
      , 'Estudiante Nuevo Banner: OK'
      , body)
  } else {
    sendMail(
      [ consejero.internalemailaddress ]
      , 'Notificaciones Banner'
      , 'Estudiante Nuevo Banner: OK'
      , body)
  }
  sendMail(alertmailboxes, 'Notificaciones Banner', 'Estudiante Nuevo Banner: OK', body)
}

var contactUpdate = { contactid: contact.contactid, cmtx_codigodealumno: spridenId };

```

Ilustración 16 - Captura de código tarea consumidora

Fuente: Código fuente de servicio consumidor



The screenshot shows a CRM interface for a contact named 'ANA'. On the left, there is a navigation menu with sections like 'Información' and 'Relacionado'. The main area displays a table of change history. A red box highlights a specific entry in the table.

Fecha	Cambiado por	Evento	Campo cambiado	Valor anterior	Nuevo valor
02/03/2021 15:30	CRM Admin	Actualizar			
02/03/2021 14:02	CRM Admin	Actualizar	Código de alumno		N00296495
02/03/2021 13:18		Actualizar	Dirección del Apoderado		
02/03/2021 13:18	CRM Admin	Actualizar	Dirección del Apoderado		
02/03/2021 13:18		Actualizar	Dirección del responsa...		
02/03/2021 13:18		Actualizar	Código de Ubigeo		
			DNI		
			Ubicación Geográfica		

Ilustración 17 - Captura de historial de cambios guardando el ID de estudiante en CRM

Fuente: Captura de sistema CRM de UPN

Luego notificamos al usuario *consejero* que ha iniciado esta solicitud por medio de un correo electrónico, en caso se haya ejecutado el proceso de manera satisfactoria o haya ocurrido algún error.

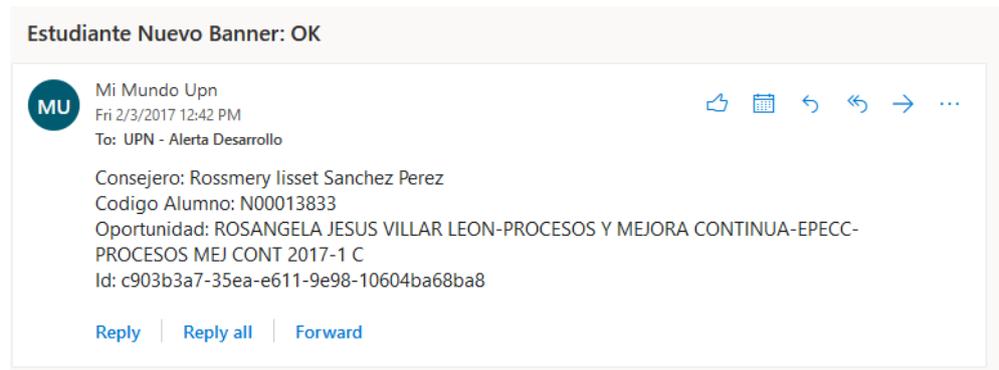


Ilustración 18 - Captura de correo de notificación satisfactoria

Fuente: Correo de notificación de integración

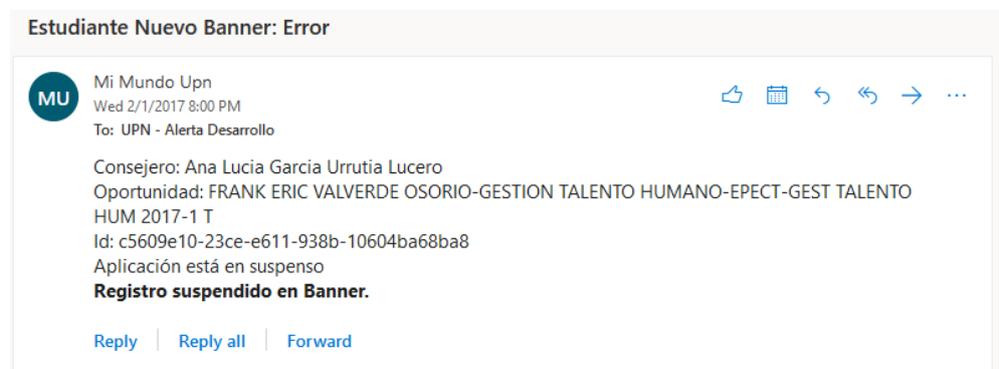


Ilustración 19 - Captura de correo de notificación de error en el proceso

Fuente: Correo de notificación de integración

10. Diagrama de flujo algoritmo orientado a la implementación

10.1. Flujo de formulario de CRM

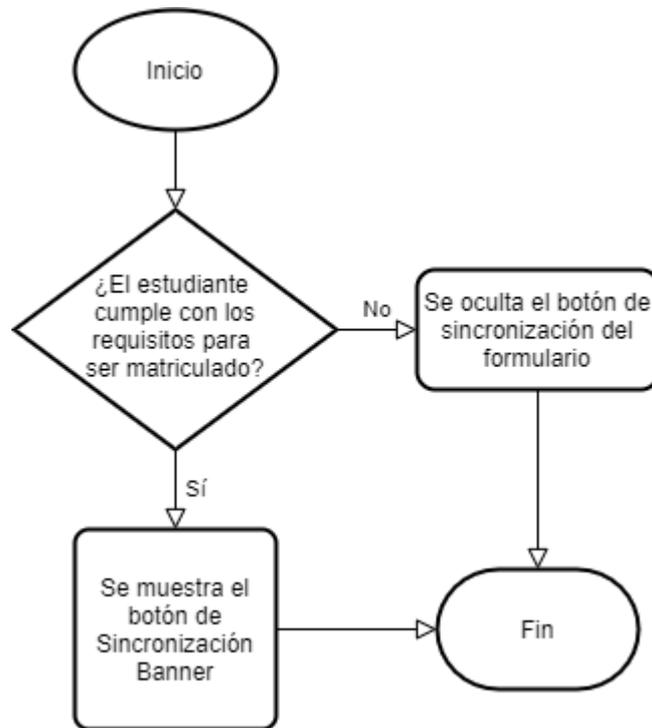


Ilustración 20 - Flujo de comportamiento nuevo botón en formulario CRM

Fuente: Elaboración propia

10.2. Flujo de Servicio Web

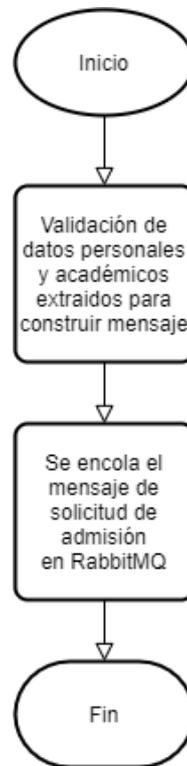


Ilustración 21 - Flujo botón 1: Nuevo Alumno

Fuente: Elaboración propia



Ilustración 22 - Flujo botón 2: Actualización Documentos

Fuente: Elaboración propia



Ilustración 23 - Flujo botón 3: Actualización Resultado Exámenes

Fuente: Elaboración propia

10.3. Servicio consumidor

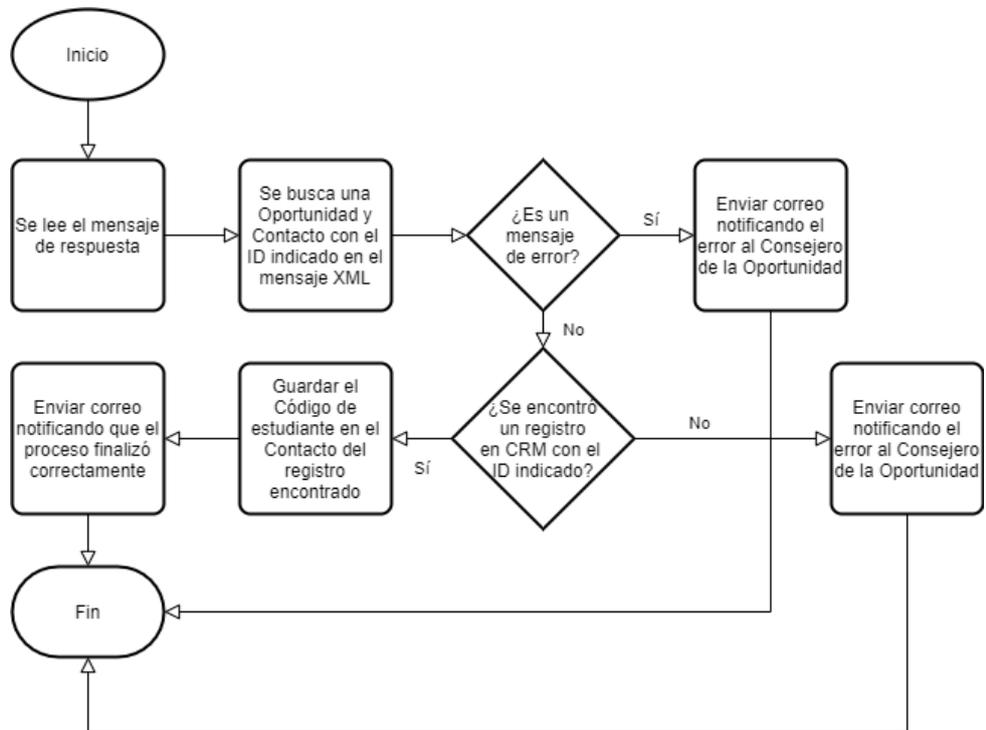


Ilustración 24 - Flujo servicio consumidor

Fuente: Elaboración propia

11. Diagrama de componentes

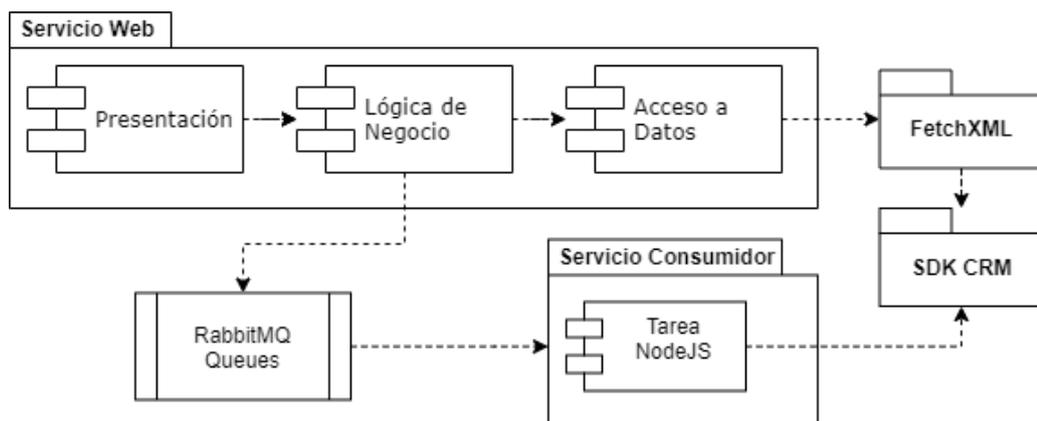


Ilustración 25 - Diagrama de Componentes de la solución

Fuente: Elaboración propia

12. Diagrama de despliegue

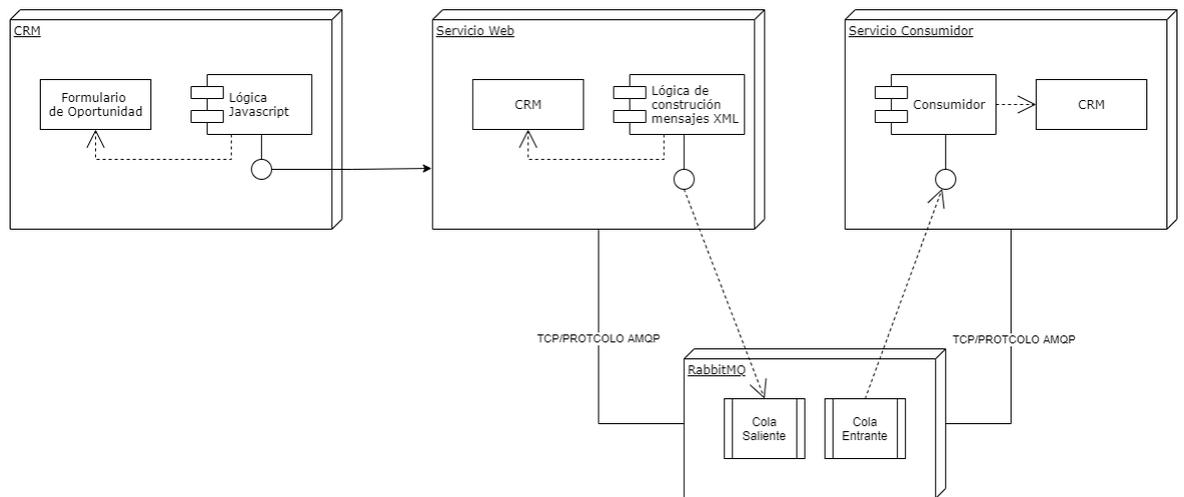


Ilustración 26 - Diagrama de despliegue de solución

Fuente: Elaboración propia

13. Pruebas

13.1. Pruebas de performance

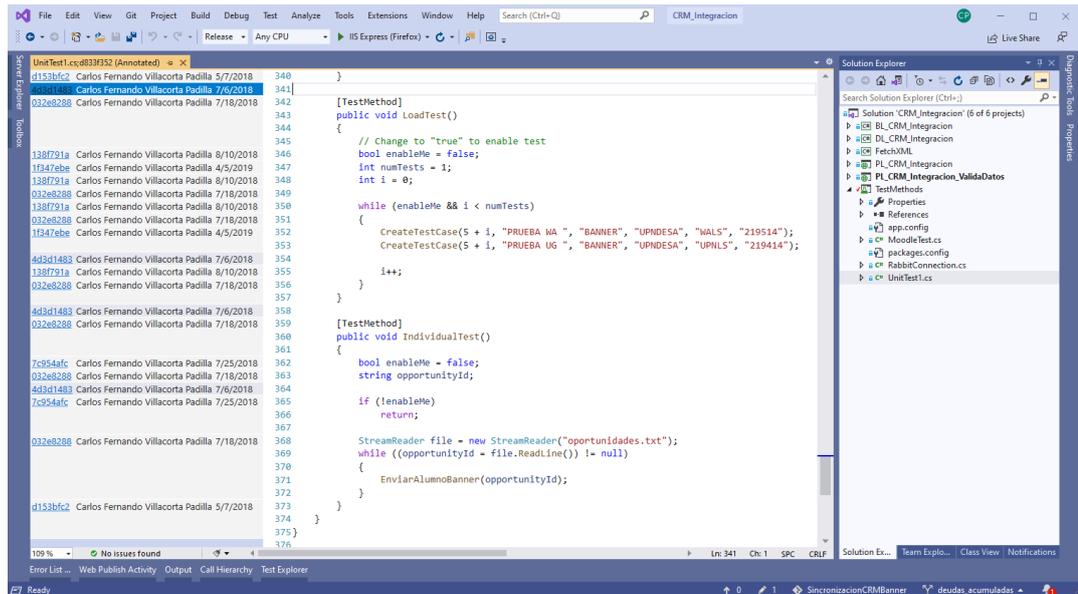
El proyecto cuenta con un generador de casos de prueba masivo de manera programática.

Creando los registros necesarios en CRM invocando los API de CRM para luego generar los mensajes de solicitud de admisión pertinentes, y finalmente procesando los mensajes de respuesta.

Esta prueba busca que el sistema académico no tarde más de 5 minutos en enviar un mensaje de respuesta después de haber hecho todas las validaciones necesarias, en caso contrario se realiza un procedimiento de matrícula distinto llamado "Plan B".

Durante las pruebas realizadas el sistema no tardó en responder más de 1 minuto sin pasar el umbral de 5 minutos.

Actualmente en producción no se cruza el umbral de 5 minutos, ya que normalmente el sistema responde en cuestión de 1 o 2 minutos durante horarios donde hay mayor carga por el uso de los sistemas.



```
340 }
341
342 [TestMethod]
343 public void LoadTest()
344 {
345     // Change to "true" to enable test
346     bool enableMe = false;
347     int numTests = 1;
348     int i = 0;
349
350     while (enableMe && i < numTests)
351     {
352         CreateTestCase(5 + i, "PRUEBA WA ", "BANNER", "UPNDESA", "NALS", "219514");
353         CreateTestCase(5 + i, "PRUEBA UG ", "BANNER", "UPNDESA", "UPNLS", "219414");
354
355         i++;
356     }
357
358 [TestMethod]
359 public void IndividualTest()
360 {
361     bool enableMe = false;
362     string opportunityId;
363
364     if (!enableMe)
365         return;
366
367     StreamReader file = new StreamReader("oportunidades.txt");
368     while ((opportunityId = file.ReadLine()) != null)
369     {
370         EnviarAlumnoBanner(opportunityId);
371     }
372 }
373 }
374 }
375 }
376 }
```

Ilustración 27 - Captura de pruebas de performance

Fuente: Código fuente de Servicio Web integrador

13.2. Pruebas funcionales y de integración

Documentamos las pruebas funcionales e integración realizadas desde el sistema y aquí podemos visualizar las siguientes capturas tomadas de aquel documento realizado por mi persona.

En el Anexo 3 - Caso de pruebas funcionales podrán encontrar una tabla de las pruebas funcionales ejecutadas.

14. Objetivo general

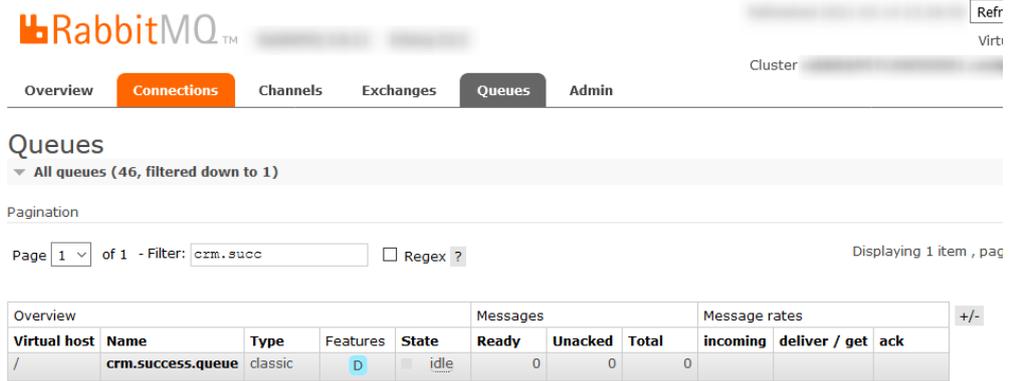
Implementar un aplicativo integrador en el proceso de matrículas para la Universidad Privada del Norte.

15. Objetivos específicos

- Disminuir la cantidad de incidentes por el uso de la integración.
- Disminuir los tiempos perdidos en el proceso de matrícula por errores de integración.

16. Implantación

Primero se realizó la implementación de un servidor de RabbitMQ seguido de su configuración según los RF1 y RF2. Estableciendo la configuración de colas y la conexión hacia el servidor RabbitMQ de Banner.



RabbitMQ™

Overview **Connections** Channels Exchanges **Queues** Admin

Queues

▼ All queues (46, filtered down to 1)

Pagination

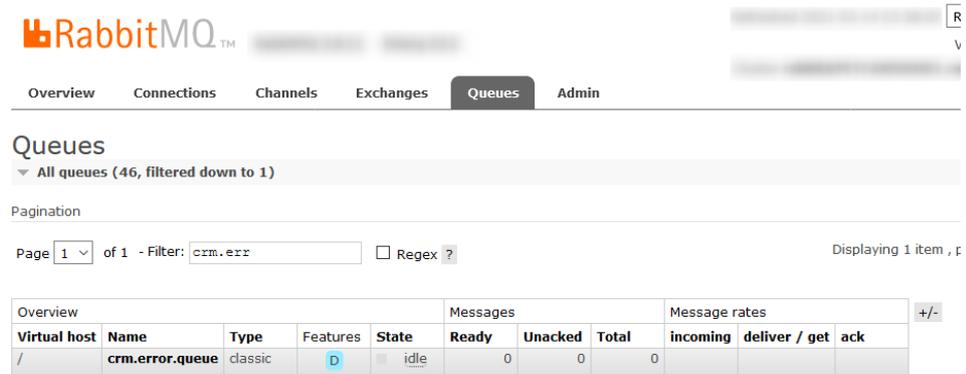
Page 1 of 1 - Filter: Regex ?

Displaying 1 item , pag

Overview					Messages			Message rates				+/-
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack		
/	crm.success.queue	classic	D	idle	0	0	0					

Ilustración 28 - Captura configuración colas RabbitMQ

Fuente: Sistema RabbitMQ



RabbitMQ™

Overview Connections Channels Exchanges **Queues** Admin

Queues

▼ All queues (46, filtered down to 1)

Pagination

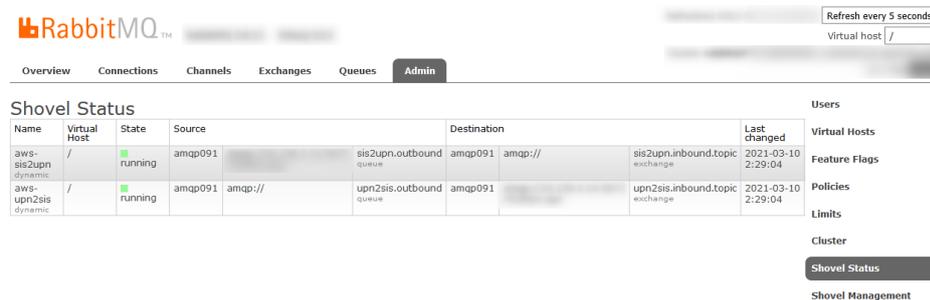
Page 1 of 1 - Filter: Regex ?

Displaying 1 item , pag

Overview					Messages			Message rates				+/-
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack		
/	crm.error.queue	classic	D	idle	0	0	0					

Ilustración 29 - Captura configuración colas RabbitMQ

Fuente: Sistema RabbitMQ



RabbitMQ™

Overview Connections Channels Exchanges Queues **Admin**

Refresh every 5 seconds

Virtual host /

Shovel Status

Name	Virtual Host	State	Source	Destination	Last changed	Users
aws-sis2upn	/	running	amqp091	sis2upn.outbound.queue	2021-03-10 2:29:04	Virtual Hosts
aws-upn2sis	/	running	amqp091	upn2sis.outbound.queue	2021-03-10 2:29:04	Feature Flags

Policies

Limits

Cluster

Shovel Status

Shovel Management

Ilustración 30 - Configuración Shovel en RabbitMQ

Fuente: Sistema RabbitMQ

También se realizó el despliegue de los servicios web y consumidor. El servicio web se desplegó en el servidor WEB004 de la arquitectura de CRM.

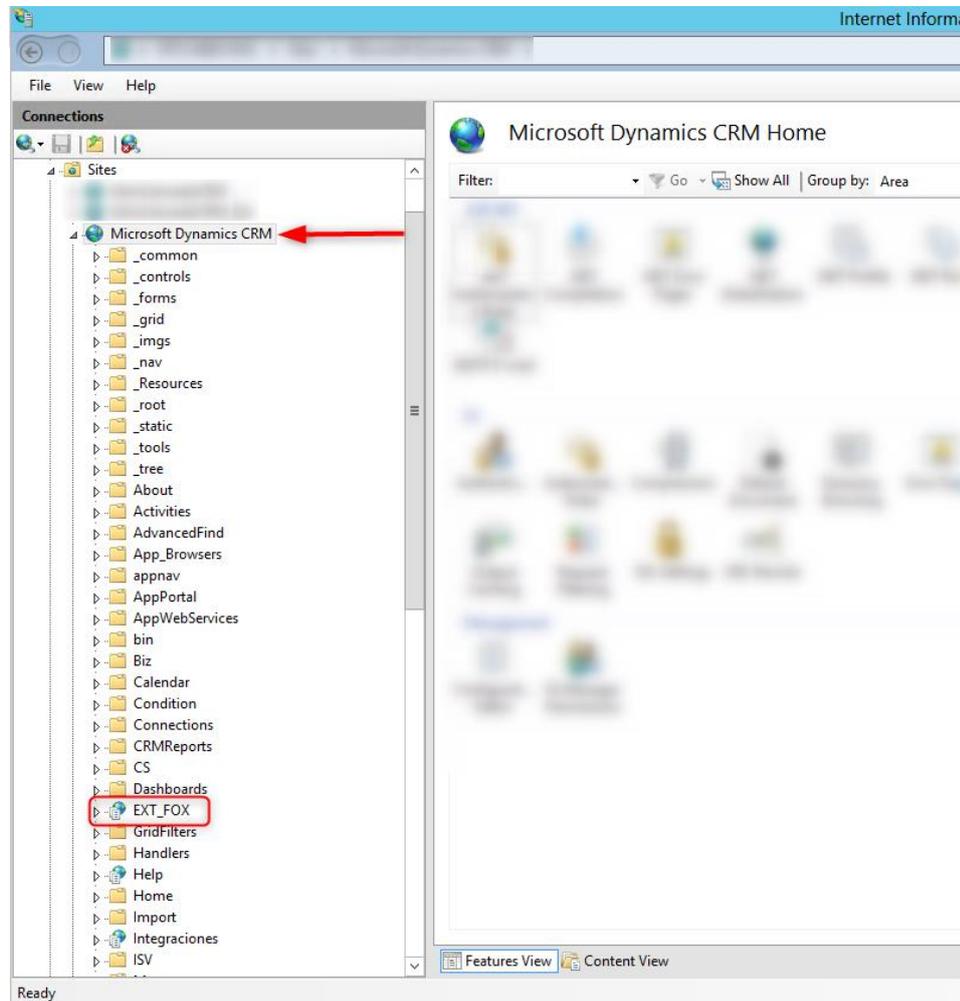


Ilustración 31 - Despliegue de aplicación web en IIS de CRM

Fuente: Captura de IIS de CRM

La tarea consumidora de los mensajes de respuesta de Banner fue desplegada en un servidor de Jobs como un servicio de Windows según la siguiente captura.

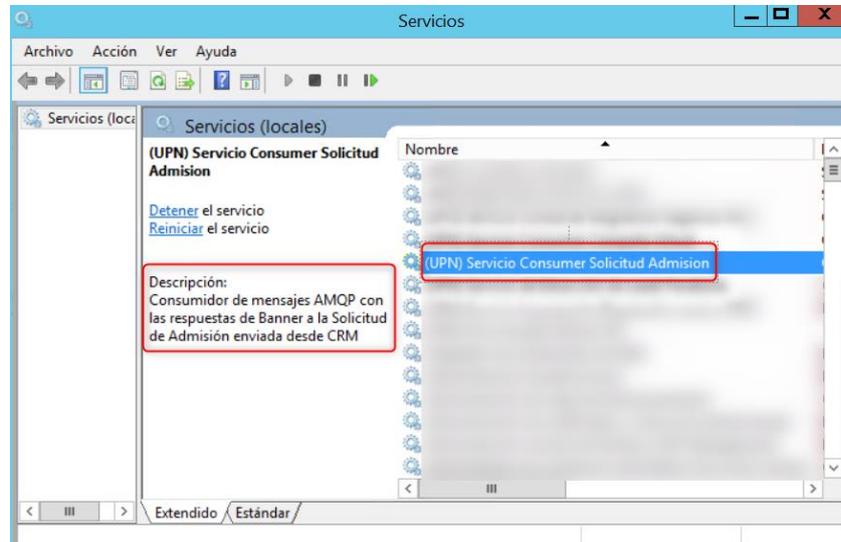


Ilustración 32 - Servicio consumidor como Servicio de Windows

Fuente: Captura de Administrador de Servicios de Windows de un servidor Windows de UPN

Instruí al equipo de implementación de Banner sobre el uso de esta integración, para que finalmente puedan brindar las capacitaciones al equipo de admisión de la universidad. Cabe resaltar que las capacitaciones se realizaron con los ambientes de pruebas de CRM y Banner.

CAPÍTULO IV. RESULTADOS

1. Cantidad de incidentes

Para la comparación de la integración entre FOX-CRM y Banner-CRM, se consideraron los semestres de despliegue de estas integraciones como punto de partida, hasta 4 semestres calendario hacía adelante.

Los incidentes fueron extraídos de 2 sistema:

- GLPI: un sistema de atención de tickets, ahora este sistema ha sido dado de baja en la universidad y solo se mantiene a manera de consulta. De aquí pude extraer los incidentes reportados catalogados como “Vinculación de código FOX” sobre la integración desfasada CRM-FOX. Los periodos de trabajo de FOX que fueron consultados fueron del 2013-07-01 hasta el 2015-06-30 (4 semestres calendario).
- ServiceNow: este sistema es usado por la red Laureate hasta la actualidad para la atención de tickets y muchas otras gestiones adicionales como pases a producción. De este sistema pude extraer los incidentes reportados como “Errores de integración” o “No aparece ID Banner en CRM”. Los periodos de considerados para la integración Banner-CRM fueron del 2018-07-01 hasta el 2020-06-30 (4 semestres calendario).

Inicialmente hice una limpieza de los tickets para lograr a las catalogaciones deseadas ya que las catalogaciones que hacían los sistemas por defecto no me permitían tener visibilidad del incidente real que estaba descrito en el ticket.

Adicional a ello, en muchos casos los tickets contenían más de 1 caso reportado. Pude mejorar la visibilidad de casos reportados analizando el texto descrito en los tickets buscando códigos de estudiante. Todo código de estudiante encontrado en la descripción del ticket fue validado contra una base de códigos de estudiante válidos. El detalle de esta información no puede ser anexada dado que es información confidencial de la universidad.

1.1. Incidentes CRM-FOX

Al extraer los incidentes reportados a la salida inicial de CRM junto con el sistema FOX, podemos ver que hay una gran alza de incidentes en el primer año de adopción. Luego en los siguientes semestres podemos observar una estabilización del servicio, pero aún con un elevado número de incidentes.

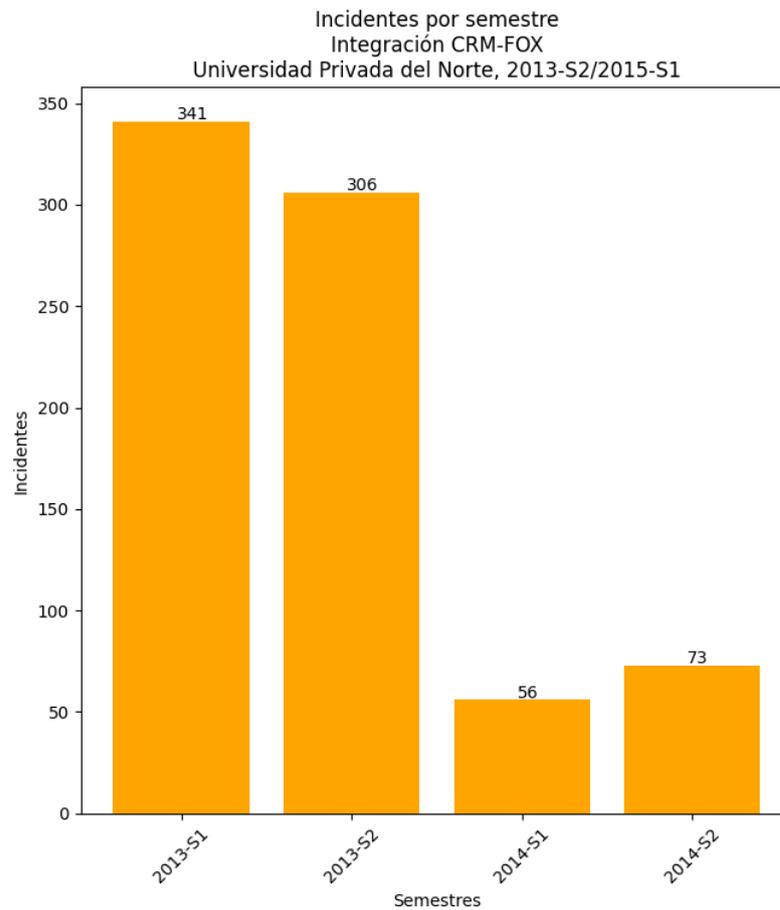


Ilustración 33 - Cantidad de incidentes CRM-FOX por semestre calendario

Fuente: Elaboración propia

1.2. Incidentes CRM-Banner

Durante el lanzamiento de la integración también nos encontramos con un alza de casos, pero es mucho más controlada. Y luego disminuyó radicalmente en el siguiente semestre comparado con el segundo semestre de CRM-FOX.

Los incidentes iniciales se han dado por caídas de servicio en Banner, ya que, en una etapa inicial como institución, aún nos encontrábamos en una etapa de adaptación.

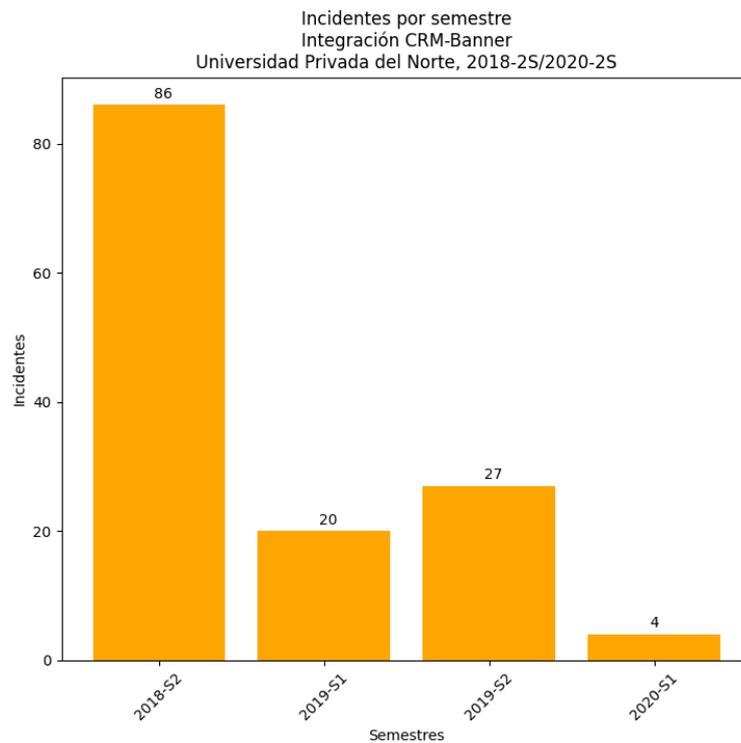


Ilustración 34 - Cantidad incidentes CRM-Banner por semestre calendario

Fuente: Elaboración propia

1.3. Disminución de incidentes

Si contrastamos la información tenemos una gran disminución de incidentes en comparación a la etapa de adaptación, como también en la etapa de estabilidad del uso de las integraciones CRM-FOX y CRM-Banner.

En el siguiente gráfico podemos ver la disminución porcentual de los incidentes distribuida por semestres calendario.

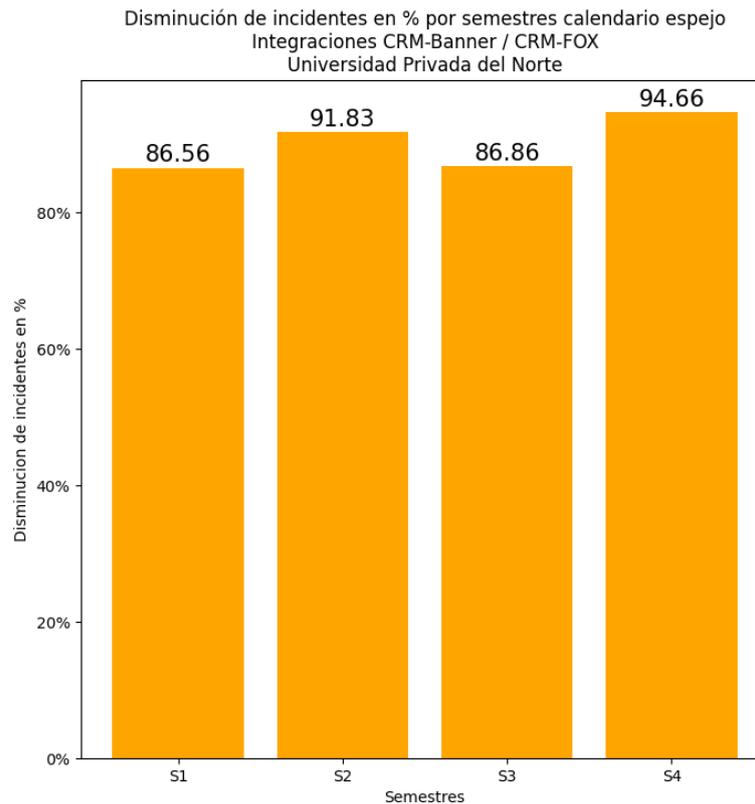


Ilustración 35 - Disminución de incidentes en % por semestres calendario espejo

Fuente: Elaboración propia

Esta disminución se mantiene en una media de 89.98%, una cifra muy cercana a cada promedio semestral, lo cual nos da confianza en que la efectividad de la integración CRM-Banner desarrollada impactó de manera uniforme en periodos de tiempo espejo comparados de la integración CRM-FOX. Es una gran tasa de efectividad en cuestión de disminución de incidentes.

2. Tiempos de atención

Ahora exponemos los tiempos de resolución de los tickets ingresados para cada sistema. Se han promediado los tiempos de atención de cada ticket por semestres calendario por los mismos periodos analizados para el conteo de incidentes.

2.1. Tiempos de resolución tickets CRM-FOX

Podemos ver que la cantidad de horas que tomaba en resolver un ticket con FOX era relativamente alta. 4 horas en el mejor de los periodos hasta 18 horas.

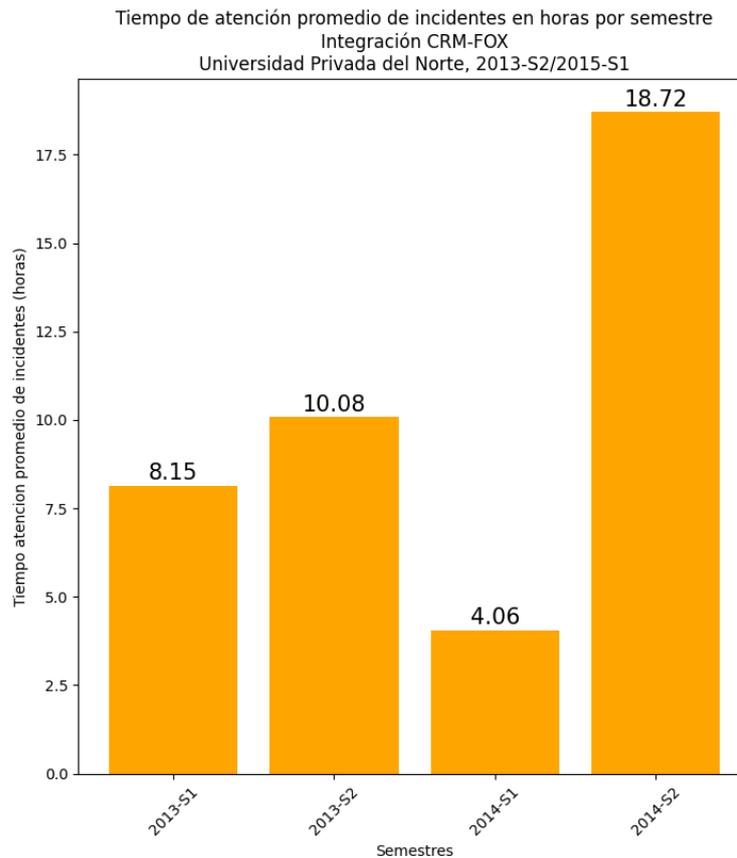


Ilustración 36 - Tiempo de atención promedio de incidentes en horas por semestre Integración CRM-FOX

Fuente: Elaboración propia

2.2. Tiempos de resolución tickets CRM-Banner

Para la atención de los incidentes con Banner vemos que los tiempos de atención promedio de los tickets se mantienen en el umbral no mayor de 2 horas. Si bien los tiempos de atención han mejorado, a esto le podemos adicionar la menor cantidad de incidentes generados con la integración de CRM-Banner para visualizar la amplificación del beneficio de la solución. Ambos factores tienen un factor multiplicador para la mejora de los tiempos de ejecución del proceso de matrícula.

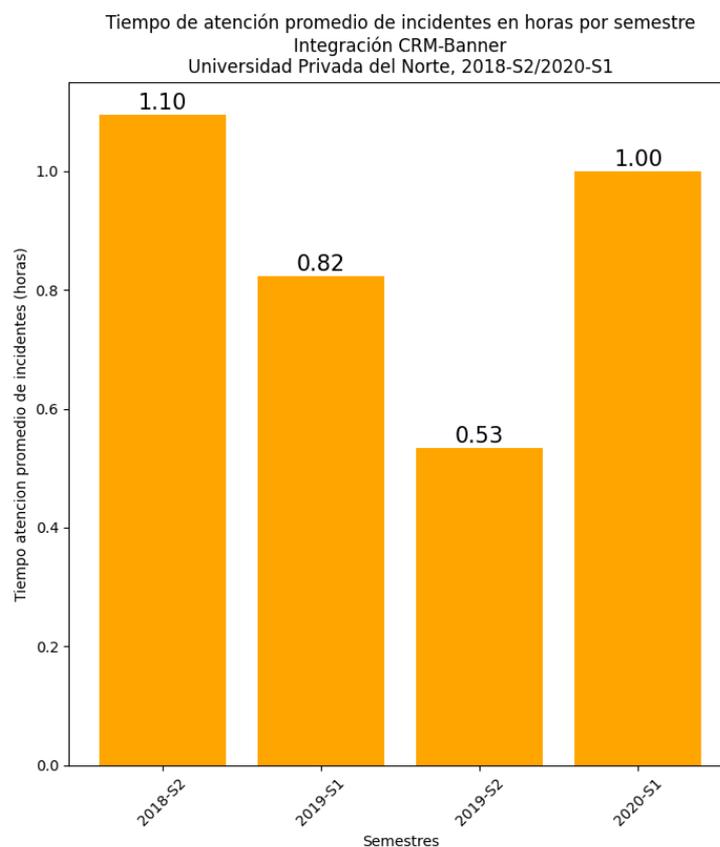


Ilustración 37 - Tiempo de atención promedio de incidentes en horas por semestre Integración CRM-Banner

Fuente: Elaboración propia

2.3. Tiempos de resolución tickets CRM-FOX

Podemos apreciar un gran decremento en los tiempos de resolución de los tickets de incidentes con las integraciones CRM-Banner / CRM-FOX.

Calculamos una media de 78.71% de decremento de tiempos de resolución. También es una cantidad significativa de mejora sobre los tiempos recuperados donde el usuario funcional del área de Admisión puede ocupar para otras funciones o en atención a más clientes.

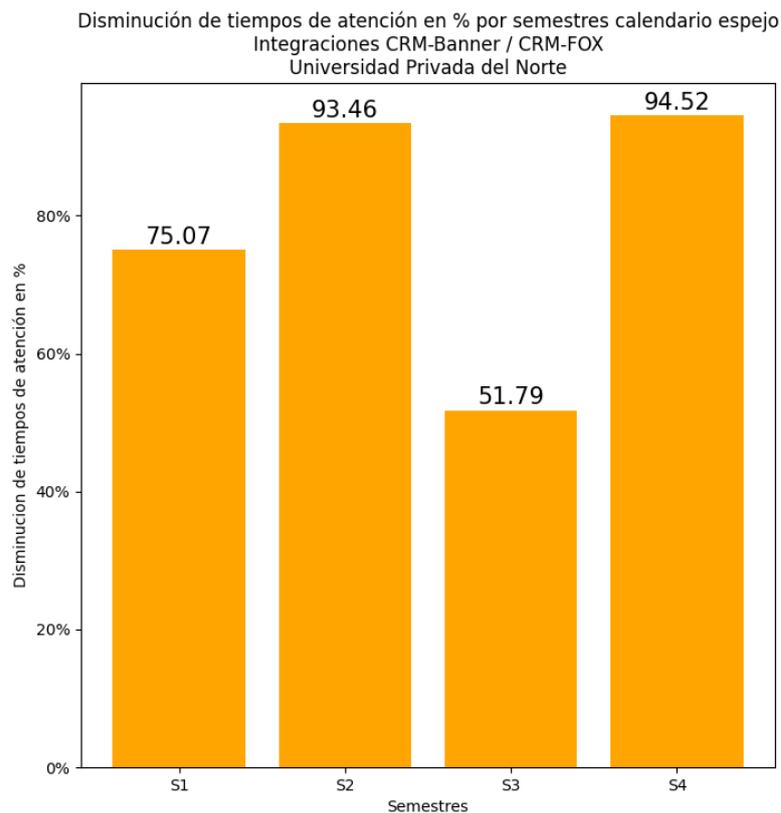


Ilustración 38 - Disminución en tiempos de atención en % por semestres calendario espejo

Fuente: Elaboración propia

3. Acumulado de horas invertidas en atención de incidentes

Para obtener el resultado efectivo de la solución, vamos a exponer la cantidad de tiempo invertido en horas por los incidentes resueltos.

3.1. Acumulado de horas invertidas en atención de incidentes en integración CRM-FOX

Tenemos un total de 5033 horas invertidas en la resolución de incidentes en el total del espacio de tiempo analizado.

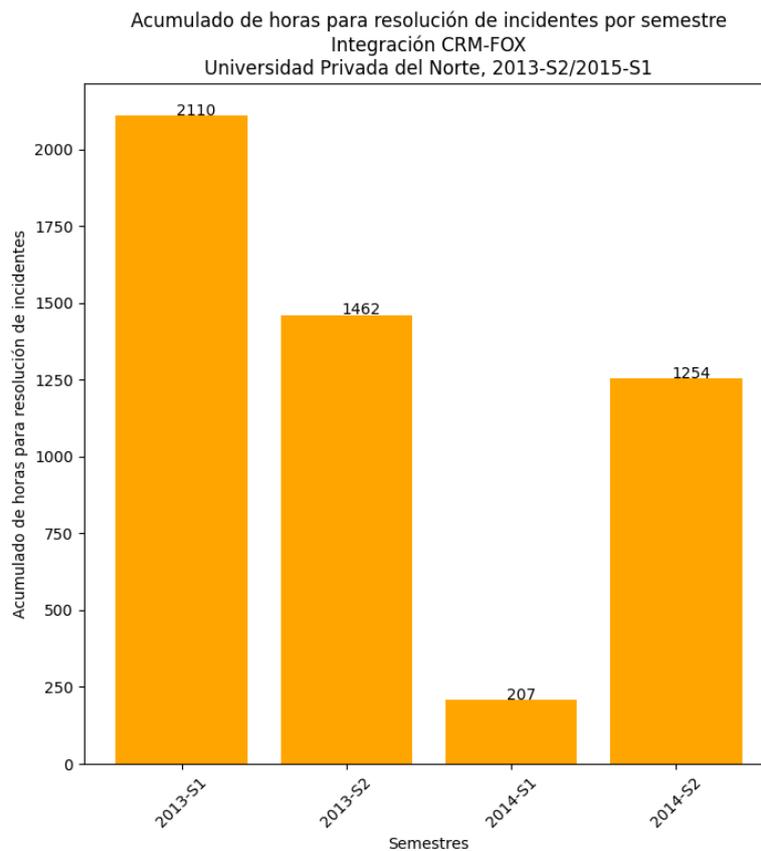


Ilustración 39 - Acumulado de horas para resolución de incidentes por semestre Integración CRM-FOX

Fuente: Elaboración propia

3.2. Acumulado de horas invertidas en atención de incidentes en integración CRM-Banner

Tenemos un total de 95 horas invertidas en la resolución de incidentes en el total del espacio de tiempo analizado.

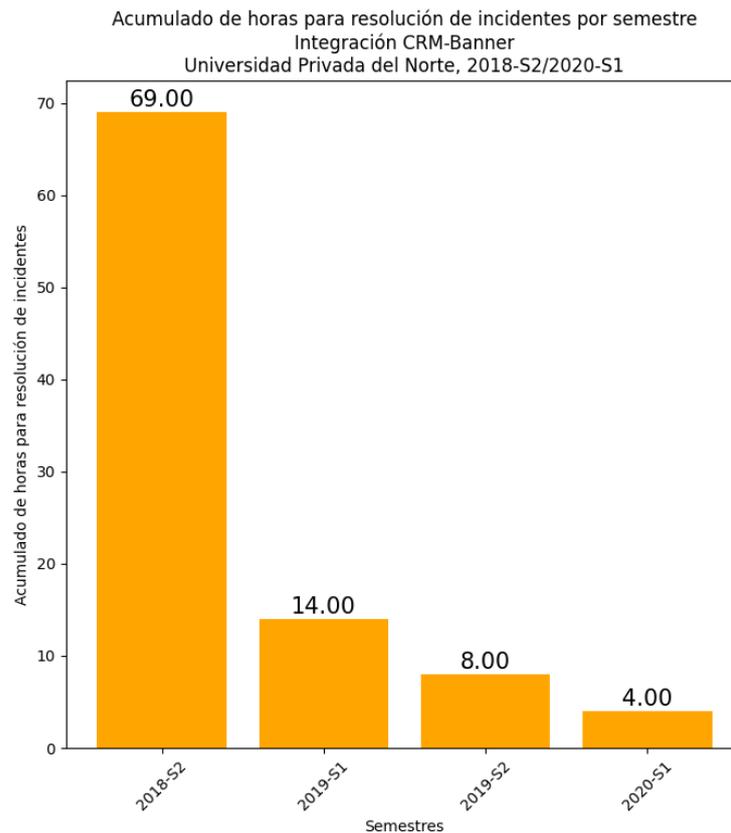


Ilustración 40 - Acumulado de horas para resolución de incidentes por semestre Integración CRM-Banner

Fuente: Elaboración propia

3.3. Disminución porcentual de horas invertidas para la atención de incidentes

El promedio porcentual de la disminución de horas invertidas en atención de incidentes es 97.9%. Esto es sustancialmente una ganancia de 4938 horas para el negocio.

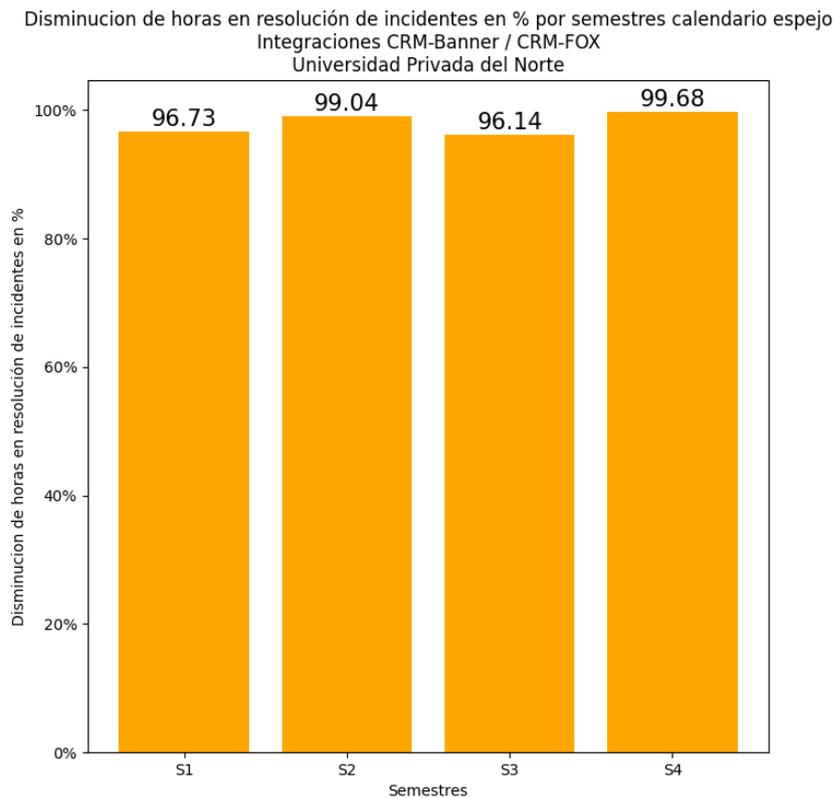


Ilustración 41 - Disminución de horas en resolución de incidentes en % por semestres calendario espejo Integraciones CRM-Banner / CRM-FOX

Fuente: Elaboración propia

CAPÍTULO V. CONCLUSIONES

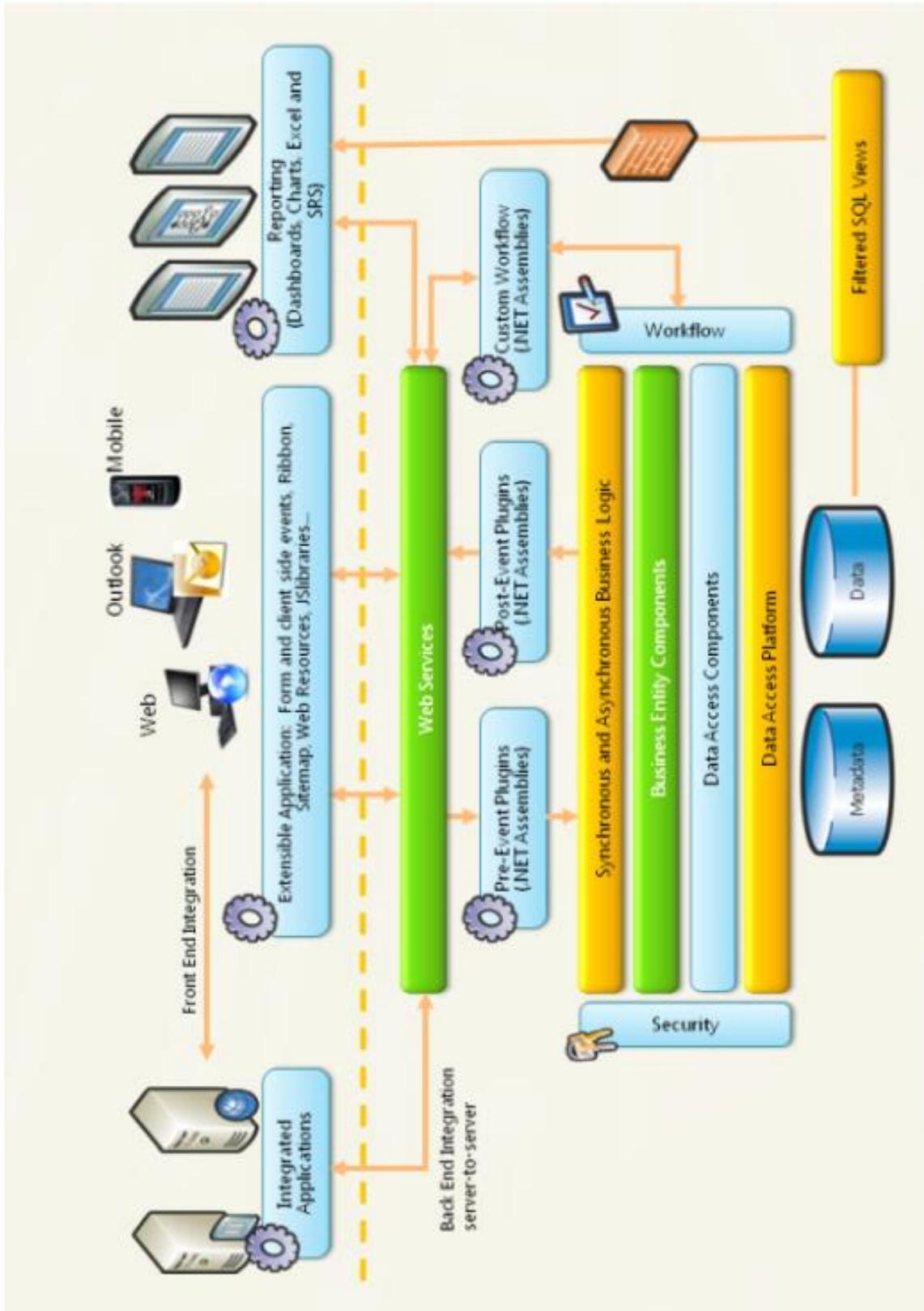
- Se lograron cumplir los procesos internos de desarrollo, pruebas y despliegue de aplicaciones, utilizando la metodología Scrum para el desarrollo de la presente solución integradora.
- La cantidad de incidentes fue disminuida drásticamente con casi un 90% de efectividad para el proceso de matrícula a lo largo de 4 semestres calendario desde la salida de la integración. Esto refleja un gran ahorro en tiempo, dinero y disminución de incomodidad al estudiante y trabajadores de la universidad.
- Al ser menores los tiempos de atención de los incidentes CRM-Banner, y al ser menores la cantidad de incidentes, tenemos un efecto multiplicador aportado por la solución. El beneficio resultante es de un 84.34% para el proceso de matrículas para la universidad. Se calculó como el promedio de la cantidad de incidentes disminuidos (89.98%) y el promedio de la cantidad de horas disminuidas para la atención de los incidentes (78.71%).
- La cantidad de horas ganadas en los periodos de tiempo analizados es de 4938. Estas horas ganadas son distribuidas entre los analistas de sistemas que han dejado de atender estos incidentes y los trabajadores del área de admisión que han podido invertir este tiempo en el proceso de matrícula.

RECOMENDACIONES

- Mejorar el diseño de la interfaz del servicio web de la integración que da cara a los usuarios funcionales.
- Implementar una alta disponibilidad en el servicio de RabbitMQ adicionando un servidor esclavo adicional.
- Se recomienda la migración del sistema CRM Dynamics 2011 a la última versión Dynamics 365, dado que pronto Microsoft dejará de brindar soporte para la versión 2011.
- Implementar la captura de más eventos ocurridos en Banner y Peoplesoft. Incluyendo el registro de cursos, retiro de cursos, anulación de matrícula o pagos de la matrícula o cuota inicial, con la finalidad de poder guardar estos datos en CRM en vivo. Estos datos actualmente viajan a CRM pero por medio de ETLs y otros procesos adjuntos programados bajo un horario rígido. Dado que, si alguno de estos procesos se detiene o tiene algún error, pueden impactar de gran manera los reportes que dependen de esta información ya que las ventanas de ejecución tienen un espacio muy amplio y demandaría ejecutar estos procesos manualmente desde el punto de falla en adelante. Teniendo un flujo de información en vivo y constante, junto con el trabajo de colas de mensajes, si algún proceso detiene su ejecución por alguna falla, este proceso puede restaurarse desde el punto donde se detuvo.

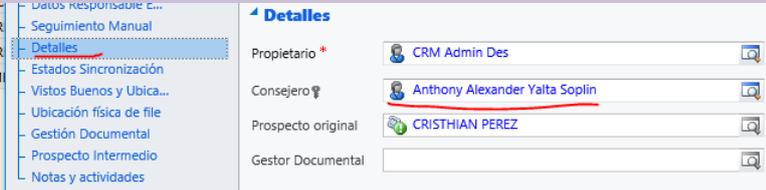
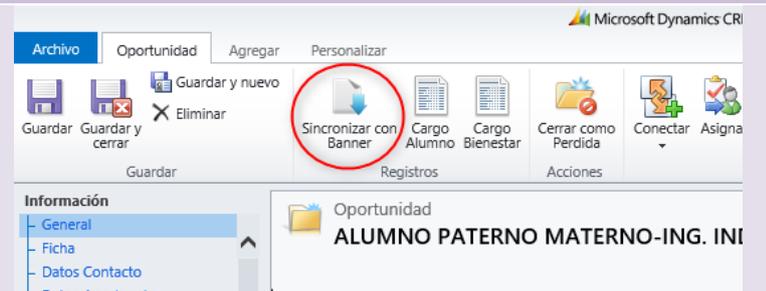
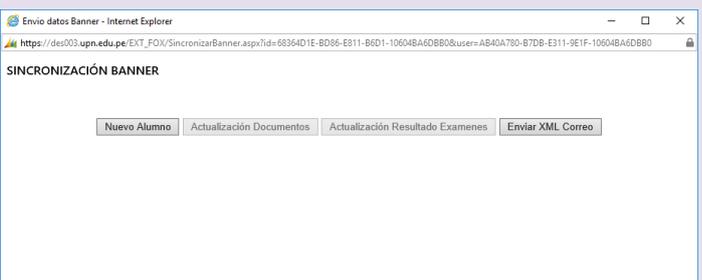
REFERENCIAS

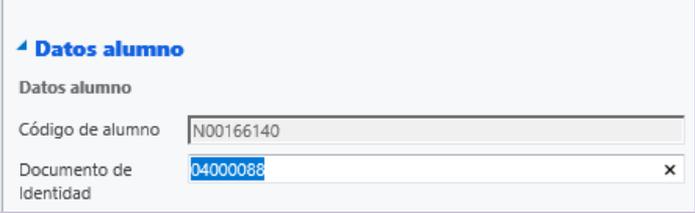
- Aguilar, L. J. (2008). Fundamentos de programación, algoritmos, estructura de datos y objetos. (4). (McGRAW-HILL, Ed.) Madrid, España.
- Bianco, P. (2007). Evaluating a Service-Oriented Architecture. Carnegie Mellon University . Recuperado el 24 de Febrero de 2014, de <https://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=8443>
- CRM Software Blog. (19 de Enero de 2011). Microsoft Dynamics CRM 2011: What is xCRM and XRM. Obtenido de <https://www.crmsoftwareblog.com/2011/01/microsoft-dynamics-crm-2011-what-is-xcrm-and-xrm/>
- Ellucian. (Febrero de 2021). Ellucian Banner. Obtenido de <https://www.ellucian.com/solutions/ellucian-banner>
- Microsoft. (Noviembre de 2010). Building Business Applications with Microsoft Dynamics CRM 2011. Recuperado el 25 de Febrero de 2021, de Building Business Applications with Microsoft Dynamics CRM 2011
- Microsoft. (17 de Marzo de 2014). Aplicaciones de servicios en tres capas. Recuperado el 24 de Febrero de 2021, de [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648105\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648105(v=pandp.10))
- Microsoft. (31 de Mayo de 2018). Servicios de Windows. Recuperado el 24 de Febrero de 2021, de <https://docs.microsoft.com/es-es/windows/win32/services/services?redirectedfrom=MSDN>
- Microsoft. (2021). ¿Qué es .NET Framework? Recuperado el 24 de Febrero de 2021, de <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>
- Node.js. (16 de Abril de 2020). Acerca de Node.js®. Recuperado el 03 de Febrero de 2021, de <https://nodejs.org/es/about/>
- Packt. (3 de Enero de 2013). *Overview of Microsoft Dynamics CRM 2011*. Obtenido de <https://hub.packtpub.com/overview-microsoft-dynamics-crm-2011/>
- Paget, V. (18 de Septiembre de 2018). How To Get The Most Out Of Your Student Information Systems. Recuperado el 25 de Febrero de 2021, de <https://blog.boardingware.com/get-the-most-from-your-student-information-system/>
- Universidad Privada del Norte S.A.C. (31 de 10 de 2019). PROCESO DE ADMISIÓN PARA ESTUDIANTES DE PREGRADO - COR-P-OPE-ADM-04 V7. Perú.
- Vinoski, S. (Diciembre de 2006). Advanced Message Queuing Protocol. Obtenido de https://web.archive.org/web/20090207140603/http://dsonline.computer.org/portal/site/dsonline/menuitem.6dd2a408dbe4a94be487e0606bcd45f3/index.jsp?&pName=dso_level1_article&TheCat=1001&path=dsonline%2F2006%2F12&file=w6tow.xml&%3Bjsessionid=LR6hp4hBgfJRWv1vw



Anexo 2 - Diagrama de componentes de CRM Dynamics 2011

Fuente: (Bianco, 2007)

Nº de Act.	Descripción de la Actividad	Resultado Esperado	Resultado Obtenido
1	Se inicia sesión a CRMQA (https://des003.upn.edu.pe/UPNQA/) con un usuario Consejero (crmadminde s02) y se ingresa un nuevo prospecto, se califica y abrimos la oportunidad	Se visualiza la oportunidad recién creada	Revisar documento “Creación y calificación manual de un nuevo alumno.docx”
2	Creamos una entrevista	La entrevista sea guardada en el sistema	Revisar documento “Creacion de una entrevista.docx”
3	Agregamos un usuario al campo Consejero	El campo consejero debe tener algún usuario	
4	Abrimos la ventana de sincronización Banner. Aseguramos que	Visualizamos el botón de “Nuevo Alumno” habilitado.	 

5	Finalmente debemos recibir el Id del Banner del postulante cuando se pasen todas las validaciones.	Tener un ID de Banner en el Contacto del Alumno.	
---	--	--	--

Anexo 3 - Caso de pruebas funcionales