



Conexión Universal de Aplicaciones Institucionales

Primera etapa - Conexión SIIU/Santander Universidades

INTRODUCCIÓN

Objetivo del Proyecto

Desarrollar una API que facilite la comunicación entre el SIIU y diversas plataformas y sistemas externos, con el objetivo de gestionar datos de forma eficaz. Esta iniciativa busca apoyar la integración con terceros, adaptándose a las demandas de la tecnología actual.

Alcance del Proyecto

Se incluirá el monitoreo de eventos a los procesos actuales del SIIU involucrados, los cuales deberán ejecutar disparadores (*triggers*) para informar a la API los eventos (mediante *webhooks*) de acuerdo a la estructura definida por la Oficina de Proyectos de Software de la UVAQ. Así como, también se deberá desarrollar una API que responda a estos eventos, mantenga la información actualizada en una Base de Datos y se encuentre preparada y disponible para ser consumida por alguna aplicación de terceros según las prioridades y conveniencias de la Universidad. El desarrollo e implementación de la API deberá ser responsabilidad de un equipo interno de desarrollo por ser de manejo crítico, dinámico y escalable.

Justificación

Un ecosistema de aplicaciones requiere del intercambio continuo de información entre las mismas de forma segura y eficiente. La construcción de sistemas modernos implica conexiones flexibles y completas que sean fácilmente adaptables a nuevos proyectos y aplicaciones actualmente desconectadas. Se busca la construcción de una API o interfaz de comunicación para aplicaciones la cual se irá nutriendo con el tiempo con nuevas funciones. Inicialmente, se construirá la versión de *Expediente Digital* de los miembros de la comunidad Universitaria, el cual será consultable por aplicaciones autorizadas para obtener datos vigentes (de 4 a 6 horas desde su última

consulta) o en tiempo real, según sea necesario. La Conexión Universal de Aplicaciones inicialmente será de consulta de los datos del sistema ERP actual (SIIU) y en el futuro deberá también incluir funciones para la actualización de datos.

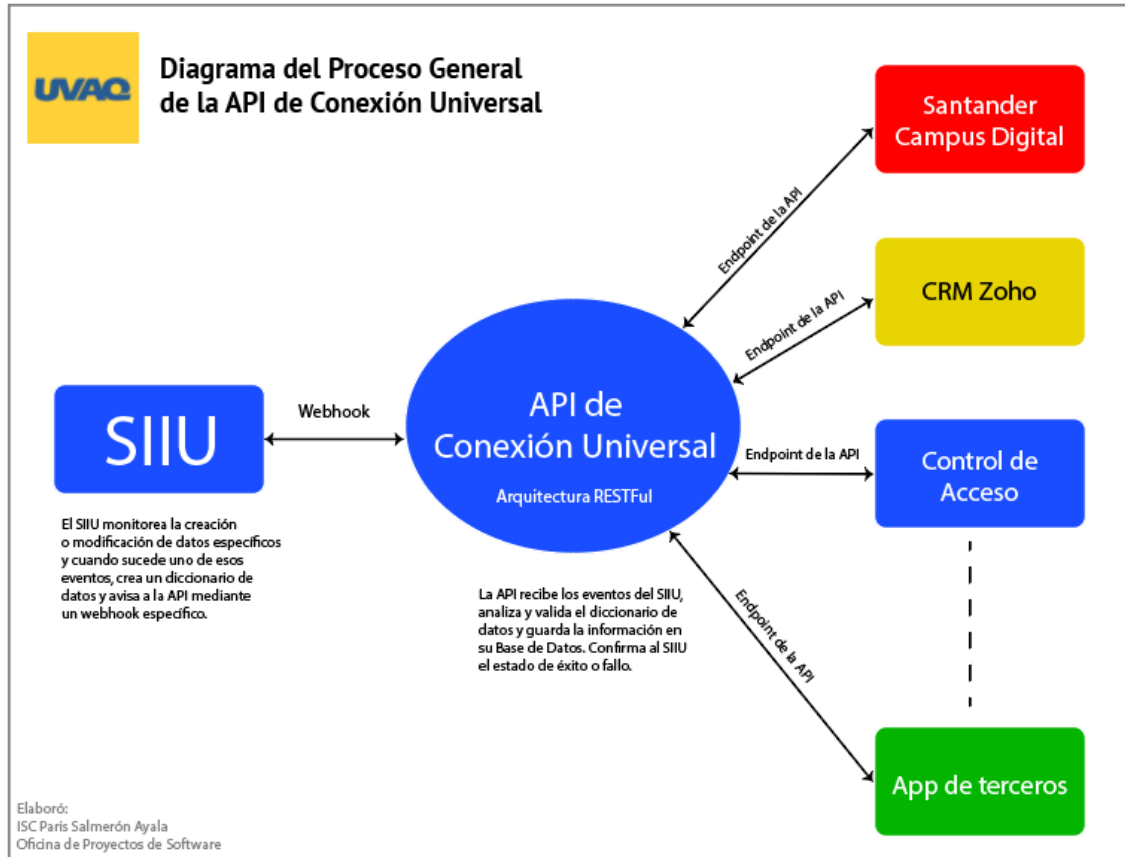


Figura 1 – Diagrama del Proceso General de la API

STAKEHOLDERS DEL PROYECTO

Equipo Interno

Definir el equipo interno de desarrollo, así como sus roles y responsabilidades que desarrollará la API.

Evolución TI

Desarrollará e implementará los ajustes necesarios en el SIIU para monitorear y disparar los eventos acordados en las especificaciones técnicas del proyecto.



Santander Universidades

Colaborará en la integración de la aplicación Campus Digital, asegurando compatibilidad y eficacia en el uso de la API.

REQUERIMIENTOS FUNCIONALES

Datos requeridos:

Identidad Personal

Incluye información básica de identificación de usuarios, como nombres, apellidos, CURP, y datos de contacto. Esencial para validar la identidad de usuarios dentro del sistema.

1. Nombres de personas
2. Apellido paterno de personas
3. Apellido materno de personas
4. Teléfonos de contacto
5. Direcciones de correo electrónico
6. Número de documento de identidad (CURP)
7. Países emisores del documento de identidad
8. Códigos de países de nacionalidad
9. Fechas de nacimiento de las personas
10. Identificador del sexo de las personas
11. Fotografías del usuario
12. Firma digitalizada del usuario

Expediente académico

Contiene datos académicos de usuarios que son alumnos, matrículas, y registros de cursos. Importante para aplicaciones que requieran verificar la trayectoria educativa de los alumnos.

13. Etiquetas de datos adicionales de alumno en la universidad
14. Valores adicionales de alumno en la universidad
15. Matrícula SEP
16. Nivel de ingreso
17. Periodo de ingreso
18. Información de la persona para credenciales en la aplicación Campus Digital
19. Información de la universidad para credenciales
20. Grupos de notificaciones para credenciales



- 21. Nombres de cursos
- 22. Tipos de estudio (grado, maestría, etc.)
- 23. Nombres de roles de usuario
- 24. Identificador del rol del usuario
- 25. Identificador de la universidad
- 26. Grupos obligatorios para notificaciones en la aplicación Campus APP
- 27. Grupos opcionales para notificaciones en la aplicación Campus APP
- 28. Identificadores de usuario en la universidad
- 29. Fechas de matriculación
- 30. Cursos del alumno
- 31. Grado actual del alumno
- 32. Estado de los cursos (vacío, cursando, aprobado, etc.)
- 33. Calificación obtenida en el curso
- 34. Datos adicionales del usuario en la universidad
- 35. Información sobre la universidad asociada al usuario
- 36. Deuda total del alumno
- 37. Saldo total vencido del alumno

Admisiones

Abarca información relacionada con el proceso de admisión de nuevos estudiantes, desde su escuela de procedencia hasta comentarios sobre sus intereses académicos.

- 38. Escuela de procedencia de grado previo
- 39. Medio por el que se enteró
- 40. Lista de nombres de los asesores educativos
- 41. Grado de estudio actual
- 42. Grado de interés
- 43. Oferta académica de interés
- 44. Medio de contacto preferido
- 45. Comentarios

Acceso Seguro

Relacionado con el control físico y lógico de accesos, incluye información sobre vehículos y niveles de acceso permitidos dentro de la institución.

- 46. Matrículas de vehículos



- 47. Tipos de vehículo
- 48. Modelo y color del vehículo
- 49. Fotografía del vehículo
- 50. Horarios permitidos de acceso
- 51. Áreas de estacionamiento autorizadas
- 52. Niveles de acceso de usuario (estudiante, docente, administrativo, visitante, etc.)
- 53. Horarios de acceso permitidos para el usuario
- 54. Datos de contacto de emergencia
- 55. Información de seguros y/o documentación vehicular

Servicios requeridos:

Para que el SIIU pueda suministrar los datos comentados, se debe diseñar una serie de eventos que cubran todas las acciones posibles para comunicar adecuadamente los cambios efectuados en la información de los usuarios.

Se definirá una lista exhaustiva de eventos requeridos, incluyendo la creación, actualización y eliminación de usuarios, así como eventos específicos para admisiones y control de acceso.

A continuación se presenta un breve esquema de eventos relevantes para cada categoría de datos:

Eventos de Identidad Digital e Información Personal:

- UsuarioCreado: Se emite cuando se crea un nuevo usuario en el sistema.
- UsuarioActualizado: Se emite cuando se actualiza la información de un usuario existente.
- UsuarioEliminado: Se emite cuando un usuario es eliminado del sistema.

Eventos de Información de Contacto

- ContactoUsuarioActualizado: Se emite cuando se actualiza la información de contacto de un usuario.

Eventos de Información Académica

- PerfilAcademicoCreado: Se emite cuando se crea un nuevo perfil académico para un usuario.



- **PerfilAcademicoActualizado:** Se emite cuando se actualiza el perfil académico de un usuario.
- **MateriaAsignada:** Se emite cuando se asigna una nueva materia a un usuario.
- **MateriaActualizada:** Se emite cuando se actualiza la información de una materia asignada (por ejemplo, cambio de calificación o estado).

Eventos de Admisión

- **SolicitudAdmisionCreada:** Se emite cuando un prospecto completa una solicitud de admisión.
- **SolicitudAdmisionActualizada:** Se emite cuando se actualizan los detalles de una solicitud de admisión existente.

Eventos de Control de Acceso

- **AccesoConcedido:** Se emite cuando se concede acceso a un usuario o vehículo.
- **AccesoDenegado:** Se emite cuando se deniega acceso a un usuario o vehículo.
- **AccesoUsuarioActualizado:** Se emite cuando se actualizan los detalles de control de acceso de un usuario.
- **Eventos Financieros**
- **EstadoFinancieroUsuarioActualizado:** Se emite cuando se actualiza el estado financiero de un usuario, como cambios en la deuda o en el saldo vencido.

Cada uno de estos eventos deberá incluir detalles específicos sobre el cambio realizado, como el ID del usuario afectado, timestamps, y un payload que contenga la información específica del cambio. Por ejemplo, un evento de **UsuarioActualizado** podría incluir qué campos específicos se actualizaron junto con los valores antiguos y nuevos.

Para la implementación de la API, el SIIU necesitará:

1. **Detectar Cambios:** Implementar la lógica de negocio dentro del SIIU para detectar cambios en los datos involucrados.
2. **Generar Eventos:** Crear los eventos con la estructura definida en las especificaciones técnicas cada vez que se detecte un cambio en los datos.
3. **Publicar Eventos:** Utilizar los webhooks proporcionados para publicar los eventos generados con la debida estrategia de fallos, según lo indicado.
4. **Generar, mantener y resguardar los logs** de los eventos generados.



5. Documentación: Proveer la documentación detallada para cada evento, incluyendo su estructura y el tipo de cambios que representa.



Especificación Técnica para Implementación de Webhooks en el SIU

1. Objetivo de la Implementación

Proporcionar una integración cercana al tiempo real entre el SIU y la API de Santander Universidades para sincronizar datos relevantes de la identidad digital de los usuarios, procesos de admisión y control de acceso seguro, mejorando así la eficiencia operativa y la experiencia del usuario.

2. Requisitos de Arquitectura

Endpoint de Recepción:

El SIU deberá enviar las notificaciones de eventos a los siguientes endpoints RESTful específicos, según el tipo de evento:

- Identidad Digital y Personal: <https://www.uvaq.edu.mx/api/v1/events/personal>
- Información de Admisión: <https://www.uvaq.edu.mx/api/v1/events/admission>
- Control de Acceso: <https://www.uvaq.edu.mx/api/v1/events/access>

Método HTTP:

Todos los webhooks utilizarán el método POST para enviar los eventos.

Cabeceras Requeridas:

- Content-Type: application/json
- Authorization: Bearer <token> (El token será proporcionado por la API para asegurar la autenticidad de la solicitud).

Seguridad:

- Autenticación y Autorización: Se utilizarán tokens JWT para autenticar y autorizar las solicitudes del webhook. La API proporcionará un token al SIU que debe incluirse en cada solicitud.
- Cifrado: Uso obligatorio de HTTPS para asegurar todas las comunicaciones.
- Validación de Payloads: Se espera que el SIU firme los payloads de eventos utilizando una clave secreta compartida. La API validará esta firma para cada evento recibido.



Estrategia de Intentos Fallidos: En caso de fallo (códigos de respuesta HTTP 4xx o 5xx), el SIU debe reintentar la entrega del evento hasta 5 veces, con un intervalo exponencial de espera entre intentos, comenzando el primer reenvío al segundo minuto del primer intento fallido. Después de los intentos fallidos, si ninguno fue exitoso, se deberá notificar al administrador de la API a través un de correo electrónico donde se incluya el log completo y detallado de la operación fallida, para realizar una intervención manual.

3. Formato de los Eventos

Cada evento deberá seguir una estructura JSON estándar, clara y consistente, como en el siguiente ejemplo; la cual deberá estar claramente documentada en un archivo YAML:

```
{
  "eventId": "uuid",
  "eventType": "UsuarioCreado",
  "eventTimestamp": "2024-03-12T15:04:05Z",
  "resourceId": "12345",
  "data": {
    // Detalles específicos del evento
    "idUser": "12345",
    "nombre": "Juan",
    "apellidos": "Pérez López",
    "email": "jperez@uvaq.edu.mx"
  }
}
```

4. Lista de Eventos y Triggers

Se requiere la implementación de los siguientes eventos, cada uno con su respectivo trigger en el SIU:

1. **Identidad Digital y Personal:** UsuarioCreado, UsuarioActualizado, UsuarioEliminado
2. **Información de Admisión:** SolicitudAdmisionCreada, SolicitudAdmisionActualizada, SolicitudAdmisionEliminada
3. **Control de Acceso:** AccesoConcedido, AccesoDenegado



(Especificar detalladamente cada tipo de evento, su trigger correspondiente y los datos esperados en el payload).

5. Manejo de Versiones

Se considera desde el inicio introducir versionamiento con la ruta `´/v1/´` para facilitar futuras actualizaciones sin interrumpir las integraciones existentes en el momento.

6. Documentación y Soporte

Proporcionar documentación completa de la API de webhooks, incluyendo especificaciones técnicas de los eventos, ejemplos de payloads y guías de integración. También se debe especificar el protocolo de soporte técnico para resolver incidencias.

7. Pruebas y Validación

Se deberá considerar contar con un entorno de pruebas y proporcionar acceso al equipo de la API para realizar pruebas integrales antes del lanzamiento a producción.

8. Observaciones finales

La presente especificación técnica detallada debe ser revisada y acordada con Evolución TI para asegurar que la implementación cumpla con todas las necesidades y requisitos de la API de Santander Universidades.

Atentamente,

I.S.C. Paris Salmerón Ayala

Jefe de Oficina de Proyectos de Software

“Educar en la verdad”



Estructura para la API de Conexión Universal para Apps Institucionales

OBJETOS

Objeto: PersonalInformation

- Nombre
- Apellidos
- Fecha de Nacimiento
- Sexo
- Fotografía de identificación
- Firma digitalizada

Objeto: ContactInformation

- Teléfono de contacto
- Celular
- Dirección de correo electrónico personal
- Dirección de correo electrónico institucional
- Medio de contacto preferido (WhatsApp, llamada, correo, otro)

Objeto: Identification

- CURP
- Número de identificación (podría ser la matrícula SEP)
- Nacionalidad (código de país)

Objeto: AcademicProfile

- Escuela de procedencia
- Estudio actual (Ejemplo: "1er semestre de preparatoria")
- Grado de estudios de interés (Licenciatura, Posgrado)
- Oferta académica de interés
- Resumen de materias asignadas (con grado, materia, estado y calificación)

Objeto: AdmissionData

- Medio por el que se enteró (redes sociales, página web, etc.)
- Asesor Educativo asignado (nombre del asesor)
- Dudas o comentarios

Objeto: AccessControl



- Niveles de acceso (estudiante, facultad, staff, visitante)
- Horarios de acceso permitidos
- Historial de acceso
- Matrícula y detalles del vehículo para el control vehicular
- Dispositivos de acceso (tarjetas RFID, etiquetas NFC)
- Datos biométricos para el control de acceso (opcional)

Objeto: FinancialInformation

- Deuda total
- Saldo vencido total

Objeto: UserUniversity

- Identificador de usuario en la universidad
- Roles de usuario (estudiante, profesor, administrativo)
- Grupos obligatorios y opcionales de notificaciones
- Información de la universidad (identificador y otros datos relevantes)
- Fecha de matriculación

Objeto: EmergencyInformation

- Datos de contacto en caso de emergencia

Objeto: InsuranceInformation (Solo si es relevante para el control vehicular)

- Información de seguros y documentación vehicular

ENDPOINTS

Objeto	Descripción	Endpoints (CRUD)
PersonalInformation	Información personal del usuario	GET /users/{id}/personal
		PUT /users/{id}/personal
ContactInformation	Información de contacto del usuario	GET /users/{id}/contact
		PUT /users/{id}/contact
Identification	Identificación del usuario	GET /users/{id}/identification
		PUT /users/{id}/identification

AcademicProfile	Perfil académico del usuario	GET /users/{id}/academic
		POST /users/{id}/academic
		PUT /users/{id}/academic/{academicId}
AdmissionData	Datos del proceso de admisión	GET /admissions/{id}
		POST /admissions
		PUT /admissions/{id}
AccessControl	Control de acceso para usuarios y vehículos	GET /access/{id}
		POST /access/{id}
		PUT /access/{id}
FinancialInformation	Información financiera del usuario	GET /users/{id}/financial
		PUT /users/{id}/financial
UserUniversity	Relación del usuario con la universidad	GET /users/{id}/university
		PUT /users/{id}/university
EmergencyInformation	Información de contacto de emergencia	GET /users/{id}/emergency
		PUT /users/{id}/emergency
InsuranceInformation	Información sobre el seguro y vehículo	GET /vehicles/{id}/insurance
		PUT /vehicles/{id}/insurance

Notas importantes:

CRUD representa Crear (POST), Leer (GET), Actualizar (PUT) y Borrar (DELETE), aunque en la tabla solo se incluyen los ejemplos más representativos para cada objeto pues conforme el análisis siga avanzando se pueden agregar o ajustar. Dependiendo de los requisitos específicos, algunos objetos podrían no necesitar todos los métodos CRUD.

El endpoint DELETE no se ha incluido de momento para mantener la simplicidad, pero debería considerarse en la implementación final de la API para manejar la eliminación de datos cuando sea apropiado y seguro hacerlo.

{id} es un placeholder para el identificador único del usuario, admisión, vehículo, etc., según corresponda.

{academicId} es un placeholder para identificar registros específicos dentro del perfil académico del alumno, como por ejemplo, materias específicas.