

# SeQReXML

## API multiplataforma para gestionar transacciones seguras con Código QR a través de datos XML

**Calloni, Juan Carlos - Gioino, Mauro Daniel -  
Armando, Silvana María - Bonino, María Belén -  
Bianciotti, Andrés Hernán - Ponce, Martín Andrés - Rossi, Martín Gonzalo**  
*Universidad Tecnológica Nacional, Facultad Regional San Francisco*

### Abstract

*El proyecto consiste en una API multiplataforma que sirve a cualquier lenguaje de programación para leer, grabar y gestionar de forma segura códigos QR con formato estándar de datos XML.*

*Dicha API brinda soporte a los desarrolladores de sistemas de gestión, otorgando la funcionalidad necesaria para emplear códigos QR como herramienta de carga masiva de datos.*

*De esta forma, se resuelve la problemática de cargar grandes volúmenes de datos en sistemas de información, que en la actualidad se ingresan en forma manual; como pueden ser facturas, cheques, entre otros.*

### Palabras Clave

QR (Quick Response), seguridad, XML (eXtensible Markup Language), API (Interfaz de Programación de Aplicaciones), multiplataforma, automatización.

### Introducción

El desarrollo de dicho proyecto intenta dar solución a dos problemáticas. En primer lugar, brindar una herramienta que permita cubrir la necesidad de cargar en forma masiva documentos con un gran volumen de información en los distintos sistemas de gestión (por ejemplo sistemas administrativos, contables, financieros, entre otros) y en segundo lugar, la realización de transacciones seguras empleando códigos QR y datos en formato XML.

### Elementos del Trabajo y metodología

En una primera etapa definimos el marco teórico del proyecto analizando la presencia de antecedentes. Se definen los conceptos y proposiciones que permiten abordar la problemática planteada, ubicándola en un contexto unificado, integrándola con

conocimientos previos y ordenando los mismos para que resulten útiles al proyecto.

Luego de esto se realiza una recolección de datos en forma empírica con el fin de obtener los conocimientos necesarios para la construcción de la API y el uso de códigos QR con XML. El grupo se divide en subgrupos, donde cada uno realiza una de las siguientes actividades de investigación:

- Se investiga cómo construir una API que funcione tanto en sistemas operativos libres como privativos, de manera que pueda ser implementada y se adecúe a diferentes entornos independientemente de la plataforma que se utilice y teniendo en cuenta que sea mínimo el impacto del lado del cliente.
- Se evalúan los diferentes lenguajes de programación existentes y emergentes en forma genérica, a los fines de determinar el más adecuado para el tipo de desarrollo. Se tienen en cuenta aspectos como portabilidad, grado de madurez, curva de aprendizaje, popularidad, seguridad y soporte oficial. Un punto importante que se considera es la implementación de patrones como MVC (Model View Controller) para facilitar las tareas de desarrollo y mantenimiento del producto. Además, se pretende trabajar con lenguajes que soporten tanto desarrollo de soluciones de

escritorio como soluciones web, o en su defecto la posibilidad de crear interfaces entre diferentes lenguajes para poder dar solución a ambos escenarios.

- Se estudia en profundidad el estándar ISO/IEC 18004:2000 que define la tecnología QR principalmente para determinar dos puntos importantes. Uno de ellos es el tipo de código QR que se va a utilizar, en base a los objetivos y lo que se pretende realizar, y por otro lado se analiza la cantidad de información que pueden almacenar teniendo en cuenta el tipo de dato que se intenta incluir en los mismos.

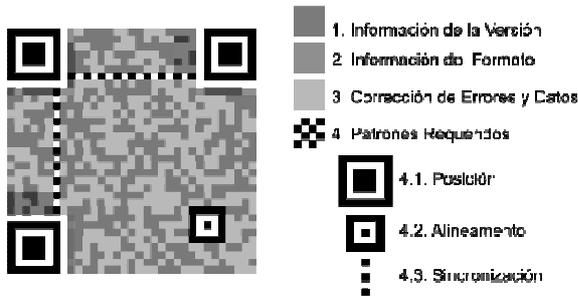


Figura 1. Estructura de un código QR

- Se investiga la estructura genérica de las diferentes categorías de comprobantes existentes en el mercado (facturas, recibos, cheques, entre otros), a los fines de determinar la cantidad de caracteres a representar. En base a esto, se busca minimizar la cantidad de códigos QR que se utilizan para representar un comprobante.
- Se indaga acerca de XML y sus distintas características como estándar de representación en intercambio de datos. Cada sección de un comprobante se representa mediante XML, dado que es un lenguaje de intercambio de información estructurada utilizado para almacenar datos de forma

legible y flexible a esquemas diversos.

- Se evalúan las diferentes alternativas existentes en materia de compresión de archivos XML, debido a la necesidad de minimizar la cantidad de códigos QR por comprobante. Se tienen en cuenta las herramientas de compresión de texto general (.zip, .gzip, .rar, entre otras) y los compresores específicos de XML (XMill, XComp, XWRT, eXalt, XGrind, entre otros). Estos compresores específicos están diseñados para aprovechar la estructura de los documentos XML y así obtener un mejor nivel de compresión.

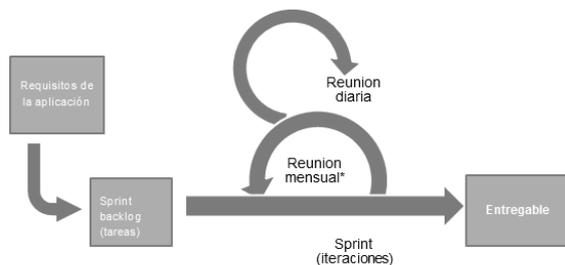
Tabla 1. HERRAMIENTAS DE COMPRESIÓN DE ARCHIVOS XML	
TEXTO GENERAL	NATIVAS
RAR	XMill
ZIP	XComp
gzip	XWRT
kgb	eXalt
7z	XMLPPM
	XGrind
	XPress
	XQueC
	XQzip

- Se analizan las distintas técnicas de cifrado de datos y algoritmos vinculados, para seleccionar la que mejor se adapte al proyecto y sea seguro. Para el almacenamiento de la información, se utiliza cifrado simétrico, que consta de 5 componentes: texto claro, algoritmo de cifrado, clave secreta, texto cifrado y algoritmo de descifrado. El algoritmo a utilizar es AES-256, un estándar a nivel mundial. La seguridad del cifrado simétrico depende de la privacidad de la clave, no de la privacidad del algoritmo. El problema principal consiste en mantener la privacidad de la clave, por lo tanto el intercambio de las mismas se realiza

a través del algoritmo RSA o el protocolo Diffie Hellman, dependiendo de las bibliotecas estándar de programación a utilizar.

Una vez finalizada la etapa de investigación, se seleccionan las tecnologías, algoritmos y técnicas a utilizar. La metodología de trabajo para gestionar el proyecto será Scrum, en la cual se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado de un proyecto. No está basada en el seguimiento de un plan sino en la adaptación continua a las circunstancias de la evolución del proyecto y está orientada a las personas más que a los procesos.

Esta metodología se basa en el desarrollo incremental, donde al final de cada iteración se dispone de cierto producto que se puede evaluar. Estas iteraciones se conocen como sprints y tienen una duración promedio de 30 días. Por cada sprint, el equipo define objetivos, tareas a realizar y responsabilidades de cada integrante a lo largo de dicha iteración.



**Figura 2. Metodología Scrum**

La evolución del proyecto se gestiona a través de reuniones breves periódicas en las que todo el equipo revisa el trabajo realizado y lo previsto. De esta manera el grupo de trabajo se auto organiza, tomando las decisiones que se crean necesarias en cuanto al desarrollo y avance del proyecto. Al finalizar el sprint se realiza el análisis y revisión del incremento generado y se toman las decisiones oportunas para pasar a la siguiente fase de desarrollo.

Posteriormente, se lleva adelante una simulación de situaciones reales en entorno de laboratorio para probar las tecnologías seleccionadas (QR, lenguajes de programación, bibliotecas, plataformas, sistemas de gestión que lo puedan embeber, entre otras).

Después se analizan dos casos prácticos en empresas que poseen sistemas de gestión con el objetivo de relevar requerimientos y necesidades reales. Se realizan las mismas pruebas en entorno de laboratorio para obtener resultados que permitan confrontar con los resultados obtenidos anteriormente. Además, se pretende evaluar el grado de impacto que puede provocar el proyecto en los usuarios finales y en el funcionamiento global de la empresa, a los fines de obtener información que permita cuantificar la mejora en los procesos actuales y la utilidad del desarrollo.

Finalmente se obtienen como resultado las necesidades técnicas para el desarrollo del proyecto basados en los conocimientos empíricos obtenidos, teniendo en cuenta que puede sufrir modificaciones en el tiempo en base a los avances tecnológicos.

A continuación se presentan dos esquemas que describen los escenarios de funcionamiento de la API.

## ESCENARIO DE GENERACIÓN

La figura 3 muestra el escenario de generación de los códigos QR en los comprobantes físicos. Los pasos del proceso son los siguientes:

- 1° El sistema de gestión proporciona los datos de facturación.
- 2° Estos datos son captados por la API, la cual realiza las siguientes actividades sobre los mismos:
  - a. Los estructura en formato XML.

- b. Los comprime mediante un formato de compresión nativo de XML.
  - c. Los cifra utilizando el algoritmo AES-256 ya citado.
  - d. Genera los códigos QR necesarios para representar los datos en el comprobante físico.
- 3° Luego se imprime el comprobante con los códigos QR.
- 4° El comprobante puede ser transferido empleando la dinámica habitual.

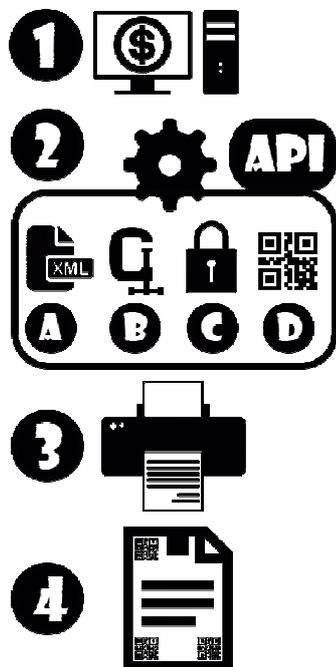


Figura 3. Escenario de generación de los códigos QR en los comprobantes físicos

### ESCENARIO DE LECTURA E INTERPRETACIÓN

La figura 4 muestra el escenario de lectura e interpretación de los códigos QR en los comprobantes físicos. Los pasos del proceso son los siguientes:

- 1° El comprobante es recibido de la manera habitual.

- 2° Mediante un lector de código QR se captura cada uno de los códigos QR impresos en el comprobante. Esto ahorra tiempo de carga y elimina los errores de tipeo.
- 3° Los datos son enviados a la API, la cual realiza las siguientes actividades sobre los mismos:
  - a. Interpreta el contenido de los códigos QR.
  - b. Lo descifra utilizando el algoritmo AES-256 ya citado.
  - c. Lo descomprime utilizando el mismo formato de compresión nativo de XML empleado en la etapa de generación.
  - d. Estructura los datos en formato XML.
- 4° Por último, los datos son enviados al sistema de gestión de manera automática.

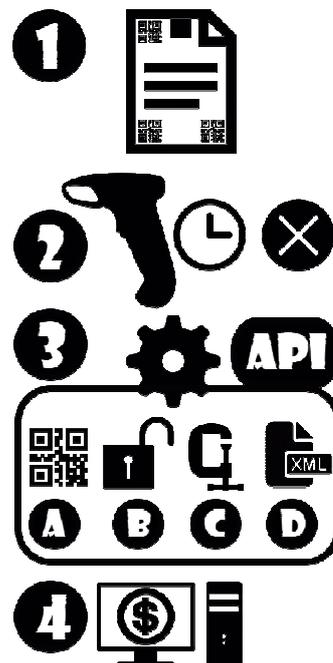


Figura 4. Escenario de lectura e interpretación de los códigos QR en los comprobantes físicos

### Resultados

Se espera desarrollar una metodología automatizada y segura para el

procesamiento de transacciones en áreas de Data Entry.

Se realiza un aporte a las teorías de Cifrado de Información y Gestión de Datos agregando a las mismas la relevancia de la Seguridad Informática.

### Discusión

Si bien se plantea un nuevo paradigma de presentación y almacenamiento de información en forma segura, esto está ligado a la utilización de documentos físicos en donde pueda ser impreso y desde donde pueda ser leído el código QR.

La tendencia actual hacia la emisión de comprobantes electrónicos significa un riesgo para el presente proyecto, debido a que el desarrollo obtenido no va a ser aplicable si los datos se transfieren en formato digital entre los diferentes sistemas.

La investigación y los desarrollos efectuados dentro del proyecto en materia de cifrado, compresión de datos y transferencia de información mediante archivos XML; son independientes de la representación mediante códigos QR. Esto permite reutilizarlos en nuevos métodos de transferencia de información en caso de que los documentos físicos dejaran de ser viables. Sin embargo, luego de analizar este impacto, se infiere que los documentos en papel como medio de transacción no serán reemplazados en forma masiva en un futuro cercano.

### Conclusión

Se plantea un nuevo paradigma de presentación y transferencia de la información mediante la implementación de estándares mundiales a través de códigos QR, proporcionando almacenamiento estructurado a través del lenguaje XML y un alto nivel de seguridad implementando compresión y cifrado de datos. Este paradigma es aplicado en primera instancia al ámbito de comprobantes financieros y luego expandible hacia nuevas áreas de interés donde pueda resultar útil.

### Referencias

1. MODELOS DE COMPROBANTES FISCALES. BIBLIOTECA AFIP [en línea]. Disponible en: [http://biblioteca.afip.gob.ar/afipres/RG\\_AF\\_IP\\_0163\\_A001\\_V000.htm](http://biblioteca.afip.gob.ar/afipres/RG_AF_IP_0163_A001_V000.htm)  
Fecha de consulta: miércoles 24 de octubre de 2012.
2. ANEXO II TIPOS DE COMPROBANTES FISCALES. BIBLIOTECA AFIP [en línea]. Disponible en: <http://www.afip.gov.ar/afip/259Iicap1a.htm>  
<http://www.afip.gov.ar/afip/259Iicap1b.htm>  
<http://www.afip.gov.ar/afip/259Iicap1c.htm>  
<http://www.afip.gov.ar/afip/259Iicap1d.htm>  
Fecha de consulta: miércoles 24 de octubre de 2012.
3. STALLINGS, William. Fundamentos de Seguridad en Redes: Aplicaciones y Estándares. 2º Edición (2004).
4. BORDESE, Matías. Análisis y Alternativas para la Compresión de XML. Trabajo Integrador (Especialista en Sistemas y Servicios Distribuidos). Córdoba, Argentina, UNIVERSIDAD NACIONAL DE CORDOBA, Facultad de Matemática, Astronomía y Física (FaMAF), Agosto, 2009.
5. JACOBSON, Daniel; BRAIL, Greg & WOODS, Dan. APIs A Strategy Guide. O'Reilly Media. (2011).
6. INTERNATIONAL STANDARD ISO/IEC 18004:2000, Information technology – Automatic identification and data capture techniques - Bar code symbology – QRCode, First Edition (2000).
7. DEITEL, Harvey M. & DEITEL, Paul J. Como Programar C#. 2º Edición (2007).
8. CEBALLOS SIERRA, Francisco Javier. Enciclopedia de Microsoft Visual C#. 2º Edición (2010).
9. DEITEL, Harvey M. & DEITEL, Paul J. Como Programar en Java. 2º Edición (2008).
10. SUEHRING, Steve; CONVERSE Tim & PARK, Joyce. La Biblia PHP6 y MySQL. 1º Edición (2009).

Tablas:

*Tabla 1. Herramientas de compresión de archivos xml*

Figuras:

*Figura 1. Estructura de un código QR*

*Figura 2. Metodología Scrum*

*Figura 3. Escenario de generación de los códigos QR en los comprobantes físicos.*

*Figura 4. Escenario de lectura e interpretación de los códigos QR en los comprobantes físicos.*