



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

GRADO INGENIERÍA INFORMÁTICA

PROYECTO FIN DE GRADO

**Gestión de servicios
IT mediante códigos QR**

Autor: Daniel Hernández Cassel
Director: Dr. Miguel Ángel Patricio Guisado

Septiembre, 2014



Universidad
Carlos III de Madrid

Agradecimientos

En primer lugar quiero agradecer a Yose y a Susana por el tiempo invertido y el apoyo que me han dado.

A Juan, con quien siempre puedo comentar temas tecnológicos gracias a sus amplios conocimientos.

A mis compañeros de trabajo, con especial mención a Miguel, por entender el potencial de la idea, ayudar con el análisis y por valorar formas de expandir su campo de acción.

A todos los amigos que no han dejado de escuchar y dar sus opiniones sobre el proyecto durante todo el proceso de realización del mismo.

Índice General

Introducción	7
1.1 Motivación y Antecedentes	7
1.1.1 Características de los Sistemas.....	10
1.2 Objetivos	13
1.2.1 Creador de QR.....	14
1.2.2 Servicios Web.....	15
1.2.3 Interfaz de integración	15
1.2.4 Conceptos de ámbito general	16
1.3 Medios Disponibles	17
1.4 Estructura de la Memoria.....	18
1.5 Definiciones, Acrónimos y Abreviaturas.....	19
Estado del Arte	24
2.1 Sistemas de gestión de servicios IT.....	24
2.2 Códigos QR.....	26
Especificación de Requisitos	31
3.1 Requisitos Funcionales.....	31
3.1.1 Requisitos del Creador de códigos QR.....	31
3.1.2 Requisitos de los Servicios Web	37
3.1.3 Requisitos de las interfaces de integración	44
3.2 Requisitos No Funcionales	47
3.2.1 Requisitos de Usabilidad	47
3.2.2 Requisitos de Disponibilidad.....	50
3.2.3 Requisitos de Seguridad.....	51
3.2.4 Requisitos de Rendimiento.....	53
3.2.5 Requisitos de Portabilidad	55

Diseño	56
4.1 Descripción General	56
4.1.1 Creador de Códigos QR.....	56
4.1.2 Servicios Web	58
4.2 Arquitectura del Sistema	60
4.2.1 Creador de Códigos QR	60
4.2.2 Servicios Web	64
4.2.3 Interfaces de Integración	67
4.3 Bases de Datos.....	68
4.3.1 Creador de códigos QR	68
4.3.2 Servicios Web	70
4.4 Casos de Uso	72
4.4.1 Casos de Uso de Creador de QR.....	72
4.4.2 Casos de Uso de Servicios web.....	76
4.5 Diagramas de Casos de Uso	78
 Marco Regulador y Entorno Socio-económico	 80
5.1 Leyes aplicables al sistema a desarrollar.....	80
5.1.1 Ley Orgánica 15/99 de Protección de Datos de Carácter Personal.....	80
5.1.2 Ley 34/2002 de servicios de la sociedad de la información y de comercio electrónico (LSSI).....	81
5.1.3 R.D.L 1/1996 Ley de Propiedad Intelectual	81
5.2 Licencias comunes	82
5.2.1 Licencia MIT	82
5.2.2 GNU Lesser General Public License	82
5.3 Acciones a realizar.....	83
5.4 Entorno socio-económico	83
 Implementación	 85
6.1 QRCreator.....	85
6.2 Servicios Web	87
6.3 Interfaces de Integración	88
 Pruebas	 89
7.1 Pruebas Funcionales.....	89
7.2 Pruebas de Seguridad.....	100

Conclusiones y Futuros Trabajos	104
8.1 Conclusiones.....	104
8.2 Trabajos Futuros	106
Índice de Figuras y Tablas.....	109
9.1 Índice	109
Bibliografía	111
10.1 Referencias.....	111
10.2 Bibliografía Consultada.....	113
10.3 Herramientas Referenciadas	116
Apéndice I Manual de Usuario QRCreator	117
11.1 Instalación y configuración Ruby	117
11.1.1 Instalación de Ruby en Linux	117
11.1.2 Instalación de Ruby en Windows.....	118
11.1.3 Gemas Necesarias.....	118
11.2 QRCreator	118
11.3 Referencias Externas	123
Apéndice II Manual de Usuario Servicios Web	124
12.1 Configuración de los Servicios Web	124
12.2 Acceso y uso de Servicios Web	124
12.3 Notificaciones y avisos.....	128

Introducción

1.1 Motivación y Antecedentes

La gestión de servicios de tecnologías de la información, ITSM, es una disciplina basada en procesos, enfocada en alinear los servicios de TI proporcionados con las necesidades de las empresas, poniendo énfasis en los beneficios que puede percibir el cliente final.

Una correcta y eficiente gestión de servicios de tecnologías de la información es indiscutiblemente uno de los pilares básicos necesarios para alcanzar el éxito en prácticamente cualquier empresa.

En líneas generales una buena gestión de servicios IT debe incluir:

- Agilizar, dinamizar, reducir riesgos, reducir costes de la gestión de servicios corporativos, mejorar la eficacia de la toma de decisiones y agilizar los procesos de recursos humanos.
- Alinear los procesos de negocio y de la estructura IT.
- Generar negocio.

Las tecnologías de la información se suelen considerar como un concepto dinámico, pues continuamente surgen nuevas tecnologías con capacidades incrementadas que desplazan a las existentes. Esto implica que debe existir un trabajo continuo de investigación y adaptación a las nuevas tecnologías por parte de los técnicos de IT.

Teniendo en cuenta estos conceptos, es necesario desarrollar un sistema de gestión, que permita agilizar las gestiones de IT a los usuarios cuando no tienen acceso a su estación de trabajo habitual.

Uno de los requisitos principales es que debemos basarnos en el uso de software corporativo existente para la gestión de servicios de IT o desarrollos que sean compatibles con el mismo. Actualmente se emplea de forma primaria el PaaS Service-Now ^[1], disponemos como punto de partida de las listas existentes de integraciones ^[2] y complementos ^[3] existentes. Necesitamos un sistema que sea fácil de emplear para los usuarios de las diferentes fases de gestión, y que permita agilizar la realización y gestión de pedidos e incidencias relacionadas con los servicios de IT.

Otro de los requisitos fundamentales es que el sistema de solicitud de servicios empleado debe estar orientado hacia o ser plenamente funcional con dispositivos móviles.

Con el fin de adaptar, generar y/o emplear un sistema que cumpla con estos requisitos, disponíamos inicialmente de varios conceptos que podían llegar a adaptarse a nuestras necesidades:

- **Acciones al recibir correo electrónico:** realizar una serie de acciones al recibir un correo enviado por un usuario/cliente, en base a los parámetros contenidos tanto en el cuerpo como en el asunto e incluso en el emisor del mensaje. Este tipo de concepto es óptimo por ejemplo para automatizar las tareas rutinarias derivadas de correos de aviso generados por servidores, bastaría con incluir en la lista de envío el correo de recepción del servicio y generar una regla de gestión de forma que si el correo recibido cumple una serie de parámetros, se genere el pedido o incidencia de gestión que corresponda, como puede ser ampliar la memoria de un servidor.
- **Aplicaciones nativas:** realizar aplicaciones móviles para las diferentes plataformas empleadas Android, IOS y BlackBerry. De forma que el usuario las tenga previamente instaladas y configuradas para poder realizar las gestiones necesarias de forma simple y dinámica.

- **Interfaces específicas para dispositivos móviles:** realizar interfaces nativas específicas para los dispositivos móviles de forma que cuando se accede al servicio desde un dispositivo móvil la interfaz además de adaptarse, permita la realización de una serie de gestiones específicas. De esta forma se podría, por ejemplo, modificar la interfaz de un usuario final si este accede al servicio con un dispositivo móvil y agregarle una opción específica para reportar el mal funcionamiento de su ordenador, por si este es el motivo por el que no puede acceder al servicio desde dicho ordenador.
- **Códigos QR [4]:** los códigos QR permiten almacenar una gran cantidad de información, bastando una aplicación de escaneo de QR para recuperar dichos datos. Los códigos QR se pueden emplear por ejemplo para almacenar los datos del ordenador del usuario de forma que si algo le sucede solo tenga que escanearlo, especificar el problema y la gestión del mismo comenzará. Cuando un código QR se escanea se accede a un servicio web correspondiente, a través del cual se gestionará el pedido con los introducidos y escaneados.

A continuación se enumeran estos conceptos junto con el conjunto de ventajas y desventajas de cada uno. Este análisis es independiente del software empleado para la gestión de procesos de IT, que viene impuesto por los requisitos, los cuales se detallaran más adelante.

1.1.1 Características de los Sistemas

Concepto	Acciones al recibir correo electrónico
Ventajas	<ul style="list-style-type: none"> - Únicamente requiere el envío de un email. - Permite automatizar ciertos servicios. - Disponible para todo tipo de dispositivos
Desventajas	<ul style="list-style-type: none"> - Requiere la creación de plantillas para la introducción correcta de parámetros. Los usuarios deben cumplimentar los datos. - No es posible el empleo de credenciales de forma segura, seguridad dependiente de la seguridad de las cuentas de correo electrónico (si se conoce la dirección de correo electrónico se puede suplantar). - Personalización de las solicitudes limitada a acciones desarrolladas.

Concepto	Aplicaciones nativas
Ventajas	<ul style="list-style-type: none"> - Solicitud personalizable - Permite al usuario generar y administrar sus diferentes tipos de solicitudes fácilmente.
Desventajas	<ul style="list-style-type: none"> - Seguridad dependiente de seguridad del dispositivo móvil. Al configurar el perfil de usuario en el dispositivo se guardan los credenciales en hash, se recuperan mediante clave. Esta clave por conveniencia del usuario suele ser de cuatro dígitos. - Requiere desarrollo de tres aplicaciones para tres lenguajes diferentes.

Concepto	Interfaces específicas para dispositivos móviles
Ventajas	<ul style="list-style-type: none"> - Solicitud personalizable. - Seguridad y credenciales proporcionadas por el servicio web. - Disponible para todo tipo de dispositivos.
Desventajas	<ul style="list-style-type: none"> - Requiere desarrollo de interfaces con módulos propios adicionales a los comunes. - Necesario acceder al servicio web y cumplimentar un pedido/incidencia.

Concepto	Códigos QR
Ventajas	<ul style="list-style-type: none"> - Únicamente requiere el escaneado de código QR. - Disponible para todo tipo de dispositivos. - Al escanear se accede a un servicio web que genera la solicitud con los datos introducidos por el usuario (incluidos credenciales) y los datos escaneados.
Desventajas	<ul style="list-style-type: none"> - Personalización de las solicitudes limitada a servicios desarrollados.

La siguiente figura presenta un cuadro comparativo de los sistemas anteriores

	Acciones al recibir correo electrónico	Aplicaciones nativas	Interfaces específicas para dispositivos móviles	Códigos QR
Personalización completa por defecto				
Multiplataforma				
Seguridad al menos como en servicio web				
Fácil de emplear por usuario final				
Integrado nativamente en Service-Now				
Integrable				

Figura 1.1.a: Análisis de las características de los sistemas

En base al análisis resumido en la figura anterior se decidió desarrollar el sistema que emplea los códigos QR. La principal deficiencia de los códigos QR, su relativamente limitado rango de usos, está supeditado al desarrollo que se realice, cuanto más completo sea este desarrollo menos evidente será esta deficiencia.

1.2 Objetivos

El objetivo final de este proyecto es el de desarrollar los servicios, las aplicaciones y las integraciones necesarias para poder realizar la gestión de servicios de IT empleando códigos QR.

Una vez que se finalice el proyecto se dispondrá de los elementos software necesarios para generar los códigos QR con los datos deseados, con capacidad para modificarlos según sea necesario, terminar de cumplimentar y gestionar las solicitudes derivadas de los escaneos de los usuarios de los diferentes códigos previamente generados mediante servicios web ubicados en un servidor externo y gestión de los servicios de IT correspondientes en la herramienta de gestión corporativa. Para lograr este último hito será necesario establecer una pasarela de datos que permita el envío de solicitudes entre los servicios web de gestión y el servicio web de integración. Este servicio de integración se tendrá que desplegar en la herramienta corporativa de gestión de IT.

Los elementos construidos deberán estar preparados para integrarse como parte de la aplicación de gestión de servicios de IT, además deberán ser multiplataforma en todas las fases del proceso, desde la generación de los códigos QR hasta la creación de los pedidos en la herramienta de gestión.

A continuación se presenta una breve descripción de cada uno de los elementos principales del proyecto, indicando las principales características que deben presentar y servicios que deben poder prestar.

1.2.1 Creador de QR

El creador de QR a desarrollar deberá ser multiplataforma, siendo interesante que adicionalmente se encapsule en un ejecutable de forma que se pueda distribuir fácilmente. Además de disponer de interfaz gráfica, deberá ser capaz de soportar trabajos batch con el objetivo de automatizar tareas y ejecutar trabajos por lotes.

Deberá ser capaz de generar códigos QR adaptados a los diferentes tipos de servicios web ofrecidos, mediante el uso de plantillas o formularios específicos, y además tener opción a generar códigos QR que no empleen dichas plantillas. Para casos puntuales en los que urja crear un código QR cuya plantilla aun no esté creada o para los casos en los que no se requiera el desarrollo de una nueva plantilla pero si la creación de un QR con datos específicos.

Tendrá que tener la capacidad de almacenar los QR creados de forma que al menos cuando se emplee la interfaz gráfica, se puedan recuperar los códigos previamente creados para su reutilización o modificación y de esta forma, evitar tener que estar continuamente introduciendo todos los datos contenidos en los QR. A la hora de cargar QR previos deberá ser capaz de identificar los QR creados para cada una de las diferentes plantillas/formularios, de forma que se muestren los códigos correctos en cada caso. A la hora de clasificarlos, se considerará que todos los códigos creados sin plantilla pertenecen a un mismo grupo.

Es posible que sea necesario generar versiones limitadas o específicas del creador de códigos QR que serán distribuidas a grupos de diferentes niveles de acceso. Puesto que la realización de estas versiones implicaría principalmente eliminar o restringir el acceso de ciertas partes del programa, si este se encuentra bien estructurado no debería suponer un gran incremento del volumen de trabajo a realizar.

1.2.2 Servicios Web

Los servicios web deberán ser capaces de soportar múltiples pedidos simultáneos sin colapsarse, deberán disponer de una interfaz simple y ser de fácil uso por parte del usuario final, compatibles con todos los navegadores comerciales, deberán disponer de interfaces adaptables en función del tipo y tamaño del dispositivo con el que se acceda a ellos.

Estos servicios web deberán registrar los datos de las solicitudes entrantes y los resultados devueltos por la herramienta. Estos datos serán almacenados en el servidor en el que se alojen los servicios web. Aunque se presupone seguro, se deberán almacenar cifrados para proporcionar una seguridad adicional a los datos.

Desde el punto de vista de su despliegue, deberán ser multiplataforma, pudiendo desplegarse al menos en servidores que empleen diferentes distribuciones de Linux o versiones de Windows.

1.2.3 Interfaz de integración

La interfaz del servicio web de integración ha de adaptarse a los módulos nativos de la herramienta Service-Now de forma que pueda acceder a las diferentes estructuras de gestión contenidas en dicha herramienta y realizar las gestiones solicitadas.

Será necesaria la creación de interfaces de integración independientes para cada una de los tipos de solicitud que se requiera integrar en la herramienta de gestión Service-now. Esto se debe a que la creación de interfaces que soporten solicitudes genéricas resulta aparatosa al tener cada tipo de solicitud diferentes datos de entrada y afectar a módulos con diferentes características, formularios y business rules. Las características comunes como la salida de datos o las gestiones que se puedan considerar como transacciones atómicas, podrán ser compartidas sin necesidad de ser repetidas en cada una de las interfaces.

No será necesario realizar una gestión de almacenamiento en una BB.DD. específica de las solicitudes que se gestionen mediante estas interfaces. Tanto los datos de las solicitudes entrantes como las solicitudes que sean correctamente gestionadas deberán reflejarse en los sistemas de log internos de la herramienta. Implícitamente, las solicitudes que sean correctamente gestionadas serán registradas en los módulos correspondientes del sistema de gestión.

1.2.4 Conceptos de ámbito general

Las BB.DD que se empleen tanto en el creador de QR como en los servicios web deberán ser capaces de soportar un gran número de registros y trabajar velozmente a la vez que mantiene el consumo de recursos al mínimo.

Los diferentes componentes de este sistema deberán contar con modos de debug disponibles para la realización de pruebas, corrección de errores o simplemente para obtener trazabilidad completa del trabajo realizado.

Para lograr realizar la gestión de los servicios de IT mediante el empleo de códigos QR es necesario el desarrollo de las distintas partes diferenciadas que componen el sistema descrito previamente. Estas partes diferenciadas serán independientes unas de las otras pero deberán ser capaces de realizar la gestión de forma fluida.

La siguiente figura ilustra la interacción entre las diferentes partes del sistema

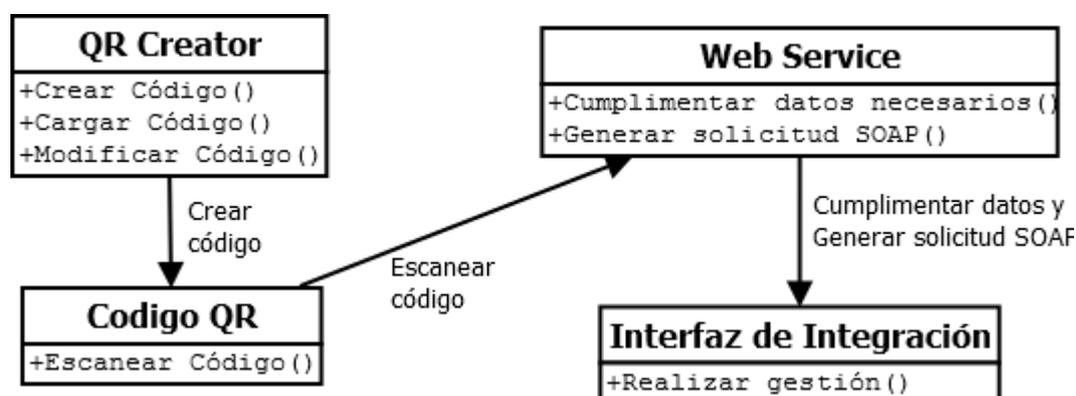


Figura 1.2.a: Interacción entre partes diferenciadas del sistema

1.3 Medios Disponibles

Para el desarrollo de este proyecto no existen restricciones específicas respecto a los lenguajes de programación a emplear, dejándose de parte del desarrollador la elección del lenguaje o lenguajes que considere más convenientes de acuerdo con los objetivos y los requisitos del proyecto. Como se detallará más adelante sería interesante emplear un lenguaje que presente un alto factor de portabilidad, como puede ser Java, Perl, Python, Ruby [5], etc.

Para el desarrollo y prueba de la interfaz de integración con la herramienta Service-now disponemos de permisos completos de administración en cuatro entornos clonados del entorno de producción, habilitados para desarrollo y consolidación, que se podrán emplear para la realización del desarrollo y las pruebas pertinentes.

Es importante considerar que sería óptimo que las partes del proyecto correspondientes a la creación de QR y a la gestión web de los servicios no dependan de herramientas privativas o sujetas a licencias de uso.

1.4 Estructura de la Memoria

Este documento está dividido principalmente en once partes claramente diferenciadas por su contenido:

- Una primera parte de introducción al proyecto desarrollado. En este punto entramos a valorar la justificación del proyecto y se establecen los objetivos del mismo.
- Una segunda parte donde se detalla el estado del arte actual de las herramientas y tecnologías que se emplearán en el desarrollo del proyecto.
- En la tercera parte se detalla cada uno de los requerimientos que finalmente deberá cubrir la implementación final del proyecto.
- En la cuarta parte se recoge el diseño que finalmente se ha seguido para el sistema y el análisis de la tecnología empleada.
- En la quinta parte se realiza una descripción del marco regulador y del entorno socio-económico.
- El sexto apartado describe detalles de la implementación que han sido particularmente interesantes o problemáticos, así como características concretas de los diferentes componentes del sistema.
- La séptima parte contiene las pruebas realizadas sobre los componentes del sistema.
- En la octava parte se analizan las conclusiones obtenidas y los futuros trabajos que se pueden realizar tomando como base el proyecto.
- Las partes novena y décima contienen el índice de figuras y la bibliografía empleada.
- Como colofón a este documento, en los apartados undécimo y duodécimo se incluyen manuales de usuario para el uso de los sistemas desarrollados.

1.5 Definiciones, Acrónimos y Abreviaturas

- **ACID**: En inglés Atomicity, Consistency, Isolation and Durability, en bases de datos se denomina ACID a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción. Si un sistema de gestión de bases de datos es compatible con ACID quiere decir que el mismo cuenta con las funcionalidades necesarias para que sus transacciones tengan las características ACID.
- **ACL**: Access Control List, listas de usuarios con permiso de acceso a diferentes módulos, secciones, objetos, etc.
- **APP**: En inglés Application, una aplicación es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.
- **Batch**: Se conoce como batch al modo de ejecución no interactivo o no asistido de un programa.
- **BBDD**: Bases de Datos.
- **Bitcoin**: Divisa electrónica descentralizada que fue creada en 2009.
- **Business Rules**: Las Reglas del Negocio describen las políticas, normas, operaciones, definiciones y restricciones presentes en una organización o producto.
- **CI**: En inglés Configuration Item, cada elemento de configuración almacenado en la CMDB, puede ser hardware, software o relaciones entre CIs.
- **Cloud computing**: La computación en la nube es un paradigma que permite ofrecer servicios de computación a través de Internet.
- **Cobit**: En inglés Control Objectives for Information and related Technology, es una guía de mejores prácticas presentado como framework, dirigida al control y supervisión de tecnología de la información (TI).
- **Código QR**: Quick Response code en inglés, son un tipo de código de barras bidimensional que permite almacenar una gran cantidad de información.
- **CMDB**: En inglés Configuration Management DataBase, es la base de datos de la gestión de configuración, que contiene una entrada por cada CI con sus detalles y relaciones.

- **CPD:** Centro de procesamiento de datos, es una ubicación donde se concentran los recursos necesarios para el procesamiento de la información de una organización
- **CSS:** En inglés Cascading Style Sheets, es un lenguaje empleado para describir la apariencia y el formato de un documento escrito en lenguaje de marcas.
- **EDPR:** Siglas correspondientes a Energías De Portugal Renovables, empresa del sector energético centrada en las energías renovables. Pertenece al grupo EDP.
- **Gema:** En contexto del lenguaje de programación Ruby se denominan gemas los paquetes de programas o librerías autocontenidos disponibles para su carga e instalación que aportan nuevas funcionalidades.
- **GPL:** En inglés General Public License, licencia de software libre, que especifica que a todo trabajo derivado deberá aplicársele la misma licencia.
- **GUI:** En inglés Graphical User Interface, la interfaz gráfica de usuario es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.
- **HH.RR:** En inglés Human Resources, por lo general hace referencia a sistema o proceso de gestión que se ocupa de seleccionar, contratar, formar, emplear y retener al personal de la organización.
- **HTML:** HyperText Markup Language, es el lenguaje de marcado para la elaboración de páginas web.
- **IDE:** Un entorno de desarrollo integrado, en inglés de Integrated Development Environment, es un programa informático compuesto por un conjunto de herramientas de programación. Usualmente se considera que un IDE consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.
- **IRB:** Interactive Ruby Shell, entorno interactivo para el desarrollo de Ruby.
- **IT:** En inglés Information technology, agrupan los elementos y las técnicas utilizadas en el tratamiento y la transmisión de las informaciones, principalmente de informática

- **ITIL:** En inglés Information Technology Infrastructure Library, la Biblioteca de Infraestructura de Tecnologías de Información, es un conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas.
- **ITSM:** En inglés IT Service Management, disciplina basada en procesos, enfocada en alinear los servicios de TI proporcionados con las necesidades de las empresas.
- **Java:** Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases.
- **JS:** JavaScript, es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo y dinámico.
- **Kana:** Término que describe a los alfabetos silábicos japoneses
- **Kanji:** Son los sinogramas utilizados en la escritura de la lengua japonesa.
- **MySQL:** Se trata del lenguaje estándar empleando para elaborar las sentencias (Query) de acceso a las BBDD.
- **OS:** En inglés Operating System, programa o conjunto de programas que entre otras cosas, gestiona los recursos hardware y provee servicios a los programas de aplicación.
- **PaaS:** En inglés platform as a service, es una categoría de servicios de cloud computing que ofrece una plataforma de computación y un conjunto de soluciones como un servicio.
- **Parser:** Programa encargado de transformar un documento o archivo en un determinado origen a otra en un nuevo formato destino (XML→HTML, Word→PDF, etc.).
- **PHP:** Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
- **PNG:** Portable Network Graphics, es un formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps no sujeto a patentes.
- **POLA:** En inglés Principle Of Least Astonishment, indica que la característica o elemento descrito se debe de comportar de forma que minimice los escenarios en los que puede confundir a los usuarios expertos.

- POO: Programación Orientada a Objetos, es un paradigma de programación que emplea los objetos en sus interacciones para diseñar aplicaciones y programas informáticos.
- Prepared Statement: En los sistemas de BBDD esta característica permite ejecutar una misma sentencia o sentencias con pocas diferencias de forma repetida con alta eficiencia. Este tipo de sentencias son resistentes contra SQL Injection.
- PRG: Post/Redirect/Get, es un patrón de diseño de desarrollo web que impide algunos envíos de formularios duplicados.
- Query: Sentencia que se ejecuta sobre una BBDD para obtener información de la misma.
- QR Code: Ver Código QR
- Ruby: Lenguaje de programación interpretado, reflexivo y orientado a objetos. Alto grado de portabilidad.
- Ruby on Rails: Es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador.
- Sinograma: Unidad mínima de un sistema de escritura, que por sí solo representa una palabra o un morfema originario de China.
- SLA: En inglés Service Level Agreement, es la parte de un contrato de servicios en la que se definen los tiempos de gestión y la calidad del servicio.
- SOAP: En inglés Simple Object Access Protocol, es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML
- SQL Injection: Es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar consultas a una base de datos.
- SQLite: Es una implementación de sistema de gestión de bases de datos relacional compatible con ACID, contenida en una biblioteca relativamente pequeña escrita en C. SQLite es un proyecto de dominio público.
- Software: Son los programas y datos almacenados en los ordenadores y periféricos.

- Sys_id: Cada registro en ServiceNow es identificado por un único GUID de 32 caracteres (Globally Unique ID) llamado sys_id. El mismo valor sys_id nunca se repetirá, asegurando que cada registro creado en cada tabla en cada instancia de ServiceNow en el mundo tiene un valor único de identificación.
- TK: En contexto del lenguaje de programación Ruby, TK es una de las principales GUI disponibles.
- Transacción atómica: Conjunto de acciones, órdenes que se realizan de forma independiente e indivisible.
- URL: En inglés uniform resource locator, es una cadena de caracteres que constituye una referencia a un recurso.
- W3C: El World Wide Web Consortium, es un consorcio internacional que produce recomendaciones para la World Wide Web.
- WSDL: son las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web.
- XML: En inglés eXtensible Markup Language, es un lenguaje de marcas desarrollado por el W3C utilizado para almacenar datos en forma legible.

Estado del Arte

En esta sección se analiza y detalla el estado del arte de las principales tecnologías y herramientas que se emplearán en la realización de este proyecto.

2.1 Sistemas de gestión de servicios IT

La gestión de servicios de tecnologías de la información (ITSM) es una disciplina basada en procesos, enfocada en alinear los servicios de TI proporcionados con las necesidades de las empresas, poniendo énfasis en los beneficios que puede percibir el cliente final. GSTI propone cambiar el paradigma de gestión de TI, por una colección de componentes enfocados en servicios de punta a cabo usando distintos marcos de trabajo con las "mejores prácticas", como por ITIL o COBIT.

Existen un gran número de herramientas que ofrecen este tipo de servicio, por lo que se procede a analizar únicamente los resultados más relevantes. Para realizar la selección de estas herramientas nos hemos basado en listas y tablas como el "Magic Quadrant for IT Service Support Management Tools" desarrollado cada año por Gartner [6].



Figura 2.1.a: Gartner Magic Quadrant for IT Service Support Management Tools

2014

La siguiente figura muestra los resultados de la investigación de herramientas llevada a cabo. Se puede apreciar que aunque las implementaciones difieren, los servicios ofrecidos por las herramientas líderes de mercado son relativamente similares.

	Service-Now (Berlin Release)	BMC Software (BMC Remedy ITSM 8.1)	CA Technologies (Service Desk Manager 12.7)	IBM (SmartCloud Control Desk 7.5)	HP (Service Manager 9.31)	Cherwell Software (CSM 4.3)	Axios Systems (assyst v.10)	EasyVista (EasyVista 2012)	FrontRange (Heat 2012.3)
Incident Management									
Problem Management									
Change Management									
Service Level Agreement Management									
Finalcial Management									
User Self Service									
Knowledge Base Management									
Service Catalog Management									
Online Wiki Documentation									
Reporting / Metrics Management									
Outstanding Social IT Management									
Outstanding Configuration Management									
Mobile Devices Interfaces support									
Licensed in a SaaS model									
Integration of QR code actions									

Figura 2.1.b: Análisis de la comparación de herramientas

Se han agregado referencias a las herramientas mencionadas en la sección 10.2 Bibliografía Consultada.

2.2 Códigos QR

Este tipo de códigos de barras bidimensionales fue creado por la compañía japonesa Denso Wave en 1994. Se caracteriza por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector. El objetivo principal cuando se creó este tipo de códigos era que permitiera que su contenido se leyera a alta velocidad, de ahí su nombre. Los códigos QR son muy comunes en Japón y de hecho son el código bidimensional más popular en ese país. Un detalle importante sobre el código QR es que su código es abierto y sus derechos de patente (propiedad de Denso Wave) no son ejercidos.

El estándar japonés para códigos QR (JIS X 0510) fue publicado en enero de 1998 y su correspondiente estándar internacional ISO (ISO/IEC18004) fue aprobado en junio de 2000.

En la siguiente figura se pueden apreciar las diferentes áreas que componen un código QR.

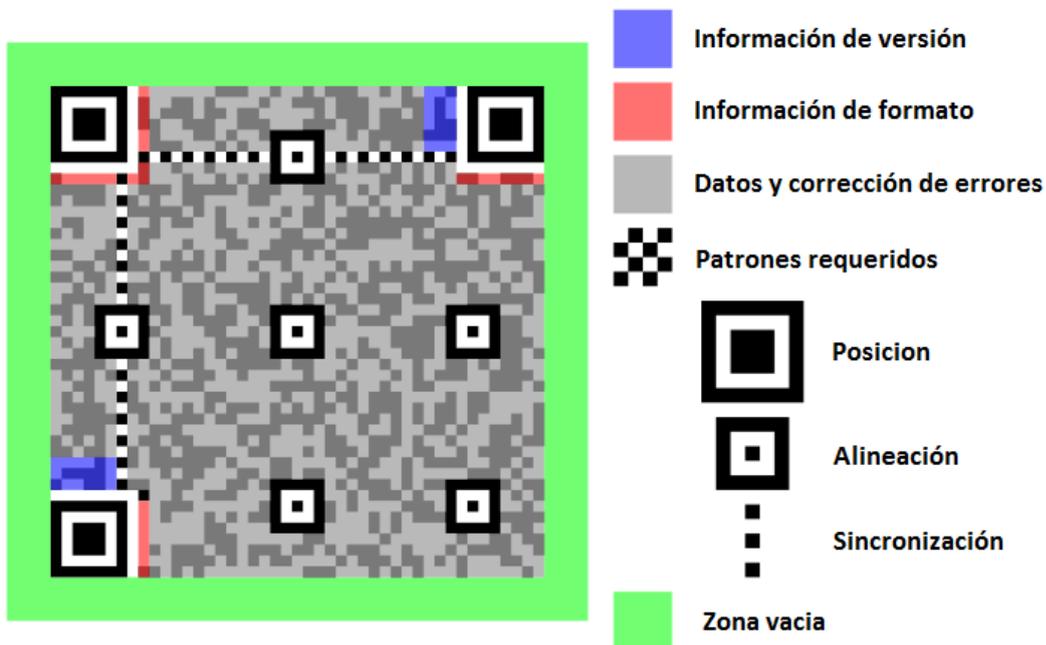


Figura 2.2.a: Estructura de código QR

La cantidad de datos que pueden almacenarse en un código QR depende del tipo de datos (numérico, alfanumérico, binario, etc.), la versión (de 1 a 40, indica las dimensiones del código QR), y el nivel de corrección de errores. La capacidad de almacenamiento es máxima cuando se emplea 40-L (versión 40, error de nivel de corrección L), que alcanza las siguientes capacidades de almacenaje en función del tipo de datos:

Tipo de dato	Número máximo de caracteres	Posibles caracteres
Numérico	7089	0,1,2,3,4,5,6,7,8,9,0
Alfanumérico	4296	0-9, A-Z, espacio, \$, %, *, +, -, ., /, :
Binario	2953	ISO 8859-1
Kanji/kana	1817	Shift JIS X 0208

Figura 2.2.b: Capacidad de QR según tipo de dato

Las palabras de código empleadas son de 8 bits y utilizan el algoritmo de corrección de errores Reed-Solomon con cuatro niveles de corrección de errores. Cuanto mayor sea el nivel de corrección de error, menor es la capacidad de almacenamiento. A continuación se muestra la capacidad de corrección de error aproximada para cada uno de los cuatro niveles (Las siglas de los niveles proceden del equivalente en inglés).

- Nivel L (baja) el 7% de las palabras de código se puede restaurar.
- Nivel M (Medio) 15% de las palabras de código se puede restaurar.
- Nivel Q (calidad) 25% de las palabras de código se puede restaurar.
- Nivel H (alto) 30% de las palabras de código puede ser restaurado.

La información de formato almacena dos cosas: el nivel de corrección de errores y el patrón de máscara utilizada para el código QR. El enmascaramiento se utiliza para romper los patrones en el área de datos que podrían confundir a un escáner, como las grandes áreas en blanco o las características engañosas que se parecen a las marcas del localizador. Los patrones de máscara se definen en una cuadrícula que se repite tanto como sea necesario para cubrir la totalidad del código QR. La información de formato está protegida de errores con un código BCH, y dos copias completas se incluyen en cada símbolo QR.

La siguiente figura describe con detalle el tipo y la ubicación de la información del formato almacenada en los códigos QR.

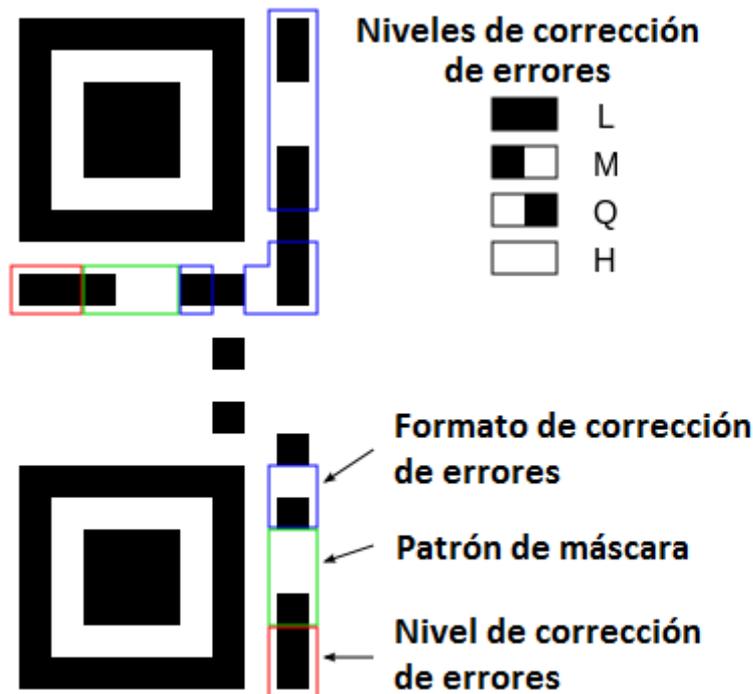


Figura 2.2.c: Información de formato

En grandes códigos QR, el mensaje se divide en varios bloques de código Reed-Solomon. El tamaño del bloque se elige de modo que a lo sumo 15 errores pueden ser corregidos en cada bloque; esto limita la complejidad del algoritmo de decodificación. Los bloques de código se entrelazan de forma que sea menos probable que los daños localizados a un símbolo QR puedan abrumar la capacidad de un bloque.

Gracias a la corrección de errores, es posible crear códigos QR artísticos que aún escanean correctamente, pero que contienen errores intencionales, colores, logotipos y otras características para hacerlos más atractivos para el ojo humano.

La siguiente figura ejemplifica la ordenación de las palabras código en el código QR.

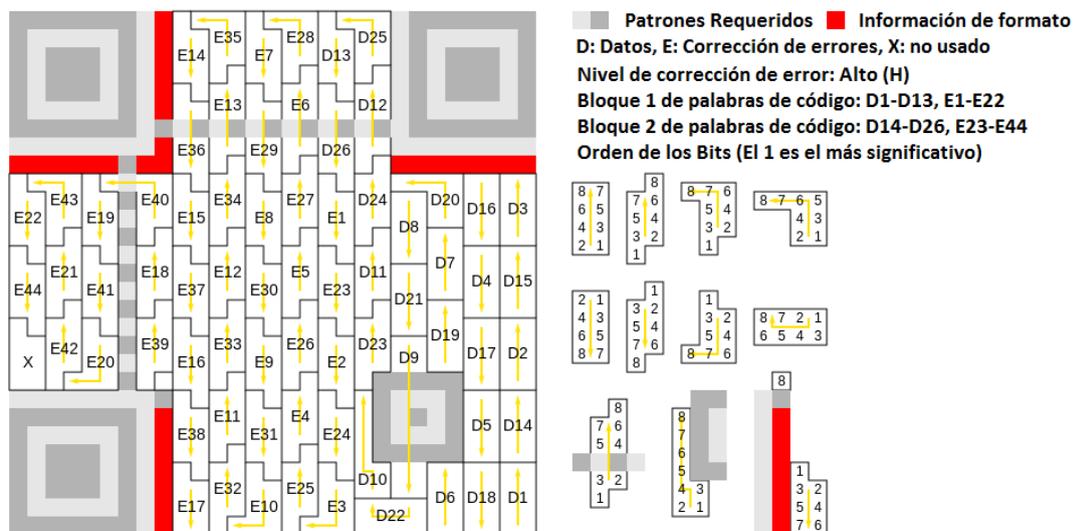


Figura 2.2.d: Ordenación de las palabras código

Los usos que se pueden dar a los códigos QR están prácticamente solo limitados por el ingenio de los técnicos, algunos de los usos actuales son:

- Cotejar datos de recetas médicas con los códigos RSS de las medicinas.
- Agilizar las consultas médicas enviando un código QR en el formulario de consulta, el cual es escaneado en la consulta médica para obtener los datos de las pruebas necesarias [7].
- Consultar los datos de un producto, el stock actual, la fecha esperada de llegada del siguiente pedido, etc.
- Consultar albaranes de envío leyendo el código adjunto en el mismo.
- Consultar productos a enviar, adquirir, almacenar o empaquetar leyendo el código adjunto en el pedido.
- Controlar el acceso mediante el escaneado de códigos QR impresos en las tarjetas personales de los usuarios.
- Generar códigos que se pueden emplear para el pago de monedas digitales, usados comúnmente en el empleo de Bitcoin.
- Permitir el acceso a URLs
- Leer información, desde direcciones físicas a partidas de ajedrez.

Existen múltiples aplicaciones de lectura de códigos QR para prácticamente todo tipo de dispositivo, desde lectores de códigos QR para PC hasta lectores de códigos QR para consolas portátiles, pasando por teléfonos móviles, tablets, etc.

Por este motivo consideramos que no es necesario desarrollar una aplicación específica para cada posible plataforma de dispositivo, ya que se pueden emplear las aplicaciones existentes.

Por ejemplo, para dispositivos Android se han realizado múltiples pruebas con el app QR Droid [8], sin embargo cualquier otro app que realice escaneado de QR podría realizar la funcionalidad deseada sin mayor problema.

Se adjunta una lista de aplicaciones probadas para los diferentes sistemas operativos de dispositivos móviles que empleamos en la compañía.

OS	Aplicación
Android	QR BARCODE SCANNER
Android	QR Droid
IOS	QR Code Reader and Scanner
BlackBerry	QR Code Scanner Pro – Free

Figura 2.2.e: Aplicaciones lectoras de QR

Especificación de Requisitos

3.1 Requisitos Funcionales

A continuación se enumeran y detallan los diferentes requisitos funcionales que el creador de QR, los servicios web y las interfaces de integración deberán cumplir.

3.1.1 Requisitos del Creador de códigos QR

Código:	RFO01	Prioridad:	E-Esencial
Título:	Tipificación de códigos QR		
Descripción			
<p>Los usuarios del creador de códigos QR deberán poder seleccionar las plantillas de tipo de código QR con la que desean operar en la aplicación de creación de códigos QR.</p> <p>Poder seleccionar el tipo de código permitirá a los usuarios no tener que introducir todos los campos de forma manual sobre la plantilla de tipo vacío, con lo que se agilizará considerablemente la realización de todas las acciones soportadas por la herramienta para la creación y manipulación de códigos QR.</p> <p>Cada una de las tipificaciones predefinidas presentará a su vez una serie de campos predefinidos de acuerdo con los valores que serán necesarios para la ejecución del servicio web y posterior integración con la herramienta de gestión.</p> <p>Cada tipificación tendrá además una url asociada que deberá apuntar al servicio web, el cual proporcionará el deseado servicio a los códigos QR creados con dicha tipificación.</p> <p>Adicionalmente a los tipos definidos se agregará por defecto siempre un tipo vacío que permitirá la creación de códigos QR no predefinidos. Esta tipificación no presentará una url de servicio web asociada por defecto.</p>			

Los tipos que se han de desarrollar se pueden apreciar en la siguiente tabla

Tipificación	Descripción	Tipo	Número de parámetros adicionales	Parámetros Obligatorios (*) y Opcionales
Default	Tipo vacío	0	1	-Datos *
Incident	Empleado para la creación de incidencias	1	7	-Categoría * -Subcategoría * - Tipificación * -Grupo asignado * -Usuario afectado -Datos adicionales -Modo
Request	Empleado para la creación de peticiones	2	3	-Artículo * -Datos adicionales -Modo
Create CI	Empleado para crear entradas en la CMDB	3	13	-Tipo de CI * -Dueño * -Usuario asignado -Datos propios del CI (varían según el tipo de CI) * -Grupo de actualización * -Grupo de aprobación * -Datos adicionales -Modo

Figura 3.1.a: Tipificaciones de códigos QR a implementar

A continuación se describen los datos que deberá contener cada uno de los parámetros adicionales indicados en la tabla anterior

- Categoría, Subcategoría, Tipificación: Datos característicos de las incidencias, mediante los cuales es posible clasificarlas.
- Grupo asignado: Grupo al que será asignado en primera instancia la incidencia una vez que sea creada.
- Usuario afectado: Usuario para el que se solicita la incidencia, es el usuario que actualmente padece el problema descrito.

- **Artículo:** Tipo de pedido que se quiere crear. La lista que se facilita está limitada a artículos que emplean el formulario básico (solo contiene los campos usuario solicitante, usuario para el que se solicita y detalles del pedido).
- **Tipo de CI:** Tipo de elemento que se quiere crear. La lista está actualmente acotada a “computer” y “server”.
- **Dueño:** Usuario propietario o responsable en última instancia del CI.
- **Usuario asignado:** Usuario que emplea habitualmente el CI, responsable en primera instancia del mismo.
- **Grupo de actualización:** grupo con permisos de modificación del CI, se asignará el grupo por defecto si no se indica otro.
- **Grupo de aprobación:** Grupo encargado de aprobar las solicitudes de cambio de datos sensibles para este CI, se asignará el grupo por defecto si no se indica otro.
- **Datos propios del CI, pueden ser obligatorios u opcionales:**
 - Proveedor (Computer), obligatorio
 - Modelo (Computer), obligatorio
 - Fecha de Adquisición (Ambos), obligatorio
 - Nombre del servidor (Application Server), obligatorio
 - OS (Application Server), obligatorio
 - RAM (Ambos), opcional
 - Capacidad de Disco (Ambos), opcional
 - Número de procesadores (Ambos), opcional
- **Datos adicionales:** Campo opcional que permite agregar una descripción, una guía o información útil en general.
- **Modo:** Modifica la cantidad tanto en los formularios como en los datos mostrados en la ejecución y en el resultado de la misma. Existe un único modo seleccionable (debug), por defecto si no se indica nada se aplica el modo estándar.

Las referencias a usuarios emplean el userID correspondiente al usuario.

Las referencias a elementos de Service-Now emplearán los sys_id de dichos elemento.

Código:	RF002	Prioridad:	E-Esencial
Título:	Creación de códigos QR		
Descripción			
<p>Tras seleccionar una tipificación de código QR deberá ser posible acceder al formulario de creación de códigos QR de dicha tipificación.</p> <p>Este formulario deberá incluir las variables predefinidas para el tipo seleccionado (en caso de tipo vacío se presentará un único campo de texto en el que se deberán introducir todos los datos de forma secuencial).</p> <p>Una vez cumplimentados los diferentes campos del formulario deberá ser posible crear el código QR. Realizar esta acción implicará la creación del código QR propiamente dicho como archivo PNG en el directorio predefinido.</p>			

Código:	RF003	Prioridad:	E-Esencial
Título:	Guardado de códigos QR creados		
Descripción			
<p>Al realizar la creación de un código QR se deberá crear una nueva entrada en la BB.DD. con los datos introducidos en los parámetros de la tipificación. El tipo de código QR y la fecha de creación también serán guardados y se emplearán para clasificar e identificar los códigos QR.</p>			

Código:	RF004	Prioridad:	E-Esencial
Título:	Carga códigos QR previamente creados		
Descripción			
<p>La aplicación de creación de códigos QR deberá permitir listar los códigos QR previamente guardados para cada tipo de código de forma independiente.</p> <p>En función del tipo de código que se encuentre seleccionado se cargarán los datos en el formulario correspondiente.</p> <p>Se deberán listar los parámetros de los diferentes códigos de forma que el usuario tenga conocimiento de las características de los diferentes códigos QR sin necesidad de ir probándolos de forma secuencial.</p>			

Código:	RF005	Prioridad:	E-Esencial
Título:	Modificación de códigos QR		
Descripción			
<p>Deberá ser posible modificar los datos introducidos en el formulario de creación de QR. Si los datos han sido cargados desde un código QR previamente existente en la BB.DD. y se desea guardar o crear el código QR modificado, se deberá poder elegir entre realizar dos acciones diferentes:</p> <ul style="list-style-type: none"> • Actualizar la entrada existente en la BB.DD. con las modificaciones realizadas. Esta acción no se considera como creación por lo que si se desea disponer del código QR modificado en un archivo deberá emplearse a continuación la acción correspondiente. La realización de esta acción no será reversible. • Crear una nueva entrada independiente con los datos modificados. Esta acción además de crear una nueva entrada en la BB.DD. deberá crear el código QR correspondiente en el directorio predefinido. 			

Código:	RF006	Prioridad:	E-Esencial
Título:	Selección de códigos QR previamente creados		
Descripción			
<p>Deberá ser posible seleccionar los códigos previamente creados y acceder al formulario de creación del tipo correspondiente al código seleccionado. Los datos de dicho código deberán ser cargados en los parámetros correspondientes del formulario.</p>			

Código:	RF007	Prioridad:	E-Esencial
Título:	Borrado de códigos QR		
Descripción			
<p>Deberá ser posible eliminar códigos QR existentes. Esta acción deberá ser accesible tanto en el listado de selección de códigos QR como en el formulario de modificación. La realización de esta acción no será reversible.</p>			

Código:	RF008	Prioridad:	E-Esencial
Título:	Creación de códigos QR en batch		
Descripción			
<p>Deberá ser posible crear códigos QR de forma desatendida. Esta funcionalidad deberá recibir para cada código que se desee crear el tipo de código a crear con sus parámetros necesarios.</p> <p>En caso de no ser un tipo válido, no facilitar los datos de los parámetros necesarios o emplear funcionalidades incorrectas se devolverá un mensaje de error y el código no será creado.</p>			

3.1.2 Requisitos de los Servicios Web

Código:	RF009	Prioridad:	E-Esencial
Título:	Tipificación de servicios		
Descripción			
<p>Se deberá crear un servicio web por cada una de las url asociadas a las diferentes tipificaciones de los códigos QR. Estos servicios deberán estar ubicados de acuerdo a lo predefinido al indicar las url correspondientes.</p> <p>Estos servicios web deberán recibir los parámetros escaneados y presentarlos en un formulario en campos diferenciados, junto con los campos adicionales que el usuario debe cumplimentar para que se pueda llevar a cabo el servicio requerido.</p> <p>Existen dos modos de muestra de datos. En función del parámetro modo se mostrarán la totalidad de los datos o solo los definidos como necesarios. En caso de modo debug, se mostrarán todos los parámetros, lo que permitirá a los usuarios que han escaneado los códigos confirmar la corrección de los datos antes de finalizar el proceso de solicitud de servicio. Si el modo no es debug se mostrarán solo los datos definidos para los usuarios estándar.</p> <p>Los datos contenidos en los códigos QR se cargarán de forma íntegra, de esta forma se permitirá la corrección y modificación de los mismos en los casos en los que esto sea necesario. Algunos de estos datos no que no eran de carácter obligatorio, pasan a ser obligatorios al requerirse para la ejecución del servicio deseado. En la mayoría de los casos esto es debido a que el valor de estos parámetros se desconoce hasta que se requiere el servicio.</p> <p>En la siguiente tabla se describen los parámetros que se deberán mostrar en los formularios de los diferentes servicios web. En caso de emplear el modo debug además de afectar a los datos mostrados al finalizar el servicio requerido, se habilitarán los parámetros de tipificación de incidencia, grupos asignados y el parámetro compañía, que por defecto presenta valor “EDPR”.</p>			

Especificación de Requisitos

Tipificación	Parámetros modo estándar	Parámetros adicionales modo debug
Incident	-Usuario afectado * -Datos adicionales -Descripción de la incidencia	-Categoría * -Subcategoría * -Tipificación * -Grupo asignado * -Compañía *
Request	-Artículo * -Detalles del pedido * -Pedido para * -Datos adicionales	-Compañía *
CMDB	-Tipo de CI * -Dueño * -Usuario asignado * -Datos propios del CI (varían según el tipo de CI) * -Datos adicionales	-Grupo de actualización * -Grupo de aprobación * -Compañía *

Figura 3.1.b: Tipificaciones de servicios web a implementar

Código:	RF010	Prioridad:	E-Esencial
Título:	Acceder a servicio		
Descripción			
<p>Deberá ser posible acceder a los diferentes servicio web escaneando los códigos QR generados.</p> <p>El servicio accedido se corresponderá con el servicio escaneado por el usuario.</p> <p>Al acceder al servicio se cargarán en el formulario los datos leídos del código QR en los campos predefinidos para cada variable. Estos datos serán adicionalmente almacenados en la tabla de la BB.DD. destinada para las solicitudes entrantes del tipo de servicio web correspondiente.</p>			

Código:	RF011	Prioridad:	E-Esencial
Título:	Solicitar servicio		
Descripción			
<p>Deberá ser posible solicitar la realización de un servicio una vez los parámetros obligatorios hayan sido cumplimentados y se seleccione la acción correspondiente.</p> <p>El servicio ofrecido variará de acuerdo al servicio web que se esté empleando, creación de una incidencia, creación de un pedido o creación de una entrada en la CMDB.</p>			

Código:	RF012	Prioridad:	D-Deseable
Título:	Registro de solicitudes		
Descripción			
<p>Cuando se solicite uno de los servicios web tipificados deberá crearse una nueva entrada en la BB.DD. del servidor en la que se almacenarán los parámetros de la solicitud incluyendo la fecha y hora de la misma.</p> <p>Para cada servicio web existirá una tabla independiente de registro de solicitudes.</p>			

Código:	RF013	Prioridad:	D-Deseable
Título:	Registro de respuestas de servicio		
Descripción			
<p>Cuando se ejecute la acción de uno de los servicios web tipificados deberá crearse una nueva entrada en la BB.DD. del servidor en la que se almacenarán los parámetros de la respuesta y la hora de ejecución del servicio.</p> <p>Para cada servicio web existirá una tabla independiente de registro de respuestas de servicios.</p> <p>Estas tablas compartirán la BB.DD. de las tablas de registro de solicitudes pero emplearán tablas independientes.</p>			

Código:	RF014	Prioridad:	E-Esencial
Título:	Gestión de datos para realizar solicitud		
Descripción			
<p>Antes de realizar la solicitud a la interfaz de integración y una vez el usuario ha ejecutado la acción correspondiente, es necesario convertir parte de los datos contenidos en los parámetros a sus valores de identificadores del sistema de la herramienta Service-Now (SysID).</p> <p>Los datos a modificar son aquellos que hacen referencia a estructuras únicas de la herramienta, los diferentes grupos, las categorizaciones y tipificaciones de incidencias y los artículos.</p> <p>Para lograr esta conversión se emplearán tablas de equivalencias de una BB.DD. que deberán ser periódicamente actualizadas con los datos de Service-Now.</p>			

Código:	RF015	Prioridad:	E-Esencial
Título:	Resumir el resultado de la ejecución del servicio		
Descripción			
<p>En función del modo de ejecución empleado variará la cantidad de datos mostrados al realizar la solicitud de servicio.</p> <ul style="list-style-type: none"> • Si el parámetro modo no está definido, se mostrarán unos datos de retorno de solicitud de servicio estándar para cualquier servicio solicitado. Los datos estándar son mensaje y código, contendrán la siguiente información. <ul style="list-style-type: none"> ○ Mensaje: Contendrá la descripción del resultado de la ejecución o el mensaje de error obtenido según corresponda. ○ Código: Si la ejecución ha tenido éxito contendrá el número de incidencia, número de pedido o el número de identificación de CI asignado según corresponda al servicio solicitado. Si se produjo un error este parámetro no se mostrará. • Si el parámetro modo está definido y su valor es “debug”, además de los datos mostrados al realizar la ejecución en el modo por defecto, se mostrará la traza completa de ejecución obtenida en la realización del servicio. Esta característica será de gran utilidad a la hora de revisar problemas que puedan ocasionarse. • Si el parámetro modo está definido y su valor no es “debug” se mostrará un mensaje de error indicando que el modo indicado no está soportado. 			

Código:	RF016	Prioridad:	E-Esencial
Título:	XML Parser		
Descripción			
<p>Para poder realizar la solicitud registrada por el servicio web es necesario retransmitir esta solicitud a la interfaz de integración empleando el protocolo de comunicación SOAP.</p> <p>Para convertir los datos recibidos del servicio a XML se empleará una funcionalidad de parser de datos integrada en la herramienta SOAP. Esta misma funcionalidad de parser será empleada para convertir la respuesta del servicio de integración de XML a texto.</p> <p>La herramienta SOAP deberá ser capaz de convertir los datos de cualquier solicitud a XML y realizar el envío de este XML a la interfaz de integración que se le indique.</p>			

Código:	RF017	Prioridad:	E- Esencial
Título:	Identificación de usuario solicitante		
Descripción			
<p>Para poder conectar con el servidor que gestiona la herramienta Service-Now y poder realizar gestiones es necesario autenticarse con un usuario registrado de la herramienta.</p> <p>El servicio web requerido solicitará la autenticación de usuario cuando intente conectarse con la herramienta de gestión Service-Now.</p> <p>La gestión de la solicitud empleará este usuario como usuario solicitante o creador del registro según corresponda al servicio solicitado.</p>			

Código:	RF018	Prioridad:	D-Deseable
Título:	Consulta de registros de solicitud y respuesta		
Descripción			
<p>Se debería implementar una interfaz de consulta mediante la cual se puedan consultar las entradas creadas en la BB.DD. tanto al recibir los datos enviados por los usuarios para la creación de las peticiones empleando los servicios web, como cuando se reciban las respuestas de la interfaz de integración con los resultados de las gestiones.</p> <p>Sería optimo poder realizar consultas en base a los siguientes parámetros tanto de forma independiente como combinada:</p> <ul style="list-style-type: none">• Tipo de servicio web solicitado• Tipo de registro (de solicitud o de respuesta de servicio)• Fecha de creación del registro			

3.1.3 Requisitos de las interfaces de integración

Código:	RF019	Prioridad:	E-Esencial
Título:	Tipificaciones de las interfaces de integración		
Descripción			
<p>Será necesario desarrollar una interfaz de integración para cada una de las tipificaciones de servicio web. Cada interfaz de integración recibirá únicamente los datos de las solicitudes del servicio web que le corresponda.</p> <p>Estas interfaces de integración deberán desarrollarse en la herramienta de gestión Service-Now.</p> <p>Es más eficiente crear interfaces independientes ya que de este modo se incrementará la eficiencia de la gestión de peticiones. Las funcionalidades comunes podrán estar desarrolladas de forma modular para que todas las interfaces puedan acceder a ellas sin tener que implementarlas de forma repetida.</p>			

Código:	RF020	Prioridad:	E-Esencial
Título:	Recepción de peticiones de gestión de servicios		
Descripción			
<p>Es necesario que la estructura de los parámetros de entrada de las interfaces de integración coincida con la estructura de los parámetros de salida de los servicios web correspondientes.</p> <p>Para poder realizar correctamente la recepción de los datos será necesario definir un documento WSDL para cada una de las interfaces de integración.</p> <p>A continuación se muestra uno de los prototipos planteados para el desarrollo de la definición de documento WSDL para la entrada de datos de la interfaz de integración del servicio de creación de incidencias.</p>			

```

- <wsdl:definitions targetNamespace="http://www.service-now.com/EDPR_Incident_wsdl">
- <wsdl:types>
- <xsd:schema elementFormDefault="qualified" targetNamespace="http://www.service-now.com/EDPR_Incident_wsdl">
- <xsd:element name="insert">
- <xsd:complexType name="EDPR_Incident_wsdlinsertType">
- <xsd:sequence>
  <xsd:element maxOccurs="1" minOccurs="0" name="requester" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="company" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="request_for" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="computer" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="description" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="category" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="subcategory" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="type" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="group" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
- <xsd:element name="insertResponse">
- <xsd:complexType name="EDPR_Incident_wsdlinsertResponseType">
- <xsd:sequence>
  <xsd:element maxOccurs="1" minOccurs="0" name="request_id" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="message_id" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="incident_data" type="xsd:string"/>
  <xsd:element maxOccurs="1" minOccurs="0" name="incident_description" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
- <wsdl:message name="insertSoapOut">
  <wsdl:part name="EDPR_Incident_wsdl" element="tns:insertResponse"/>
</wsdl:message>
- <wsdl:message name="insertSoapIn">
  <wsdl:part name="EDPR_Incident_wsdl" element="tns:insert"/>
</wsdl:message>
- <wsdl:portType name="ServiceNowSoap">
- <wsdl:operation name="insert">
  <wsdl:input message="tns:insertSoapIn"/>
  <wsdl:output message="tns:insertSoapOut"/>
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="ServiceNowSoap" type="tns:ServiceNowSoap">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
- <wsdl:operation name="insert">
  <soap:operation soapAction="http://www.service-now.com/EDPR_Incident_wsdl/insert" style="document"/>
- <wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
- <wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="ServiceNow_EDPR_Incident_wsdl">
- <wsdl:port name="ServiceNowSoap" binding="tns:ServiceNowSoap">
  <soap:address location="https://silverstormedp2.service-now.com/EDPR_Incident_wsdl.do?SOAP"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Figura 3.1.c: Versión inicial de documento WSDL

Código:	RF021	Prioridad:	E-Esencial
Título:	Gestión de servicios		
Descripción			
<p>Las interfaces de integración deberán procesar los datos recibidos en sus correspondientes entradas de datos de acuerdo con los diferentes módulos y tablas de datos implicados de la herramienta de gestión Service-Now.</p> <p>La realización de la gestión del servicio, siempre que sea exitosa, conllevará la creación de una nueva entrada en el sistema que variará según el servicio realizado.</p>			

Código:	RF022	Prioridad:	E-Esencial
Título:	Retorno de datos al servicio web		
Descripción			
<p>Al finalizar la gestión de una solicitud, la interfaz de integración retornará un mensaje de respuesta xml compuesto por dos parámetros, un mensaje con la descripción del resultado del servicio realizado o el mensaje de error obtenido en caso de que la gestión produjese un error y el número de identificación de la incidencia, pedido o CI creado.</p>			

Código:	RF023	Prioridad:	D-Deseable
Título:	Generación de log de la gestión.		
Descripción			
<p>Al recibir los parámetros de entrada, al enviar el mensaje de respuesta y durante la gestión del servicio solicitado, la interfaz de integración deberá generar entradas en el log del sistema con los diferentes hitos de la ejecución.</p>			

3.2 Requisitos No Funcionales

3.2.1 Requisitos de Usabilidad

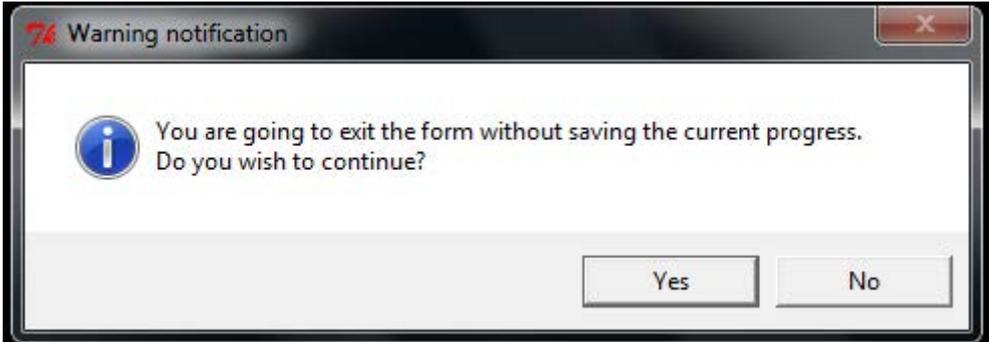
Código:	RU001	Prioridad:	E-Esencial
Título:	Facilidad de uso		
Descripción			
<p>Todos los elementos que componen el sistema deben ser fáciles de usar tanto para los usuarios como para los administradores, ofreciendo una interfaz que permita acceder a la información y ejecutar las operaciones más habituales sin necesidad de recorrer múltiples menús.</p>			

Código:	RU002	Prioridad:	E-Esencial
Título:	Interfaces gráficas		
Descripción			
<p>Todos los elementos que componen el sistema deben ofrecer como interfaz primaria una interfaz gráfica que permita la realización de las diferentes operaciones estipuladas.</p> <p>La única operación que supone una excepción a esta norma es la creación de códigos QR empleando ejecución batch, la cual por definición no requiere de interfaz gráfica.</p>			

Código:	RU003	Prioridad:	D-Deseable
Título:	Facilidad de aprendizaje		
Descripción			
<p>Los nuevos usuarios que empleen las diferentes funcionalidades que componen el sistema deberán poder dominar estas funcionalidades con facilidad, por lo que es necesario que las diferentes herramientas sean predecibles, consistentes y presentes únicamente los datos y formularios requeridos.</p>			

Código:	RU004	Prioridad:	E-Esencial
Título:	Documentación		
Descripción			
<p>Para cada una de las funcionalidades desarrolladas, se deberá facilitar documentación de uso y administración de las herramientas a los grupos de usuarios implicados en las correspondientes actividades.</p>			

Código:	RU005	Prioridad:	E-Esencial
Título:	Lenguaje empleado		
Descripción			
<p>Para facilitar su uso y comprensión por el mayor número de usuarios, todos los componentes del sistema deberán estar disponibles en lengua inglesa. Se deberá procurar emplear formulas anglicanas comunes para no dificultar el uso de los usuarios no nativamente angloparlantes.</p>			

Código:	RU006	Prioridad:	E-Esencial
Título:	Notificación de errores y avisos		
Descripción			
<p>Tanto en el creador de códigos QR como en los servicios web se deberán generar notificaciones con la información de los errores ocurridos.</p> <p>Adicionalmente se deberán generar notificaciones de aviso cuando una acción pueda generar pérdida de datos al usuario, como puede ser el borrado de datos o abandonar un menú sin guardar el progreso actual.</p> <p>La siguiente figura ejemplifica uno de las notificaciones de avisos</p>			
			
Figura 3.2.a: Ejemplo notificación de aviso			

Código:	RU007	Prioridad:	D-Deseable
Título:	Empleo de software con licencias abiertas		
Descripción			
<p>En el caso de necesitar emplear librerías externas se deberá dar preferencia a librerías o software con licencias poco restrictivas y emplear únicamente software del que se pueda analizar su código fuente.</p>			

3.2.2 Requisitos de Disponibilidad

Código:	RD001	Prioridad:	E-Esencial
Título:	Disponibilidad del sistema		
Descripción			
<p>El sistema deberá garantizar una alta disponibilidad de al menos tres nueves [9] para los servicios web que lo integran.</p> <p>La disponibilidad de la interfaz de integración está garantizada por la alta disponibilidad de la plataforma Service-Now.</p> <p>La disponibilidad de los servicios web estará garantizada por EDPR, ya que los servicios web serán incluidos en uno de los servidores virtuales disponibles. Estos servidores soportan servicios críticos por lo que disponen de mecanismos de replicación y reinicio si detectan problemas de accesibilidad.</p> <p>Por su parte el creador de QR se ejecutará en local, y no deberá necesitar acceder a ninguna librería web.</p>			

3.2.3 Requisitos de Seguridad

Código:	RS001	Prioridad:	E-Esencial
Título:	Seguridad en acceso a servicios web		
Descripción			
<p>Para poder acceder a la funcionalidad de los servicios web es necesario disponer de los credenciales de un usuario de EDPR.</p> <p>Las credenciales de los usuarios son actualizados de forma periódica por política de empresa. Adicionalmente las contraseñas de los usuarios deben cumplir ciertos parámetros de configuración.</p> <p>Además la herramienta Service-Now cuenta con controles de nivel de acceso, de forma que los usuarios solo puedan acceder a las funcionalidades para las que disponen de los roles correspondientes.</p> <p>Los servicios web se encontrarán alojados en un servidor ubicado en uno de los CPD de la red interna de EDPR. Este CPD presenta un alto grado de control de acceso, medidas de seguridad física y seguridad informática.</p>			

Código:	RS002	Prioridad:	E-Esencial
Título:	Copias de seguridad		
Descripción			
<p>Para poder presentar un informa correcto en las auditorías, es necesario garantizar que no se pierda la información de la gestión de servicios web.</p> <p>Los servidores alojados en el CPD objetivo realizan diariamente un backup incremental. Semanalmente, en un horario de bajo tráfico en red, se realiza un backup completo de los datos alojados en los servidores.</p>			

Código:	RS003	Prioridad:	E-Esencial
Título:	Acceso restringido a BB.DD. de registro de solicitudes de servicio y respuesta servicio de servicios web.		
Descripción			
<p>Los registros de solicitudes de servicios y los registros de las respuestas son guardados en una BB.DD. ubicada en el mismo servidor web en el que se alojan los servicios web.</p> <p>A diferencia de las reglas de acceso a los servicios web previamente descritos, el acceso a esta BB.DD. al igual que muchos otros servicios alojados en el servidor estará limitado a los administradores de sistemas.</p>			

Código:	RS004	Prioridad:	E-Esencial
Título:	Uso de prepared statement para accesos a BB.DD.		
Descripción			
<p>Los registros de solicitudes de servicios y los registros de las respuestas son guardados en una BB.DD. ubicada en el mismo servidor web en el que se alojan los servicios web.</p> <p>A diferencia de las reglas de acceso a los servicios web previamente descritos, el acceso a esta BB.DD. al igual que muchos otros servicios alojados en el servidor estará limitado a los administradores de sistemas.</p>			

3.2.4 Requisitos de Rendimiento

Código:	RR001	Prioridad:	E-Esencial
Título:	Escalabilidad del sistema		
Descripción			
<p>El sistema debe soportar poder reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida.</p> <p>La escalabilidad del servicio web estará garantizada por los recursos de la granja de virtualización en la que se aloja el servidor virtual que soporta los servicios web. En caso de necesidad se incrementarán los recursos asignados de manera automática y se generará un aviso destinado a los administradores del servidor y del servicio.</p> <p>La escalabilidad del servicio de integración estará garantizada por el proveedor al gestionarse en el servidor de Service-Now.</p>			

Código:	RR002	Prioridad:	E-Esencial
Título:	Latencia del Sistema		
Descripción			
<p>Puesto que la granja de virtualización en la que se ubica el servidor virtual que aloja los servicios web se encuentra en una ubicación concreta, los accesos al mismo podrán presentar una latencia variable en función de la ubicación desde la que se realice el acceso.</p> <p>El tiempo de gestión de los diferentes servicios web será uniforme ya que depende de la conexión entre el servicio web y la herramienta Service-Now y de la capacidad de cómputo de la herramienta.</p>			

Código:	RR003	Prioridad:	D-Deseable
Título:	Eficiencia de los algoritmos		
Descripción			
<p>Los algoritmos empleados por los diferentes elementos que componen el sistema deberán ser eficientes en el uso de los recursos habilitados. El objetivo es minimizar el uso de recursos manteniendo el rendimiento al máximo.</p>			

Código:	RR004	Prioridad:	E-Esencial
Título:	Eficiencia de consultas a BB.DD.		
Descripción			
<p>Las diferentes consultas que se realizan por un lado en el proceso de administración de códigos QR y por otro en la gestión y administración de servicios de los servicios web han de ser lo más eficientes posible, para ello se acotaran las consultas en la base de datos siempre que sea posible.</p> <p>Las consultas realizadas se mantendrán en memoria hasta que se realice otra consulta, se finalice la ejecución o sea descartado el bloque de memoria por el OS. Gracias a esto será posible realizar subconsultas de los datos sin necesidad de volver a conectar con la BB.DD. Esta técnica se empleará por ejemplo al listar los códigos QR de diez en diez</p>			

3.2.5 Requisitos de Portabilidad

Código:	RP001	Prioridad:	E-Esencial
Título:	Portabilidad del creador de QR		
Descripción			
<p>La aplicación de creación de códigos QR deberá poder ejecutarse en diferentes plataformas, al menos en distribuciones Linux y en Windows 7 o superior.</p> <p>El código fuente del software será capaz de reutilizarse cuando el software pase de una plataforma a otra.</p>			

Código:	RP002	Prioridad:	E-Esencial
Título:	Portabilidad de los servicios web		
Descripción			
<p>Siempre que se cumplan los requisitos mínimos de ejecución, los diferentes servicios web deberán poder ejecutarse sin problemas independientemente del sistema operativo del dispositivo en la que se alojen.</p>			

Diseño

En este apartado se detallarán el diseño y la arquitectura del sistema propuesto. Se realizará una descripción del propósito del sistema, las partes que componen el sistema, las plataformas planteadas y las diversas etapas que desarrolla cada componente para procesar los datos requeridos.

4.1 Descripción General

A continuación se recogen figuras y descripciones del sistema a desarrollar.

Se va a proceder a describir de forma independiente por un lado la gestión de la creación de códigos QR y por otro la gestión de los servicios web.

4.1.1 Creador de Códigos QR



Figura 4.1.a: Esquema del Creador de Códigos QR

La aplicación para la creación de códigos QR dispondrá de las funcionalidades necesarias para la creación y administración de una BBDD de códigos QR. Existen varios tipos diferenciados de códigos QR que pueden crearse, el formulario a cumplimentar difiere en función del tipo seleccionado. El número de tipificaciones existente se puede llegar a incrementar, debiendo crear los correspondientes servicios web si se desea facilitar el servicio completo.

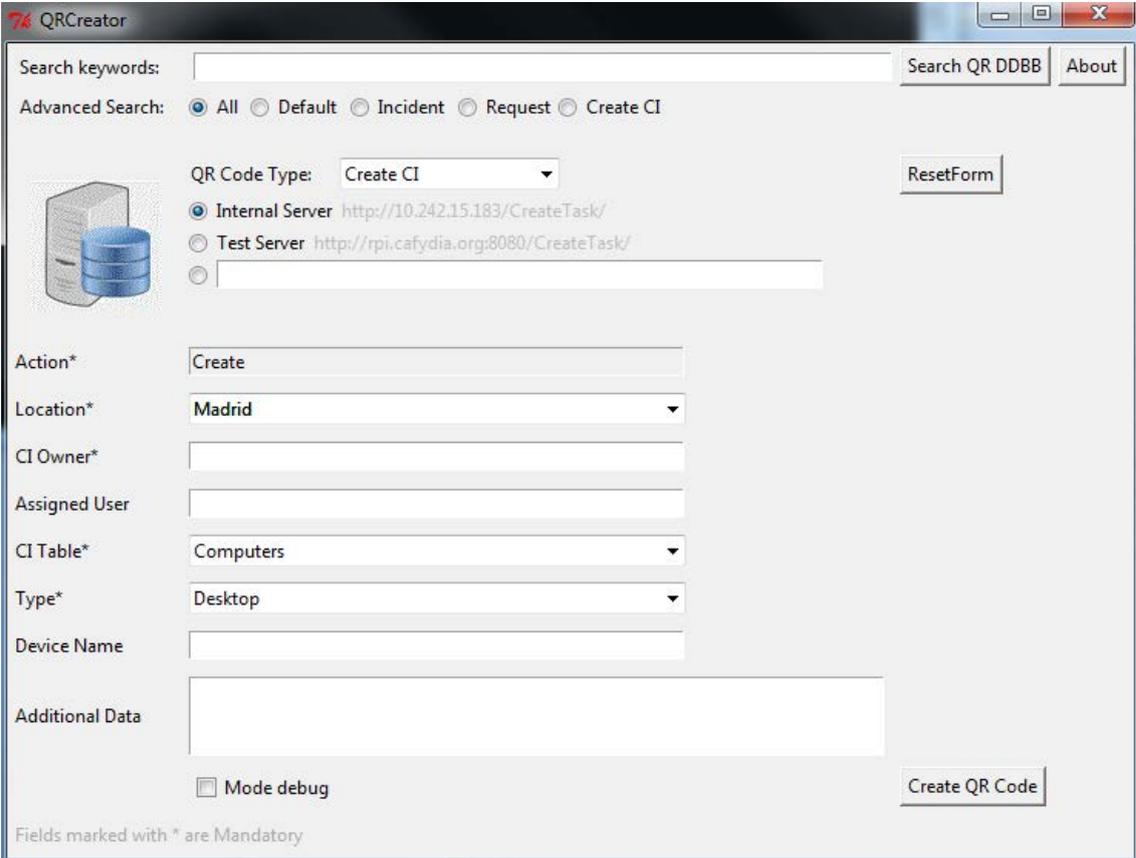
No se podrá crear un código QR si no se cumplimentan los datos obligatorios del formulario de creación, estos datos obligatorios pueden cambiar según la tipificación seleccionada.

Los códigos QR serán creados mayoritariamente por técnicos de IT o técnicos de soporte local, usuarios a los que se deberá dar formación específica. Una vez crean los códigos QR se los suministrarán a los usuarios finales o los distribuirán (por las oficinas, equipos hardware, etc.) para su posterior uso. Esto no implica que el usuario final no pueda autogenerar sus propios códigos QR siempre que disponga de la aplicación.

Las principales acciones que se deben poder realizar con esta aplicación son la creación, búsqueda, modificación y borrado de códigos QR.

Para crear un código QR existen varias casuísticas posibles:

- Seleccionar la tipificación deseada, cumplimentar como mínimo los datos indicados como obligatorios del formulario y ejecutar la acción de crear nuevo código QR.
- Cargar un código QR previamente creado y modificarlo
- Cargar un código QR previamente creado y emplearlo para crear un nuevo código QR a raíz de él.



The screenshot shows the QRCreator application window. The title bar reads 'QRCreator'. The interface includes a search bar at the top with 'Search keywords:' and buttons for 'Search QR DDBB' and 'About'. Below the search bar, there are radio buttons for 'Advanced Search' with options: 'All' (selected), 'Default', 'Incident', 'Request', and 'Create CI'. A 'QR Code Type' dropdown menu is set to 'Create CI', with a 'ResetForm' button to its right. Underneath, there are radio buttons for server selection: 'Internal Server' (selected) with URL 'http://10.242.15.183/CreateTask/', 'Test Server' with URL 'http://rpi.cafydia.org:8080/CreateTask/', and an empty option. The main form contains several fields: 'Action*' (text input with 'Create'), 'Location*' (dropdown menu with 'Madrid'), 'CI Owner*' (text input), 'Assigned User' (text input), 'CI Table*' (dropdown menu with 'Computers'), 'Type*' (dropdown menu with 'Desktop'), 'Device Name' (text input), and 'Additional Data' (text area). At the bottom left, there is a checkbox for 'Mode debug'. At the bottom right, there is a 'Create QR Code' button. A footer note states 'Fields marked with * are Mandatory'.

Figura 4.1.b: Ejemplo de parámetros obligatorios para tipificación Create CI

Al igual que ocurre para la creación, para la eliminación de un código QR existen varias casuísticas posibles:

- Buscar el código QR que se desea eliminar, acceder a él y emplear la acción de eliminar.
- Buscar el código QR que se desea eliminar, emplear la acción de eliminar disponible en la ventana de resultados de la búsqueda.

Al igual que ocurre para la creación y el borrado de códigos QR, existen varias casuísticas posibles para la realización de búsquedas:

- Introducir las palabras clave deseadas en la barra de búsqueda, se seleccionarán aquellos códigos que las contengan y se listarán ordenados por tipo y variables.
- Emplear la funcionalidad de búsqueda avanzada y delimitar el tipo deseado de forma que solo se muestren los códigos que contengan las palabras claves indicadas y además pertenezcan a uno de los tipos seleccionados.

4.1.2 Servicios Web

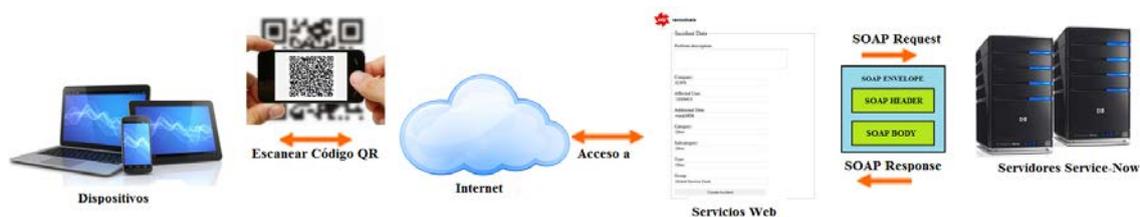


Figura 4.1.c: Esquema de los Servicios Web

El objetivo de los servicios web desarrollados es, dados los parámetros obtenidos de escanear una imagen de código QR desde un dispositivo, habilitar el formulario correspondiente para poder realizar la solicitud del servicio requerido por el usuario en la herramienta de gestión de servicios de IT Service-Now.

Los usuarios acceden a los servicios web mediante un acceso a Internet. Existe un servidor que aloja los diferentes componentes del servicio web y su BBDD.

El servicio web al que se accederá dependerá del tipo de código QR escaneado. Una vez en el servicio web se accederá a un formulario en el que será necesario revisar los parámetros cargados a raíz de los datos escaneados del código QR y cumplimentar los parámetros restantes.

Cuando los diferentes parámetros del formulario estén cumplimentados el usuario podrá solicitar la realización del servicio web. Para realizar esta acción el servicio web generará un mensaje de solicitud SOAP que enviará a Service-Now donde la interfaz de integración correspondiente lo recibirá y gestionará.

Estas interfaces de integración estarán implementadas en la herramienta Service-Now, lo que les permitirá acceder a los diferentes módulos y tablas de datos para realizar las gestiones requeridas.

Una vez finalizada esta gestión, desde la interfaz de integración se generará un mensaje de respuesta SOAP que se enviará de vuelta al servicio web. El servicio Web realizará una operación de parser de datos y mostrará al usuario el resultado de la solicitud realizada.

La BBDD se empleará tanto para realizar un registro de las solicitudes enviadas y de las respuestas recibidas de la herramienta de gestión Service-Now, como para realizar transformaciones de datos antes de efectuar los envíos de forma que por ejemplo, se puedan mostrar a los usuarios los nombres de los grupos o las categorías de datos, mientras que en las peticiones que se envían a la herramienta de gestión se envían los sys_id identificativos correspondientes. De esta forma se aumenta la legibilidad de los formularios, la velocidad de gestión de la petición de la herramienta y se reduce el número de valores tanto a introducir en el código QR (únicamente la cadena de texto o el identificador) como a enviar en las diferentes pedidos (únicamente el identificador).

4.2 Arquitectura del Sistema

Para crear el sistema planteado es necesario desarrollar tres partes diferenciadas, la aplicación de códigos QR, los servicios web y las interfaces de integración para Service-Now.

Es necesario seleccionar los lenguajes de programación que se van a emplear en cada una de las partes diferenciadas del sistema. Cada lenguaje será seleccionado en base a los requisitos del desarrollo que se deberá realizar con dicho lenguaje.

Se va a proceder a describir las características que deberán cumplir cada una de las partes que integran el sistema y a exponer un análisis de los lenguajes seleccionados.

4.2.1 Creador de Códigos QR

El creador de códigos QR será una aplicación informática para ordenador. Deberá garantizar como mínimo las siguientes características:

- Ser capaz de crear códigos QR, como archivos de imagen que luego se puedan imprimir a partir de los parámetros introducidos.
- Almacenar y posteriormente recuperar los parámetros introducidos en la creación de códigos.
- Disponer de interfaz gráfica y posibilitar la ejecución batch.
- Ser multiplataforma, pudiéndose ejecutar como mínimo en Linux y Windows.

El lenguaje a emplear para la creación de esta aplicación deberá, además de ser capaz de cumplir las características de la aplicación, preferentemente ser orientado a objetos, permitir el uso de expresiones regulares y ser aceptablemente eficiente en el procesamiento de datos.

A continuación se adjuntan dos figuras en las que se pueden apreciar comparativas de los principales lenguajes de programación orientados a objetos.

En la primera se puede apreciar una comparación de las características principales de los lenguajes de programación POO más empleados actualmente.

	Ruby	Java	C#	C++	Python	Perl
Object-Orientation	Pure	Hybrid	Hybrid	Hybrid / Multi-Paradigm	Hybrid	Add-On / Hybrid
Static / Dynamic Typing	Dynamic	Static	Static	Static	Dynamic	Dynamic
Generic Classes	N/A	No	No	Yes	N/A	N/A
Inheritance	Single class, multiple "mixins"	Single class, multiple interfaces	Single class, multiple interfaces	Multiple	Multiple	Multiple
Feature Renaming	Yes	No	No	No	No	No
Method Overloading	No	Yes	Yes	Yes	No	No
Operator Overloading	Yes	No	Yes	Yes	Yes	Yes
Higher Order Functions	Blocks	No	No	No	Lambda Expressions	Yes (???)
Lexical Closures	Yes (blocks)	No	No	No	Yes (since 2.1)	Yes
Garbage Collection	Mark and Sweep	Mark and Sweep or Generational	Mark and Sweep or Generational	None	Reference Counting	Reference Counting
Uniform Access	Yes	No	No	No	No	No
Class Variables / Methods	Yes	Yes	Yes	Yes	No	No
Access Control	public, protected, private	public, protected, "package", private	public, protected, private, internal, protected internal	public, protected, private, "friends"	Name Mangling	None
Multithreading	Yes	Yes	Yes	Libraries	Yes	No
Regular Expressions	Built-in	Standard Library	Standard Library	No	Standard Library	Built-in
Pointer Arithmetic	No	No	Yes	Yes	No	No
Language Integration	C, C++, Java	C, some C++	All .NET Languages	C, Assembler	C, C++, Java	C, C++

Figura 4.2.a: Comparación de características de lenguajes POO

En la segunda se puede apreciar los resultados de una comparación de la velocidad y consumo de memoria de varios lenguajes a la hora de realizar manipulaciones de cadenas de caracteres [10].

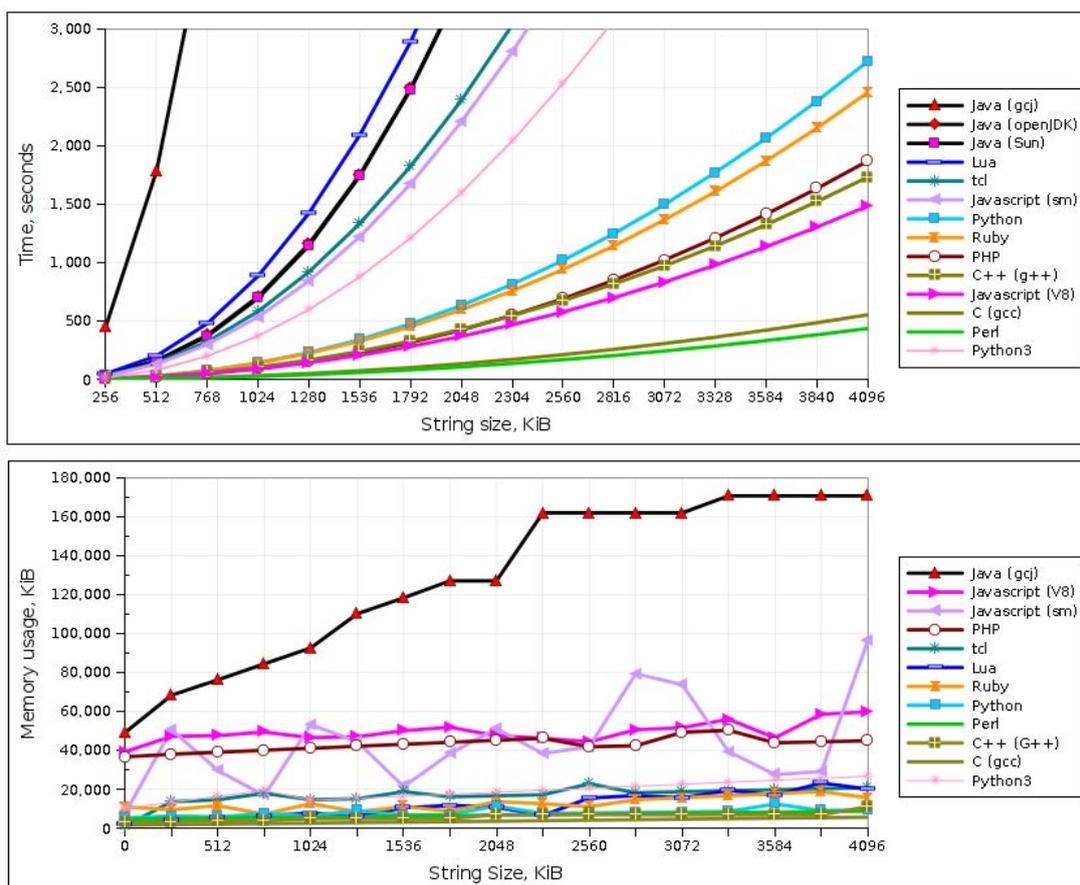


Figura 4.2.b: Comparación de velocidad y rendimiento

De los datos obtenidos en las pruebas de velocidad vamos a resumir los de los lenguajes descritos previamente:

Respecto a la velocidad, C, C++, Perl, Ruby y Python disponen de una velocidad altamente aceptable, mientras que respecto al uso de memoria destacan C, C++, Perl Y Python, seguidos de cerca por Ruby. Para ambas estadísticas el resultado de Java, como era de esperar, es nefasto.

A pesar de no haber obtenido el mejor rendimiento en las pruebas, Ruby es un lenguaje que soporta el paradigma POO y que está orientado a facilitar la productividad del programador, presentar una fácil comprensión del código y aplicar el principio POLA [11]. Destaca también la centralización del mantenimiento de paquetes a través de las gemas. Adicionalmente al ser interpretado es intrínsecamente multiplataforma, por lo que cumple todos los requisitos solicitados para ser seleccionado.

Una vez seleccionado el lenguaje a emplear, puesto que gran parte de las acciones de la aplicación tendrán lugar mediante la interfaz gráfica, es interesante analizar las diferentes GUI toolkits existentes.

La siguiente figura resume los resultados obtenidos:

Toolkit	Principales Ventajas	Principales Inconvenientes
FXRuby	-Dispone de abundante documentación -Incluido nativamente en el instalador de Ruby para Windows. -Instalable mediante gema.	-Interfaces generadas presentan aspecto de Windows XP en todos los OS.
Shoes	-Interfaz simple -Aspecto de los resultados agradable.	
TK	-Incluido en las librerías como estándar. -Aspecto de los resultados nativo al OS	

Figura 4.2.c: Comparativa de Ruby GUI toolkits

Se han agregado referencias a las herramientas mencionadas en la sección 10.2 Bibliografía Consultada.

En base a estos resultados, la selección quedaba reducida principalmente a Shoes o TK. Shoes presenta mayor popularidad al estar orientado a facilitar la programación de interfaces. Sin embargo, puesto que en el pasado ya se había empleado TK y el hecho de que está integrado como librería estándar en Ruby, finalmente el GUI seleccionado es TK.

El principal problema que tenía TK es que no había mucha documentación para Ruby. Los usuarios de TK en Ruby tenían que emplear la documentación para Perl o Python e intentar adaptarla. Actualmente existe bastante documentación para Ruby sobre Tk ^[12].

Respecto al almacenamiento de datos, Ruby dispone de la gema `sqlite3` que permite la creación, acceso y uso de BBDD SQLite. Se empleará SQLite ya que a diferencia de los sistemas de gestión de BBDD cliente-servidor, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. Los datos de la BBDD son almacenados en un fichero en la máquina en la que se ejecuta el programa.

4.2.2 Servicios Web

Los servicios web junto con la BBDD en la que alojarán las solicitudes, los resultados y las tablas de equivalencia de datos estarán alojados en un servidor virtual. Los servicios web deberán garantizar como mínimo las siguientes características:

- Disponibilidad de servicio.
- Multiplataforma.
- Generación, envío y recepción de mensajes SOAP.
- Comprobación de la validez de los datos del formulario.
- Conexión a la interfaz de la herramienta Service-Now y petición de autenticación del usuario.

Para el desarrollo de la estructura que realizarán los servicios web descritos se requiere preferentemente un lenguaje de programación que cumpla los siguientes requisitos:

- Orientación al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en BBDD.
- Multiplataforma.
- Script del lado del servidor.
- Libre.
- Bien documentado.

Al requerir que se ejecute en el lado del servidor se ha limitado ligeramente los posibles lenguajes seleccionables. Esto en realidad no nos supone un problema, ya que entre los lenguajes que si cumplen esta condición se encuentran Ruby, PHP y Python entre otros.

Sería interesante emplear RoR para mantener un único lenguaje tanto para el creador de códigos QR como para los servicios web. Sin embargo finalmente se ha seleccionado PHP como lenguaje para el desarrollo de los servicios web debido a que cumple los requisitos solicitados y es uno de los lenguajes empleados por excelencia para el desarrollo de sitios web. El hecho de que sea interpretado y tenga un rendimiento ligeramente inferior a lenguajes de bajo nivel no supone un problema. Además el código fuente PHP debería ser invisible al navegador web y al cliente al ejecutarse en el servidor, el cual, estará protegido por las capas de seguridad de la empresa, con lo que debería ser algo más seguro y confiable.

Acceso directo a los servidores se concede a los administradores de sistema y a unos pocos usuarios especializados, por lo que se debería poder garantizar la integridad y la confidencialidad de los servicios web y su BBDD. En cualquier caso, éstas se pueden incrementar cifrando la BBDD y ofuscando el código de los servicios web.

Para mantener la consistencia del sistema la BBDD empleada en los servicios web será también SQLite. Puesto que queremos poder acceder de forma eficaz a los registros almacenados es necesario desarrollar una interfaz mínima de consulta que permita listar o exportar los registros contenidos en las diferentes tablas.

Para realizar el servicio propiamente dicho, será necesario desarrollar las siguientes estructuras:

- Parser que realice la conversión de los datos de la solicitud a XML para que se puedan enviar la misma como SOAP request.
- Parser que realice la conversión desde XML a texto para poder interpretar los datos contenidos en el SOAP response.
- Cliente SOAP que permita la conexión a Service-Now, el envío de solicitudes y la recepción de respuestas.
- Un formulario de creación de solicitudes por cada tipificación de código QR desarrollada. De esta forma evitamos tener que realizar transformaciones en el formulario en función del tipo solicitado y además distribuimos la carga de trabajo entre varios servicios.

- Un formulario de recepción de la respuesta obtenida de las solicitudes. Este formulario deberá mostrar datos tanto si ha habido éxito en la gestión, como si la gestión ha fallado por algún motivo. En los casos en lo que tenga éxito se mostrará el resultado de la gestión realizada y en los que la gestión fracase se mostrará la traza el mensaje de error correspondiente.

Se deberá tener en cuenta la posibilidad de sufrir ataques de SQL injection. PHP disponía de la función `mysql_real_escape_string` pero puesto que todas las funciones de la extensión mysql para PHP fueron deprecadas en la versión PHP 5.5 para evitar sufrir este tipo de ataques será necesario emplear Prepared statements.

Por otro lado, el formulario de recepción de respuesta no mostrará directamente el resultado al usuario, ya que se aplicara PRG para evitar el envío de formularios duplicados.

La siguiente figura ilustra la actuación del patrón de diseño PRG.

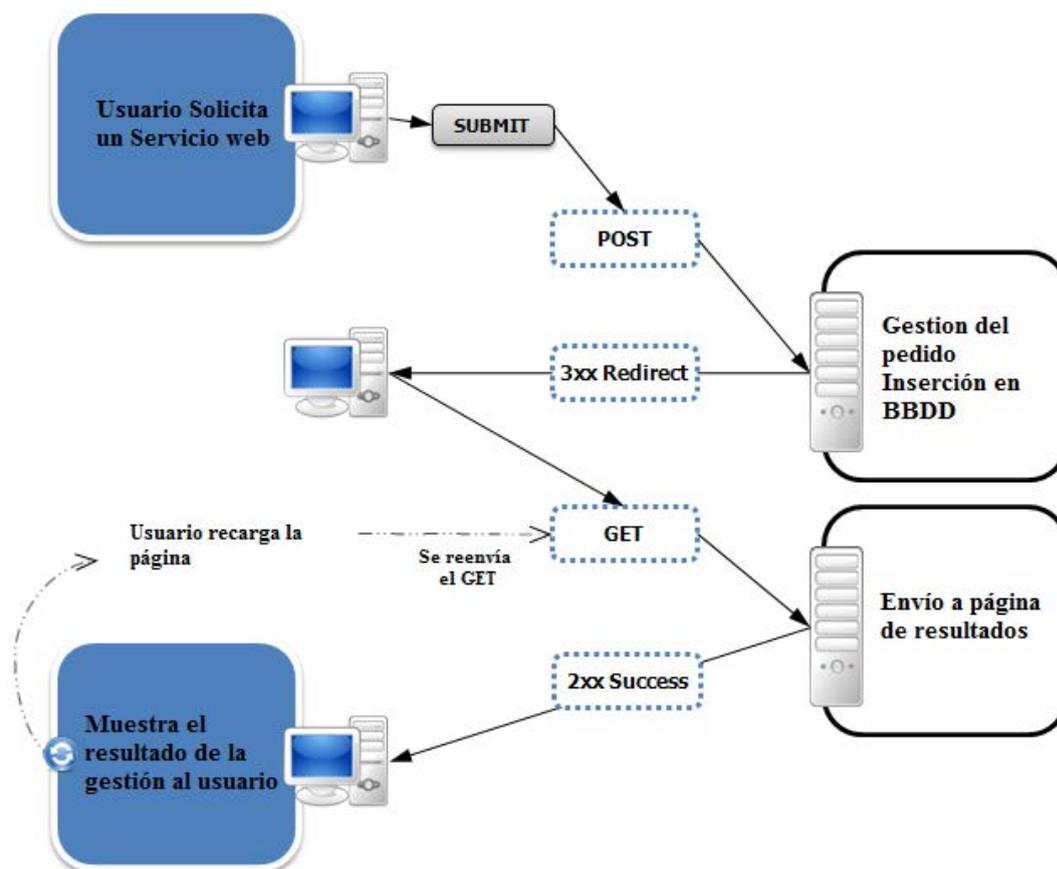


Figura 4.2.d: Esquema de patrón de diseño PRG

Otra opción para evitar el reenvío de solicitudes duplicadas era agregar un nonce a las solicitudes. Puesto que se logró evitar el reenvío de formularios con PRG no fue necesario implementar esta solución que hubiese requerido realizar una llamada adicional de los clientes a los servicios web para solicitar el nonce y luego poder agregarlo a las solicitudes.

4.2.3 Interfaces de Integración

Las interfaces de integración se desarrollarán en la herramienta Service-Now y deberán adaptarse y ser capaces de interactuar con las estructuras empleadas por dicha herramienta de gestión.

Con el fin de lograr estos objetivos, el desarrollo de las interfaces se realizará en los lenguajes nativos de la plataforma Service-Now. De acuerdo con esto los lenguajes principales a emplear para el desarrollo de las interfaces de integración serán JS y XML.

Puesto que cada servicio web a desarrollar tendrá parámetros diferenciados y accederá a diferentes tablas de Service-Now se procederá a generar una interfaz de integración por cada servicio web.

Las interfaces de integración contendrán la lógica necesaria para a partir de los datos de entrada realizar el servicio solicitado y devolver el resultado obtenido.

Además de la lógica de gestión para la realización de los diferentes tipos de solicitudes de servicios, cada una de las interfaces de integración debe contar con una definición de documento WSDL que contenga la estructura de los parámetros de los mensajes SOAP de solicitud y respuesta.

No será necesario implementar un servicio parser en Service-Now, pues ya se encuentra integrado en el sistema.

4.3 Bases de Datos

A continuación se detalla la estructura de las bases de datos que empleará el sistema. Se detallarán de forma independiente la BBDD del creador de códigos QR y la BBDD empleada por los servicios web.

4.3.1 Creador de códigos QR

A continuación se incluye la figura con el modelo lógico de las entidades de BBDD en las que se apoyará el creador de códigos QR y las descripciones de las funcionalidades tanto de las tablas como de los campos que contienen.

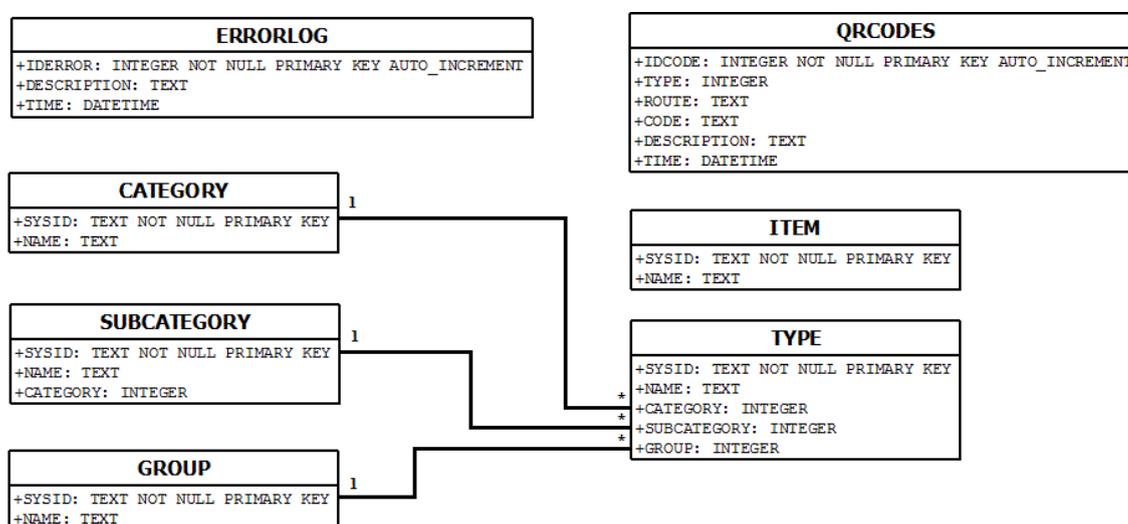


Figura 4.3.a: Modelo Físico/Lógico de la BBDD del creador de códigos QR

La tabla ERRORLOG contiene las trazas de los errores capturados en los diferentes try-catch contenidos en la aplicación. Se almacena tanto la traza como la fecha y hora en la que se generó el error.

Campo	Tipo	Observaciones
IDERROR	INTEGER	Identificador del error y clave primaria.
DESCRIPCION	TEXT	Contiene la traza del error generado.
TIME	DATETIME	Contiene la fecha y hora de la inserción.

Figura 4.3.b: Descripción tabla ERRORLOG

La tabla QRCODES contiene el tipo, los parámetros y la fecha de creación o última modificación de los códigos QR. Los parámetros de los códigos se encuentran concatenados en una misma variable, gracias al tipo del código se pueden clasificar y filtrar en caso de buscar códigos QR de un tipo determinado.

Campo	Tipo	Observaciones
IDCODE	INTEGER	Identificador del código QR y clave primaria.
TYPE	INTEGER	Contiene el tipo de código QR.
ROUTE	TEXT	Contiene el parámetro de ruta del código QR
CODE	TEXT	Contiene los parámetros de creación del código QR
DESCRIPTION	TEXT	Contiene una descripción del código QR creado.
TIME	DATETIME	Contiene la fecha y hora de la inserción o última modificación.

Figura 4.3.c: Descripción tabla QRCODES

Las tablas restantes se emplean principalmente en la conversión de datos entre la interfaz de usuario y los datos a enviar, al usuario se muestra el nombre pero se envía el sys_id. Puesto que los sys_id son únicos se emplean como clave primaria.

Las tablas CATEGORY, SUBCATEGORY, TYPE y GROUP se emplean en la tipificación de códigos QR Incident, mientras que la tabla ITEMS se emplea en el tipo Request. Al cargarse los datos desde la BBDD resulta más sencillo actualizarlos de acuerdo con las actualizaciones que se produzcan en la herramienta ServiceNow, exportando las modificaciones producidas en la herramienta de gestión e importándolas en la BBDD.

La siguiente tabla ilustra la funcionalidad de los diferentes campos contenidos en estas tablas. El campo REFERENCIA hace alusión a los posibles campos referenciados, por ejemplo, para la tabla SUBCATEGORY el campo CATEGORY es una referencia a la tabla homónima.

Campo	Tipo	Observaciones
SYSID	TEXT	Contiene el código sys_id de identificación del elemento en la herramienta Service-Now. Adicionalmente se emplea como clave primaria.
NAME	TEXT	Cadena de texto que se mostrará a los usuarios.
REFERENCIA*	INTEGER	Contiene el identificador del elemento referenciado en al que pertenece.

Figura 4.3.d: Descripción genérica de las tablas de equivalencia.

4.3.2 Servicios Web

A continuación se incluye la figura con el modelo lógico de las entidades de BBDD en las que se apoyarán los servicios web y las descripciones de las funcionalidades tanto de las tablas como de los campos que contienen.

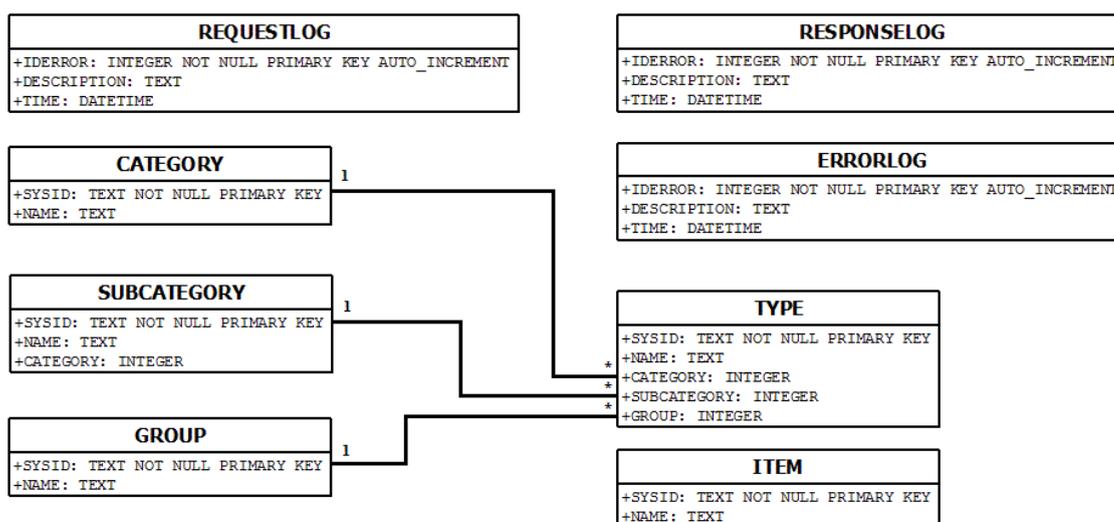


Figura 4.3.e: Modelo Físico/Lógico de la BBDD de los servicios web

La tabla REQUESTLOG contiene las trazas de las solicitudes de servicio realizadas por los usuarios a la herramienta de gestión de servicios Service-Now. Se almacenan tanto los datos de la solicitud como la fecha y hora en la que se generó la solicitud.

Campo	Tipo	Observaciones
IDREQUEST	INTEGER	Identificador de la solicitud y clave primaria.
DESCRIPCION	TEXT	Contiene la traza de la solicitud enviada.
TIME	DATETIME	Contiene la fecha y hora de la inserción.

Figura 4.3.f: Descripción tabla REQUESTLOG

La tabla RESPONSELOG contiene las trazas de las respuestas a las solicitudes de servicios realizadas por los usuarios a la herramienta de gestión de servicios Service-Now. Se almacenan tanto los datos de la respuesta como la fecha y hora en la que se recibió la respuesta.

Campo	Tipo	Observaciones
IDRESPONSE	INTEGER	Identificador de la respuesta y clave primaria.
DESCRIPCION	TEXT	Contiene la traza de la respuesta recibida.
TIME	DATETIME	Contiene la fecha y hora de la inserción.

Figura 4.3.g: Descripción tabla RESPONSELOG

La tabla ERRORLOG contiene las trazas de los errores capturados en los diferentes if(\$err) contenidos en la aplicación. Se almacena tanto la traza como la fecha y hora en la que se generó el error.

Campo	Tipo	Observaciones
IDERROR	INTEGER	Identificador del error y clave primaria.
DESCRIPCION	TEXT	Contiene la traza del error generado.
TIME	DATETIME	Contiene la fecha y hora de la inserción.

Figura 4.3.h: Descripción tabla ERRORLOG

Las tablas CATEGORY, SUBCATEGORY, TYPE, GROUP e ITEM se emplean en la conversión de datos entre la interfaz de usuario y los datos a enviar; al usuario se muestra el nombre pero se trabaja con el sys_id. La descripción de los campos de estas tablas coincide con la realizada para las mismas tablas en el apartado 4.3.1 Creador de códigos QR

4.4 Casos de Uso

A continuación se enumeran y detallan los diferentes casos de usos que la aplicación deberá contemplar. Se van a listar de forma independiente los casos de uso del creador de QR y los de la solicitud y gestión de servicios web.

4.4.1 Casos de Uso de Creador de QR

CU-001	Crear código QR	
Descripción	Los usuarios deben ser capaces de crear códigos QR funcionales.	
Precondición	Disponer y acceder a la aplicación para la creación de códigos QR.	
Secuencia Principal	Paso	Acción
	1	Seleccionar el tipo de código QR que se desea crear.
	2	Cumplimentar los diferentes campos de parámetros necesarios presentes en el formulario de la tipificación de código QR seleccionado.
	3	Emplear la opción de crear Código.
	4	Los parámetros del código QR se guardarán en la BB.DD. y una imagen PNG del código generado se guardará en el directorio predefinido.
Secuencia Alternativa	Paso	Acción
	3	Si hay campos obligatorios sin valor el código QR no se creará, y se generará un aviso notificando que algún valor es invalido.
Postcondición	El usuario podrá emplear el código QR creado o continuar empleando la aplicación de creación de códigos QR.	
Notas		

CU-002	Buscar códigos QR previamente creados	
Descripción	Los usuarios deben ser capaces de localizar códigos QR creados.	
Precondición	Disponer y acceder a la aplicación para la creación de códigos QR.	
Secuencia Principal	Paso	Acción
	1	Se deberá poder buscar el código QR empleando palabras clave en el buscador
	2	Se deberán listar los códigos que contengan las palabras claves introducidas.
Secuencia Alternativa	Paso	Acción
	1	Si se realiza una búsqueda sin parámetros, al buscar se listarán todos los códigos existentes de forma ordenada.
	2	Si se emplean parámetros de búsqueda que no produzcan resultados se deberá mostrar un aviso indicando que las palabras clave empleadas no producen resultados
Postcondición	El usuario podrá seleccionar y realizar acciones sobre los códigos listados.	
Notas		

CU-003	Realizar modificaciones en códigos QR previamente creado	
Descripción	Los usuarios deben ser capaces de modificar los parámetros de un código QR creado previamente.	
Precondición	Realizar una búsqueda de códigos QR	
Secuencia Principal	Paso	Acción
	1	Se deberá seleccionar un QR del listado, al seleccionarse se cargarán sus parámetros en el formulario correspondiente a su tipo.
	2	Se modificarán los parámetros del formulario conforme a lo deseado.
	3	Se ejecutará la acción de guardado de cambios.
Secuencia Alternativa	Paso	Acción
	1	Si no hay resultados o no se localiza el resultado esperado se deberán revisar los parámetros de búsqueda (CU-002)
Postcondición	Al guardar las modificaciones se generará un nuevo PNG y se actualizará la entrada correspondiente de la BB.DD.	
Notas		

CU-004	Crear un código QR basándose en uno existente	
Descripción	Los usuarios deben ser capaces de crear un nuevo código QR basándose en uno existente.	
Precondición	Realizar una búsqueda de códigos QR	
Secuencia Principal	Paso	Acción
	1	Se deberá poder seleccionar un QR listado, al seleccionarse se cargarán sus parámetros en el formulario correspondiente a su tipo.
	2	Se modificarán los parámetros del formulario conforme a lo deseado.
	3	Se ejecutará la acción de creación de nuevo código QR.
Secuencia Alternativa	Paso	Acción
	1	Si no hay resultados o no se localiza el resultado esperado se deberán revisar los parámetros de búsqueda (CU-002)
Postcondición	Los parámetros del código QR se guardarán en la BB.DD. y una imagen PNG del código generado se guardará en el directorio predefinido.	
Notas		

CU-005	Borrar códigos QR	
Descripción	Los usuarios deben ser capaces de borrar códigos QR existentes.	
Precondición	Realizar una búsqueda de códigos QR	
Secuencia Principal	Paso	Acción
	1	Se deberá poder seleccionar un QR listado, al seleccionarse se cargarán sus parámetros en el formulario correspondiente a su tipo.
	2	Se deberá poder realizar la acción de eliminar código QR.
	3	Se solicitará confirmación, una vez confirmado se borrará.
Secuencia Alternativa	Paso	Acción
	1	Si no hay resultados o no se localiza el resultado esperado se deberán revisar los parámetros de búsqueda (CU-002)
	1	Se podrá borrar una entrada de la lista desde la propia lista, sin necesidad de acceder al registro que se desea borrar.
Postcondición	Esta acción no es reversible	
Notas		

CU-006	Crear códigos QR empleando batch	
Descripción	Se debe poder programar la creación de códigos QR de forma que no necesite interacción con los usuarios.	
Precondición	Configurar tarea automática o script que ejecute la llamada de creación de código QR con los parámetros correspondientes.	
Secuencia Principal	Paso	Acción
	1	En función del parámetro correspondiente al tipo de QR se creará un código QR u otro
	2	Una vez creado se generará un nuevo registro en la BB.DD. y se creará un PNG con el código QR, el cual se guardara en la carpeta predefinida
Secuencia Alternativa	Paso	Acción
	1	Si el parámetro de tipo no es válido o el número de parámetros no se corresponde con el que se esperaba para el tipo indicado se cancelará la creación de código QR.
Postcondición	El código creado estará disponible para ser escaneado.	
Notas		

4.4.2 Casos de Uso de Servicios web

CU-007	Acceder a un servicio web	
Descripción	Los usuarios deben ser capaces de acceder a los servicios web.	
Precondición	Disponer de un código QR, disponer de una aplicación para el escaneo de códigos QR, disponer de conexión a internet.	
Secuencia Principal	Paso	Acción
	1	Escanear el código QR con la aplicación de escaneo.
	2	Se cargará el formulario de solicitud del servicio web correspondiente con los parámetros escaneados precargados.
Secuencia Alternativa	Paso	Acción
	2	En caso de que el código escaneado este muy dañado y no se pueda realizar una lectura correcta del mismo, el usuario deberá contactar con su equipo de soporte local.
Postcondición	El usuario podrá proceder a solicitar el servicio web cargado.	
Notas		

CU-008	Solicitud de un servicio web y recepción de respuesta	
Descripción	Los usuarios deben ser capaces de solicitar un servicio web y	
Precondición	Acceder a un servicio web	
Secuencia Principal	Paso	Acción
	1	Revisar y modificar si fuese necesario los valores de las entradas cargadas a raíz de los parámetros escaneados en el código QR
	2	Cumplimentar los campos adicionales del formulario.
	3	Realizar la acción de solicitar el servicio. Cuando se solicita el servicio se generará un registro con los parámetros de entrada de solicitud en la tabla correspondiente de la BBDD.
	4	Al conectar al servidor de Service-Now, será necesario introducir las credenciales de usuario de Service-Now
	5	El servidor gestionará el servicio requerido y devolverá el resultado al servicio web, que se lo mostrará al usuario. Adicionalmente al recibir la respuesta a la solicitud de servicio se generará un registro con los parámetros de respuesta en la tabla correspondiente de la BBDD.
Secuencia Alternativa	Paso	Acción
	3	Si hay datos sin cumplimentar la petición no será enviada y se notificará al usuario este hecho.
	5	Si se ha producido un error en la gestión debida a los datos de entrada, en lugar del resultado de la gestión se notificará el error producido al usuario (al generarse un error la gestión se interrumpe).
	5	Si se ha producido un error en la gestión debido a problema de conexión o problema de acceso a módulos en la herramienta de gestión se notificará el error producido al usuario y al administrador correspondiente.
Postcondición	El usuario dispondrá de los datos de respuesta de la ejecución del servicio web con los parámetros introducidos	
Notas		

CU-009	Acceder a registros de solicitudes y respuestas	
Descripción	Los administradores de sistemas podrán acceder a la BBDD de registros.	
Precondición	Disponer de credenciales necesarios y estar habilitado en la matriz de accesos del servidor.	
Secuencia Principal	Paso	Acción
	1	Acceder a la interfaz de visualización de registros.
	2	Realizar consultas en función de los datos deseados.
Secuencia Alternativa		
Postcondición	Se podrá acceder a los registros deseados.	
Notas	Una vez finalizadas las consultas deseadas se recomienda cerrar la aplicación.	

4.5 Diagramas de Casos de Uso

Un diagrama de caso de uso en su forma más simple es una representación de la interacción de un usuario con el sistema y representa las especificaciones de un caso de uso. Un diagrama de casos de uso puede representar los diferentes tipos de usuarios de un sistema y de las diversas formas en que interactúan con el sistema.

Vamos a emplear dos diagramas independientes para las dos partes diferenciables que componen el sistema. Uno para ilustrar los actores, casos de usos y relaciones del creador de códigos QR y otro para los servicios web.

La representación de los casos de uso en diagramas separados se realiza debido a que de este modo la representación de los diagramas se corresponde en mayor medida con la separación existente entre las partes del sistema, por lo que se considera que el resultado obtenido es más esclarecedor.

La siguiente figura ilustra el diagrama de casos de uso para el creador de códigos QR.

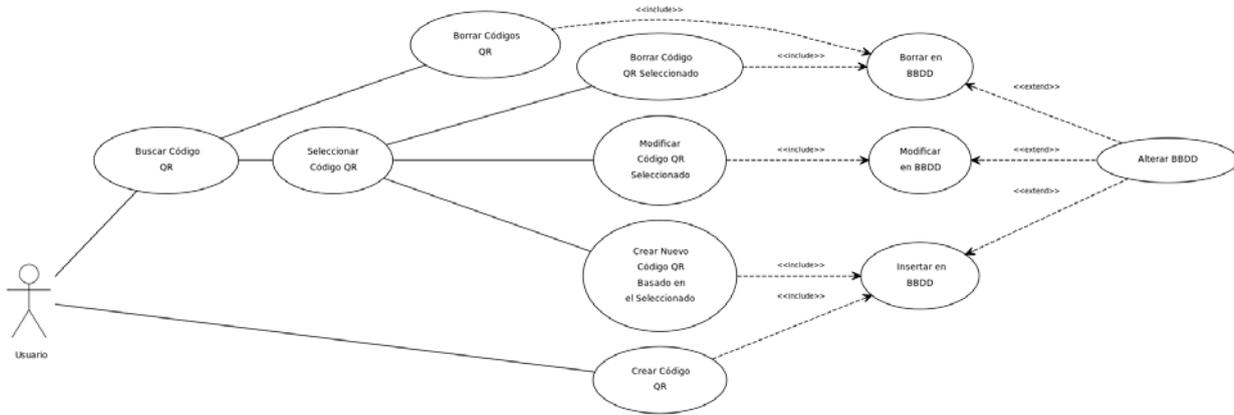


Figura 4.5.a: Diagrama de Casos de Uso del Creador de Códigos QR

A continuación se presenta una figura que ilustra el diagrama de casos de uso para los servicios web.

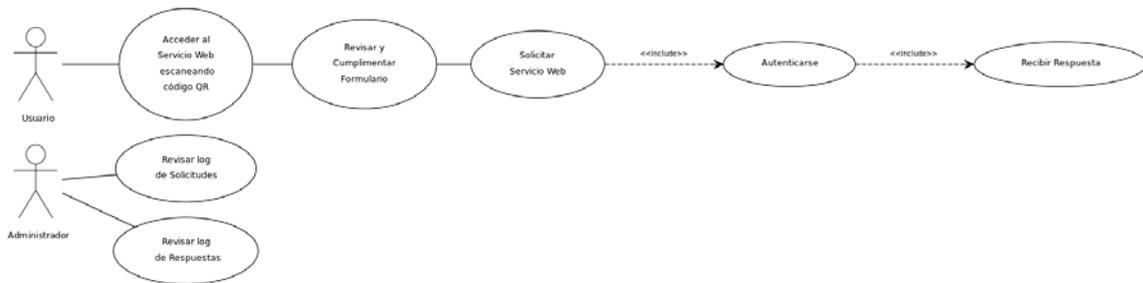


Figura 4.5.b: Diagrama de Casos de los Servicios Web

Marco Regulator y Entorno Socio-económico

En este apartado se va a realizar un pequeño inciso acerca del marco regulator y del entorno socio-económico que el sistema a desarrollar debe contemplar.

5.1 Leyes aplicables al sistema a desarrollar

Desde la publicación de la Ley Orgánica de Protección de Datos de Carácter Personal en el año 1999, hay una serie de leyes que están relacionadas con la seguridad de la información.

5.1.1 Ley Orgánica 15/99 de Protección de Datos de Carácter Personal

El objetivo de esta Ley es garantizar y proteger, en lo concerniente al tratamiento de los datos personales (automatizados o no), las libertades públicas y los derechos fundamentales de las personas físicas y, especialmente, de su honor e intimidad personal y familiar.

Según esta ley las personas de las que se almacena datos de carácter personal, tienen los siguientes derechos:

- **Derecho de información:** Según esta ley, cuando alguien proporciona sus datos si estos van a ser almacenados, debe ser informado de forma que pueda elegir si desea continuar con la acción que estaba en curso.
- **Derecho de acceso, cancelación, rectificación y oposición:** La persona puede ver la información que se dispone de él, puede cambiar esos datos para que sean correctos y exactos, cancelar la información que se almacene de él y oponerse a que se almacene.

5.1.2 Ley 34/2002 de servicios de la sociedad de la información y de comercio electrónico (LSSI)

Esta Ley se encarga de regular las obligaciones de los prestadores de servicios y los servicios que prestan. Entre las obligaciones que estipula la Ley están:

- Los prestadores de servicios deben facilitar sus datos de contacto.
- Deben colaborar con las autoridades, reteniendo los datos de conexión y tráfico durante 12 meses.

Cuando transmitan información de terceros, los proveedores de servicio no tendrán responsabilidad al respecto si:

- No modifican la información.
- Permiten el acceso a ella sólo a los destinatarios autorizados.
- Actualizan correctamente la información.
- No utilizan su posición con el fin de obtener datos sobre la utilización de la información
- Retiran la información que hayan almacenado o hacen imposible el acceso a ella, en cuanto sepan que ha sido retirada del lugar de la red en que se encontraba, o que un tribunal u órgano administrativo competente haya ordenado retirarla.

Debido a la reciente modificación de esta ley mediante el Decreto-ley 13/2012, de 30 de marzo, es necesario informar a los usuarios adecuadamente si se va a emplear archivos o programas informáticos que almacenen información de los usuarios, como por ejemplo las cookies.

5.1.3 R.D.L 1/1996 Ley de Propiedad Intelectual

La propiedad intelectual de una obra literaria, artística o científica corresponde al autor y le da la plena disposición y el derecho exclusivo a la explotación de la obra. Las obras pueden estar expresadas en cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro, lo que incluye los programas de ordenador.

5.2 Licencias comunes

La mayor parte del software desarrollado emplea librerías o aplicaciones externas para complementar o directamente poder desarrollar sus funcionalidades. Este software adicional suele estar sujeto a licencias de uso que deben respetarse.

5.2.1 Licencia MIT

Esta licencia, empleada originalmente por el Instituto Tecnológico de Massachusetts (MIT), es una de las menos restrictivas ya que permite emplear el código y los programas que emplean esta licencia sin restricciones independientemente del uso final de la aplicación y sin requerir mención alguna de la licencia o el software empleado.

Este tipo de licencia permite usar, copiar, modificar, integrar con otro software, publicar, vender o licenciar el software y además permite a las personas a las que se les facilite el software resultante hacer lo mismo.

Por estos motivos algunas veces se desaconseja su uso.

Un alto porcentaje de las gemas de Ruby emplean este tipo de licencia.

5.2.2 GNU Lesser General Public License

Esta licencia LGPL, es una versión menos restrictiva de la licencia GPL. A diferencia de la licencia GPL, que obliga a que todo software que emplee código GPL sea liberado bajo la licencia GPL, la LGPL no aplica esta restricción por lo que permite que librerías con licencia LGPL sean empleadas por programas que no necesariamente deben emplear licencias de código libre.

En cualquier caso, si se emplean librerías licenciadas bajo LGPL se debe incluir una copia del texto de la LGPL

5.3 Acciones a realizar

En consideración con las leyes y licencias listadas anteriormente, será necesario asegurar que se dispone de los siguientes mecanismos/informaciones en el sistema a desarrollar.

- Comunicar al usuario que los datos de los pedidos solicitados serán almacenados en el sistema.
- Facilitar un mecanismo mediante el que se pueda realizar la modificación o eliminación de la información almacenada.
- Facilitar información de contacto a los usuarios.
- No emplear los datos de terceros almacenados para usos diferentes de los estipulados.
- Si se emplea software licenciado bajo LGPL, incluir una notificación de que se está utilizando parte de software licenciado bajo LGPL y el texto completo de la misma.

5.4 Entorno socio-económico

En España y Europa resulta cada vez más extraño encontrar empresarios y empleados que carezcan de dispositivos móviles inteligentes. Esto no debería extrañarnos ya que el avance y la asimilación de las nuevas tecnologías por parte de la población en general es cada vez mayor.

Teniendo esto en cuenta, un sistema que emplea códigos QR y servicios web presenta una funcionalidad fácil de entender para los usuarios en general, y especialmente fácil para aquellos que emplean a diario un Smartphone. Para emplear el sistema, únicamente deben escanear el código QR del servicio deseado y cumplimentar el formulario correspondiente.

Aunque el ámbito de este sistema es inicialmente los empleados internos y externo de EDPR, el sistema podría emplearse por gran parte de la población sin que supusiese un gran desafío de formación de usuarios.

Respecto al entorno económico, las herramientas de gestión de procesos suelen tener costes relativamente elevados (precios acordes a los beneficios de emplear estas herramientas para parametrizar y organizar las diferentes gestiones de la organización), en el caso concreto de EDPR al ser una compañía energética, es prácticamente obligatorio un sistema que permita la gestión ágil y eficaz de pedidos, incidencias y correctivos, a la vez que mantiene un registro exhaustivo de las acciones realizadas de cara a las auditorías.

Al agilizar la creación y gestión de servicios mediante el uso de códigos QR se mejoran los resultados de los SLA, la calidad de la gestión y los resultados de las auditorías, aumentando la valoración del sistema por parte tanto de los usuarios como de los directivos.

Implementación

En este apartado se detallarán aspectos que se consideran relevantes de la implementación o que han supuesto un problema durante el desarrollo de la misma.

6.1 QRCreator

La aplicación para la creación de códigos QR ha sido denominada QRCreator. De acuerdo con lo especificado previamente, se ha codificado en Ruby empleando TK como GUI.

Al iniciarse la aplicación se comprueba si existe la base de datos desde la que se cargarán los parámetros de tipificaciones, como grupos y categorías, en variables auxiliares que se emplearán luego en los campos con selectores.

Si la base de datos no existe se ejecuta una tarea secundaria que la crea con las estructuras necesarias y los datos por defecto definidos para las tipificaciones.

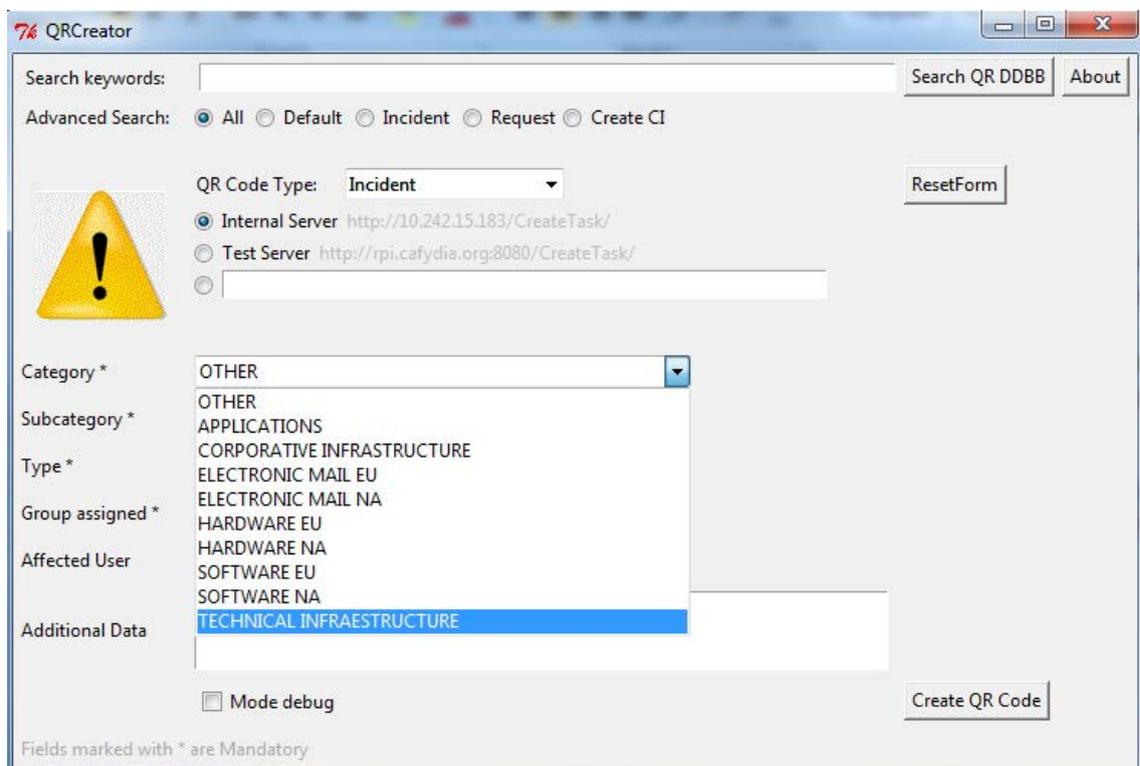


Figura 6.1.a: Selector de Categoría de desplegado

Los diferentes selectores de los tipos Incidencia y CI están relacionado, si se modifican los valores superiores se acotan las opciones disponibles en los selectores relacionados. Por ejemplo, cuando se selecciona una categoría de incidencia, las subcategorías disponibles pasan a ser las que son dependientes de la categoría seleccionada.

Los usuarios pueden tipografiar valores no predefinidos en los campos selectores. En estos casos evidentemente no existirán `sys_id` equivalentes por lo que la ruta se establecerá empleando el valor introducido por el usuario.

Para todas las tipificaciones de código QR desarrolladas se puede seleccionar el servidor que se agregará al código QR. Existen dos registros configurados por defecto (Servidor interno y servidor de pruebas) y un tercer registro en el que se puede introducir cualquier valor deseado.

Si se selecciona el tercer registro y se deja vacío el campo de texto adjunto, se creará un código QR que solo contendrá datos. La única excepción sería en el caso de estar creando un código QR de tipo 0, "Default", en el que es posible introducir la URL en el campo de datos.

Se han controlado las excepciones registradas durante las pruebas con usuarios. Para ello se han introducido los fragmentos de código que podían generar estas excepciones en sentencias try-catch (en Ruby begin-rescue), que en caso de la excepción emplean un workaround o notifican al usuario el problema surgido, indicando al usuario como solventar el problema en lugar de mostrarle la traza del error que para él sería probablemente incomprendible. Por ejemplo, si se ejecuta la aplicación sin disponer de las gemas necesarias, se indica al usuario que la ejecución no puede continuar mientras que no se instale una determinada gema.

Como se explicó en la descripción de los códigos QR que estos códigos pueden tener varias tamaños y niveles de calidad. La aplicación siempre empleará el nivel de calidad H mientras que el tamaño se ajustará en función de los datos a almacenar en el código QR.

Cada vez que se crea un código QR la aplicación genera automáticamente una descripción de los parámetros del mismo, que se mostrarán al usuario cuando realice búsquedas de códigos.

Tanto en las vistas de creación y modificación como en las ventanas de búsqueda se han agregado iconos para facilitar la identificación del tipo de Código QR.

6.2 Servicios Web

Los servicios web tipificados se han incluido en el directorio “CreateTask”. La siguiente figura muestra el contenido de una copia en local, empleada para pruebas de desarrollo, de este directorio.

```
C:\Bitnami\wampstack-5.4.25-0\apache2\htdocs\CreateTask>tree /f
Folder PATH listing for volume SYSTEM
Volume serial number is DE99-2B4A
C:
  clientCI.php
  clientIncident.php
  clientRequest.php
  debugCI.php
  debugIncident.php
  debugRequest.php
  response.php
  css
  | client.css
  images
  | logoEDPR.jpeg
  | logoEDPR.png
  lib
  | danisoap.php
```

Figura 6.2.a: Árbol de directorios de versión inicial de CreateTask

Existe un único documento CSS que contiene los estilos unificados para todos los PHP. El documento está configurado para el estilo en función del tamaño de la pantalla del dispositivo desde el que se accede.

En el directorio lib se encuentra el fichero danisoap, este documento permite realizar las diferentes acciones de la comunicación SOAP, incluyendo entre otras funcionalidades parser de XML, envío y recepción de datos mediante http y https, parser de WSDL, parser de SOAP, cliente SOAP.

Danisoap se desarrolló por necesidad ya que debido a la configuración específica que presenta Service-Now, inicialmente no lográbamos realizar una conexión SOAP correcta, por lo que no podíamos realizar pruebas de envío y recepción de datos. Antes de desarrollar este nuevo cliente SOAP se probaron los clientes SoapUI [13], nuSOAP [14] y varios clientes web SOAP [15].

Response.php se emplea para aplicar el patrón de diseño PRG, lo que impide el reenvío de formularios por parte de los usuarios.

6.3 Interfaces de Integración

Para el desarrollo de estas interfaces fue necesario realizar una investigación intensiva acerca del funcionamiento y de los campos contenidos por las diferentes tablas afectadas por los servicios web.

Los diferentes hitos de la ejecución, recepción de mensaje, gestión del servicio requerido, envío de respuesta y registro de errores que se ocasionen, se registran en el log del sistema para que quede constancia de ellos.

El hecho de que los clientes SOAP iniciales no nos funcionasen supuso un problema temporal para el desarrollo de las interfaces al no poder realizar pruebas con pasos de datos desde servicios externos.

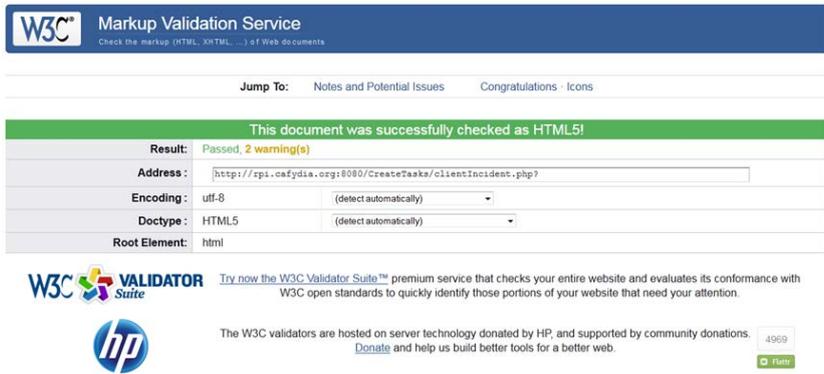
Adicionalmente a las interfaces de integración se han creado en Service-Now notificaciones de aviso a los grupos asignados cuando se crea un pedido o una incidencia empleando este sistema. Estas notificaciones son un refuerzo empleado para mejorar el tiempo de respuesta de estas solicitudes de servicios.

Pruebas

En este apartado se detallan las diferentes pruebas que se han llevado a cabo sobre el conjunto del sistema una vez que este ha sido implementado. Estas pruebas se han realizado para comprobar si el sistema cumple los requisitos y cubre las expectativas respecto a los diferentes parámetros de evaluación.

Para cada prueba realizada se especifica la funcionalidad o parte del sistema evaluada en el parámetro “sistema”.

7.1 Pruebas Funcionales

PF-001	Validar documentos HTML generados
Sistema	Servicios Web
Descripción	Someter los documentos generados al validador W3C
Criterios de completitud	Las diferentes páginas generadas deben superar la validación sin errores.
Resultado obtenido	<p>Las páginas superan la validación con éxito.</p>  <p>Figura 7.1.a: Validación HTML5 del documento clientIncident.php</p>
Consideraciones especiales	Se han validado los documentos subidos en el servidor de pruebas ya que el validador no puede acceder al servidor interno de EDPR en el que se alojan los servicios web.

<p>PF-002</p>	<p>Validar documentos CSS generados</p>
<p>Sistema</p>	<p>Servicios Web</p>
<p>Descripción</p>	<p>Someter los documentos generados al validador CSS W3C</p>
<p>Criterios de completitud</p>	<p>Las diferentes páginas generadas deben superar la validación sin errores.</p>
<p>Resultado obtenido</p>	<p>Las páginas superan la validación con éxito.</p>  <p style="text-align: center;">Figura 7.1.b: Validación CSS del documento client.css</p>
<p>Consideraciones especiales</p>	<p>Se han validado los documentos subidos en el servidor de pruebas ya que el validador no puede acceder al servidor interno de EDPR en el que se alojan los servicios web.</p>

PF-003	Validar compatibilidad de los documentos en dispositivos móviles														
Sistema	Servicios Web														
Descripción	Someter los documentos generados al validador W3C mobile														
Criterios de completitud	Las diferentes páginas generadas deben superar la validación sin errores y con alta compatibilidad.														
Resultado obtenido	<p>Las páginas superan la validación con éxito.</p>  <p>The screenshot shows the W3C mobileOK Checker interface. At the top, it says 'W3C mobileOK Checker' and 'Is your Web site mobile friendly?'. Below that, there's a browser address bar showing 'http://rpl.catydia.org:8080/CreataIncidents/client.php?'. The main content area shows a progress bar with a score of 97%. Below the progress bar, there are two tables:</p> <table border="1"> <thead> <tr> <th colspan="2">FAILURES PER SEVERITY</th> </tr> </thead> <tbody> <tr> <td>CRITICAL</td> <td>0</td> </tr> <tr> <td>SEVERE</td> <td>0</td> </tr> <tr> <td>MEDIUM</td> <td>0</td> </tr> <tr> <td>LOW</td> <td>1</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">FAILURES PER CATEGORY</th> </tr> </thead> <tbody> <tr> <td>Rely on Web standards</td> <td>1</td> </tr> </tbody> </table> <p>At the bottom, it shows 'Page Size: 5.7KB' and 'Network usage: 2 requests'.</p>	FAILURES PER SEVERITY		CRITICAL	0	SEVERE	0	MEDIUM	0	LOW	1	FAILURES PER CATEGORY		Rely on Web standards	1
FAILURES PER SEVERITY															
CRITICAL	0														
SEVERE	0														
MEDIUM	0														
LOW	1														
FAILURES PER CATEGORY															
Rely on Web standards	1														
Consideraciones especiales	Se han validado los documentos subidos en el servidor de pruebas ya que el validador no puede acceder al servidor interno de EDPR en el que se alojan los servicios web.														

Figura 7.1.c: Validación de compatibilidad móvil

PF-004	Compatibilidad con Linux
Sistema	QRCreator
Descripción	Comprobar la compatibilidad de la aplicación con distribuciones del OS Linux.
Criterios de completitud	<p>Comprobar la correcta ejecución y creación de códigos QR en Ubuntu 12.04.</p> <p>Comprobar la correcta ejecución y creación de códigos QR en Debían 7.2.</p>
Resultado obtenido	QRCreator ejecuta correctamente en entornos Ubuntu 12.04 y Debían 7.2 configurados correctamente. Para correcta configuración ver Apéndice I: Manual de Usuario QRCreator
Consideraciones especiales	No se han probado todas las distribuciones Linux pero mientras se configure correctamente el entorno debería funcionar correctamente en cualquier distribución.

PF-005	Compatibilidad con Windows
Sistema	QRCreator
Descripción	Comprobar la compatibilidad de la aplicación OS Windows
Criterios de completitud	Comprobar la correcta ejecución y creación de códigos QR en Windows 7.
Resultado obtenido	QRCreator ejecuta correctamente en entornos Windows configurados correctamente. Para correcta configuración ver Apéndice I: Manual de Usuario QRCreator
Consideraciones especiales	

PF-006	Compatibilidad con Internet Explorer
Sistema	Servicios Web
Descripción	Comprobar la compatibilidad de los servicios con el navegador.
Criterios de completitud	Comprobar la correcta carga de los servicios en Internet Explorer 10 y en Internet Explorer 11.
Resultado obtenido	Los documentos PHP de los servicios web se muestran correctamente en el navegador.
Consideraciones especiales	

PF-007	Compatibilidad con Mozilla Firefox
Sistema	Servicios Web
Descripción	Comprobar la compatibilidad de los servicios con el navegador.
Criterios de completitud	Comprobar la correcta carga de los servicios en Mozilla Firefox 30.0 y en Mozilla Firefox 31.0.
Resultado obtenido	Los documentos PHP de los servicios web se muestran correctamente en el navegador.
Consideraciones especiales	

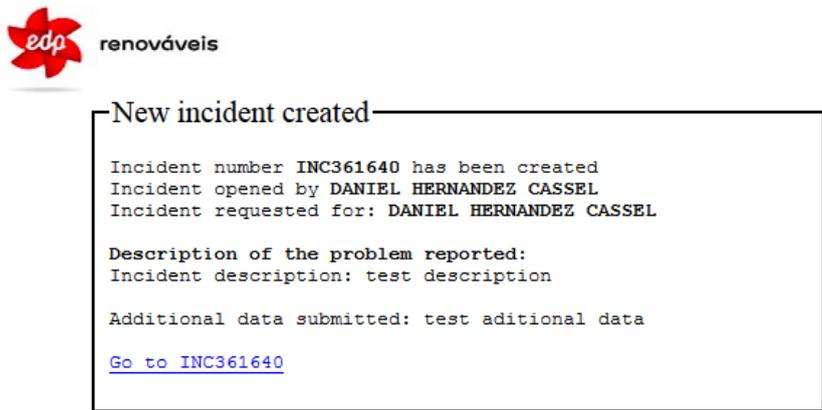
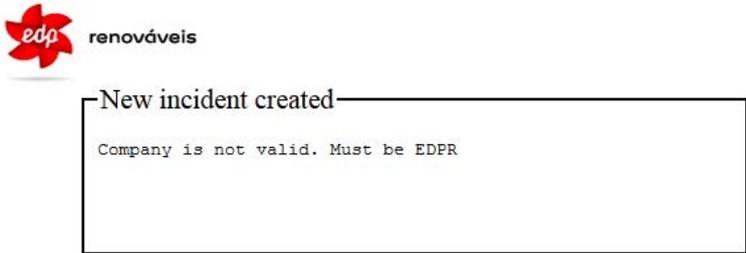
PF-008	Compatibilidad con Chrome
Sistema	Servicios Web
Descripción	Comprobar la compatibilidad de los servicios con el navegador.
Criterios de completitud	Comprobar la correcta carga de los servicios en Chrome 36.0.
Resultado obtenido	Los documentos PHP de los servicios web se muestran correctamente en el navegador.
Consideraciones especiales	

PF-009	Compatibilidad con Opera
Sistema	Servicios Web
Descripción	Comprobar la compatibilidad de los servicios con el navegador.
Criterios de completitud	Comprobar la correcta carga de los servicios en Opera 12.15.
Resultado obtenido	Los documentos PHP de los servicios web se muestran correctamente en el navegador.
Consideraciones especiales	

PF-010	Compatibilidad con Safari
Sistema	Servicios Web
Descripción	Comprobar la compatibilidad de los servicios con el navegador.
Criterios de completitud	Comprobar la correcta carga de los servicios en Safari 5.1.7.
Resultado obtenido	Los documentos PHP de los servicios web se muestran correctamente en el navegador.
Consideraciones especiales	

PF-011	Integración de los servicios, envío y recepción de solicitudes.
Sistema	Servicios Web e Interfaces de Integración
Descripción	Comprobar la correcta recepción de los mensajes SOAP enviados por los servicios web en la herramienta Service-Now.
Criterios de completitud	Comprobar la correcta recepción de los servicios en Service-Now.
Resultado obtenido	<p>La solicitud se recibe y registra correctamente en Service-Now.</p>  <p style="text-align: center;">Figura 7.1.d: Recepción de solicitud en Service-Now</p>
Consideraciones especiales	

PF-012	Integración de los servicios, gestión de las solicitudes.
Sistema	Servicios Web e Interfaces de Integración
Descripción	Comprobar la correcta gestión del servicio solicitado en el mensaje SOAP recibido.
Criterios de completitud	Comprobar la correcta gestión del servicio solicitado.
Resultado obtenido	<p>La solicitud se gestiona correctamente en Service-Now.</p>  <p style="text-align: center;">Figura 7.1.e: Incidencia creada a raíz de la gestión en Service-Now</p>
Consideraciones especiales	

PF-013	Integración de los servicios, envío y recepción de respuesta.
Sistema	Servicios Web e Interfaces de Integración
Descripción	Comprobar la correcta recepción de los mensajes SOAP enviados por la herramienta Service-Now en los servicios web.
Criterios de completitud	Comprobar la correcta recepción y visualización del mensaje.
Resultado obtenido	<p>La solicitud se gestiona correctamente en Service-Now.</p>  <p>Figura 7.1.f: Mensaje recibido tras la gestión del servicio</p>
	<p>En caso de error en los parámetros enviados que impida la gestión, el mensaje indicará el error.</p>  <p>Figura 7.1.g: Mensaje de error por compañía errónea.</p>
Consideraciones especiales	

PF-014	Creación de códigos QR empleando la interfaz gráfica
Sistema	QRCreator
Descripción	Comprobar la correcta creación de códigos QR cuando se emplea la interfaz gráfica.
Criterios de completitud	<p>El código QR generado debe ser creado en el directorio previamente definido.</p> <p>Los parámetros de creación del código QR deben ser guardados en la BBDD para futuras referencias o usos.</p> <p>Se debe generar un aviso visual confirmando la creación.</p>
Resultado obtenido	<p>Las cuatro tipificaciones disponibles generan códigos QR correctamente.</p> <p>Al crearse un código QR en la carpeta de destino, los parámetros del mismo son guardados en la DDBB.</p> <p>Una vez finaliza el proceso de creación se genera un mensaje en el que se indica que el proceso ha concluido y se especifican el nombre con el que se ha sido guardado el código QR y los parámetros del mismo.</p> <div data-bbox="606 1120 1428 1482" data-label="Image"> </div> <p style="text-align: center;">Figura 7.1.h: Notificación de creación de código QR</p>
Consideraciones especiales	

PF-015	Creación de códigos QR empleando parámetros por consola
Sistema	QRCreator
Descripción	Comprobar la correcta creación de códigos QR cuando se emplea la consola.
Criterios de completitud	Los parámetros de creación del código QR deben introducirse en el orden apropiado. El valor del parámetro tipificación debe ser de cero a tres. El número de parámetros debe coincidir con el esperado para la tipificación seleccionada.
Resultado obtenido	Se comprueba que se generan códigos QR introduciendo los parámetros correspondientes para las diferentes tipificaciones. Se comprueba que se generan códigos QR correctos mediante ejecución batch.  <pre>U:\Dropbox\Proyecto\test\test\qr>QRCreator.rb 0 "este es un ejemplo de creacion de codigo QR tipo 0" QR saved as 31_0_2014_13_45_52.png</pre> <p style="text-align: center;">Figura 7.1.i: Creación de Código QR por consola</p>
Consideraciones especiales	

PF-016	Carga de códigos QR a plantillas de tipificación
Sistema	QRCreator
Descripción	Cargar códigos QR a la tipificación correcta
Criterios de completitud	El código QR seleccionado debe cargarse en la tipificación que le corresponda. Los parámetros de creación del código QR deben ser cargados en los campos del formulario correspondientes.
Resultado obtenido	Se realizan cargas para todas las tipificaciones con diferentes valores para los diferentes parámetros. Se realizan cargas de tipos cruzados. (Cargar un tipo de código QR teniendo seleccionado un tipo diferente). Se comprueba que el parámetro de servidor de los códigos QR se carga correctamente para los diferentes servidores.
Consideraciones especiales	

PF-017	Usabilidad de QRCreator
Sistema	QRCreator
Descripción	Usuarios con y sin formación deben ser capaces de crear códigos QR funcionales
Criterios de completitud	Usuarios con formación en la aplicación debe ser capaz de emplear las diferentes funcionalidades disponibles sin complicaciones. Usuarios sin formación específica deben ser capaces al menos de crear códigos QR.
Resultado obtenido	Usuarios con formación son capaces de generar, modificar y borrar códigos QR de forma dinámica. Usuarios sin formación son capaces de generar códigos QR funcionales.
Consideraciones especiales	Los usuarios de prueba empleados son usuarios familiarizados con el uso de aplicaciones informáticas. Usuarios sin experiencia en el uso de aplicaciones para PC podría encontrar complicado su uso.

PF-018	Usabilidad de Servicios Web
Sistema	Servicios Web
Descripción	Los usuarios deben ser capaces de solicitar la gestión de servicios.
Criterios de completitud	Los usuarios deben ser capaces de solicitar la gestión de un servicio web partiendo de un código QR.
Resultado obtenido	Los usuarios con los que se realizan las pruebas son capaces de solicitar los servicios a partir de los parámetros escaneados.
Consideraciones especiales	Algunos usuarios desconocen la existencia o la utilidad de los códigos QR, una vez explicada su capacidad de almacenamiento de datos e instalada una aplicación lectora de QR se inician las pruebas.

PF-018	Múltiples Búsquedas
Sistema	QRCreator
Descripción	La aplicación debe soportar la apertura de varias ventanas de resultados de búsqueda y continuar funcionando correctamente para cualquier acción en cualquiera de las ventanas de búsqueda.
Criterios de completitud	Comprobar que se pueden mantener varias ventanas de búsqueda abiertas de forma simultánea y que esto no genera pérdida de rendimiento ni de funcionalidad.
Resultado obtenido	Se generan múltiples ventanas de búsqueda empleando diferentes criterios de búsqueda, los diferentes registros límite (primero y último) cargan correctamente independientemente de la ventana seleccionada, del orden en el que se abrieron las ventanas o del número de resultado de las diferentes ventanas de búsqueda.
Consideraciones especiales	

7.2 Pruebas de Seguridad

PS-001	SQL Injection
Sistema	Servicios Web
Descripción	Realizar inyecciones SQL para comprobar si se ejecutan en el sistema.
Criterios de completitud	<p>Introducir código SQL intruso en los diferentes parámetros de los formularios disponibles en los servicios web.</p> <ul style="list-style-type: none"> • Parámetros obligatorios y opcionales de los formularios de solicitud de servicios. • Parámetros ocultos cuando no se emplea el modo debug. • Valores introducidos en los log de solicitud y de recepción de mensaje SOAP. <p>El código SQL intruso no debe ejecutarse en ningún caso.</p>
Resultado obtenido	<p>Se realizan inyecciones de código SQL en todos los parámetros de los formularios de cargas para todas las tipificaciones, los comandos introducidos no se ejecutan.</p> <p>Se modifican los mensajes SOAP para introducir comandos de código SQLS, los comandos introducidos no se ejecutan.</p>
Consideraciones especiales	

PS-002	SQL Injection
Sistema	QRCreator
Descripción	Realizar inyecciones SQL para comprobar si se ejecutan en el sistema.
Criterios de completitud	<p>Introducir código SQL intruso en los diferentes parámetros de los formularios disponibles en los servicios web.</p> <ul style="list-style-type: none"> • Parámetros obligatorios y opcionales de los formularios de creación de códigos QR • Claves de búsqueda del buscador de códigos QR <p>El código SQL intruso no debe ejecutarse en ningún caso.</p>
Resultado obtenido	<p>Se realizan inyecciones de código SQL en todos los parámetros de los formularios de creación de códigos QR para todas las tipificaciones, los comandos introducidos no se ejecutan.</p> <p>Se realizan inyecciones de código SQL en el campo de búsqueda de códigos QR, los comandos introducidos no se ejecutan.</p>
Consideraciones especiales	

PS-003	Confidencialidad de información en DDBB
Sistema	Servicios Web
Descripción	La BBDD empleada por los servicios web debe ser accesible solo por los administradores de sistemas y los procesos de gestión de los servicios web.
Criterios de completitud	<p>Los servicios web deben poder acceder a la BBDD tanto para realizar inserciones en los logs como para consultar datos.</p> <p>Los únicos usuarios que deben poder acceder directamente a los datos almacenados son los usuarios administradores.</p>
Resultado obtenido	<p>Se comprueba que los servicios web pueden consultar datos de equivalencias y realizar inserciones en la BBDD.</p> <p>Se verifica que el servidor en el que está alojada la BBDD es un servidor interno que presenta ACL. Únicamente el grupo de usuarios administradores tiene acceso al directorio en el que se aloja la BBDD.</p>
Consideraciones especiales	

PS-004	Integridad de datos en BBDD
Sistema	QRCreator
Descripción	Comprobar la integridad de la BBDD tras acciones de inserción, actualización y borrado de datos.
Criterios de completitud	La BBDD empleada por el QRCreator debe conservar su integridad tras la creación, modificación y el borrado de códigos QR.
Resultado obtenido	Se verifica la integridad de la BBDD después de realizar las siguientes acciones: <ul style="list-style-type: none"> • Creación e inserción de códigos QR empleando la interfaz de usuario. • Creación e inserción de códigos QR empleando la ejecución por consola. • Modificación y actualización de códigos QR. • Borrado de códigos QR existentes.
Consideraciones especiales	

PS-005	Integridad de datos en BBDD
Sistema	Servicios Web
Descripción	Comprobar la integridad de la BBDD tras acciones de inserción, actualización y borrado de datos.
Criterios de completitud	La BBDD empleada por los servicios web debe conservar su integridad tras la creación, modificación y el borrado de códigos QR.
Resultado obtenido	Se verifica la integridad de la BBDD después de realizar las siguientes acciones: <ul style="list-style-type: none"> • Creación e inserción de entradas de registros de solicitudes de servicios web. • Creación e inserción de entradas de registros de respuestas de servicios web.
Consideraciones especiales	

PS-006	Integridad de datos en BBDD
Sistema	Servicios Web
Descripción	Los datos almacenados en las BBDD solo deben ser modificables por los procesos del sistema y los administradores.
Criterios de completitud	La integridad de la BBDD no debe verse afectada por modificaciones de usuarios sin permisos.
Resultado obtenido	Se verifica que los usuarios no incluidos en el grupo de administradores de sistemas no pueden acceder ni listar la BBDD aun cuando conozcan su ubicación en los directorios.
Consideraciones especiales	

PS-007	Autenticación de usuarios
Sistema	Servicios Web e Interfaces de Integración
Descripción	Confirmar la obligatoriedad de autenticarse en la herramienta de gestión para realizar la solicitud de servicio
Criterios de completitud	Confirmar que para solicitar un servicio web hay que introducir credenciales de usuario. Comprobar que la gestión no se realiza si los credenciales no son correctos.
Resultado obtenido	<p>Al realizar la solicitud de cualquiera de los servicios web se requieren credenciales de usuario. Si los credenciales no son correctos el sistema muestra un error.</p>  <p>Figura 7.2.a: Solicitud de credenciales y mensaje de error</p>
Consideraciones especiales	

Conclusiones y Futuros Trabajos

8.1 Conclusiones

Tras finalizar el desarrollo objeto del presente proyecto de fin de carrera, es hora de valorar resultados y sacar conclusiones. Tal y como se recoge en el apartado inicial de este documento, el objetivo fundamental es desarrollar un sistema de gestión de servicios de IT empleando códigos QR, mediante el cual agilizar la solicitud y gestión de los servicios IT.

Se trata de un sistema multiplataforma, cuyas interfaces gráficas se adaptan al estilo del SO en el que se ejecute gracias al uso de librerías estandarizadas. De esta forma mientras que el QRCreator parece una funcionalidad más del sistema, los servicios web muestran una interfaz estándar para las diferentes combinaciones de SO y dispositivos que acceden a ellos.

Mediante las aplicaciones existentes de creación de códigos QR es posible crear códigos QR funcionales, sin embargo, emplear estas herramientas podría poner en peligro la confidencialidad de la información, a la vez que limitaría el grado de personalización. Gracias a la creación de un creador de códigos QR propio, podemos generar códigos funcionales para varias configuraciones de datos, con la tranquilidad de que la integridad, disponibilidad y confidencialidad de la información estén contempladas.

Además, para la creación de códigos QR de forma sistemática, la posibilidad de crearlos empleando la generación batch es indispensable. Esta capacidad también permite la generación automática de códigos QR sin necesidad de la intervención de ninguna persona.

Existen múltiples clientes SOAP disponibles que permiten la generación envío y recepción de mensajes, aun así la práctica totalidad de estos clientes únicamente pueden emplear la configuración estándar. Gracias a la creación de DaniSOAP disponemos de una librería PHP, que además de permitir intercambio de mensajes SOAP estándar, permite establecer conexiones SOAP con la herramienta Service-Now.

Por su parte, los servicios web y las interfaces de integración desarrolladas permiten la solicitud de gestiones de IT de forma sencilla y eficaz, a partir de los parámetros escaneados de los códigos QR.

Gracias a este sistema en conjunto los usuarios pueden solicitar servicios de IT desde sus dispositivos móviles simplemente escaneando un código QR, confirmando parámetros e introduciendo sus credenciales. En las solicitudes con más datos, el proceso completo de solicitud emplea, de media, menos de un minuto, lo que supone un gran ahorro de tiempo para los usuarios que además accederán mediante una interfaz adaptada a sus dispositivos móviles.

Una ventaja igualmente importante a considerar es que al ser todos los componentes del sistema parte del desarrollo propio, podemos adaptarlo según nuevas necesidades que pudieran surgir (nuevas tipificaciones, nuevos elementos, etc.) sin depender de los diferentes soportes que los otros productos aportan y que pueden no ser ágiles en este sentido.

La mejor forma de ver si se han cubierto estas necesidades y objetivos marcados, es analizarlos resumiendo los requisitos en una tabla.

	Completitud del Requisito
Configuración de tipificaciones de Códigos QR	
Creación de Códigos QR mediante Interfaz Grafica	
Creación de Códigos QR mediante paso de parámetros por consola.	
Guardado de Códigos QR	
Carga de Códigos QR	
Modificación y borrado de códigos QR	
Carga de parámetros en formulario de servicio web	
Solicitud de servicio web	
Gestión de servicio solicitado	
Registro de mensajes de solicitud y respuesta	
Generación de log de gestión	
Parser de datos XML	

Autenticación de usuarios	
User Friendly	
Disponibilidad de documentación	
Disponibilidad del sistema	
Integridad y confidencialidad de las BBDD	
Multiplataforma	

Figura 8.1.a: Completitud de requisitos

Por tanto, podemos afirmar que el sistema desarrollado cubre con éxito las expectativas marcadas inicialmente.

8.2 Trabajos Futuros

Cuando hablamos de desarrollos informáticos, la práctica totalidad es mejorable. El resultado obtenido al finalizar este proyecto no deja de ser una versión 1.0 que puede tener una evolución con objeto de ampliar su capacidad y cubrir aún más necesidades.

A continuación se recogen diferentes aspectos que en esta primera versión quizás no se hayan cubierto o puedan ser potenciados y ampliados en futuras versiones:

- **Gestión de Seguridad más potente:** El acceso a los diferentes códigos QR creados por parte de los usuarios de la aplicación podría estar restringido a determinados usuarios o grupos de usuarios. Actualmente los parámetros de los códigos QR creados son accesibles para todos los usuarios independientemente del usuario que los crease. Se podrían implementar reglas de control de acceso por grupos de forma que los usuarios solo puedan acceder a los códigos QR creados o asignados a los grupos a los que pertenezcan.

- **Integración de la BBDD de los servicios web y del QRCreator con la herramienta de gestión:** Se podría ampliar la capacidad de actualización del sistema integrando las BBDD empleadas por el sistema desarrollado con Service-Now, de manera que pudiera configurarse una actualización periódica de los parámetros de las tipificaciones.
Los catálogos de servicios cambian pero los cambios son pocos y controlados; sin embargo una automatización del proceso de actualización permitiría asegurar que los datos estén actualizados.

- **Acceso desde aplicaciones nativas de los dispositivos móviles:** Una evolución del sistema desarrollado sería generar para cada plataforma de dispositivo móvil una aplicación nativa que permita el escaneo de códigos QR y en base a los datos escaneados, cargue un formulario en la propia aplicación que luego pueda enviar a la herramienta de gestión de servicios de IT correspondiente.
De esta forma se podrían actualizar las tipificaciones existentes actualizando la aplicación del dispositivo móvil.
Esta evolución del sistema presenta el inconveniente de que sería necesario gestionar y mantener una aplicación por cada SO de dispositivo móvil empleado por los usuarios.

- **Creación de ejecutables de instalación:** Aunque el sistema desarrollado presenta instrucciones de instalación y configuración de los requisitos necesarios, sería interesante generar archivos ejecutables que dejaran el sistema completamente operativo.
Sería especialmente interesante desarrollarlo para la aplicación QRCreator, para que se instalasen y configurasen de forma consecutiva, Ruby y las gemas necesarias para el funcionamiento de la aplicación QRCreator.

- **Multi-idioma:** Los sistemas desarrollados están configurados todos para la lengua anglosajona, al ser este el lenguaje estándar empleado en su entorno de uso.
Podría incluirse la capacidad multi-idioma de manera que mediante tablas de equivalencias se pudieran presentar los textos en diferentes idiomas mediante un selector.
Adicionalmente se podría agregar como parámetro en el QRCreator de forma que al acceder a un servicio web se cargue directamente el idioma deseado.

- **Administración Web:** Los servicios web presentan el conjunto de ventanas necesarias para que los usuarios puedan gestionar los servicios deseados. No obstante, las personas responsables de revisar los registros de solicitudes y mensajes de respuesta deben emplear el propio cliente de BBDD para consultar los registros necesarios en las tablas de la BBDD. Hoy en día es cierto que los clientes de acceso a BBDD son aplicaciones muy cómodas y amigables, pero quizás sería más cómodo e intuitivo disponer de un conjunto de páginas web con las que los responsables pudieran gestionar estas tablas, realizar exportaciones de datos, etc.

- **Tipos adicionales de Códigos QR:** Desarrollar la capacidad de crear códigos QR que permitan crear correctivos, evolutivos, cambios operacionales o actualizar registros existentes de la CMDB (actualmente solo se pueden crear nuevos registros).
Desarrollar nuevo tipos de códigos QR implica realizar análisis de los parámetros necesarios de la tabla referenciado por el tipo en cuestión, modificar la BBDD, desarrollar los servicios web y de integración y generar la plantilla de creación del nuevo tipo de códigos QR.

- **Agregar preferencias de búsqueda:** Desarrollar para el creador de códigos QR un pequeño menú que permita al usuario modificar parámetros de las ventanas de búsqueda:
 - Seleccionar el número de registros que desea que se le muestren por página.
 - Elegir los parámetros por los cuales se ordenan los resultados.
 - Seleccionar ordenación descendente o ascendente.

Índice de Figuras y Tablas

9.1 Índice

Figura 1.1.a: Análisis de las características de los sistemas	12
Figura 1.2.a: Interacción entre partes diferenciadas del sistema.....	16
Figura 2.1.a: Gartner Magic Quadrant for IT Service Support Management Tools 2014	24
Figura 2.1.b: Análisis de la comparación de herramientas	25
Figura 2.2.a: Estructura de código QR	26
Figura 2.2.b: Capacidad de QR según tipo de dato.....	27
Figura 2.2.c: Información de formato.....	28
Figura 2.2.d: Ordenación de las palabras código.....	29
Figura 2.2.e: Aplicaciones lectoras de QR	30
Figura 3.1.a: Tipificaciones de códigos QR a implementar.....	32
Figura 3.1.b: Tipificaciones de servicios web a implementar.....	38
Figura 3.1.c: Versión inicial de documento WSDL.....	45
Figura 3.2.a: Ejemplo notificación de aviso	49
Figura 4.1.a: Esquema del Creador de Códigos QR.....	56
Figura 4.1.b: Ejemplo de parámetros obligatorios para tipificación Create CI.....	57
Figura 4.1.c: Esquema de los Servicios Web	58
Figura 4.2.a: Comparación de características de lenguajes POO.....	61
Figura 4.2.b: Comparación de velocidad y rendimiento.....	62
Figura 4.2.c: Comparativa de Ruby GUI toolkits.....	63
Figura 4.2.d: Esquema de patrón de diseño PRG	66
Figura 4.3.a: Modelo Físico/Lógico de la BBDD del creador de códigos QR.....	68
Figura 4.3.b: Descripción tabla ERRORLOG	68
Figura 4.3.c: Descripción tabla QRCODES	69
Figura 4.3.d: Descripción genérica de las tablas de equivalencia.	70
Figura 4.3.e: Modelo Físico/Lógico de la BBDD de los servicios web.....	70
Figura 4.3.f: Descripción tabla REQUESTLOG	71
Figura 4.3.g: Descripción tabla RESPONSELOG.....	71
Figura 4.3.h: Descripción tabla ERRORLOG	71
Figura 4.5.a: Diagrama de Casos de Uso del Creador de Códigos QR.....	79
Figura 4.5.b: Diagrama de Casos de los Servicios Web	79
Figura 6.1.a: Selector de Categoría de desplegado	85
Figura 6.2.a: Árbol de directorios de versión inicial de CreateTask.....	87
Figura 7.1.a: Validación HTML5 del documento clientIncident.php.....	89
Figura 7.1.b: Validación CSS del documento client.css.....	90

Figura 7.1.c: Validación de compatibilidad móvil	91
Figura 7.1.d: Recepción de solicitud en Service-Now	94
Figura 7.1.e: Incidencia creada a raíz de la gestión en Service-Now	94
Figura 7.1.f: Mensaje recibido tras la gestión del servicio	95
Figura 7.1.g: Mensaje de error por compañía errónea	95
Figura 7.1.h: Notificación de creación de código QR	96
Figura 7.1.i: Creación de Código QR por consola	97
Figura 7.2.a: Solicitud de credenciales y mensaje de error	103
Figura 8.1.a: Completitud de requisitos	106
Figura 11.2.a: Estado inicial al seleccionar la tipificación Incident	119
Figura 11.2.b: Resultados de búsqueda empleando palabra clave “test”	120
Figura 11.2.c: Borrado de Códigos QR	121
Figura 11.2.d: Formulario de actualización de código QR	122
Figura 11.2.e: Mensaje de aviso de búsqueda sin resultados	122
Figura 12.2.a: Formulario con parámetros para la creación de incidencias, modo estándar	125
Figura 12.2.b: Formulario con parámetros para la creación de incidencias, modo debug	126
Figura 12.2.c: Formulario para la inserción de credenciales	127
Figura 12.2.d: Resumen de la gestión realizada, modo estándar	127
Figura 12.2.e: Resumen de la gestión realizada, modo debug	128
Figura 12.3.a: Notificación para campos vacíos y no válidos	129
Figura 12.3.b: Notificación de compañía no válida	129
Figura 12.3.c: Notificación de usuario no valido en solicitud pedido	130
Figura 12.3.d: Error de autenticación y errores de formato en el mensaje SOAP	131

Bibliografía

10.1 Referencias

Conjunto de documentación referenciado en el cuerpo de la memoria. Estas referencias se encuentran ordenadas en consonancia con su orden de aparición en la memoria.

[1] Service-now. Service-now, 2014 [consulta: 10-09-2014]. Disponible en: <http://www.servicenow.com>.

[2] Service-now. List of Available Integrations [en línea], 31-07-2014 [consulta: 10-09-2014]. Disponible en: [http://wiki.servicenow.com/index.php?title=List of Available Integrations](http://wiki.servicenow.com/index.php?title=List_of_Available_Integrations)

[3] Service-now. List of Plugins [en línea], 08-04-2014 [consulta: 10-09-2014]. Disponible en: [http://wiki.servicenow.com/index.php?title=List of Plugins](http://wiki.servicenow.com/index.php?title=List_of_Plugins)

[4] DENSO WAVE INCORPORATED. QR Code [consulta: 03-04-2014]. Disponible en: <http://www.qrcode.com/en/>

[5] Ruby community. Ruby [consulta: 03-02-2014]. Disponible en: <https://www.ruby-lang.org/en/>

[6] GREENE, J., BROOKS, J. Magic Quadrant for IT Service Support Management Tools. Cherwell [blog]. 25 de agosto de 2014 [consulta: 10-09-2014]. Disponible en: <http://www.gartner.com/technology/reprints.do?id=1-20BY8WJ&ct=140825&st=sb>

[7] DENSO WAVE INCORPORATED. Solutions & Case Studies: Data entry: Hiroshima Prefecture Association for Group Medical Checkup [en línea], [consulta: 05-04-2014] Disponible en: <http://www.denso-wave.com/en/solution/typeofuse/data-entry.html>. [img/usr_page_files/en_solution_usage_data-entry_profile_kenko.pdf](#)

[8] DroidLa. Google play: QR Droid Code Scanner [en línea]. 18-08-2014 [consulta: 25-08-2014]. Disponible en:

<https://play.google.com/store/apps/details?id=la.droid.qr&hl=en>

[9] Wikipedia. Alta disponibilidad: Cálculos porcentuales [en línea]. 13-08-2014 [consulta: 19-08-2014]. Disponible en:

http://es.wikipedia.org/wiki/Alta_disponibilidad#C.C3.A1culos_porcentuales

[10] ONLYJOB. Arena. Raid6 [blog]. 22-03-2012 [consulta: 20-08-2014] Disponible en: <http://raid6.com.au/~onlyjob/posts/arena/>

[11] VENNERS, Bill. The Philosophy of Ruby: A Conversation with Yukihiro Matsumoto, Part 1. Artima [blog] 29-09-2003 [Consultado 19-08-2014]. Disponible en: <http://www.artima.com/intv/ruby4.html>

[12] ROSEMAN, Mark. Tkdocs, 2014 [consulta: 05-09-2014]. Disponible en: <http://www.tkdocs.com/index.html>

[13] SmartBear Software. SoapUI, 2014 [consulta: 11-06-2014]. Disponible en: <http://www.soapui.org/>

[14] Dice Holdings. Sourceforge: NuSOAP - SOAP Toolkit for PHP. 06-06-2013 [consulta: 12-06-2014]. Disponible en:

<http://sourceforge.net/projects/nussoap/>

[15] WSDLBroser. Web based SOAP client, 2013 [consulta: 12-06-2014]. Disponible en: <http://wsdlbrowser.com/>

10.2 Bibliografía Consultada

Conjunto de referencias y documentación consultados durante el desarrollo del sistema. Estas referencias se encuentran organizadas por temario, tipo de recurso y orden alfabético.

Diseño

- GAMMA, Erich. HELM, Richard. JOHNSON, Ralph. VLISSIDES, John M. Design Patterns: Elements of Reusable Object-Oriented Software. Primera edición. Prentice Hall, 17-01-1980. 395 p. ISBN: 0201633612.
- OLSEN, Russ. Design Patterns in Ruby [en línea]. Addison-Wesley Professional, 10-12-2007 [consulta: 10-07-2014]. 384 p. ISBN: 0321490452. Disponible en:
<http://proquest.safaribooksonline.com/strauss.uc3m.es:8080/book/web-development/ruby/9780321490452>

PHP:

- REIERSOL, Dagfinn. BAKER, Marcus. SHIFLETT, Chris. PHP in Action: Objects, Design, Agility [en línea]. Primera edición. Manning Publications, 10-07-2007 [consulta: 25-07-2014]. 552 p. ISBN: 1932394757. Disponible en:
<http://proquest.safaribooksonline.com/strauss.uc3m.es:8080/book/programming/php/9781932394757>
- PHP Group, PHP, 2014 [consulta: 23-07-2014]. Disponible en:
<http://php.net/>
- Stackoverflow, Stackoverflow: PHP [en línea], 2014 [consulta: 24-08-2014]. Disponible en: <http://stackoverflow.com/tags/php>

QR:

- DENSO WAVE INCORPORATED. QR Code: What is a QR Code [en línea], 2014 [consulta: 06-09-2014]. Disponible en:
<http://www.qrcode.com/en/about/>
- DENSO WAVE INCORPORATED. QR Code: History [en línea], 2014 [consulta: 06-09-2014]. Disponible en:
<http://www.qrcode.com/en/history/>

Ruby

- BROWN, Gregory T. Ruby Best Practices [en línea]. Primera edición. O'Reilly Media, 16-06-2009 [consulta: 12-08-2014]. 336 p. ISBN: 0596523009. Disponible en: <http://proquest.safaribooksonline.com.strauss.uc3m.es:8080/book/web-development/ruby/9780596157487>
- CARLSON, Lucas. RICHARDSON, Leonard. Ruby Cookbook [en línea]. Primera edición. O'Reilly Media, 19-07-2006. Serie: Cookbooks [consulta: 03-06-2014]. 910 p. ISBN: 0596523696. Disponible en: <http://proquest.safaribooksonline.com.strauss.uc3m.es:8080/book/web-development/ruby/0596523696>
- COLLINGBOURNE, Huw. The Book of Ruby [en línea]. Primera edición. No Starch Press, 06-07-2011 [cConsulta: 11-08-2014]. 400 p. ISBN: 1593272944. Disponible en: <http://proquest.safaribooksonline.com.strauss.uc3m.es:8080/book/programming/ruby/9781593272944>
- THOMAS, David. Programming Ruby: the pragmatic programmers' guide. Segunda edición. Raleigh (North Carolina); Dallas (Texas): The pragmatic bookshelf, 2006 [consulta: 22-04-2014]. 830 p. ISBN: 0974514055. Disponible en: UC3M, Fondo especializado, biblioteca Campus de Colmenarejo.
- Ruby community. Ruby documentation, 2014 [consulta: 13-09-2014]. <http://ruby-doc.com>
- Ruby community. Ruby Central. Ruby Gems: your community gem host, 2014. Hosted: GitHub [consulta: 14-09-2014]. Disponible en: <https://rubygems.org>
- Stackoverflow, Stackoverflow: Ruby [en línea], 2014 [consulta: 06-09-2014]. Disponible en: <http://stackoverflow.com/tags/ruby/>

SQLite

- NEWMAN, Chris. Chapter 5. The PHP Interface. En: SQLite [en línea]. Sams, 09-11-2004. 336 p. ISBN: 067232685X. Disponible en:
<http://proquest.safaribooksonline.com/strauss.uc3m.es:8080/book/databases/sql/067232685x/using-sqlite-programming-interfaces/ch05>
- BUCK, Jamis. LAVENA, Luis. PATTERSON Aaron. Ruby gems: sqlite3 [en línea]. Version 1.3.9. 20-02-2014 [consulta: 04-05-2014] Disponible en: <https://rubygems.org/gems/sqlite3>
- SQLite Consortium. SQLite Home Page, 2014 [consulta: 27-07-2014]. Disponible en <http://www.sqlite.org/>

10.3 Herramientas Referenciadas

Software de gestión de TI

- Axios Systems. Axios Systems: Assyst, 2014 [consulta: 10-09-2014]. Disponible en:
<http://www.axiossystems.com/en/home.html>
- BMC Software. BMC Software: BMC Remedy, 2014 [consulta: 10-09-2014]. Disponible en:
<http://www.bmc.com/it-solutions/remedy-itsm.html>
- CA Technologies. CA Technologies: Service Desk Manager, 2014 [consulta: 10-09-2014]. Disponible en:
<http://www.ca.com/es/intellicenter/ca-service-desk-manager.aspx>
- Cherwell Software. Cherwell Software: Cherwell Service Management, 2014 [consulta: 10-09-2014]. Disponible en:
<http://www.cherwell.com/product/cherwell-service-management>
- EasyVista. EasyVista, 2014 [consulta: 10-09-2014]. Disponible en:
<http://www.easyvista.com/en/>
- FrontRange. FrontRange: Heat Service Management, 2014 [consulta: 10-09-2014]. Disponible en:
<http://www.frontrange.com/heat/products/service-management>
- Hewlett-Packard. HP: Service Manager, 2014 [consulta: 10-09-2014]. Disponible en: <http://www8.hp.com/us/en/software-solutions/service-manager-service-desk/>
- IBM. IBM: Smartcloud Control Desk, 2014 [consulta: 10-09-2014]. Disponible en:
- Service-now. Service-now, 2014 [consulta: 10-09-2014]. Disponible en: <http://www.servicenow.com>.
www.ibm.com/software/products/es/smartcloudcontroldesk

Ruby GUI Toolkit

- KANIS, Lars. JOHNSON, Lyte. FXRuby, 2014 [consulta: 15-08-2014]. Disponible en: <https://rubygems.org/gems/fxruby>
- Team Shoes. Shoes, 2014 [consulta: 15-08-2014]. Disponible en:
<http://shoesrb.com/>
- Tcl Developer Xchange. TK, 2014 [consulta: 15-08-2014]. Disponible en: <http://www.tcl.tk/>

Apéndice I Manual de Usuario QRCreator

En este apartado se adjunta documentación adicional desarrollada para la instalación, configuración y correcta ejecución de la aplicación QRCreator.

11.1 Instalación y configuración Ruby

El objetivo de este manual es indicar los pasos necesarios para la instalación y la correcta configuración del entorno del usuario para el uso de la aplicación QRCreator desarrollada en Ruby.

11.1.1 Instalación de Ruby en Linux

Actualmente la mayoría de las distribuciones Linux ya traen por defecto instalado Ruby junto con el interpretador de consola IRB.

Para verificar si ya dispone de Ruby abra una consola de comandos y ejecute el comando **which ruby**.

Si el comando devuelve un path, es que Ruby se encuentra instalado en el sistema, si por el contrario el comando devuelve un mensaje de error, significa que no dispone de Ruby en el sistema.

En caso de que devuelva path puede consultar la versión actual que tiene instalada con el comando **ruby -v**. Puede comparar la versión que tiene instalada con la disponible en la página de descargas de Ruby ^[1], si la versión que tiene instalada es muy antigua podría ser interesante actualizarla para garantizar la funcionalidad de todas las características.

El comando de instalación varía en función de las distribuciones ^[2], para Ubuntu y debían se puede emplear **sudo apt-get install ruby**

11.1.2 Instalación de Ruby en Windows

Windows no trae Ruby instalado por defecto, en cualquier caso se puede verificar rápidamente si ya se instaló Ruby en el pasado ejecutando en la consola el comando **ruby -v**. Si la ejecución devuelve una versión, es que Ruby se encuentra actualmente instalado.

En caso de que no disponga de Ruby, para Windows existe un instalador RubyInstaller ^[3] que únicamente es necesario descargar, aceptar la licencia, marcar todas las opciones adicionales y ejecutar.

11.1.3 Gemas Necesarias

Para que QRCreator pueda ejecutarse requiere que determinadas gemas se encuentren instaladas. Los comandos a ejecutar para instalar las gemas necesarias son:

```
gem install rqrcode-with-patches
```

```
gem install sqlite3
```

Adicionalmente es necesario instalar una librería para habilitar Tk. En caso de haber empleado el instalador de Windows ya lo tendrás instalado, por el contrario si estas empleando Linux lo puedes instalar con el comando:

```
sudo apt-get install libtcltk-ruby
```

Una vez finalizadas estas instalaciones el entorno de ejecución se encuentra listo para ejecutar QRCreator.

11.2 QRCreator

En este apartado se listan las distintas funcionalidades incluidas en la aplicación QRCreator y se describe como emplearlas.

En la siguiente figura se describen las diferentes funcionalidades disponibles en el estado inicial de la aplicación.

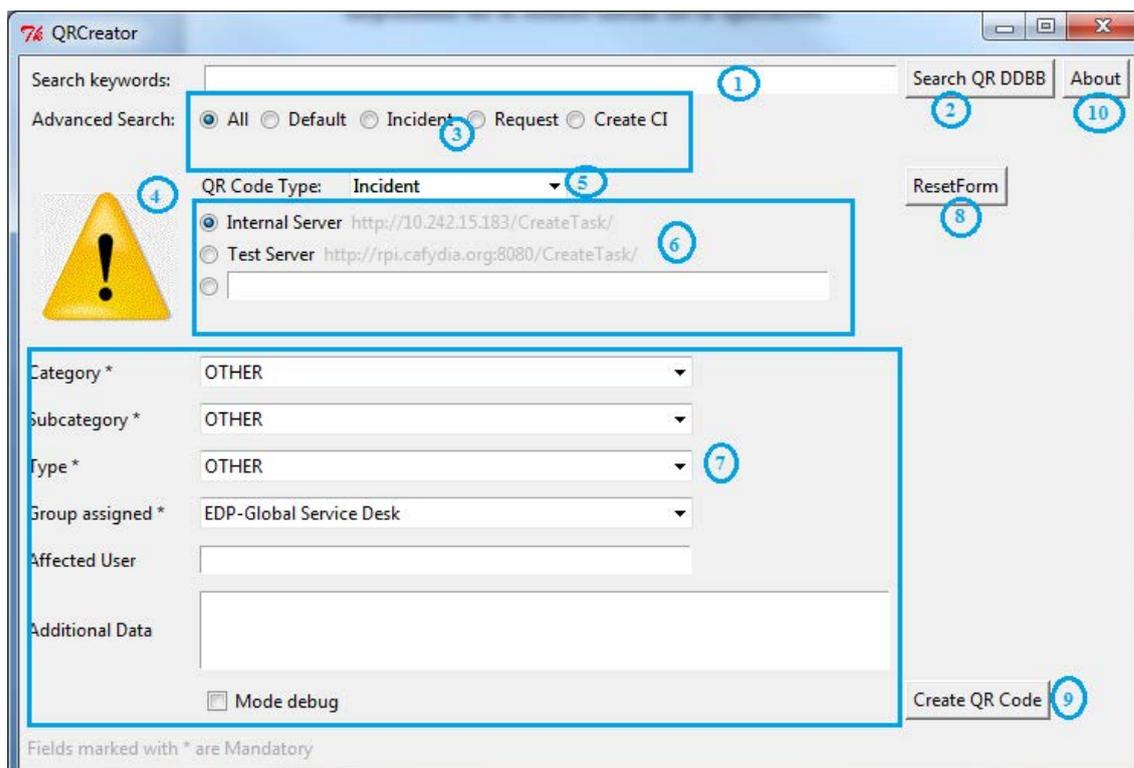


Figura 11.2.a: Estado inicial al seleccionar la tipificación Incident

- 1) Barra de búsqueda, permite buscar por la cadena de texto introducida en la BBDD de códigos QR creados. Si se deja vacío se listan todos los códigos correspondientes al tipo seleccionado en búsqueda avanzada.
- 2) Botón para realizar la búsqueda en la BBDD en función de los parámetros de búsqueda introducidos (1) y (3).
- 3) Búsqueda avanzada, permite delimitar la tipificación de códigos QR sobre la que realizar una búsqueda.
- 4) Icono característico de la tipificación seleccionada.
- 5) Permite seleccionar el tipo de código QR que se desea crear
- 6) Permite seleccionar entre los diferentes servidores previamente configurados o introducir un nuevo servidor. Para el tipo Default no se incluye este selector.
- 7) Parámetros propios de la tipificación de código QR seleccionada en el selector de tipificación (5).
- 8) Botón para restaurar el formulario a los valores iniciales.
- 9) Botón que permite la creación del código QR con los parámetros introducidos. Si algún parámetro obligatorio no se encuentra cumplimentado se generará un mensaje de error.
- 10) Botón para acceder a información sobre el QRCreator.

Por cada código QR listado se generan dos botones, uno para borrar el código QR y otro para cargar sus parámetros en el formulario de su tipo correspondiente. Si se pulsa el botón de borrado, tras la confirmación los datos almacenados se borrarán, se marcará en rojo su descripción y se deshabilitarán sus botones.

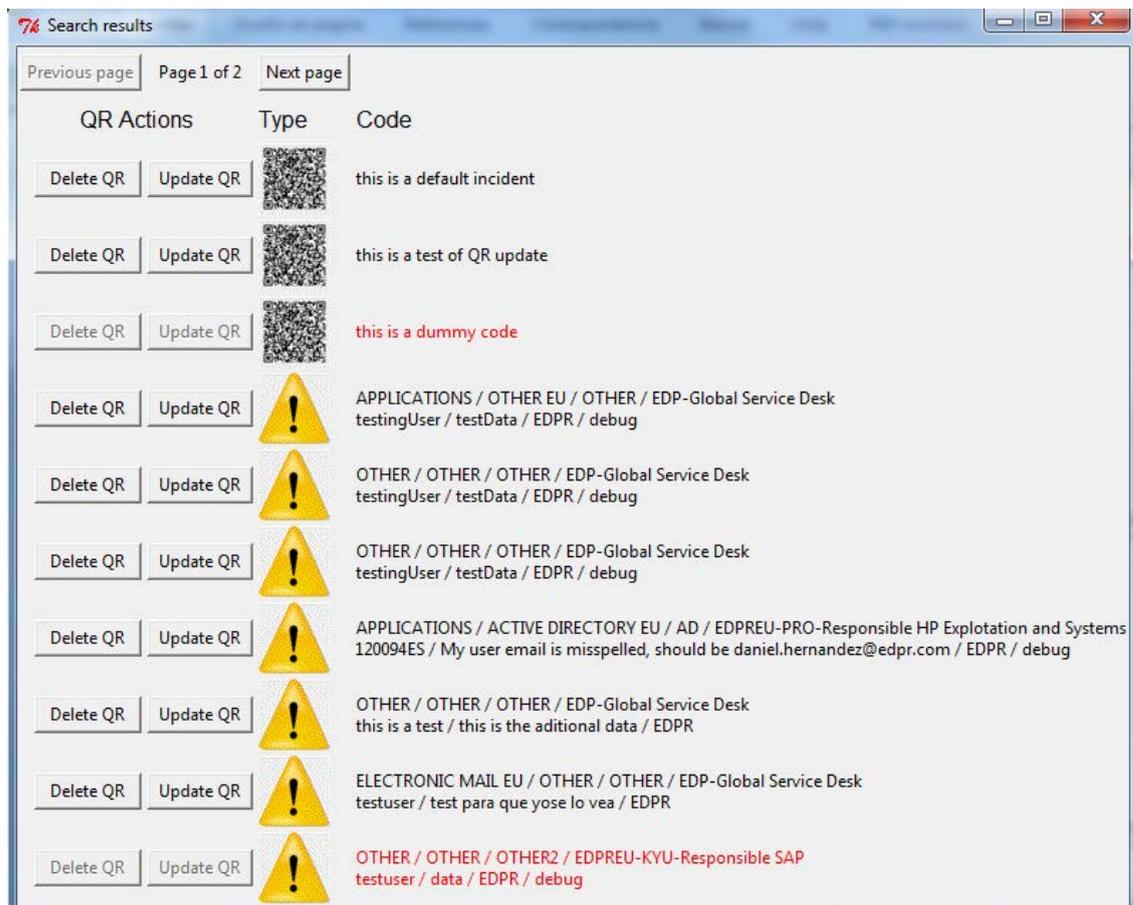


Figura 11.2.c: Borrado de Códigos QR.

Si por el contrario se pulsa el botón de actualización de código QR, se cargarán los parámetros del código QR en el formulario que corresponda y se habilitarán botones para la actualización, borrado o creación de un nuevo código QR.

The screenshot shows the QRCreator application window. At the top, there is a search bar with the text "Search keywords:" and a "Search QR DDBB" button. Below it, there are radio buttons for "Advanced Search" with options: All (selected), Default, Incident, Request, and Create CI. A yellow warning triangle icon is visible on the left. The "QR Code Type" is set to "Incident" in a dropdown menu, with a "ResetForm" button next to it. Below this, there are radio buttons for "Internal Server" (selected) and "Test Server", each with a corresponding URL. There are also empty radio buttons. The form includes several mandatory fields marked with an asterisk: "Category *", "Subcategory *", "Type *", and "Group assigned *", all set to "OTHER". The "Affected User" field contains "this is a test". The "Additional Data" field contains "this is the additional data". There is a "Mode debug" checkbox which is unchecked. On the right side, there are buttons for "Update QR Code", "Delete QR Code", and "Create QR Code". At the bottom left, a note says "Fields marked with * are Mandatory".

Figura 11.2.d: Formulario de actualización de código QR

En caso de que los parámetros de búsqueda introducidos no generen resultados, se mostrará un mensaje de ayuda indicándolo.

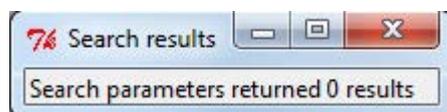


Figura 11.2.e: Mensaje de aviso de búsqueda sin resultados

11.3 Referencias Externas

[1] Ruby community. Ruby: Downloads, 2014 [consulta: 16-08-2014]. Disponible en: <https://www.ruby-lang.org/en/downloads/>

[2] Ruby community. Ruby: Installations: Package management systems, 2014 [consulta: 16-08-2014]. Disponible en: <https://www.ruby-lang.org/en/installation/#package-management-systems>

[3] RubyInstaller. Ruby Installer for Windows, 2014 [consulta: 01-08-2014]. Disponible en: <http://rubyinstaller.org/>

Apéndice II Manual de Usuario Servicios Web

En este apartado se adjunta documentación adicional desarrollada para facilitar el uso de los servicios web.

Estos servicios web se emplearán para realizar solicitudes de servicios de gestión de IT.

12.1 Configuración de los Servicios Web

Puesto que los usuarios no disponen de acceso para realizar modificaciones en los servicios web ni en su configuración, se van a indicar únicamente unas directrices.

Los documentos desarrollados para suministrar los servicios de gestión demandados requieren estar alojado en un servidor de servicios web como por ejemplo, Apache.

No se requiere una configuración específica, simplemente es necesario cargar la estructura de ficheros en uno de los directorios de carga del servidor web (por defecto, Apache carga los documentos ubicados en htdocs).

Es importante respetar la estructura de directorios definida inicialmente de forma que los servicios web funcionen correctamente.

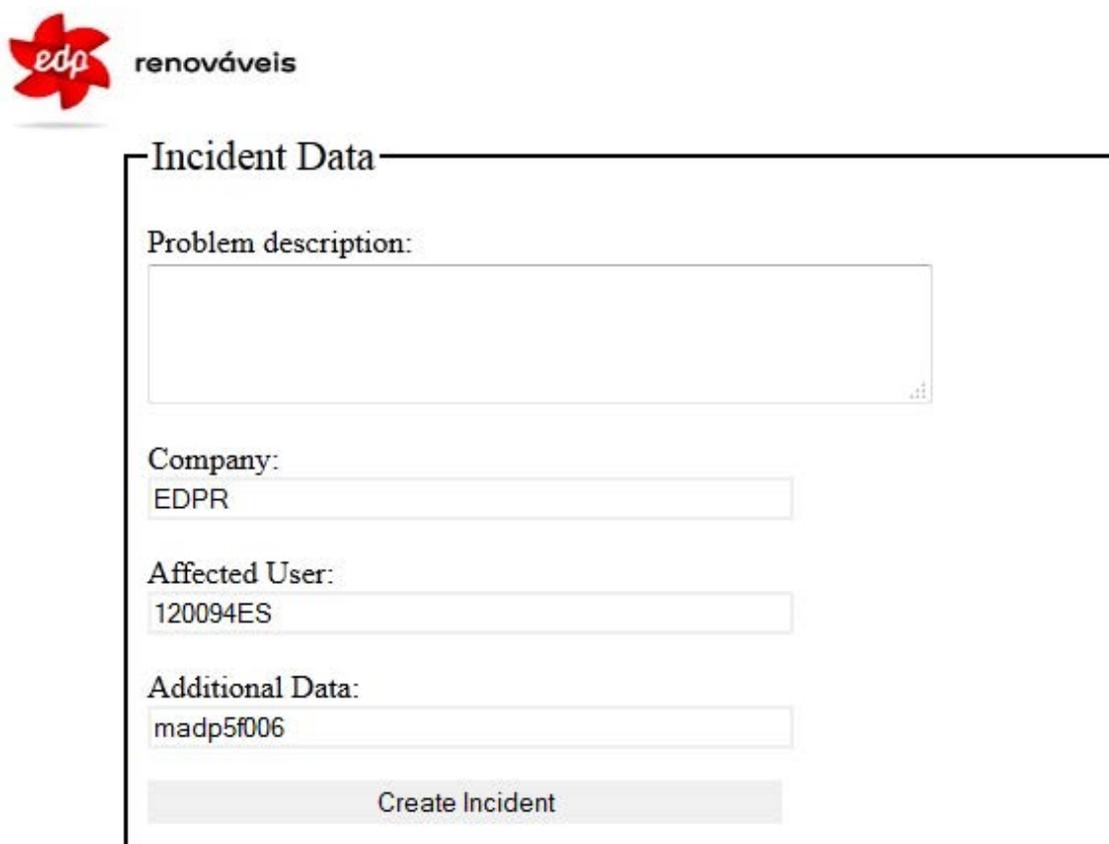
12.2 Acceso y uso de Servicios Web

Actualmente existen tres servicios web habilitados que permiten la creación de incidencias, pedidos e ítems de configuración.

Los usuarios generalmente accederán a los servicios web escaneando códigos QR que además de contener la dirección web del servicio, contienen los datos a cargar en los parámetros del formulario.

En base a los parámetros contenidos en el código QR se cargará uno de los servicios web. El número de parámetros que estarán disponibles varía en función del parámetro “mode” contenido en los códigos QR. Las siguientes figuras ilustran la diferencia de parámetros visibles, al incluir el parámetro “mode=debug”.

Formulario de solicitud de incidencias con parámetros disponibles para el modo estándar (Si el modo no está definido se considera estándar).



The image shows a web form titled "Incident Data" for the "renováveis" service. The form is enclosed in a black border and contains the following elements:

- Problem description:** A large, empty text area.
- Company:** A text input field containing "EDPR".
- Affected User:** A text input field containing "120094ES".
- Additional Data:** A text input field containing "madp5f006".
- Create Incident:** A grey button at the bottom of the form.

Figura 12.2.a: Formulario con parámetros para la creación de incidencias, modo estándar.

Formulario de solicitud de incidencias con parámetros disponibles para el modo debug. Se habilita la edición de la tipificación de incidencia.



Incident Data

Problem description:

Company:

EDPR

Affected User:

120094ES

Additional Data:

madp5f006

Category:

OTHER

Subcategory:

OTHER

Type:

OTHER

Group:

EDP-Global Service Desk

Create Incident

Figura 12.2.b: Formulario con parámetros para la creación de incidencias, modo debug.

Para ambos modos, una vez confirmada la corrección de los datos y cumplimentada la descripción, se podrá solicitar la creación de una incidencia. Para poder crear la incidencia el servidor de Service-Now requiere la inserción de los credenciales de usuario, por lo que se generará una ventana para permitir al usuario introducir estos datos.

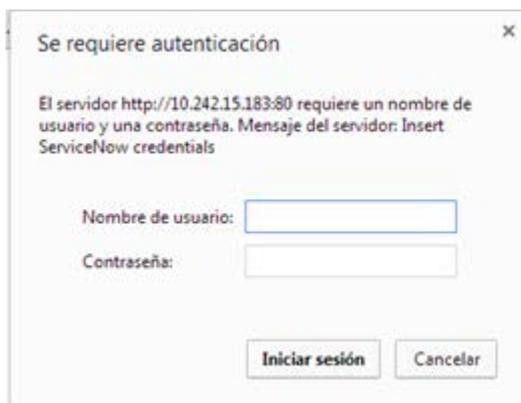


Figura 12.2.c: Formulario para la inserción de credenciales.

Una vez introducidos los credenciales, se realiza la gestión del servicio solicitado en Service-Now, y una vez finalizado se muestra el resultado.



```
New incident created
Incident number INC361636 has been created
Incident opened by DANIEL HERNANDEZ CASSEL
Incident requested for: DANIEL HERNANDEZ CASSEL

Description of the problem reported:
Incident description: This is the test description

Additional data submitted: This is the test data

Go to INC361636
```

Figura 12.2.d: Resumen de la gestión realizada, modo estándar.

Si se estaba empleando el modo debug, además del resultado se muestra la traza del proceso de generación y envío y recepción de los mensajes SOAP.



Result

```
Array
(
    [request_id] => INC361638
    [incident_data] => Incident opened by DANIEL HERNANDEZ CASSEL
    Incident requested for: MIGUEL COLLADO GOMEZ
    [message_id] => Incident number INC361638 has been created
    [incident_description] => Incident description: this is another test description

    Additional data submitted: testing debug mode
)

Go to INC361638
```

Details:

Request

```
POST /EDPR_Incident_wsdl.do?SOAP HTTP/1.0
Host: silverstormedp2.service-now.com
User-Agent: DaniSOAP/0.004 (1.0)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "http://www.service-now.com/EDPR_Incident_wsdl/insert"
Authorization: Basic MTIwMDk0RXM6U2VuMzc3LjIzMDFk
Content-Length: 880

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope"
```

Response

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=850EE3BE195EFE42957F7A2340BC22C4;Secure; Path=/; HttpOnly
Set-Cookie: glide_user_session="";secure; Expires=Thu, 01-Jan-1970 00:00:10 GMT; HttpOnly
Set-Cookie: glide_user_session="";secure; Expires=Thu, 01-Jan-1970 00:00:10 GMT; HttpOnly
Set-Cookie: glide_user="UON2Mjo2NzIzMTA1ODZkMWZlOWMwZTA2M2RhOWYyNmNhYzhkNg=="; Version=1; Max-Age=2147483647; Expires=Thu, 01-Jan-1970 00:00:10 GMT; HttpOnly
Set-Cookie: glide_user_session="UON2Mjo2NzIzMTA1ODZkMWZlOWMwZTA2M2RhOWYyNmNhYzhkNg=="; Version=1; HttpOnly
```

Figura 12.2.e: Resumen de la gestión realizada, modo debug.

12.3 Notificaciones y avisos

La totalidad de los parámetros de los servicios web son obligatorios, por lo que las solicitudes de gestión de servicios web no se podrán gestionar si alguno de los parámetros no presenta datos válidos. Cuando se solicita la gestión de un servicio, antes de generar el mensaje de solicitud se evalúan los parámetros, si alguno no cumple con los patrones establecidos o directamente no presenta datos se notificará al usuario para que proceda a su corrección.

Si hay varios parámetros considerados erróneos se mostrará aviso para todos ellos.

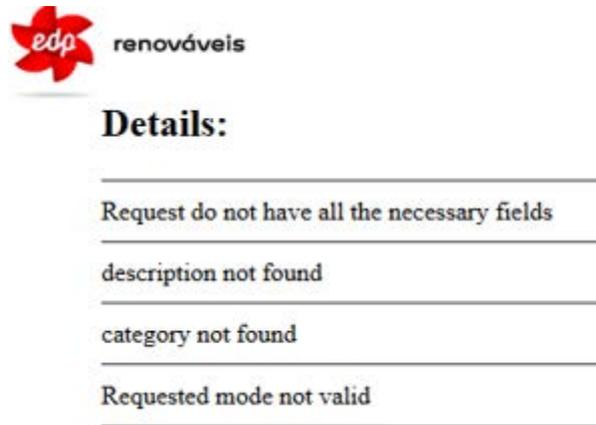


Figura 12.3.a: Notificación para campos vacíos y no válidos.

Aunque se supere la primera verificación de datos correctamente, esto no garantiza que la gestión del servicio vaya a finalizar con éxito. Únicamente se ha verificado el formato de la información, la comprobación de corrección de la información se realiza en Service-Now antes de gestionar el servicio solicitado.

Si los datos obtenidos del mensaje enviado no son válidos, en lugar de gestionar el servicio solicitado y enviar un mensaje de respuesta con el resultado de la gestión, se enviará un mensaje de respuesta con el error de datos detectado.

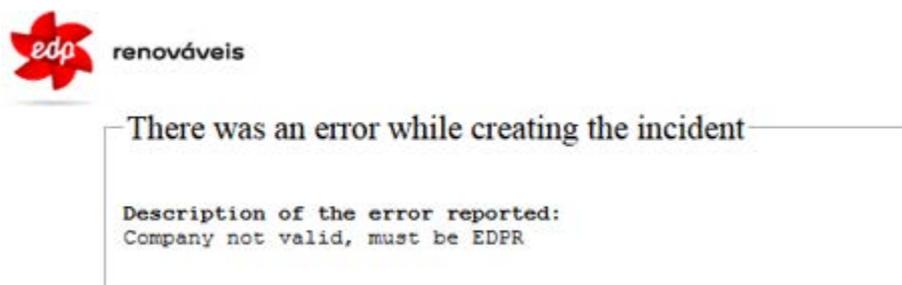


Figura 12.3.b: Notificación de compañía no válida.

Del mismo modo que pasaba con el mensaje de respuesta cuando la gestión se realizaba correctamente, si el modo asignado es debug, se muestra la traza completa:

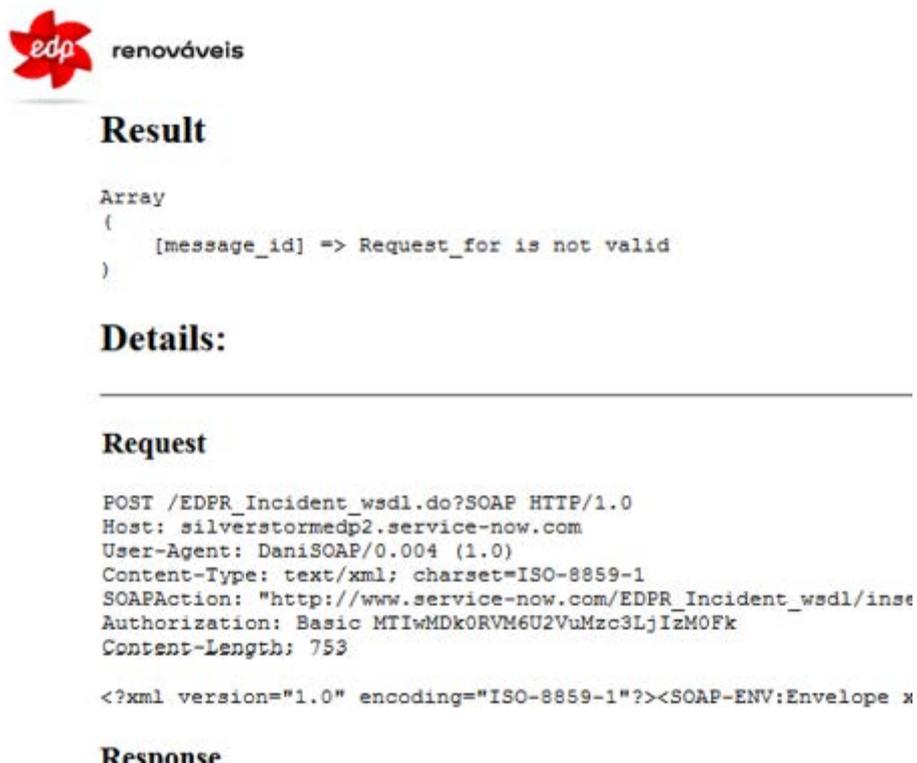


Figura 12.3.c: Notificación de usuario no valido en solicitud pedido.

Están contemplados también los posibles errores de autenticación y de conexión mediante SOAP a Service-Now.

La siguiente figura ilustra algunas de las posibles notificaciones informativas que se pueden generar.



Figura 12.3.d: Error de autenticación y errores de formato en el mensaje SOAP.

Si el mensaje de error obtenido es el primero de la figura, significa que los credenciales introducidos por el usuario no son válidos, Los otros dos errores adjuntados a continuación se producen si la estructura del mensaje no es correcta (estos errores se obtienen al emplear otros clientes SOAP diferentes a DaniSOAP, por lo que no deberían ser comunes).