



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

Marco Normativo de IT

ES0901 – Estándar de Desarrollo

Agencia de Sistemas de Información

Gobierno de la Ciudad Autónoma de Buenos Aires



Gobierno de la Ciudad Autónoma de Buenos Aires

“2014, Año de las letras argentinas”

Índice

1	INTRODUCCIÓN	4
2	MARCO NORMATIVO DE TI.....	4
3	PROCESO DE GESTIÓN DE CAMBIOS DE SOFTWARE.....	4
4	ENTREGABLES	5
5	INTERFACES CON SERVICIOS EXISTENTES EN EL GCABA	7
6	PLATAFORMAS DE DESARROLLO	7
6.1	PLATAFORMA JAVA.....	8
6.2	PLATAFORMA PHP	8
6.3	PLATAFORMA JAVASCRIPT.....	8
6.4	PLATAFORMA MÓVIL.....	9
6.5	PLATAFORMA DRUPAL.....	9
6.6	PLATAFORMA BASE	10
6.7	ENTORNOS DE DESARROLLO (IDE).....	10
6.8	PAQUETES ADICIONALES	10
7	COMPONENTES DE SOFTWARE	10
7.1	PROCESOS BATCH.....	11
7.2	MOTOR DE WORKFLOWS	11
7.3	MENSAJERÍA	11
7.4	ADMINISTRACIÓN DE ARCHIVOS PDF.....	11
7.5	ACTUALIZACIÓN DE VERSIONES.....	11
7.6	BALANCEADOR DE CARGA	11
7.7	SERVICIOS.....	11
8	DISPONIBILIDAD DE DATOS (API)	12
9	REQUERIMIENTOS NO FUNCIONALES	12
9.1	LOGGING.....	12
9.2	ADMINISTRACIÓN DE ERRORES.....	12
9.3	SERVICIO DE MONITOREO.....	12
9.4	ESCALABILIDAD.....	12
9.5	DISPONIBILIDAD	13
9.6	MODULARIDAD.....	13
9.7	ESTÉTICA	13
9.8	USABILIDAD Y ACCESIBILIDAD.....	14
9.9	SEGURIDAD.....	14
9.9.1	Autenticación de Usuarios y Administración de Contraseñas.....	14
9.9.2	Validación de los Datos de Entrada.....	15
9.9.3	Subida de Archivos.....	15
9.9.4	Metodologías Adoptadas	16
9.9.5	Otros Requerimientos de Seguridad.....	16
9.10	INTEROPERABILIDAD.....	17
10	TESTING DE ACEPTACIÓN	17
11	ENTREGA DE ACTUALIZACIONES	20
11.1	VERSIONAMIENTO.....	20
12	TRANSFERENCIA DE CONOCIMIENTOS.....	21
13	ANEXO I – GUÍA DE SERVICIOS USIG.....	22



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

13.1	MAPAS INTERACTIVOS	22
13.2	NORMALIZACIÓN DE DIRECCIONES	22
13.2.1	Normalizador Batch Desktop	23
13.2.2	Web Services	23
13.2.3	Normalizador de direcciones (JS)	23
13.2.4	Normalizador de Direcciones (Python)	23
13.3	AUTOCOMPLETER DE LUGARES	23
13.4	CALLEJERO SERVER	23
13.5	SERVICIO DE GEOCODIFICACIÓN	23
13.5.1	Servicios Web SOAP	24
13.6	INVENTARIO DE INFORMACIÓN GEOGRÁFICA DE LA CIUDAD	24
14	ANEXO II – INTEGRACIÓN CON OPENID O DIRECTORIO ACTIVO GCABA	25
15	ANEXO III – PLATAFORMA DE VERSIONADO Y PROCESO DE ENTREGA	27
16	ANEXO IV – DOCUMENTACIÓN MODELO	32
16.1	MODELO DOCUMENTO DE ARQUITECTURA	33
16.2	MODELO DOCUMENTO DE RELEVAMIENTO	40
16.3	MODELO DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS	45
16.4	MODELO DOCUMENTO DE CASOS DE USO	50
16.5	MODELO DOCUMENTO MANUAL DE INSTALACIÓN	56
16.6	MODELO DOCUMENTO SERVICIOS WEB	62
16.6.1	Descripción de Servicios Web	62
16.6.2	Funciones del Servicio Web	62
16.6.3	Elementos del Servicio Web	63
17	ANEXO V – FLUJOGRAMA PROCESO CONTROL DE CAMBIOS	65



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

1 Introducción

El objetivo del presente documento es definir los requerimientos que un desarrollo debe cumplir en relación con las tecnologías adoptadas por la ASI, como así también la interacción con los proveedores de software en cuanto a entregables, seguimiento de errores, control de cambios, etc.

Un análisis más profundo de la arquitectura puede encontrarse en el documento de Arquitectura de los sistemas del Data Center de la Agencia de Sistemas de Información (en adelante ADC en este documento).

Los criterios establecidos en este documento también son aplicables para los integradores que implementen dentro de la infraestructura informática del GCABA, sistemas comerciales ("enlatados") o desarrollos ad-hoc en cumplimiento de contratos que así lo soliciten. Estas implementaciones de "enlatados" o desarrollos ad-hoc, deben efectuarse respetando los criterios arquitectónicos y entornos de desarrollo, testing, homologación y producción descritos en este documento de tal forma que los sistemas puedan ser implementados en el Data Center único del GCABA.

IMPORTANTE: Salvo que para un proyecto en particular se especifique por contrato algún acuerdo diferente para alguno de los puntos detallados en este documento, todos los sistemas que se desarrollen en el ámbito de los proyectos de la ASI deben respetar la totalidad de los criterios aquí descriptos.

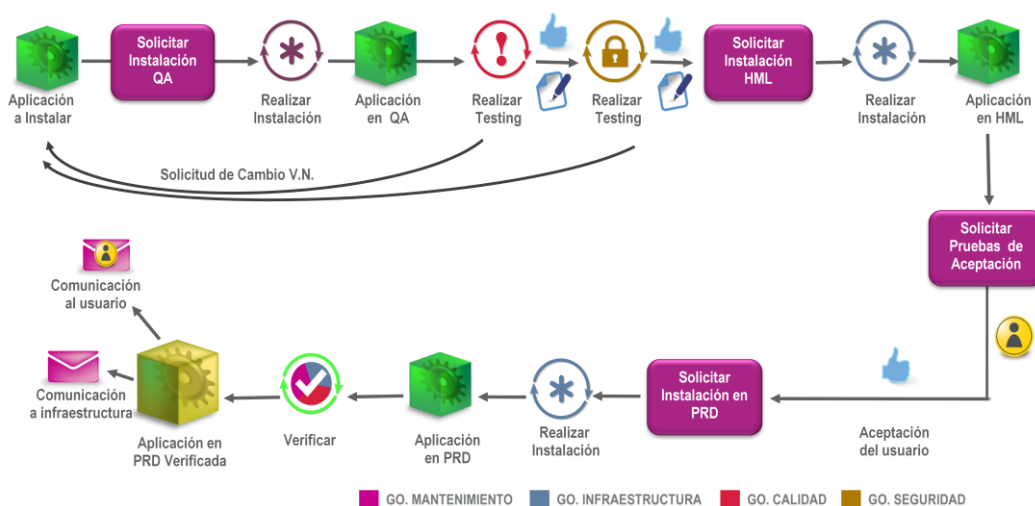
2 Marco Normativo de TI

Todo desarrollo deberá cumplir con lo expresado en el Marco Normativo de TI del GCABA, publicado en el boletín oficial del día 08-11-2013, resolución 177-ASINF-2013. Esta documentación se encuentra disponible en la Intranet del GCABA.

http://intranet/asi/formularios_procedimientos/guia_procedimientos.php

3 Proceso de Gestión de Cambios de Software

La ASI tiene implementado un proceso de control de cambios que especifica los diferentes ambientes (testing, homologación y producción). Aplica a todo cambio de software tanto para las implementaciones por primera vez como para las actualizaciones de los mismos.





Gobierno de la Ciudad Autónoma de Buenos Aires

“2014, Año de las letras argentinas”

En el anexo V se puede visualizar un flujograma de los roles, funciones y actividades macro del proceso de control de cambios que es obligatorio para cada implementación.

4 Entregables

Se detallan a continuación los entregables mínimos requeridos para un proyecto de desarrollo de software y el contenido esperado en cada uno de ellos:

Entregable		Contenido
Documento de Proyecto	Plan	<ul style="list-style-type: none">• Objetivo y alcance del proyecto.• Cronograma de tareas.• Plan de entregables del proyecto e identificación de hitos.• Organigrama, roles y responsabilidades.• Asignación de recursos a roles.• Estimación de esfuerzo total.• Análisis de riesgos y plan de mitigación de los mismos.• Plan de comunicación y seguimiento del proyecto.• Gestión de la configuración.• Gestión de cambios en requerimientos.
Documento de Requerimientos		<ul style="list-style-type: none">• Documento de requerimientos (funcionales y no funcionales) que incluya los objetivos funcionales del proyecto, alcance y origen de los requerimientos.• Requerimientos funcionales acordados y priorizados.• Especificación de los requerimientos utilizando casos de uso de primer nivel o detallados, dependiendo de la metodología.• Matriz de trazabilidad: requerimientos-casos de uso. <p>Además de los requerimientos no funcionales obligatorios, comunes a todos los aplicativos detallados en el apartado <i>Requerimientos No</i>, identificar aquellos que aplican particularmente a este proyecto, tales como:</p> <ul style="list-style-type: none">• <u>Escalabilidad</u>: es la propiedad deseable de un sistema, que indica su habilidad adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.• <u>Usabilidad</u>: se refiere a la facilidad con que los usuarios pueden operar un sistema con el fin de alcanzar un objetivo concreto.• <u>Performance</u>: tiempo de respuesta de un sistema para procesar información dentro de determinados valores preestablecidos.• <u>Robustez</u>: capacidad del sistema de funcionar correctamente ante la presencia de entradas inválidas o condiciones de stress en los ambientes.• <u>Autenticación</u>: garantiza la gestión de los accesos a la información. Asegura que sólo los usuarios con la necesidad adecuada y la autorización debida puedan acceder al sistema.• <u>Confidencialidad</u>: garantiza la protección de la información sensible.• <u>Integridad</u>: garantiza la inalterabilidad de la información.• <u>Disponibilidad</u>: garantiza el acceso a la información crítica.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

Entregable	Contenido
	<p>Ten.</p> <ul style="list-style-type: none">• Casos de tests realizados y sus resultados, adjuntando evidencias de test funcionales y no funcionales (stress, performance, volumen, seguridad).• Transferencia de conocimiento.• Manual de operaciones e instalación.• Video de cada instalación (opcional).• Documento para facilitar la transferencia de conocimientos (estructura del código fuente, particularidades de compilación y funcionamiento del sistema, configuración, parametrización y todo lo necesario para poder realizar la implementación del desarrollo y su posterior mantenimiento).
Documento Manual de Usuario	<ul style="list-style-type: none">• Documentación completa del software que incluya todas las funcionalidades para todos los roles y la administración de roles, permisos y seguridad.
Material de Capacitación	<ul style="list-style-type: none">• Propuesta con el plan de capacitación.• Documentación que incluya la completitud de la funcionalidad a capacitar.
Minutas e Informes de Avance	<ul style="list-style-type: none">• Todos las minutas generadas durante el proyecto e informes de avances presentados/enviados.

5 Interfaces con Servicios Existentes en el GCABA

El GCABA cuenta con una serie de aplicaciones que brindan servicios generales los cuales deben ser usados por el resto de los sistemas.

- Normalización y georeferenciación de calles. Ver Anexo I – Guía de Servicios USIG
- Reparticiones (SADE)
- Caratulación de expedientes (SADE)
- Plataforma de Turnos
- OpenID. Ver Anexo II
- Registros varios
- Alta de reclamos, denuncias, solicitudes (SUACI)

6 Plataformas de desarrollo

Las plataformas de desarrollo autorizadas son las siguientes:

- Plataforma Java
- Plataforma PHP
- Plataforma Javascript



Gobierno de la Ciudad Autónoma de Buenos Aires

“2014, Año de las letras argentinas”

- Plataforma Móvil

Si se presenta una situación de duda sobre la elección de la plataforma/motor de Base de Datos, se debe solicitar asesoramiento a la ASI al momento del Diseño.

6.1 Plataforma JAVA

Esta plataforma es la indicada para casos donde los requerimientos de robustez son altos, complejos, procesamiento intensivo a nivel plataforma, consideradas críticas por la alta demanda o cantidad de usuarios concurrentes activos.

En casos que el requerimiento de usabilidad exija Front-end con efectos visuales más amigables y funcionales es conveniente utilizar esta plataforma.

El motor de Bases de datos debe ser DB2. Existe la posibilidad de usar Oracle como opción pero es necesario al momento de selección solicitar la autorización a la ASI previo al desarrollo, con el fin de determinar la cantidad de licencias necesarias.

Todo desarrollo deberá realizarse haciendo uso del *stack* Wicket – Spring – Hibernate - JUnit. El objetivo principal de la arquitectura es separar las distintas capas de desarrollo, permitiendo un modelo de dominio limpio y facilidad de mantenimiento de las aplicaciones.

La plataforma Java está basada en Linux, Tomcat, JBoss.

6.2 Plataforma PHP

Esta plataforma es la indicada para casos de aplicaciones de complejidad o cantidad de usuarios concurrentes media o baja en su demanda. Es aconsejable para casos de desarrollos o requerimientos evolutivos dinámicos y sencillos una vez implementada, con modelos de datos simples que no necesitan escalar en el futuro.

En el caso de presentarse un modelo de datos sencillo, con baja concurrencia de usuarios activos, bajo nivel transaccional en el motor de Base de Datos o sin presencia de picos de demanda pronunciados puede usarse MySQL.

En el caso de presentarse situaciones de alta concurrencia de usuarios activos o alto nivel transaccional del motor de Bases de datos, debe ser DB2. Existe la posibilidad de usar Oracle como opción pero es necesario al momento de selección solicitar la autorización a la ASI previo al desarrollo, con el fin de determinar la cantidad de licencias necesarias.

Las aplicaciones PHP deben desarrollarse utilizando un framework. Actualmente, el único framework autorizado es Symfony 2.

En todos los casos se utilizará APC como OpCode Caches. La plataforma PHP está basada en Linux, Apache,

6.3 Plataforma Javascript

En el caso de presentarse la necesidad de la construcción de API's que requieran alta disponibilidad o sufran un elevado número de accesos concurrentes se recomienda esta plataforma.

También puede optarse para la creación de prototipos sencillos, acotados y evolutivos de duración acotada.



Gobierno de la Ciudad Autónoma de Buenos Aires

“2014, Año de las letras argentinas”

En el caso de presentarse un modelo de datos sencillo, con baja concurrencia de usuarios activos o sin presencia de picos de demanda pronunciados puede usarse MySQL.

En caso de que el modelo no se encuentre definido inicialmente y esté expuesto a cambios periódicos, puede utilizarse el motor MongoDB, pero debe existir una forma o método que permita extraer los datos relacionados en un formato legible.

La plataforma Javascript esta basada en Linux, Ngix 1.x, Nodejs 1.x, Express 3.x.
Para la gestión de paquetes debe utilizarse npm.

6.4 Plataforma Móvil

El desarrollo de las aplicaciones móviles se divide en nativo y no nativo.
Se llama desarrollo “nativo” a las aplicaciones desarrolladas con el SDK oficial de cada una de las plataformas:

- Android - Eclipse (JAVA)
- iOS - Xcode (Objective-C)
- BlackBerry - Eclipse (JAVA)

Y “no nativo” a las desarrolladas con HTML5. En este caso está autorizado el uso del framework PhoneGap.

6.5 Plataforma DRUPAL

En aquellos sitios que requieran actualización de contenido deberá utilizarse Drupal 7.x como plataforma. El mismo es el único CMS homologado por la ASI.

La base de datos a emplear en todos los casos es MySQL.

Las actualizaciones de software deberán realizarse a través de comandos Drush en todos los casos donde sea factible.

El sitio deberá contar con su archivo drush.make correspondiente para la puesta en producción.

Se deberá respetar la siguiente estructura de carpetas para los módulos:

- **<drupal-root>/sites/<sitename>/modules/contrib**
Módulos desarrollados por terceros
- **<drupal-root>/sites/<sitename>/modules/features**
Módulos generados automáticamente por el módulo features
(<https://drupal.org/project/features>)
- **<drupal-root>/sites/<sitename>/modules/custom**
Módulos desarrollados ad hoc.

El código de los módulos custom deben seguir los estándares de calidad y estilos de Drupal (<https://drupal.org/node/1354>). Estos módulos deben ser desarrollados con el objetivo de poder ser reutilizado y su código fuente podrá ser liberado a la comunidad.

Los módulos instalados deberán ser los estrictamente necesarios para el funcionamiento del sitio no debiendo haber módulos que no cumplan ninguna función o hayan quedado en desuso. Las modificaciones de esquema que se realicen sobre la base de datos deben ser efectuadas a través de las features creadas a tales efectos.

Para la generación de layouts para el sitio debe utilizarse el módulo panels (<https://drupal.org/project/panels>) y en el caso en que los mismos superen el límite de 30 variantes de nodo (node variants) deberá utilizarse el módulo panelizer (<https://drupal.org/project/panelizer>) para mejorar la administración de las mismas.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

Para la creación de grupos/areas/secciones el módulo a utilizar es Organic Groups (<https://drupal.org/project/og>).

Estás urls son las que tiene drupal.org. No todas las páginas de drupal.org tienen urls amigables.

6.6 Plataforma Base

La ASI brinda al proveedor las imágenes virtuales con las configuraciones estándar existentes en producción según la lista especificada en el ADC con el fin de evitar posibles problemas de compatibilidad entre los ambientes de desarrollo-homologación-producción.

Los módulos Apache habilitados son los disponibles en las VM provistas por la ASI. Es importante que el proveedor utilice tanto para desarrollar como para testear las aplicaciones estas imágenes VMware para asegurar la compatibilidad.

Es obligatorio que contemplen que el sistema operativo que deben utilizar es el homologado por la ASI de **RedHat**.

6.7 Entornos de Desarrollo (IDE)

Plataforma Java: Los proyectos de desarrollo deben ser compatibles con Eclipse, Maven y Subversion. Se debe instalar dentro de la ASI un entorno de desarrollo (configuración de Eclipse para debug y build de las aplicaciones) de manera que el software o las modificaciones al mismo se puedan entregar en forma de commits en el sistema de control de versiones y que se puedan verificar las modificaciones hechas, así como hacer quality assurance y auditorías de seguridad en modo "white box".

El proveedor debe entregar documentación escrita de cómo configurar Eclipse, Maven, y otros sistemas para poder trabajar sobre el código.

6.8 Paquetes Adicionales

Previo acuerdo con la ASI, cualquier paquete o componente de software que se instale con la aplicación de forma adicional al software provisto en la plataforma base y componentes adicionales, se considera parte de la aplicación y por lo tanto el desarrollador es responsable de su buen funcionamiento y compatibilidad con el software de base definido como estándar por la ASI para sus servidores según las restricciones aplicadas en la ADC.

7 Componentes de Software

Para las siguientes funciones dentro de los sistemas se deben utilizar los componentes determinados por la ASI.

Las versiones autorizadas de cada uno de estos componentes van a ser actualizadas periódicamente. Por este motivo, al inicio de cada proyecto se debe establecer cuál es la versión autorizada para cada uno de ellos.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

7.1 Procesos Batch

Las aplicaciones que requieran para su funcionamiento ejecutar procesos batch (archivos por lotes, scripts de Shell, procesos en background), deberán documentar el detalle de cómo configurarlos, lanzarlos y bajo qué condiciones ejecutarlos, para que la ASI pueda tomar control de los mismos y monitorearlos en función de los objetivos definidos.

La aplicación no deberá tener capacidades de ejecución o control de los procesos que requiera para su ejecución. Todos los procesos estarán centralizados en una herramienta independiente de la aplicación en cuestión.

7.2 Motor de Workflows

Los workflows de negocios se deben implementar sobre jBPM

Para no repetir funcionalidad ya implementada se debe analizar, además, que los requerimientos no correspondan a los sistemas SADE (seguimiento de expedientes) y SUACI (reclamos). En caso de que sí ocurra, se debe implementar la nueva funcionalidad sobre estos sistemas.

7.3 Mensajería

Se deben utilizar los servidores de ZeroMQ ya instalados en el GCBA, sin interferir con otros sistemas. Se puede utilizar tanto en Java como en PHP para desacoplar sistemas.

Más información sobre ZeroMQ puede consultarse en <http://zeromq.org/>

7.4 Administración de Archivos PDF

Todas las transformaciones de archivos PDF, firmas digitales en el servidor, habilitación (reader extensions), procesamiento, etc, deben programarse como procesos de Adobe Livecycle ES y ser llamados desde las aplicaciones vía SOAPo Rest.

No se deben utilizar librerías llamadas desde PHP o Java para realizar tareas que se pueden realizar con Livecycle (<http://helpx.adobe.com/livecycle.html>). Asimismo todas las transacciones deben ser stateless.

7.5 Actualización de versiones

Al momento del diseño debe contemplarse que aquellas configuraciones que sufran cambios deben ser actualizadas por medio de un backend para evitar el deploy de nuevas versiones con el costo que eso conlleva.

7.6 Balanceador de Carga

En la arquitectura de aplicaciones web de la ASI se utiliza HAProxy o F5 como balanceador de capa 7 (HTTP), asignando sesiones nuevas a los web servers mediante el algoritmo *round-robin* y manteniendo "*sticky sessions*" mediante la inspección de las cookies de sesión de cada lenguaje (JSESSIONID o PHPSESSID) de manera que todos los pedidos de un mismo cliente vayan a un mismo servidor web durante toda la sesión.

7.7 Servicios

No se da exclusividad en el acceso a los servicios (permisos de "admin").



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

La lista de servicios figura en el Documento de Arquitectura, y no pueden ser reemplazados por otros que cumplan la misma función. Si es necesario un servicio de infraestructura que no está disponible se debe plantear su adición al stack provisto por la ASI.

8 Disponibilidad de Datos (API)

Toda aplicación debe contar con una API disponible con la información relevante y con un formato de extracción de datos para exponer los datos.

Los estándares para la creación de API's se encuentran disponibles en <https://github.com/gcba/api-standards>

9 Requerimientos No Funcionales

9.1 Logging

Se debe definir el nivel de logging de los sistemas a implementar dependiendo de la naturaleza de cada uno de ellos.

Plataforma Java: las aplicaciones deben usar log4j

Plataforma PHP: las aplicaciones deben usar syslog

<https://github.com/gcba/api-standards>.

9.2 Administración de Errores

La aplicación debe evitar enmascarar los mensajes de error http de forma tal que los balanceadores / proxy puedan entender el estado actual del aplicativo.

Ante un error inesperado no debe visualizarse por front características del servidor, ip, path, o cualquier información que permita conocer propiedades de infraestructura

Por ejemplo, si ocurre un error en el aplicativo la respuesta http de estado debe ser código 500 y el mismo no debe reemplazarse por ningún otro código.

9.3 Servicio de Monitoreo

Las aplicaciones deben contar con transacciones testigo para monitoreo, cada transacción crítica que tenga el sistema debe tener su homólogo para monitoreo. Esta transacción debe realizar los mismos pasos que la transacción real pero no debe generar impacto real y debe estar documentada (por ejemplo, en una transacción de pago, la transacción de monitoreo no debe generar un pago real).

El objetivo de estas transacciones es monitorear la disponibilidad del servicio.

Adicionalmente la aplicación debe contar con una forma de setear que se encuentra en modo mantenimiento. Se sugiere como mejor opción que sea por Backend , caso contrario que permita setear un archivo de configuración y el sistema muestre un imagen de este estado provisorio.

9.4 Escalabilidad

Se debe proveer escalabilidad horizontal en el sistema a desarrollar teniendo en cuenta las especificaciones realizadas en el Documento de Arquitectura, como ser la integración con



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

balanceador de capa 7, Application servers granjeables y stateless, la separación en distintas vlans de sus componentes por cada servicio que ofrezcan o consuman, evitar el uso de caché distribuidos o mecanismos de clusters, etc.

Cuando se necesite acceso rápido a información con baja probabilidad de cambios y empleada con mucha frecuencia se debe implementar sistemas de caché de contenido y de datos. Esto evita realizar invocaciones innecesarias que degradan la performance del aplicativo.

Para servir contenido cacheado (html, css, imágenes o cualquier contenido estático) deberá emplearse el acelerador http Varnish.

En el caso que el sitio requiera autenticación por parte del usuario deberá utilizarse una lógica de cache adecuada que garantice la performance del mismo y el consumo moderado de recursos. Todo contenido que requiera cierta frecuencia de actualización deberá consultarse a su origen (base de datos, webservices, etc.) una sola vez por ciclo de dicha frecuencia.

Ejemplo: si un feed de RSS requiere una frecuencia de actualización de 5 minutos solamente deberá efectuarse una sola consulta como máximo cada 5 minutos.

Para el caché de datos deben utilizarse los servicios de Memcached o Redis provistos por la agencia.

Es responsabilidad del desarrollador el mantenimiento de los archivos de configuración de Varnish.

9.5 Disponibilidad

La aplicación debe ser 7x24 sin ventana de mantenimiento, sin interrumpir la prestación de servicios interactivos o Web Services.

Se debe tener en cuenta que todos los procesos de backups que sean necesarios deben soportar este requerimiento.

En caso de contar con la necesidad de bajar la aplicación por un tiempo determinado, debe contemplar desde la administración del usuario que un administrador pueda realizar una baja controlada del servicio que permita visualizar a los usuarios un aviso de la situación.

9.6 Modularidad

Los distintos servicios, Application servers, base de datos, etc. deben estar desacoplados, permitiendo separarlos y configurarlos en distintas vlans según la funcionalidad de los mismos, como se encuentra especificado en el Documento de Arquitectura.

9.7 Estética

Los sitios web de la Ciudad de Buenos Aires deben tener una concordancia estética homogénea que se corresponde con el plan gráfico integral del Gobierno de la Ciudad. Por este motivo, los diseños web ya sean de aplicaciones accedidas desde la intranet o desde internet deben tener la conformidad de la ASI.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

9.8 Usabilidad y Accesibilidad

Dado que uno de los principales objetivos del GCBA es hacer que los sitios web de la Ciudad de Buenos Aires sean disponibles para todas las personas, cualquiera sea su hardware, software, infraestructura de red, cultura, localización geográfica, o capacidad física o mental, el área de usabilidad y creatividad ha generado las Pautas/ lineamientos que se encuentran accesible en <http://www.buenosaires.gob.ar/guiaux>

9.9 Seguridad

Las aplicaciones web deben ser auditadas según los criterios de Seguridad Informática de la ASI y se debe garantizar que cumplen con los criterios de la misma.

Algoritmos criptográficos probados (RSA, IDEA, SHA1)

- Si se requiere secreto, no repudio, u otra característica de seguridad sobre la información, se debe hacer uso de los protocolos criptográficos necesarios que lo garanticen.
- No se debe dar por supuesto que otras capas del sistema proveen un manto de seguridad (firewalls, web application firewalls, seguridad del application Server).
- No se debe suponer que las aplicaciones son seguras por estar dentro de una red corporativa, por las características y el tamaño de la red MAN del GCABA.
- No es un criterio válido asumir que si sólo un administrador tiene acceso, entonces es seguro.

Se deben documentar todos los puntos donde el sistema tiene configuradas credenciales (de acceso a la base de datos, de acceso a otros servicios, etc.) y las contraseñas deberán ser robustas de acuerdo con los criterios establecidos en la resolución 177-ASINF-2013.

Permisos de usuarios

A los usuarios deben asignárseles los mínimos privilegios requeridos para realizar sus tareas. Este requerimiento incluye tanto usuarios de las aplicaciones como de bases de datos, file systems y cualquier otro usuario que esté relacionado con el sistema desarrollado.

Aplicaciones de administración

Se recomienda realizar una separación de las consolas de administración de manera que puedan ser ubicadas físicamente en servidores dentro de la red local del GCABA e impedir el acceso por parte de los usuarios finales.

Pantalla de autenticación

En el caso de ocurrir errores en la autenticación, el sistema no debe informar el parámetro equivocado. El mensaje debe ser del tipo: "Usuario / contraseña inválidos o usuario bloqueado."

9.9.1 Autenticación de Usuarios y Administración de Contraseñas

Ingreso al sistema

Toda aplicación que se encuentre accesible por el ciudadano debe integrarse con el identificador único (OpenID) que brinda el GCABA. Ver Anexo II. En caso de no cumplir la condición anterior debe integrarse con el Directorio Activo GCABA.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

Una vez que el usuario es autenticado y autorizado, el sistema debe mostrar la fecha / hora del último acceso de ese usuario al sistema.

Contraseñas

- Deben permanecer cifradas utilizando un algoritmo unidireccional y residir en archivos ocultos y protegidos.
- No deben ser visibles por pantalla al momento de ser ingresadas.
- El sistema debe rechazar la selección de contraseñas en blanco (nulas).

Datos sensibles según las necesidades de negocio

Otros datos sensibles y confidenciales para las necesidades de negocio también deben transmitirse y almacenarse de manera protegida.

Cada aplicación debe aplicar el concepto de Roles- Acciones que apliquen a la funcionalidad requerida y la misma debe contar con un administrador desde un BackOffice donde permita configurar/adaptar a los cambios.

Parámetros de autenticación adicionales

Toda aplicación debe implementar de forma obligatoria funcionalidades del tipo desafío-respuesta ("captcha") utilizadas para determinar cuándo el usuario es o no humano. Dicha funcionalidad debe habilitarse frente a cinco (5) o más intentos de acceso fallidos.

Esta funcionalidad debe poder configurarse desde el archivo de configuración / instalación de la aplicación, de forma tal de poder deshabilitarla con el fin de facilitar las pruebas de calidad. Por defecto este requerimiento debe estar habilitado al instalar la aplicación. El manual de instalación debe describir las instrucciones necesarias para llevar adelante esta acción.

9.9.2 Validación de los Datos de Entrada

Todas las pantallas de ingreso de datos deben validar los datos provistos por el usuario en tipo de dato, longitud y rango válido.

No se debe confiar en las validaciones del cliente. Las validaciones deben estar implementadas tanto en la capa de presentación de la aplicación como en el servidor.

Los datos ingresados deben presumirse incorrectos hasta que se demuestre lo contrario.

9.9.3 Subida de Archivos

Si la aplicación permite la subida (upload) de archivos se debe verificar no solo la extensión sino también su tipo MIME.

En el caso de ser imágenes se recomienda realizar algún tipo de transformación del lado del servidor para prevenir ataques conocidos (Ej.: GIFAR)

No se debe permitir la subida de ningún archivo que pueda ser procesado por el servidor. Por ejemplo, están prohibidas las subidas de archivos con extensiones php, asp, htm, exe entre otras. Adicionalmente, si estos archivos se almacenan en un sistema de archivos, la carpeta correspondiente no debe permitir ejecución.

Toda subida de archivos debe validar el tamaño máximo permitido de los archivos. Este valor debe ser configurable a través de un administrador.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

9.9.4 Metodologías Adoptadas

El sistema a desarrollar debe cumplir con los siguientes requerimientos basados en el Proyecto Abierto OWASP.

Es necesario que los desarrolladores y personal de testing utilicen esta guía para asegurarse de que están produciendo código seguro. Estas prácticas deben ser parte de los procedimientos normales de comprobación de código y módulos, a fin de verificar que no se ha dejado ningún agujero de seguridad en la aplicación.

Más información en www.owasp.org

9.9.5 Otros Requerimientos de Seguridad

Requerimiento	Ataque a evitar
Ocultar las extensiones de scripts públicos en los servidores. Por ej. en lugar de http://host/servicio.php , configurar el web server de forma tal que se vea http://host/servicio . Al configurar los servicios de esta forma es posible modificar la implementación e incluso la tecnología sin necesidad de modificar las interfaces.	Explotar vulnerabilidades por el conocimiento de la tecnología usada para la implementación.
Configurar ServerSignature Off en los virtual hosts de Apache	Explotar vulnerabilidades a través de la información provista por los mensajes de error que incluyan la versión de Apache, módulos instalados y sistema operativo del servidor.
Deshabilitar el display de errores por pantalla en los diferentes módulos/lenguajes/frameworks, etc. que se utilicen en producción. Por ej. configurar debug=False en los settings de las aplicaciones Django o display_errors Off en la configuración de PHP.	Acceso a la estructura interna del sistema. Cuando las aplicaciones fallan en tiempo de ejecución los mensajes que pueden aparecer por pantalla no solo pueden brindar pistas sobre posibles vulnerabilidades de la implementación sino también información que no debería estar accesible al usuario tales como estructuras de bases de datos, nombres de usuarios, paths, etc.
Estructurar las aplicaciones de forma tal que solamente aquellos archivos que deben ser accedidos desde fuera estén incluidos en directorios publicados en el web Server.	Ejecución de funcionalidades restringidas de la aplicación. En lenguajes interpretados como PHP y Python donde la simple invocación de los scripts provoca su ejecución por parte del web server, podrían registrarse ejecuciones no deseadas de partes internas de una aplicación que no deberían estar accesibles al usuario
Validar siempre los parámetros de entrada (tanto en el cliente como en el servidor) y restringir lo máximo posible el universo de valores aceptados para los parámetros. Ver 9.9.2 - Validación de los Datos de	Sql injection y cualquier otro tipo de ataque basado en la introducción de código malicioso en los parámetros.
Configurar los accesos a bases de datos en las aplicaciones con usuarios con los mínimos	Escalada de privilegios una vez explotada una vulnerabilidad de tipo SQL injection. Ante



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

Requerimiento	Ataque a evitar
permisos necesarios.	una vulnerabilidad de tipo SQL injection en la aplicación o en alguno de los software de base, un atacante pueda escalar privilegios sobre la base de datos o bien simplemente usufructuar más permisos de los que la aplicación necesitaba.
Disponibilizar archivos para descarga a través de links validados y con timeout.	Ejecución de un script que recorra una estructura de directorios (visible o deducida) y descargue archivos sin utilizar la aplicación.
Habilitar el Directory Index de Apache y otros web servers solo para directorios específicamente elegidos en lugar de habilitarlo por defecto.	Acceso a carpetas privadas de la aplicación.
Cuando el sistema necesite pasar información sensible de una página a otra, hacerlo de una manera protegida usando algún algoritmo de hashing o encriptación.	Modificación de valores de los parámetros en las urls para acceder o actualizar información no autorizada.

9.10 Interoperabilidad

La ASI aspira a maximizar la interoperabilidad entre los sistemas del GCABA buscando que cada sistema se atenga a la resolución de los problemas propios del dominio en que actúa.

Al mismo tiempo, es habitual que las áreas de gobierno interactúen y, por lo tanto, haya interdependencia de datos. Se espera que en estos casos las estrategias de intercambio de datos estén dirigidas a intercambio vía servicios-web. En caso de sistemas legacy, se debe analizar la posibilidad de creación de interfaces de interacción web, considerándose éstas como un módulo más de la aplicación/solución entregada por el proveedor o, como una dependencia del proyecto, en caso que el proveedor de datos tenga la capacidad de desarrollar la interfaz al sistema.

10 Testing de aceptación

El proveedor deberá realizar los tests unitarios y de desarrollo antes de realizar la entrega del producto. Por lo tanto, es deseable que se entreguen los casos de test ejecutados y los resultados de los mismos con anticipación a la entrega formal del software. Los casos de test realizados por el proveedor deberán incluir los requerimientos no funcionales.

Remarcamos que el producto entregado debe contar con calidad de aceptación y es responsabilidad del proveedor que el mismo cumpla con los requerimientos funcionales y los requisitos no funcionales definidos o sugeridos por la ASI. El paso del producto por la Gerencia de calidad no desliga de responsabilidades, ni es una extensión del sector de desarrollo del proveedor.

El equipo de testing de la ASI se encargará de realizar los tests de integración y funcionalidad general. El proceso de testing involucra los conceptos de severidad y prioridad.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

La severidad puede ser:

Crítica	Mayor	Menor	Cosmética
a) Cuando los usuarios no pueden utilizar las funcionalidades principales del sistema. b) Cuando no es posible realizar algún trabajo productivo. c) Cuando no se puede prestar el servicio a los usuarios.	a) Cuando el sistema está operando pero con restricciones. b) Existe impacto en la prestación del servicio a los usuarios. c) Existe impacto para los usuarios.	a) Cuando los usuarios no pueden utilizar las funcionalidades secundarias del sistema. b) Cuando no se encuentran disponibles algunas funciones o componentes del Sistema, que generan un impacto mínimo para los clientes y para los usuarios. c) Cuando las limitaciones no son críticas para la operación. d) El impacto no genera un riesgo considerable, pero es necesario resolverlo.	El error se refiere a un mal funcionamiento o diseño de la interfaz de usuario, que no impide la correcta ejecución del sistema.

La prioridad indica la precedencia o el orden en el tiempo en que deben solucionarse los errores reportados.

La prioridad puede ser:

Urgente	Alta	Normal	Baja
El Error debe solucionarse inmediatamente dado que no permite continuar, puede requerir una entrega especial.	El error debe solucionarse tan pronto como sea posible dado que se requiere para la próxima salida a producción.	Prioridad media, es deseable la corrección del error para la salida a producción.	Corregir el error sólo si no afecta calendario ni la corrección de otros errores de mayor prioridad o severidad, puede salirse a producción sin corregir.

La ASI provee una instancia de MantisBT accesible desde el exterior por los proveedores mediante el cual se administran los incidentes del proyecto. A través de este sistema se puede hacer un seguimiento de defectos o pedidos de nuevas funcionalidades



Gobierno de la Ciudad Autónoma de Buenos Aires

“2014, Año de las letras argentinas”

El software entregado se aprobará cuando exista:

- Según su severidad:
0% incidentes críticos y 0% incidentes mayores. El porcentaje de incidentes menores e incidentes cosméticos permitidos abiertos para dar la aceptación de calidad por parte de la ASI se acordarán durante el transcurso y planificación del proyecto.
- Según su prioridad:
0% incidentes de prioridad urgente y 0% incidentes de prioridad Alta. El porcentaje de incidentes con prioridad normal y baja permitidos abiertos para dar la aceptación de calidad por parte de la ASI se acordarán durante el transcurso y planificación del proyecto.

11 Entrega de Actualizaciones

Toda entrega de archivos entre un proveedor externo y la ASI se hace a través de un servicio Versionador con autenticación y encriptación provista por SSH. Se busca evitar el uso de VPNs manteniendo una comunicación segura entre las partes y ofreciendo la integridad de los datos.

Con cada entrega, ya sea inicial o de actualización de versiones, el proveedor debe entregar el código fuente, imágenes, librerías y demás archivos que sean necesarios para la instalación del software junto con la documentación establecida para el entregable.

Para que un entregable de desarrollo sea aceptado, se debe poder generar sus binarios en la ASI, hacer debugging y correr los unit tests del mismo por parte del personal de la ASI.

En caso que el proveedor entregue una versión con un incidente conocido y que la ASI lo acepte, éste debe especificar los casos en que sucede y cómo salvarlo.

Los usuarios de las aplicaciones no tienen permisos de DML (Data Manipulation Language) en las bases de datos administradas por la ASI. Por lo tanto, si la versión entregada incluye modificaciones a la estructura de la base de datos, las mismas no pueden realizarse en forma automática con su despliegue. El proveedor de software debe incluir los scripts de bases de datos con las modificaciones correspondientes dentro del paquete entregable.

Ver Anexo III con el detalle del procedimiento de entrega.

11.1 Versionamiento

Es obligatorio que todo producto a instalar debe contar una etiqueta de versión visible fácilmente siguiendo las reglas siguientes:

- La versión beta es una versión completa desde un punto de vista funcional. El equipo de proyecto debe ser capaz de justificar que se han alcanzado los requisitos y expectativas.
- La versión estable se denota como 1.0.0, luego se tienen las siguientes reglas:
 - 1.0.Z: Resolución de incidencias sobre la versión 1.0.0
 - 1.Y.0: Si aumenta la Y, la Z se pone a cero e implica que se han incorporado mejoras y probablemente resolución de incidencias
 - Z.0.0: Conlleva lo anterior pero además implica cambios importantes a múltiples niveles (diseño, funcionalidades, UI, etc.)



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

- En caso de un desarrollo iterativo, el proveedor ofrece la capacidad de generar versiones de corrección de incidentes en el medio de una etapa de desarrollo (capacidad de branching de versiones estables y un posterior merging con el trunk o versión inestable de desarrollo).

12 Transferencia de Conocimientos

El proveedor debe realizar la transferencia de conocimientos del proyecto a un grupo de funcionarios de la ASI con la menor cantidad de inconvenientes posible. Dicho traspaso debe incluir la entrega de las últimas versiones de la documentación funcional y de diseño relacionada con el proyecto más una capacitación de los funcionarios técnicos sobre la estructura del código fuente, las particularidades de compilación y funcionamiento del sistema, configuración, parametrización y todo lo necesario para poder realizar la implementación del desarrollo y su posterior mantenimiento.

Esta capacitación debe realizarse antes de finalizar la ejecución del proyecto de manera de poder detectar en forma temprana los problemas que se encuentren en el sistema y entender los arreglos que se vayan realizando.

Esta transferencia tiene como segundo objetivo permitir la evaluación de cuestiones no funcionales que sólo son visibles si se comprende el funcionamiento interno de los componentes.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

13 Anexo I – Guía de Servicios USIG

La Unidad de Sistemas de Información Geográfica (USIG) depende de la Agencia de Sistemas de Información y, de acuerdo con las leyes N°2525 y N°2689, desarrolla y administra el sistema único de información geográfica para el Gobierno de la Ciudad de Buenos Aires. La base de este sistema parte de la noción de Gestión Geográfica que destaca la importancia que posee la localización de los eventos u objetos permitiendo, por ejemplo, comprender cómo se relacionan (espacialmente), analizar la proximidad y su distribución dentro de un barrio o comuna, o planificar acciones futuras sobre diversas áreas de interés (inspecciones, verificaciones, relevamiento de información, obras u otro tipo de intervenciones en la ciudad).

Por este motivo, la USIG ofrece un sistema capaz de incorporar toda la información disponible y susceptible de ser representada cartográficamente, a partir de una plataforma informática que aporta eficiencia y coherencia en la administración y que ofrece además, diversas herramientas de análisis, consulta y administración de datos a los diferentes sectores de la Ciudad.

13.1 Mapas Interactivos

La USIG provee a los organismos del Gobierno de la Ciudad de Buenos Aires, como así también a sus ciudadanos, diferentes tipos de mapas interactivos. Las soluciones varían en complejidad y funcionalidad y van desde componentes geográficos integrables a otras aplicaciones hasta sistemas completos con todas las herramientas necesarias para analizar la información. Desde el Mapa Interactivo de Buenos Aires accesible públicamente para todos los ciudadanos a través de la URL <http://mapa.buenosaires.gob.ar> hasta un mapa interactivo embebible y personalizado que se integra al tablero de control del gabinete de ministros, la USIG ofrece soluciones geográficas tanto a través de la red MAN del GCBA como a través de Internet, empleando para ellos estándares internacionales y las más avanzadas tecnologías disponibles en el mercado.

13.2 Normalización de Direcciones

El procedimiento de normalización de direcciones tiene por objetivo unificar la escritura de direcciones con respecto a un callejero de referencia de forma tal que los nombres de las calles queden escritos utilizando la nomenclatura oficial. De esta manera, es posible evitar incongruencias en las bases producidas por distintas áreas de gobierno, resolver errores de escritura, criterios diferentes en la carga, etc. El normalizador de USIG utiliza los campos de calle, altura o dirección existentes en la base de información y los evalúa para intentar reescribirlos de forma normalizada. En caso de que no fuera posible realizar dicha normalización de forma automática, la aplicación identificará cada registro con una descripción del problema encontrado u ofrecerá un listado de posibles direcciones normalizadas en caso de tratarse de una ambigüedad.

Con el objetivo de facilitar la normalización de las direcciones en bases de datos o sistemas con respecto a la versión más actualizada del callejero de la ciudad, la USIG provee a todos de los organismos del Gobierno de la Ciudad de Buenos Aires diversas herramientas y servicios accesibles a través de diferentes estándares tecnológicos y plataformas de forma tal de satisfacer la más amplia gama de necesidades relacionadas con este problema. A continuación se listan los servicios y herramientas accesibles por terceros a través de los cuáles es posible normalizar direcciones:



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

13.2.1 Normalizador Batch Desktop

Es una aplicación de escritorio para Windows que permite normalizar una base de datos completa y corregir interactivamente registros de forma individual. Soporta archivos en formato CSV.

Url: <http://servicios.usig.buenosaires.gob.ar/normalizador/>

13.2.2 Web Services

Servicio web SOAP para desarrolladores accesible a través de Internet (previa autorización de acceso) que permite entre otras cosas, normalizar direcciones

Url: <http://ws.usig.buenosaires.gob.ar>

13.2.3 Normalizador de direcciones (JS)

Implementación Javascript del normalizador de direcciones de USIG que puede utilizarse en cualquier aplicación cliente. Usa Callejero Server.

Url: <http://servicios.usig.buenosaires.gob.ar/nd-js/>

Documentación: <http://servicios.usig.buenosaires.gob.ar/nd-js/1.0/doc>

13.2.4 Normalizador de Direcciones (Python)

Implementación Python del normalizador de direcciones de USIG que puede utilizarse en cualquier aplicación cliente. Usa Callejero Server.

Url:

http://servicios.usig.buenosaires.gob.ar/normalizador_direcciones/1.0.0/normalizador_direcciones-1.0.0.tar.gz

Documentación: http://servicios.usig.buenosaires.gob.ar/normalizador_direcciones/1.0.0/

13.3 Autocompleter de Lugares

El autocompleter de lugares es un componente que forma parte de USIG-JS y se integra fácilmente en cualquier aplicación web para facilitar el ingreso y normalización de direcciones de forma interactiva. Usa el Normalizador de direcciones (JS).

Url: <http://servicios.usig.buenosaires.gov.ar/usig-js/2.0/usig.AutoCompleterFull.min.js>

Demo: <http://servicios.usig.buenosaires.gob.ar/usig-js/2.0/demos/autoCompleter.html>

Documentación: <http://servicios.usig.buenosaires.gob.ar/usig-js/2.0/doc/>

13.4 Callejero Server

Es un servicio web para desarrolladores que permite obtener siempre la última versión del callejero en formato JSON. En el caso de que ninguno de los servicios anteriores satisfaga las necesidades del usuario, siempre es posible utilizar este servicio para implementar su propio normalizador.

Url: <http://servicios.usig.buenosaires.gob.ar/callejero>

13.5 Servicio de Geocodificación

La geocodificación es el proceso por el cual se obtiene una posición geográfica a partir de una dirección normalizada. La USIG posee una herramienta propia específicamente optimizada



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

para la Ciudad de Buenos Aires que permite obtener posicionamientos de altísima precisión pudiendo además personalizar el tipo de posicionamiento deseado.

Entre las diversas formas de acceder a este servicio se destacan los servicios web SOAP (ver más abajo) y el servicio de geocodificación basado en el estándar REST accesible a través de <http://ws.usig.buenosaires.gob.ar> que puede utilizarse directamente o bien a través del componente específico de USIG-JS (ver abajo).

13.5.1 Servicios Web SOAP

Los servicios web constituyen hoy en día un pilar fundamental para lograr la integración de distintas aplicaciones evitando la duplicación innecesaria de información y propiciando la unificación de las bases de datos. Con este objetivo la USIG pone a disposición de todos los organismos del Gobierno de la Ciudad de Buenos Aires una serie de servicios web SOAP accesibles a través de la url: <http://ws.usig.buenosaires.gob.ar>

Los servicios web SOAP son un estándar ampliamente soportado por todos los grandes proveedores y organizaciones de la industria incluyendo a IBM, Microsoft, Sun y el W3C, entre otros, y permiten compartir información en formato XML entre muy diversas plataformas de software y hardware. Por ejemplo, es posible crear un servicio web utilizando Java bajo Linux y alguien puede acceder a ese servicio con un cliente .NET bajo Windows de forma totalmente transparente.

Se encuentran disponibles consultas para conocer, por ejemplo, la forma normalizada de una dirección y su posición geográfica, a qué Comuna o Barrio corresponde.

13.6 Inventario de Información Geográfica de la Ciudad

Se trata de un sistema de administración de información geográfica masivo y unificado que es alimentado por las diversas áreas del GCBA competentes en la materia a que se refieren los objetos almacenados, bien a través de sus propios sistemas o a través de sistemas provistos por la USIG. Expone una serie de servicios basados en tecnologías estándares que permiten interactuar con este repositorio masivo tanto para realizar consultas alfanuméricas o geográficas como para administrar objetos georreferenciados de manera genérica. Este servicio está accesible a través de la url <http://inventario.usig.buenosaires.gob.ar>



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

14 Anexo II – Integración con OpenID o Directorio Activo GCABA

Para la integración con el proveedor de OpenID pueden utilizarse cualquiera de los clientes disponibles que implementan el protocolo OpenID. El listado de clientes de OpenID disponibles pueden consultarse en <http://openid.net/developers/libraries/>. A continuación se detalla una guía para implementar la integración:

Introducción

El presente documento está orientado a la implementación de BA ID como sistema de identificación en cualquier website de la ciudad. El sistema otorga identificación o registro de usuarios, datos básicos provistos bajo el estándar de OpenID y se proveen ejemplos de código para poder implementar rápidamente.

En caso de requerir mayor información se deberá utilizar OpenID Connect (todavía no implementado del lado servidor) y sólo se provee una API con Tipo y Número de Documento para los casos que lo requieran.

El ID de cada usuario es su correo electrónico.

Conexión

URL: <https://id.buenosaires.gob.ar/openid>

Datos

Campos accesibles por SREG

- email
- nickname
- fullname
- dob
- postcode
- country
- gender

Campos accesibles por AX

- <http://openid.net/schema/contact/internet/email>
- <http://openid.net/schema/namePerson/first>
- <http://openid.net/schema/namePerson/last>
- <http://openid.net/schema/namePerson/friendly>
- <http://openid.net/schema/birthDate/birthYear>
- <http://openid.net/schema/birthDate/birthMonth>
- <http://openid.net/schema/birthDate/birthday>
- <http://openid.net/schema/gender>
- <http://openid.net/schema/contact/phone/home>
- <http://openid.net/schema/contact/phone/cell>
- <http://openid.net/schema/contact/postaladdress/home>
- <http://openid.net/schema/contact/city/home>
- <http://openid.net/schema/contact/state/home>
- <http://openid.net/schema/contact/country/home>
- <http://openid.net/schema/contact/postalcode/home>

Ejemplos

Basados en el sitio oficial <https://github.com/openid/php-openid>

PHP

25/06/2014 1/2



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

<https://github.com/gcba/baid-openid-client-php>

NODE.js

Usamos el módulo Passport OpenID sin configuración particular, solo pasarle la URL de ID.

<http://passportjs.org/guide/openid/>

Python

(en desarrollo)

JAVA

(en desarrollo)

Ambientes

Desarrollo

URL: <http://id.dev.gcba.gob.ar>

HML

URL: <http://id.hml.gcba.gob.ar>

Extensión

API extendida para consulta de DNI

URL: <https://id.buenosaires.gob.ar/services/correo@dominio.com/documento>

Esta URL devuelve, si existe, el siguiente resultado:

```
{
  · tipo: "dni",
  · documento: "20202202"
}
```



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

15 Anexo III – Plataforma de Versionado y proceso de entrega

El proceso de entrega así como también el versionado del proyecto debe respetar lo especificado en el siguiente documento, que puede consultarse en la url http://intranet/asi/formularios_procedimientos/guia_procedimientos.php



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



ANEXO Control de Versiones - GIT



INSTRUCTIVO

PARA USO DEL GIT

Anexo

Gobierno de la Ciudad Autónoma de Buenos Aires

Página 1 de 4



Buenos Aires Ciudad

EN TODO ESTÁS VOS



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



ANEXO Control de Versiones - GIT



OBJETIVO:

El objetivo del presente documento es poder unificar las entregas en una sola herramienta de control de versiones o SVC (del inglés *System Version Control*), a través de un proceso y estructura definidos que permitan a las diferentes áreas de la ASI realizar las actividades e intercambios con los proveedores. Es de carácter obligatorio realizar este proceso para llevar a cabo las actividades de control de cambios tanto para aplicaciones nuevas como entregas de nuevas versiones de aplicaciones existentes.

Como aclaración informamos que éstas entregas no reemplazan la entrega formal por Mesa de Entradas en contrataciones licitadas, sino que es un medio adicional obligatorio para facilitar las tareas operativas.

ALCANCE:

El alcance del presente documento involucra a todos los Organismos que interectúan con la ASI, así como los proveedores del GCABA.

PROCEDIMIENTO

El sistema de control de versiones seleccionado por la ASI a utilizar es GIT el cuál pertenece al tipo de arquitectura de repositorios de información distribuidos.

Todo el código fuente necesario para el proyecto así como también para el esquema, sus migraciones y todo el material generado para el proyecto deberá estar presente en un repositorio creado a tales efectos.

Estructura de proyecto:

Cada aplicación debe contar con su carpeta correspondiente y la estructura definida.

NOMBRE PROYECTO

- build/
- database/
 - database.sql
 - migrations
 - version-name-(orden).sql
 - ...
- source/
- documents/
- Instructivos-Documentos ASI/
- scripts/ (opcional) ej: Scripts para rollback, Scripts TestServices.sh
- tests/
- patches/
- .gitignore
- .gitmodules (opcional)
- Makefile (opcional)
- README.md
- CHANGELOG.md
- requirements.txt

En la Carpeta **Build/** se deben incluir todos los binarios compilados en caso de tratarse de un

Página 2 de 4





Buenos Aires Ciudad

EN TODO ESTÁS VOS



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



ANEXO Control de Versiones - GIT

lenguaje compilado.

En la carpeta **Documents/** incluir en cada entrega la documentación relacionada a la versión o el proyecto que se encuentra definida como obligatoria en el Estándar de Desarrollo (Ej.: documento de arquitectura, casos de prueba, manuales, etc.). Para realizar la primera instalación es imprescindible contar con el Manual de Instalación completo y el Documento de Arquitectura (Solicitar los template a la ASI como referencia)

En la carpeta **source/** debe estar contenido todo el código fuente de la aplicación y el archivo de configuración de dependencias, el cuál debe listar las mismas especificando el número de versión exacta para cada una de ellas.

En caso de necesitar aplicar algún tipo de parche al framework utilizado los mismos deberán alojarse dentro de la carpeta patches de la raíz del repositorio.

En el directorio raíz del repositorio deberá encontrarse un archivo **README.md** el cual debe contener la información necesaria para realizar la implementación del aplicativo.

Luego, para cada nueva versión que se entrega es obligatorio sumar al archivo **README.md** las instrucciones para llevar a cabo la instalación, indicando la fecha en la que se produjo dicha actualización y los cambios que se realizaron en el archivo **CHANGELOG.md**, incluyendo en el mismo lo siguiente:

- Nro de tickets de bugs resueltos (nro asociado en la Herramienta de seguimiento y control de cambios definidas en el proyecto)
- Funcionalidades incluidas (si corresponde),
- Sprint correspondiente (versión preliminar).

Cada versión deberá estar etiquetada con correspondiente nombre con el formato **"vx.x.x"**.


Quando el Referente del proyecto de la ASI tiene la aprobación (del Usuario y QA si interviene en le proceso) de la versión pasar de ambiente, el responsable de la entrega deberá etiquetar en el GIT con el formato **"r vx.x.x"**. Al momento de realizar el pedido de instalación por la vía formal establecida por la ASI (actualmente es la herramienta Mantis) debe informarse la última etiqueta con formato de RELEASE.

Si una versión aprobada llegara a producción y la misma contuviese errores, el Referente de proyecto en conjunto con el Equipo de Desarrollo (Interno o Proveedor) deberán corregir dichos errores, quitar la etiqueta de release de ese commit, agregar al mismo la etiqueta de **"rf vx.x.x"** y crear una nueva etiqueta release en el commit donde los mismos estuvieran corregidos. Hecho esto se deberá enviar la versión generada nuevamente con su correspondiente pedido.

Es obligatorio contar en la carpeta **scripts**, con los comandos necesarios para efectuar un rollback en caso de que el deploy pudiera fallar. Este script deberá efectuar tanto rollback de código como de base de datos y configuración. Es responsabilidad del desarrollador mantener la lógica necesaria para evitar que se pierda información valiosa en este proceso. Todos los script deben contar con una forma adicional que verifique y valide que la ejecución es exitosa o no, a modo de validación como por ejemplo que contenga las instrucciones que logueen en un log que pueda ser rescatado y enviado a los interlocutores.

Adicionalmente debe ser utilizado el comando de mayor nivel de información (verbose)

En caso de existir integraciones/acoplamiento con servicios externos al paquete a instalar debe contarse con una forma de verificar que la instalación relacionada se comunica correcta o

Buenos Aires Ciudad



Página 3 de 4

EN TODO ESTÁS VOS



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



ANEXO Control de Versiones - GIT

incorrectamente y es obligatorio detallar la forma de integración en el documento de arquitectura.

Los ambientes donde se implementan las entregas no contemplan accesos a servicios externos, por lo tanto cualquier excepción debe estar documentada, justificada y la ASI puede aprobar o no el acceso. Para no generar reworking solicitamos anticipar este tipo de necesidad al comienzo del proyecto.

Si se considera que el deploy a realizar puede poner en riesgo la información almacenada en la base de datos del proyecto se deberá indicar al área de infraestructura la realización de un backup de los datos, previo a la realización del mismo, en el **README.md** y en la solicitud del pedido del pasaje de ambiente.

Debe existir una rama estable (**master**) que es la que siempre contiene las versiones entregadas a la ASI y se recomienda generar una rama de desarrollo (**dev**) para desarrollo.

En el archivo **.gitignore** dentro del repositorio deberán incluirse los archivos de configuración y todos aquellos archivos que no formen parte del proyecto y que por alguna razón existan en el directorio del mismo (Ej.: archivos de sistema, configuración, archivos subidos por usuarios, etc.). En la carpeta **Tests/** incluir los casos de pruebas funcionales y no funcionales y los informes con la evidencia de los resultados.

El archivo de dependencias deberá estar incluido dentro de la carpeta source con todas las dependencias externas necesarias para el correcto funcionamiento de la aplicación. A continuación se enumeran los ejemplos para las plataformas reguladas por la ASI:

- requirements.txt para python
- package.json para nodejs
- pom.xml para java
- composer.js para php

Alta de Usuario


En caso de no tener un usuario de la herramienta, el Referente técnico de la ASI realizará las gestiones necesarias para brindarle uno con los permisos necesarios así como la creación de la estructura de carpeta correspondiente.

CONFIGURACION

A continuación una guía de los pasos necesarios para vincular un puesto al GIT GCBA:

1. Acceder a la URL <http://git-asi.buenosaires.gob.ar/help/ssh> para generar la Key de ssh
2. Una vez que se cuenta con la Key, agregarla en el profile del usuario del GIT.
<http://git-asi.buenosaires.gob.ar/profile/keys>
3. Ejecutar los comandos para clonar el proyecto que la ASI creó para tal fin e informó
4. Subir los archivos respetando el contenido en las carpetas definidas.

Página 4 de 4

**Buenos Aires Ciudad**

EN TODO ESTÁS VOS



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

16 Anexo IV – Documentación Modelo

La documentación solicitada debe estar completada con los modelos de referencia detallados a continuación que pueden consultarse en:

http://intranet/asi/formularios_procedimientos/guia_procedimientos.php

En el caso de los restantes entregables pueden utilizar un formato libre pero deben respetar el contenido exigido en el presente estándar.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

16.1 Modelo Documento de Arquitectura



I

<Nombre del proyecto>
Documento de Arquitectura de Software
Versión <x.x>
xxxxxxxx de 2013



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Historia del documento

Versión	Descripción de los cambios	Fecha	Cambiado por:	Revisado por:



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Tabla de contenidos

1	INTRODUCCIÓN	4
1.1	DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	4
1.2	REFERENCIAS	4
2	REQUERIMIENTOS ESPECIALES	4
3	ARQUITECTURA TECNOLÓGICA	4
4	ARQUITECTURA DE LA APLICACIÓN	4
4.1	MODULOS DEL SISTEMA	4
4.2	VISTA DE ALTO NIVEL	4
4.3	DIAGRAMA DE CAPAS	4
4.4	DIAGRAMA DE CONTEXTO	4
4.5	DIAGRAMA DE CLASES	4
4.6	TRANSACCIONALIDAD DE BASE DE DATOS	4
4.7	INTERFACES	4
4.8	DECISIONES TECNOLÓGICAS DE BASE	4
4.9	PATRONES DE DISEÑO	4
4.10	TECNOLOGÍAS Y CRITERIOS DE DISEÑO	4
4.11	PLATAFORMA UTILIZADA	4
4.12	LENGUAJE Y FRAMEWORK DE APLICACIÓN	4
4.13	MOTOR Y FRAMEWORK DE BASE DE DATOS	4
4.14	LENGUAJE Y FRAMEWORK DE VISTA	4
4.15	DIAGRAMAS DE SECUENCIAS	4
5	LOGS Y AUDITORIA	4
6	MANEJO DE ERRORES	4
6.1	ERRORES ESPERADOS	4
6.2	ERRORES INESPERADOS	4
6.3	SERVICIOS	4
6.3.1	Servicios que expone	4
6.3.2	Servicios que consume	4
7	MODELOS DE DATOS	4
7.1	MODELO DE ENTIDAD-RELACIÓN	4
7.2	MODELO DE DATOS FÍSICO	4
7.2.1	Diccionario de datos	4



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



<PROPOSITO DE ESTE DOCUMENTO

El Documento de Arquitectura de Software (DAS), tiene como propósito describir el sistema en forma resumida pero detallada. Se espera que el DAS provea información complementaria al código fuente y documentos funcionales.

Es una especificación de las ideas principales del diseño. El DAS proporciona una descripción entendible de la arquitectura del sistema y sirve como medio de comunicación entre el arquitecto y otros miembros de equipo del proyecto con respecto a las decisiones arquitectónicamente significativas que se han tomado. El documento de arquitectura debe contener información complementaria al código y debe describir lo que el código de arquitectura no hace. No debe repetir información, en caso que la información se encuentre en otro documento sólo debe hacer referencia a la fuente.>



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



1 Introducción

<Explicación general sobre el sistema a desarrollar, necesidades de negocio y objetivos a lograr con su implementación...>

1.1 Definiciones, acrónimos y abreviaturas

<Glosario con todos los términos propios del negocio y/o abreviaturas usados a lo largo de este documento>

Término	Definición

1.2 Referencias

<Otros documentos del proyecto a los que se hace referencia dentro de este documento o sirven como información adicional de contexto (ej: documentos de alcance, requerimientos, de casos de uso, etc.)>

ID Referencia	Tipo de archivo	Nombre del archivo

2 Requerimientos especiales

<Descripción de los principales requerimientos funcionales y no funcionales que afectan la arquitectura del sistema. Debe mantener relación y hacer referencia al documento de requerimientos.>

3 Arquitectura tecnológica

<Debe incluir la descripción de la plataforma tecnológica utilizada indicando las versiones: sistema operativo, motor de base de datos, servidor de aplicación, etc.>

4 Arquitectura de la aplicación

4.1 Módulos del sistema

<Explicación de la descomposición del sistema en módulos o subsistemas y las dependencias entre ellos. Incluir diagrama.>

<Nombre del proyecto> – Documento de Arquitectura

Versión <X.X>

Pág.: 5 / 7



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



4.2 Vista de alto nivel

<Diagrama (y su explicación) con los principales módulos estructurales del sistema, diferentes capas propuestas y los componentes de cada una.>

4.3 Diagrama de capas

4.4 Diagrama de contexto

4.5 Diagrama de clases

4.6 Transaccionalidad de base de datos

4.7 Interfaces

4.8 Decisiones tecnológicas de base

<Tabla con las decisiones tecnológicas de base del sistema y su justificación.>

Capa	Servicio	Producto / Tecnología	Descripción

<Diagrama de alto nivel con los distintos servicios, sistemas, componentes externos que interactúan con el sistema. Ej: balanceadores de carga, servidores de email, DBMS, LDAP, otros sistemas>

4.9 Patrones de diseño

<Sólo para sistemas Java. Principales patrones de diseño implementados en el sistema.>

4.10 Tecnologías y criterios de diseño

<Descripción detallada de las decisiones tecnológicas mostradas en el punto 4.8.>

4.11 Plataforma utilizada

4.12 Lenguaje y framework de aplicación

4.13 Motor y framework de base de datos

4.14 Lenguaje y framework de vista

4.15 Diagramas de secuencias

<Diagramas de secuencias para las principales funcionalidades del sistema.>



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



5 Logs y auditoria

6 Manejo de errores

6.1 Errores esperados

6.2 Errores inesperados

6.3 Servicios

6.3.1 Servicios que expone

<Descripción, definiciones funcionales e información de intercambio de los servicios web que expone el sistema. Mapeo entre los servicios y los requerimientos que los solicitan. Debe complementarse con los documentos técnicos de WS.>

6.3.2 Servicios que consume

<Descripción, definiciones funcionales e información de intercambio de los servicios web que consume el sistema. Mapeo entre los servicios y los requerimientos que los solicitan.>

7 Modelos de datos

7.1 Modelo de entidad-relación

<Diagrama representando las entidades relevantes del sistema, sus atributos, sus interrelaciones y la cardinalidad entre ellas.>

7.2 Modelo de datos físico

7.2.1 Diccionario de datos

Esquema	Tabla	Campo	Formato	Atributo	Descripción

<Nombre del proyecto> – Documento de Arquitectura

Versión <x.x>

Pág.: 7 / 7



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

16.2 Modelo Documento de Relevamiento



I

<Nombre del proyecto>

Versión <x.x>
xxxxxxxx de 20xx



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Historia del documento

Versión	Descripción de los cambios	Fecha	Cambiado por:	Revisado por:

<Nombre del proyecto> – Relevamiento inicial

Versión <X.X>

Pág.: 2 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Tabla de contenidos

1	INTRODUCCIÓN	4
1.1	DATOS DEL SOLICITANTE	4
1.2	DATOS DEL REFERENTE	4
1.3	DATOS DE OTRAS PERSONAS RELACIONADAS CON EL PROYECTO	4
1.4	SITUACIÓN ACTUAL	4
1.5	RIESGOS	4
1.6	PLAZOS	4
2	OBJETIVO	4
3	ALCANCE	4
4	USUARIOS	4
5	INTERFACES CON OTROS SISTEMAS	4
6	PRINCIPALES REQUERIMIENTOS	4
6.1	REQUERIMIENTO 1	4
6.2	REQUERIMIENTO 2	4
6.3	REQUERIMIENTO N	4

<Nombre del proyecto> – Relevamiento inicial

Versión <x.x>

Pág.: 3 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



1 Introducción

1.1 Datos del solicitante

<Datos de la persona que hizo la solicitud: Nombre, apellido, repartición, cargo, dirección laboral, teléfono, celular, email>

1.2 Datos del referente

<Datos de la persona que va a ser el referente del proyecto. Es la persona que va a realizar las definiciones del proyecto, determinar prioridades y tomar las decisiones solicitadas por el equipo de la ASI: Nombre, apellido, repartición, cargo, dirección laboral, teléfono, celular, email>

1.3 Datos de otras personas relacionadas con el proyecto

<Explicación sobre cuál es la necesidad a satisfacer, quién solicitó el sistema.

1.4 Situación actual

<Situación actual: existe sistema? Si existe sistema, solicitar documentación.

Cómo es el proceso? Está definido por alguna reglamentación? Cuáles son las principales falencias en la actualidad>

1.5 Riesgos

<Cuáles son los riesgos del proyecto?>

1.6 Plazos

<Existe algún plazo en la solicitud? Por qué motivo está dado este plazo?>

2 Objetivo

<Objetivo del proyecto>

3 Alcance

<Alcance>

3.1 Fuera del alcance

4 Usuarios

<Cantidad de estimada de personas que van a utilizar el sistema, actores del sistema, concurrencia, sensibilidad de la información,

<Nombre del proyecto> – Relevamiento inicial

Versión <X.Y>

Pág.: 4 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



5 Interfaces con otros sistemas

<Qué interfaces son necesarias con sistemas externos>

6 Principales Requerimientos

6.1 Requerimiento 1

6.2 Requerimiento 2

•
•
•

6.3 Requerimiento n



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

16.3 Modelo Documento de Especificación de Requerimientos



Especificación de Requerimientos

<Nombre del Proyecto>>



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Tabla de Contenidos

1	INTRODUCCIÓN	2
1.1	Objetivo del Documento	2
1.2	Destinatarios	2
1.3	Alcance	2
1.4	Documentos relacionados	2
1.5	Actores	2
1.6	Sistemas e Interfaces	2
2	ARQUITECTURA FUNCIONAL	3
3	REQUERIMIENTOS FUNCIONALES	3
3.1	Módulo << Nombre del módulo >>	3
3.1.1	Introducción	3
3.1.2	Requerimientos	3
3.2	Módulo << Nombre del módulo >>	3
3.2.1	Introducción	3
3.2.2	Requerimientos	3
4	REQUERIMIENTOS NO FUNCIONALES	3
4.1.1	Introducción	3
4.1.2	Requerimientos	3
5	CASOS DE USO	3
5.1	Casos de Uso << Nombre del caso de uso >>	3
6	APÉNDICES	3
6.1	Glosario	3
6.2	Historia de Cambios	3

ERS <<Nombre Proyecto>>

Pág.: 1 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



1 INTRODUCCIÓN

1.1 Objetivo del Documento

Este documento contiene la Especificación de Requerimientos del Software (ERS) destinado al <<Proyecto - indicar el nombre del proyecto>>.

1.2 Destinatarios

En esta sección se indica a cuales personas o roles está dirigido este documento.

Lector	Sector o Rol

1.3 Alcance

<<Describir el objetivo del sistema, indicando lo incluido y lo no incluido en el mismo.>>

1.4 Documentos relacionados

En esta sección se indican todos los documentos entregados por el cliente relacionados al presente documento.

Documento	Carpeta	Nombre Documento

1.5 Actores

A continuación se especifican los actores/usuarios de la aplicación:

Actor	Descripción

1.6 Sistemas e Interfaces

A continuación se listan los sistemas con los cuales interactúa la aplicación:

ERS <<Nombre Proyecto>>

Pág.: 2 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



2 ARQUITECTURA FUNCIONAL

En esta sección se presentarán gráficamente y en forma funcional como se relaciona el sistema << nombre del proyecto >> con otras aplicaciones, servicios, los canales de uso, etc. Este mapa debe representar gráficamente una visión global.

3 REQUERIMIENTOS FUNCIONALES

En esta sección se presentarán agrupados en módulos los requerimientos funcionales para el sistema << nombre del proyecto >>. Dentro de cada módulo se podrá observar un marco introductorio y las funcionalidades incluidas en éste.

3.1 Módulo << Nombre del módulo >>

3.1.1 Introducción

Este módulo tiene por finalidad describir las funcionalidades << describir la funcionalidad del módulo >>.

3.1.2 Requerimientos

ID	Requerimiento	Descripción

3.2 Módulo << Nombre del módulo >>

3.2.1 Introducción

Este módulo tiene por finalidad describir las funcionalidades << describir la funcionalidad del módulo >>.

3.2.2 Requerimientos

ID	Requerimiento	Descripción

4 REQUERIMIENTOS NO FUNCIONALES

4.1.1 Introducción

En esta sección se enumeran los requerimientos no funcionales que la aplicación debe tener en cuenta.

4.1.2 Requerimientos

ID	Requerimiento	Descripción
----	---------------	-------------

ERS <<Nombre Proyecto>>

Pág.: 3 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



5 CASOS DE USO

NOTA: en esta sección se deben referenciar los casos de uso que surgen del análisis detallado de cada uno de los requerimientos funcionales. Incluir los mismos a medida que se van especificando y acordando con el usuario.

5.1 Casos de Uso << Nombre del caso de uso >>

<<link al documento de especificación de casos de uso >>

6 APÉNDICES

6.1 Glosario

Término	Descripción
Término 1	
Término 2	

6.2 Historia de Cambios

Revisión	Autor	Fecha	Descripción

ERS <<Nombre Proyecto>>

Pág.: 4 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

16.4 Modelo Documento de Casos de uso

Se puede usar un único documento de casos de uso o uno por cada caso de uso. En caso de elegir la segunda alternativa, se deberá además incluir un documento de especificaciones funcionales que relacione todos los casos de uso y brinde la información de contexto que aparece en el modelo mostrado a continuación.



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Especificación de Caso de Uso

<Id. del CU>-<Nombre del Caso de Uso>
<Nombre del Proyecto>



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Tabla de Contenidos

1	INFORMACIÓN GENERAL *	2
2	FLUJO BÁSICO *	3
3	FLUJO ALTERNATIVO *	3
4	REQUERIMIENTOS NO FUNCIONALES DEL CASO DE USO	3
5	INFORMACIÓN RELACIONADA	3
6	DISEÑO CONCEPTUAL DE INTERFACES *	3
6.1	<Interfaz 1>	3
6.2	<Interfaz 2>	3
7	MENSAJES DEL SISTEMA *	3
8	APÉNDICES *	3
8.1	Glosario	3
8.2	Historia de Cambios *	3

<Id del CU>-<Nombre del Caso de Uso>

Pág.: 1 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



1 INFORMACIÓN GENERAL *

Nombre del Caso de Uso*	<Asignar un Nombre al caso de uso, que identifique el objetivo, escrito como frase verbal. El nombre debe incluir un identificador único del Caso de uso: <Id_CU>--<Nombre Caso Uso> Ejemplo: CU001_Dar de alta Cliente
Objetivo*	< Completar con un Objetivo y/o Descripción de lo que el/los actor/es intentan satisfacer en este caso de uso >
Trigger*	<Indicar el evento que dispara el caso de uso>
Actores*	<Indicar el Nombre del rol de quienes interactúan con el sistema en este Caso de Uso>
Pre-condiciones*	<Estado inicial del sistema antes de ejecutar el caso de uso. Por ej: el usuario debe estar identificado y autenticado en el sistema>
Post-condiciones Exitosas*	<Estado final del sistema luego de ejecutar el caso de uso en forma exitosa únicamente. Por ej: se registra la venta, el sistema calcula los impuestos y emite un ticket>
Prioridad	<Orden en el cual la funcionalidad del caso de uso es requerida>
Nombre del caso de uso extendido	<Si se trata de un caso de uso que extiende a otro, indicar acá el nombre del caso de uso que este caso de uso extiende >
Punto de extensión	<Si se trata de un caso de uso que extiende a otro, indicar acá el número de paso del flujo donde éste caso de uso se inserta >
Condición de extensión	< Si se trata de un caso de uso que extiende a otro, indicar la condición que hace que esta extensión se inserte en el caso de uso extendido>

<Id del CU>--<Nombre del Caso de Uso>

Pág.: 2 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



2 FLUJO BÁSICO *

<FLUJO BÁSICO: Actividades (pasos) que ocurren entre el actor y el sistema para lograr obtener el objetivo.>

Este escenario describe los pasos que se realizan desde que se dispara el evento Trigger hasta la obtención de cumplimiento del objetivo planteado para el Caso de Uso siempre que no ocurra una falla en su ejecución.

1. El vendedor ingresa el Id de artículo. *
2. El sistema muestra la descripción del artículo.
3. El vendedor ingresa la forma de pago en efectivo.
4. El vendedor confirma la venta.
5. El sistema registra el pedido, calcula los impuestos y emite el ticket.

3 FLUJO ALTERNATIVO *

<FLUJOS ALTERNATIVOS: Extensiones producidas por alguna alteración en los pasos del Flujo Principal. Ver el ejemplo en la tabla para el paso 3 del ejemplo anterior.>

Los flujos alternativos muestran cómo puede extenderse el escenario principal.

3. a El Sistema detecta que la forma de pago es con tarjeta:
 - 3.a.1 El vendedor ingresa la forma de pago con tarjeta.
 - 3.a.2 El vendedor ingresa el nro. de tarjeta, la fecha de vencimiento y el código de seguridad.
 - 3.a.3 El sistema valida el límite de la tarjeta.
 - 3.a.3 El sistema registra la venta y emite el comprobante de tarjeta.

4 REQUERIMIENTOS NO FUNCIONALES DEL CASO DE USO

<Requerimientos de performance, seguridad, etc. que deben satisfacerse en este caso de uso >

Requerimientos No Funcionales asociados al caso de uso, importantes de tener en cuenta para su desarrollo.

1. La clave de usuario debe cambiarse cada 30 días por normativa del BCRA.

5 INFORMACIÓN RELACIONADA

<Información de contexto del caso de uso >

Documento	Ubicación

<Id del CU>-<Nombre del Caso de Uso>

Pág.: 3 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



6 DISEÑO CONCEPTUAL DE INTERFACES*

<Diseño conceptual de las interfases que reflejan la funcionalidad del caso de uso>

6.1 <Interfaz 1>

Campo	Tipo	Obligatoriedad	Validación	Valor Default	Descripción

6.2 <Interfaz 2>

Campo	Tipo	Obligatoriedad	Validación	Valor Default	Descripción

7 MENSAJES DEL SISTEMA*

<Código y descripción de los mensajes relacionados con el caso de uso actual>

8 APÉNDICES*

8.1 Glosario

Término	Descripción
Término 1	
Término 2	

8.2 Historia de Cambios *

Revisión	Autor	Fecha	Descripción
*	*	*	*

* Datos Obligatorios

** Datos Obligatorios si no es un Caso de Uso de Procesos

<Id del CU>-<Nombre del Caso de Uso>

Pág.: 4 / 5



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

16.5 Modelo Documento Manual de Instalación



I

**<Nombre del sistema>
Manual de Instalación**
<Versión>
<Fecha de creación>

<Nombre del sistema> – Manual de Instalación
<Versión>
Pág.: 1 / 6



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



Tabla de contenidos

1	INTRODUCCIÓN	3
1.1	DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	3
1.2	REFERENCIAS	3
2	REQUERIMIENTOS DEL SOFTWARE	3
3	ESPECIFICACION DEL SISTEMA OPERATIVO	3
4	MODULOS REQUERIDOS	3
5	CONFIGURACIÓN DE PARÁMETROS	3
6	ESPECIFICACION DE LOS DIRECTORIOS DE TRABAJO	3
7	INSTALACION LIBRERIAS DE TERCEROS	3
8	CONFIGURACIÓN DE LA BASE DE DATOS	3
8.1	ACTUALIZACIONES DE LA BASE	3
8.2	CREACIÓN DE DATOS DE PRUEBA	3
9	CONFIGURACIÓN DE AMBIENTES	3
10	CONFIGURACIONES DEL SERVIDOR WEB	3
11	CONFIGURACION DE LOGS	3
12	CONFIGURACIÓN DEL ENVÍO DE MAILS	3
13	CONFIGURACION DE SERVICIOS WEB	3

<Nombre del sistema> – Manual de instalación

<Versión>

Pág.: 2 / 6



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



El documento tiene como objetivo dar una descripción de los pasos a llevar a cabo para realizar la instalación de un sistema aplicativo en los distintos ambientes de ejecución. Proveerá información correspondiente configuración de servidores, tecnologías utilizadas, configuración de la base de datos y los servicios utilizados tales como servicio de mail y web services.

<Nombre del sistema> – Manual de instalación

<Versión>

Pág.: 3 / 6



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



1 Introducción

<Se da una descripción del alcance del sistema y en caso de que aplique se indican los siguientes datos >

1.1 Definiciones, acrónimos y abreviaturas

Término	Definición

1.2 Referencias

ID Referencia	Tipo de archivo	Nombre del archivo

2 Requerimientos del Software

<Se listan las distintas herramientas empleadas en la construcción del software, por ejemplo: servidor web, tecnología de desarrollo (php, java, etc.), motor de base de datos, etc>

Software requerido	Versión requerida	Comentarios

3 Especificación del sistema operativo

<Se describe el sistema operativo sobre el cual se llevara a cabo la instalación. Dependiendo de esto los demás puntos que se listan a continuación deberán indicar los comandos compatibles con el SO>

4 Módulos requeridos

<Se listan los distintos módulos o librerías, relacionados al lenguaje de desarrollo, que necesiten agregarse en el momento de instalación para el funcionamiento del aplicativo. Por ejemplo jars utilizados (significativos) en el caso de java o bien módulos activos en relación a PHP>

Modulo	Versión	Comentarios

5 Configuración de parámetros

<Nombre del sistema> – Manual de instalación

<Versión>

Pág.: 4 / 6



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



<En esta sección se listan los distintos parámetros a tener en cuenta que serán necesarios configurar para habilitar recursos propios de las tecnologías de desarrollo utilizadas. Por ejemplo si la aplicación está desarrollada en PHP y necesitamos indicar la zona horaria completaremos la tabla en el siguiente orden:

php.ini: date.timexone America/Argentina/Buenos_Aires comentarios>

Configuración	Valor	Comentario

6 Especificación de los directorios de trabajo

<Se realiza una descripción de los directorios de trabajo. Si se debe crear alguno en especial y los permisos de acceso con los respectivos comandos a ejecutar>

7 Instalación librerías de terceros

<Se listan los pasos para llevar a cabo la instalación de las librerías propias y de terceros necesarias>

8 Configuración de la base de datos

<Se debe especificar el motor de base de datos a utilizar y su configuración, los pasos para la creación de la base así como la creación de los usuarios y toda acción necesaria para la comunicación de la base con la aplicación>

8.1 Actualizaciones de la base

<Se realiza una descripción así como seguimiento de las actualizaciones que se realizan sobre la base. Se indican los scripts a ejecutar, la cantidad y el orden en que deben ser ejecutados>

8.2 Creación de datos de prueba

<Se describe el modo de carga de datos iniciales sobre la base de datos, scripts a ejecutar, comandos a ejecutar, archivos a importar, etc.>

9 Configuración de ambientes

<Se describen los pasos para la configuración de los distintos ambientes sobre los que se podrá desplegar la aplicación. Entre dichos pasos podemos encontrar configuraciones de conexión a las bases, de parámetros, etc. Básicamente se diferenciarán estos aspectos entre los ambientes de trabajo>



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"



10 Configuraciones del servidor web

<Se realiza una descripción de los pasos para la configuración del mismo así como archivos que serán necesarios modificar para la instalación del sistema y los recursos asignados. Por ejemplo eliminación de la cache, modificación de parámetros de ejecución, etc.>

11 Configuración de Logs

<Se realiza una descripción con los pasos para la configuración necesaria en el reporte de logs>

12 Configuración del envío de mails

<Se procede a indicar si el sistema contara con el servicio de envío de mails. Además, se indicaran todas las configuraciones relacionadas para habilitar el servicio>

13 Configuración de servicios web

<Se procede a indicar si el sistema contara con entrada o salida de servicios web. Se listaran todos aquellos servicios implicados y se detallaran las acciones para proceder con las configuraciones de los mismos>

14 Actualizaciones

<Se describen los pasos para realizar el backup de la base de datos y el estado actual del aplicativo así como las acciones necesarias para llevar a cabo el proceso de rollback en caso de que lo amerite. Luego se indica el proceso de actualización con todo lo que implica, ejecución de scripts, actualización de la base, actualización del código, etc.>

Historia del documento

<Se incluyen todos los datos que figuran en la tabla. Esto aplica para el momento de creación del documento así como todas sus actualizaciones posteriores>

Versión	Descripción de los cambios	Fecha	Cambiado por:	Revisado por:

<Nombre del sistema> – Manual de instalación

<Versión>

Pág.: 6 / 6



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

16.6 Modelo documento Servicios Web

El documento de servicios web debe incluir la siguiente información de todos los servicios web que expone el sistema.

16.6.1 Descripción de Servicios Web

Esta plantilla ofrece información general sobre la existencia de un servicio web desde un punto de vista global.

Esta es la plantilla básica.

Denominación del WS	<i>Nombre clave con el cual es conocido el WS</i>
Descripción del WS	<i>Detalle sobre el objetivo y funcionalidad del mismo desde una perspectiva global.</i>
Tecnología	<i>SOAP/REST/RPC/etc.</i>
Transporte utilizado	<i>Definir por medio de qué protocolos de transporte está definido el WS. HTTP/HTTPS/SMTP/TCP/UDP/JMS/etc.</i>
Seguridad	<i>Definir si implementa algún mecanismo de seguridad. Abierto/ Mecanismo de Autenticación/Encriptación de Información./Adhiere a los estándares WS-Security 1.0/WS-Security 1.1/etc.</i>
Clientes del WS	<i>Especificar quienes son los clientes que utilizan el WS. Ciudadano (Abierto al público en gral)/Usuarios de gobierno(especificar áreas)/Empresas(especificar cuáles)/Sistemas(especificar cuáles)</i>
Ámbito de cobertura	<i>Especificación de la cobertura del servicio. MAN/Internet/etc.</i>
Cantidad de usuarios	<i>Número estimado de usuarios que consumen el WS</i>
Cantidad de transacciones	<i>Número de consultas estimadas que se efectúan sobre el WS por unidad de tiempo.</i>
Acuerdo de servicio	<i>Especificación de disponibilidad del WS para su consumo. 7x24hs/5x8hs/etc.</i>
Dirección de acceso/WSDL	<i>URL de consulta al WS</i>
Lenguaje desarrollado del WS	<i>Especificación sobre el lenguaje en qué fue desarrollado el mismo. PHP/Java/.Net/etc</i>
Librerías utilizadas	<i>Especificación de las librerías adicionales utilizadas para el desarrollo del mismo. Apache AXIS/Apache AXIS2/NuSOAP/etc.</i>

16.6.2 Funciones del Servicio Web

Esta plantilla ofrece información específica sobre las funciones y los valores de ingreso y egreso que ofrece un servicio web.

Esta plantilla es complementaria a la especificada anteriormente (básica).



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

Denominación del WS	<i>Nombre clave con el cual es conocido el WS</i>
Dirección de acceso al WS	<i>URL de consulta al WS</i>
Método {Nombre 1}	<i>Descripción global de funcionalidad</i>
Parámetros de llamada	<i>Especificación de los parámetros de llamada al WS</i>
INPUT	<i>Especificación de la entrada de datos</i>
OUTPUT	<i>Especificación de la salida de datos y su transformación.</i>
...	...
Método {Nombre N}	<i>Descripción global de funcionalidad</i>
Parámetros de llamada	<i>Especificación de los parámetros de llamada al WS</i>
INPUT	<i>Especificación de la entrada de datos.</i>
OUTPUT	<i>Especificación de la salida de datos y su transformación.</i>

16.6.3 Elementos del Servicio Web

Esta plantilla ofrece detalles a nivel de los elementos de datos que son utilizados como input y output de los distintos métodos.

Elemento	<i>Nombre del elemento</i>
Versión	<i>Versión actual del elemento</i>
Estado	<i>Especificación del estado del elemento</i>
Versión anterior	<i>Especificación de la versión que reemplaza</i>
Fecha de lanzamiento	<i>Especificación de la puesta de producción del mismo.</i>
Nombre	<i>Nombre del elemento</i>
Descripción	<i>Descripción asociada al mismo.</i>
Tipo de elemento	<i>Tipo de dato</i>
Autoridad	<i>Organismo rector del elemento.</i>
Basado en	<i>Norma ISO 4217</i>
Validación	<i>Especificación de la validación.</i>
Valores	<i>Valores que puede tomar el elemento.</i>
Formato	<i>Especificación del formato del elemento.</i>
Valor Por defecto	<i>Especificación del valor por defecto</i>
XML	<i>Esquema XML asociado.</i>
Comentarios	<i>Comentarios asociados al mismo.</i>

Ejemplo de elementos del servicio web:

Elemento	<i>Monto</i>
Versión	<i>0.1</i>
Estado	<i>No Publicado</i>
Versión anterior	



Gobierno de la Ciudad Autónoma de Buenos Aires

"2014, Año de las letras argentinas"

Fecha de lanzamiento	
Nombre	<i>MONTO</i>
Descripción	<i>Todo campo que indique un valor monetario (Ej.: Monto de una asignación, pensión, plan, avaluo, sueldo, etc.)</i>
Tipo de elemento	<i>Tipo de dato</i>
Autoridad	<i>Organismo que corresponda</i>
Basado en	<i>Norma ISO 4217</i>
Validación	<i>Números del 0 al 9</i>
Valores	
Formato	<i>Monto : 25 caracteres justificados a derecha en donde el entero está representado del 1 al 23 y los decimales del 24 al 25 rellenando con ceros a la izquierda del primer dígito significativo y dejando en cero el 24 y 25 en el caso que se informe enteros. Ej: un mil: 000000000000000000000000100000, ciento veintitres con cuarenta y cinco: 00000000000000000000000012345</i>
Valor Por defecto	<i>0</i>
XML	<i><xsd:simpleType name="Monto"> <xsd:restriction base="xsd:string"> <xsd:pattern value="[0-9]{25}"/> </xsd:restriction> </xsd:simpleType></i>
Comentarios	



Gobierno de la Ciudad Autónoma de Buenos Aires
“2014, Año de las letras argentinas”

17 Anexo V – Flujograma Proceso Control de Cambios

CAMBI-010 Cambios en Software de Aplicación

