



## Municipalidad de Vicente Lopez

### Decreto

**Número:**

CIUDAD DE VICENTE LOPEZ

**Referencia:** EXPTE - EEMVL-949-2023

---

VISTO La necesidad de implementar estándares y buenas prácticas para los desarrollos de aplicaciones de software alojados en la infraestructura de la Municipalidad de Vicente López, y

CONSIDERANDO:

Que, es necesario definir un marco estandarizado para los desarrollos de aplicaciones de software cuyo alojamiento se realice, tanto on premise (local) como on cloud (en la nube), en la infraestructura de la Municipalidad de Vicente López.

Que, dicho marco estandarizado permitirá simplificar el mantenimiento de las aplicaciones de software, así como también garantizar el uso de tecnologías vigentes y con amplia penetración en el mercado, definir un flujo de trabajo que permita agilidad en los desarrollos y disminuya la discrecionalidad, garantizar la transparencia del código y su reusabilidad, permitir el control y trazabilidad de cada cambio a implementar tanto en nuevas aplicaciones como en aplicaciones existentes (cambios evolutivos) y contemplar un adecuado ciclo de vida del software y una documentación adecuada del mismo.

Que, el estándar se aplica en todos los casos donde la Municipalidad de Vicente López o proveedores contratados por la misma desarrollen y/o provean aplicaciones y/o servicios digitales con financiación pública.

Que, todo personal involucrado en el desarrollo o que en función de sus tareas dentro del proyecto de desarrollo, implementación o mantenimiento de Software deba tener contacto con Información, Datos (Información de Bases de Datos, o sus estructuras, Datasets, etc) o Procesos, deberá firmar un acuerdo de confidencialidad, que le proporcionará el Municipio de Vicente López mediante el cual se garantice el correcto tratamiento y confidencialidad de los datos.

Que, los datos recabados por el proveedor para poder brindar su servicio son propiedad del Municipio y por lo tanto éste último deberá tener acceso a los mismos en todo momento, cumpliendo con todos los criterios de **Disponibilidad**: perfectamente legible en lenguaje natural ("clear text" humano), **Integridad**: con toda la información completa que refleje en todas sus estructuras, tablas y sub-tablas, registradas en un mismo momento temporal y el último estado de actualización al momento del pedido de información, **Interoperabilidad**, en formato tabular integral y con toda la estructura de base de datos (por, ej. Diagrama DER) , correctamente descripta, para permitir el diseño de ETL y/o desarrollos que se alimenten con dicha información, **Confidencialidad** , las exportaciones de datos que el Municipio de Vicente López (MVL) podrá solicitar, serán entregadas con el debido resguardo de dicha información, a la persona o equipo de personas designadas como líderes de proyecto o interlocutores válidos del lado de MVL, o en su defecto a la Dirección de Seguridad Informática, **No Repudio**, se firmará un recibo por la recepción de dicha información por parte de MVL. El plazo máximo para entregar dicha exportación de datos deberá ser de 72 hs.

Que, en consecuencia, la Subsecretaría de Modernización procedió a la creación del Reglamento de Estándares y Buenas Prácticas para el Desarrollo y Contratación de Aplicaciones, cuya aprobación se propicia a través del presente Decreto.

Que, en el citado documento se especifican los derechos y las obligaciones de cada una de las partes para llevar a cabo la correcta implementación del marco estandarizado.

Es por ello que LA INTENDENTA MUNICIPAL DE VICENTE LOPEZ, en uso de las atribuciones que le son propias

### **DECRETA**

Artículo 1º.- **APROBAR** el “Reglamento de Estándares y Buenas Prácticas para el Desarrollo y Contratación de Aplicaciones” que como ANEXO ÚNICO forma parte integrante de la presente medida.

Artículo 2º.- **DISPONER**, a través de la Dirección General de Compras y Licitaciones, y a partir de la publicación del presente, uso obligatorio del Reglamento que se aprueba en el artículo 1º, para todas aquellas contrataciones municipales cuyo objeto consista en el desarrollo de aplicaciones de software alojados en la infraestructura de la Municipalidad de Vicente López, on premise (local) así como on cloud (en la nube).

Artículo 3º.- **REFRENDEN** el presente el Sr. Jefe de Gabinete.

Artículo 4º.- Tomen conocimiento del presente Decreto todas las Secretarías de la Municipalidad de Vicente López y la Dirección General de Compras y Licitaciones.

Artículo 5º.- Dar al Registro Municipal de Decretos, cumplir, hacer saber y oportunamente archivar.

REGLAMENTO DE  
ESTÁNDARES Y BUENAS  
PRÁCTICAS PARA EL  
DESARROLLO Y  
CONTRATACIÓN DE  
APLICACIONES  
Municipalidad de Vicente López

## Introducción

El objeto de este documento es definir un marco estandarizado para los desarrollos de aplicaciones de software alojados en la infraestructura de la Municipalidad de Vicente López, *on premise* (local) así como *on cloud* (en la nube), con el fin de:

- Simplificar su mantenimiento.
- Garantizar el uso de tecnologías vigentes y con amplia penetración en el mercado.
- Definir un flujo de trabajo que permita agilidad en los desarrollos y disminuya la discrecionalidad.
- Garantizar la transparencia del código y su reusabilidad.
- Permitir el control y trazabilidad de cada cambio a implementar tanto en nuevas aplicaciones como en aplicaciones existentes (cambios evolutivos).
- Contemplar un adecuado Ciclo de Vida del Software y una documentación adecuada del mismo.

Es responsabilidad de las Secretarías y las diferentes áreas del Municipio de Vicente López velar por el cumplimiento del presente Estándar y la presentación de la documentación firmada de los **Anexo I, II y III** según corresponda al momento de efectuar toda adquisición de software y durante su desarrollo. A los efectos, el mismo deberá incluirse como condición *sine qua non* en los pliegos técnicos y condiciones particulares al momento de llevarse a cabo la contratación. La Dirección General de Compras y Licitaciones, previa adjudicación al proveedor deberá solicitar a la Subsecretaría de Modernización y/o la Dirección General de Procesos y Desarrollos Informáticos un Dictamen Técnico mediante el cual notificará su análisis en cuanto al cumplimiento de lo exigido en el presente Estándar.

Es competencia de la Subsecretaría de Modernización y la Dirección General de Procesos y Desarrollos Informáticos la actualización continua del presente Estándar, en el marco del surgimiento de avances tecnológicos, teniendo en cuenta como mínimo una revisión anual.

### Alcance

Este documento está dirigido a todas aquellas personas vinculadas a la implementación y/o provisión de software, sean estos desarrolladores, líderes de proyectos, testers, ingenieros de pre-venta de soluciones, entre otros y abarca tanto a empleados de la Municipalidad de Vicente López como proveedores externos, que desarrollen y/o provean aplicaciones para la Municipalidad y las alojen en la infraestructura de la misma (*on premise*) o en nube (*on cloud*).

### ¿En qué casos se aplica este estándar?

Este estándar se aplica en todos los casos donde la Municipalidad de Vicente López o proveedores contratados por la misma desarrollen y/o provean aplicaciones y/o servicios digitales con financiación pública.

### Aseguramiento de la Confidencialidad

Todo personal involucrado en el desarrollo o que en función de sus tareas dentro del proyecto de desarrollo, implementación o mantenimiento de Software deba tener contacto con Información (p.ej. detalle funcional o cronogramas de implementación), Datos (Información de Bases de Datos, o sus estructuras, Datasets, etc) o Procesos, deberá firmar un acuerdo de confidencialidad, ver **ANEXO I**, que le proporcionará el Municipio de Vicente López mediante el cual se garantice el correcto tratamiento y confidencialidad de los datos.

### Software as a Service

Para el caso de estas modalidades de desarrollo y/o contratación de soluciones de software, se deberá considerar especialmente el cumplimiento de los estándares ISO 27017 e ISO 27018.

El proveedor no podrá utilizar la explotación de los datos con fines comerciales, investigación, ni recreativos.

Se debe tener disponibilidad 100% de los datos en el momento que la Municipalidad lo requiera, esto puede ser solicitado mediante API key, web service o por acceso directo a la base de datos con las respectivas credenciales, en el nivel de granularidad y descripción documental de la estructura de dichas bases, adecuado para la comprensión y utilización por parte de la Municipalidad de Vicente López. El proveedor deberá mantener la plataforma disponible y operativa las 24hs de los 7 días de la semana, para que los usuarios puedan operar siempre sobre la misma.

En caso de generar, por parte del proveedor, actualizaciones sobre el software, debe darse un aviso previo de un mes, como mínimo, indicando la documentación de los cambios a implementar, de tal manera que desde el Municipio de Vicente López se puedan tener previstas las acciones necesarias para contener o mitigar cualquier situación producida por dicha actualización.

En caso de terminar la contratación del proveedor, el municipio deberá disponer de toda la información del ciudadano y sus operaciones en la plataforma del proveedor de manera accesible durante el tiempo necesario para que el Municipio, mediante los mecanismos descritos previamente, realice la migración de información necesaria para permitir la continuidad de servicios al ciudadano mediante otro proveedor o con un desarrollo interno, minimizando interrupciones sobre lo que se considerará un servicio público.

Más allá de tratarse de una contratación de Software como Servicio (SaaS), los datos recabados por el proveedor para poder brindar su servicio son propiedad del Municipio y por lo tanto éste último deberá tener acceso a los mismos en todo momento, cumpliendo con todos los criterios de **Disponibilidad**: perfectamente legible en lenguaje natural ("clear text" humano), **Integridad**: con toda la información completa que refleje en todas sus estructuras, tablas y sub-tablas, registradas en un mismo momento temporal y el último estado de actualización al momento del pedido de información, **Interoperabilidad**, en formato tabular integral y con toda la estructura de base de datos (por, ej. Diagrama DER) , correctamente descrita, para permitir el diseño de ETL y/o desarrollos que se alimenten con dicha información, **Confidencialidad** , las exportaciones de datos que el Municipio de Vicente López (MVL) podrá solicitar, serán entregadas con el debido resguardo de dicha información, a la persona o equipo de personas designadas como líderes de proyecto o interlocutores válidos del lado de MVL, o en su defecto a la Dirección de Seguridad Informática, **No Repudio**, se firmará un recibo por la recepción de dicha información por parte de MVL. El plazo máximo para entregar dicha exportación de datos deberá ser de 72 hs.

### **Ciclo de Vida de las Aplicaciones y definiciones de Aplicaciones Legacy**

Se entiende por aplicaciones legacy a aplicaciones que cumplan al menos 1 de los siguientes criterios: Con más de 5 años de antigüedad. Que no cuenten con soporte por parte de la empresa fabricante. Que presenten incompatibilidades con protocolos de red y chequeos de seguridad actuales (p.ej. compartir recursos mediante SMB-1, RLOGIN, FTP no cifrado, entre otros). Desarrollos que no estén en al menos 3 capas. Hardcodes de cualquier tipo (registro de contraseñas dentro del código fuente o en cadenas de conexión no cifradas), dependencia de direccionamientos de red fija, registros de usuarios. Aplicaciones de baja resiliencia a la conexión, (interfaces usuarias orientadas a la conexión). Uso de drivers de DB obsoletos. Uso de Perma-links (links URL estáticos), incapacidad de modificar y/o escalar configuraciones (ej., dificultad de agregar o cambiar formularios de reportes, nuevas tablas o fuentes de información), limitaciones de crecimiento horizontal. Aplicaciones no documentadas mediante el GIT Municipal (URL: [https://git.vicentelopez.gov.ar/users/sign\\_in](https://git.vicentelopez.gov.ar/users/sign_in) ) (código fuente + arquitectura de solución + funcional + manuales de usuario e implementación, como mínimo), o sin continuidad de mantenimiento de los creadores. Aplicaciones que no cumplan con los requisitos mínimos de Seguridad Informática, que deberán ser chequeados mediante testeos de vulnerabilidades, previos a la puesta en producción. Desarrollos que usen archivos clear-text para el uso base de datos unipersonales. Ej. Clipper/FoxPro etc., que graban datos en .dbf.

Las aplicaciones Legacy que hoy funcionan en el Municipio, que no cumplan con el presente estándar seguirán habilitadas, pero se deberá acordar una fecha de fin de su ciclo de vida y eventual reemplazo o actualización, luego de un análisis a detalle y evaluación de riesgo. La autoridad competente para dicho análisis y evaluación será la Subsecretaría de Modernización conjuntamente con la Dirección General de Procesos y Desarrollos Informáticos. Toda actualización o reemplazo en el futuro, deberá respetar los estándares de Desarrollo aquí plasmados.

### Generación de QR

Todos los desarrollos deben poder ajustarse al formato de código QR (Quick Response o respuesta rápida) que se encuentra en la librería <https://github.com/walletconnect/> y documentación en <https://docs.walletconnect.com/2.0/specs/>. Debiendo ser homologado con los desarrollos ya existentes en el Municipio de Vicente López y mostrando su validación a la Subsecretaría de Modernización. Conviniendo prestar conformidad de no utilizar o solapar campos utilizados por el BCRA (Banco Central de la República Argentina).

Por otro lado, para las implementaciones que requieran interoperabilidad utilizarán el estándar de código QR EMVCo publicado en Comunicación SINAP "A" 7153 (<http://www.bcra.gov.ar/pdfs/comytexord/A7153.pdf>) del BCRA.

En tal sentido el boletín SIMPRA 525 enuncia <http://www.bcra.gov.ar/pdfs/sistemasfinancierosydepagos/SNP3525.pdf> cómo gestionar la interfaz Estandarizada de Pagos (**ANEXO IV**).

### Estándares de tecnología

Todos los desarrollos deben utilizar las herramientas homologadas junto con las versiones estables al momento de su implementación.

Con el fin de limitar la variedad de tecnologías en la infraestructura de la Municipalidad de Vicente López y simplificar su mantenimiento este estándar propone como tecnologías aceptadas:

### Bases de datos y versiones

La Municipalidad de Vicente López, permitirá el uso de bases de datos tanto relacionales como no relacionales. A continuación, se listan las tecnologías habilitadas junto con la versión mínima aceptada.

Languages	Versiones aceptadas	Aclaraciones
Python	3.6 o superior	end of support v3.6 2021-12-23 <a href="https://devguide.python.org/#status-of-python-branches">https://devguide.python.org/#status-of-python-branches</a>
PHP	7.4 o superior	<a href="https://www.php.net/supported-versions.php">https://www.php.net/supported-versions.php</a>
Java (OpenJDK)	1.8 o superior	<a href="https://www.oracle.com/java/technologies/java-se-glance.html">https://www.oracle.com/java/technologies/java-se-glance.html</a> <a href="https://access.redhat.com/articles/1299013">https://access.redhat.com/articles/1299013</a> <a href="https://wiki.debian.org/Java">https://wiki.debian.org/Java</a>

Bases de datos	Versiones aceptadas	Aclaraciones
JavaScript jQuery	v3.6	<a href="https://en.wikipedia.org/wiki/JQuery">https://en.wikipedia.org/wiki/JQuery</a>
Node JS	v12 o superior	<a href="https://nodejs.org/en/about/releases/">https://nodejs.org/en/about/releases/</a>
C#	6.0 o superior	
PostgreSQL	13.2 o superior	<a href="https://www.postgresql.org/docs/">https://www.postgresql.org/docs/</a>
MySQL	8.0 o superior	<a href="https://dev.mysql.com/doc/">https://dev.mysql.com/doc/</a>
MariaDB	10.5 o superior	<a href="https://mariadb.com/kb/en/documenta-tion/">https://mariadb.com/kb/en/documenta-tion/</a>
MongoDB	3.6 o superior	<a href="https://docs.mongodb.com/manual/">https://docs.mongodb.com/manual/</a>
Elasticsearch	7.8 o superior	<a href="https://www.elastic.co/guide/index.htm-l">https://www.elastic.co/guide/index.htm-l</a>
Redis	6.2	<a href="https://redis.io/documentation">https://redis.io/documentation</a> <a href="https://redis.io/download">https://redis.io/download</a> <a href="https://github.com/redis/redis/releases">https://github.com/redis/redis/releases</a>
SQL Server	14.0 o superior	<a href="https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15">https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15</a>
Oracle	19c o superior	<a href="https://docs.oracle.com/en/database/oracle/oracle-database/">https://docs.oracle.com/en/database/oracle/oracle-database/</a> <a href="https://support.oracle.com/knowledge/Oracle%20Database%20Products/742060_1.html">https://support.oracle.com/knowledge/Oracle%20Database%20Products/742060_1.html</a>

Frameworks	Versiones aceptadas	Aclaraciones
JavaScript React	V 16 o superior	<a href="https://github.com/facebook/react/releases">https://github.com/facebook/react/releases</a>
JavaScript VUE		
JavaScript Angular	10.0.0	Última version estable: 1.6.9
Laravel	6.0 LTS, 8 o superior	<a href="https://laravelversions.com/">https://laravelversions.com/</a> <a href="https://laravel-news.com/laravel-releases">https://laravel-news.com/laravel-releases</a>
Django	2.2 LTS o superior	<a href="https://www.djangoproject.com/download/">https://www.djangoproject.com/download/</a>
Flask	1.1.0 o superior	<a href="https://flask.palletsprojects.com/en/1.1.x/changelog/?highlight=changelog">https://flask.palletsprojects.com/en/1.1.x/changelog/?highlight=changelog</a>

- *tip tecnología: entornos de ejecución de intérpretes*
  - Ej., si se acepta la versión de java 1.8, entonces un JSP a utilizar es tomcat 8
  - Tomcat, gunicorn, glassfish, etc. deben coincidir con versiones de framework

## Estándares de desarrollo

El código debe ser legible y entendible para que el producto salga en producción y pueda ser adecuadamente mantenido. Es tan importante una buena documentación y prolijidad en el código como las funcionalidades solicitadas. No se aprobarán proyectos con deficiencias de estilo y documentación. El código no podrá presentar errores graves de codificación, que impidan su funcionamiento o su ejecución fuera de los parámetros acordados por el Usuario y el Líder de Proyecto.

### 1. Estilo General

Documentar funciones, clases, módulos y paquetes con comentarios en el mismo código (docstrings). La documentación debe generarse en formato *markdown* (.md). Documentar la funcionalidad del desarrollo con ejemplos de cómo funciona el código, screenshots, y todo lo que se considere necesario para que alguien que no conoce el desarrollo pueda utilizarlo sin mayores inconvenientes. Documentar la arquitectura de la solución. Documentar el proceso de instalación. Documentar toda configuración adicional requerida para la ejecución del código, incluyendo librerías, módulos, o dependencias requeridas para el correcto funcionamiento del producto. Por ejemplo, permisos y usuarios sobre archivos y directorios, path de ejecución de scripts si los hubiese, datos a modificar dentro de los archivos de configuración, descripción de dockerfiles o docker compose files, etc. Incluir en la documentación el charset encoding utilizado en código de la aplicación y en el motor de la base de datos. En caso de que la aplicación cuente con Usuarios Finales, la misma deberá tener un Manual de Usuario. Usar espacios en vez de tabs para indentar. Se recomienda limitar caracteres por la línea de acuerdo a las especificaciones del estándar de lenguaje utilizado. Usar lenguaje natural para nombrar funciones, clases y variables. Los nombres deben ser entendibles y pueden usarse palabras en español o inglés teniendo en cuenta la compatibilidad ASCII. Respetar las especificaciones del lenguaje utilizado y tomar como referencia lo utilizado en la práctica para el lenguaje. En caso de concatenar palabras, se sugiere utilizar una convención de nombres a definir (camelcase, snake\_case, etc). Siempre que se pueda se deberán utilizar librerías reutilizables y públicas en tanto y en cuanto no se encuentre en estado obsoleto (deprecated), y/o sin soporte, al momento de escribir el código.

### 2. Requerimientos y Sugerencias.

Se detallan a continuación requerimientos y sugerencias para tener en cuenta en los desarrollos dentro de la Municipalidad de Vicente López. Los requerimientos deberán ser cumplidos por el desarrollador en sus productos, mientras que las sugerencias refieren a buenas prácticas a considerar.

## Requerimientos

### R1. Enrutamiento de peticiones web vía proxies reversos y/o balanceadores de carga

Toda aplicación deberá estar disponible detrás de un dispositivo intermediario que gestione sus peticiones web para garantizar el correcto flujo y monitoreo del tráfico, la aplicación de servicios de seguridad y la mejora en la disponibilidad al desacoplar el entorno de ejecución del manejo de tráfico web.

Por lo tanto, la aplicación deberá adecuar sus procesos internos y la gestión de su contenido estático para ejecutarse mediante la presencia de proxies reversos o balanceadores de carga, considerando que las direcciones de acceso a la aplicación (URL's) pueden ser modificadas, incluida la definición del protocolo en la url (http y https). Será obligatorio que se documenten los cambios necesarios en la aplicación que deban acompañar las modificaciones de URL's.

## R2. Sistema de control de versiones: fuente de información

Todo código base de la aplicación debe ser alojado en un sistema de control de versiones como Git. Todo desarrollador que necesite trabajar con la aplicación deberá descargar el código fuente a su entorno de desarrollo local desde un único repositorio común al proyecto. Cada desarrollador podrá estar trabajando en su propia versión de código, pero siempre los cambios deberán ser remitidos a un único repositorio de control de versiones. Esta será la única fuente de información válida. El código base alojado en el repositorio único del proyecto es la fuente de cambios para su construcción, su posterior prueba y su despliegue. El código debe producir una única compilación que, combinado con una configuración específica, produce una versión inmutable, a la que no se le pueden realizar cambios hasta que se vuelva a desplegar e implementar en un entorno.

## R3. Declaración explícita de dependencias

Toda dependencia (librerías y/o paquetes) debe ser declarada explícitamente tanto en el código como en la documentación asociada. Por ejemplo, uso de “*requirements.txt*” en Python o “*package.json*” en JavaScript para que un manejador de módulos/dependencias como *pip* o *npm* pueda inicializarlas para su uso en la aplicación. No deben almacenarse elementos compilados o *artifacts* junto con el repositorio del código base y deberán ser referenciados como dependencias para ser cargadas y ejecutadas por la aplicación durante su runtime.

Referencias

<https://docs.npmjs.com/cli/v7/configuring-npm/package-json>

[https://pip.pypa.io/en/stable/reference/pip\\_install/#requirements-file-format](https://pip.pypa.io/en/stable/reference/pip_install/#requirements-file-format)

<https://getcomposer.org/doc/01-basic-usage.md>

## R4. Gestión de configuración

En caso de que diferentes tipos de despliegue dependan de configuraciones específicas como cuentas de servicio, parámetros generales de ejecución, etc., se deberá explicitar y documentar dicha configuración y el proceso asociado. Las configuraciones se utilizarán en el entorno de ejecución de la aplicación, pero sus valores serán externos y no deben encontrarse declarados en el repositorio de código base de la aplicación. Por ejemplo, los parámetros de configuración de la aplicación se podrán declarar en un archivo externo o se podrán declarar como variables de entorno. Los parámetros de configuración deberán ser independientes de la lógica de la aplicación para que puedan ser modificados de acuerdo a las instancias y entornos de ejecución.

## R5. Servicios de aplicación conectables

Los servicios utilizados por la aplicación deberán ser tratados como recursos conectables o no por el entorno de ejecución. Se tiene como objetivo la portabilidad y el mantenimiento del sistema al separar la lógica de la aplicación con los servicios que utiliza. Por ejemplo, la conexión hacia una API o hacia una base de datos a través de un usuario, contraseña y URL, o la conexión mediante una URL a un sistema de mensajes de error o un sistema de almacenamiento de archivos.

## R6. Paridad: Desarrollo = Test = Producción

Cada instancia deberá ser lo más parecida una de otra como sea posible, y a su vez ser independientes entre sí. Con independencia nos referimos a la capacidad de que una nueva versión en desarrollo pueda continuar el camino de implementación y despliegue hacia producción sin depender de sus versiones previas en ejecución. Los factores de dependencia pueden ser varios, como archivos almacenados en los entornos, configuraciones anteriores, variables de entorno residuales, etc.

La paridad permite la reproducción de síntomas, de capacidad de concurrencia, etc. La facilidad en la reproducción incrementa la velocidad de despliegue y la facilidad para eliminar y desplegar nuevas

aplicaciones.

## Sugerencias

### S1. Gestión de procesos de aplicación

Una aplicación en ejecución dentro de un entorno ejecutará uno o más procesos sin estado (stateless) y no deberá compartir información entre sí. Esto permite que la aplicación pueda escalar mediante la replicación de procesos e incrementar su portabilidad a través de los entornos que integran su infraestructura. Considerando el concepto de “sticky sessions”, en un contexto donde los procesos en ejecución pueden interrumpirse o perderse en cualquier momento, no hay garantía que el resultado de su ejecución se encuentre almacenado de forma local, y que subsecuentes peticiones sean gestionadas por el mismo proceso. Por lo tanto, es recomendable persistir explícitamente todo dato que necesita ser reutilizado en un backing service como una base de datos. Todo dato que se requiera que persista, se deberá almacenar en un backing service para que el proceso sea sin estado (stateless).

### S2. Autocontención y concurrencia de procesos

La aplicación debería ser su propio proceso y buscar su aislamiento completo del resto de aplicaciones externas. En un entorno de aplicación tradicional, cada proceso gestiona diferentes funcionalidades de la aplicación y esta es disponibilidad a través de un protocolo web como HTTP mediante un servicio como Apache o Nginx. En un entorno de aplicación moderno, la aplicación (o inclusive cada proceso) se vuelve disponible a otros procesos o servicios mediante la integración de una librería que gestiona peticiones web, con lo cual su ejecución se encuentra autocontenida y es independiente de la ejecución de otros procesos de aplicación y de servicios web externos. La aplicación conoce como realizar manejar peticiones HTTP o presentarse a través de una dirección IP o puerto (port binding). Es posible declarar explícitamente dependencias que integren librerías web en Python (Tornado, flask + gunicorn), node.js (Express), Java (jetty, Spring Boot, etc.), Ruby (Puma, Rack, Thin). La capacidad de cada aplicación para ser autocontenida y gestionar sin estado las peticiones de usuario le permitirá ser auto escalable de forma horizontal para soportar el incremento de concurrencia sin degradación de performance o aparición de errores.

### S3. Registros de aplicación (Logs)

Tratar a los registros de aplicación como una secuencia de eventos (event streams) y no como un archivo (log file). La aplicación debería producir registros como eventos y su entorno de ejecución tener la posibilidad de extraerlos y agregarlos a través de servicios externos de agregación de logs como fluentd, graylog o logstash, entre otros. El almacenamiento y gestión de logs no debe incluirse en la lógica de la aplicación.

## 3. Estándares de Estilo por lenguaje

Se listan a continuación los estándares de estilo obligatorios para cada tipo de lenguaje utilizado.

- Python: <https://www.python.org/dev/peps/pep-0008/>
- JavaScript: <https://google.github.io/styleguide/jsguide.html>
- NodeJS: <https://cloud.google.com/nodejs/docs/setup>
- PHP: <https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/>
- Java: <https://google.github.io/styleguide/javaguide.html>
- C++ <https://google.github.io/styleguide/cppguide.html>
- C# <https://google.github.io/styleguide/csharp-style.html>

## Ambientes

Toda aplicación debe contar con 3 ambientes (Dev, Test/QA y PRD).

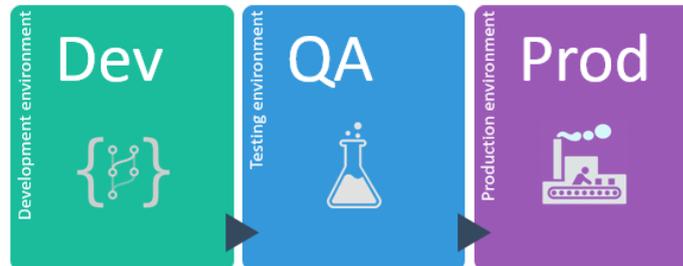


Fig.1. Ambientes de Desarrollo de software

### Gestión de ambientes

El proceso de gestión de ambientes refleja la importancia de la sustentabilidad de las aplicaciones, incluyendo actividades específicas para controlar y asegurar: la calidad, desempeño y seguridad de las aplicaciones. Este documento busca establecer y describir las tareas a desarrollar para efectuar la creación y el pasaje de ambientes hasta la puesta en producción de los proyectos de manera segura, a fin de minimizar el riesgo de alteración de los sistemas productivos. La generación de distintos ambientes tiene como objetivo principal garantizar que el código que se implemente en producción se encuentre en óptimas condiciones de funcionamiento, resultado de las pruebas. Se busca preservar el estado del código y del ambiente ya probado.

### El ambiente DEV/Desarrollo

Todo software, ya sea un desarrollo nuevo o un cambio a uno existente debe contar con un ambiente de desarrollo provisto por la Subsecretaría de Modernización y autogestionado, permitiendo al equipo de Desarrollo implementar la solución con las verificaciones correspondientes. El ambiente DEV posee la misma infraestructura que el ambiente de QA/Test y PRD/Producción, con los recursos acorde a un ambiente de Test. Existirá un entorno de Desarrollo por cada nueva aplicación que se incorpore. Independientemente de la plataforma utilizada por el Desarrollador para generar su producto, todas las aplicaciones deben desempeñarse correctamente en el entorno DEV provisto a tal efecto.

### El ambiente de QA/Test

Cuando en Desarrollo se encuentre una versión que haya superado los controles exitosamente, se podrá solicitar el pasaje al ambiente de Test administrado también por la Subsecretaría de Modernización. Aquí se verifican automáticamente los mismos controles que en DEV y se suman los diferentes tipos de pruebas. En esta instancia es importante contar con la participación del Usuario Final. Sin la aprobación del usuario final, el desarrollo no estará en condiciones de implementarse en producción. El equipo de Desarrollo tendrá acceso de sólo lectura para verificar el comportamiento de la aplicación desplegada, no así acceso de escritura. Esto último, con el fin de preservar la paridad de código entre ambientes y garantizar el desempeño correcto de las pruebas en test para su posterior pasaje a producción.

### **El ambiente PRD/Producción**

En este ambiente es donde ocurren todas las transacciones de los usuarios. Las aplicaciones productivas contarán con un monitoreo de Infraestructura y chequeos adicionales del comportamiento on-line, contando con controles proactivos y alarmas del funcionamiento.

## **Proceso de Desarrollo / Ciclo de Vida de Proyecto**

En la siguiente sección detallamos el proceso definido por la Subsecretaría de Modernización y la Dirección General de Procesos y Desarrollo Informáticos para la implementación de nuevas aplicaciones o versiones evolutivas y correctivas de aplicaciones existentes (**Anexo V**).

### **Alta de Desarrollador y de Ambientes**

Para el desarrollo de un sistema se requerirá la inscripción del desarrollador en un formulario online (ver **Anexo II Formulario Alta de Desarrollador y Ambientes**) el cual contendrá información del proyecto, datos de contacto del desarrollador, así como también datos técnicos de los ambientes requeridos en el proyecto. Para ello, el líder de proyecto pondrá a disposición el formulario que el desarrollador deberá completar para iniciar el proceso. El formulario lo recibirá el equipo de Data Center para crear los ambientes, y el equipo de Seguridad Informática para crear los usuarios y accesos.

### **Creación del repositorio para un proyecto**

El código fuente de las aplicaciones es uno de los “entregables” de cada proyecto y debe ser impactado en el repositorio definido por la Subsecretaría de Modernización para tal fin. El sistema de control de versiones elegido es Git#. El código fuente almacenado en el sistema de control de versiones será considerado la única fuente de gestión de código para implementar cambios y actualizaciones sobre los ambientes.

El mismo será la fuente para el despliegue de la aplicación en las etapas siguientes del proceso.

Todo el código fuente necesario para un proyecto, así como su documentación, scripts auxiliares, parámetros de configuración de entorno, esquema de bases de datos, tests de seguridad, arquitectura, procesos de migraciones y todo el material auxiliar para el proyecto (ej. documentación funcional), deberá estar presente en el repositorio del mismo, incluyendo el versionado de cambios.

Git MVL: <https://git.vicentelopez.gov.ar>

En base a los datos suministrados en el *Formulario Alta de Desarrollador y Ambientes* se creará el repositorio del proyecto en el sistema de control de versiones de la Municipalidad y los usuarios requeridos para acceder al mismo.

El repositorio del proyecto se creará bajo la estructura:

/[nombre secretaria solicitante]/[Secretaria/DG/D]/[nombre de proyecto].

## Gestión de código fuente y despliegue aplicaciones

### Supuestos y alcances

- El presente flujo de trabajo no considera la gestión de cada equipo de desarrollo sobre el repositorio de código (manejo de conflictos, autorizaciones, etc.).
- Los comandos indicados en el flujo de trabajo se presentan a modo indicativo, para resaltar la operación general de las áreas de desarrollo y operaciones.
- Se considera la aplicación de una única rama git (*git branch*) para una mayor comprensión general de los pasos que intervienen en el despliegue de aplicaciones en producción. El proceso puede incluir el uso de múltiples ramas (*branches*) que reflejen el estado del código en cada rama.

### 1 - Creación de proyecto por desarrollador y carga de código en repositorio de proyecto.

```
git init
git add *
git commit -m "mi primer commit! inicio de desarrollo"
git branch -M main
git remote add origin https://git.vicentelopez.gov.ar/Modernizacion/proyecto.git
git push -u origin main
```

### 2 - Desarrollo y actualización del código fuente versionado

```
git add *
git commit -m "estoy desarrollado, termine cambio sobre archivos"
git branch -M main
git push -u origin main
```

### 3 - Notificación a Líder de Proyecto para prueba de código en testing

### 4 - Actualización código en testing

```
cd proyecto/
git branch -M main
git pull -u origin main
```

### 5a - Estado de desarrollo en testing OK. Aprobación de Usuario Final.

### 5b - Estado de desarrollo en testing FAIL: se requieren correcciones sobre la aplicación.

### 6a - Actualización de código en Producción

```
cd proyecto/
git branch -M main
git pull -u origin main
```

### 6b - Desarrollo y actualización del código fuente versionado

```
git add *
git commit -m "estoy desarrollado, termine ajuste por solicitud de cambio XXXX por Secretaría"
git branch -M main
git push -u origin main
```

### 7 - Notificación a Líder de Proyecto para prueba de código en testing

### 8 - Actualización código en testing

```
cd proyecto/
git branch -M main
git pull -u origin main
```

9 - Estado de desarrollo en testing OK. Aprobación de usuario final.

10 - Actualización de código en producción

```
cd proyecto/  
git branch -M main  
git pull -u origin main
```

### **Etapas de Desarrollo**

El equipo de desarrollo realiza el desarrollo y pruebas internas del sistema/aplicación, basadas en base al documento de alcance. Terminando el desarrollo el proveedor genera un push en GIT, que deberá ser autorizado por el Líder de Proyecto para que desde Infraestructura/Datacenter procedan con el despliegue del mismo en QA/Test.

### **Instalación en ambiente Test**

El Líder de Proyecto aprueba la instalación en Test, y su instalación en el ambiente por parte del equipo de Datacenter. En el caso de requerir la publicación de servicios de la aplicación se deberá solicitar accesos al equipo de Seguridad Informática.

### **Verificación Técnica de la instalación**

El proveedor debe verificar técnicamente el desempeño de su código desplegado en el ambiente de test y confirmar al Líder del Proyecto el estado de su funcionamiento. Deberá proveer un informe de dichos tests con sus resultados.

Los tests de la aplicación sobre el ambiente de test son responsabilidad de los desarrolladores, siendo la Municipalidad quien valide luego los mismos.

### **Tests funcionales**

El Líder de Proyecto solicitará a los Usuarios (designados por la Secretaría solicitante para este fin) comenzar con las pruebas de la aplicación. El resultado de las mismas será reflejado en el Documento de Conformidad de Usuario, el cual será requisito para iniciar el pasaje a producción.

### **Chequeo de Seguridad**

Líder de Proyecto solicitará al equipo de Seguridad Informática que ejecute el chequeo de Vulnerabilidades sobre la aplicación (de acuerdo a los Estándares de Seguridad). Si el mismo es aprobado, y las pruebas en Test también (Documento de Conformidad de Usuario, más el Informe Técnico de Proveedor), se podrá solicitar el pase a Producción.

Caso contrario se deben detallar las vulnerabilidades asociadas al despliegue para que el desarrollador realice la corrección necesaria.

## Pasaje a Producción

Una vez completadas y validadas las pruebas de usuario el Líder de Proyecto podrá solicitar el pasaje a producción. Se deberá indicar al equipo de Infraestructura las condiciones y su proceso de despliegue mediante el formulario online de Solicitud de Pasaje a Producción (ver **Anexo III**).

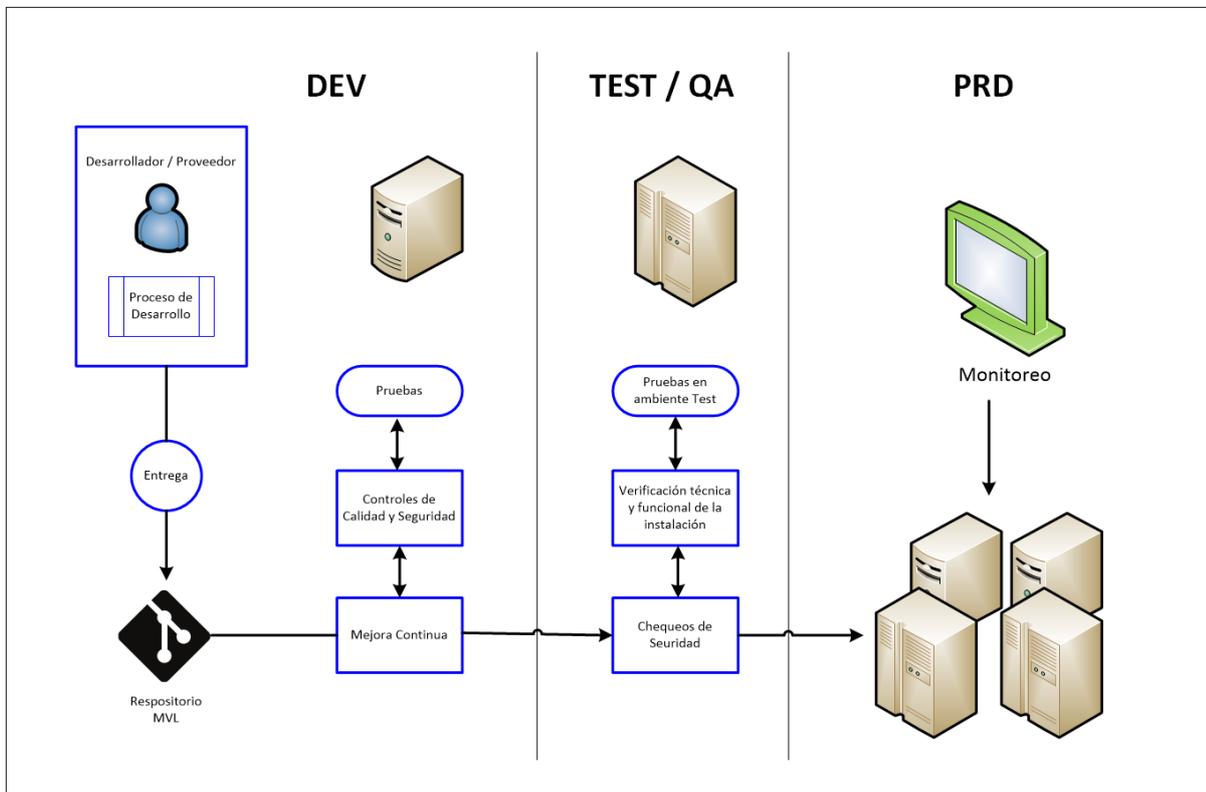


Fig. 1. Ambientes en el proceso de Desarrollo

## Mantenimiento

Alcance: Ajustes necesarios para que el sistema o aplicación, una vez autorizado su pase a entorno de producción, mantenga su funcionalidad tal como fue definida en los casos de uso de su especificación funcional. Transferencia de conocimientos y respuesta a consultas del personal técnico interno de soporte, con relación a adaptaciones menores por cambios de entorno menores (p.ej. reinstalación en un nuevo server, cambios de hostname, IP o DNS, ABMC de roles de usuarios). El mantenimiento de las aplicaciones debe estar contemplado como parte de la propuesta de cada sistema a excepción de que se acuerde lo contrario con la Subsecretaría de Modernización. El mantenimiento inicial lo debe realizar el propio proveedor y deberá luego realizar una transferencia de conocimientos al personal técnico designado por el Líder de Proyecto. Esto con el fin de incorporar los conocimientos técnicos suficientes para poder gestionar el código y su proceso de implementación. Plazos y periodo de mantenimiento. A definir como parte de la propuesta, considerando un mínimo de 1 año de garantía de la versión desarrollada y en producción. Existe la alternativa de considerar el mantenimiento evolutivo tercerizado, habitualmente con el mismo desarrollador inicial del producto, a partir de la expiración de la garantía. Este mantenimiento evolutivo debe considerarse en el costo operativo dentro del presupuesto anual.

## Capacitación

De estar contenido en el pliego, la capacitación para los sistemas desarrollados será responsabilidad del proveedor. El Líder de Proyecto y la Secretaría solicitante definirán los destinatarios de dicha capacitación. Luego de la misma, el desarrollador deberá entregar un Manual de Usuario.

## Acuerdo de Confidencialidad

Este acuerdo reunirá todo lo establecido por el estándar de Desarrollo y será condición obligatoria la firma de ambas partes, de esta forma daremos la seguridad que corresponde y merece dicho Desarrollo. Se adjunta en el ANEXO I.

## Glosario

- Desarrollador (Área de Desarrollo interna o personal contratado): tiene la responsabilidad de la construcción del aplicativo. Este rol puede ser cubierto por personal de desarrollo de la MVL, o por personal contratado.
- Aplicaciones: llamaremos aplicación a cualquier desarrollo software funcionalmente independiente que, no obstante, puede interconectarse, puntualmente, a otros desarrollos.
- Servicios Digitales: servicios que se descargan, visualizan o utilizan a través de la red internet o similar, que están básicamente automatizados y con mínima intervención humana.
- Github: es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git
- Repositorio Versionado: Se trata de una categoría de herramientas que sirven para que los equipos de desarrollo administren los cambios al código fuente a lo largo del tiempo a través del registro de todas las modificaciones realizadas al código en una base de datos diseñada para tales propósitos. Esto permite comparar y modificar diferentes versiones del código a través del tiempo.
- SaaS: acrónimo para “Software como servicio”, se trata de un modelo de distribución de software en el que el proveedor produce y alberga la aplicación y la deja disponible a los usuarios a través de la Internet.
- Usuario: persona designada por la Secretaría solicitante del proyecto para la realización de pruebas y validación del estado de la aplicación.

## Contacto

Esta norma está disponible y en continua actualización. Todas las consultas podrán ser remitidas al siguiente email: [estandares.desarrollo@vicentelopez.gov.ar](mailto:estandares.desarrollo@vicentelopez.gov.ar).

## ANEXO I

### **ACUERDO DE CONFIDENCIALIDAD**

La finalidad del presente ACUERDO es garantizar y proteger las libertades públicas, los derechos fundamentales de las personas físicas involucradas, y principalmente el cuidado del secreto o intimidad, resguardo de la confidencialidad en el manejo de la información originada en el cumplimiento de sus funciones como EMPLEADO / PROVEEDOR DE SERVICIOS de \_\_\_\_\_ (área), la Municipalidad de Vicente López. El documento tomará validez una vez aceptado y firmado el nombrado contrato por ambas partes.

### **DECLARAN LAS PARTES:**

1- Que en virtud de los servicios prestados por el EMPLEADO / PROVEEDOR DE SERVICIOS se entiende por "INFORMACIÓN CONFIDENCIAL", de manera colectiva, toda la información que sea suministrada por la Municipalidad de Vicente López al EMPLEADO / PROVEEDOR DE SERVICIOS, en el desarrollo de la relación laboral/comercial, al igual que todos los datos de índole estadísticos, técnicos, know-how, conceptos, técnicas, principios y demás conceptos subyacentes a sus sistemas y aplicaciones informáticas a los que pudieran tener acceso como consecuencia del desarrollo de sus funciones, así como otros datos e información, análisis, estudios, notas, recopilaciones, resúmenes, memorandos, informes y conclusiones o cualquier otro documento preparado por \_\_\_\_\_ (área), de la Municipalidad de Vicente López u otros organismos públicos y/o privados. Asimismo se entiende como información confidencial:

- a) La información que no sea pública y sea conocida por el EMPLEADO/ PROVEEDOR DE SERVICIOS con ocasión del desarrollo de sus funciones, bien sea que la misma sea escrita, oral, visual o en cualquier clase de soporte.
- b) Cualquier tipo de información relacionada con el funcionamiento de \_\_\_\_\_ (área), su infraestructura, hardware, software, datos, usuarios, comunicaciones, procesos y métodos de seguridad informática;
- c) Cualquier tipo de información técnica, jurídica, económica, administrativa, nuevas tecnologías y cualquier otra relacionada con las operaciones técnicas presentes y futuros de \_\_\_\_\_ (área), y/o relacionada con su estructura organizacional, bien sea que la misma sea escrita, oral, visual, en cualquier clase de soporte, tangible o no, incluidos los mensajes de datos, de la cual el EMPLEADO / PROVEEDOR DE SERVICIOS tenga conocimiento o a la que tenga acceso por cualquier medio o circunstancia en virtud del desarrollo de sus funciones.

2- Que se reconocen mutuamente la personalidad con la que comparecen a celebrar el presente acuerdo y manifiestan su libre voluntad para obligarse en los términos de las siguientes cláusulas:

**PRIMERA.** El EMPLEADO/ PROVEEDOR DE SERVICIOS reconoce de manera expresa, la naturaleza confidencial de toda la información que le suministra y le haya suministrado por \_\_\_\_\_ (área), de la Municipalidad de Vicente López y es propiedad exclusiva de ésta última. En consecuencia, el EMPLEADO / PROVEEDOR DE SERVICIOS acepta y está de acuerdo en que:

- a- Solo utilizará la información confidencial con el propósito exclusivo de desarrollar las funciones que le han sido asignadas por \_\_\_\_\_ (área).
- b- No permitirá el acceso a la información confidencial ni divulgará de manera parcial o total su contenido a ningún tercero, sin el consentimiento escrito previo por parte de \_\_\_\_\_ (área), salvo en aquellos casos en que sea necesario divulgar ese contenido con el propósito conforme lo estipulado en el numeral anterior, en cuyo caso se incorporará al presente acuerdo de confidencialidad a todo personal o tercero que sea receptor de dicho contenido.

**SEGUNDA.** Se excluye de los efectos de este acuerdo aquella información sujeta a confidencialidad y reserva que:

- a- Al momento de ser revelada sea ya de dominio público, o que luego de dicha revelación, se convierta en dominio público por causas no imputables al EMPLEADO / PROVEEDOR DE SERVICIOS, para lo cual, se entenderá que es de dominio público la información que aparezca en una fuente de divulgación de libre acceso al público general.
- b- Expresamente se autorice a divulgar por \_\_\_\_\_ (área).
- c- Sea requerida por organismos legales competentes tales como tribunales provinciales o nacionales, u organismos autárquicos de control tales como el Honorable Tribunal de Cuentas.

**TERCERA.** El EMPLEADO / PROVEEDOR DE SERVICIOS, receptor de la información confidencial se obliga a mantener estricta confidencialidad sobre la documentación e información que conozca, reciba o intercambie en ocasiones de la relación laboral. En particular, se obliga a:

- a- Proteger la confidencialidad de la información verbal, escrita, visual o por cualquier otro medio que reciba, obligándose en forma irrevocable ante \_\_\_\_\_ (área), a no revelar, divulgar, facilitar, bajo cualquier forma a ninguna persona física o jurídica, pública o privada, toda información relacionada con el ejercicio de sus funciones, como así también las políticas y/o cualquier otra información vinculada con sus funciones.
- b- Usar toda información confidencial que se le entregue o que se produzca solamente a los efectos del desarrollo de sus funciones. Igualmente, se obliga a mantener dicha información en reserva hasta tanto adquiera carácter público.
- c- Implementar mecanismos tendientes a proteger la información resguardando un lugar de acceso restringido y con el mismo nivel de seguridad que utiliza para su propia información confidencial. Estos mecanismos podrán ser eventualmente auditados por la Dirección de Seguridad Informática de la Municipalidad de Vicente López.
- d- Cumplir con la obligación de confidencialidad de la misma forma y con el mismo cuidado con que se realiza la protección de la información confidencial propia.

**CUARTA.** El EMPLEADO / PROVEEDOR DE SERVICIOS está de acuerdo en que la información confidencial que reciba por parte o en virtud del cumplimiento de sus funciones es y seguirá siendo propiedad de la Municipalidad de Vicente López el poder usar dicha información únicamente de la manera y para los propósitos autorizados en el presente acuerdo. Se agrega además que este instrumento no otorga, de manera expresa o implícita, derecho intelectual o de propiedad alguno, incluyendo, mas no limitando, licencias de uso respecto de la información confidencial.

**QUINTA.** La parte receptora SE OBLIGA A UTILIZAR la información confidencial que llegó a su conocimiento únicamente para transmitir a las autoridades competentes de \_\_\_\_\_ (área), de la Municipalidad de Vicente López.

**SEXTA.** La vigencia del presente será por todo el plazo de la relación contractual y por un plazo adicional de 5 (cinco) años contados a partir de la extinción de la relación contractual y/o laboral entre la Municipalidad de Vicente López y el EMPLEADO / PROVEEDOR DE SERVICIOS.

**SÉPTIMO.** En caso de terminar con la prestación de servicio por cualquier causa, el EMPLEADO / PROVEEDOR DE SERVICIOS se obliga a reintegrar o destruir, según le solicite \_\_\_\_\_ (área) por escrito toda la documentación e información que le haya sido suministrada y todo otro material o

documento que incorpore o refleje información confidencial, a la que haya tenido acceso y que conserve el EMPLEADO / PROVEEDOR DE SERVICIOS. La entrega de la documentación se deberá efectuar en un plazo no mayor a 15 (quince) días calendario siguientes a la solicitud de entrega. Igualmente, cualquier información confidencial mantenida en computadoras personales, medios magnéticos o cualquier otro similar, será resguardada o destruida si así es requerido por \_\_\_\_\_ (área). Una vez devuelta y/o destruida la mencionada documentación, el EMPLEADO / PROVEEDOR DE SERVICIOS hará constar mediante comunicación escrita que optará el carácter de declaración jurada con todos sus efectos, que no se encuentra en su poder ningún otro documento sujeto a la obligación de confidencialidad aquí establecida.

**OCTAVO.** Este acuerdo constituye el entendimiento total entre las partes y sustituye a cualquier otro entendido previo, oral o escrito, que haya existido entre partes.

**NOVENO.** Ninguna de las partes podrá ceder sus derechos y obligaciones derivados del presente acuerdo.

**DÉCIMO.** El no cumplimiento del presente acuerdo habilitará a las autoridades competentes a tomar las medidas del caso que sean necesarias (sanciones correspondientes), quedando a tal efecto facultada la Municipalidad de Vicente López para accionar por los daños y/o perjuicios ocasionados, así como para constituirse en parte querellante en el proceso penal iniciado contra el EMPLEADO / PROVEEDOR DE SERVICIOS receptor ante la posible comisión de un delito de acción pública o privada.

**DÉCIMO PRIMERO.** Este acuerdo adhiere a lo establecido en la “LEY DE CONFIDENCIALIDAD”, para proteger toda la información que se encuentre legítimamente en poder de una persona física o jurídica que sea divulgada a terceros o sea utilizada por estos sin el consentimiento y de manera contraria a los usos comerciales honestos LEY Nro. 24.766, sección III “Acciones por infracción a la Ley” en sus Art. 11, 12, y 14.

**DÉCIMO SEGUNDO.** Este acuerdo adhiere a lo establecido en la “LEY DE PROTECCIÓN DE DATOS PERSONALES”, son los principios generales relativos a la protección de datos, derechos de los titulares de datos. usuarios y responsables de archivos, registros y bancos de datos LEY 25.326, Art. 1 al Art. 48 en todos su capítulos y Decreto Reglamentario N° 1558/01 y modificaciones. Con especial observancia de los derechos fundamentales a la privacidad de los ciudadanos de Vicente López, cuya protección es objetivo fundamental del equipo de Tecnología de la Municipalidad de Vicente Lopez.

**DÉCIMO TERCERO.** Queda establecido expresamente que la remuneración que el EMPLEADO / PROVEEDOR DE SERVICIOS percibe compensa a las obligaciones de confidencialidad, aclaradas en este documento.

**DÉCIMO CUARTO. CLAÚSULA PENAL:** Se deja constancia que la violación o el incumplimiento de la obligación de confidencialidad a cargo del EMPLEADO / PROVEEDOR DE SERVICIOS, así como falsedad de la información que pudiere brindar a terceros, podrá dejarlo incluso en el delito de violación de secreto tipificado en el Art. 156 del Código Penal de La Nación, siendo facultad de \_\_\_\_\_ (área), de la Municipalidad de Vicente López, formular la denuncia del caso y constituirse en parte querellante.

**DÉCIMO QUINTO. LEY APLICABLE Y JURISDICCIÓN:** El presente Acuerdo tiene naturaleza privada y está sometido al derecho argentino. Las Partes acuerdan someter toda controversia, diferencia o reclamación que surja del presente acuerdo y de toda enmienda al mismo o relativa al presente, incluyendo en particular, su formación, validez, obligatoriedad, interpretación, ejecución, incumplimiento o resolución, así como las reclamaciones extracontractuales, será sometida a un procedimiento judicial ante los Tribunales de la ciudad de San Isidro – Provincia de Buenos Aires que resulten competentes.

Enteradas las partes del contenido y alcance del presente contrato, lo firma en fecha \_\_\_\_ / \_\_\_\_ / \_\_\_\_

**Firma:**

**Área correspondiente:**

**Firma:**

**Empleado / Proveedor de servicios:**

**ANEXO II**

**Formulario Alta de Desarrollador y Ambientes.**

ALTA DE DESARROLLADOR Y AMBIENTES	
Listado de Usuarios	
Usuario 1	<ul style="list-style-type: none"> <li>• DNI:</li> <li>• Apellido y Nombre:</li> <li>• N° de Proveedor:</li> <li>• Razón Social:</li> <li>• Tel:</li> <li>• E-mail:</li> </ul>
Usuario 2	<ul style="list-style-type: none"> <li>• DNI:</li> <li>• Apellido y Nombre:</li> <li>• N° de Proveedor:</li> <li>• Razón Social:</li> <li>• Tel:</li> <li>• E-mail:</li> </ul>
Usuario 3	<ul style="list-style-type: none"> <li>• DNI:</li> <li>• Apellido y Nombre:</li> <li>• N° de Proveedor:</li> <li>• Razón Social:</li> <li>• Tel:</li> <li>• E-mail:</li> </ul>
Usuario N	<ul style="list-style-type: none"> <li>• DNI:</li> <li>• Apellido y Nombre:</li> <li>• N° de Proveedor:</li> <li>• Razón Social:</li> <li>• Tel:</li> <li>• E-mail:</li> </ul>
Nombre del Proyecto	

Secretaría solicitante	
Instancias Requeridas	<ul style="list-style-type: none"> <li>• Test / Producción [ ]</li> <li>• Desarrollo / Test / Producción [ ]</li> </ul>
Modelo de Sistema Operativo	<ul style="list-style-type: none"> <li>• Linux Debian 9 o superior [ ]</li> <li>• Linux CentOS 7 [ ]</li> <li>• Windows Server 2012 R2 o superior [ ]</li> <li>• Otro [ ]</li> </ul>
Sistema de Gestión de Base de Datos	<ul style="list-style-type: none"> <li>• SQL Server 2016 o superior [ ]</li> <li>• MySQL 5.5 en adelante [ ]</li> <li>• Postgre SQL 9.6 en adelante [ ]</li> <li>• MongoDB [ ]</li> <li>• Elasticsearch [ ]</li> <li>• Otro [ ]</li> </ul>
Recursos de Hardware Desarrollo	<ul style="list-style-type: none"> <li>• Cantidad de VCPUs (detalle resumido de su función):</li> <li>• RAM:</li> <li>• Tamaño de particiones de disco:</li> </ul>
Recursos de Hardware Test	<ul style="list-style-type: none"> <li>• Cantidad de VCPUs:</li> <li>• RAM:</li> <li>• Tamaño de particiones de disco:</li> </ul>
Recursos de Hardware Producción	<ul style="list-style-type: none"> <li>• Cantidad de VCPUs:</li> <li>• RAM:</li> <li>• Tamaño de particiones de disco:</li> </ul>
Lenguaje de programación	<ul style="list-style-type: none"> <li>• PHP [ ]</li> <li>• Python [ ]</li> <li>• Ruby [ ]</li> <li>• Java [ ]</li> <li>• .NET Framework [ ]</li> <li>• Javascript [ ]</li> </ul>
Framework	
Paquetes y software requerido	<ul style="list-style-type: none"> <li>• Nombre de paquete:</li> <li>• Versión: Ejemplo 1: openssl 1.0.2k-12.el7.x86_64 Ejemplo 2: docker-ce 20.10.3</li> </ul>

<p>Repositorios requeridos</p>	<ul style="list-style-type: none"> <li>• [nombre_repo]</li> <li>• [URL]</li> </ul> <p>Ejemplo: [CentOS-Base.repo] [[http://mirrorlist.centos.org] [deb http://ftp.us.debian.org/debian/ stretch main contrib]</p>
<p>Puertos de red a habilitar</p>	
<p>Crecimiento estimado de recursos de la aplicación</p>	<ul style="list-style-type: none"> <li>• Disco</li> <li>• CPU</li> <li>• RAM</li> </ul>

**ANEXO III**

**Solicitud de Pasaje a Producción**

<b>SOLICITUD PASAJE A PRODUCCION</b>		
Aprobación y conformidad de	<i>[Nombre completo 1] – [Líder de proyecto]</i>	
Fecha Propuesta	<i>[DD-MM-AAAA]</i>	
Hora Propuesta	<i>[HH:MM a HH:MM] horas</i>	
Contactos Técnicos	<i>[Nombre completo 1] – [Desarrollador] [Nombre completo 2] – [Desarrollador]</i>	
Usuarios Internos	<i>[Nombre completo 1] – [Secretaría MVL] [Nombre completo 1] – [Secretaría MVL]</i>	
Fecha Emisión del Formulario	<i>[DD-MM-AAAA HH:MM a HH:MM] horas</i>	
Objetivo	<i>Puesta en producción Proyecto N. Documento de instalación: Procedimiento de instalación y software necesario</i>	
<b>INDICACIONES DE INSTALACION</b>		
<b>Tema</b>	<b>SI</b>	<b>NO</b>
<b>Publicación a Internet</b>		
Nombre de sitio web propuesto:		
<b>CONSIDERACIONES DE DESPLIEGUE EN PRODUCCIÓN</b>		
<b>PROCEDIMIENTO DE INSTALACIÓN</b>		

*Tarea 1: instalar el paquete "A"*

*Ejemplo:*

- `Yum install paquete-A-1.0el7.x86_64`

*Tarea 2: crear carpeta "datos"*

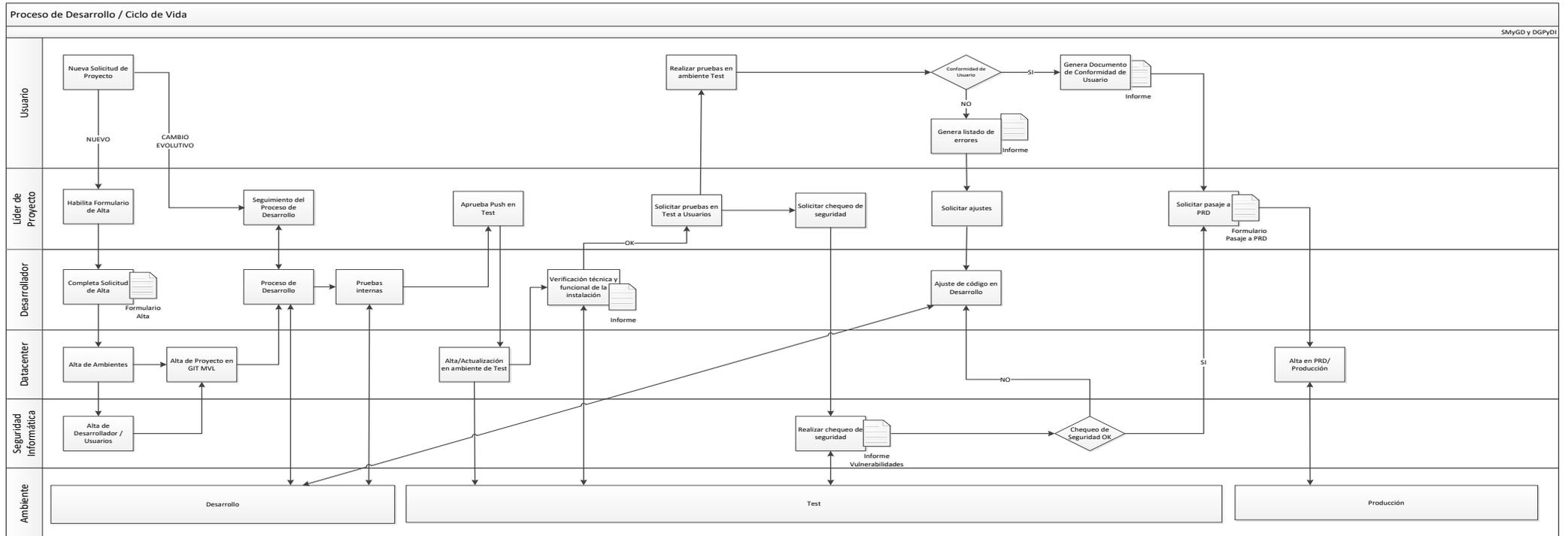
*Ejemplo:*

- `mkdir /var/www/public_html/datos`

*Tarea 3: Asignar permisos*

*Tarea 4: Configurar archivo con los siguientes parámetros:*

## Diagrama de Flujo - Proceso de Desarrollo MVL



ANEXO IV

<http://www.bcra.gov.ar/pdfs/sistemasfinancierosydepagos/SNP3525.pdf>

QR:



Contenido: 00020101021141390016com.adquirente0115info\_adquirente5013000912345678952 049700  
53030325802AR5909FULL NAME6010CITY LEGAL63045BE9

Desglose:

00 02 01

01 02 11 41 39

00 16 com.adquirente

01 15 info\_adquirente

50 13

00 09 123456789

52 04 9700

53 03 032

58 02 AR

59 09 FULL NAME

60 10 CITY LEGAL

63 04 6725

En este caso, el QR cuenta con un único adquirente que es “com. adquirente” y se encuentra en la posición 41.

Elementos en root node:

status: string. Posibles valores: open\_amount, closed\_amount, pending, unsupported\_qr\_code, timeout, unsupported merchant, empty order, error. collector.name: string (50)

collector. identification number: string (11) collector. Account: string (22) order.id: string (68) order. total amount: decimal (,2) order.max\_amount: decimal (,2) order.min\_amount: decimal (,2)

items: item [] retry delay: integer

Objeto item:

item. Description: string (50)

item. unit price: decimal (,2)

item. currency: string (3) representando ISO currency code de 3 letras

item. quantity: integer

item. picture URL: string (2048) con formato URL.

item. title: string (50)