



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	1 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

ELABORA	REVISAR	ADMINISTRA
RÚBRICA Karina Onofre Martínez Directora General de Gobierno Digital y Soporte Tecnológico	RÚBRICA Cintia Smith Secretaria de Innovación y Gobierno Abierto	RÚBRICA María Fernanda Araujo Meza Directora de Planeación, Enlace y Proyectos Estratégicos

Con fundamento en lo dispuesto en los artículos 1, 88, 89 y 96 de la Ley de Gobierno Municipal del Estado de Nuevo León; artículos 5, 11, 14, 16 fracción IX, 113, 114, 115, 116 fracciones I, II, III, V, VII, XVIII, XX, XXI, XXIV y XXX, 117 fracciones I, II, III, IV, VII, XV, XVII, XVIII, XXIV y XXVIII del Reglamento de la Administración Pública Municipal de Monterrey; artículo 1, 6 fracciones I, III, V, VIII, XI, XIV, 7, 8 fracción I, III, V, VI, X, XI, , 23, 24 fracción X, 52 fracciones I, II, III, IV y V, 53, 54, 55 y 56 del Reglamento de Gobernanza Tecnológica para el Municipio de Monterrey, así como demás ordenamientos y disposiciones jurídicas aplicables.

La Secretaría de Innovación y Gobierno Abierto, a través de la Dirección de Gobierno Digital de la Dirección General de Gobierno Digital y Soporte Tecnológico, genera el presente lineamiento con la finalidad de mejorar la experiencia de las personas usuarias al interactuar con los productos digitales y sitios web del Municipio de Monterrey, buscando armonizar la tecnología con la que cuenta la Administración Pública del Municipio de Monterrey.

I. OBJETIVO

Garantizar la calidad del desarrollo de software de manera integral mediante la implementación de estándares y prácticas, asegurando la mantenibilidad, legibilidad, eficiencia y seguridad del código; consolidando productos confiables para la Administración Pública Municipal y Paramunicipal de Monterrey.

II. ALCANCE

El presente lineamiento es aplicable a la Dirección General de Gobierno Digital y Soporte Tecnológico de la Secretaría de Innovación y Gobierno Abierto al requerir, dictaminar o implementar soluciones digitales en el ámbito de sus respectivas atribuciones, así como a las Dependencias y Entidades de la Administración Pública Municipal y Paramunicipal de Monterrey que pretendan llevar a cabo un desarrollo de Software.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	2 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

III. DEFINICIONES

API. Por sus siglas en inglés, Application Programming Interface. Es una interfaz de programación de aplicaciones informáticas, estandarizadas que posibilitan el intercambio de datos.

API-First. Enfoque de desarrollo de software que prioriza el diseño y desarrollo de la interfaz de programación de aplicaciones (API) antes de construir la interfaz de usuario (UI) o cualquier otra capa de la aplicación. En lugar de comenzar el desarrollo con la UI o la lógica de negocio, se comienza creando una API bien definida y robusta que expone la funcionalidad del sistema de forma clara y coherente.

Atributos en POO. Característica o propiedad que se asigna a un objeto, variable o elemento de código. Estos atributos se utilizan para describir y definir las características de un elemento en particular y pueden tomar diferentes formas dependiendo del lenguaje de programación que estés utilizando.

Autenticación. Verificación de credenciales de acceso asociadas a una persona física o moral, las cuales son validadas contra las mismas credenciales registradas previamente al momento de la creación de la cuenta de usuario del mismo.

Autoridad de Certificación (CA). Entidad que ha sido autorizada por los navegadores para emitir certificados digitales como TLS y SSL. Estos certificados se utilizan para cifrar y garantizar la entrega confidencial entre los extremos involucrados. Como las transacciones en internet y la transferencia segura de datos. Estas organizaciones se someten a auditorías anuales por parte de entidades independientes para garantizar que cumplen las políticas y procedimientos definidos para validar, emitir y revocar certificados.

Autorización. Al acto de conceder a una parte autenticada permiso para realizar una acción específica.

BackEnd. También conocido como la parte lógica o servidor de un sitio web, despliega un papel fundamental en la gestión de la lógica empresarial. Su responsabilidad recae en la manipulación de datos, tanto en la recepción como en la entrega de información procesada a las aplicaciones y sitios web asociados. Este proceso se ejecuta con el propósito de optimizar la experiencia de navegación, asegurando al mismo tiempo el funcionamiento fluido y la integridad de las diversas funcionalidades.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	3 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

Bcrypt. Es un algoritmo de cifrado de contraseñas ampliamente utilizado y considerado uno de los más seguros disponibles en la actualidad.

Bibliotecas (Desarrollo de Software). Conjunto de código que contiene funciones, clases y métodos listos para ser utilizados en un programa. El objetivo principal es proporcionar un conjunto de funcionalidades específicas que se pueden reutilizar en diferentes proyectos.

BPMN (Business Process Model and Notation). Conjunto de símbolos o iconos para estandarizar la representación de los elementos de forma gráfica y facilitar su comprensión dentro de un diagrama específico. Este método no solo brinda la capacidad de comprender los procesos, sino que también otorga a las organizaciones la habilidad de comunicar estos procedimientos de manera consistente y comprensible para todas las partes involucradas.

Clase en POO. Plantilla o modelo para crear objetos. Una clase define un conjunto de atributos (datos) y métodos (comportamientos) que los objetos pueden tener.

Código abierto. Tipo de software cuyo código fuente es accesible y disponible para el público en general. Cualquier persona puede examinar, modificar y distribuir el código fuente de manera libre y abierta. Promueve la colaboración, la transparencia y la participación comunitaria en el desarrollo de software, permitiendo a las personas usuarias mejorar, compartir y adaptar el programa según sus necesidades.

Control de versiones. Es la práctica de rastrear y gestionar los cambios en el código de software. Realiza un seguimiento de todas las modificaciones en el código en un tipo especial de base de datos.

Cross-Site Scripting (XSS). Es un software, fragmento de datos o una secuencia de comandos, mejor conocida como exploit de seguridad, que facilita a un atacante insertar código malicioso en un sitio web. Este código se ejecuta en los navegadores de las víctimas, lo que permite al atacante evadir los controles de acceso y suplantar a los usuarios legítimos.

Cuenta de Servicio. Cuenta no asociada a una persona usuaria que se crea con el propósito específico de facilitar la comunicación y la interacción entre múltiples aplicaciones de software, sistemas o servicios. Este tipo de cuenta actúa como intermediaria, permitiendo que las diversas entidades tecnológicas interactúen de



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	4 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

manera eficiente y segura, sin depender de la presencia o la intervención directa de una persona usuaria.

Dirección. Dirección de Gobierno Digital de la Dirección General de Gobierno Digital y Soporte Tecnológico de la Secretaría de Innovación y Gobierno Abierto del Municipio de Monterrey.

Django REST Framework. Biblioteca que complementa Django para construir APIs web de manera sencilla, flexible y rápida para la construcción de APIs web.

Django. Es un framework de desarrollo web de código abierto, escrito en Python y diseñado para facilitar el desarrollo rápido y la creación de aplicaciones web con un enfoque pragmático y limpio. Posee un conjunto completo de características listas para usar, lo que facilita la construcción rápida y eficiente de aplicaciones web y se centra en la simplicidad, la flexibilidad y la capacidad de reutilización de componentes.

Docstring. Comentario colocado en la primera línea de una definición de módulo, función, clase o método en Python. Se convierte internamente en el atributo `__doc__` del objeto al que describe. Este docstring sirve como documentación incorporada en el código fuente, proporcionando información sobre el propósito, funcionamiento y uso de la entidad definida.

Entorno de producción. Espacio donde se despliega la solución en producción para que se puedan procesar casos.

Entorno de pruebas. Configuración personalizada para realizar actividades de prueba de software. Los entornos de prueba están separados tanto del desarrollo como de los entornos de producción. El entorno aislado se especializa en analizar la calidad, la funcionalidad y el rendimiento del software.

Framework. Marcos de trabajo utilizados en ingeniería de software para llevar a cabo el desarrollo de software. El uso de frameworks facilita la agilización de los procesos de desarrollo al evitar la repetición de código. Además, asegura la implementación de buenas prácticas y promueve la consistencia en el código generado. Al proporcionar una base sólida y predefinida, los frameworks permiten a las personas desarrolladoras centrarse en la lógica específica de la aplicación, en lugar de ocuparse de tareas repetitivas o de bajo nivel.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	5 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

FrontEnd. Interfaz de usuario de un software o, en otras palabras, la porción de una aplicación destinada a interactuar directamente con las personas usuarias. Engloba los elementos visibles en la pantalla al acceder a un sitio web o aplicación tales como: tipografía, paleta de colores, adaptabilidad para diferentes dispositivos, así como los efectos y movimientos asociados con acciones del ratón, teclado y desplazamiento.

Función. Proceso que permite dividir el trabajo realizado por un programa en tareas más pequeñas y separadas de la parte principal.

Funciones Hash Criptográficas (CHF). Algoritmos matemáticos unidireccionales que asignan datos de cualquier tamaño a una cadena de bits de tamaño fijo. Se utilizan ampliamente en prácticas de seguridad de la información como firmas digitales, códigos de autenticación de mensajes, y otras formas de autenticación.

GitHub. Servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Este servicio permite colaborar y realizar cambios en proyectos compartidos, al mismo tiempo que mantienen un seguimiento detallado de su progreso.

Guía de estilo. Documento que compila normativas y patrones fundamentales vinculados con la presentación visual y estructural de una interfaz. Su propósito radica en proporcionar directrices específicas para el diseño y desarrollo de nuevas pantallas dentro de un contexto determinado. Aborda aspectos tales como la disposición de elementos visuales, la consistencia en la apariencia y la navegabilidad, con el objetivo de optimizar la experiencia de la persona usuaria y garantizar la coherencia estética en el sistema o aplicación en cuestión.

HTTP (Hypertext Transfer Protocol). El Protocolo de Transferencia de Hipertexto es un conjunto de reglas para formatear y procesar datos de la capa de aplicación para la transmisión de documentos hipermedia. Fue diseñado para la comunicación entre los navegadores y servidores web, aunque puede ser utilizado para otros propósitos también.

HTTPS (HyperText Transfer Protocol Secure). Es la versión segura del protocolo HTTP, que se utiliza para enviar datos de forma protegida entre el navegador y el sitio web al que se está conectado.

Indentación. Técnica de formateo de código comúnmente utilizada en la programación para mejorar la legibilidad y la estructura del código fuente. Consiste



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	6 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

en agregar espacios o tabulaciones al principio de cada línea de código para indicar la jerarquía y la estructura del programa.

JSON (JavaScript Object Notation). Formato ligero de intercambio de datos.

JWT (JSON Web Token). Estándar abierto basado en JSON, propuesto por IETF para la creación de tokens de acceso que permiten la propagación de identidad y privilegios, o claims en inglés.

Markdown. Lenguaje diseñado para facilitar la creación de contenido de manera simple y legible. Su objetivo principal es ofrecer una sintaxis clara y fácil de escribir, manteniendo al mismo tiempo un diseño legible en todo momento. En esencia, puede ser visto como un método de escritura que simplifica el proceso de formatear y estructurar el texto, permitiendo a las personas usuarias concentrarse en el contenido sin preocuparse demasiado por los detalles técnicos del formato.

Métodos en POO. Bloque de código que contiene una serie de instrucciones. Estas instrucciones se ejecutan cuando se llama a la función, especificando los argumentos necesarios para el método.

Métodos HTTP. Dentro del protocolo HTTP, son métodos de petición que se utilizan para especificar la acción deseada sobre un recurso particular. Estos métodos, como GET, POST, PUT y DELETE, proporcionan una forma estandarizada de comunicar las intenciones del cliente con respecto a cómo deben ser manipulados los recursos en un sistema de información.

Modelo Plantilla Controlador (MVT). Paradigma de ingeniería de software similar al MVC. Se usa para la implementación de interfaces de usuario, gestión de datos y control de lógica. A diferencia del MVC, en el MVT es el mismo framework el que se encarga de la parte del controlador.

Modelo-Vista-Controlador (MVC). Paradigma ampliamente empleado en el diseño de software, especialmente para la implementación de interfaces de usuario, gestión de datos y control de lógica. Este enfoque pone énfasis en la clara separación de responsabilidades entre la lógica de negocio (modelo), la presentación de la información (vista) y el flujo de control de la aplicación (controlador).

Módulos (Python). Colección de definiciones de variables, funciones y tipos, entre otros elementos, que pueden ser importados y utilizados desde cualquier programa.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	7 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

OpenAPI Specification (OAS). Define una interfaz estándar para las API RESTful, que permite a los seres humanos y los ordenadores descubrir y comprender las capacidades del servicio sin acceso al código fuente, documentación o a través de la inspección del tráfico de la red.

OWASP. Es una organización internacional sin ánimo de lucro que establece un estándar de concienciación de seguridad para aplicaciones web.

Parámetro de entrada. Permite pasar un valor de datos a la función o al procedimiento almacenado a quien realiza la llamada. Parte de un sistema que protege sus recursos, permitiendo que sólo sean usados por aquellos consumidores a los que se les ha concedido autorización para ello. Los recursos incluyen archivos y otros objetos de datos, programas, dispositivos o funcionalidades provistas por aplicaciones.

Patrones de Diseño de Software. Soluciones generales, reutilizables y aplicables a diversos problemas de diseño de software.

Pbkdf2. Es un Algoritmo de una función matemática de derivación de claves que permite cifrar contraseñas de forma segura para que se puedan almacenar debidamente en las bases de datos de una aplicación móvil o web.

Personas usuarias. Usuarios finales de un servicio digital, ya sea una persona servidora pública o la ciudadanía.

Plantilla (Template). Se utiliza como base para crear varias páginas web con un diseño similar, pero con distinto contenido.

Principio DRY (DRY). De sus siglas en inglés Don't Repeat Yourself (DRY), es un conjunto de protocolos, medidas y operaciones de desarrollo de software cuyo objetivo es eliminar la duplicación lógica.

Programación Orientada a Objetos (POO). Modelo de programación en el que el diseño de software se organiza alrededor de datos u objetos, en vez de usar funciones y lógica. Se enfoca en los objetos que las personas programadoras necesitan manipular, en lugar de centrarse en la lógica necesaria para esa manipulación. Un objeto se puede definir como un campo de datos con atributos y comportamientos únicos.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	8 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

Protocolo. Conjunto formal de estándares y normas que regulan tanto el formato como el control de la interacción entre dispositivos dentro de una red o sistema de comunicación. El objetivo principal es permitir la transmisión de datos entre estos dispositivos de manera eficiente y confiable.

Pruebas de estrés. Ensayos que evalúan el rendimiento del software más allá de los parámetros normales de funcionamiento. Esto implica someter el software a cargas de tráfico muy elevadas, como un aumento en el número de personas usuarias o transacciones, con el fin de entender su rendimiento en situaciones que exceden los límites de capacidad previstos. Estas pruebas son cruciales para identificar posibles cuellos de botella, problemas de rendimiento o fallos en la aplicación cuando se encuentra bajo estrés o cargas extremas.

Pruebas unitarias. Ensayos pequeños y rápidos que verifican la funcionalidad y seguridad del código de componentes individuales. Se ejecutan automáticamente con cada compilación para ofrecer retroalimentación inmediata sobre los cambios en el código.

Pull Request. Propuesta para combinar un conjunto de cambios de una rama con otra. En una solicitud de cambios, los colaboradores pueden revisar y analizar el conjunto propuesto de cambios antes de integrar los cambios en el código base principal. Las solicitudes de cambios muestran la diferencia o diferencias entre el contenido de la rama de origen y el contenido de la rama de destino.

Python. Lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis limpia y un código legible.

Rama (control de versiones). Se trata de bifurcaciones en el estado del código que crean nuevos caminos para la evolución del mismo, permitiendo trabajar en paralelo con otras ramas que puedan generarse. Es una de las herramientas fundamentales en Git para gestionar el código de manera más eficiente.

README. Convención comúnmente utilizada en el desarrollo de software para proporcionar documentación y descripciones acerca de una aplicación o proyecto. Generalmente, se trata de un archivo de texto con la extensión .txt (texto plano), .md (Markdown) o sin extensión. Acompaña a un programa y ofrece instrucciones detalladas sobre cómo instalar, iniciar o solucionar problemas relacionados con el mismo. Este archivo sirve como guía útil para las personas usuarias y desarrolladoras que interactúan con el software, facilitando su comprensión y uso adecuado.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	9 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

reCaptcha Servicio de Google que protege los sitios web del spam y los abusos al distinguir entre personas usuarias y robots automatizados. Utilizando aprendizaje automático y análisis avanzado de riesgos, reCAPTCHA representa una evolución del sistema tradicional de CAPTCHA.

Repositorio. Tipo de almacenamiento digital centralizado utilizable para realizar y administrar cambios en el código fuente de una aplicación. Un repositorio cuenta con características que permiten a las personas desarrolladoras rastrear con facilidad cambios en el código, editar archivos de manera simultánea y colaborar de forma eficiente en el mismo proyecto desde cualquier ubicación.

REST (Representational State Transfer). Estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

Software libre. Software que respeta la libertad de las personas usuarias y la comunidad para ejecutar, estudiar, modificar, copiar, distribuir, y mejorar el software. El acceso a su código fuente siempre debe estar disponible.

Software. Sistema formal de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.

SPAM. Cualquier forma de comunicación no solicitada enviada de manera masiva.

SQL injection. Es una vulnerabilidad que permite a las/los atacantes insertar instrucciones maliciosas en consultas SQL, comprometiendo la seguridad de una base de datos. Esto puede resultar en la obtención de información confidencial, manipulación de datos e incluso la toma de control total del sistema.

SQL. Es un lenguaje de programación para almacenar y procesar información en una base de datos relacional.

Sujetos Obligados. Todas las Dependencias y Entidades de la Administración Pública Municipal y Paramunicipal de Monterrey.

Token. Mecanismo utilizado para acceder a un recurso restringido electrónicamente. Se utiliza como complemento o en lugar de una contraseña.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	10 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

Typescript. Lenguaje de programación que es una extensión tipada de JavaScript. Es un superset de JavaScript, lo que significa que cualquier código JavaScript existente es también código TypeScript válido. Su principal característica es su sistema de tipos estáticos, que permite a las/los desarrolladores especificar tipos de datos para variables, parámetros de función, retornos de función y más.

UML (Unified Modeling Language). Herramienta esencial para la representación gráfica y la comunicación efectiva de la estructura y comportamiento de un sistema, particularmente en el ámbito del software orientado a objetos. Mediante un conjunto de símbolos y diagramas estandarizados, facilita la comprensión y el intercambio de ideas entre los diversos actores involucrados en el desarrollo de un sistema.

URL (Uniform Resource Locator). Dirección única y específica asignada a cada recurso disponible en la World Wide Web. Estas direcciones permiten que los navegadores se localicen y las personas usuarias visiten los recursos en línea.

Valor de retorno. Es el resultado producido por la ejecución de una función personalizada. Este resultado puede ser utilizado por otros módulos, procesos o funciones dentro de una aplicación de software para su posterior procesamiento o como valores de entrada.

Vue js. Marco de código del lenguaje de programación JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones.

IV. DESCRIPCIÓN

Este lineamiento es emitido y será actualizado por la Secretaría de Innovación y Gobierno Abierto, a través de la Dirección de Gobierno Digital de la Dirección General de Gobierno Digital y Soporte Tecnológico, de conformidad con lo dispuesto en el **L-SIG-GGD-08** *Lineamiento de Criterios y Estándares de Interoperabilidad*. Establece las directrices para implementar estándares, protocolos y prácticas en el proceso de desarrollo de software y su debida documentación como recurso para su sostenibilidad. Debe asegurar la legibilidad, mantenibilidad, eficiencia, seguridad y escalabilidad de las aplicaciones, plataformas y servicios digitales de la Administración Pública Municipal de Monterrey.

Este conjunto de directrices abarca desde la estructura y estilo del código, hasta las prácticas de documentación. Establece las bases para un desarrollo ágil y eficiente, que facilite la comprensión y colaboración, mejorando la experiencia general del



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	11 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

desarrollo de software, así como la eficacia y eficiencia de los recursos del Municipio de Monterrey asignados para este fin.

En el presente lineamiento se exponen y detallan los siguientes ejes rectores, mismos que guiarán el contenido del documento:

- Normas y Estándares
- Arquitectura
- Proceso de Desarrollo
- Pruebas y Control de Calidad
- Documentación del Desarrollo
- Gestión de Versiones y Control de Cambios
- Seguridad y Privacidad
- Mantenimiento y Actualizaciones
- Entrega y Despliegue

4.1. Normas y estándares.

Los Sujetos Obligados deberán apearse a las recomendaciones y convenciones de estilo de cada lenguaje de programación. Algunos ejemplos de guías de estilo para cada lenguaje de programación son:

- PEP8 (Python)
- Google Java Style Guide (Java)
- Microsoft C# Coding Conventions (C#)
- Google TypeScript Style Guide (TypeScript)

Estas incluyen reglas sobre la indentación, el espaciado, el uso de mayúsculas y minúsculas, entre otros. Por ende, se deberá seguir una línea similar acorde a la tecnología que se esté utilizando para facilitar la consistencia, legibilidad y mantenibilidad del desarrollo.

4.1.1. Especificación OpenAPI.

La especificación OpenAPI 3.0 Specification (OAS) define una interfaz estándar independiente del idioma para las API RESTful, que permite a las personas y a los sistemas descubrir y comprender las prestaciones del servicio sin acceso al código fuente, a la documentación, o a través de la inspección de tráfico de red. Si su definición es correcta, el consumidor



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	12 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

puede comprender el servicio remoto e interactuar con él con una mínima lógica de implementación.

Al desarrollar APIs, estas deben seguir el estandar OpenAPI Specification (OAS) versión 3.0 o superior.

4.1.2. De los comentarios.

Se deberán incluir comentarios descriptivos incrustados dentro del código para explicar la lógica, las decisiones y cualquier complejidad que pueda no ser evidente de inmediato. Siguiendo el formato que le aplique al definido, según el lenguaje de programación implementado en el desarrollo, en atención al punto 4.1.1 del presente lineamiento.

Estos comentarios deben ser de tipo Docstring siguiendo el formato definido según el lenguaje de programación implementado en el desarrollo. Debe evitarse usar comentarios sin aportes significativos, un ejemplo es: *# declaramos la variable*.

4.1.3. Evitar código duplicado.

Los Sujetos Obligados y la Dirección deberán seguir los principios DRY, así como de la programación orientada a objetos y los patrones de diseño de software, a fin de promover la reutilización del código en sus diferentes secciones. De igual manera, se deberán reutilizar las bibliotecas y partes de código ya desarrolladas para el Municipio de Monterrey en cuanto a software.

4.1.4. Manejo de excepciones.

Para garantizar la estabilidad del código del desarrollo, su claridad y mantenibilidad, todos los segmentos propensos a errores deben estar protegidos mediante el manejo de excepciones de código adecuadas, asegurando la captura y el tratamiento apropiado de excepciones de código específicas. Es recomendable utilizar cláusulas de limpieza para acciones finales, se debe evitar el uso excesivo de bloques de manejo de excepciones.

4.1.5. Validación de entradas.

Para prevenir ataques de inyección, los Sujetos Obligados y la Dirección deberán implementar validación de entradas, como SQL injection o cross-site scripting (XSS).



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	13 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

Se recomienda la utilización de paquetes de validación de entradas, la realización de pruebas de seguridad y el seguimiento de los estándares mencionados en el apartado 4.2.7. *Seguridad* del presente lineamiento.

Algunos ejemplos de validación de entradas pueden ser la validación del formato de un número de teléfono, la longitud de una dirección de correo electrónico, descartando entradas de información que no cumplan con los criterios del desarrollo, evitando códigos maliciosos que pueden ser utilizados en un ataque.

4.1.6. Manejo de contraseñas.

Las contraseñas no deberán ser almacenadas en texto plano, sino a través de la utilización de funciones hash criptográficas (CHF) que almacenan contraseñas de forma cifrada y segura. Para esto se recomiendan PBKDF2 y bcrypt.

4.1.7. Control de Acceso.

La Dirección deberá implementar un control de acceso mediante autenticación y autorización para garantizar que las personas usuarias de las plataformas y servicios digitales tengan los permisos necesarios, evitando así un acceso no autorizado.

4.1.8. Actualizaciones.

La Dirección y los Sujetos Obligados deberán mantener actualizadas todas las dependencias tecnológicas, mencionadas en el punto 4.5 del presente lineamiento, del desarrollo e implementar actualizaciones de seguridad. La periodicidad dependerá del caso, como se menciona en el punto 4.8.1 *Actualizaciones*.

4.1.9. Revisiones Regulares.

La Dirección deberá realizar revisiones de código de manera periódica con la finalidad de garantizar la conformidad con las pautas de calidad y documentación establecidas en este lineamiento, promoviendo la calidad y mantenibilidad del desarrollo. Se recomienda realizarlas cada 3 meses.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	14 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

4.2. Arquitectura.

La arquitectura de software se refiere a la estructura fundamental de un sistema de software, delineando la disposición y relación entre sus componentes para cumplir con los alcances establecidos. Es un marco conceptual que guía el diseño, desarrollo y mantenimiento del software, abordando decisiones críticas como la distribución de responsabilidades y la gestión de la complejidad. Este enfoque planificado y organizado optimiza la eficiencia del sistema, garantiza su cumplimiento de requisitos y facilita su evolución a lo largo del tiempo. En el contexto gubernamental, la descripción de la arquitectura de software es esencial para la toma de decisiones informadas y la gestión efectiva del sistema. Esta arquitectura deberá describir los siguientes aspectos mencionados.

4.2.1. Identificación de requisitos.

Durante el proceso de desarrollo de la arquitectura de software se deben definir claramente los requisitos del desarrollo por parte de la Dirección, y comprender las necesidades de los usuarios y las expectativas de rendimiento, seguridad y escalabilidad.

4.2.2. División en componentes.

La Dirección deberá descomponer el desarrollo en componentes lógicos, identificando módulos, capas y subsistemas. La Coordinación de Desarrollo asignará responsabilidades específicas a cada componente para una clara separación de funcionalidades.

4.2.3. Modelado de reglas de negocio.

Para proporcionar una comprensión completa del sistema y la representación gráfica de la arquitectura de software, los Sujetos Obligados y la Dirección deberán seguir el estándar BPMN. Este modelado de las reglas de negocio se deberá incluir en la documentación del desarrollo por los Sujetos Obligados y la Dirección de Gobierno Digital.

4.2.4. Estilo arquitectónico.

Previo al inicio del desarrollo, la Dirección y los Sujetos Obligados deberán seleccionar un estilo arquitectónico que asegure la mantenibilidad y escalabilidad del software. Se recomienda el estilo Modelo-Vista-Controlador o equivalente, por ejemplo, MVT en Django.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	15 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

4.2.5. Documentación clara.

La arquitectura de software deberá estar documentada de manera exhaustiva, proporcionando descripciones claras de cada componente, sus interacciones y flujos de datos. Debe ser actualizada a lo largo del ciclo de vida del software siguiendo el mismo procedimiento establecido en el punto 4.5 del presente lineamiento.

4.2.6. Mantenibilidad y escalabilidad.

La Dirección y los Sujetos Obligados deberán diseñar la arquitectura para facilitar futuras expansiones y modificaciones, considerando la escalabilidad del desarrollo, adaptabilidad a cambios en la carga de trabajo y cumplimiento de los requisitos establecidos en el presente lineamiento, así como los demás que resulten relativos y aplicables. Con la finalidad de asegurar la sustentabilidad de los desarrollos de software del Municipio, se recomienda la implementación de código abierto.

4.2.7. Seguridad.

Se deberán incorporar prácticas de seguridad desde el diseño del software, abordando posibles amenazas y vulnerabilidades en atención a lo dispuesto por el punto 4.7 del presente lineamiento. Dichas prácticas deben permitir la autenticación, autorización y responsabilidad auditable. Para esto, los Sujetos Obligados y la Dirección de Gobierno Digital deberán implementar mecanismos de control de acceso, cifrado y autenticación, según sea necesario. Las prácticas de seguridad deben ser implementadas de forma consistente en todos los desarrollos de software. Como referencia para implementar prácticas de seguridad en el desarrollo se recomienda utilizar la Guía OWASP* y para la mitigación OWASP top 10* en su totalidad, la cual expone los riesgos de seguridad más significativos en el desarrollo de software.

Al respecto, se deberán atender las consideraciones descritas en el punto 4.7 del presente lineamiento en el entendido que las características expuestas conforman las prácticas de seguridad y privacidad de manera enunciativa, más no limitativa.

4.2.8. Colaboración y comunicación.

A fin de fomentar una colaboración eficiente entre el personal de las Coordinaciones de la Dirección y/o terceras personas involucradas en el



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	16 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

desarrollo, se recomienda establecer un canal de comunicación sobre las decisiones arquitectónicas, responsabilidades y cambios.

Para hacer algún requerimiento de software, se formalizará la solicitud mediante oficio dirigido a la Dirección de General de Gobierno Digital y Soporte Tecnológico de la Secretaría de Innovación y Gobierno Abierto, de conformidad con lo dispuesto en el punto 4.3 del **L-SIG-GGD-07** *Lineamiento de Criterios y Estándares de Gestión de Software del Municipio*.

4.2.9. Evaluación continua.

Para la mejora continua de los desarrollos, la Coordinación de Desarrollo de la Dirección de Gobierno Digital de la Dirección de Gobierno Digital y Soporte Tecnológico, y los Sujetos Obligados recibirán retroalimentación de las Coordinaciones y Dirección de Gobierno Digital, así como de las personas usuarias, tal y como se describe en el **L-SIG-GGD-06** *Lineamiento de la Participación y Retroalimentación Ciudadana por Medios Digitales*. Dicha retroalimentación se deberá comunicar a través del canal de comunicación previamente establecido por los Sujetos Obligados y la Dirección.

La retroalimentación por parte de las Coordinaciones y Dirección de Gobierno Digital, así como los Sujetos Obligados, se podrá recibir a partir que el desarrollo se encuentre en el entorno de pruebas. Esta misma se podrá recibir por medio de las personas usuarias finales una vez que el desarrollo se encuentre en el entorno de producción.

4.3. Proceso de desarrollo.

El proceso de desarrollo de software no podrá iniciar hasta que se hayan revisado todos los requerimientos por parte de la Dirección a través de la Coordinación de Proyectos, de conformidad con lo dispuesto en el punto 4.4 Evaluación y factibilidad de la Solución Digital del **L-SIG-GGD-07** *Lineamiento de Criterios y Estándares de Gestión de Software del Municipio*.

Una vez que la revisión haya concluido, se procederá a elaborar el plan de trabajo para el desarrollo, en atención a lo dispuesto por el **L-SIG-GGD-03** *Lineamiento de Diseño de Interfaces y Experiencia de la Persona Usuaría*.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	17 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

En el plan de trabajo, la Dirección deberá definir las tecnologías a usar y los responsables de las diferentes funcionalidades del desarrollo. Dicho plan deberá tener en cuenta el paradigma de desarrollo de software API- First.

Para el desarrollo de BackEnd se recomienda usar Python con Django REST Framework; para el desarrollo del Frontend se recomienda usar Vue 3 o superior, como framework y TypeScript como lenguaje de programación en la lógica interna. Estas recomendaciones se realizan para aprovechar las piezas de software existentes y reutilizarlas, tal y como se describe en el punto 4.1.3 Evitar código duplicado del presente lineamiento permitiendo una mejora en la integración, escalabilidad y mantenimiento.

En caso de que se trate de un software interno, una vez definidas las tecnologías, responsables y los tiempos dentro del plan de acción establecido por las Coordinaciones de Proyectos y de Desarrollo de la Dirección, se podrá iniciar el desarrollo. En el caso de software que se desarrolle de manera externa, una vez definidas las tecnologías, responsables y tiempos dentro del plan de acción establecido por la Coordinación de Proyectos, se podrá iniciar el desarrollo.

4.4. Pruebas y control de calidad.

A fin de asegurar el correcto funcionamiento del software, la Coordinación de Experiencia Ciudadana, en conjunto con la Coordinación de Desarrollo de la Secretaría de Innovación y Gobierno Abierto, deberán abordar pruebas unitarias y pruebas de estrés con las siguientes consideraciones a cumplir:

- Definirse de acuerdo a su objetivo, ya sea para verificar la correcta funcionalidad, mejorar la robustez del sistema o simular la posible carga del entorno de producción.
- Realizarse de manera unitaria para verificar el correcto funcionamiento de las unidades más pequeñas del código, y emplear el uso de herramientas que correspondan al lenguaje de programación que se está utilizando.
- Especificar las condiciones iniciales y finales de las pruebas unitarias con anterioridad.
- Incluir casos de prueba positivos y negativos, es decir, casos donde se prueben condiciones donde el resultado sea el esperado, y otros donde no lo sea.
- Documentar las pruebas realizadas, los escenarios, resultados obtenidos y cualquier problema que se haya identificado.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	18 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

Para asegurar la calidad del desarrollo y su rendimiento lo más cercano a un entorno de producción, se deberá implementar un entorno de pruebas durante el proceso del desarrollo, dependiendo de las necesidades y funcionalidades establecidas en el plan.

4.5. Documentación del desarrollo.

La documentación del código facilita la colaboración entre las personas involucradas, la transferencia de conocimiento y el mantenimiento del código: Al mismo tiempo asegura la continuidad del desarrollo y el cumplimiento de los requisitos recibidos por la Dirección.

La Dirección deberá contar con la documentación del código de aquellos desarrollos que hayan sido realizados de manera interna, en estricto apego a los presentes lineamientos, así como demás disposiciones y ordenamientos jurídicos aplicables. En caso de que sea un desarrollo externo, la documentación debe ser requerida como parte de los entregables del proyecto en el contrato de la persona proveedora.

La documentación debe contener toda la información detallada de aquello que compone el desarrollo y cumplir con los siguientes apartados:

- **Introducción.**
Expresa el objetivo general del desarrollo y una breve descripción de lo que es el desarrollo sin entrar en tecnicismos.
- **Arquitectura.**
 - **Diagramas conceptuales.**
Definen el proceso por el que pasa el usuario final para el uso del servicio o plataforma, detallando cada uno de los pasos sin entrar en tecnicismos.
 - **Diagrama de infraestructura.**
Muestra los diferentes procesos, con sus respectivas decisiones de la plataforma mostrando los componentes tecnológicos del desarrollo, ya sea del FrontEnd o el BackEnd, y las dependencias de bibliotecas, según el caso. Este diagrama debe de seguir el estándar BPMN para el modelado de procesos de negocio.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	19 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- **Funcionalidad y responsabilidad.**

Define los diferentes componentes de la infraestructura del desarrollo, explicando sus funcionalidades y responsabilidades dentro BPMN del mismo.

- **Dependencias tecnológicas directas.**

Este apartado debe hacer referencia a todas las dependencias tecnológicas del código que se necesitan forzosamente para que el desarrollo pueda funcionar correctamente, como bibliotecas o paquetes dentro del desarrollo de software, y su versión de compatibilidad.

- **Dependencias tecnológicas con terceros.**

Este apartado debe de hacer referencia a todas aquellas bibliotecas que tiene el desarrollo dentro de sí mismo, que son actualizadas por un tercero o aquellas dependencias que proveen un servicio para el desarrollo, un ejemplo pudiera ser un servicio de autenticación, y en su caso definir la versión que utiliza el desarrollo en la actualidad.

- **Diagramas de clase.**

Representan las clases que interactúan entre sí y sus relaciones definiendo sus propios métodos internos implementados. Estos diagramas deben de seguir el formato estándar UML o algún estándar equivalente para el diseño de software.

• **Tareas**

- **Programables.**

Representan las tareas que se pueden realizar de manera automática por medio de algún servicio.

- **No programables.**

Son las tareas que se deben realizar de manera manual para el correcto funcionamiento de la plataforma. Esto puede ir desde la configuración de la plataforma, desde el panel administrador, hasta la actualización del contenido o creación de cuentas para alguna Dirección o Secretaría. Para esto último, referirse al **L-SIG-GGD-07** *Lineamiento de Criterios y Estándares de Gestión de Software del Municipio*.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	20 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- **Instalación y despliegue.**
Este apartado hace referencia a las instrucciones que debe de seguir todo aquel desarrollador que necesite que el software esté funcionando tanto de manera local, en su computadora, como en producción.
- **Gestión y operación.**
Hace referencia a los flujos y configuraciones en el panel de administración del desarrollo dependiendo el caso.
- **Casos de uso.**
Hace referencia a cómo el desarrollo será utilizado para satisfacer las necesidades de las personas usuarias o para alcanzar ciertos objetivos específicos.
- **Flujos y procesos.**
Define los diferentes flujos y procesos junto con sus respectivas tomas de decisiones dentro del desarrollo. Estos deberán ser diagramas que sigan el estándar BPMN para el modelado de procesos de negocio.
- **APIs.**
Hace referencia a todas las APIs internas que fueron desarrolladas para el consumo de información del desarrollo. Deben de ser documentadas siguiendo la especificación estándar Open API.
- **Código BackEnd.**
Esta sección deberá de contener todo el código correspondiente al BackEnd. Se deberá organizar con base en los módulos que componen el desarrollo en dicho apartado. Cada módulo deberá contener una descripción de las clases que lo componen junto con sus métodos y atributos implementados. Los atributos deben de ser identificados con el tipo de variable primitiva, así como el propósito del mismo.

En el caso de los métodos, éstos deben de describir los parámetros de entrada del mismo, el valor de retorno y una descripción de su funcionalidad. En este apartado, de igual manera, se deberá poder visualizar el código de cada clase o método implementado para su mejor entendimiento y análisis.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	21 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- **Código FrontEnd.**

Esta sección deberá contener todo el código correspondiente al FrontEnd. Se deberá organizar con base en la estructura de carpetas que tenga el código. Cada elemento que lo conforme deberá contar con su descripción, el código del template en caso de contar con él y la lógica del elemento a documentar. En caso del template, este debe ser desestructurado por las secciones que lo componen y su descripción. En cuanto al código de lógica, este debe documentar cada método implementado, junto con su nombre, la descripción y el código de dicha función. Para los métodos se debe de incluir la misma información descrita en la sección de Código BackEnd.

La documentación del código deberá cumplir con la información detallada previamente para cada sección aplicable. Es decir, el desarrollo puede no contar con un código BackEnd o alguna otra sección, por ende, no será necesario agregarlo en la documentación del desarrollo si dicha sección no hace parte de este mismo. Para generar la documentación de manera automática con base en Docstrings implementados en el código, se podrá elegir la herramienta de preferencia, dependiendo el lenguaje de programación. De no ser posible se pueden utilizar herramientas como Markdown para su generación manual.

4.6. Gestión de versiones y control de cambios.

La gestión de versiones permite conservar un histórico de las diferentes versiones del desarrollo y su evolución antes de una integración estable. Esto permite tener versiones estables e históricas de desarrollo a las cuales regresar en caso de algún imprevisto.

De esta manera, el control de cambios facilita la integración correcta entre distintas versiones, mitigando la aparición de errores en el software, la pérdida de código desarrollado o vulnerabilidades de seguridad y de cualquier otra índole.

Para el cumplimiento de este aspecto, existen diferentes herramientas que permiten este manejo de versiones y control de cambios. Se recomienda el uso de Github y sus servicios facilitando una colaboración eficiente y gestión efectiva durante el desarrollo. De igual manera, la herramienta podrá ser seleccionada a criterio del área o equipo involucrado, siempre y cuando la herramienta cuente con un histórico de cambios, asignación de roles y permisos



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	22 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

por cada persona usuaria y/o grupo, almacenamiento de repositorios y permita la colaboración en tiempo real.

Se deberá contar con una persona responsable encargada de integrar las diferentes versiones e integrar los cambios realizados por la Coordinación de Desarrollo; esto en el contexto de los desarrollos llevados a cabo de manera interna por la Dirección de Gobierno Digital.

Cuando se trate de un desarrollo de otra Coordinación de la Dirección de Gobierno Digital o proveedor para el municipio de Monterrey, el equipo involucrado deberá designar un responsable para dicha tarea. Las personas integrantes del equipo que estén trabajando en conjunto o de manera individual en un desarrollo deberán crear una versión del código (denominada rama en Github), relacionada con el contexto de la funcionalidad en cuestión, evitando así la modificación de la rama designada para el entorno de producción, de manera que la versión principal sea administrada por la coordinación de Desarrollo.

Cualquier cambio que se pretenda realizar por algún integrante de la Dirección y que se deba integrar a la rama principal, se deberá integrar por medio de una Pull Request a través del software de control de versiones que defina la Dirección solicitando una revisión y/o validación a la coordinación de Desarrollo y a la Dirección de Gobierno Digital.

4.7. Seguridad y privacidad.

La seguridad debe mantenerse en todo el ciclo de vida del desarrollo en cuanto a almacenamiento, tratamiento, privacidad, exposición, cifrado e intercambio de toda la información con base en permisos y usuarios según sea necesario para el desarrollo. Para esto, se deben cumplir las siguientes características.

- Toda aplicación o servicio web para el municipio de Monterrey, deberá ser implementado en el servidor para el entorno de producción definido por la Dirección, utilizando cifrado mediante HTTPS con un certificado firmado por una Autoridad de Certificación (CA) reconocida.
- En el caso de ser necesario que las aplicaciones web y servicios digitales del municipio de Monterrey deban contar con autenticación y autorización de usuarios finales, se deberá utilizar el servicio de autenticación y autorización de OpenID Connect (OIDC) del Municipio de Monterrey, "iD Mty".



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	23 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- Se utilizarán métodos HTTP adecuados para la transmisión de datos, así como el envío de parámetros en el cuerpo o en las cabeceras de la solicitud para no exponer información sensible en la URL de la API como correos de usuario o contraseñas. Evitar el uso del método GET para enviar información.
- Implementar medidas antifraude para evitar solicitudes SPAM, como el reCAPTCHA, para proteger las APIs contra ataques.
- Validar la identidad de las solicitudes con base en la autenticación y autorización del servicio OpenID Connect del municipio. En los casos de consumo de información a APIs con información sensible o confidencial, deben ir autenticadas mediante JWT emitido para el usuario autenticado para evitar intentos maliciosos de acceso o manipulación de datos ajenos.
- Se debe seguir la guía OWASP y se debe cumplir completamente al menos las OWASP Top 10.
- La información sensible de la base de datos debe ser accesible sólo para las personas usuarias autenticadas y autorizadas dueñas de esa información. En el caso de personas servidoras públicas, éstas solo podrán acceder a la información de otras personas usuarias si esto forma parte de sus atribuciones dentro de la plataforma, sistema o servicio digital en cuestión siempre que la regulación aplicable lo contemple para el cumplimiento de sus funciones dentro de la Administración Pública Municipal de Monterrey.
- Deben existir mecanismos de control de acceso como autenticación o autorización para las distintas APIs.
- Evitar el compartir credenciales de acceso en los repositorios del desarrollo, como bases de datos, cuentas de servicio o alguna otra información de configuración.
- Al momento de dar de alta a las personas servidoras públicas como usuarias para la gestión del desarrollo en cuestión, se deberá respetar el manual correspondiente al desarrollo.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	24 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

4.8. Mantenimiento y actualizaciones.

En el ciclo de vida del software, se llevará a cabo el proceso de mantenimiento y actualizaciones por parte de la Dirección con una periodicidad de al menos 3 meses. Esta práctica busca preservar la funcionalidad, seguridad y eficiencia del software, asegurando que continúe cumpliendo con los requisitos y expectativas establecidos. A través de estas intervenciones, se garantiza que el software evolucione de manera adaptativa, manteniéndose apto y relevante en un entorno tecnológico en constante cambio.

4.8.1. Actualizaciones.

Las actualizaciones en el desarrollo de software aseguran la relevancia, funcionalidad y seguridad del desarrollo de manera continua, haciéndolo sostenible a lo largo del tiempo. Por lo que la Dirección, a través de la Coordinación de Desarrollo, genera actualizaciones por lo menos cada 3 meses. Estas actualizaciones permiten mejorar continuamente la eficiencia, la experiencia de la persona usuaria y la resiliencia a posibles vulnerabilidades en términos de seguridad. Por lo que la calendarización y priorización de las actualizaciones será acorde a la necesidad de las mismas, pueden ser de diferente naturaleza con diferentes aproximaciones en cuanto a su implementación, estas se mencionan a continuación:

- **Solicitado desde los requerimientos.**

Las mejoras y ajustes del software se realizan de manera directa en respuesta a los requerimientos previamente establecidos en el punto 4.3, una vez que han sido recibidos y validados por la Coordinación de Proyectos de la Dirección, de conformidad con lo dispuesto en el punto 4.3.2 del **L-SIG-GOD-02** *Lineamientos para Visto Bueno Previo a la Adquisición de Software*, enfoque asegura que cada actualización esté alineada con las necesidades y expectativas definidas inicialmente por los requerimientos definidos en el punto 4.3, manteniendo un desarrollo coherente y orientado a cumplir con los objetivos establecidos.

- **Incidencia que requiere la actualización.**

Cuando la Dirección de Gobierno Digital identifica una situación que afecta la funcionalidad o seguridad de la plataforma o servicio, se procede directamente con la correspondiente actualización, misma



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	25 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

que dependerá de la incidencia que se identifique. Este enfoque garantiza una respuesta inmediata a los problemas identificados, asegurando la estabilidad y efectividad continua del software.

- **Solicitud por parte de terceros.**

En caso de que se requieran adecuaciones, ajustes o mejoras; se deberá atender a lo dispuesto por el punto 4.8. *Gestión y seguimiento a la solución digital* del **L-SIG-GOD-02 Lineamientos para Visto Bueno Previo a la Adquisición de Software**. El requerimiento será evaluado por la Dirección para determinar su prioridad y fecha integración en el plan de trabajo del desarrollo.

- **Actualización de dependencias tecnológicas.**

En caso de que la Dirección detecte que una dependencia tecnológica específica requiera una actualización, dicha Dirección a través de la Coordinación de Desarrollo, determinará la prioridad y se agendará una fecha para la integración de dicha actualización. Esto asegura que el software esté en sintonía con las últimas versiones de las bibliotecas utilizadas, mejorando la estabilidad, rendimiento o seguridad del sistema.

4.8.2. Mantenimiento.

Este proceso abarca una serie de actividades destinadas a corregir errores, mejorar el rendimiento, adaptarse a cambios en el entorno tecnológico y satisfacer las necesidades de las personas usuarias. El mantenimiento no solo asegura la estabilidad del software, sino que también proporciona las bases para futuras actualizaciones y expansiones. Es por esto que la Dirección asegurará el mantenimiento de forma continua a los desarrollos para el municipio de Monterrey. A continuación, se mencionan los criterios que se deberán respetar por parte de los Sujetos Obligados y la Dirección de Gobierno Digital, para brindar un mantenimiento de calidad a los desarrollos para el municipio de Monterrey.

- **Incidentes.**

Estos pueden variar en complejidad y origen. La identificación y corrección de estos mismos por parte de la Dirección y los Sujetos Obligados, es fundamental para asegurar la calidad y estabilidad del software. Las incidencias durante el desarrollo se deberán gestionar



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	26 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

a través del sistema de incidencias de GitHub, o la herramienta de control de versiones utilizada en el repositorio del desarrollo. Dichas incidencias se deberán reportar por los Sujetos Obligados a las personas responsables del desarrollo mediante oficio, en atención a lo dispuesto en el punto 4.8.2 *Errores, fallas e incidencias en la Solución Digital* del **L-SIG-GOD-02** *Lineamientos para Visto Bueno Previo a la Adquisición de Software*.

El reporte de dicho error en el desarrollo deberá contener la información descrita a continuación.

- **Descripción del incidente.**

Detallar de manera clara y específica los síntomas, el contexto y cualquier información relevante que facilite la comprensión del problema por parte de la Coordinación de Desarrollo. La claridad en la descripción agiliza el proceso de identificación, replicación y corrección del incidente por parte la Coordinación de Desarrollo de la Dirección.

- **Captura de pantalla.**

Se solicita adjuntar una captura de pantalla o breve video, en la medida de lo posible, que ilustre visualmente el incidente reportado. Una imagen puede proporcionar una comprensión rápida y clara de la situación, facilitando la tarea de la Coordinación de Desarrollo de la Dirección al visualizar el problema directamente.

- **Plataforma.**

Se deberá proporcionar información sobre la plataforma o servicio digital que presenta dicho error, así como la URL específica donde se ha identificado el problema. Esta precisión permite una rápida ubicación y réplica del incidente por parte de la Coordinación de Desarrollo de la Dirección, agilizando el tiempo de respuesta y su solución.

Una vez recibido el reporte con la información previamente detallada, la Coordinación de Desarrollo brindará atención de acuerdo a las funcionalidades que sean afectadas por el incidente, y bajo la observancia del nivel de prioridad que representa. A



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	27 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

continuación, veremos cuáles son los niveles de prioridad que define la Dirección:

Nivel de prioridad	Descripción
Alta	El incidente afecta una función principal de la plataforma o servicio digital.
Media	El incidente afecta la experiencia de la persona usuaria en la plataforma o servicio digital.
Baja	No afecta a las anteriores o puede ser considerado una mejora o arreglo menor.

- **Refactorización.**

La refactorización en la vida del software es un proceso breve, pero fundamental que implica la reestructuración del código sin cambiar su funcionalidad externa. Este enfoque busca mejorar la legibilidad, eficiencia y mantenibilidad del código, contribuyendo a un desarrollo continuo y eficaz del software.

- **Proceso de refactorización.**

Por cada una de las actualizaciones que se realicen al desarrollo de software, la Dirección realizará una revisión del código del desarrollo, con la finalidad de encontrar áreas de oportunidad que permitan mejorar la eficiencia, la mantenibilidad, así como la eficiencia del código y su legibilidad. En caso de identificar áreas de oportunidad, la Dirección a través de la Coordinación de Desarrollo, evaluará la refactorización del código y se generará una calendarización para su debida atención.

Cuando se trate de un desarrollo por un(a) proveedor(a) externo(a), éste(a) deberá seguir el proceso de refactorización dentro de su propia estructura organizacional.

4.9. Entrega y despliegue.

La entrega y despliegue en el desarrollo de software constituyen las etapas que culminan el proceso de creación y preparan la aplicación para su utilización efectiva.

Estas fases son esenciales para garantizar una transición fluida del desarrollo a la producción, asegurando que la aplicación está lista para su uso. La entrega y



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	28 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

despliegue eficiente no solo facilitan la accesibilidad al software, sino que también establecen las bases para futuras actualizaciones y mejoras. En conjunto, estas etapas representan el punto culminante del esfuerzo de desarrollo, marcando el inicio de la interacción real de la persona usuaria con la aplicación.

4.9.1. Entrega.

La fase de entrega en el desarrollo de software marca el punto de transición crucial entre el proceso de desarrollo y la puesta en funcionamiento del producto. En la etapa de entrega del software la Dirección o la persona desarrolladora responsable deberá incluir en sus entregables por lo menos lo siguiente: documentación del desarrollo, manuales de uso, manuales de administración y se dejará la implementación en el entorno de producción; por lo tanto, se deberá entregar una url respectiva de producción y en caso de ser necesario, urls administrativas y credenciales de acceso. Cabe mencionar que el software no podrá ser entregado hasta que se entregue la documentación completa del desarrollo bajo la observancia del apartado 4.5 *Documentación del desarrollo* del presente lineamiento.

El cumplimiento de estos requisitos será fundamental para la transversalidad de las operaciones entre las diferentes dependencias del Municipio de Monterrey y la información que estas administren, relativas al desarrollo de dicho software. En caso de incumplimiento de alguno de los puntos contemplados dentro de los ejes, se requerirá la valoración técnica por parte de la Dirección.

- **Documentación del código.**

Se entregará en un sitio de documentación navegable y un archivo README que existirá en el repositorio del desarrollo escrito en Markdown. En este archivo se detallarán las dependencias tecnológicas directas y con terceros, la configuración para activar el funcionamiento del desarrollo, una breve descripción de lo que es el desarrollo, así como la URL que dirija al sitio de la documentación detallada del desarrollo en cuestión. La persona usuaria deberá de revisar el documento y comentar cualquier duda que tenga con la finalidad de que sea explicada o cambiar dicha parte del documento, a consideración de la Dirección, esto con el fin de buscar el mejor entendimiento del lado de la/el cliente.



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	29 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- **Manuales de usuario y administración.**

La Dirección deberá entregar manuales de usuario y manuales de administración como sitios de documentación navegable cuando sean desarrollo interno, en caso contrario, la/el proveedor responsable de dicho desarrollo deberá cumplir con la entrega a la que se hace referencia. De manera complementaria, se podrá entregar un archivo README dentro del repositorio del desarrollo o un documento en formato PDF. Lo anterior con la finalidad de explicar a la persona usuaria sobre las funcionalidades y las diferentes interfaces del desarrollo del software.

El manual del usuario debe de estar detallado de la manera más simple, pero a la vez entendible, es decir, con un lenguaje ciudadano que permita el entendimiento de los desarrolladores. El documento deberá contener imágenes del funcionamiento del software y explicar de una manera sencilla para su fácil comprensión.

- **Paquetes de instalación.**

Los paquetes de instalación son indispensables para que el software funcione correctamente, sin estos el software arrojaría errores y afectaría total o parcialmente su correcto funcionamiento. Esta información se integrará dentro del archivo README del repositorio, de acuerdo con el punto que antecede. Se adjuntará de igual manera, la versión de cada una de las dependencias tecnológicas directas y/o con terceros.

4.9.2. Despliegue.

La fase de despliegue en el desarrollo de software marca el momento en el cual la aplicación se implementa y pone en funcionamiento en un entorno operativo. Durante este proceso, el software es trasladado desde el ambiente de desarrollo a un entorno de producción por parte de la Dirección, donde estará accesible para las personas usuarias finales.

El despliegue no sólo implica la transferencia del código, sino también la realización de pruebas exhaustivas para garantizar su correcto funcionamiento en el entorno de producción. Dichas pruebas exhaustivas comprenden como mínimo las pruebas unitarias y pruebas de estrés, que deberán llevarse a cabo por parte de la Dirección de



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	30 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

Gobierno Digital como paso previo al despliegue en el entorno de producción. Esta fase es esencial para asegurar que la aplicación está lista y apta para su uso generalizado, y que cumpla con los estándares de rendimiento y seguridad establecidos durante el desarrollo.

- **Pruebas de software.**

La Dirección pondrá a prueba el software, realizando las pruebas definidas en el punto 4.4 del presente lineamiento, de manera que permitan comprobar el correcto funcionamiento. En caso de que la Dirección detecte algún incidente, este se reportará mediante un correo electrónico a la Coordinación encargada del proyecto.

- **Cumplimiento de requerimientos.**

La Dirección verificará que el software cumpla cada uno de los requerimientos, y que opere de manera funcional para continuar con su aprobación previo al despliegue.

En caso de que exista algún limitante o incidente, la Dirección podrá regresar al proceso de desarrollo, esto con el propósito de asegurar el cumplimiento de los requisitos del desarrollo. La Dirección deberá informar a los Sujetos Obligados por medio del canal de comunicación oficial.

- **Aprobación.**

El software deberá ser revisado por la Dirección; de no detectarse errores, el software se desplegará en el entorno de producción. En caso de identificar incumplimiento en los entregables o requisitos de desarrollo, la Dirección notificará al responsable de desarrollo mediante oficio, la situación y se deberá iniciar un nuevo proceso de desarrollo para implementar las correcciones comunicadas.

4.10. Toda información sobre estándares y/o tecnologías de desarrollo de software deberán ser consultados directamente en los portales oficiales respectivos, los cuales se encuentran en la fracción **VI Referencias y/o Bibliografía** del presente lineamiento.

V. ANEXOS

N/A



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	31 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

VI. REFERENCIAS Y/O BIBLIOGRAFIA

- **L-SIG-GGD-06** *Lineamiento de la Participación y Retroalimentación Ciudadana por Medios Digitales.*
- **L-SIG-GGD-07** *Lineamiento de Criterios y Estándares de Gestión de Software del Municipio.*
- **L-SIG-GGD-08** *Lineamiento de Criterios y Estándares de Interoperabilidad.*
- **L-SIG-GOD-02** *Lineamientos para Visto Bueno Previo a la Adquisición de Software.*
- **L-SIG-GOD-04** *Lineamiento de Contenido para Sitios Web Institucionales.*
- ¿Qué es un archivo README? (S/f). Filext.com. <https://filext.com/es/extension-de-archivo/README>
- ¿Qué es un Repositorio?. (s/f). Amazon.com. <https://aws.amazon.com/es/what-is/repo/>
- ¿Qué es una función hash criptográfica? (2015, noviembre 10). Ssl.com; SSL Corp. <https://www.ssl.com/es/preguntas-frecuentes/%C2%BFQu%C3%A9-es-una-funci%C3%B3n-hash-criptogr%C3%A1fica%3F/>
- ¿Qué es una función hash criptográfica? (2015, noviembre 10). Ssl.com; SSL Corp. <https://www.ssl.com/es/preguntas-frecuentes/%C2%BFQu%C3%A9-es-una-funci%C3%B3n-hash-criptogr%C3%A1fica%3F/>
- ¿Qué es una URL? Aquí te lo contamos. (2020, diciembre 2). Edix España. <https://www.edix.com/es/instituto/que-es-url/>
- Acerca de las solicitudes de incorporación de cambios (2024). Github, Inc. <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>
- Aprende Python. (s/f). Python.org. <https://es.python.org/aprende-python/>
- Atlassian. (s/f). Qué es el control de versiones. Atlassian. Recuperado el 21 de marzo de 2024, de <https://www.atlassian.com/es/git/tutorials/what-is-version-control>
- Bach, J. (2021). Django: Una guía completa para desarrollar sitios web con Django. Independently Published
- Canelo, M. M. (2020, junio 24). Qué son los Patrones de Diseño de software / Design Patterns. *Profile Software Services*. <https://profile.es/blog/patrones-de-diseno-de-software/>
- Carolina, D. (2022, diciembre 23). ¿Qué es reCAPTCHA? Todo lo que necesitas saber. Tutoriales Hostinger. <https://www.hostinger.mx/tutoriales/que-es-recaptcha>



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	32 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- Christie, T. (s/f). Home - Django REST framework. Django-rest-framework.org. Recuperado el 21 de marzo de 2024, de <https://www.django-rest-framework.org/>
- Cross-site scripting (XSS). (s/f). MDN Web Docs. Recuperado el 21 de marzo de 2024, de https://developer.mozilla.org/en-US/docs/Glossary/Cross-site_scripting
- ESET. (s. f.). ESET. <https://www.eset.com/es/caracteristicas/spam/>
- Fenton, S. (2013). TypeScript for JavaScript programmers. Lulu.com. <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
- Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.
- Fundamentos de programación/Tipos de datos primitivos. (s/f). Wikiversity.org. https://es.wikiversity.org/wiki/Fundamentos_de_programaci%C3%B3n/Tipos_de_datos_primitivos
- Godbolt, M. (2016). Frontend architecture for design systems: a modern blueprint for scalable and sustainable websites. " O'Reilly Media, Inc."
- Godbolt, M. (2016). Frontend architecture for design systems: a modern blueprint for scalable and sustainable websites. " O'Reilly Media, Inc."
- Graham, D. (s.f.). Foundations of Software Testing: ISTQB certification. Cengage.
- Graham, D. (s.f.). Foundations of Software Testing: ISTQB certification. Cengage.
- Guelmann, A. (2023, enero 17). ¿Qué son las cuentas de servicio? Silverfort. <https://www.silverfort.com/es/glossary/service-account/>
- Guías de estilo: diseño, normalización y usabilidad. (s/f). Desarrolloweb.com. <https://desarrolloweb.com/articulos/1513.php>
- Gustavo, B. (2019, abril 12). Qué es GitHub y cómo empezar a usarlo. Tutoriales Hostinger. <https://www.hostinger.mx/tutoriales/que-es-github>
- IBM Documentation. (2024, enero 25). Ibm.com. <https://www.ibm.com/docs/es/baw/20.x?topic=overview-case-development-production-environments>
- IBM documentation. (s. f.). <https://www.ibm.com/docs/es/app-connect/12.0?topic=apis-openapi-30>
- Imagar. (2021, noviembre 9). Imagar Solutions Company - Soluciones informáticas y servicios Web. Imagar Solutions Company. <https://www.imagar.com/blog-desarrollo-web/que-es-el-protocolo-en-informatica/>
- Inába, S. (2023, diciembre 30). Bibliotecas, Librerías y Módulos en Programación: Diferencias e Importancia. Soluciones Inába. <https://www.inabaweb.com/bibliotecas-librerias-y-modulos-en-programacion-diferencias-e-importancia/>



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	33 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- Introducción a JSON. (s/f). JSON Org. <https://www.json.org/json-es.html>
- Markdown - La guía definitiva en español. (2015, agosto 24). Markdown. <https://markdown.es/>
- Métodos de petición HTTP. (s/f). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>
- Módulos — Programación. (s/f). Usm.cl. Recuperado el 21 de marzo de 2024, de <http://progra.usm.cl/apunte/materia/modulos.html>
- OpenAPI Specification. (s/f). Swagger.io. <https://swagger.io/specification/>
- OWASP Developer Guide. (s/f). OWASP Foundation, Inc. <https://owasp.org/www-project-developer-guide/release/>
- OWASP Top Ten. (s/f). OWASP Foundation, Inc. <https://owasp.org/www-project-top-ten/>
- Pathak, A. (2021, julio 8). Las 26 mejores herramientas de pruebas de rendimiento. Kinsta®; Kinsta. <https://kinsta.com/es/blog/herramientas-pruebas-rendimiento/>
- Pruebas unitarias. (2023, octubre 11). Parasoft. <https://es.parasoft.com/solutions/unit-testing/>
- Puigmartí, E. L. (s/f). Docstrings, documentando el código · Tkinter. Gitbooks.io. <https://enriquelazcorreta.gitbooks.io/tkinter/content/desarrollo-de-aplicaciones/docstrings-documentando-el-codigo.html>
- Qué es BPMN - Definición, Elementos y Propósito. (s/f). Edrawsoft.com. <https://www.edrawsoft.com/es/what-is-bpmn.html>
- Qué es Framework. (s/f). Arimetrics. <https://www.arimetrics.com/glosario-digital/framework>
- Qué es una autoridad de certificación (CA)? (s/f). Digicert.com. <https://www.digicert.com/es/faq/compliance/what-is-a-certificate-authority-ca>
- Qué es una función hash criptográfica? (2015, noviembre 10). Ssl.com; SSL Corp. <https://www.ssl.com/es/preguntas-frecuentes/%C2%BFQu%C3%A9-es-una-funci%C3%B3n-hash-criptogr%C3%A1fica%3F/>
- Qué es una URL? Aquí te lo contamos. (2020, diciembre 2). Edix España. <https://www.edix.com/es/instituto/que-es-url/>
- Redondo, P. C. (2017, octubre 12). ¿Qué es SQL Injection? Openwebinars.net. <https://openwebinars.net/blog/que-es-sql-injection/>
- Redsauce: Blog. (s/f). Redsauce.net. <https://www.redsauce.net/blog/es/api-first-design>
- rwestMSFT. (s/f). Parámetros. Microsoft.com. <https://learn.microsoft.com/es-es/sql/relational-databases/stored-procedures/parameters?view=sql-server-ver16>



LINEAMIENTO DE CALIDAD DE DESARROLLO DE SOFTWARE Y SU APROPIADA DOCUMENTACIÓN

CÓDIGO:	L-SIG-GGD-11
VERSIÓN:	01
EMISIÓN:	23/05/24
PÁGINA:	34 de 34

SECRETARÍA DE INNOVACIÓN Y GOBIERNO ABIERTO

- UML, lenguaje de modelado gráfico. (2018, octubre 26). IONOS Digital Guide; IONOS. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>
- Universidad Europea. (2022, agosto 24). Programación orientada a objetos. <https://universidadeuropea.com/blog/programacion-orientada-objetos/>
- Urrutia, D. (2020, enero 30). Qué es Https - Definición, significado y ejemplos. Arimetrics. <https://www.arimetrics.com/glosario-digital/https>

VII. CONTROL DE CAMBIOS

VERSIÓN	FECHA	MOTIVO
01	23/05/24	Creación del lineamiento.