###### **BUENAS PRÁCTICAS DE DESARROLLO, ESTÁNDAR DE CODIFICACIÓN DE SOFTWARE Y SEGURIDAD DE LA INFORMACIÓN**

1. **METODOLOGÍA DE DESARROLLO**

**Metodología de Programación Extrema XP:** Metodología adoptada por la Dirección de Tecnologías de la Información y las Comunicaciones de desarrollo ágil, cuyo objetivo principal es entregar un software de calidad controlado por las necesidades de los usuarios, permite un desarrollo y gestión de proyectos con eficacia, flexibilidad y control, aumenta la productividad a la hora de desarrollar un proyecto de software dando prioridad a los trabajos que dan un resultado directo.

Contempla las siguientes fases:

* **Fase de exploración:** El usuario plantea a grandes rasgos la actividad que desea que haga el sistema y que es de interés para la entrega del producto.
* **Fase del planeamiento:** Se detalla y priorizan las historias de usuario, se acuerda el alcance del reléase o modulo (iteraciones). Los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma.
* **Fase de producción:** Etapa de desarrollo, prueba y comprobación extra del funcionamiento del sistema antes de que éste  
  se pueda liberar al cliente. En esta fase, se pueden presentar cambios y debe tomarse la decisión de si se incluyen o no en el reléase actual.
* **Fase de mantenimiento:** Mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el usuario.
* **Fase final:** Ocurre cuando el usuario no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura.

**Programación individual:** Codificación de una funcionalidad elaborada por una persona o programador.

**Programación en parejas**: Involucra dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado.

1. **BUENAS PRÁCTICAS DE DESARROLLO**

* Utilizar nombres descriptivos para la definición de variables, métodos, funciones, clases.
* No repetir código: Funciones, métodos, clases, no repetir librerías ni documentación.
* Mantener al mínimo el número de niveles en las instrucciones anidadas.
* Hacer test unitarios.
* Evitar usar métodos con muchos parámetros.
* Comentar el código, funciones, definiciones y lógica compleja.
* Evitar comentarios innecesarios
* Las librerías, Componentes, Servicios o bibliotecas de uso general deben estar debidamente documentadas explicando su propósito, funciones, parámetros de entrada y salida, etc. La documentación de estos componentes debe ser almacenada en el repositorio designado por la Subdirección de Gestión de la Información.
* Proteger a las variables y los recursos compartidos de acceso concurrente inapropiados.
* Impedir que los usuarios generen nuevo código o alteren el código existente.
* La información involucrada en las transacciones de los servicios de las aplicaciones se deben proteger para evitar la transmisión incompleta, el enrutamiento errado, la alteración no autorizada de mensajes, la divulgación no autorizada, y la duplicación o reproducción de mensajes no autorizada.
  + Se debe mantener actualizado el software, frameworks y plugins que se utiliza en el desarrollo.
  + Restringir la escritura de archivos desde la web a través de la asignación de permisos de solo lectura.
  + Implementar mensajes genéricos de error, que no revelen información acerca de la tecnología usada, excepciones o parámetros que dispararon el error específico, los cuales deberán ser comprensibles por parte de las personas, incluyendo la accesibilidad para las personas con discapacidad.
  + Proteger el binario de la aplicación, a través de métodos de ofuscación que impidan realizar procedimientos de ingeniería inversa (reversing) para analizar la lógica de la aplicación.
  + Restringir la escritura de archivos en el servidor web a través de la asignación de permisos de roles y los privilegios asociados.
  + Cumplir con los estándares definidos para la integración al Portal Único del Estado Colombiano GOV.CO, incluyendo la validación de la codificación, en caso de que aplique.
  + Incluir lenguaje común de intercambio para la generación y divulgación de la información y datos estructurados y no estructurados dispuestos en medios electrónicos, como los sitios web de los sujetos obligados y el Portal Único del Estado Colombiano GOV.CO, en caso de que aplique.
  1. **BUENAS PRÁCTICAS PARA UNA CODIFICACION SEGURA:**

Las practicas referenciadas en este numeral son extractadas “*Prácticas seguras de codificación OWASP-Guía de referencia rápida*” <https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf>.

Cumplir con la estandarización de código fuente para portales web, siguiendo las buenas prácticas del W3C (World Web Wide Consortium), de forma que permita la correcta visualización de la información a los usuarios.

Adoptar validadores HTML y CCS para la continua revisión del sitio web y su mejora continua, a través de las buenas prácticas del W3C (World Web Wide Consortium).

* **Validación de entradas:**
* Validar los tipos, rangos, longitud de datos esperados.
* Validar todas las entradas en una lista "blanca" de caracteres permitidos, siempre que sea posible.
* Todos los errores de validación deben resultar en el rechazo de entrada.
* Validar todos los datos de usuario proporcionados antes del procesamiento, incluyendo todos los parámetros, URL y contenido de la cabecera HTTP.
* Validar los tipos de datos esperados.
* En los parámetros de entrada eliminar etiquetas, saltos de línea, espacios en blanco y otros caracteres especiales.
* Restringir el tamaño de los archivos que se suben a las aplicaciones.
* Sanitización de caracteres especiales (secuencia de Escape de variables en el código de Programación).
* **Codificación de salidas:**
* Depurar toda la salida de los datos no confiables a las consultas de SQL, XML, y LDAP.
* Utilizan un estándar, la rutina probada para cada tipo de codificación de salida.
  + **Autenticación y gestión de contraseñas:**
* Requerir la autenticación de todas las páginas y los recursos, salvo los destinados específicamente a ser pública.
* Todos los controles de autenticación deben aplicarse en un sistema de confianza (LDAP).
* Si la aplicación gestiona un almacén de credenciales, se debe garantizar que la tabla o archivo que almacena las contraseñas y claves contengan controles criptográficos y sean manejados únicamente por la aplicación.
* Validar los datos de autenticación.
* Hacer cumplir los requisitos de complejidad y longitud de contraseñas establecidas por la política aplicada por la Contraloría de Bogotá.
* La introducción de contraseña se deberá cubrir en la pantalla del usuario.
* Forzar desactivación de la cuenta o sesión después de un número establecido de intentos de acceso no válidos.
* Restablecimiento de contraseñas y operaciones de cambio requiere el mismo nivel de control como la creación de cuentas y la autenticación.
* Las contraseñas temporales y enlaces deben tener un tiempo de caducidad corta.
* Prevenir reutilización de contraseña.
  + **Gestión de sesión:**
* El identificador de sesión debe realizarse siempre en un sistema de confianza (LDAP).
* La funcionalidad de cierre de sesión debe terminar completamente la sesión o conexión asociada.
* Establecer un tiempo de espera de inactividad de la sesión que sea lo más corto posible.
* No permitir conexiones persistentes y hacer cumplir las terminaciones de sesiones periódicas, incluso cuando la sesión está activa.
* No permitir conexiones simultáneas con el mismo ID de usuario.
* **Control de acceso:**
* Utilizar un único componente de todo el sitio para comprobar la autorización de acceso (LDAP). Esto incluye las bibliotecas que requieren servicios de autorizaciones externos
* Denegar el acceso si la aplicación no puede acceder a la información de la configuración de seguridad.
* Restringir el acceso a direcciones URL, funciones, objetos, servicios, datos protegidos sólo a los usuarios autorizados.
* Implementar auditoría de cuentas y hacer cumplir la desactivación de cuentas no utilizadas.
* **Manejo de log de cambios**
* No revelar información confidencial en las respuestas de error, incluyendo detalles del sistema, los identificadores de sesión o información de la cuenta.
* Restringir el acceso a los registros de sólo las personas autorizadas.
* Registrar todos los intentos de autenticación, especialmente los fracasos.
* Registrar todas las fallas de control de acceso.
* Registrar todos los eventos de alteración aparentes, incluyendo cambios inesperados en el estado de los datos
* Registrar sesiones de tokens no válidos o caducados.
* **Protección de datos:**
* Implementar los privilegios para restringir a los usuarios sólo la funcionalidad, datos e información del sistema que se requiere para llevar a cabo sus tareas
* Proteger de acceso no autorizado los datos confidenciales almacenados en caché o temporales almacenados en el servidor.
* No almacenar contraseñas, cadenas de conexión u otra información confidencial en texto claro o de cualquier manera segura en el lado del cliente.
* No incluya información confidencial en parámetros de la petición.
* Desactivar las funciones de auto completar en los formularios que se espera que contengan información sensible, incluyendo la autenticación.
* **Seguridad de comunicación:**
* Implementar mecanismos de cifrado para la transmisión de la información clasificada como reservada, privada, semiprivada, sensible, confidencial.
* **Base de datos**
* Utilizar consultas con parámetros fuertemente tipadas.
* Implementar el cifrado para la transmisión de toda la información sensible.
* Las cadenas de conexión no deben estar codificado dentro de la aplicación.
* Cerrar la conexión tan pronto como sea posible.
* Eliminar o cambiar todas las contraseñas administrativas de base de datos por defecto. Utilizar contraseñas fuertes / frases o implementar autenticación de múltiples factores.
* **Manejo de archivos:**
* Requerir autenticación antes de permitir que un archivo para ser cargado.
* Desactivar privilegios de ejecución en los directorios de subida de archivos.
* Explorar los archivos subidos en busca de virus y malware.
* **Manejo de memoria**
* Libere adecuadamente la memoria asignada a la finalización de funciones y en todos los puntos de salida.

1. **ESTÁNDAR DE CODIFICACIÓN DE SOFTWARE**
   1. **Prefijos**

Es obligatorio utilizarlos para denominar cada tipo de elemento, su función es identificar diferentes tipos de campos, objetos, datos. Se puede aplicar para: directorios y estructura de archivos, archivos, clases, interfaces, servicios, formularios, funciones, etc.

Los prefijos deben estar compuestos por 2 letras y los nombres de los elementos que acompañan el prefijo deben reflejar de manera precisa su contenido y función, los prefijos se toman como una nueva palabra adicionada al elemento que se quiere describir y por lo tanto deben respetar la sintaxis del elemento.

**Reglas de Generación de prefijos:**

Determinar el prefijo según las siguientes condiciones:

* Si la palabra empieza en letra y tiene al menos 2 consonantes, se escogen las dos primeras consonantes, ejemplo:
  + Texto : Tx\_Nombre
  + Combo: Cm\_Listado
* Si la palabra inicia en vocal y tiene al menos una consonante, se escoge la primera vocal y consonante, ejemplo:
  + Update : Up
* Si la palabra tiene menos 2 consonantes, se escogen las consonantes y/o vocales en el orden que aparezcan
  + Aro: Ar\_Centro
* En caso que se repita la nomenclatura se debe escoger la primera y tercera consonante o vocal en el orden que aparezcan evitando duplicidad de prefijos, ejemplo: **U**pdatePanel – Prefijo: Up y **U**p**d**ateProgress: Ud, esto se debe documentar en comentario.

| TIPO | PREFIJO | EJEMPLO |
| --- | --- | --- |
| Esquema | SC\_ | [Servidor].[SC\_Nomina\_Cb].[Base\_datos] |
| Base de datos | BD\_ | [Servidor].[Schema].[BD\_Nomina] |
| Tablas | TB\_ | [Servidor].[Schema].[Base\_Datos].[TB\_Funcionarios] |
| Vistas | VW\_ | [Servidor].[Schema].[Base\_datos].[VW\_Funcionarios\_Pensionados] |
| Llave Primaria | PK\_ | PK\_CodFuncionario |
| Llave Foránea | FK\_ | FK\_CodCiudad |
| Llaves Unique | UK\_ | UK\_CodDepartamento |
| Store procedures | SP\_ | SP\_ReporteSalidasFuncionarios |
| Trigger | TR\_ | TR\_AuditoriaEstado |

* 1. **Nomenclatura para elementos de base de datos**

El prefijo: debe estar en mayúscula, separado del nombre por guion bajo (\_).

El nombre: debe reflejar de manera precisa y exacta el contenido y su función, no debe usar tildes y se utiliza la nomenclatura Pascal Case.

Prefijos a utilizar:

En caso de la existencia de un elemento de la base de datos que no se encuentra en este listado se debe utilizar las reglas de generación de prefijos, antes descrita.

* + 1. **Llaves o Clave**

Los índices se nombran considerando la tabla a la que están relacionados y el propósito del índice.

• Las claves primarias utilizan el prefijo “PK”.

• Las claves foráneas utilizan el prefijo “FK”.

* + 1. **Campos**

Cada nombre de campo debe ser único dentro de su tabla correspondiente y utiliza nomenclatura Pascal Case, deben venir precedido del prefijo **PK o FK** seguido por el carácter **(\_)** guion bajo cuando sean llaves primarias o foráneas.

No se deben utilizar palabras reservadas como nombres de los campos.

Ejemplo: PK\_CedFuncionario, FK\_CodCiudad.

* 1. **Nomenclatura de programación**

**Pascal Case:** Es un estilo de escritura que se aplica a frases o palabras compuestas y se asemejan a las jorobas de un camello. La letra inicial es mayúscula. Ejemplo: SaldoInicial.

**3.3.1 Elementos de programas**

Nomenclatura Pascal Case**.**

El nombre: debe reflejar de manera precisa y exacta el contenido y su función, no debe usar tildes.

| TIPO | PREFIJO | Estructura | EJEMPLO |
| --- | --- | --- | --- |
| Formulario | Fm\_ | Fm\_NombreFormulario | Fr\_Funcionarios |
| Reportes | Rp\_ | Rp\_NombreReporte | Rp\_Vacaciones |
| WebServices | Ws\_ | Ws\_NombreWebService | Ws\_Pila |
| Proyecto | Pr\_ | Pr\_NombreProyecto | Pr\_Personal |

**3.3.2. Definición nombre de Variables**

Nomenclatura Pascal Case**.**

Comience el nombre de la variable con una letra.

Debe describir claramente el propósito de la variable.

Utilizar el prefijo que indique el tipo de dato.

|  |  |  |  |
| --- | --- | --- | --- |
| TIPO DE DATO | PREFIJO | Estructura | EJEMPLO |
| Integer (entero) | In | InNombreVariable | InTelefono |
| doble | Db | DbNombreVariable | DbValor |
| Long (entero largo) | Ln | LnNombreVariable | LnTotal |
| Float | Fl | FlNombreVariable | FlSubtotal |
| Decimal | Dc | DcNombreVariable | DcPago |
| Char | Ch | ChNombreVariable | ChDato |
| Varchar | Vr | VrNombreVariable | VrNombre |
| String | St | StNombreVariable | StNombreFun |
| boolean | Bl | BlNombreVariable | BlRespuesta |
| Byte | By | ByNombreVariable | ByByte |
| **Date time** | Dt | DtNombreVariable | DtFechaIncial |

**3.3.3. Definición de Procedimientos**

Nomenclatura: Pascal Case**.**

Nombre: Describe la acción

**3.3.4. Definición de Funciones**

Nomenclatura: Pascal Case**.**

Nombre: Debe ser tan largos como sea necesario para describir su funcionalidad, la Primera letra del nombre en Mayúscula.

Usar el sufijo: Tipo de dato que retorna, sí la función no retorna nada no se usará sufijo (El sufijo utiliza la misma sintaxis del prefijo).

|  |  |  |
| --- | --- | --- |
| TIPO | PREFIJO | EJEMPLO |
| Procedimiento | Pr | *PrCalcularExcedente()* |
| Funciones | Fn | *FnCalcularPorcentajeDb, fnEnviarEmail* |
| Clases | Cl | ClCuenta |

**3.3.5. Definición de Objetos**

| Tipo | Prefijo | Ejemplo |
| --- | --- | --- |
| Button | Bt | BtNombre |
| Calendar | Cn | CnNombre |
| CheckBox | Ch | ChNombre |
| ComboBox | Cb | CbNombre |
| DataGrid | Dg | DgNombre |
| GroupBox | Gb | GbNombre |
| ImageList | Il | IlNombre (il) |
| Label | Lb | LaNombre |
| LinkLabel | Ll | LlNombre |
| ListBox | Lb | LbNombre |
| ListView | Lv | LvNombre |
| MainMenu | Mm | MmNombre |
| PictureBox | Pb | PbNombre |
| ProgressBar | Pr | PrNombre |
| RadioButton | Rb | RbNombre |
| RichTextBox | Rt | RtNombre |
| StatusBar | Sb | SbNombre |
| TabControl | Tc | TcNombre |
| TextBox | Tb | TbNombre |

En caso de la existencia de un objeto que no se encuentra en el listado se debe utilizar las reglas de generación de prefijos descrita inicialmente, con la siguiente excepción:

* Cuando el nombre está compuesto por dos o más palabras unidas se debe seleccionar la primera letra de cada palabra, por ejemplo: **R**ich**T**extBox - Prefijo: Rt

**4. REQUERIMIENTOS DE SEGURIDAD DE LA INFORMACIÓN**

| **REQUERIMIENTO SEGURIDAD DE LA INFORMACIÓN** |
| --- |
| **Autenticación y gestión de contraseñas.** |
| Todos los controles de autenticación se deben aplicar en un sistema de confianza (LDAP). |
| El ingreso al sistema de información debe permitir autenticación de usuario, gestión de contraseña y control de sesiones seguras, cumpliendo con los requisitos aplicados por la Contraloría de Bogotá en cuanto a complejidad y longitud de contraseñas. |
| Debe forzar desactivación de la cuenta o sesión después de un número establecido de intentos de acceso no válidos. |
| Debe permitir la gestión de seguridad de usuarios, roles y perfiles, permitiendo asociarlo según los roles y responsabilidades del usuario. |
| No debe permitir conexiones simultaneas con el mismo ID de usuario. |
| **Control de acceso.** |
| La asignación de permisos o denegación de acceso al sistema de información para los usuarios, solo podrán ser aplicados por el administrador del sistema. |
| Un usuario puede estar asociado a uno o más roles, de tal manera que los menús de navegación del sistema se muestren o desplieguen dependiendo de las acciones asociadas a cada rol de usuario, la autenticación del usuario permite el acceso a roles Y/o perfiles asignados otorgando únicamente las acciones autorizadas a realizar. |
| Requiere del uso de herramientas criptográficas como dispositivos biométricos, tarjetas inteligentes, certificados electrónicos, toquen de seguridad, entre otros. |
| Debe generar registros de auditoria de ingreso a usuarios y acciones ejecutadas en el sistema de información |
| **Manejo de Log y Eventos** |
| Registrar todos los eventos de alteración aparentes, incluyendo cambios inesperados en el estado de los datos y del sistema de información. |
| El sistema de información debe estar sincronizado con la Hora Legal Colombiana del Instituto Nacional de Metrología de la Superintendencia de Industria y Comercio, el ingreso de los usuarios y las transacciones realizadas deben quedar registradas con esta fecha y hora. |
| **Seguridad de comunicación:** |
| Implementar mecanismos de cifrado para la transmisión de toda la información clasificada como reservada, privada, semiprivada, sensible, confidencial. |
| Debe cumplir con controles de seguridad de conexión segura a través de redes públicas y privadas, garantizando la confidencialidad, integridad y disponibilidad de la información y acceso a ella. |

**5. PRÁCTICAS DE SEGURIDAD [[1]](#footnote-1)**

|  |
| --- |
| **PRÁCTICAS DE SEGURIDAD A TENER ENCUENTA EN EL DESARROLLO DE SISTEMAS DE INFORMACIÒN** |
| **Revisión de diseño** |
| * **Identificar posibles ataques de software:** Por medio de diagramas, compruebe el modelo para verificar una consistencia a nivel de diseño para ver cómo se aseguran las interfaces con acceso similar, cualquier ruptura en la consistencia puede ser anotada como un fallo en la evaluación. |
| * **Analizar el diseño contra requisitos de seguridad conocidos:** Verificar que cada requerimiento de seguridad conocido haya sido solucionado en el diseño del sistema. |
| **Revisión de código** |
| * **Crear listas de verificación para la revisión de los requisitos de seguridad conocidos** basándose en el punto 3. REQUERIMIENTOS DE SEGURIDAD DE LA INFORMACIÓN. |
| * **Realizar revisiones en código de puntos de alto riesgo:** Revisar módulos de alto riesgo por vulnerabilidades comunes, las funcionalidades de alto riesgo incluyen módulos de autenticación, puntos de reforzamiento de controles de acceso, esquemas de manejo de sesión, interfaces externas, validadores de entradas, y analizadores de datos, etc. |
| * **Establecer puntos de control para la liberación de las revisiones de código:** Para establecer lineamientos de seguridad a nivel de código a todos los proyectos de software, un punto en particular en el ciclo de desarrollo de software puede ser establecido como un punto de control donde el estándar mínimo para los resultados de las revisiones de código debe ser cumplidos para poder hacer la liberación |
| **Pruebas de seguridad** |
| * **Deducir casos de prueba desde los requisitos de seguridad conocidos**: De los requisitos de seguridad conocidos para un proyecto, identificar un conjunto de casos de prueba para comprobar la correcta funcionalidad del software |

**6. PRÁCTICAS DE GESTION DOCUMENTAL[[2]](#footnote-2)**

|  |
| --- |
| **REQUERIMIENTOS EN MATERIA DE GESTION DOCUMENTAL QUE SE DEBEN TENER EN CUENTA EN EL DESARROLLO DE SISTEMAS DE INFORMACIÓN** |
| Los sistemas de información que requieran firma electrónica en los documentos o registros del aplicativo deberán tener en cuenta las siguientes consideraciones regidas por normatividad y requisitos de ley establecidos en el ordenamiento jurídico colombiano:   * Garantizar la intención, aprobación del contenido del mensaje de datos (documento) y relación con el mensaje de datos a través de mecanismos técnicos como claves, contraseñas, códigos OTP, mensajes enviados a correo electrónicos, audios con voz del o de los firmantes, mecanismos biométricos.   Se sugiere la siguiente leyenda:  “Como responsable del diligenciamiento y certificación del documento reconozco y declaro haber leído, comprendido y aceptado el contenido íntegro del mismo y la información diligenciada, las implicaciones de la información emitida, así mismo, al presionar el botón "Aceptar" expreso la conformidad inequívoca e irrevocable del contenido y las observaciones realizadas.  La información contenida en el documento generado por el aplicativo goza de plena validez de acuerdo al artículo 5 de la Ley 527 de 1999, por medio del cual se reconoce jurídicamente los efectos de los mensajes de datos.  Los firmantes y responsables de la firma aceptamos y reconocemos que el presente documento se firmará electrónicamente a través del método de firma electrónica establecido mediante usuario y contraseña (la cual se tratara como un dato confidencial, personal e intransferible, siendo de su responsabilidad exclusiva el manejo, administración y custodia), el cual cumple con los criterios de confiabilidad y apropiabilidad y se garantizan los atributos de autenticidad e integridad necesarios conforme a lo establecido en la Ley 527 de 1999 y el Decreto 2364 de 2012, compilado en el Decreto Único Reglamentario del sector comercio 1074 de 2015 o cualquier norma que lo complemente o sustituya.  El usuario firmante reconoce de forma expresa e irrevocable que cualquier operación o consulta realizada bajo su clave secreta equivale a: su firma manuscrita, corresponde exclusivamente a éste y ha manifestado expresamente el consentimiento en la misma.  El proceso de firma electrónica implementado en el aplicativo almacenará la fecha, hora, usuario institucional que firma y los documentos firmados, generando un número identificador de la operación de firma (Hash) y un código QR que será único e intransferible para cada documento.  El documento electrónico generado será archivado y conservado en la base de datos del aplicativo, y será accesible a las partes de acuerdo con los mecanismos de verificación y autenticidad implementadas sobre el documento.”   * Acuerdo de voluntades: Las aplicaciones que vayan a involucrar algún mecanismo de firma, deben obtener dicha aprobación a través de una ventana de información, de un documento PDF o almacenar mediante algún objeto del sistema que los firmantes aceptan y deciden que dicho mecanismo se acepte como válido para los mensajes que se firmen. * Integridad, originalidad: Capacidad de que el objeto se mantenga intacto con las propiedades con las que se creó y no ha sido modificado. Algunas herramientas tecnológicas aplicables son: hash (ADN del documento o mensaje de datos). Estampa de tiempo (En lo posible obtener la fecha y hora legal de Colombia de acuerdo al instrumento del Instituto Nacional de Metrología e insertarla como metadato del documento o en un mecanismo alterno), uso de criptografía, implementación de CSV (Código seguro de verificación) como mecanismos para verificar la autenticidad del documento. * Capacidad de verificar y validar que un usuario es quien dice ser (identidad del firmante – integridad - autenticidad) – Equivalencia No repudio. Criterio a cumplir por medio de la implementación de los tres factores de autenticación (mínimo dos) en Colombia que pueden ser:   **Algo que sabemos:** Contraseña o código PIN, clave dinámica o códigos OTP, avatar, imágenes, firmas biométricas, claves criptográficas, marcas de agua digitales.  **Algo que poseemos:** Token, tarjetas, celular, correo electrónico.  **Algo que somos:** La huella dactilar, la voz, el ADN, el rostro, sonrisa, video.   * Seguridad o validez jurídica: por lo que puede ser susceptible de ser verificada, es única y el control es exclusivo de la persona que la usa y debe asociarse al hash, por lo que si cambia la firma para un documento original, todo el documento se invalida. Los metadatos almacenados posteriormente apoyaran el proceso de validez jurídica junto con los mecanismos de las características anteriormente mencionadas. * Confiabilidad: Característica que puede ser implementada por medio de un hash, Código QR, Contraseña o código PIN, clave dinámica, avatar, Token, tarjetas, celular, correo electrónico, datos biométricos, metadatos, CSV (Código seguro de verificación), mecanismo de consulta del documento. Aplicación de consulta del documento, marcas de agua digitales. Se sugiere implementar algún mecanismo de consulta que permita visualizar el documento, presentar el hash y proveer mecanismos para validarlo a partir de la consulta o el cargue de un documento existente. |
| Los mecanismos técnicos o tecnológicos seleccionados y la selección de implementación de firma electrónica o firma digital deberá obedecer en todo caso a criterios relacionados con la seguridad y privacidad de la información, la criticidad del documento y la exposición a riesgos tales como suplantación, alteración, indisponibilidad, el grado de seguridad y confiabilidad que proporciona el mecanismo seleccionado, la valoración e importancia que tiene el documento y el trámite en el cual se encuentra relacionado para ser expuesto a requisitos jurídicos, técnicos, comerciales y legales que por sí pueden ser demostrables y adecuados.  Triangulo del nivel de seguridad e una firma, frente al nivel de riesgo  Imagen tomada del Webinar Firmado Digital y Electrónico Colombia. |
| En todos los casos y para garantizar las características anteriormente mencionadas deberán almacenarse los metadatos correspondientes que circunscriben el entorno del mensaje de datos o la producción documental, por lo que, entre otras características de no poderlas embeber al documento, sería necesario almacenarlas en forma de log, en una tabla del sistema o algún mecanismo del sistema que dé cuenta con posterioridad de la siguiente información:   * Proceso o actuación en el que se produce el documento o mensaje. * Para cada firmante (en calidad de aprobador, revisor, proyector, redactor, autoridad de la entidad): nombres y apellidos completos, cédula, correo electrónico, cargo, dependencia, calidad de firmante, fecha y hora de la firma, mecanismo de firma empleado, nombre del documento, tipo de hash y valor, nombre del trámite, actividad, proceso o documento del SIG que se firma, dirección ip origen del firmante, ubicación del documento, TRD a la que pertenece, aplicación que se emplea para firmar. * Tener en consideración los metadatos que sean aplicables y que se encuentran definidas en el esquema de metadatos de la entidad en su última versión, que puede ser consultado en la Intranet de la Entidad dentro de los formatos del Proceso de Gestión Documental en el Instrumento Archivístico denominado Esquema de Metadatos documento PGD-16 y el estándar técnico de Preservación Digital PREMIS adoptado por la Entidad. |
| En cuanto al almacenamiento de los documentos, se recomienda que los documentos que se encuentren clasificados en la TRD en alguna serie o subserie, la generación se realice en los formatos definidos para la preservación digital contemplados en el Plan de Preservación Digital que hace parte del Sistema Integrado de Conservación entre ellos, para documentos de texto PDF/A1, XML. La información referente a los formatos puede ser consultados en la estrategia 2 de dicho plan, en donde se referencian los formatos para cada tipo de contenido. |
| En relación al acceso y permisos de los documentos deberá observarse las consideraciones y restricciones establecidas en los instrumentos de gestión de la información tales como los registros de los activos de información, esquema de publicación de información y el índice de información clasificada y reservada y la tabla de control de acceso que se encuentren vigentes a efectos de definir, el tratamiento, los roles de acceso a registros y documentos en los sistemas de información. |
| La conformación de expedientes, el nombrado, la ubicación, la disposición de documentos electrónicos generados por el sistema de información deberán atender los lineamientos y consideraciones expuestas en el PGD-05 Procedimiento para la producción, organización y conservación de documentos y los instructivos que lo componen. |
| La ubicación física de los documentos deberán contemplar la interrelación e interoperabilidad con el sistema de comunicaciones oficiales de la entidad, el SGDEA y los demás sistemas de la entidad con los que se comunique, en todo caso deberá responder bajo la estructura definida para el proceso que apoye y lo contemplado en el Cuadro de Clasificación Documental, la Tabla de Retención Documental al cual pertenezcan y el repositorio de los documentos Datacontrabog o el que se disponga al momento de la generación de documentos electrónicos de archivo propendiendo por la unidad, la originalidad y el orden de los documentos, entre otros principios archivísticos. |
| Inicialmente y de acuerdo a lo implementado para algunos sistemas de información se ha seleccionado la función de hash SHA-256 a efectos de garantizar la integridad de los documentos electrónicos; sin embargo, en los desarrollos se podrá contemplar mecanismos criptográficos de mayor fortaleza o que se integren de forma más completa o natural al entorno de desarrollo en función de los recursos disponibles. |
| En los procesos de desarrollo que involucren firmas digitales en los cuales los documentos electrónicos estén contemplados como Documentos de Archivos contemplados en la Tabla de Retención Documental con tiempo de retención mayor a 10 años se deberá solicitar a la entidad de certificación con la cual se ejecute la compra de los certificados digitales la siguiente consideración:  La firma digital deberá soportar los siguientes formatos: XADES, CADES y PADES y sus respectivos formatos longevos, en este caso PAdES-LTV de los certificados digitales expedidos en los documentos electrónicos. |

1. Basado en el Software Assurance Maturity Model – Una guía para integrar en el desarrollo de software Versión - 1.0. OWASP THE Open Web Application Security Project. [↑](#footnote-ref-1)
2. Basado en el Software Assurance Maturity Model – Una guía para integrar en el desarrollo de software Versión - 1.0. OWASP THE Open Web Application Security Project. [↑](#footnote-ref-2)