



AGENCIA PARA LA REINCORPORACIÓN Y NORMALIZACIÓN (ARN)

GUÍA DE DESARROLLO DE SOFTWARE

BOGOTÁ D.C. MAYO DE 2021

TABLA DE CONTENIDO

1.	OBJETIVOS.....	3
2.	ALCANCE	3
3.	DEFINICIONES.....	3
4.	FASES DEL DESARROLLO DE SOFTWARE	4
4.1.	FASES DE DESARROLLO Y METODOLOGÍA DE DESARROLLO.....	4
5.	ESTÁNDARES DE PROGRAMACIÓN Y PRÁCTICAS RECOMENDADAS	9
5.1	ESTÁNDAR DE SEGURIDAD DE APLICACIONES WEB OWASP.....	9
5.2	SOLID.....	10
5.3	ESPECIFICACIONES DE CODIFICACIÓN PARA APLICACIONES WEB	12
5.4	ESTÁNDAR DE NOMBRAMIENTO Y CODIFICACIÓN DE BASE DE DATOS.....	14
5.5	NOMBRAMIENTO DE LOS REPORTES	14
5.6	CONTROL DE VERSIONES	14
6.	ARQUITECTURA DEL SISTEMA DE INFORMACION MISIONAL	15
6.1	SIRR	15
6.2	ARPA.....	15
6.3	SARA.....	15
6.4	AMBIENTES DE DESARROLLO	15
7	PLAN DE CALIDAD DEL SOFTWARE	17
7.1	LOS PRINCIPALES OBJETIVOS DEL ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE	17
7.2	DEFINICIONES.....	17
7.3	MODELO DE CALIDAD.....	17
7.4	CONSIDERACIONES DE LA ETAPA DE PRUEBAS.....	20
7.5	AMBIENTE DE PRUEBAS Y CAPACITACIÓN	20
7.6	MANEJO DE DATOS EN AMBIENTE DE PRUEBAS	22
7.7	USUARIOS DE PRUEBA	22
8	HERRAMIENTAS PARA GESTIÓN DE PROYECTOS DE DESARROLLO.....	23
8.1	HERRAMIENTAS DE GESTIÓN.....	23
8.2	INFRAESTRUCTURA DE LAS HERRAMIENTAS DE GESTIÓN	23
9	BIBLIOGRAFIA	24

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

1. OBJETIVOS

Los principales objetivos del presente documento son:

- Mejorar la calidad del software que se desarrolla en la Agencia, documentando recomendaciones, estándares y mejores prácticas aplicables al proceso.
- Servir como documento guía para conducir el proceso de desarrollo de software.

2. ALCANCE

Las recomendaciones aplican a los proyectos de software que ejecute la Agencia, y que contengan componentes de desarrollo de software.

3. DEFINICIONES

ARN: Agencia para la Reincorporación y Normalización.

ARPA: Aplicación de Apoyo a la Reincorporación y Procesos de Atención.

NTC 5854: La Norma Técnica Colombiana (NTC) 5854 fue redactada y diseñada para establecer los requisitos de accesibilidad que deben cumplir los portales web del Estado Colombiano. Estos requisitos se agrupan en tres niveles de conformidad: A, AA, y AAA. Esta norma fue desarrollada empleando como documento de referencia “Las Pautas de Accesibilidad para el Contenido web (WCAG) 2.0 del 11 de diciembre de 2008”. El objetivo de esta norma es promover las buenas prácticas que permitan que el contenido de la web sea accesible a todos los usuarios independiente de su condición, para que puedan percibir, entender, navegar, interactuar y contribuir con los sitios web

PPR: Persona en proceso de Reintegración o Reincorporación.

RFC: La Solicitud de Cambio (Request for Change, RFC) es una requisición formal de Cambio en espera de ser implementada. Incluye detalles del Cambio propuesto, y puede estar en formato electrónico o en papel. Las siglas (en inglés) RFC a menudo se usan equivocadamente para referirse al Registro de Cambio o al Cambio mismo.

SARA: Sistema de Apoyo a la Reintegración.

SIRR: Sistema de Información para la Reintegración y la Reincorporación, es el sistema central de la Agencia para la Reincorporación y Normalización, donde se encuentra toda la información relacionada con las Personas en Proceso de Reintegración y su proceso de reintegración o reincorporación.

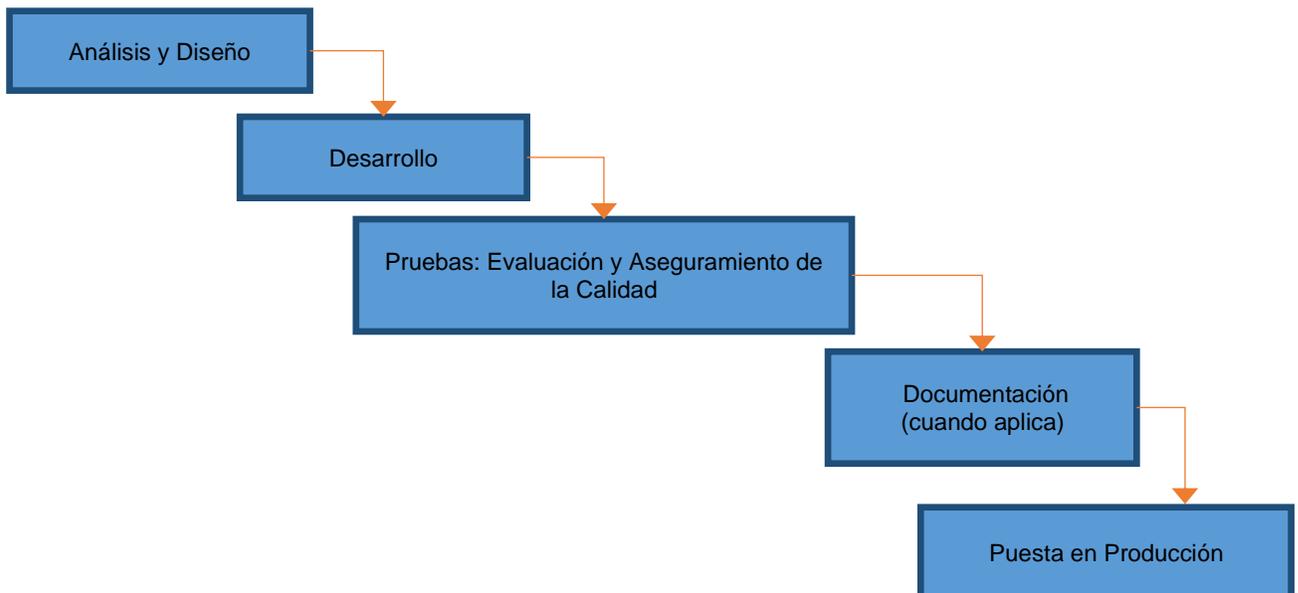
SOLID: Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion) es un acrónimo mnemónico introducido por Robert C. Martin¹² a comienzos de la década del 2003 que representa cinco principios básicos de la programación orientada a objetos y el diseño.

4. FASES DEL DESARROLLO DE SOFTWARE

4.1. FASES DE DESARROLLO Y METODOLOGÍA DE DESARROLLO

Dada las condiciones de la Agencia y su necesidad de mantener la seguridad de la información, se determina como metodología de desarrollo RUP (Proceso de Desarrollo Unificado) por ser un proceso de desarrollo de software formal que junto con el Lenguaje Unificado de Modelado UML, constituyen la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas de información. No obstante, lo anterior, se incorporan a la metodología elementos y prácticas de SCRUM como la gestión de backlogs, tareas, iteraciones y reuniones semanales de equipo.

En la oficina de Tecnologías de la Información – Grupo de Sistemas de Información, se establece que los Proyecto de Desarrollo de Software para nuevas aplicaciones o funcionalidades y de mantenimiento de aplicaciones y/o funcionalidades existentes, deberán observar las siguientes fases del ciclo de vida de desarrollo de software, alineado con el TI-P-01 Procedimiento Atención A Requerimientos De Sistemas De Información:



Fuente: Propia de la OTI

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

4.1.1. Análisis y Diseño

En esta etapa los analistas del GSI se relacionan con los usuarios finales del sistema, se realizan mesas de trabajo a fin de definir los requisitos a cumplir por el proyecto o funcionalidad. Una vez que se recaban todos los requisitos, los analistas vuelven a reunirse con los usuarios para repasarlos en una fase de verificación, realizando la estimación de esfuerzo y posible solución, luego en caso de requerirse se planteará el diseño de la funcionalidad o aplicación y se documentara en los casos de usos respectivos que deberán ser aprobados por los usuarios finales y contar con el aval del Administrador Funcional.

Dentro de esta etapa se deben contemplar las siguientes recomendaciones de seguridad en la información:

- a) Nivel de confianza requerido con relación a la autenticación de usuario y los procesos de autenticación de usuario del negocio, al igual que para usuarios privilegiados o técnicos:

Para el SIRR y ARPA la autenticación del usuario se realiza a través del directorio activo por lo tanto se debe tener en cuenta el definido para creación de usuarios y permisos en el formato TI-F-01, Solicitud de Usuario y/o Recursos Tecnológicos.

En aplicaciones WEB como lo es SARA se definen las siguientes recomendaciones:

Autenticación: Debe poderse identificar el generador de la información, por lo tanto, cada usuario deberá autenticarse en el sistema mediante un formulario de inicio de sesión donde se soliciten los datos Usuario y Contraseña. Además, el sistema debe contar con:

Política de contraseñas:

1. Caducidad: Las contraseñas deben tener un plazo de vencimiento de 90 días para cambiarla.
2. Longitud mínima: El sistema debe exigir un número mínimo de 6 caracteres para la contraseña.
3. Complejidad: El sistema debe exigir a los usuarios la combinación de caracteres alfanuméricos y símbolos especiales en la contraseña.
4. Historial de contraseñas: El sistema debe almacenar el hash de cada una de las contraseñas de modo los usuarios deben reestablecerla a una

nueva y exclusiva. El historial guardará un máximo de 5 registros por usuario.

5. Número de intentos de inicio de sesión no válidos: El sistema permitirá hasta 3 intentos de inicio de sesión no válidos; si este valor es superado, el usuario deberá esperar un tiempo de 5 minutos para volver a intentarlo.
6. Mecanismo de restablecimiento de la contraseña: El sistema deberá tener un mecanismo de restablecimiento de la contraseña mediante el envío de las nuevas credenciales al correo electrónico.

b) Definir las necesidades de protección de activos involucrados, en particular acerca de disponibilidad, confidencialidad, integridad;

El sistema deberá contar con medidas preventivas y reactivas para garantizar la confidencialidad y la integridad de la información del sistema. El sistema debe contar con los siguientes mecanismos:

- **Confidencialidad:** El sistema deberá garantizar que el acceso a la información tenga varios niveles de confidencialidad, por lo tanto, el sistema debe contar con:
 - **Transporte seguro:** La información debe viajar por un canal seguro para el transporte de los datos a través de internet y así asegurar la confidencialidad de la información, por lo que deberá ser capaz de operar en HTTPs.
 - **Autorización:** El sistema deberá tener políticas de acceso a la información mediante roles para impedir que los usuarios tengan acceso a información o procesos no autorizados.
- **Integridad:** El sistema debe tener mecanismos que impidan realizar modificaciones no autorizadas a la información para evitar que la información pierda exactitud. Para esto, el sistema debe contar con:
 - **Mecanismos de validación de la información:** Durante las transacciones de datos en los procesos del sistema, se debe realizar validaciones del lado del cliente y del lado del servidor para garantizar la exactitud de la información.
- **Disponibilidad:** El sistema debe garantizar la disponibilidad de la información a las personas que cuenten con los privilegios de acceso.

- c) Definir los requisitos obtenidos de los procesos del negocio, tales como los requisitos de ingreso y seguimiento, y de no repudio;

Dentro de los documentos de Arquitectura de nuevos proyectos o aplicaciones se deberá contemplar un ítem donde se definan los requisitos de ingreso, seguimiento y de no repudio.

- d) Establecer los requisitos exigidos por otros controles de seguridad, (interfaces con el ingreso o seguimiento, o los sistemas de detección de fuga de datos).

En los documentos de caso de uso se debe documentar en el apartado 3. ACTORES Y PERMISOS los roles, permisos y nivel de acceso a la información deberán tener los usuarios para acceder a la aplicación.

- e) Una vez finalizada la etapa de requerimientos los casos de uso, o arquitectura deben ser cargados en el repositorio <http://otidocs.acr.int/sites/OTI>, adicionalmente los casos de uso deben ser incluidos en el inventario de casos de uso.
- f) Los casos deben ser nombrados según la siguiente nomenclatura: CU-SISTEMA-MODULO-XXXX

CU	Que corresponde a la sigla caso de uso
Sistema	Corresponde a una sigla que corresponde al sistema o ambiente donde se desarrollara el caso de uso (SIRR SARA, INTRANET)
Módulo	Nombre que referencia al módulo a desarrollar (Jurídica, Plan de Trabajo, etc.)
Consecutivo	Número Consecutivo compuesto por 4 dígitos de los casos de uso por módulo

Ejemplo: CU-SIR-PLANDETRABAJO-0001

- g) El requerimiento debe ser documentado en el formato de caso de uso "TI-F-03 Formato Caso de Uso", formato publicado en SIGER (https://klein.reincorporacion.gov.co/SIGER/modulodocumental/ModuloXML/formatoframes.asp?Vista=1&Nombre_Formato=&intTipoDoc=82&modulo=D0&IdDoc=51249&)

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

Una vez documentado el caso de uso, este es validado y aprobado (mediante firma mecánica o aceptación electrónica) por el usuario solicitante, el administrador funcional del SIRR, el jefe o Coordinador del Área Solicitante y el Coordinador del Grupo de Sistemas de Información, asegurando la pertinencia y exactitud del requerimiento.

4.1.2. Desarrollo

Esta etapa se refiere al desarrollo del software según las definiciones realizadas en los documentos de casos de uso aprobados y firmados, durante en la etapa de análisis y diseño.

En la construcción del software se deberán tener en cuenta las recomendaciones que se describen en el ítem **5 Estándares de Programación y prácticas recomendadas** del presente documento.

Para el caso del desarrollo de Reportes a través de la herramienta de Reporting Services se recomienda nombrar los informes con el código del caso de uso con el que se especificó.

Una vez finalizado el desarrollo se debe realizar el empaquetamiento y la preparación del despliegue en ambiente de pruebas, con lo cual el desarrollador deberá solicitar caso a la mesa de ayuda con el fin de asegurar:

- a) Realizar la preparación del ambiente de pruebas:
 - Se debe realizar la instalación, publicación o despliegue del software desarrollado o modificado en los servidores de aplicación.
 - Se debe verificar si se requiere preparar datos del ambiente productivo al ambiente de pruebas.
- b) Preparación de permisos y estructuras necesarias de BD cuando se requiere:
 - Scripts de BD
 - Indicaciones de Roles y menús.

4.1.3. Pruebas y Aseguramiento de la Calidad

Dentro de esta etapa se contempla dos eventos:

- Revisión de los casos de uso aprobados que harán parte de la prueba.
- Prueba de los desarrollos realizados, tratados en detalle en el capítulo 7. **PLAN DE CALIDAD DEL SOFTWARE** del presente documento.

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

4.1.4. Documentación

En esta fase se crean los instructivos de usuario de las funcionalidades (cuando se requiere) o aplicaciones desarrolladas, estos documentos deben ser publicados en el portal SIRR (<http://portalsir:85/>).

De requerirse capacitación complementaria a la documentación, esta deberá ser coordinada por el área usuaria, y podrá contar con el apoyo del equipo técnico participó del proyecto o desarrollo.

4.1.5. Despliegue y Puesta en Producción

En esta fase se crea la solución de la nueva funcionalidad o aplicación y se solicita la publicación en ambiente productivo al Grupo de Infraestructura y Soporte a través de un registro en la herramienta de mesa de ayuda en el marco de la gestión de cambios, se valida el despliegue correcto y se informa a los usuarios.

5. ESTÁNDARES DE PROGRAMACIÓN Y PRÁCTICAS RECOMENDADAS

En este apartado se definen los principios para la construcción de sistemas seguros los cuales deben ser aplicarlos a cualquier actividad de implementación de sistemas de información (NIST SP 800-53 Rev. 4 SA-3, SA-4, SA-8, SA-10, SA-11, SA-12, SA-15, SA-17, PL-8), así como practicas recomendadas.

5.1 ESTÁNDAR DE SEGURIDAD DE APLICACIONES WEB OWASP

Se propone que durante el diseño y construcción de las aplicaciones se tengan en cuenta las recomendaciones dadas en la guía para el desarrollo de aplicaciones seguras y se tenga en cuenta el listado de los diez riesgos de seguridad más importantes sobre los cuales se recomienda a las entidades orientar la prevención y protección en el código que contienen las aplicaciones desarrolladas.

- Detalle acerca del proyecto OWASP y el estándar, puede consultarse en el sitio web oficial:

https://www.owasp.org/index.php/Sobre_OWASP

- El documento “Guía para Construir Aplicaciones y Servicios Web Seguros” liberado, con la versión más reciente puede consultarse en:

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

https://www.owasp.org/images/b/b2/OWASP_Development_Guide_2.0.1_Spanish.pdf

- El documento borrador versión 3 de la Guía para Construir Aplicaciones y Servicios Web Seguros” puede consultarse en:
<https://github.com/OWASP/DevGuide>

- El listado actualizado, sus históricos y candidatos para liberación pueden consultarse aquí:
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

5.2 SOLID

El grupo de sistemas de Información se ha propuesto implementar estos 5 principios con el propósito de que los diseños de software sean más comprensibles, flexibles y mantenibles.

Principios de SOLID

A. Single Responsibility Principle / Principio de Responsabilidad Única

Lo que nos pide este principio es que cada clase debe tener una única responsabilidad, por lo que si estamos programando una clase que se ocupa de diferentes cosas es conveniente partirla en 2 o más clases. Con esto se consigue mayor mantenibilidad y se clarifica el código, haciéndolo más mantenible.

Muchas veces estamos tentados de programar demasiadas cosas en una misma clase, pero esto aumenta el acoplamiento de dos funcionalidades que pueden cambiar por razones diferentes o en momentos distintos.

B. Open-Closed Principle / Principio Abierto-Cerrado

La idea de este principio es que las clases están abiertas a la extensión, pero cerradas a la modificación, lo que significa que, ante peticiones de cambio en nuestro programa, hay que ser capaces de añadir funcionalidad sin modificar la existente (siempre que sea posible).

Por lo general, en un mal diseño, modificar una funcionalidad durante el ciclo de vida, suele conllevar una cadena de cambios en módulos dependientes, y este efecto puede propagarse en cascada si el código está muy acoplado.

La forma más común de seguir este principio es usar interfaces o clases abstractas de las que dependen implementaciones concretas. De esta forma puede cambiarse la implementación de la clase concreta manteniéndose la interfaz intacta.

C. Liskov Sustitution Principle /Principio Sustitución de Liskov

El principio de sustitución de Liskov nos dice que, si en alguna parte de nuestro código estamos usando una clase, y esta clase es extendida, tenemos que poder utilizar cualquiera de las clases hijas y que el programa siga siendo válido. Esto nos obliga a asegurarnos de que cuando extendemos una clase no estamos alterando el comportamiento de la clase padre.

Este principio viene a desmentir la idea preconcebida de que las clases son una forma directa de modelar la realidad.

D. Interface Segregation Principle / Principio de segregación de interfaz

El principio de segregación de interfaces dice que ninguna clase debería depender de métodos que no usa. Por tanto, cuando creamos interfaces que definan comportamientos, es importante estar seguros de que todas las clases que implementen esas interfaces vayan a necesitar y ser capaces de agregar comportamientos a todos los métodos. En caso contrario, es mejor tener varias interfaces más pequeñas.

Las interfaces nos ayudan a desacoplar módulos entre sí. Esto es así porque si tenemos una interfaz que explica el comportamiento que el módulo espera para comunicarse con otros módulos, nosotros siempre podremos crear una clase que lo implemente de modo que cumpla las condiciones. El módulo que describe la interfaz no tiene que saber nada sobre nuestro código y, sin embargo, nosotros podemos trabajar con él sin problemas.

E. Dependency inversion principle / Principio de inversión de dependencias

Este principio enuncia lo siguiente:

- Las clases de alto nivel no deberían depender de las clases de bajo nivel. Ambas deberían depender de las abstracciones.
- Las abstracciones no deberían depender de los detalles. Los detalles deberían depender de las abstracciones.

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

5.3 ESPECIFICACIONES DE CODIFICACIÓN PARA APLICACIONES WEB

- HTML 5.2: <https://www.w3.org/TR/html52/>.
- CSS 3.0: <https://www.w3.org/Style/CSS/specs.en.html>
- ECMASCRIPT 5.0: <https://www.ecma-international.org/ecma-262/5.1/>
- WCAG 2.1: <https://www.w3.org/TR/WCAG21/>
- Norma técnica nacional de accesibilidad web NTC 5854:
<http://ntc5854.org/>

5.3.1 Aplicaciones desarrolladas para la Plataforma universal de Windows:

Las aplicaciones desarrolladas para la plataforma Universal e Windows como lo es la aplicación.

ARPA deben cumplir los siguientes lineamientos documentados por Microsoft en su la página:

<https://docs.microsoft.com/es-es/windows/uwp/design/accessibility/accessibility-overview>

ESTÁNDARES DE CODIFICACIÓN DE LENGUAJES DE PROGRAMACIÓN

Teniendo en cuenta que las tecnologías utilizadas en la ARN son herramientas de Microsoft, se han propuesto la implementación de la guía de Convenciones de código de C# propuesta por Microsoft en su página:

<https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/inside-a-program/coding-conventions> para los desarrollos en SIRR, ARPA y SARA.

Igualmente se tendrá en cuenta la siguiente documentación para nombrar los objetos, clases y demás componentes del código:

<https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/capitalization-conventions>

Algunos ejemplos de la notación a utilizar son:



IDENTIFICADOR	CASE	EJEMPLO	DESCRIPCIÓN
Clase	Pascal	Actividad, Persona, Programación	Utilizar nombres que identifiquen la entidad desde el punto de vista del desarrollador; los nombres deberán reflejar escenarios de uso. No usar guion bajo en los nombres de clases.
Tipo de enumeración	Pascal	TipoDocumento	Usar sustantivos para los nombres, no utilizar prefijos.
Valores de enumeración	Pascal	Cédula, TarjetaIdentidad, RegistroCivil	Usar sustantivos para los nombres, no utilizar prefijos.
Evento	Pascal	OnButtonEnrolar_Click	No utilizar abreviaturas, se debe colocar la palabra On, seguida del nombre del control, la descripción del evento y finalmente la acción del evento, separada por _.
Excepción	Pascal	FileNotFoundException	Termina el nombre con la palabra "Exception".
Constantes	Pascal	NOMBRE_REPOSITORIO	El nombre debe ser en letras mayúsculas y usar guion _ cuando el nombre está conformado por más de una palabra.
Interfaz	Pascal	I Actividad	Nombrar interfaces con sustantivos o adjetivos que describan el funcionamiento y utilizar como prefijo la letra I.
Método	Pascal	Consultar Actividades, Reprogramar Actividad	Los nombres de los métodos deben ser verbos para describir la acción del método y facilitar a los desarrolladores la comprensión de lo que hace el método
Espacio de nombres	Pascal	ACR.SIPAS.Model	<Compañía>.<Proyecto>.<Funcionalidad>
Parámetro	Camel	nombres, apellidos, fecha Actividad	Utilizar nombres descriptivos y significativos.
Propiedad	Pascal	Edad, EsCabezaFamilia	Asignar nombres a las propiedades mediante un sustantivo o un adjetivo. Denomine las propiedades booleanas con una frase afirmativa.

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

5.4 ESTÁNDAR DE NOMBRAMIENTO Y CODIFICACIÓN DE BASE DE DATOS

Para los diseños y desarrollos que impliquen creación de bases de datos, tablas, procedimientos almacenados, Triggers, etc. se deberá seguir los lineamientos consignados en el documento ESTANDAR DE NOMBRAMIENTO Y CODIFICACION DE BASE DE DATOS según lo defina el Grupo de Infraestructura y Soporte.

5.5 NOMBRAMIENTO DE LOS REPORTES

Los reportes desarrollados deben ser nombrados con el mismo código de caso de uso documentado para su desarrollo siguiendo la codificación descrita en el literal 4.1.1 f) de este documento.

5.6 CONTROL DE VERSIONES

El control de versiones del código fuente debe protegerse mediante el uso del repositorio de código fuente.

a) Librerías de fuentes de programas:

Todo código desarrollado o actualizado debe alojarse en el servidor <http://azuredevops.reincorporacion.gov.co:8080> a través de la herramienta de control de versiones de Azure DevOps en el proyecto OTI/ACR ubicándolo en el espacio correspondiente a la aplicación o proyecto al que pertenezca.

- b) Al realizar una entrada en el repositorio, el código fuente de los programas y las librerías de las fuentes de los programas se debe escribir una descripción corta del código y si es una actualización o control de cambio se debe escribir un comentario del cambio realizado, adicionalmente el código desarrollado se debe enlazar a la tarea de desarrollo asignada en el Azure DevOps.
- c) Solo los funcionarios y contratistas del Grupo de Sistemas de Información están autorizados para crear, consultar y actualizar nuevos proyectos y/o código fuente. El acceso a los proyectos se controla a través el panel de control del Azure DevOps donde se configuran los usuarios y/o grupos que podrán acceder a los proyectos.
- d) Todos los Plug-Ing y Stored Procedure después de desarrollados deben ser registrados en el archivo Inventario de Plug-Ins y SP publicado en la ruta:

 AARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

<http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/Inventario%20de%20Plug-Ins%20y%20Stored%20Procedure>

6. ARQUITECTURA DEL SISTEMA DE INFORMACION MISIONAL

6.1 SIRR

La documentación técnica del SIRR es la propuesta por Microsoft CRM Dynamics 2016

<https://www.microsoft.com/es-ES/dynamics/crm-customer-center/microsoft-dynamics-365-user-guides.aspx>

6.2 ARPA

La Arquitectura planteada para ARPA se encuentra documentada en el Documento de Diseño ACR_ModernizacionSIPAS_Diseno_V5 ubicado en la ruta:

http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/ARPA/Documentación%20Modernización%20SIPAS/ACR_ModernizacionSIPAS_Diseno_V5.docx

Y en el documento de arquitectura de ARPA después de la refactorización, ubicado en la ruta:

<http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/ARPA/Documentos%20de%20Dise%C3%B1o%20para%20Refactorizaci%C3%B3n%20ARPA%20V6/Documento%20Arquitectura%20ARPA.docx>

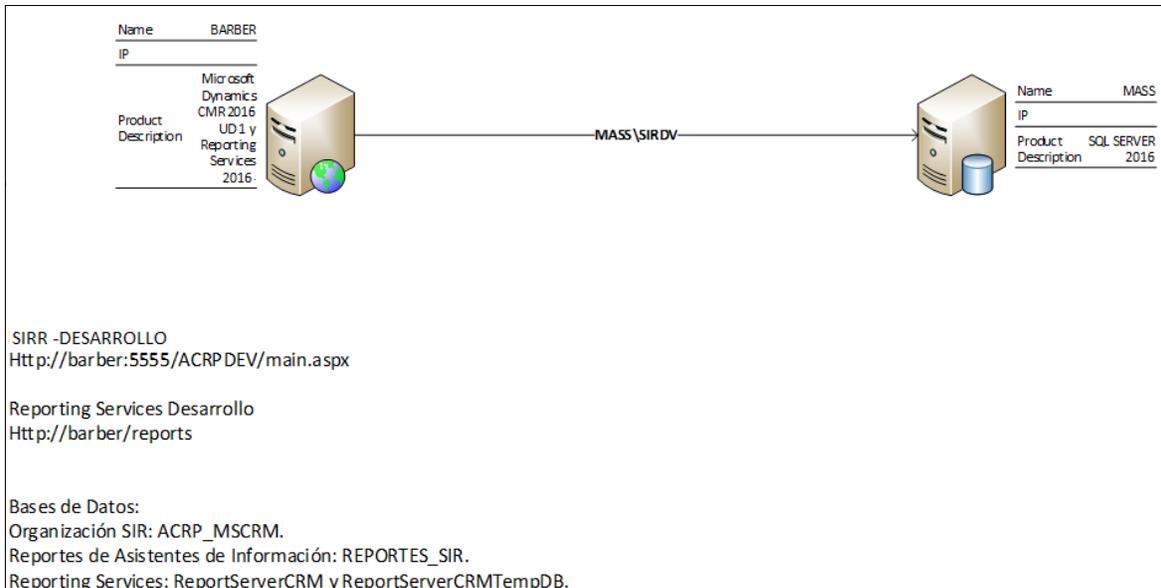
6.3 SARA

La Arquitectura planteada para SARA se encuentra documentada en el Documento Arquitectura SARA ubicado en la ruta:

<http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/SARA%20V2/Documento%20Arquitectura%20SARA.docx>

6.4 AMBIENTES DE DESARROLLO

6.4.1 SIRR



6.4.2 ARPA

En el documento de Arquitectura de ARPA se encuentra la descripción de los ambientes:

http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/ARPA/Documentación%20Modernización%20SIPAS/ACR_ModernizacionSIPAS_Disenio_V5.docx

<http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/ARPA/Documentos%20de%20Dise%C3%B1o%20para%20Refactorizaci%C3%B3n%20ARPA%20V6/Documento%20Arquitectura%20ARPA.docx>

6.4.3 SARA

En el documento de Arquitectura de SARA en el ítem 5. Vista de Despliegue se describen el ambiente de desarrollo:

<http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/SARA%20V2/Documento%20Arquitectura%20SARA.docx>

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

7 PLAN DE CALIDAD DEL SOFTWARE

7.1 LOS PRINCIPALES OBJETIVOS DEL ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

- Mejorar la calidad del software monitoreando apropiadamente tanto los productos de software como el proceso de desarrollo que los genera.
- Asegurar el cumplimiento de los estándares y procedimientos establecidos para el software y el proceso de software establecidos.

7.2 DEFINICIONES

Calidad: “Propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie” (Real Academia Española)

Calidad del Software: “Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas”. IEEE

Calidad del Software: “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”. Para asegurar la calidad del producto desarrollado por el grupo de Sistemas de Información, se establecen dos etapas de pruebas.

7.3 MODELO DE CALIDAD

El Grupo de Sistemas de Información ha diseñado una metodología para asegurar la calidad del desarrollo realizado por el grupo el cual busca cumplir los atributos de calidad en las funcionalidades desarrolladas:

- Adecuación:** Capacidad del producto para proporcionar un conjunto de funciones para tareas y objetivos de usuarios especificados.
- Exactitud:** Capacidad del producto de software para proporcionar los resultados o efectos acordados con el grado necesario de precisión.
- Interoperabilidad:** Capacidad del producto de software para interactuar con uno o más sistemas especificados.
- Seguridad de Acceso:** Capacidad del producto de software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que se deniega el acceso a las personas o sistemas autorizados.
- Cumplimiento funcional:** Capacidad del producto de software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con la funcionalidad.

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

Para el cumplimiento de estos atributos se consideran dos procesos para la evaluación del producto de software:



- **Prueba de calidad Interna**

Toma en cuenta todas las características del producto software desarrollado, debe ser medido y evaluado tomando en cuenta los requisitos especificados y aprobados por el usuario, y realizando las siguientes actividades:

1. Análisis del requerimiento especificado.
2. Identificación de escenarios de pruebas.
3. Ejecución y documentación de pruebas.
4. Informar y documentar los errores detectados.
5. Revalidar correcciones.

Este proceso lo llevará un funcionario del grupo de sistemas de información para lo cual se realizará un plan de ejecución y realización de pruebas (Test Case) y documentará los errores detectados (Bug) en la herramienta establecida (Test Manager) asociada a una tarea en la plataforma de Azure DevOps. Cuando lo amerite, y dependiendo de la complejidad o impacto del requerimiento se deberán realizar pruebas de integración para verificar que todas las interacciones con otras funcionalidades relacionadas operan correctamente.

Cuando se trate de implementación de un nuevo sistema de información es necesario considerar pruebas de carga, concurrencia, seguridad y verificación de estándares. Para el caso de mantenimientos aditivos o sistemas existentes, las pruebas de seguridad estarán orientadas a la separación de funciones y roles de acuerdo a lo especificado en el caso de uso.

Ejemplo de Documentación de Set de Pruebas en el Test Manager:



The screenshot shows a web-based testing tool interface. At the top, there are navigation tabs: 'Centro de prue...', 'Plan', 'Prueba', 'Seguimiento', and 'Organizar'. Below this, there are sub-tabs: 'Contenido', 'Resultados', 'Propiedades', and 'Configuración de ejecución'. The main content area displays a test case titled 'Test Case 1862: CU-SIR-BIE-0001_v4.0 - Ajuste en pantalla principal BIE - Actividad Económica Principal y subsector -BUG'. The test case details include: 'Iteration: Proyectos GSI 2016/Release 1/Diciembre', 'STATUS: Assigned To: Julian Roberto Medina, State: Design, Priority: 2', and 'DETAILS: Automation status: Not Automated, Area: Proyectos GSI 2016/SIR'. Below the details, there are tabs for 'STEPS', 'SUMMARY', 'TESTED BACKLOG ITEMS', 'LINKS', 'ATTACHMENTS', and 'ASSOCIATED AUTOMATION'. The 'STEPS' tab is active, showing a list of test steps with a table structure. The table has columns for 'Acción' and 'Resultado esperado'. Step 1 describes selecting the principal activity and subsector. Step 2 describes selecting 'Adquisición' in the 'modalidad' field. Step 3 describes enabling fields for 'Sector', 'Actividad Económica Principal', and 'Subsector'. Step 4 describes adding a 'Tipo de Emprendimiento' field and selecting 'Plan de negocio' in the 'modalidad' field. A note indicates that the 'Tipo de emprendimiento' field should have options like 'Productivo', 'Adquisición de predio Rural y Productivo', and 'Compra de establecimiento de comercio'. A link 'Haga clic aquí para agregar un paso' is provided at the bottom.

• Prueba de calidad Externa

Desde una visión externa enfocadas a la ejecución del producto, es evaluada en un ambiente simulado. Esperando eliminar la mayor cantidad de fallas posibles durante las pruebas, sin afectar la arquitectura del software. En este proceso se involucra al usuario solicitante o funcionarios asignados por el solicitante de BIE con el propósito de que de un aval del producto desarrollado mediante la firma o aprobación electrónica de una Acta de Cierre o si llegase a fallar el solicitante enviara un informe de las pruebas realizadas para su revisión.

En este proceso se podrá involucrar a algunos asistentes de información cuando los productos desarrollados posean características que requieran ser probados en un ambiente similar al de los usuarios en cuanto a recursos ubicación geográfica entre otros.

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

7.4 CONSIDERACIONES DE LA ETAPA DE PRUEBAS

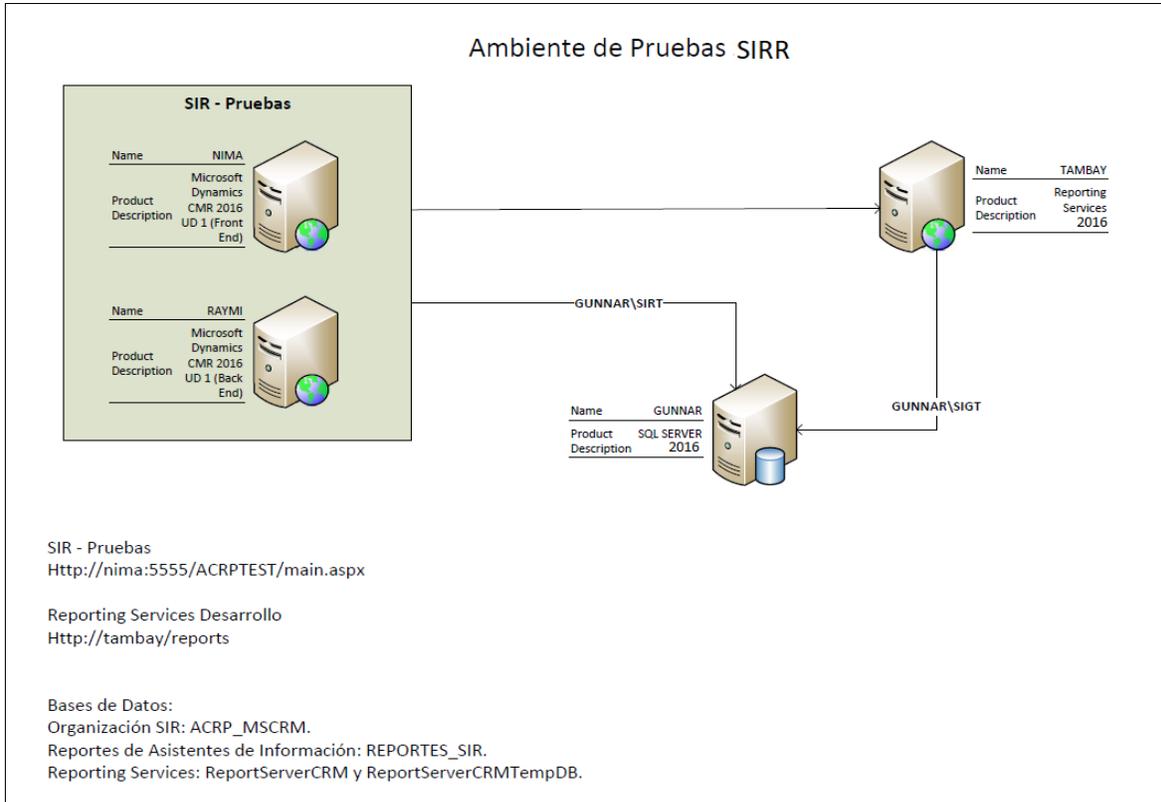
- A. Todos los desarrollos deben superar la etapa de pruebas antes de pasar a Producción, se incluyen los controles de cambio y soportes que se resuelvan con ajustes al Software.
- B. El desarrollador debe informar al Coordinador del Grupo de sistemas de Información, y a los demás participantes del Grupo asignados al requerimiento, la terminación del desarrollo para la asignación del inicio de la etapa de prueba, con el propósito de realizar la asignación correspondiente en Azure DevOps.
- C. Todas las pruebas deben ser documentadas en el TestManager y relacionadas a la tarea de pruebas asignada en el Azure DevOps.
- D. Todos los desarrollos y controles de cambio que impliquen ajustes o creación de nuevos roles se deben incluir en la solución generada para publicar en los ambientes de pruebas y producción.
- E. Siempre se deberá tener en cuenta el flujo de la etapa de pruebas en este documento.

7.5 AMBIENTE DE PRUEBAS Y CAPACITACIÓN

El ambiente de pruebas de los diferentes sistemas de información, son administrados por el grupo de Infraestructura y Soporte, quienes realizarán las publicaciones y la asignación de los roles a los usuarios genéricos, para ello los desarrolladores deberán realizar la solicitud con los correspondientes RFC, a través de la herramienta establecida (ARANDA). Algunas actividades de configuración podrán estar en manos del equipo de pruebas. Este ambiente también puede disponerse para capacitaciones requeridas sobre los diferentes sistemas.

7.5.1 SIRR

Se tiene establecida la siguiente configuración de servidores virtuales para el ambiente de pruebas:



7.5.2 ARPA

En el documento de Arquitectura de ARPA se encuentra la descripción de los ambientes:

http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/ARPA/Documentación%20Modernización%20SIPAS/ACR_ModernizacionSIPAS_Disenio_V5.docx

<http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/ARPA/Documentos%20de%20Dise%C3%B1o%20para%20Refactorizaci%C3%B3n%20ARPA%20V6/Documento%20Arquitectura%20ARPA.docx>

7.5.3 SARA

En el documento de Arquitectura de SARA en el ítem 5. Vista de Despliegue se describen el ambiente de desarrollo:

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

<http://otidocs.acr.int/sites/OTI/Documentos%20compartidos/OTI/GASI/Desarrollos%20Internos/SARA%20V2/Documento%20Arquitectura%20SARA.doc>
[X](#)

7.6 MANEJO DE DATOS EN AMBIENTE DE PRUEBAS

- **SIRR**

Para contar con los datos necesarios en los ambientes de pruebas, se aconseja realiza copia de la base de datos del ambiente productivo cada mes o cuando el grupo de desarrollo lo considere necesario para lo cual realizará solicitud al administrador de bases de datos a través de Soporte la herramienta establecida.

Dentro del procedimiento de replicar la base de datos de producción a pruebas se realiza una reducción de esta excluyendo documentos adjuntos relacionados a las PPR, igualmente se realiza un shrink y se correrá procedimiento encargado de enmascarar o modificar los datos sensibles de las PPR como son Nombre, Identificación y datos de ubicación. Igualmente se correrá procedimiento para modificar los new_name de las entidades donde se reflejan los datos de las PPR.

7.7 USUARIOS DE PRUEBA

Para el desarrollo de las pruebas tanto internas como con los interesados, se deben utilizar los usuarios genéricos a los cuales se les asignarán los roles requeridos para las interactuar con las funcionalidades.

- **Gestión de Contraseñas:**

La administración de los usuarios y contraseñas está a cargo del Grupo de Infraestructura y Soporte, quienes deben informar al coordinador del Grupo de Sistemas de Información siempre que se actualicen las contraseñas de estos usuarios y esté será el encargado de replicar a los funcionarios y contratistas de su grupo.

Se recomienda que la información del ambiente de pruebas SIRR sea actualizado con una periodicidad mensual o según necesidad, para lo cual el Coordinador del Grupo de Sistemas de Información generara solicitud a través de soporte@reincorporacion.gov.co al Grupo de Sistemas de Información.

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

8 HERRAMIENTAS PARA GESTIÓN DE PROYECTOS DE DESARROLLO

8.1 HERRAMIENTAS DE GESTIÓN

El grupo de Apoyo a Sistemas de Información utiliza para la gestión y control del desarrollo de software varias herramientas que de manera articulada brindan la información de trazabilidad de cada requerimiento desde su solicitud inicial hasta la entrega del producto implementado.

La primera herramienta corresponde a la implementada por la mesa de ayuda, pues allí es donde se radican de manera inicial los requerimientos por parte de los usuarios. Cuando los requerimientos llegan a la OTI por SIGOB o a través de un acta de trabajo o por correo electrónico al buzón del coordinador del Grupo de Sistemas de Información, entonces se debe reenviar el requerimiento a soporte@reincorporacion.gov.co para se cree el caso en la herramienta de la mesa de ayuda.

Por cada requerimiento recibido se debe crear un grupo de tareas (Backlog) en la plataforma Azure DevOps que es la segunda herramienta utilizada para la gestión de desarrollo. Según la complejidad del caso se crean una o varias tareas para el análisis y diseño, el desarrollo, las pruebas, y la documentación y entrega del producto implementado.

El caso abierto en la herramienta de mesa de ayuda se debe cerrar indicando el id de la tarea creada en el TFS. Y en la descripción de las tareas se debe hacer referencia al id del caso en la mesa de ayuda.

Como complemento a lo anterior, también se puede utilizar el Microsoft Project para facilitar el control general del cronograma de actividades o gestión de proyectos. Esto para cuando el requerimiento solicitado corresponda a desarrollos que demanden esfuerzos, recursos y tiempos de entrega mayores, a criterio de la coordinación del Grupo de Sistemas de Información.

8.2 INFRAESTRUCTURA DE LAS HERRAMIENTAS DE GESTIÓN

Las herramientas utilizadas se encuentran implementadas en la siguiente infraestructura:

 ARN AGENCIA PARA LA REINCORPORACIÓN Y LA NORMALIZACIÓN	GUÍA DE DESARROLLO DE SOFTWARE	CÓDIGO: TI-G-06	
		FECHA 2021-05-31	VERSIÓN V- 1

		
Product: Aranda (Mesa de Ayuda)	Product: Azure DevOps	Product: Project Server

9 BIBLIOGRAFIA

1. Principios de SOILD, Antonio Leyva. Recuperado de:
<https://devexperto.com/principio-responsabilidad-unica/>
2. SOLID, Samuel Martín Gómez-Calcerrada. Recuperado de:
<https://www.adictosaltrabajo.com/tutoriales/solid-5/#01>
3. <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>
4. <https://es.wikipedia.org/wiki/CamelCase>
5. <http://yakzia.in/buenas-practicas-pascal-case-y-camel-case/>
6. [https://msdn.microsoft.com/es-ES/library/ms229002\(v=vs.110\).aspx](https://msdn.microsoft.com/es-ES/library/ms229002(v=vs.110).aspx)
7. <https://code2read.com/2015/03/13/csharp-ood-principios-solid-caso-practico-ood-parte1/>