**Paquete de Despliegue**

**Arquitectura de Software y Diseño Detallado**

**Perfil Básico**

**Notas:**

Este documento es propiedad intelectual de la organización del autor. De todas formas, la información contenida en el documento es de uso libre. La distribución parcial o total del documento está autorizada para uso no comercial mientras que la siguiente nota legal sea mencionada:

© *École de Technologie Supérieure*

El uso comercial de este documento está estrictamente prohibido. Este documento es distribuido para mejorar el intercambio de información técnica y científica.

Este material está proporcionado en el estado en que se encuentra. El autor no garantiza ningún tipo, explícito o implícito, de cualquier asunto, sin estar limitado a, garantía o aptitud para propósito o comercialización, exclusividad, o resultados obtenidos del uso del material.

Los procesos descritos en este Paquete de Despliegue no intentan excluir o desalentar el uso de procesos adicionales que las Pequeñas Organizaciones puedan encontrar útiles.

|  |  |
| --- | --- |
| **Autores Documento Original** | F. GUILLEMOT – École de Technologie Supérieure (ETS), (Canadá)  R. CHAMPAGNE – École de Technologie Supérieure (ETS), (Canadá) |
| **Autor Versión Español** | LUIGGI MENDOZA – Universidad Peruana de Ciencias Aplicadas (Perú) |
| **Editores** | LUIS GARCIA – Universidad Peruana de Ciencias Aplicadas (Perú)  C. Y. LAPORTE – École de Technologie Supérieure (ETS), (Canadá) |
| **Fecha de creación** | 16 de Junio de 2013 |
| **Fecha de última actualización** | 14 de Diciembre de 2013 |
| **Estado** | Versión Final – Lista para revisión final |
| **Versión** | 1.0 |

Historial de Versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 16/06/2013 | 0.1 | Creación del documento basado en Deployment\_Software\_Design\_v0 5 | Luiggi Mendoza |
| 23/06/2013 | 0.2 | Revisión del documento | Cynthia Ramos |
| 25/06/2013 | 0.3 | Aplicación de correcciones | Luiggi Mendoza |
| 13/08/2013 | 0.4 | Revisión del documento | Luis García |
| 25/08/2013 | 0.5 | Aplicación de correcciones | Cynthia Ramos |
| 14/11/2013 | 0.6 | Revisión del documento | Luis García |
| 14/11/2013 | 0.7 | Aplicación de correcciones | Luiggi Mendoza |
| 24/12/2013 | 1.0 | Versión Final – Lista para revisión final | Cynthia Ramos |

Abreviaciones/Acrónimos

|  |  |
| --- | --- |
| **Abre./Acro.** | **Definiciones** |
| PD | Paquete de Despliegue - conjunto de artefactos desarrollados para facilitar la implementación de un conjunto de prácticas, de un marco de trabajo seleccionado, en una Pequeña Organización. |
| PO | Pequeña Organización – una empresa, organización, departamento o proyecto que posee como máximo 25 personas. |
| POs | Pequeñas Organizaciones. |
| LT | Líder Técnico |
| AN | Analista |
| DIS | Diseñador |

Tabla de Contenidos

[1. Descripción Técnica 4](#_Toc372233507)

[Propósito del documento 4](#_Toc372233508)

[¿Por qué es importante la Arquitectura de Software y Diseño Detallado? 4](#_Toc372233509)

[2. Definiciones 6](#_Toc372233510)

[Términos Genéricos 6](#_Toc372233511)

[Términos Específicos 6](#_Toc372233512)

[3. Relaciones con ISO/IEC 29110 9](#_Toc372233513)

[4. Descripción de Procesos, Actividades, Tareas, Pasos, Roles y Productos 11](#_Toc372233514)

[Descripción de Roles 17](#_Toc372233515)

[Descripción de Productos 17](#_Toc372233516)

[Descripción de Artefactos 21](#_Toc372233517)

[5. Plantilla 22](#_Toc372233518)

[Plantilla del Diseño de Software 22](#_Toc372233519)

[6. Ejemplo 23](#_Toc372233520)

[Ejemplo de Práctica de Ciclo de Vida de diseño de Software 23](#_Toc372233521)

[7. Listas de verificación 24](#_Toc372233522)

[8. Herramientas 26](#_Toc372233523)

[Herramientas de Trazabilidad 26](#_Toc372233524)

[9. Referencias a otros Estándares y Modelos 28](#_Toc372233525)

[Matriz de Referencia ISO 9001 28](#_Toc372233526)

[Matriz de Referencia ISO/IEC 12207 29](#_Toc372233527)

[Matriz de Referencia CMMI 33](#_Toc372233528)

[10. Referencias 34](#_Toc372233529)

[11. Formulario de Evaluación 35](#_Toc372233530)

# 1. Introducción

Este Informe de diseño soporta el Perfil Básico definido en la norma ISO/IEC 29110 Parte 5-1-2: Guía de Gestión e Ingeniería [[1]](#footnote-1).

Un PD es un conjunto de artefactos desarrollados para facilitar la implementación de un conjunto de prácticas en una Pequeña Organización (PO). Un PD no es un modelo de proceso de referencia (esto es, no es preceptivo). Los elementos de un PD típico son: descripción de procesos, actividades, tareas, roles y productos, plantillas, lista de verificación, ejemplo y referencia a estándares y modelos, y herramientas.

## Propósito del documento

El propósito de este documento, titulado “*Paquete de Despliegue -* *Arquitectura de Software y Diseño* Detallado de Quick List” es proveer a los implicados en el sistema guías y materiales para poder implementar e interpretar un buen *Diseño de Software*.

El contenido de este documento es enteramente *informativo*.

## ¿Por qué es importante la ?

Asegura que para la solución propuesta (QuickList) se habrá meditado alguna idea previa a su implementación. Construir algo sin diseñarlo típicamente produce una solución que no cumple con los requerimientos, se entrega tarde, excede el presupuesto o es de baja calidad.

Invertir esfuerzos en **documentación** explícita habilita la comunicación y negociación entre los stakeholders del proyecto, más específicamente a aquellos que tienen interés en el diseño. Capturando un diseño en alguna forma (documento electrónico, documento en papel, modelos…) no es solo útil mientras el proyecto de software está activo, sino también para el futuro mantenimiento y mejoras.

## Alcance

Este Informe de Diseño está dirigido a los aprendices e instructores del tecnólogo en Análisis y desarrollo de sistemas de información con la ficha de caracterización N°750566 del centro de diseño y metrología de la regional Distrito capital, para aplicar los conocimientos aprendidos durante la etapa electiva de nuestra formación y para construir un software que contribuya en la automatización de ésta, la cual tiene por objetivo principal el gestionar los distintos procesos académicos (Asistencia, calificaciones, seguimiento de plan de estudios, formato de etapa lectiva).

## Personal involucrado

|  |  |
| --- | --- |
| **Nombre** | Andrés Santana |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | ADSI-Tecnólogo |
| **Responsabilidad** | Análisis de información, diseño y programación del SIS-I |
| **Información de contacto** | Es.andres@live.com |

|  |  |
| --- | --- |
| **Nombre** | Alejandra Sierra |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | ADSI-Tecnóloga |
| **Responsabilidad** | Análisis de información, diseño y programación del SIS-I |
| **Información de contacto** | Alejasierramendez@hotmail.com |

|  |  |
| --- | --- |
| **Nombre** | Cesar Torres |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | ADSI-Tecnólogo |
| **Responsabilidad** | Análisis de información, diseño y programación del SIS-I |
| **Información de contacto** | Cesartorres-13@hotmail.com |

# 3. Definiciones

En esta sección, especificando los dos conjuntos de definiciones. El primer conjunto define los términos utilizados en todos los Paquetes de Despliegue, esto es, términos genéricos. El segundo conjunto de términos utilizados en este Paquete de Despliegue, es decir, los términos específicos.

## Términos Genéricos

***Proceso:*** conjunto de actividades interrelacionadas o que interactúan entre ellas para transformar entradas en salidas. [ISO/IEC 12207]

***Actividad:*** un conjunto de tareas cohesivas de un proceso. [ISO/IEC 12207]

***Tarea:***acción requerida, recomendada o permisible que intenta contribuir al logro de uno o más resultados de un proceso.[ISO/IEC 12207]

***Sub-Tarea:*** cuando una tarea es compleja, se divide en sub-tareas.

***Paso:*** en un paquete de despliegue, una tarea es descompuesta en una serie de pasos.

***Rol***: una función definida para ser realizada por un miembro del equipo del proyecto, como pruebas, archivamiento, inspección, codificación. [ISO/IEC 24765]

***Producto:*** pieza de información o entregable que puede ser producida (no obligatoriamente) por una o muchas tareas (por ejemplo, un documento de diseño, código fuente).

***Artefacto:*** información, que puede no estar listada en la norma ISO/IEC 29110 Parte 5, pero que puede ayudar a una PO durante la ejecución del proyecto.

## Términos Específicos

*Componente:* Los elementos computacionales y de almacenamiento de información principales que se ejecutan en el sistema. [CLEMENTS 03, p. 107]

*Cliente*: organización o persona que paga a la PO para crear un producto de software.  
NOTA: el adquiridor o el usuario es cliente. [ISO 9000]

*Preocupaciones de diseño:* Un área de interés con respecto al diseño de software. [IEEE 1016]

*Diseño detallado:* 1. el proceso de refinar y expandir el diseño preliminar de un sistema o componente en la medida en que el diseño está lo suficientemente completo para ser implementado 2. el resultado del proceso en (1). [ISO/IEC 24765]

*Requerimiento funcional:* Un requerimiento que especifica una función que debe realizar el sistema o un componente del sistema. [ISO/IEC 24765]

*Módulo:* Una unidad de implementación de software que provee una unidad de funcionalidad coherente. [CLEMENTS 03, p. 43]

*Requerimiento no funcional:* un requerimiento de software que no describe lo que hará el software sino cómo lo hará. ISO/IEC 24765, Vocabulario de Ingeniería de Sistemas y Software. Sinónimos: Limitaciones de diseño, requerimientos no funcionales vs requerimientos funcionales. EJEMPLO: requerimientos de rendimiento del software, requerimientos de interfaces externas del software, restricciones de diseño del software y atributos de calidad del software. Los requerimientos no funcionales son a veces difíciles de probar, por eso usualmente son evaluados subjetivamente. [ISO/IEC24765]

Arquitectura de software: La estructura o estructuras de un sistema, que consiste en elementos, sus propiedades visibles externamente y las relaciones entre ellas. [CLEMENTS 03, p. xxv]

Diseño de software: En este Paquete de Despliegue, la combinación de Diseño arquitectónico y Diseño detallado.

Producto de Software: El conjunto de programas de computadora, procedimientos y las posibles documentación asociada e información. [ISO/IEC 12207]

***Stakeholder:*** Individuo u organización que tiene un derecho, participación, reclamo o interés en un sistema o en la posesión de características que cumplan con sus necesidades y expectativas. [ISO/IEC 12207]

***Subsistema:*** 1. un sistema secundario o subordinado dentro de un sistema más grande. 2. un sistema que es parte de un sistema más grande. [ISO/IEC 24765]

***Trazable:*** tener componentes cuyos orígenes pueden ser determinados. [ISO/IEC24765]

Vista: Una representación del sistema como un todo desde la perspectiva de un conjunto de asuntos de interés relacionados. [IEEE 1471]

# 4. Descripción de Procesos, Actividades, Tareas, Pasos, Roles y Productos

En esta sección, el lector encontrará una actividad, una lista de tareas y roles de Implementación de Software (IS).

* **Proceso: Implementación de Software (IS)**
* **Actividad: IS.3 Arquitectura de Software y Diseño Detallado**
* **Tareas y Roles:**

| **Tareas** | **Roles[[2]](#footnote-2)** |
| --- | --- |
| * IS. 3.1 Asignar Tareas a los miembros del Equipo de Trabajo de acuerdo a cada rol, basado en el *Plan del Proyecto* actual. | LT, AN  DIS |
| * IS.3.2 Comprender la *Especificación de Requerimientos* | AN, DIS |
| * IS.3.3 Documentar o actualizar el *Diseño de Software*. * Analizar la *Especificación de Requerimientos* para generar el diseño arquitectónico, definir interfaces internas y externas. * Describir a detalle, la apariencia y el comportamiento de la interfaz, con base en la *Especificación de Requerimientos* de tal forma que los recursos para su implementación puedan preverse. * Proporcionar el detalle de los *Componentes de Software* y sus interfaces para permitir la construcción en una forma clara. * Generar o actualizar el *Registro de Trazabilidad*. | AN  DIS |
| * IS.3.4 Verificar y obtener la aprobación del *Diseño de Software* * Verificar que la documentación del *Diseño de Software* sea correcta, viable y consistente con la *Especificación de Requerimientos*. * Verificar que el *Registro de Trazabilidad* contenga las relaciones adecuadas entre los requerimientos y los elementos del *Diseño de Software*. Los resultados encontrados son documentados en *Resultado de Verificación* y las correcciones se realizan hasta que el documento ha sido aprobado por el AN. Si fueran necesarios cambios significativos, se propone una *Solicitud de Cambio*. | AN  DIS |
| * IS.3.5 Establecer o actualizar los *Casos de Prueba* y *Procedimientos de Prueba* para pruebas de integración basadas en la *Especificación de Requerimientos* y el *Diseño de Software*. * El cliente provee datos de prueba, en caso de ser necesarios. | DIS |
| * IS.3.6 Verificar y obtener la aprobación de los *Casos de Prueba* y *Procedimientos de Prueba*. * Verificar la consistencia entre la *Especificación de Requerimientos*, *Diseño de Software* y los *Casos de Prueba* y *Procedimientos de Prueba*. Los resultados encontrados están documentados en el *Resultado de Verificación* y las correcciones son realizadas hasta que el documento es aprobado por el AN. | DIS  AN |
| * IS.3.7 Actualizar el *Registro de Trazabilidad* incorporando los *Casos de Prueba* y *Procedimientos de Prueba*. | DIS |
| * IS.3.8 Incorporar el *Diseño de Software* y el *Registro de Trazabilidad* a la *Configuración de Software* como parte de la línea base. Incorporar los *Casos de Prueba* y *Procedimientos de Prueba* al *Repositorio del Proyecto.* | LT |

# 5. Descripción de Procesos, Actividades, Tareas, Pasos, Roles y Productos

**Proceso: Implementación de Software (IS)**

De acuerdo a la definición en la norma ISO/IEC 29110, el propósito del proceso de Implementación de Software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas para los productos de software nuevos o modificados, de acuerdo a los requerimientos especificados.

## Descripción de Roles

Esta es una lista en orden alfabético de los roles, abreviaciones y lista de competencias.

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Rol*** | ***Abreviación*** | ***Competencias*** |
| 1. | Analista | AN | Conocimiento y experiencia que permita obtener, especificar y analizar los requerimientos.  Conocimiento en diseño de interfaces de usuario y criterios ergonómicos.  Conocimiento en técnicas de revisión.  Conocimiento en técnicas de edición.  Experiencia en desarrollo y mantenimiento de Software. |
| 2. | Diseñador | DIS | Conocimiento y experiencia en componentes de software y diseño de arquitectura.  Conocimiento de técnicas de revisión.  Conocimiento y experiencia en la planificación y ejecución de pruebas de integración.  Conocimiento en técnicas de edición.  Experiencia en desarrollo y mantenimiento de software. |
| 3. | Líder Técnico | LT | Conocimiento y experiencia en el dominio de proceso del software. |

## Descripción de Productos

Esta es una lista en orden alfabético de los productos de entrada, salida y de uso interno del proceso, sus descripciones, posibles estados y el origen del producto.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nombre** | **Descripción** | **Origen** |
| 1. | *Casos de Prueba y Procedimientos de Prueba* | Los *Casos de Prueba* pueden incluir:   * Identificación del caso de prueba * Elementos a probar * Especificaciones de entrada * Especificaciones de salida * Necesidades del entorno * Requerimientos de procedimientos especiales * Dependencias de interfaz     Los *Procedimientos de Prueba* pueden incluir:   * Identificación: nombre de la prueba, descripción de la prueba y fecha de finalización de la prueba * identificación de posibles problemas de implementación * Identificación de la persona que completó el procedimiento de prueba * Identificación de los requerimientos previos * Identificación de los pasos del procedimiento incluyendo el número de paso, la acción requerida por el tester y los resultados esperados     Los estados aplicables son: verificado e incorporado en la línea base. | Implementación de Software |
| 2. | Configuración deSoftware | Un conjunto de productos software único, identificado y consistente:   * *Especificación de Requerimientos* * *Diseño de Software* * *Registro de Trazabilidad* * *Componentes de Software* * *Software* * *Casos de Prueba y Procedimientos de Prueba* * *Reporte de Pruebas* * *Guía de Operaciones del Producto* * *Manual de Usuario* * *Documentación de Mantenimiento*   Los estados aplicables son: entregado y aceptado. | Implementación de Software |
| 3. | *Diseño de Software* | Información textual y gráfica de la estructura del Software. Esta estructura puede incluir las siguientes partes:  Diseño Arquitectónico de Software – Describe la estructura global del *Software*:   * Identifica los stakeholders del diseño arquitectónico y sus preocupaciones * Identifica los mecanismos arquitectónicos relevantes (patrones, tácticas, heurísticas, buenas prácticas, …) * Identifica los tipos de vistas que son relevantes para transmitir la arquitectura del software, tomando en consideración las preocupaciones de los stakeholders y los requerimientos varios (funcionales y no funcionales) * Provee vistas arquitectónicas relevantes del software en varias formas (diagramas, modelos, tablas, texto plano, …) * Identifica y describe los elementos principales de la arquitectura de software (subsistemas, capas, módulos) y sus relaciones * Identifica y describe los *Componentes* de Software requeridos, sus interfaces y las relaciones entre ellos * Describe la razón fundamental, provee cualquier análisis utilizado para producir la solución, identifica riesgos conocidos e inconsistencias   Diseño Detallado de Software – incluye detalles de los *Componentes de Software* para facilitar su construcción y pruebas dentro del entorno de programación:   * Proporciona diseño detallado (puede ser representado como un prototipo, diagrama de flujo, diagrama entidad relación, pseudo código, etc.) * Proporciona el formato de entrada / salida de los datos * Proporciona especificaciones de las necesidades de almacenamiento de los datos * Establece convenciones de denominación de los datos requeridos * Define el formato de las estructuras de datos requeridas * Define los campos de datos y el propósito de cada elemento de datos requerido * Proporciona las especificaciones de la estructura del programa   Los estados aplicables son: verificado e incorporado en la línea base. | Implementación de Software |
| 4. | *Especificación de Requerimientos* | Incluye una introducción y descripción de los requerimientos. Puede contener:   * Introducción – descripción general del software y su uso en el alcance del negocio del cliente; * Descripción de Requerimientos: * Funcionalidad – establece las necesidades a ser satisfechas por el software cuando es usado en condiciones específicas. Funcionalidades tienen que ser adecuadas, precisas y seguras. * Interfaz de Usuario – definición de aquellas características de la interfaz de usuario que permiten entender y aprender el software fácilmente para que el usuario sea capaz de realizar sus tareas de manera eficiente incluyendo la descripción de la interfaz ejemplo; * Interfaces externas – definición de las interfaces con otro software o hardware; * Confiabilidad – Especificación del nivel de ejecución del software concerniente a la madurez, tolerancia a fallos y recuperación; * Eficiencia – Especificación del nivel de ejecución del software concerniente al tiempo y uso de los recursos; * Mantenimiento – descripción de los elementos que facilitan la comprensión y ejecución de las futuras modificaciones del software; * Portabilidad – descripción de las características del software que permiten transferirlo de un lugar a otro; * Limitaciones/restricciones del diseño y la construcción – necesidades impuestas por el cliente; * Interoperabilidad – capacidad de dos o más sistemas o componentes de software que permiten intercambiar información entre ellos y usarla. * Reusabilidad – característica que cualquier producto/sub-producto, o una parte de él, que pueda ser usada por muchos usuarios como producto final, en el propio desarrollo de software, o en la ejecución de otros productos software. * Legal y regulativo – necesidades impuestas por leyes, regulaciones, etc.   Cada requerimiento es identificado, único y es verificable o puede ser asesorado.  Los estados aplicables son: verificado, validado e incorporado en la línea base. | Implementación de Software |
| 5. | *Registro de Trazabilidad* | Documenta la relación entre los requisitos incluidos en la *Especificación de Requerimientos*, los elementos del *Diseño de Software*, los *Componentes de Software*, los *Casos de Prueba y Procedimientos de Prueba*.   * Identifica los requerimientos de la *Especificación de Requerimientos* por rastrear. * Proporciona el mapeo (hacia adelante y hacia atrás) de los requisitos a los elementos del *Diseño de Software*, los *Componentes de Software*, los *Casos de Prueba y Procedimientos de Prueba*.   Los estados utilizados son: verificado, en línea base y actualizado. | Implementación de Software |
| 6. | Resultados de Validación | Puede incluir el registro de:   * Participantes * Fecha * Lugar * Duración * Lista de comprobación de la validación * Elementos aprobados para la validación * Elementos no aprobados para la validación * Elementos pendientes para la validación * Defectos identificados durante la validación | Gestión de Proyectos  Implementación de Software |

## Descripción de Artefactos

Esta es una lista alfabética de los artefactos que pueden ser producidos para facilitar la documentación de un proyecto. Los artefactos no son requeridos por la Parte 5, son opcionales.

|  | **Nombre** | **Descripción** |
| --- | --- | --- |
| 1. | Diagramas de Diseño | Los diagramas facilitan la representación gráfica del diseño. Se pueden utilizar diagramas informales de cajas y líneas o notaciones especializadas como Unified Modelling Language (UML). Sin embargo, cabe notar que no toda la información del diseño se expresa fácilmente en diagramas. Tablas y prosa, entre otros, también son maneras muy útiles para expresar algunas preocupaciones del diseño. |

# 6. STAKEHOLDERS DE DISEÑO Y PREOCUPACIONES

**6.1 STAKEHOLDERS DE DISEÑO Y PREOCUPACIONES**

**6.1.1 Stakeholders de diseño y preocupaciones**

|  |  |
| --- | --- |
| **Tipo de usuario** | Administrador |
| **Formación** | ADSI-Tecnólogo |
| **Actividades** | Control y manejo del sistema en general |

|  |  |
| --- | --- |
| **Tipo de usuario** | Instructor |
| **Formación** | Educador |
| **Actividades** | Facilitar el proceso de aprendizaje |

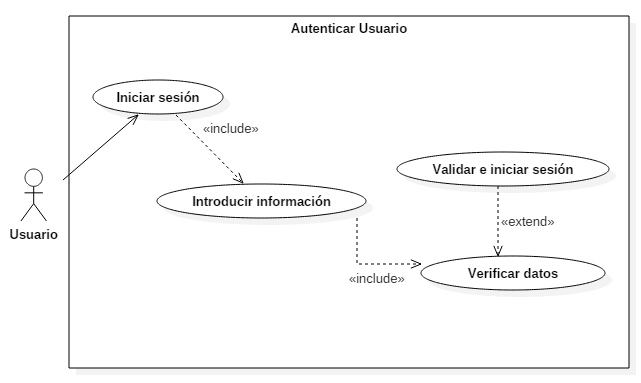
|  |  |
| --- | --- |
| **Tipo de usuario** | Aprendiz |
| **Formación** | ADSI-Tecnólogo |
| **Actividades** | Participación activa en el software |

.

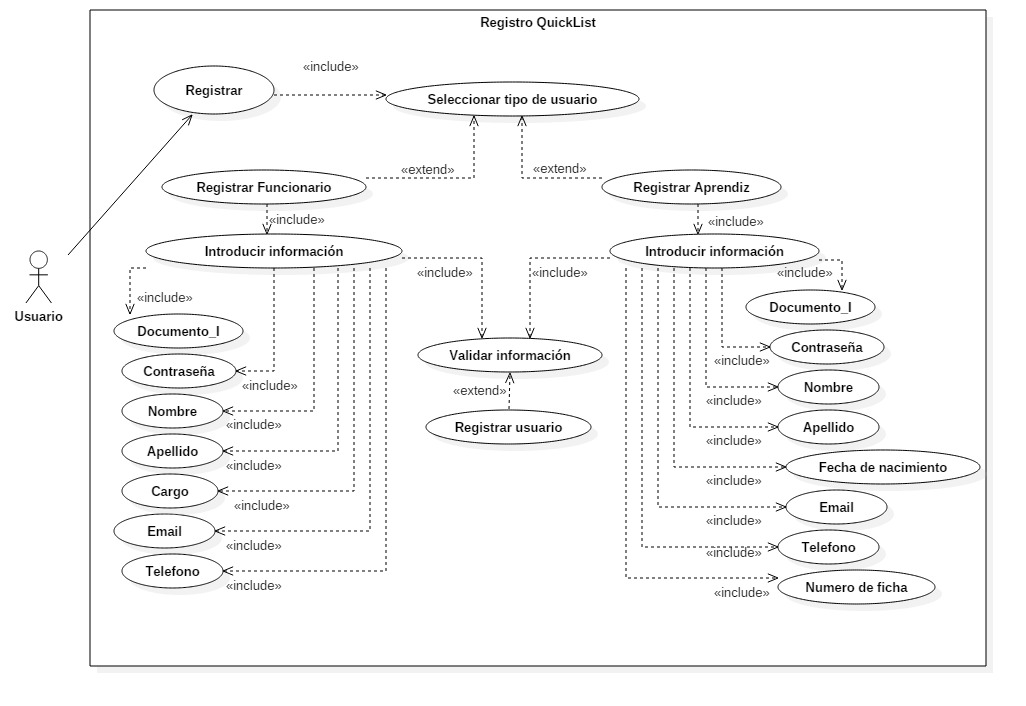
2.2 Vistas de diseño y relaciones para las preocupaciones de diseño

En esta sección podemos ver la relación de los requerimientos funcionales con sus respectivos casos de uso dentro del diseño de software.

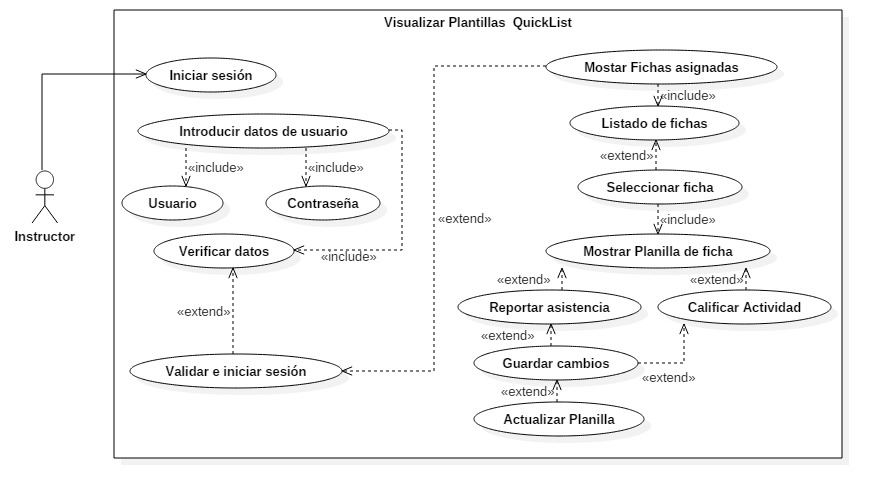
|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Autentificación de Usuario. |
| **Características:** | Los usuarios deberán identificarse para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF05 * RNF08 |
| **Prioridad del requerimiento:**  Alta | |



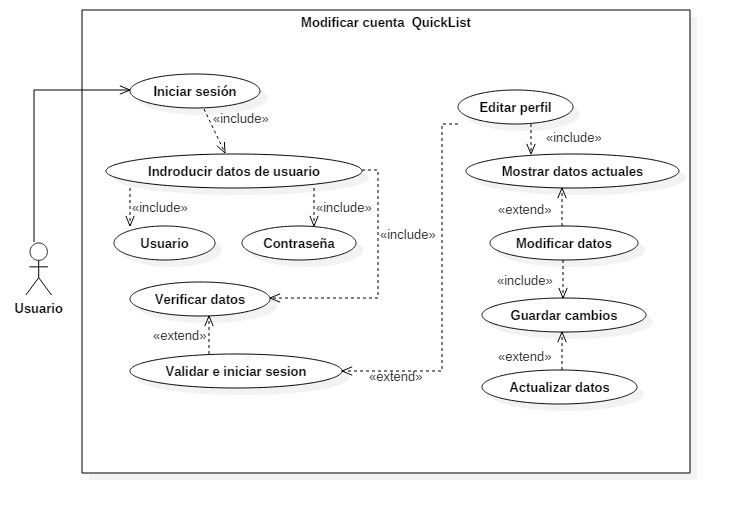
|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF02 |
| **Nombre del Requerimiento:** | Registrar Usuarios. |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario (Aprendiz, Instructor y Administrador) registrarse. El usuario debe suministrar datos como: Documento de identidad, Nombre, Apellido, E-mail, Usuario y Password. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF05 * RNF08 |
| **Prioridad del requerimiento:**  Alta | |

****

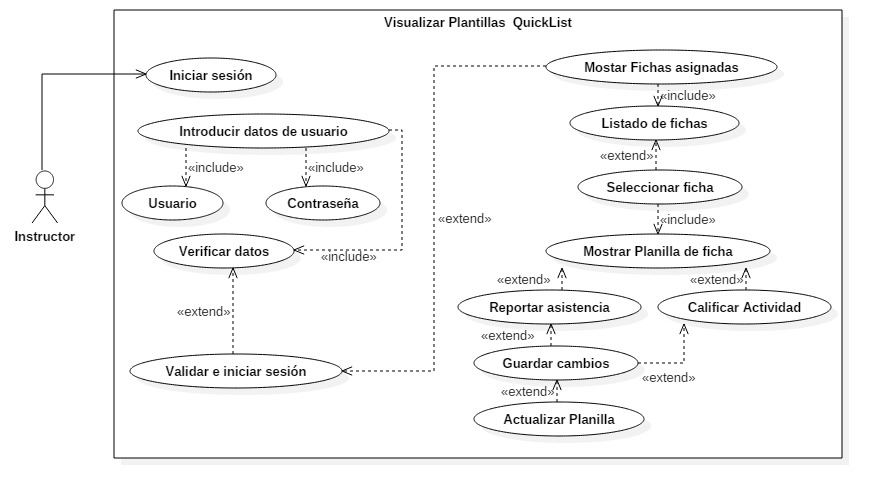
|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF03 |
| **Nombre del Requerimiento:** | Consultar Información. |
| **Características:** | El sistema ofrecerá al usuario información general acerca de la formación Académica, Plan de estudios, Calendario de actividades. |
| **Descripción del requerimiento:** | **Consultar Plan de Estudios:** Muestra información general sobre el plan curricular asignado a la ficha de caracterización, Competencias, actividades ,duración por trimestre académico. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 |
| **Prioridad del requerimiento:**  Alta | |

****

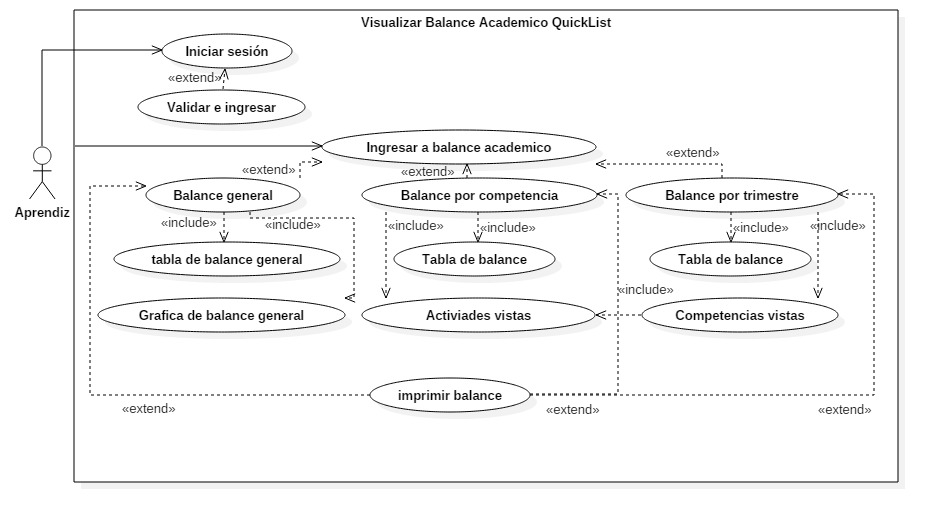
|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF05 |
| **Nombre del Requerimiento:** | Modificar. |
| **Características:** | El sistema permitirá al administrador, instructor y Aprendiz modificar los datos personales, actividades creadas en cuanto al instructor y notas implantadas. |
| **Descripción del requerimiento:** | Permite al administrador modificar datos de los usuarios, fichas y cuentas creadas. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

****

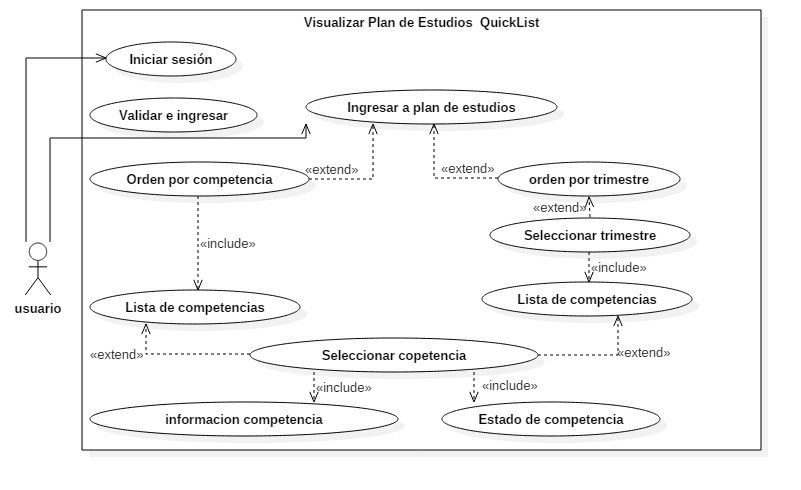
|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF06 |
| **Nombre del Requerimiento:** | Gestión de SFT-Q. |
| **Características:** | Permite gestionar información referente al software. |
| **Descripción del requerimiento:** | **Crear Actividades:** Permite al instructor una vez que haya accedido con su cuenta, crear sus actividades y suministrar información relevante a las mismas a los aprendices, manteniendo una comunicación activa. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 * RNF05 * RNF06 * RNF07 |
| **Prioridad del requerimiento:**  Alta | |

****

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF08 |
| **Nombre del Requerimiento:** | Gestión de SFT-Q. |
| **Características:** | Permite gestionar información referente al software. |
| **Descripción del requerimiento:** | **Consultar:** permite a los aprendices ver información de actividades pendientes y notas de actividades ya calificadas, también permite ver el promedio académico y de asistencia. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 * RNF05 * RNF06 * RNF07 |
| **Prioridad del requerimiento:**  Alta | |

****

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF10 |
| **Nombre del Requerimiento:** | Gestionar Reportes. |
| **Características:** | El sistema permitirá generar reportes. |
| **Descripción del requerimiento:** | Permite al administrador imprimir reportes de los eventos a realizar o concluidos, así como también, ver listados de aprendices por fichas y instructores activos |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 |
| **Prioridad del requerimiento:**  Alta | |

****

# 7. DESCRIPCIÓN DE LA ARQUITECTURA DE SOFTWARE

**7.1 Resumen de la arquitectura de software**

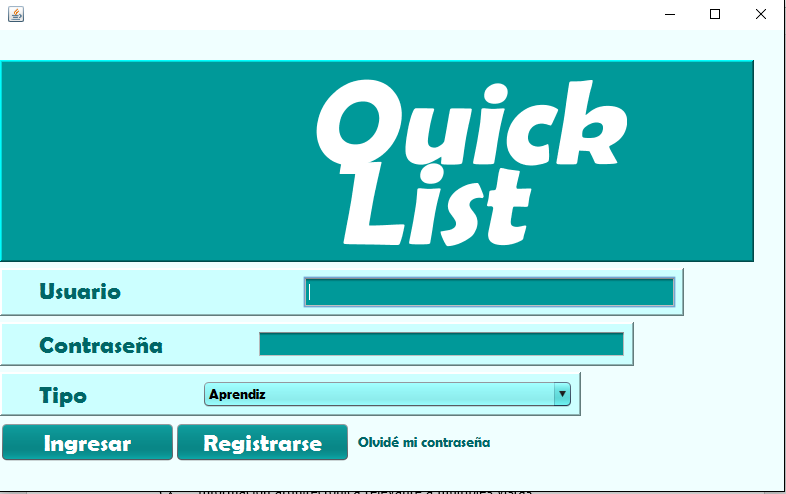
Diseño Arquitectónico de Software – Describe la estructura global del Software -Identifica los tipos de vistas que son relevantes para transmitir la arquitectura del software, tomando en consideración las preocupaciones de los stakeholders y los requerimientos varios (funcionales y no funcionales).

**7.2 Vista 1 de la arquitectura de software**

**3.2.1** Resumen

Esta vista muestra la pantalla de ingreso al software Quicklist.

**3.2.2** Diseño que aplican a esta vista



**3.2.3** Preocupaciones de diseño y requerimientos atendidos por esta vista

|  |  |
| --- | --- |
| **Requerimiento Funcional:** | RF01 |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF05 * RNF08 |

**3.2.4** Descripción de elementos de la vista y sus interfaces

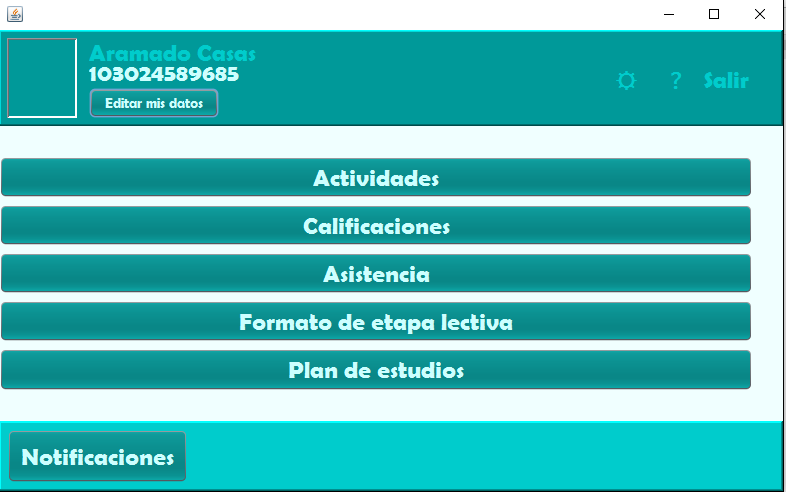
La vista numero 1 contiene un formulario de ingreso el cual permite entrar al software si los datos de usuario son válidos, también contiene un botón de registro el cual conduce a la respectiva vista y por ultimo un botón de olvide mi contraseña.

**7.3 Vista 1 de la arquitectura de software**

**3.2.1** Resumen

Esta vista muestra la pantalla inicial del software Quicklist.

**3.2.2** Diseño que aplican a esta vista



**3.2.3** Preocupaciones de diseño y requerimientos atendidos por esta vista

|  |  |
| --- | --- |
| **Requerimiento Funcional:** | RF03  RF04  RF06 |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF05 * RNF08 |

**3.2.4** Descripción de elementos de la vista y sus interfaces

La vista número 2 contiene la pantalla inicial dependiendo del tipo de ingreso de usuario, esta contiene las opciones de ver actividades, calificaciones, asistencia, formato de etapa electiva y plan de estudios, además de mostrar en la parte superior la información del usuario que está usando el software junto con la opción de editar perfil.

# 8. DESCRIPCIÓN DE DISEÑO DETALLADO

**8.1 Resumen del diseño detallado**

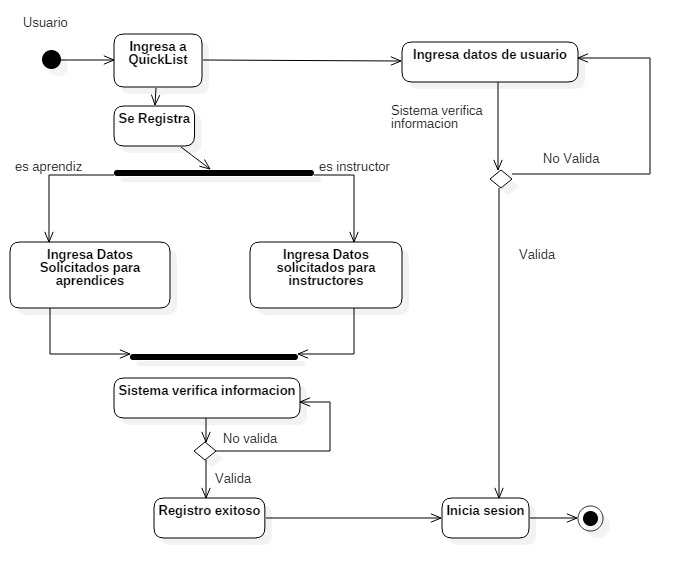
Diseño Detallado de Software – incluye detalles de los *Componentes de Software* para facilitar su construcción y pruebas dentro del entorno de programación, Proporciona las especificaciones de la estructura del programa

**8.2 Diseño detallado del elemento 1**

**8.2.1** Vistas estructurales

|  |
| --- |
| **Vista 1** |
| **Vista 2** |

**8.2.2** Vistas de comportamiento



**8.2.3** Razón fundamental

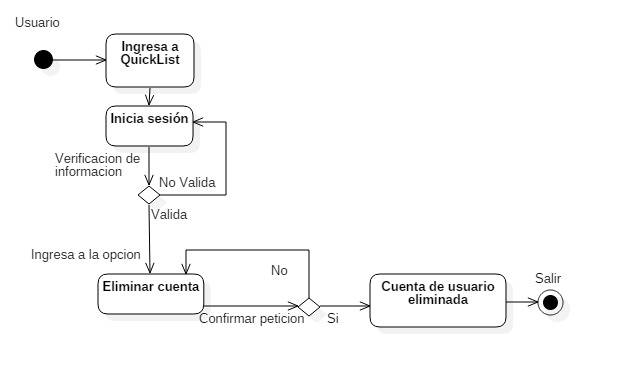
Este elemento se compone la vista 1 y descríbelas acciones que se ejecutan al momento de ingresar al software y adicional mente al momento de registrar un nuevo.

**8.3 Diseño detallado del elemento 1**

**8.3.1** Vistas estructurales

|  |
| --- |
| **Vista 3** |

**8.3.2** Vistas de comportamiento



**8.3.3** Razón fundamental

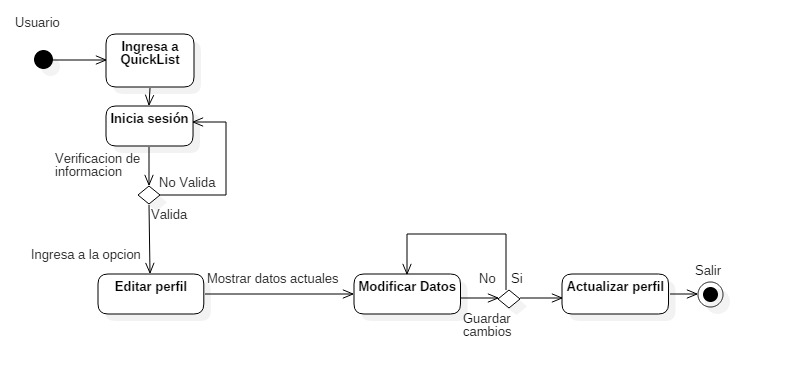
Este elemento se compone la vista 3 y describe las acciones que se ejecutan a la hora de dar de baja a un usuario dentro del software Quick List.

**8.4 Diseño detallado del elemento 1**

**8.4.1** Vistas estructurales

|  |
| --- |
| **Vista 4** |

**8.4.2** Vistas de comportamiento



**8.4.3** Razón fundamental

Este elemento se compone la vista 4 y describe las acciones que se ejecutan a la hora de modificar y actualizar los datos de un usuario dentro del software Quick list.

# 8. Herramientas

**Herramientas de Control de Versiones**

* + GIT

**Software para Descripción del Diseño**

* + Herramientas UML: Visual paradigm UML, Smart Draw, Omondo,...
  + Herramientas estándares de oficina: OpenOffice Writer, Microsoft Visio, Microsoft Word,...
  + Ambientes de desarrollo completo con capacidades de modelamiento: Eclipse, Sparx Systems Enterprise Architect (Windows), Microsoft Visual Studio, ...

## Herramientas de Trazabilidad

* Objetivos:
  + Mantener el vínculo entre la fuente de cada requerimiento a través de su descomposición a la implementación y pruebas (verificación).
  + Asegurar que todos los requerimientos han sido atendidos y que solo se ha desarrollado lo requerido.
  + Útil cuando se conducen evaluaciones de impacto de cambios de los requerimientos, elementos de diseño u otros elementos configurados.

# 8. Referencias

|  |  |
| --- | --- |
| **Clave** | **Referencia** |
| [CLEMENTS 03] | P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford, "Documenting Software Architectures - Views and Beyond", Addison Wesley, 512 pages, 2003, ISBN 0-201-70372-6. |
| [ISO/IEC 12207] | ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes. |
| [ISO/IEC 24765] | ISO/IEC 24765:2010, Systems and Software Engineering Vocabulary. (<http://pascal.computer.org/sev_display/index.action>) |
| [ISO/IEC 29110] | ISO/IEC 29110:2011, Software Engineering — Lifecycle Profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and engineering guide – Generic profile group - Basic Profile. |
| [IEEE 1471] | IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems |
| [IEEE 1016] | IEEE recommended Practice for Software Design Description. |
| [OPENUP] | Open Unified Process (OpenUP), available online at <http://epf.eclipse.org/wikis/openup/>. |

1. 1 ISO/IEC 29110-5-1-2 en español está disponible libre de costo: <http://bvirtual.indecopi.gob.pe/normas/29110-5-1-2.pdf> [↑](#footnote-ref-1)
2. Los roles están definidos en la siguiente sección. Los roles también están definidos ISO/IEC 29110 Part 5-1-2 [↑](#footnote-ref-2)