

# Glosario General de Términos Arquitectura Modelo 4+1 Vistas

<<versión 1.0>>

Historial de Revisión.

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 29/11/2017 | 1.0 | Versión estable | Gabriel Huertas |

# Términos

* ***Stakeholders:***significa ‘interesado’ o ‘parte interesada’, y que se refiere a todas aquellas personas u organizaciones afectadas por las actividades y las decisiones de una empresa [1].
* ***Developers:*** es un programador que se dedica a una o más facetas del proceso de desarrollo de software, un ámbito algo más amplio de la programación. Esta persona puede contribuir a la visión general del proyecto más a nivel de aplicación que a nivel de componentes o en las tareas de programación individuales [2].
* **Modelo de Vistas de Arquitectura 4+1:** es un modelo de vistas diseñado por el profesor Philippe Kruchten y que encaja con el estándar “IEEE 1471-2000” (Recommended Practice for Architecture Description of Software-Intensive Systems [5]) que se utiliza para describir la arquitectura de un sistema software intensivo basado en el uso de múltiples puntos de vista [3]
* **Requisitos Funcionales:** son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones [4].
* **Requisitos no funcionales:** representan características generales y restricciones de la aplicación o sistema que se esté desarrollando [5].
* **Usuario final:** es la persona que utilizará un software o un hardware luego de que este ha sido desarrollado, comercializado e instalado [6].
* **UML:** un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento [7].
* **Diagrama de clases:** Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento [8].
* **Diagrama de datos: Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos [9].**
* **Diagrama de secuencia: Un diagrama de secuencias muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo, en el cual se indicaran los módulos o clases que formaran parte del programa y las llamadas que se hacen cada uno de ellos para realizar una tarea determinada, por esta razón permite observar la perspectiva cronológica de las interacciones [10].**
* **Diagrama de comunicación: Un diagrama de Comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los diagramas de Comunicación representan una combinación de información tomada desde el diagrama de Clases, Secuencia, y Diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema [11].**
* **Diagrama de componentes: muestra los elementos de un diseño de un sistema de software. Un diagrama de componentes permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces [12].**
* **Diagrama de paquetes: El objetivo de estos diagramas es obtener una visión más clara del sistema de información orientado a objetos, organizándolo en subsistemas, agrupando los elementos del análisis, diseño o construcción y detallando las relaciones de dependencia entre ellos. El mecanismo de agrupación se denomina *Paquete* [13].**
* **Diagrama de actividades: se usan para describir las actividades de negocios y la funcionalidad de los sistemas de software [14].**
* **Diagrama de estados: Se especializa en mostrar transiciones entre diversos objetos [15].**
* **Diagrama de despliegue: n diagrama de despliegue se encuentra dentro de la familia de diagramas estructurales y describe un aspecto del sistema en sí. En este caso, el diagrama de despliegue describe el despliegue físico de información generada por el programa de software en los componentes de hardware [16].**
* **Diagrama de casos de uso: representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso) [17].**
* **Concurrencia: es la tendencia de las cosas a producirse al mismo tiempo en un sistema. En lo que se refiere a temas de concurrencia en sistemas de software, hay dos aspectos importantes en general: poder detectar y responder a sucesos externos que se producen en un orden aleatorio y garantizar que estos sucesos se responden en un intervalo requerido mínimo [18].**
* ***CRUD:*** Responde a las siglas en inglés (Create, Read, Update and Delete), Crear, Leer, Actualizar y Eliminar. Se refiere a las operaciones que se pueden realizar en un almacén de datos [19].
* **Arquitectura de Software: las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos [20]**
* **Lógica de Negocio: es la parte de un sistema que se encarga de codificar las reglas de negocio del mundo real que determinan cómo la información puede ser creada, mostrada y cambiada. En programación es una de las capas del modelo MVC Modelo–vista–controlador separando así la complejidad del desarrollo en capas independientes [21].**
* **Paquetes lógicos**
* **Patrones de Diseño**
* **Rol**
* ***Factory Method***
* ***GUI***
* ***Abstract Factory***
* ***Scope***
* ***Hardware***
* ***Decorator***
* ***Template***
* **Escalabilidad**
* **Reutilización**
* **Facade**
* **Arquitectura Cliente-Servidor**
* **Arquitectura MVC**
* ***Core***
* ***HTTP***
* **Servidor Web**
* ***RESTful***
* ***DBMS***
* ***Framework***
* ***Hibernate***
* ***DAO***
* ***SQL***
* ***TCP/IP***
* **Localhost**

**Referencias**

[1] "Significado de Stakeholder", *Significados.com*. [Online]. Available: https://www.significados.com/stakeholder/. [Accessed: 29- Nov- 2017].

[2] "Rol: Desarrollador", *Geoservice.igac.gov.co*. [Online]. Available: http://geoservice.igac.gov.co/mds/igac/ciaf/roles/desarrollador\_igac.html. [Accessed: 29- Nov- 2017].

[3] R. Moya, "Modelo “4+1” vistas de Kruchten (para Dummies)", *Jarroba*, 2017. [Online]. Available: https://jarroba.com/modelo-41-vistas-de-kruchten-para-dummies/. [Accessed: 29- Nov- 2017].

[4] "Requerimientos funcionales: Ejemplos", *Pmoinformatica.com*, 2017. [Online]. Available: http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html. [Accessed: 29- Nov- 2017].

[5] "Requerimientos no funcionales: Ejemplos", *Pmoinformatica.com*, 2017. [Online]. Available: http://www.pmoinformatica.com/2017/02/requerimientos-no-funcionales-ejemplos.html. [Accessed: 29- Nov- 2017].

[6] L. Alegsa, "Definición de usuario final (end user)", *Alegsa.com.ar*, 2015. [Online]. Available: http://www.alegsa.com.ar/Dic/usuario\_final.php. [Accessed: 29- Nov- 2017].

[7] "Qué es el lenguaje unificado de modelado (UML)", *Lucidchart*. [Online]. Available: https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml. [Accessed: 29- Nov- 2017].

[8] P. Salinas, "Modelo de Clases", 2017. [Online]. Available: https://users.dcc.uchile.cl/~psalinas/uml/modelo.htm. [Accessed: 29- Nov- 2017].

[9] "Qué es un modelo de base de datos", *Lucidchart*. [Online]. Available: https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-modelo-de-base-de-datos. [Accessed: 29- Nov- 2017].

[10] K. Cevallos, "UML: Diagrama de Secuencia", *INGENIERÍA DEL SOFTWARE*, 2015. [Online]. Available: https://ingsotfwarekarlacevallos.wordpress.com/2015/07/07/uml-diagrama-de-secuencia/. [Accessed: 29- Nov- 2017].

[11] "Diagrama de comunicación", *Es.wikipedia.org*. [Online]. Available: https://es.wikipedia.org/wiki/Diagrama\_de\_comunicaci%C3%B3n. [Accessed: 29- Nov- 2017].

[12] Diagramas de componentes de UML: Referencia", *Msdn.microsoft.com*. [Online]. Available: https://msdn.microsoft.com/es-es/library/dd409390.aspx. [Accessed: 29- Nov- 2017].

[13] M. Cillero, "Diagrama de Paquetes - manuel.cillero.es", *manuel.cillero.es*. [Online]. Available: https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-paquetes/. [Accessed: 29- Nov- 2017].

[14] "Diagrama de actividades - UML", *Lucidchart*. [Online]. Available: https://www.lucidchart.com/pages/es/diagrama-de-actividades-uml. [Accessed: 29- Nov- 2017].

[15] "Diagrama de máquina de estados - UML", *Lucidchart*. [Online]. Available: https://www.lucidchart.com/pages/es/diagrama-de-m%C3%A1quina-de-estados-uml. [Accessed: 29- Nov- 2017].

[16] "Tutorial de diagramas de implementación", *Lucidchart*. [Online]. Available: https://www.lucidchart.com/pages/es/diagrama-de-despliegue. [Accessed: 29- Nov- 2017].

[17] P. Salinas, "Tutorial de UML - Casos de Uso", *Users.dcc.uchile.cl*. [Online]. Available: https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html. [Accessed: 29- Nov- 2017].

[18] "Concepto: Concurrencia", *Cgrw01.cgr.go.cr*. [Online]. Available: http://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base\_rup/guidances/concepts/concurrency\_EE2E011A.html. [Accessed: 29- Nov- 2017].

[19] "Glosario informático - Definición de términos informáticos", *Glosarioit.com*. [Online]. Available: http://www.glosarioit.com/#!CRUD. [Accessed: 29- Nov- 2017].

[20] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, 2nd Edition, Addison Wesley, 2003

[21] "Lógica de negocio", *Es.wikipedia.org*. [Online]. Available: https://es.wikipedia.org/wiki/L%C3%B3gica\_de\_negocio. [Accessed: 29- Nov- 2017].