

Metodología de la programación II

Practica VIII

Emiliano Salvatori

Octubre 2019

1. Practica VIII

Framework

¿Por qué la metodología de programación se considera como clave para el éxito o el fracaso de un proyecto?

Las Metodologías de Programación son decisivas en el éxito o fracaso de un proyecto. En general ponen en práctica una serie de procesos comunes, que son buenas prácticas para lograr los objetivos de negocio, costos, funcionalidad, sencillez, etc.

¿Qué consecuencias puede traer la mala elección de una metodolgia de programación?

La elección de una metodología inadecuada o su mala aplicación pueden conducir a que el proyecto no llegue a su fin.

¿Por qué actualmente se considera importante la utilización de metodologías ágiles?

Hoy en día, con un escenario en el que los requisitos cambian habitualmente, es donde surge la necesidad de conocimiento sobre las Metodologías Ágiles, más ligeras y versátiles.

Es muy importante escoger una buena metodología para tener éxito en el proyecto, lo que implica muchas más cosas que sacar un producto a tiempo: supone que el cliente esté satisfecho y que el equipo trabaje cómodo. Actualmente, las Metodologías Ágiles están en auge dentro del desarrollo software.

Investigue las características del Framework RUP (Rational Unified Process), cuáles son sus principios y aspectos clave y qué tipos de ciclos de vida presenta?

El Proceso Unificado de Rational o RUP (por sus siglas en inglés de Rational Unified Process) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM.¹ Junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software, también desarrollado por Rational, que incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo con las necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el Rational Unified Process, que se vendiera como producto independiente.

En lo referente a "iterativo e incremental", significa que el proyecto se divide en varios ciclos de vida (llamadas iteraciones) que deben dar como resultado un ejecutable. Por cada una de las iteraciones se va agregando requerimientos y sobre todo valor al cliente; por este motivo es incremental.¹

¿Qué ventajas considera que posee el modelo incremental por sobre el de cascada?

Las ventajas son que el modelo incremental pone mayor énfasis para controlar la complejidad y los riesgos en un proyecto de Software, donde en cada fase se agrega una nueva funcionalidad, por lo que permite tener entregables funcionando de forma mucho más rápida y segura, atendiendo la problemática del cambio en un proyecto que la que podría proporcionar el modelo de cascada.

¹https://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

Investigue las características de UML (Lenguaje Unificado de Modelado) ¿Qué tipos de diagramas estandariza?

El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Existen dos clases principales de tipos de diagramas: diagramas estructurales y diagramas de comportamiento. Estos últimos incluyen varios que representan diferentes aspectos de las interacciones. Estos diagramas pueden ser categorizados jerárquicamente.

Estructurales

Muestran la estructura estática de los objetos en un sistema. Los cuales pueden ser:

- **Diagrama de clases** Los diagramas de clase son, sin duda, el tipo de diagrama UML más utilizado. Es el bloque de construcción principal de cualquier solución orientada a objetos. Muestra las clases en un sistema, atributos y operaciones de cada clase y la relación entre cada clase. En la mayoría de las herramientas de modelado, una clase tiene tres partes, nombre en la parte superior, atributos en el centro y operaciones o métodos en la parte inferior. En sistemas grandes con muchas clases relacionadas, las clases se agrupan para crear diagramas de clases. Las Diferentes relaciones entre las clases se muestran por diferentes tipos de flechas.
- **Diagrama de componentes** Un diagrama de componentes muestra la relación estructural de los componentes de un sistema de software. Estos se utilizan principalmente cuando se trabaja con sistemas complejos que tienen muchos componentes. Los componentes se comunican entre sí mediante interfaces. Las interfaces se enlazan mediante conectores.
- **Diagrama de despliegue** Un diagrama de despliegue muestra el hardware de su sistema y el software de ese hardware. Los diagramas de implementación son útiles cuando la solución de software se despliega en varios equipos, cada uno con una configuración única.
- **Diagrama de objetos** Los diagramas de objetos, a veces denominados diagramas de instancia, son muy similares a los diagramas de clases. Al igual que los diagramas de clases, también muestran la relación entre los objetos, pero usan ejemplos del mundo real. Se utilizan para mostrar cómo se verá un sistema en un momento dado. Debido a que hay datos disponibles en los objetos, a menudo se utilizan para explicar relaciones complejas entre objetos.
- **Diagrama de paquetes** Como su nombre indica, un diagrama de paquetes muestra las dependencias entre diferentes paquetes de un sistema.
- **Diagrama de perfiles** El diagrama de perfil es un nuevo tipo de diagrama introducido en UML 2. Este es un tipo de diagrama que se utiliza muy raramente en cualquier especificación.
- **Diagrama de estructura compuesta** Los diagramas de estructura compuesta se utilizan para mostrar la estructura interna de una clase.ⁱ

De comportamiento

Muestran el comportamiento dinámico de los objetos en el sistema. Los cuales pueden clasificarse como:

- **Diagrama de actividades** Los diagramas de actividad representan los flujos de trabajo de forma gráfica. Pueden utilizarse para describir el flujo de trabajo empresarial o el flujo de trabajo operativo de cualquier componente de un sistema. A veces, los diagramas de actividad se utilizan como una alternativa a los diagramas de máquina del estado.
- **Diagrama de casos de uso** Como el tipo de diagrama de diagramas UML más conocido, los diagramas de casos de uso ofrecen una visión general de los actores involucrados en un sistema, las diferentes funciones que necesitan esos actores y cómo interactúan estas diferentes funciones. Es un gran punto de partida para cualquier discusión del proyecto, ya que se pueden identificar fácilmente los principales actores involucrados y los principales procesos del sistema.

- **Diagrama de máquina de estados** Los diagramas de máquina de estado son similares a los diagramas de actividad, aunque las anotaciones y el uso cambian un poco. En algún momento se conocen como diagramas de estados o diagramas de diagramas de estado también. Estos son muy útiles para describir el comportamiento de los objetos que actúan de manera diferente de acuerdo con el estado en que se encuentran en el momento.
- **Diagrama de interacción** Los diagramas de interacción incluyen distintos tipos de diagramas:
- **Diagrama de secuencia** Los diagramas de secuencia en UML muestran cómo los objetos interactúan entre sí y el orden en que se producen esas interacciones. Es importante tener en cuenta que muestran las interacciones para un escenario en particular. Los procesos se representan verticalmente y las interacciones se muestran como flechas. Los diagramas de secuencia de UML forman parte de un modelo UML y solo existen dentro de los proyectos de modelado UML.
- **Diagrama de comunicación** El diagrama de comunicación se llamó diagrama de colaboración en UML 1. Es similar a los diagramas de secuencia, pero el foco está en los mensajes pasados entre objetos.
- **Diagrama de tiempos** Los diagramas de sincronización son muy similares a los diagramas de secuencia. Representan el comportamiento de los objetos en un marco de tiempo dado. Si es solo un objeto, el diagrama es directo, pero si hay más de un objeto involucrado, también se pueden usar para mostrar interacciones de objetos durante ese período de tiempo.
- **Diagrama global de interacciones** Los diagramas generales o globales de interacción son muy similares a los diagramas de actividad. Mientras que los diagramas de actividad muestran una secuencia de procesos, los diagramas de interacción muestran una secuencia de diagramas de interacción. En términos simples, pueden llamarse una colección de diagramas de interacción y el orden en que suceden. Como se mencionó anteriormente, hay siete tipos de diagramas de interacción, por lo que cualquiera de ellos puede ser un nodo en un diagrama de vista general de interacción.²

¿Qué es un Framework, desde el punto de vista del desarrollador del software?

Un Framework es un Conjunto estandarizado de conceptos, prácticas y criterios para enfrentar y resolver problemas de características similares. Palabra del idioma inglés que significa "marco de trabajo". Consiste en modelos, reglas, herramientas para asistir al proceso de desarrollo.

Es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

Mencione las características de los frameworks que a usted le resulten relevantes para su uso

Dado las características del trabajo que realizo diariamente, es de uso intensivo el tipo de Framework denomina *iterativo* el cual tiene las siguientes características:

Permite desarrollar modelos de aplicaciones de software dejando ver su funcionalidad básica, sin necesariamente incluir toda la lógica o características del modelo terminado. Permite al cliente evaluar el producto en forma temprana e interactuar con los desarrolladores para saber si se está cumpliendo con las expectativas y funcionalidades acordadas.

Tipos de prototipos:

- Homogéneos
- Heterogéneos
- Desechables
- Incrementales

¿Cuáles son las ventajas y desventajas de usar un Framework?

Las ventajas de utilizar un Framework son:

- El framework le proporciona al programador un esqueleto que hay que "rellenar".
- Facilita los desarrollos colaborativos. Entender y modificar código fuente de otro programador es difícil, por lo tanto, todo lo que sea definir y estandarizar permite ahorrar tiempo y trabajo.
- El desarrollo con framework es menos propenso a sufrir errores.

²https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado

- Compatibilidad de Lenguajes.
- Transparencia de proyectos de plataforma a plataforma.
- Portabilidad de Arquitectura.
- Integración con múltiples dispositivos.
- Desarrollo de aplicaciones de manera más sencilla, ya que cuenta con los componentes necesarios incluidos.
- Reutilización de Código.
- Manejo de Política de diseño uniforme y organizada.

¿Cuál es la arquitectura utilizada en la mayoría de los Frameworks? ¿En qué consiste dicha arquitectura?

La más utilizada en casi todos los frameworks es conocida como MVC (Modelo, Vista, Controlador). Esta arquitectura divide el desarrollo en tres grandes partes:

- **Modelo:** Son los datos de la aplicación y su reglamentación.
- **Vista:** Es la presentación de los datos.
- **Controlador:** Procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

Mencione otros Frameworks que no hayan sido nombrados y describa en qué situaciones pueden utilizarse.

Algunos de los Frameworks utilizados para desarrollo Web pueden ser:

- **Angular.js:** Framework basado en JavaScript.
- **React:** También basado en JavaScript, el cual permite desarrollar aplicaciones móviles para IOS y Android. Fue liberado por Facebook.
- **Ruby on rails:** Framework de tipo MVC basado en Ruby, orientado al desarrollo de aplicaciones web.
- **Django:** Framework Python que promueve el desarrollo rápido y el diseño limpio.

Algunos de los Frameworks utilizados para desarrollo en lenguaje Java:

- **Spring MVC:** Es un framework de gran escala de Java para las aplicaciones de Internet, ofreciendo poderosa herramienta para el desarrollo de aplicaciones web, así como para proyectos de seguridad. Ofrece una amplia gama de servicios: API REST, servicios web SOAP, seguridad, etc.
- **Hibernate:** El framework de mapeo de Java rompe con la impedancia del objeto-relacional y con los problemas de desajuste sustituyendo accesos a bases de datos de alto nivel. Hibernate está equipado con grandes características que ayudan a los desarrolladores back-end a acceder a los datos. Se trata de un framework ORM avanzado que permite realizar la operación en la base de datos de objetos Java (entidades).
- **Vaadin:** Utilizando un enfoque basado en componentes conocidos, quita peso a los desarrolladores a la hora de comunicar los cambios realizados al navegador. Un amplio conjunto de componentes de interfaz de usuario, junto con diversos widgets y controles, permite a los programadores crear aplicaciones más que interesantes en muy poco tiempo.

Por otro lado el **Eclipse Modeling Framework (EMF)**, se tiene en cuenta cuando se busca facilitar la generación de código con el objetivo de desarrollar aplicaciones y herramientas orientadas a un modelo estructurado de datos.