3. Plataformas de Ingenierías de Software

3.1. Introducción y objetivos

En este tema se introduce el concepto de *application lifecycle management*(ALM), frente al *software development life cycle*(SDLC) y se justifica que abordaremos las plataformas de ingeniería del software con una perspectiva amplia, como aquellas herramientas o combinaciones de estas, que dan soporte transversal a los procesos de ALM.

Por otro lado, se introduce el concepto de herramienta *computer-aided software engineering* (CASE)como una visión específica de plataformas de ingeniería del software.

Posteriormente, veremos las principales ventajas del uso de plataformas de ingeniería del software, en ambas perspectivas, y enumeraremos las principales disponibles con sus características y funcionalidades.

Con el estudio de este tema, pretendemos alcanzar los siguientes objetivos:

* Entender el contexto de las plataformas de ingeniería del software ALM y CASE.
* Comprender el valor y la utilidad que tienen las de ingeniería del software ALM y CASE.
* Conocer algunas plataformas de ingeniería del software ALM y CASE.

3.2. Contexto de las plataformas de ingeniería de software

Las organizaciones son cada vez más modernas y dependen del software en muchas formas. Los procesos de negocio están, a menudo, implementados en un flujo digital en el cual, si no disponemos de un software de apoyo, se pueden producir problemas.

Todo desarrollo de software incluye diferentes actividades que son ejecutadas por personas que juegan roles específicos y es necesario que se lleve a cabo de forma colaborativa para que los proyectos se realicen con éxito.

La gestión del ciclo de vida de las aplicaciones, *application lifecycle management*(ALM), se focaliza en la forma en la que se puede gestionar el ciclo completo de vida de la construcción de un software: cubre todo el proceso desde la idea hasta su puesta en marcha, experimentación, implementación, despliegue en sistemas y soporte. También incluye el seguimiento de modificaciones y documentación a lo largo de su uso. Engloba diferentes disciplinas que hasta ahora se consideraban de forma asilada, pero que, conjuntamente y con un buen proceso de ALM, se puede llevar a cabo un desarrollo más rápido, más eficiente en coste y con una mayor calidad.

En algunas ocasiones, se confunde con el ciclo de vida de desarrollo de software (*software development life cycle,*SDLC),definido como los procesos o etapas involucradas en la creación de un nuevo sistema de información o la modificación de un sistema existente y las diversas metodologías utilizadas en el proceso. En otras palabras, son todas las actividades involucradas en el desarrollo del software, los procesos involucrados en llevarlo de la idea a la realidad (software real).

SDLC y ALM son dos conceptos que no son completamente independientes. Técnicamente, implican un cierto lapso de las diferentes etapas de una aplicación entre su inicio y el final de su vida útil. En realidad, el SDLC es un subconjunto de ALM.

En las organizaciones de desarrollo de producto de software es muy habitual que, en función de las personas que la componen, se dé prioridad a algunos aspectos frente a otros en el desarrollo de un producto. Habitualmente se dan las siguientes perspectivas:

* Perspectiva del SDLC: es, posiblemente, la forma más común de observar un ALM, quizá porque es la parte central y ocupa un largo espacio de tiempo.
* Perspectiva de administración de servicios u operaciones: las operaciones, a menudo, se han considerado por separado en el ciclo de vida de un producto de software. Esto ha dado como resultado que este aspecto tenga su propio visión de ALM y los problemas en esta área.
* Perspectiva de la gestión de la cartera de aplicaciones (APM): seguramente, debido a la brecha entre negocios y TI (tecnología de la información), hemos visto muchas organizaciones con una estrategia ALM de cartera en la que el desarrollo es solo una pequeña parte. Desde una perspectiva comercial, el enfoque ha estado en cómo manejar la cartera y no en todo el proceso ALM.

Con todo ello, una vista unificada, que ya existe en algunas organizaciones, proporciona una forma holística de enfrentar el proceso ALM, al tener en cuenta las tres perspectivas anteriores. Esta es la única manera de tomar el control y optimizar el ALM. Para un director de información (CIO) de una organización es esencial tener esta vista todo el tiempo para que cada una de las perspectivas se vea reforzada.

Los tres pilares fundamentales de ALM son:

* Trazabilidad: en ocasiones se han dejado de llevar a cabo actualizaciones de un producto software en producción al no tener suficiente trazabilidad en sus componentes. Resultaba demasiado caro ese mantenimiento debido a los efectos inesperados que incluso un pequeño cambio podría tener: el efecto de un pequeño cambio en una parte del código podría afectar a otra parte por sorpresa, ya que una mala trazabilidad significaba que no se tenía forma de ver dicha conexión previamente. Debe haber una forma de rastrear los requisitos hasta el código entregado, a través de modelos de arquitectura, diseño de modelos, *scripts*, pruebas unitarias, casos de prueba, etc., no solo para implementar correcciones de errores, sino también para demostrar que el producto contiene las cosas que desde negocio se requerían o para lograr el cumplimiento interno y externo de las normas y reglamentos.
* Automatización de procesos de alto nivel: el siguiente pilar de ALM es la automatización de procesos de alto nivel. Todas las organizaciones tienen procesos, aunque sea de forma implícita. En muchas de ellas, el paso de un proceso a otro (por ejemplo, de análisis a diseño o entre implementación y pruebas) se hace manualmente. ALM enfatiza la importancia de automatizar estas tareas para un proceso más efectivo, que consuma menos tiempo y con menos errores.
* Visibilidad del progreso de los esfuerzos de desarrollo: el tercer y último pilar de ALM es brindar visibilidad sobre el progreso de los esfuerzos de desarrollo. Muchos gerentes y las partes interesadas tienen una visibilidad limitada del progreso de los proyectos de desarrollo y, a menudo, viene de las reuniones del grupo directivo, durante las cuales el director del proyecto revisa la situación actual. Si se desea tener un proceso efectivo, es necesario asegurar la visibilidad. También es fundamental que los miembros del equipo de desarrollo dispongan de esa visibilidad, ya que, de forma demasiado frecuente, la tienen sesgada. Esto se debe, habitualmente, al hecho de que la elaboración de informes es difícil y puede implicar mucho trabajo manual tomando demasiado tiempo y esfuerzo para producirlos, especialmente cuando tiene información en muchos repositorios.

En este capítulo abordaremos, por un lado, las plataformas de ingeniería del software con una perspectiva amplia, como aquellas herramientas o combinaciones de estas que dan soporte transversal a los procesos de ALM.

Por otro lado, también cubriremos una visión mucho más concreta, enfocada en plataformas *computer-aided software engineering* (CASE), que es un paquete de software que brinda soporte para el diseño e implementación de sistemas de información.

Las herramientas CASE son un conjunto de programas de aplicación de software que se utilizan para automatizar las actividades de SDLC. Son utilizadas por gerentes de proyectos de software, analistas e ingenieros para desarrollar sistemas de software. Hay varias herramientas CASE disponibles para simplificar varias etapas del ciclo de vida del desarrollo de software, como herramientas de análisis, herramientas de diseño, herramientas de gestión de proyectos, herramientas de gestión de bases de datos, herramientas de documentación, entre otras.

### **3.3. Valor y utilidad de las plataformas de ingeniería de software**

Desde la perspectiva de plataformas de ingeniería del software como aquellas que, de forma transversal, pueden dar soporte a un ALM en una perspectiva amplia, su valor es dar soporte a las siguientes funcionalidades:

* **Gestión de requerimientos:** las herramientas de ALM deben tener una funcionalidad que lo ayude a dar sentido a sus requisitos e, idealmente, las herramientas deben poder adaptarse a su metodología y procesos, y no al revés. Algunas herramientas de ALM admiten la captura y gestión de casos de uso del sistema como parte del proceso de definición de requisitos. Los **casos de uso** se pueden almacenar en el sistema junto con sus escenarios asociados y enlaces a los demás requisitos que define el caso de uso. Dependiendo de sus necesidades (si tiene procesos complicados de varios pasos o necesidades simples de ingreso de datos CRUD [crear, leer, actualizar, borrar]), esta puede ser una característica importante.
* **Estimación y planificación:** las herramientas de ALM generalmente brindan funcionalidad para ayudarlo a estimar y planificar sus proyectos. Algunas herramientas están diseñadas para administrar proyectos en cascada tradicionales, otras están diseñadas solo para proyectos de desarrollo ágiles y otras pueden manejar ambos. Dependiendo del nivel de planificación que precise, es posible que necesite una herramienta ALM que le permita **estimar tareas técnicas** individuales que luego se agregan de nuevo contra los requisitos originales que se especificaron. Si tiene requisitos de revisión y supervisión más complejos, es posible que necesite una funcionalidad de flujo de trabajo que garantice que todas las tareas asignadas sigan un conjunto específico de pasos con revisiones y controles de calidad.
* **Desarrollo de software:** aunque normalmente el acto real de desarrollar y escribir el código no se considera parte de ALM en sí mismo, por lo general necesita tener la funcionalidad de **administración de código fuente** integrada estrechamente dentro de la suite ALM. La mayoría de las herramientas de ALM ofrecerán una funcionalidad de administración de código fuente integrada: Algunas herramientas de ALM proporcionarán soluciones de gestión de código fuente patentadas, otras utilizarán herramientas estándar de la industria, como Git o Subversion. Idealmente, la herramienta ALM debería proporcionar flexibilidad para admitir diferentes modelos de bifurcación y fusión. Idealmente, aunque la escritura de código no es estrictamente parte de la suite ALM en sí, las mejores herramientas ALM del mercado brindan integración con las herramientas utilizadas por los desarrolladores para escribir el código.
* **Pruebas y garantía de calidad:** uno de los atributos clave que distingue a las suites ALM de las meras herramientas de gestión de proyectos o los sistemas de seguimiento de problemas es que incluyen la parte de **control de calidad** del proceso de desarrollo de software. Como mínimo, deberían ofrecer una gestión básica de casos de prueba. La herramienta ALM debería permitirle crear y administrar sus casos de prueba en carpetas con capacidades de clasificación y filtrado. Debería permitir a los usuarios mostrar diferentes columnas en su propia vista personalizada y permitir que el administrador defina propiedades y listas personalizadas que se pueden usar en el proyecto. Algunas herramientas de ALM brindan la capacidad de definir pasos de prueba y parámetros específicos para los casos de prueba, mientras que otras solo pueden ofrecer un artefacto de caso de prueba simple.
* **Despliegue**: aunque la compilación y el empaquetado del software que se está desarrollando a menudo está fuera del alcance de las propias herramientas de ALM, la mayoría de las herramientas de ALM le permitirán, como mínimo, integrarse con **servidores de integración continua** (también conocidos como CI) como Jenkins. Según sus necesidades, debe asegurarse de que la herramienta ALM que elija pueda integrarse con los diferentes servidores de CI que usan sus equipos de desarrollo y que los informes dentro de la herramienta ALM puedan fusionar la información de otras fuentes con la información de compilación. De esa manera, puede ver todos los cambios en cada compilación, qué funciones se agregaron y qué problemas se resolvieron.
* **Desarrollo y operaciones (DevOps):** es la combinación de tareas realizadas por los equipos de desarrollo de aplicaciones y operaciones de sistemas de una empresa en un todo coherente. Anteriormente, sus desarrolladores necesitarían la ayuda del grupo de TI para aprovisionar diferentes entornos (desarrollo, prueba, preparación, etc.) y probar la aplicación por etapas en cada entorno. Ahora, con más aplicaciones alojadas en la nube y ciclos de lanzamiento medidos en días, no en meses, no puede desarrollar y probar de forma aislada. Deberá probar **nuevas actualizaciones** con datos reales en entornos listos para producción, DevOps es la clave para que esto sea una realidad.
* **Soporte y mantenimiento**: lamentablemente, la parte que más se pasa por alto del ciclo de vida del software es el soporte y el mantenimiento del producto después de su lanzamiento. Tradicionalmente, esto se hacía justo al final, cuando el equipo de desarrollo entregaba el producto terminado al equipo de soporte y luego se lavaba las manos del sistema hasta que se desarrollaba la siguiente versión. Con el paso al desarrollo ágil, la entrega continua y DevOps, el soporte ahora es una **parte integral del proceso.** La atención al cliente es el mecanismo de retroalimentación que guía las nuevas historias de usuario en el próximo sprint. Por lo tanto, no olvide asegurarse de que la suite ALM tenga una capacidad de atención al cliente o, al menos, pueda integrarse con otras herramientas de soporte al cliente o de mesa de ayuda.
* **Gestión de proyectos y carteras:** dado que las herramientas ALM brindan una funcionalidad tan rica y poderosa, es importante asegurarse de que también brinden buenas herramientas y funciones para consumir la información y poder generar informes en todo el sistema con información resumida y fusionada para permitir la toma de decisiones. Una característica común en muchas buenas herramientas de ALM es la capacidad de **mostrar paneles de control** o gráficos y tablas de acceso común que cubren las métricas clave del proyecto. Además, si se necesita administrar programas más grandes compuestos por varios proyectos diferentes, cada uno con sus propios requisitos, características e hitos, deberá asegurarse de que la herramienta ALM tenga la funcionalidad de administración de cartera de proyectos (PPM). Tradicionalmente, las herramientas de PPM eran una categoría separada, pero cada vez más proveedores de ALM incluyen funciones de PPM en sus ofertas principales. Pueden permitirle visualizar el estado actual de sus proyectos y, en algunos de los ejemplos más potentes, administrar las dependencias y tareas en los diferentes proyectos.
* **Colaboración y comunicación:** es altamente ineficiente depender del correo electrónico, mensajes de texto o, peor aún, reuniones grupales, para facilitar la comunicación de los miembros del equipo en tiempo real. La información puede estar fuera de contexto y, posteriormente, no existe un **registro confiable** del intercambio, lo que deja a otros miembros del equipo en el vacío sin una buena forma de descubrir lo que está sucediendo.

Desde la perspectiva de herramientas CASE, su uso acelera el desarrollo del proyecto para producir el resultado deseado y ayuda a descubrir fallas antes de continuar con la siguiente etapa en el desarrollo de software. Dentro de esta categoría sus principales funcionalidades son:

* **Elaboración de diagramas:** ayuda en las representaciones esquemáticas y gráficas de los datos y procesos del sistema. Representa los elementos del sistema, el flujo de control y el flujo de datos entre diferentes componentes de software y la estructura del sistema en forma gráfica.
* **Conceptualización de necesidades:** ayuda a comprender los requisitos y las relaciones e interdependencias.
* **Análisis de modelos:** automatización en el análisis de la información modelada para buscar inconsistencias, redundancias u omisiones erróneas.
* **Repositorio central:** proporciona el único punto de almacenamiento para diagramas de datos, informes y documentos relacionados con la gestión de proyectos.
* **Generación de documentación:** ayuda a generar documentación técnica y de usuario según los estándares. Crea documentos para usuarios técnicos y usuarios finales.
* **Generación de código:** ayuda en la generación automática de código a partir de los modelos.

### **3.4. Plataformas ALM**

Las principales plataformas para el control de versiones son:

A screenshot of a phone

Description automatically generated

Tabla 1. Plataformas ALM. Fuente: elaboración propia.

**Atlassian Jira**

Al combinar algunos productos de la suite **Atlassian** como Jira Software, Confluence, Stash (Bitbucket Server) y Bamboo, se diseñó una herramienta ALM competente y de servicio completo.

* El software Jira se utiliza para planificar y realizar un seguimiento de los proyectos y sus problemas correspondientes.
* Confluence es una herramienta de colaboración para la gestión de documentos y conocimientos.
* Bitbucket Server es un repositorio de código con tecnología Git que colabora mediante solicitudes y comentarios en línea.
* Bamboo Server se utiliza para la integración, la creación y las pruebas programadas, la implementación y la entrega.

**CA Agile Central**

Anteriormente conocida como Rally, es la principal herramienta ALM para empresas que utilizan métodos Agile. Sus funciones de gestión de proyectos, incluida la planificación de recursos, están diseñadas con Agile en mente. También es una excelente herramienta ALM para **realizar pruebas.** Proporciona funcionalidad de colaboración a través de Flowdock, que tiene funciones de chat y correo electrónico. Es compatible con Scaled Agile Framework (SAFe) y ofrece análisis y métricas específicas de Agile basadas en el índice de rendimiento de desarrollo de software. Proporciona opciones locales y SaaS.

CA Agile Central se utiliza para **desarrollar y entregar** software preciso con alta calidad y velocidad con una plataforma eficiente para soluciones ágiles.

* Ofrece una plataforma ALM basada en la web a la que se puede acceder desde un navegador como Chrome o Safari, etc. a través de un sistema operativo como Windows o Linux.
* Esta herramienta es utilizada principalmente por organizaciones de todos los tamaños, como pequeñas, medianas o grandes, que desarrollan aplicaciones con metodología ágil.
* Es una plataforma de nivel empresarial que se utiliza para escalar metodologías de desarrollo ágiles en un proyecto.
* Se pueden medir las métricas centrales de proyectos en tiempo real, como las métricas de rendimiento, la productividad, la calidad y la apertura de las aplicaciones.

**CodeBeamer**

Es una plataforma de ALM para el desarrollo avanzado de productos y software. La plataforma abierta **amplía las funcionalidades** de ALM con capacidades de configuración de línea de productos y proporciona una capacidad de configuración única para procesos complejos.

CodeBeamer proporciona gestión de proyectos, tareas, requisitos, pruebas, cambios, configuración, creación, conocimientos y documentos en un único entorno seguro. Permite que el desarrollo de software y hardware sea más **colaborativo, transparente y productivo.**

CodeBeamer ofrece **flujos de trabajo digitales** únicos para el desarrollo colaborativo, la eficiencia en la ingeniería de líneas de productos y el cumplimiento normativo. Es posible conectar todas las herramientas de desarrollo para proporcionar a sus equipos un centro de desarrollo central y se puede adaptar fácilmente la solución a las necesidades especiales de ingeniería y automatizar el control de procesos para los requisitos reglamentarios.

Alguna de las funcionalidades de CodeBeamer son:

* Gestión de **requerimientos:** convierta los conocimientos del mercado en productos de calidad más rápido que nunca. Obtenga claridad y alinee a todos los colaboradores para reducir costos y tiempos de ciclo. Seguimiento de elementos y acciones a lo largo de la entrega.
* **Desarrollo de software**: reduzca la complejidad e incorpore calidad a sus productos de software. Garantice una colaboración fluida entre las disciplinas de innovación de hardware, software y servicios. Use procesos maduros, ya sea que esté trabajando con en cascada, ágiles (a escala), híbrido o un método personalizado.
* Garantía de **calidad y pruebas:** integre la gestión de calidad en sus procesos de desarrollo y genere confianza en sus productos. Rastree y gestione las actividades de control de calidad a lo largo del ciclo de vida y acelere la verificación del producto.

A screenshot of a computer

Description automatically generated

Figura 1. Vista de CodeBeamer. Fuente: Intland Software, s. f.

**IBM**

Ofrece varios productos para su solución ALM. Algunos de estos son: CLM para la gestión colaborativa del ciclo de vida, la herramienta Engineering Workflow Management para gestionar flujos de trabajo, servicio en la nube de entrega continua para automatizar compilaciones y pruebas de software, base de administración del ciclo de vida de ingeniería para administrar requisitos, calidad, cambios, planes y seguimiento de proyectos, entre otros.

Los productos de IBM son compatibles con las metodologías cascada y ágiles y se integran con IBM Tivoli para admitir la **integración continua.** Estas características hacen que los productos de IBM sean una buena opción para las organizaciones que necesitan integrar diversos procesos. El producto es más antiguo con mucho soporte heredado, pero IBM lo ha mantenido actualizado, alineándose con las tendencias emergentes de ALM. IBM Rational CLM es un sistema ALM que incluye una gran selección de aplicaciones ALM que están integradas entre sí.

* Rational CLM es una combinación de IBM Rational Team Concert, IBM Rational DOORS Next Generation e IBM Rational Quality Manager, lo que la convierte en una **solución ALM completa.**
* La integración de los productos anteriores con RCLM ofrece gestión de requisitos, planificación de proyectos en tiempo real, gestión de calidad, gestión de cambios, trazabilidad del ciclo de vida, etc.
* IBM Rational Team Concert se utiliza para planificar, gestionar y rastrear el trabajo del proyecto.
* IBM Rational DOORS Next Generation se utiliza para gestionar requisitos como definirlos, analizarlos y gestionarlos.
* IBM Rational Quality Manager se utiliza para desarrollar, ejecutar e informar el plan de prueba.

**MicroFocus ALM**

Es un software lean, ágil y DevOps que permite a los equipos de todos los tamaños ofrecer aplicaciones de alta calidad con mayor velocidad. Permite que las partes interesadas interactúen y se coordinen para **lograr los objetivos del proyecto.** Proporciona seguimiento e informes sólidos y una integración perfecta de varias tareas relacionadas con el proyecto. Puede conectarse a sistemas de correo electrónico y enviar correos electrónicos a los miembros del equipo.

**Microsoft ALM**

La suite ALM de Microsoft utiliza **Visual Studio,** que es uno de los entornos de desarrollo integrados más utilizados en la comunidad de desarrolladores. Team Foundation Server (TFS) de Microsoft proporciona una buena funcionalidad de configuración y control de versiones. La suite ALM de Microsoft es una opción particularmente buena para las organizaciones que usan un marco .NET porque muchos desarrolladores de .NET usan Visual Studio y también tienden a disfrutar usando TFS.

TFS, renombrado en 2022 como **Azure DevOps Server**, es un **conjunto integrado** de herramientas de desarrollo, sistema de compilación, métricas y control de versiones que utilizan los equipos especializados o calificados para organizar y ejecutar los proyectos.

* TFS ofrece una gama de herramientas colaborativas para el desarrollo de software que se pueden integrar con su entorno de desarrollo actual.
* Al usar TFS, se puede reducir la repetición del trabajo en el software en desarrollo al aumentar su transparencia, lo que a su vez da como resultado el envío de productos de software de alta calidad.
* TFS opera como un punto de contacto esencial tanto para la gestión de procesos como para la gestión de proyectos.

**Polarion ALM de Siemens**

Es un software ALM de nivel empresarial completo creado para conectar equipos y personas para **mejorar el proceso de desarrollo de aplicaciones.** Facilita los nuevos proyectos al brindar la capacidad de reutilizar o bifurcar sus datos (o código) para proyectos o líneas de productos similares. Comparta datos de forma segura y establezca permisos de usuario.

Polarion es una plataforma integrada que facilita a las organizaciones automatizar y optimizar los procesos de desarrollo en varios proyectos:

* Se utiliza para lograr la brillantez de la fabricación al definir, construir, probar y administrar el software multifacético en una solución basada en navegador o en la plataforma en la nube.
* Sus características importantes son el desarrollo unificado, la gestión de cambios y configuraciones, la colaboración y el cumplimiento, la gestión de pruebas y calidad, etc.
* Admite procesos de desarrollo en varias operaciones de desarrollo como en cascada, ágiles e hibrido.
* Así como técnicas innovadoras de resolución de problemas para equipos distribuidos.

**SpiraTeam**

Es la galardonada solución ALM de Inflectra que ayuda a los equipos a administrar los requisitos, versiones, iteraciones, tareas, defectos y código de su programa en un entorno, con trazabilidad completa. Los requisitos y los módulos de gestión de casos de prueba vienen con flujos de trabajo totalmente personalizables. La solución es excelente para la gestión de carteras de proyectos, la planificación de programas y la gestión de líneas base y ofrece una configuración robusta y control de versiones para artefactos. El seguimiento de errores, problemas y tareas es fácil y facilita el trabajo de los equipos de control de calidad. Las funciones para firmas electrónicas y colaboración de documentos son especialmente valiosas para las empresas que trabajan en las industrias reguladas.

A screenshot of a graph

Description automatically generated

Figura 2. Cuadros de mando con SpiraTeam. Fuente: Inflectra, s. f.

SpiraTeampresenta una **solución absoluta** de gestión del ciclo de vida de las aplicaciones que se utiliza para gestionar requisitos, planes, pruebas, errores, tareas y problemas en un único entorno.

* Se utiliza para la recopilación de requisitos, control de calidad, pruebas, informes personalizables, toma de decisiones, etc. por parte de equipos de todos los tamaños.
* Admite metodologías ágiles de gestión de proyectos como Scrum y Kanban.
* Usando SpiraTeam podemos crear los scripts de prueba manual, automatizarlos y también podemos administrarlos.
* SpiraTeam ofrece tableros incorporados con métricas importantes del proyecto.

**SwiftALM**

Es una plataforma **colaborativa** para la gobernanza de programas, proyectos y procesos de TI. Alinea los objetivos de TI con las estrategias corporativas, cuenta con funciones sofisticadas de modelado de organización y control de acceso con la capacidad de realizar múltiples niveles de desglose de información. Puede configurar los recursos, procesos y proyectos y controlar el acceso a planes e informes. La herramienta tiene una arquitectura de aplicaciones altamente **configurable y flexible** y tecnologías de análisis avanzadas para aumentar la productividad, la calidad y la entrega.

SwiftALM es una propuesta colaborativa para la gestión de proyectos, gestión de programas, SDLC y gobernanza de procesos.

* Es una herramienta de proyecto simple basada en la web que se utiliza para metodologías ágiles, en cascada e híbridas.
* Tiene un diseño estructural de aplicación extremadamente configurable y flexible que puede admitir hasta miles de usuarios.
* Se utiliza para configurar los procesos, proyectos, recursos y controlar el acceso a las funciones mediante una mejor planificación.
* Cubre la gestión de requisitos, cambios, problemas, riesgos, pruebas, defectos y documentos.

Resume las mejores prácticas de una organización en una plantilla para que sean reutilizables para proyectos futuros.

### **3.5. Plataformas CASE**

La mayoría de las plataformas CASE en el ámbito del software soportan el **lenguaje unificado** de modelado como representación básica. Las principales plataformas CASE de ingeniería del software son:

A screenshot of a computer

Description automatically generated

Tabla 2. Plataformas CASE. Fuente: elaboración propia.

**Altova UModel**

Es la herramienta lenguaje unificado de modelado (UML) que hace que el diseño de software visual sea **práctico** para cualquier proyecto. Es posible diseñar visualmente modelos de aplicaciones en UML y generar código Java, C++, C# o Visual Basic .NET y documentación de proyectos. O se puede aplicar ingeniería inversa a los programas existentes en diagramas UML, luego mejorar sus diseños y completar el viaje de ida y vuelta regenerando el código (Altova, 2022).

Mantenga sus vistas de diagramas y herramientas UML favoritas al alcance de su mano para mejorar la eficiencia y la productividad. Todo se puede **personalizar** en los diagramas UML que crea con UModel: tamaño, posición, color, características tipográficas y estilos de línea.

A screenshot of a computer

Description automatically generated

Figura 5. Vista de Altova UModel. Fuente: Altova, s. f.

* Es **compatible** con los catorce tipos de diagramas de arquitectura de software UML 2.5, además de diagramas de estilo UML para esquemas y tablas XML en bases de datos SQL populares.
* Permite agregar **hipervínculos** a cualquier elemento en cualquier diagrama UML. Puede crear hipervínculos a archivos externos, URL web u otros diagramas en su proyecto. Su descripción se convierte en texto emergente para el hipervínculo.
* Incluye un **generador de código integrado** que crea código Java, C++, C# o Visual Basic .NET basado en diagramas de clase, diagramas de secuencia y diagramas de máquina de estado en su modelo UML, liberándolo de la tediosa tarea de escribir a bajo nivel código de infraestructura para que pueda concentrarse en la lógica comercial y la arquitectura general de su proyecto.
* Incorpora una potente **capacidad de ingeniería inversa** para leer código fuente y archivos binarios de Java, C#, C++ y Visual Basic, y generar modelos UML para una representación visual mucho más clara y fácil de analizar que una copia impresa del código fuente como texto.
* Es posible modificar el código generado. La función de ingeniería de ida y vuelta lee el código modificado y actualiza automáticamente los diagramas UML en consecuencia.

**Eclipse Papyrus**

Es una herramienta de ingeniería basada en modelos de código abierto de grado industrial de la fundación Eclipse. Se ha utilizado notablemente con éxito en proyectos industriales y es la plataforma base para varias herramientas de modelado industrial (Eclipse, s. f.).

* **Basado en estándares:** UML 2.5, SysML 1.1 y 1.4, fUML 1.2.1, ALF 1.0.1, MARTE 1.1, BPMNProfile 1.0, BMM 1.3, SMM 1.1, PSCS 1.0, PSSM 1.0b, FMI 2.0 e ISO/IEC 42010.
* Posibilidad de **crear lenguajes específicos del dominio:** cada parte de Eclipse Papyrus se puede personalizar (perfil UML, explorador de modelos, notación y estilo de diagrama, vistas de propiedades, paleta y menús de creación y mucho más).
* **Model-Driven Engineering**: permite técnicas basadas en modelos (simulación basada en modelos, pruebas formales basadas en modelos, análisis de seguridad, análisis de rendimiento/compensaciones, exploración de arquitectura, etc.).

A computer screen shot of a computer

Description automatically generated

Figura 6. Vista de Papyrus. Fuente: Eclipse, s. f.

En este vídeo, Modelo UML con Eclipse Papyrus, se aprenderá a utilizar Papyrus, que está dentro de Eclipse, para modelar proyectos de software.

**Enterprise Architect**

Es la solución comercial de Sparx Systems como herramienta gráfica diseñada para ayudar a sus equipos a construir sistemas **robustos y fáciles de mantener.** Con capacidades integradas de gestión de requisitos, ayuda a rastrear especificaciones de alto nivel para modelos de análisis, diseño, implementación, prueba y mantenimiento utilizando UML, SysML, BPMN y otros estándares abiertos.

* **Repositorio de modelos de alto rendimiento:** tiene un desempeño espectacularmente rápido, ya que carga modelos extremadamente grandes en segundos.
* **Simulación dinámica de modelos:** es posible verificar la corrección de los modelos de comportamiento y obtener una mejor comprensión de cómo funciona un sistema empresarial.
* **Trazabilidad completa:** proporciona una trazabilidad completa desde los requisitos, el análisis y los modelos de diseño hasta la implementación y el despliegue.
* **Del modelo al código fuente:** el editor de código fuente incorporado permite navegar rápidamente desde el modelo directamente al código fuente en el mismo entorno. Las plantillas de generación de código permiten personalizar el código fuente generado según las especificaciones de la empresa.

A screenshot of a computer

Description automatically generated

Figura 7. Vista de Enterprise Architect. Fuente: Sparx System, s. f.

**Gleek.io**

Crea varios tipos de **diagramas UML:** diagramas de secuencia, diagramas de clases y diagramas de objetos. Los equipos también la usan para crear organigramas, diagramas de flujo, mapas mentales, entre otros. Como saben los usuarios avanzados y los desarrolladores, usar el teclado es mucho más rápido que usar un mouse. Debido a que se basa en los comandos del teclado, los desarrolladores trabajan más rápido que con los programas de dibujo de arrastrar y soltar. Y tiene ayuda de sintaxis disponible directamente en la ventana de diagramación en caso de que te quedes atascado.

A screenshot of a computer

Description automatically generated

Figura 8. Vista de Geek.io. Fuente: elaboración propia basado en Gleek.io.

**Lucidchart**

Proporciona un conjunto de **herramientas simples** para facilitar el modelado en colaboración. Funciona en tiempo real en muchas plataformas gracias a que su código está basado en HTML 5. Se integra con Confluence y también con MS Teams, Slack y G Suite. Maneja múltiples formatos de información, desde diagramas de sistemas complejos hasta mapas mentales. Es una herramienta interesante entre los miembros del equipo de desarrollo menos técnicos debido a su sencilla interfaz de usuario de arrastrar y soltar.

A screenshot of a computer

Description automatically generated

Figura 9. Vista de LucidChart. Fuente: Lucid, s. f.

**Plant UML**

Es un **proyecto de código abierto** que, con base en un lenguaje textual, genera multitud de diagramas como:

* Diagramas de secuencia.
* Diagramas de casos de uso.
* Diagramas de clases.
* Diagramas de objetos.
* Diagramas de actividades.
* Diagramas de componentes.
* Diagramas de despliegue.
* Diagramas de estados.
* Diagrama de tiempos.
* Wireframe graphical interface.
* Diagrama de Gantt.
* Mapa mental.
* Estructura de descomposición del trabajo.
* Diagrama entidad-relación.

A diagram of a user

Description automatically generated

Figura 10. Vista del modelado de casos de uso con PlantUML. Fuente: PlantUML, s. f.

En este vídeo, Documentando análisis y diseño técnico con PlantUML, se aprenderá a utilizar la herramiena PlantUML, que sirve para facilitar la documentación técnica y de diseño.

**Star UML**

Es un modelador de software sofisticado para un modelado **ágil y conciso.**

* Compatible con diagramas y metamodelo estándar UML 2.x: clase, objeto, caso de uso, componente, implementación, estructura compuesta, secuencia, comunicación, diagrama de estado, actividad, sincronización, desbordamiento de interacción, flujo de información y diagrama de perfil.
* Admite el modelado con diagramas SysML: requisito, definición de bloque, bloque interno y diagrama paramétrico.
* Soporte para crear diagramas de entidad-relación (ERD), diagramas de flujo de datos (DFD) y diagramas de diagrama de flujo.

A screenshot of a computer

Description automatically generated

Figura 11. Vista de StarUML. Fuente: StarUML, s. f.

**Visual Paradigm**

Es una aplicación de software diseñada para equipos de desarrollo de software para **modelar** sistemas de información comercial y **administrar** procesos de desarrollo. Además del soporte de modelado, esta tecnología proporciona capacidades de ingeniería de código y generación de informes, incluida la generación de código. Esta tecnología puede realizar ingeniería inversa de diagramas a partir de código y proporcionar ingeniería de ida y vuelta para varios lenguajes de programación.

* **Editor de diagramas de arrastrar y soltar:** admite UML, BPMN, ArchiMate, DFD, ERD, SoaML, SysML y CMMN.
* **Arquitectura empresarial:** es una herramienta de ciclo de vida TOGAF.
* **Planificación estratégica:** herramienta de gestión de procesos que te ayuda a descubrir y definir los objetivos estratégicos de tu negocio.
* **Diseño de procesos:** es posible visualizar el flujo de trabajo empresarial con BPMN, documentar el procedimiento de trabajo para las actividades y generar informes de procesos de alta calidad.
* **Análisis de procesos:** es posible estudiar el flujo de trabajo con la herramienta de animación de procesos. A su vez, se puede determinar la mejor configuración de recursos con la herramienta de simulación de procesos.
* **Reingeniería de procesos:** herramienta de gestión de procesos de reingeniería que ayuda a documentar el plan, diseñar nuevos procesos, identificar brechas y monitorear mejoras con indicadores (key performance indicators, KPI).
* **Diseño de experiencia de usuario:** aclare las necesidades de las partes interesadas con nuestras potentes herramientas de experiencia del usuario, soporta wireframes y prototipado rápido.
* **Generación de código:** soporta generación de código hacia Java, ORM e interfaces REST.
* **Ingeniería inversa:** obtenga diagramas a partir de código fuente.

A diagram of a form

Description automatically generated

Figura 12. Vista de Visual Paradigm. Fuente: Visual Paradigm, s. f.

En este vídeo, Creando nuestros propios metamodelos con Web Generic Modeling Environment (WebGME),se aprenderá a utilizar la herramienta WebGME, que no es la típica herramienta para modelar, pero puede ser muy útil para cuando es necesario crear con lenguajes propios.