

AMBIENTE - ENTORNO DE DESARROLLO DE SOFTWARE

AMBIENTE DE DESARROLLO DE SOFTWARE

El término “ambiente de desarrollo” se refiere al conjunto de condiciones y configuraciones bajo las cuales se lleva a cabo el desarrollo de software. Esto incluye:

- **Hardware:** Las computadoras y servidores utilizados (servidor de desarrollo, prueba, puesta en escena, producción)
- **Software:** Sistemas operativos, bases de datos (resguardos), y herramientas de desarrollo.
- **Configuraciones:** Parámetros específicos que se ajustan para que el software funcione correctamente durante su desarrollo.

Tipos de servidores. Cada uno tiene sus propias funciones:

- **Servidor de desarrollo:** se refiere al nivel central del entorno, donde los desarrolladores crean el código y lo prueban directamente en una aplicación. Proporciona herramientas básicas de hardware y software para ejecutar tareas de desarrollo, como la programación, el diseño y la depuración.
- **Servidor de pruebas:** en este entorno, los desarrolladores pueden identificar y corregir errores que pueden afectar al flujo de trabajo de la aplicación y a la experiencia del usuario. En definitiva, les permite comprobar si todas las funciones pueden funcionar como se busca.
- **Servidor de puesta en escena:** permite a los desarrolladores crear un duplicado del producto para demostrarlo a su cliente, pero no a los usuarios. Un servidor de puesta en escena les permite realizar cualquier cambio que deseen y garantiza que la aplicación se ejecute sin problemas en el servidor de producción real. Por ejemplo, el uso de un entorno de ensayo de WordPress te ayudará a actualizar los temas sin modificar el sitio web original.
- **Servidor de producción:** el entorno del servidor de producción, también conocido como entorno en vivo, es donde los usuarios finales pueden utilizar la aplicación. Aunque el producto en este servidor funcione correctamente y haya superado todas las pruebas, los desarrolladores pueden seguir realizando pruebas y añadiendo nuevas funciones en este último paso.

ENTORNO DE DESARROLLO DE SOFTWARE

El “entorno de desarrollo de software” es más específico (a diferencia del ambiente que trata de abarcar todo el ecosistema necesario para el desarrollo) y se refiere a las herramientas y procesos utilizados para crear, probar y mantener el software. Incluye generalmente:

- **IDE (Entorno de Desarrollo Integrado):** Aplicaciones como Visual Studio Code, IntelliJ IDEA que proporcionan un espacio de trabajo unificado para escribir y depurar código.
- **Control de Versiones:** Herramientas como Git para gestionar cambios en el código.

- **Automatización de Pruebas:** Herramientas que permiten ejecutar pruebas de manera automática para asegurar la calidad del software.

Nota: El ambiente de desarrollo abarca todo el ecosistema necesario para el desarrollo de software, el entorno de desarrollo se centra más en las herramientas y procesos específicos utilizados en la creación y mantenimiento del software.

Podemos establecer varios entornos con distintos fines por los que debe pasar el software durante su ciclo de vida antes de llegar al definitivo entorno de producción.

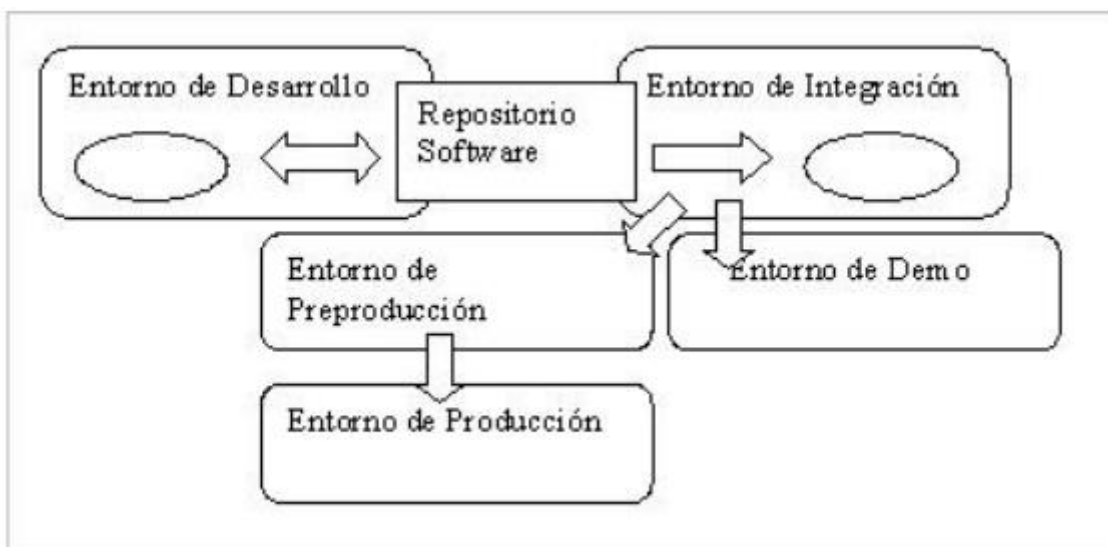


Fig.1 – Modelo de Entornos.

1 Repositorio

El elemento, posiblemente, más importante de nuestro modelo es el Repositorio Software. Es la base de datos central de las herramientas. El repositorio almacena toda la información que se va generando en un proceso de software o a lo largo del ciclo de vida del sistema o parte de este, amplía el concepto tradicional de un diccionario de datos para. Como, por ejemplo: componentes de análisis y diseño (diagramas de flujo de datos, diagramas entidad - relación, esquemas de bases de datos, diseños de pantallas), estructuras de programas, algoritmos, etc.

De hecho, muchos de los repositorios actuales hacen uso de un SGBD (normalmente relacional u orientado a objetos) como la tecnología de gestión de datos básica.

Las características más importantes de un repositorio son:

- **Tipo de información:** datos, gráficos, procesos, informes, modelos o reglas.
- **Tipo de controles:** si incorpora algún módulo de gestión de cambios, de mantenimiento de versiones, de acceso por clave, de redundancia de la información. La gestión de cambios y el mantenimiento de versiones, ayudan en el caso de que convivan diferentes versiones de la misma aplicación o se tengan que realizar cambios en la versión en producción y en la de desarrollo simultáneamente.

- **Tipo de actualización:** si los cambios en los elementos de análisis o diseño se ven reflejados en el repositorio en tiempo real o mediante un proceso por lotes (batch). Esto será importante en función a la necesidad de que los cambios sean visibles por todos los usuarios, en el acto.
- **Reutilización de módulos para otros diseños:** el repositorio es la clave para identificar, localizar y extraer código para su reutilización.
- **Posibilidad de exportación e importación:** para extraer información del repositorio y tratarla con otra herramienta (formateo de documentos, mejora de presentación) o incorporar al repositorio, información generada por otros medios.
- **Interfaces automáticas con otros repositorios o bases de datos externas:** para poder compartir información

En la siguiente figura se muestra entre otras cosas los contenidos que pueden ser alojados en un repositorio.

<p>Información de la empresa</p> <ul style="list-style-type: none"> Estructura organizativa Análisis del área de negocios Funciones de negocios Reglas de negocios Modelos de procesos (escenarios) Arquitectura de la información <p>Diseño de aplicaciones</p> <ul style="list-style-type: none"> Reglas de metodología Representaciones gráficas Diagramas de sistema Estándares de denominación Reglas de integridad referenciales Estructuras de datos Definiciones de proceso Definiciones de clase Arboles de menú Criterios de rendimiento Restricciones temporales Definiciones de pantalla Definiciones de informes Definiciones lógicas Lógicas de comportamiento Algoritmos Reglas de transformación 	<p>Construcción</p> <ul style="list-style-type: none"> Código fuente; código objeto Instrucciones de construcción del sistema Imágenes binarias Dependencias de configuración Información de cambios <p>Validación y verificación</p> <ul style="list-style-type: none"> Plan de comprobación; casos de prueba de los datos Guiones de comprobación de regresión Resultados de comprobaciones Análisis estadísticos Métricas de calidad de software <p>Información de gestión del proyecto</p> <ul style="list-style-type: none"> Planes de proyecto Estructura de desglose de tareas Estimaciones; planificaciones Carga de recursos; informe de problemas Solicitudes de cambios; informes de estado Información de auditoría <p>Documentación del sistema</p> <ul style="list-style-type: none"> Documentos de requisitos Diseños internos/externos Manuales de usuario
---	---

TABLA 31.1. Contenido de un repositorio CASE [FOR89B].

Metamodelo: Un concepto muy relacionado a los repositorios es el metamodelo este define las técnicas y metodologías soportadas por la herramienta, y que es conveniente que pueda ser extensible por parte del usuario. No siempre es visible.

El metamodelo determina la forma en que se almacena la información en el repositorio, la forma en que las herramientas pueden acceder a los datos y estos datos pueden ser visualizados por los ingenieros de software, el grado hasta el cual se puede mantener la seguridad e integridad de los datos, y la facilidad con que se puede ampliar el modelo ya existente para admitir nuevas necesidades.

2. Entornos

El entorno de desarrollo:

El entorno de desarrollo conlleva el abanico más amplio de tareas, que abarca desde el comienzo del ciclo de vida del software – toma de requerimientos – hasta la obtención de una versión mínimamente estable de la aplicación, o de un subconjunto de la misma – módulos.

En este entorno se llevan a cabo las siguientes tareas:

- Toma de Requerimientos. Una vez hemos concretado las necesidades del cliente y hemos establecido varias reuniones con este fin, es aconsejable documentarlo, de manera que todo el equipo pueda tener una visión global del proyecto – requerimientos funcionales. En esta fase suele ser necesario usar herramientas de documentación y toma de requerimientos.
- Análisis de Arquitectura y Diseño Técnico. Estas son dos de las tareas más importantes del desarrollo software y que pueden determinar el éxito o fracaso de un proyecto. Se deben realizar las siguientes funciones: acuerdo inequívoco de los requerimientos especificados, elección de la tecnología que pueda abarcar las necesidades del proyecto con una adecuada arquitectura y el diseño de los componentes.
- Implantación. Indica el comienzo de la implementación con la tecnología adecuada elegida en la fase de análisis
- Prueba de Unidad y Modulo. Estas pruebas se deben realizar sobre los módulos y componentes que cada desarrollador vaya finalizando. Se realizan en un entorno local para comprobar su correcto funcionamiento y poder integrarlas para obtener el producto final.

De las anteriores tareas podemos deducir las necesidades tanto software como hardware del entorno de desarrollo, teniendo en cuenta la política de trabajo en equipo que decidamos seguir.

El entorno de integración:

En el entorno de integración se lleva a cabo las siguientes tareas:

- Integración de los distintos módulos que componen la aplicación
- Pruebas de integración

En el proceso de integración tiene especial importancia el sistema de control de versiones. Debemos obtener los códigos fuentes etiquetados con la versión estable que deseamos integrar. Cuando los códigos fuentes son obtenidos se compilan y se genera la aplicación con todos los módulos integrados.

Desplegamos la aplicación basándonos en los documentos generados durante el proyecto. Una vez integrada y activada la aplicación, llevaremos a cabo la secuencia de pruebas. El objetivo de estas pruebas es comprobar el funcionamiento de la aplicación como un todo. En ellas, se trata de probar las funcionalidades que debe cumplir el producto.

Si la aplicación satisface las pruebas de integración, entonces está lista para su paso al entorno de preproducción – para realizar las pruebas de calidad – y al entorno de demostración – para permitir el acceso al cliente.

El entorno de preproducción:

En el entorno de preproducción se lleva a cabo las pruebas finales de la aplicación antes de su paso final al entorno de producción, donde se pondrá en funcionamiento en un escenario real.

El equipo de calidad, someterá la aplicación a un conjunto exhaustivo de pruebas, de diversos tipos:

- Funcionales y estructurales
- De rendimiento
- De tolerancia a fallos
- De seguridad

Si estas pruebas, conocidas como de aceptación, resultan satisfactorias, entonces la aplicación está ya lista para su paso al entorno de producción.

El entorno de producción:

El entorno de producción contiene en todo momento la versión activa de la aplicación. Los usuarios finales tienen acceso a la aplicación implantada en este entorno, de modo que resulta indispensable planificar y adoptar las medidas de seguridad oportunas, en consonancia con la importancia de la información que maneja el sistema.

La aplicación se despliega en el entorno de producción procedente de la versión existente en el entorno de preproducción.