



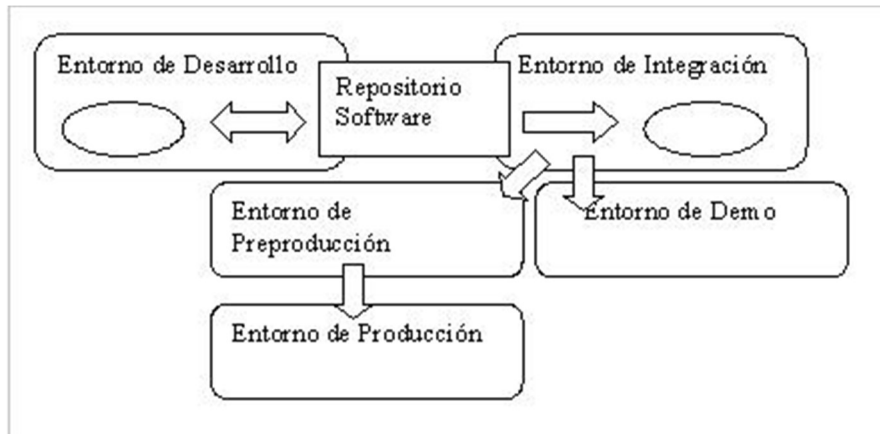
ENTORNOS DE INFORMÁTICA

ÍNDICE

1.	INTRODUCCIÓN	3
2.	ENTORNO DE DESARROLLO	5
3.	ENTORNO DE INTEGRACIÓN	6
4.	ENTORNO DE PREPRODUCCIÓN	7
5.	ENTORNO DE PRODUCCIÓN	7
6.	ENTORNO DE DEMOSTRACIÓN	7

1. INTRODUCCIÓN

El entorno de trabajo para el desarrollo de un producto software es el conjunto de circunstancias y estados en los que se puede encontrar dicho producto a lo largo de su ciclo de vida.



En la imagen se muestra los elementos que componen el modelo y la interacción que existe entre ellos. A continuación se explica cada elemento y sus principales funciones dentro del marco propuesto.

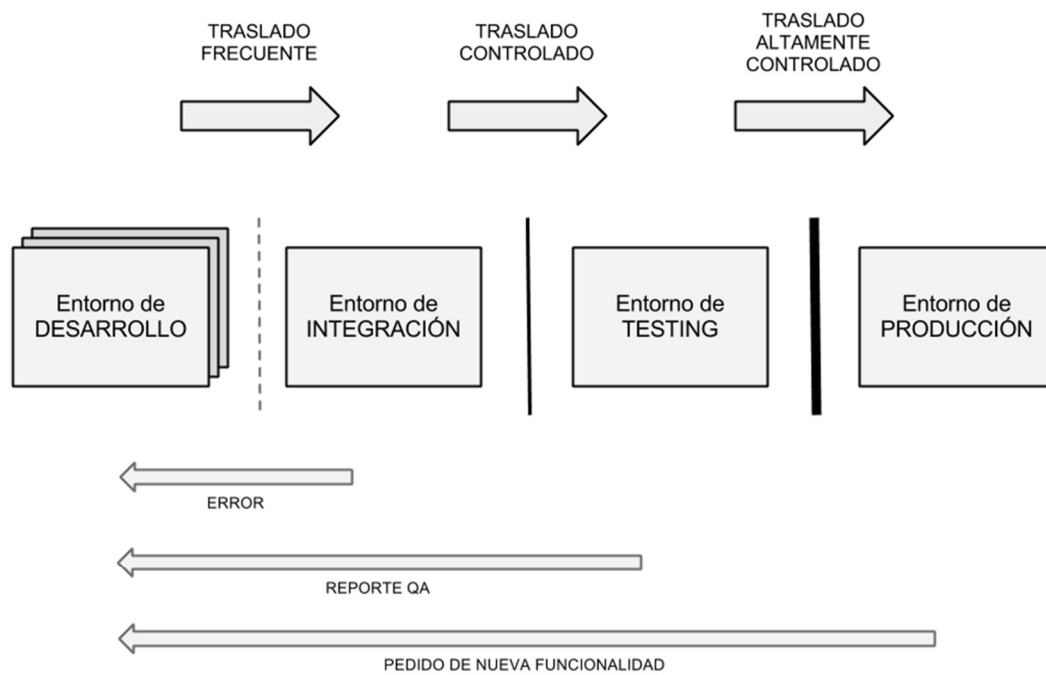
El elemento, posiblemente, más importante de nuestro modelo es el Repositorio Software (Control de versiones). Podemos definir un repositorio como un almacén, común a un grupo de trabajo, de elementos software necesario para el desarrollo de aplicaciones y la gestión de proyectos.

En el modelo de entornos, un producto software evoluciona de manera que pasa por los entornos de desarrollo, integración(pruebas), preproducción(aceptación) y producción.

El ciclo de vida del software abarca desde la toma de requerimientos hasta la obtención de una versión estable de la aplicación, o de un subconjunto de la misma.

En cada entorno se llevan a cabo las siguientes tareas dentro del ciclo de vida:

- Implementación: Es en esta fase donde se utiliza el entorno de desarrollo para hacer toda la implementación de nuestro aplicativo.
- Pruebas de unidad y módulos: Es en esta fase donde se utilizan los entornos de integración y preproducción (aceptación).
- Explotación: Es en esta fase donde se utilizan el entorno de Producción y demostración.



2. ENTORNO DE DESARROLLO

- Implementación: Es en esta fase donde se utiliza el entorno de desarrollo para hacer toda la implementación de nuestro aplicativo.

En este entorno es donde se hace todo el desarrollo inicial, y se va almacenando en nuestro repositorio a través del control de versiones cada vez que se sube un aplicativo al entorno de integración.

Los desarrolladores son los responsables de subir los módulos al sistema de control de versiones y etiquetarlos correctamente. El sistema de control de versiones es necesario puesto que el entorno de integración obtendrá de ahí los archivos que hayan sido modificados.

Tener un responsable para esta tarea se hace indispensable para canalizar, a través de una única persona, el proceso de obtención de los archivos modificados. Este rol debe ser asumido por uno de los componentes del proyecto.

El paso desde el entorno de desarrollo al de integración es el menos crítico de los que se llevan a cabo.

3. ENTORNO DE INTEGRACIÓN

En el entorno de integración se lleva a cabo la integración de los distintos módulos que componen la aplicación y por tanto las pruebas de integración.

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, aunque también deben ser probadas las anteriores funcionalidades para evitar que los cambios introducidos alteren inadvertidamente el comportamiento de las mismas.

Existen dos formas de integrar el software: incrementalmente, donde el código se escribe y se prueba en partes pequeñas que, a continuación, se añaden a un conjunto de trabajo de una en una, o no incrementalmente, es decir, se prueba todo junto.

El principal inconveniente de la integración no incremental es que introduce muchas variables y dificulta la localización de errores. Esto se debe, principalmente, al hecho de que un error podría estar en cualquiera de los componentes nuevos, en la interacción entre los componentes nuevos en el centro del sistema o en la interacción entre los componentes nuevos.

Las ventajas de la integración incremental son:

- Los errores son fáciles de localizar. Cuando se produce un problema nuevo durante la integración incremental, el componente nuevo o cambiado, o su interacción con los componentes integrados anteriormente, son los lugares obvios para buscar un error. La integración incremental hace más probable que los defectos se descubran de uno en uno, lo que facilita la identificación de errores.
- Los componentes se prueban de forma más completa. Los componentes se integran a medida que se desarrollan y, después, se prueban. Esto significa que los componentes se ejercitan más a menudo que si la integración se realiza en un solo paso.
- La ejecución se produce antes. Los desarrolladores ven los primeros resultados del trabajo y no tienen que esperar hasta el final, lo que es mejor para su moral. Esto también hace posible obtener información de retorno antes.

Es importante comprender que la integración se produce, como mínimo, una vez en todas las iteraciones. Un plan de iteración define qué deben utilizar los guiones de uso para el diseño y qué clases se deben implementar. Lo principal de la estrategia de integración es determinar el orden de implementación y combinación de las clases.

Una vez que el proyecto ha sido validado mediante las pruebas de integración, se etiqueta el proyecto para pasarlo a preproducción (aceptación). Al igual que en el caso anterior habrá un responsable encargado de desplegar el proyecto en el entorno de preproducción utilizando el documento de despliegue.

Dicho documento indica que módulos se deben subir y cuál es la localización correspondiente a cada módulo.

4. ENTORNO DE PREPRODUCCIÓN

La aplicación se transferirá desde el entorno de integración a un entorno similar al que se expondrá en el entorno de producción.

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. Para ello, en primer lugar se transfiere la aplicación una vez ha sido validada en el entorno de integración – a ser posible de una forma automatizada. 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. Si alguna de las pruebas falla debemos seguir el protocolo creado para la gestión de incidencias.

5. 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.

Por otra parte, este entorno también contiene los datos reales, información que es preciso salvaguardar frente a posibles pérdidas mediante la aplicación sistemática de una política de copias de seguridad. También debemos proteger los datos frente a exposición o usos fraudulentos de los mismos, restringiendo su acceso exclusivamente al personal de confianza que administra el sistema.

La aplicación se despliega en el entorno de producción procedente de la versión existente en el entorno de preproducción. La subida debería producirse de forma automática para evitar, en la medida de lo posible, la introducción de errores.

Este paso es el más crítico de todos. La tarea a realizar es idéntica al paso anterior. Si ha habido algún problema de subida de entorno debe producirse en el entorno anterior y se debe quedar resuelto antes de llegar a este punto.

6. ENTORNO DE DEMOSTRACIÓN

Este entorno tiene dos funciones: posibilitar al cliente el acceso a la aplicación que se está desarrollando y mostrar los productos existentes con el fin de atraer a posibles clientes.