**2.4 Plan del Proyecto de Desarrollo de Software.**

Como se mencionó anteriormente, el Plan del Proyecto es el resultado principal del proceso de planificación. Este plan existe sólo cuando está documentado, distribuido, entendido y comprometido.

El contenido básico del plan del proyecto se indica en la Figura 2.1

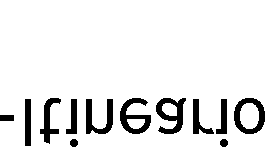
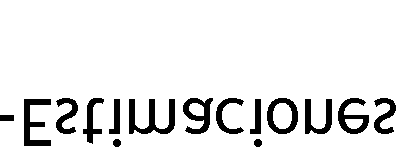
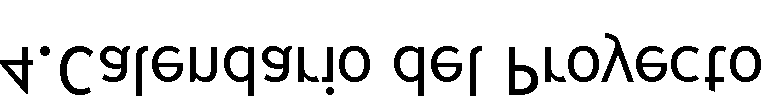
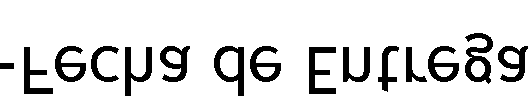
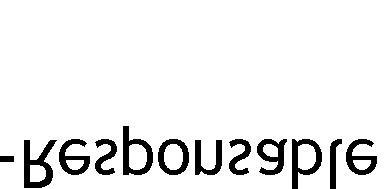
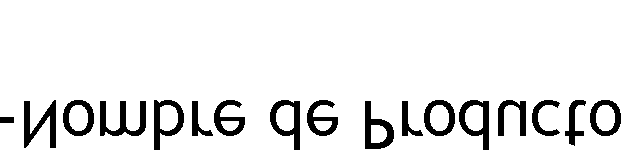
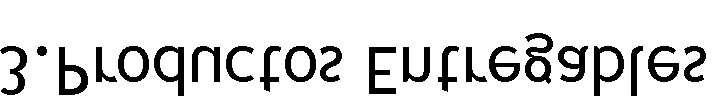
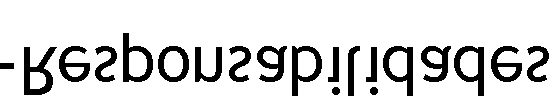
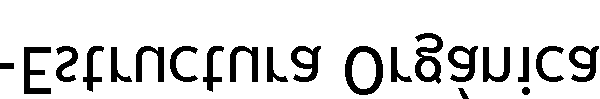
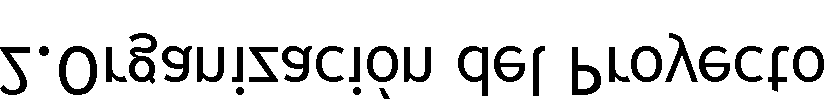
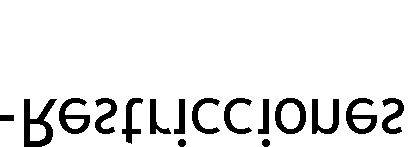
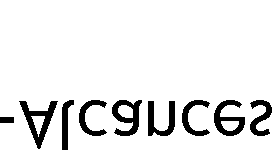
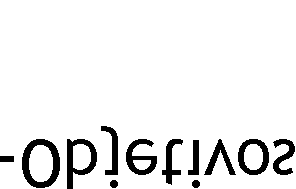
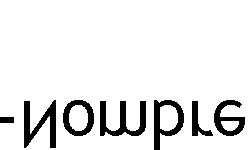
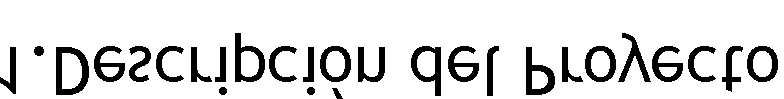
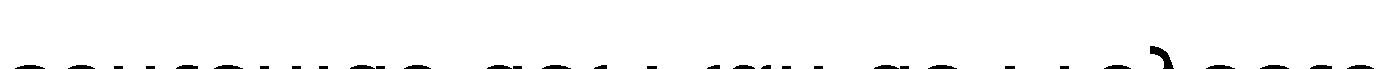
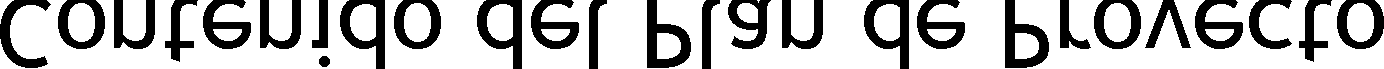


Figura 2.1: Contenido del Plan de Proyecto

La Descripción del Proyecto proporciona las características generales de éste. La Organización refleja la forma en que el grupo de proyecto ha sido estructurado para llevar a cabo el trabajo y los responsables de las funciones clave. Los Productos a Entregar incluyen los documentos u otro tipo de producto, con compromiso de entrega al usuario o a otros grupos de trabajo interno del proyecto, así como los responsables de la entrega. El Calendario comprende tanto las estimaciones realizadas para confeccionar y justificar el itinerario del proyecto, como éste mismo.

**2.4.1 ¿Para qué se usa el plan del proyecto?**

Los proyectos de Desarrollo de Software involucran a diversos participantes y cada uno de ellos da un uso distinto al plan del proyecto, ver Tabla 2.1.

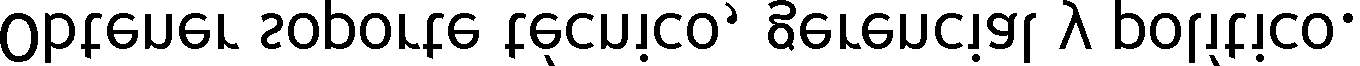
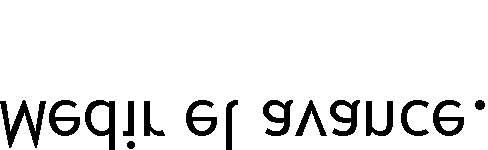
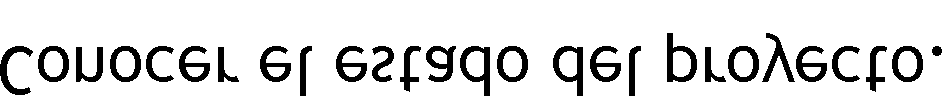
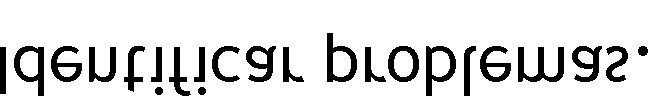
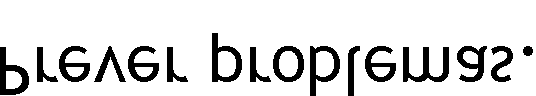
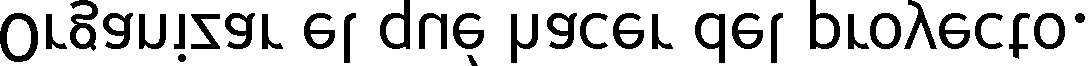
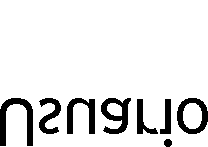
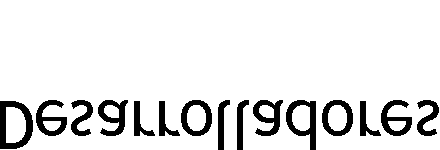
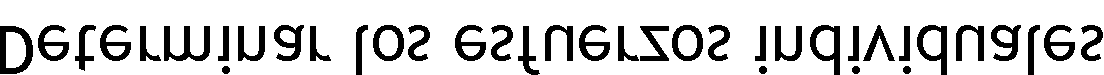
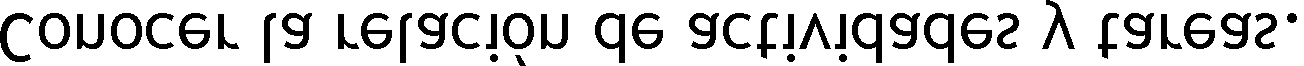
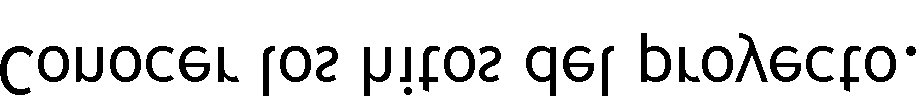
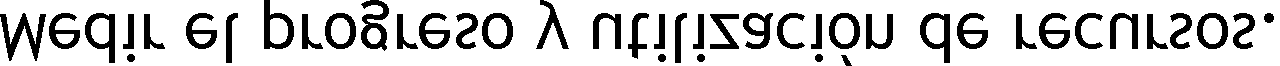
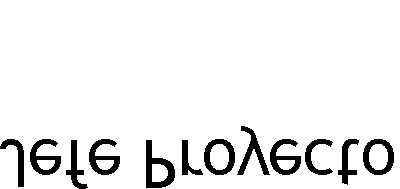


Tabla 2.1: Uso del Plan del Proyecto.

Plan de fases.

Se analiza el ciclo de desarrollo del proyecto como es: análisis de requisitos, fase de diseño de alto nivel, fase de diseño de bajo nivel, etc. Asociada con cada fase debe de haber una fecha que especifique cuando se debe terminar estas fases y una indicación de cómo se pueden solapar las distintas fases del proyecto.

**Las 5 fases esenciales para la gestión de proyectos**

**[](https://www.eoi.es/blogs/embacon/files/2014/04/imagen_blog1.jpg)**

**1. Análisis de viabilidad del proyecto**

Es la fase inicial de cualquier proyecto y su objetivo es**analizar si la empresa debe o no embarcarse en dicho proyecto**, pues en ciertas ocasiones la empresa puede tener más problemas que beneficios a la hora de realizar un proyecto.

El análisis de viabilidad debe incluir al menos las siguientes actividades:

* Creación del registro de seguimiento en la herramienta de gestión de la empresa.
* Análisis previo del alcance del proyecto.
* Análisis de los riesgos de ejecución del proyecto.
* Análisis de viabilidad de acuerdo a plazos, coste y calidad.

Si el análisis de viabilidad es positivo, es decir, no existen motivos iniciales para no abordar el proyecto, podemos pasar a la 2ª fase del proceso.

**2. Planificación detallada del trabajo a realizar**

El objetivo de esta fase es definir con el máximo detalle posible las **tareas a realizar** los **recursos necesarios**para llevar a buen término el proyecto. Un error de cálculo en esta fase puede ser muy dañino para la empresa.

Las actividades clave en esta etapa son:

* Análisis del alcance del proyecto.
* Realización de estimaciones de esfuerzo, costes y recursos.
* Definición del plan de proyecto.
* Negociación del contrato.

Si finalmente el proyecto es aprobado, debe plasmarse en un contrato que recoja todos los términos del acuerdo.

**3. Ejecución del proyecto**

La tercera fase de la gestión de proyectos es donde las empresas despliegan todo su*Know how* y donde menos problemas suelen encontrar.

Las actividades principales de esta etapa son:

* Establecimiento del entorno de trabajo.
* Asignación de las tareas planificadas a los recursos disponibles.
* Ejecución de las tareas planificadas.
* Gestión de las peticiones de cambio.

De forma paralela a esta fase, es necesario realizar un **seguimiento y control del proyecto**que velará por el cumplimiento de la planificación y la calidad del trabajo realizado. De manera que se detecten las desviaciones antes de que se conviertan en un problema para el éxito del proyecto.

**4. Seguimiento y control del trabajo**

Esta etapa, junto a la de planificación detallada, es una de las más **importantes para el éxito del proyecto**.

En esta fase se realizan esencialmente 4 tipo de actividades:

* Seguimiento de tareas e hitos planificados.
* Gestión de entregables (incluido control de la calidad).
* Gestión de incidencias.
* Generación de informes de seguimiento.

**5. Cierre del proyecto**

El objetivo de esta fase es **institucionalizar una etapa de control para verificar que no quedan cabos sueltos** antes de dar por cerrado el proyecto. Esta fase suele ejecutarse a través de un checklist o lista de control.

Las tareas clásicas a realizar dentro de esta fase son:

* Cierre formal del proyecto por parte de todos los actores involucrados en el proyecto (stakeholders).
* Realización del backup del proyecto.
* Análisis de los resultados con respecto a las estimaciones iniciales.
* Actualización de la base de conocimiento con todo lo aprendido.

**Plan de iteración**

El plan de iteración debe detallar qué se debe hacer de modo específico, para que quede poco margen de confusión sobre la verdadera posición o las responsabilidades en cualquier momento. Se suele utilizar algún tipo de herramienta de planificación del proyecto. Esto es un plan detallado para una iteración. Suelen haber dos planes de este tipo: uno para la iteración actual, y otro en construcción para la iteración siguiente.

Para definir el contenido de una iteración, necesitará:

* el plan de proyecto
* el estado actual del proyecto (bajo seguimiento, retrasos, número elevado de problemas, arrastra los requisitos, etc.)
* una lista de casos de ejemplo o casos de uso que se deben completar al final de la iteración
* una lista de riesgos que se deben solucionar al final de la iteración
* una lista de cambios que se deben incorporar al producto (arreglos de errores, cambios en los requisitos)
* una lista de las clases principales o paquetes que deben estar completamente implementados

Estas listas deben disponer de un rango. Los objetivos de una iteración deben ser agresivos para que, cuando surjan dificultades, los elementos se puedan eliminar desde las iteraciones basándose en sus rangos.

#### **Criterios de evaluación**

Cada iteración se concluye con una **valoración**. Para la valoración de esta iteración puede evaluar los resultados de la iteración relativos a los criterios de evaluación que se han establecido para el plan de iteración.

Los criterios de evaluación se han establecido antes de cada iteración y establecen objetivos para el conjunto de características, calidad y rendimiento que se deben conseguir en la iteración. El alcance real de estos objetivos variará. Por ejemplo, en una iteración determinada, el conjunto de características se puede superar, la calidad se alcanza con dificultades y falta rendimiento.

Asimismo, los objetivos también se pueden expresar como objetivos mínimos y deseables. Por ejemplo, puede existir un conjunto de características necesarias y algunas características deseables que se intentarán en esta iteración si la velocidad del desarrollo y el nivel del personal lo hacen viable.

**ADMINISTRACION DEL RIESGO**

En el contexto de proyecto, riesgo implica las amenazas de sufrir daño o pérdida (resultado negativo) y también incluye las oportunidades (resultados positivos). La administración de riesgo es el medio a través del cual la incertidumbre se maneja de forma sistemática con la finalidad de disminuir la probabilidad de ocurrencia de resultados negativos y aumentar la probabilidad de lograr los objetivos del proyecto.

La evaluación de riesgo es probablemente el paso más importante en un proceso de gestión de riesgos y también el paso más difícil y con mayor posibilidad de cometer errores. Durante esta etapa se deben identificar los diferentes riesgos existentes, determinar la probabilidad de la ocurrencia y las consecuencias que podrían generar, así como también clasificarlos. Es conveniente para mayor claridad agrupar los riesgos en grupos de acuerdo por ejemplo a su peligrosidad.

En segundo lugar se encuentran las técnicas y estrategias necesarias gestionar los riesgos que presentan mayores probabilidades de impacto, así como también, establecer las acciones a ejecutar en respuesta de estos eventos. Dichas acciones deben ser supervisadas de acuerdo a la eficacia de las estrategias y de los niveles de modificación de los riesgos a lo largo del proyecto.

Existen en la actualidad varias metodologías usadas para la gestión de proyectos, la mayoría de ellas consideran como factor importante dentro de la administración de proyectos, el análisis y gestión de riesgos; otras metodologías sin embargo han sido concebidas y creadas exclusivamente como un conjunto de pautas a seguir para descubrir y planificar las medidas oportunas para mantener los riesgos bajo control, en ciertos proyectos de tipo especifico.

La guía del [PMBOK](http://es.wikipedia.org/wiki/Project_Management_Body_of_Knowledge) de [PMI](http://es.wikipedia.org/wiki/Project_Management_Institute), plantea seis procesos que tienen lugar por lo menos una vez en cada proyecto:

* **Planificación de la gestión de riesgos:** En el cual se decide, como enfocar, planificar y ejecutar las actividades de gestión de riesgos para un proyecto
* **Identificación de riesgos:**Permite determinar qué riesgos pueden afectar al proyecto y documentar sus características.
  + **Lista del riesgo.**

La Lista de riesgos identifica, en orden decreciente de prioridad, los sucesos que pueden llevar a un resultado negativo significativo. Sirve como punto focal para las actividades de proyecto y es la base alrededor de la cual se organizan las iteraciones.

* **Análisis cualitativo de riesgos:**Cada riesgo se clasifica según su probabilidad de ocurrencia e impacto, para realizar otros análisis o acciones posteriores.
* **Análisis cuantitativo de riesgos:**Cada riesgo identificado en los objetivos generales del proyecto es analizado según su efecto.
* **Planificación de la respuesta a los riesgos:**Se desarrollan opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.
* **Seguimiento y control de riesgos:**Una vez identificados los riesgos del proyecto, es necesario realizar un seguimiento a éstos, además de supervisar los riesgos residuales, identificar nuevos riesgos, ejecutar planes de respuesta a los riesgos y evaluar su efectividad a lo largo del ciclo de vida del proyecto.

Todas las herramientas y técnicas de proyectos muy útiles, como por ejemplo: el Diagrama de Gantt, Método Pert/CPM, EDT etc., sin embargo, en lo personal prefiero el “**Análisis de Riesgos Cualitativo y Cuantitativos”**debido a que, mediante esta técnica se busca priorizar cuáles serán los riesgos más peligrosos que podrían dar al traste con los objetivos del proyecto. Esta información puede ser usada para apoyar decisiones tales como: cancelación de proyectos, asignación de recursos, demora en la realización de las actividades, ampliación del alcance de un proyecto determinado, etc. En definitiva, es fundamental para el éxito de los proyectos.

# **Gestión de Configuración y Control de Cambios**

La Gestión de Configuración y Control de Cambios es la disciplina que se encarga de

* Identificar los elementos del proyecto que deben estar bajo configuración
* Restringir los cambios a dichos elementos
* Auditar los cambios a estos elementos
* Definir y gestionar la configuración de estos elementos

Los métodos, procesos y herramientas utilizados para proveer este ambiente de configuración son parte esencial del proceso de desarrollo de software.

La Gestión de Configuración y Control de Cambios es esencial al momento de tener control sobre todos los elementos generados por los integrantes del equipo de proyecto. Este control ayuda a eliminar la posibilidad de confusiones que pueden resultar de alto costo para el proyecto y asegurar que no existan inconsistencias en el sistema desarrollado, generadas por:

* Actualizaciones simultáneas

Cuando varios integrantes del equipo trabajan sobre un mismo elemento al mismo tiempo.

* Problemas en la notificación de cambios

Cuando un problema fue resuelto para algún elemento que es compartido por varios desarrolladores y alguno de ellos no fue notificado de dicho cambio.

* Múltiples versiones

Usualmente se tienen varias versiones del producto en desarrollo, por ejemplo una versión de desarrollo, y otra de test, y se quiere que cuando haya un cambio en una éste se vea reflejado en las demás versiones.

Algunos de los beneficios que se obtienen de la realización de una buena Gestión de Configuración y Control de Cambios son:

* Brindar apoyo a los métodos de desarrollo de software.
* Mantener la integridad del producto.
* Asegurar la completitud y correctitud de los elementos bajo configuración.
* Proveer un ambiente estable y controlado de trabajo.
* Restringir y controlar los cambios que se realizan.
* Proveer mecanismos de rastreo de porqué, cuando y quién realizó un cambio.

## 2.3 Ciclo de Planificación de Proyectos de Desarrollo de Software

El ciclo de planificación de proyectos de Desarrollo de Software, comienza con los requerimientos iniciales y tiene las siguientes etapas:

Negociación de Compromisos

El jefe de proyecto y el cliente y/o usuario negocian los compromisos mutuos, los cuales se establecen sobre la base de los requerimientos del producto de software y objetivos del proyecto.

Descomposición de Requerimientos

El producto de software se divide en elementos claves denominados Estructuras de División del Trabajo (EDTo WBS). Una EDT es un organigrama jerárquico donde se establecen las distintas partes de un producto de software. Representa una jerarquía de componentes o bien de procesos. La jerarquía de componentes identifica cada uno de los componentes del software y la manera en que éstos se relacionan. La jerarquía de procesos representa las actividades de trabajo requeridas para desarrollar el software y sus interrelaciones. Si se usa este tipo de EDT se deben considerar las fases, actividades y tareas estándares definidas por la organización y también las tareas especiales del proyecto.

Estimación del Tamaño de un producto de Software

Una vez establecido el estándar de medición (Líneas de Código, Puntos de Función, Puntos Objetos), se utiliza la EDT de componentes para estimar el tamaño de cada componente del software. El tamaño total del producto de software se obtiene al sumar los valores estimados para cada componente y al ajustar la estimación de acuerdo a la información histórica de la organización, si es necesario.

Estimación de Recursos

El tamaño del producto de software sirve de base para estimar esfuerzo (Persona-Mes, Hombres-Hora), tiempo y costo de desarrollo. Los modelos empíricos de estimación de costos de software cumplen éste propósito. La estimación de recursos puede hacerse en el ámbito de proyecto, de fases y de actividades y tareas.

Desarrollo de Itinerario del Proyecto

El itinerario del proyecto se confecciona distribuyendo el esfuerzo estimado dentro del marco de tiempo establecido. El itinerario debe considerar los hitos del proyecto.

Término de fase y/o actividades.

El término de cada fase o actividades se establece formalmente y define un hito o un producto.

Generación y entrega de productos.

En ciertas partes de itinerario es necesario que la actividad de generar el producto sea explicita. Generalmente en proyectos de SW el producto es un informe.

Puntos de control o Hitos del proyecto

El itinerario y las estimaciones resultantes se comparan con las necesidades iniciales, si éstos se ajustan, los compromisos pueden ser hechos y el trabajo puede proceder. Generalmente los costos son muy altos y el itinerario demasiado largo, en este caso se requiere volver a la negociación de compromisos y replanificar, si es necesario.

La existencia de una base de datos que registre información histórica de los proyectos de Desarrollo de Software de una organización, permite contar con factores de ajuste para estimaciones futuras, mejorando progresivamente el proceso de planificación.

al vez para algunos, el término "administración de la configuración" se refiere a la configuración de una máquina, el cómo hacer funcionar cierto software en alguna computadora, o modificar ciertos parámetros del sistema operativo. Esto podría ser correcto, siempre y cuando no estemos hablando del concepto dentro de la cultura de procesos. Dentro de este contexto, la administración de la configuración, o SCM (Software Configuration Management), es el arte de identificar, relacionar, almacenar y trabajar con los productos de trabajo; entendiendo por productos de trabajo: planes, código fuente, librerías, modelos, documentación; es decir, todo lo que se genera en el ciclo de desarrollo de software.

El Instituto de Administración de la Configuración (ICM), la define como: “el proceso de administrar el espectro de los productos de la organización, facilitando y procesando la administración de todos los requerimientos, incluyendo cambios, y actualizando los resultados conforme a aquellos requerimientos” [ICM 1998]. Como podemos apreciar, esta definición también incluye al proceso en sí, y sus procedimientos referentes.

Mejor Organización de los Productos  
La Administración de la Configuración nos ayuda a localizar más fácilmente nuestros productos, ya sean para uso propio o de algún otro involucrado en el proyecto. Adicionalmente, nos ayuda a tener un control sobre las versiones de lo que vamos desarrollando. De hecho, nació con la idea de poder establecer mecanismos para tener el control sobre lo que se estaba creando en el ciclo de desarrollo de software. Esto fue porque se tenían sistemas complejos y se quería reducir el tiempo de respuesta al momento de buscar algún producto de trabajo en los proyectos.

Los proyectos donde no se aplica la administración de la configuración suelen tener un mayor costo en tiempo y esfuerzo, dado que la falta de control y organización provoca re-trabajo y pérdida de tiempo. Por ejemplo, cuando estamos buscando en algún cajón donde no tenemos orden, nos es más complicado encontrar algo, en cambio si tenemos ordenado ese cajón se nos reduce el tiempo de búsqueda, nos acordamos más de las cosas que tenemos guardadas en el cajón y encontramos lo que andamos buscando.

Es recomendable tener una buena administración de la configuración no solo para ayudarnos a encontrar más rápidamente algún producto, sino que también nos da orden sobre la manera en cómo organizamos los archivos, dónde guardamos cada cosa, cómo la nombramos para identificarla más rápidamente y también cómo vamos versionándola para que podamos realmente saber cómo ha ido cambiando y poder regresar a un cierto estado, en caso de ser necesario.

Entre los beneficios que encontramos al aplicar la administración de la configuración tenemos: la disminución de los costos de mantenimiento del software, se tiene un mejor aseguramiento de la calidad, reducción de defectos y errores, mejora la productividad del desarrollo, desarrollo que no depende de la persona o personas que lo están desarrollando sino del proceso y de la documentación que se está generando.

Control de Versiones  
En los proyectos o desarrollos pequeños, es común pensar que se puede tener el control sobre todos los productos, sin necesidad de llevar un control de versiones. Sin embargo, la realidad es que los sistemas son cada vez más complejos, y con mayor número de componentes que interactúan entre sí. Si no tenemos un control de versiones adecuado, no podemos saber cuales son los productos modificados en cierto punto y cuales son las últimas versiones. Esto puede provocar que trabajemos utilizando versiones inadecuadas, generando así un doble trabajo.

**Preparación del Entorno de Desarrollo**

La preparación del entorno de desarrollo es una parte muy importante dentro de la construcción de aplicaciones ya que permite que el trabajo en equipo se realice de forma sencilla al contar el equipo de desarrollo con las mismas herramientas.

Por este motivo, [MADEJA](http://www.juntadeandalucia.es/servicios/madeja/glossary/12/letterm#term225) presenta una serie de recomendaciones y ejemplos necesarios para la preparación del entorno de desarrollo.

A partir de los contenidos de este área, el equipo de desarrollo puede ampliar los conocimientos en el [Área Gestión de la Entrega](http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/entorno/area-gestion-entrega)donde se especifican las configuraciones necesarias de los proyectos de desarrollo para su entrega, como el uso de Maven, Subversion y Artifactory; en el área [Arquitectura Tecnológica](http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/arquitectura/arquitectura-tecnologica) del subsistema de [Arquitectura](http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/arquitectura) se recogen recomendaciones y ejemplos específicos de configuraciones tecnológicas recomendadas en función del tipo de proyecto. Por último, el equipo de desarrollo deberá contemplar las recomendaciones dadas en el subsistema de [Desarrollo](http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/desarrollo) referentes a los Especificaciones de codificación a emplear en función del lenguaje elegido, recomendaciones sobre Seguridad y Rendimiento, así como indicaciones sobre el uso de determinadas librerías y tecnologías en cada una de las capas de las aplicaciones.