

Vicerrectoría de Maestrías y Programas Ejecutivos
Dirección de Diseño Académico

Guía para el Profesor

Administración de proyectos de software



Índice

Certificados	3
¿Certificado o certificación?	3
Certificado en Productividad basada en herramientas tecnológicas.....	4
Competencia global del certificado	4
Competencia del curso.....	4
Metodología del curso.....	5
Temario	6
Bibliografía y recursos especiales	7
Evaluación y agendas.....	7
Notas de enseñanza	8

Certificados

Para entender la importancia del curso del cual usted será **Facilitador**, es necesario ofrecer un contexto mayor sobre el programa de **Certificados** de la Universidad Tecmilenio, pues son parte medular del nuevo modelo educativo basado en el **aprender haciendo** y en **brindar una experiencia educativa a la medida de los alumnos**.

Un certificado es un **programa académico corto compuesto de varias materias**, embebido en la segunda mitad del plan de estudios de profesional, que busca desarrollar **competencias muy específicas** en el alumno y lo prepara para desempeñarse de la mejor manera en un empleo.

SABER + HACER + BIEN

Con este enfoque, buscamos en los egresados de profesional que además de **saber** (tener un conocimiento teórico), también sean **capaces de hacer** (tener la habilidad de realizar una tarea) y de **saber-hacer** (entender lo que se hace y tener la capacidad para hacerlo de la mejor forma).

En Universidad Tecmilenio, **aprender haciendo** significa que el participante cursará **Certificados en los que desarrolla competencias disciplinares de especialidad que son valoradas por el mercado laboral**, convirtiéndose en un profesional altamente competente y elevando así su índice de empleabilidad.



La mayoría de nuestros Certificados se compone en promedio de cuatro materias, las cuales tienen un seguimiento lógico y terminan con un proyecto de gran calado y un alto nivel de complejidad (última materia). Una correcta realización del proyecto integrador demostrará el dominio de la competencia global declarada en cada certificado.

¿Certificado o certificación?

Es muy importante tener en claro que un certificado y una certificación son dos cosas distintas. Un **certificado** es un reconocimiento formal que **otorga internamente la Universidad Tecmilenio** a los estudiantes que demuestren haber aprobado las materias correspondientes, y adquirido la **competencia** global del certificado.

Por su parte, la **certificación** es también un reconocimiento, pero ésta se obtiene a través de la acreditación de un curso específico del programa académico de la Universidad y aprobando un examen de suficiencia aplicado por una **entidad acreditadora externa** (mapas mentales, idiomas, uso de software, etc.).

Su trabajo como docente Facilitador de este curso es muy importante para nosotros. Gracias por aportar su conocimiento y experiencia en la impartición de este certificado. A continuación, podrá revisar información detallada del curso que impartirá.

GUÍA PARA EL PROFESOR

Certificado en Productividad basada en herramientas tecnológicas

El certificado de Productividad basada en herramientas tecnológicas se compone de **tres** cursos, con la siguiente progresión:



Como se puede apreciar, este curso de **Administración de proyectos de software** es el primer curso del certificado de Productividad basada en herramientas tecnológicas. Por lo mismo, es importante que como **Facilitador verifique** que sus estudiantes hayan aprobado los cursos anteriores, pues de no haberlo hecho se podrá ver afectado el aprovechamiento académico de este curso.

Competencia global del certificado

Al finalizar el certificado de **Productividad basada en herramientas tecnológicas**, el participante deberá haber desarrollado y adquirido la siguiente competencia global, en toda su extensión:

Genera soluciones computacionales de alto valor para las organizaciones a través del desarrollo de productos multiplataforma en lenguaje Java®.

Competencia del curso

La competencia específica que el participante habrá de obtener al aprobar satisfactoriamente el **curso de Administración de proyectos de software** es la siguiente, en toda su extensión:

Aplica herramientas de integración continua, control de pruebas y reporte de errores en proyectos de desarrollo de software.

Metodología del curso

En este curso de **Administración de proyectos de software** se revisarán 14 temas divididos en 3 módulos.

En cada tema, el participante encontrará:

- Una breve explicación del tema que ayudará al estudiante a ampliar su conocimiento.
- Una serie de lecturas y videos obligatorios para una mejor comprensión de los temas.
- Una lista de lecturas y videos recomendados para complementar el estudio del tema.
- Una práctica no evaluable que servirá para repasar los conceptos abordados en el tema.
- Una tarea o actividad de aprendizaje (evaluable) cuyo propósito es aplicar y experimentar con los conceptos estudiados.

A lo largo del curso, el participante debe trabajar en lo siguiente:

- 4 actividades
- 1 avance de evidencia
- 1 entrega final de evidencia

Actividades

Las actividades deben enviarse a través de la plataforma Blackboard en la fecha indicada.

Evidencia

El proyecto final (evidencia) de este curso consiste en realizar los diferentes artefactos necesarios durante cada una de las fases del ciclo de vida del software. Además, se deben crear los diferentes documentos necesarios para trabajar con la metodología PSP. A través de ella el participante demostrará la capacidad de aplicar los conocimientos y habilidades que obtendrá a lo largo de los temas revisados en el curso. Es importante revisar la agenda del curso, pues la mayoría de las **evidencias requieren entregas de avances** que los alumnos tienen que realizar conforme avanza el periodo académico.

Tanto usted como los participantes podrán encontrar información sobre la evidencia dentro del curso, siguiendo esta ruta:

Mi curso > Inicio > Evidencia, como se muestra enseguida:

Administración de proyectos de software Inicio Temas Entregables **Evidencia**

Haz clic en las imágenes para ver la información.



Bienvenida

¡Te doy la más cordial bienvenida al curso de administración de proyectos de software!



¿Qué voy a aprender?

Daremos inicio al curso de administración de proyectos de software, aprenderás los fundamentos para administrar y llevar a cabo un proyecto de desarrollo de sistemas, así mismo conocerás las diferentes actividades, entregables y metodologías que te permitirán tener éxito en el desarrollo de proyectos.



¿Cómo voy a aprender?

El curso está diseñado para que adquieras la competencia señalada por medio de la lectura, que te permitirá adquirir los conocimientos teóricos necesarios como base. A través de estos conocimientos, serás capaz de...

[Seguir leyendo...](#)

NOTA

Es de suma importancia que enfatice en los participantes guardar todos los trabajos y productos que generen durante el curso (actividades, tareas, evidencias). Esto les servirá para conformar un portafolio personal de proyectos, así como para la elaboración de su proyecto integrador (último curso del certificado). Para ello, se le solicita colocar un aviso en Blackboard (sección Announcements), tomando como referencia el siguiente texto:

“Estimado participante, recuerda guardar siempre una copia digital de todos los trabajos, actividades y evidencias que realices en tus cursos. Contar con estos documentos te será de utilidad especialmente para dos fines:

- 1. Conformar un portafolio personal de proyectos, que te servirá como un medio importante para enriquecer tu proyección profesional.*
- 2. Poder elaborar el proyecto integrador de tu certificado (última materia).*

Por lo tanto, asegúrate de respaldar todos tus documentos localmente en un disco duro (computadora + USB flash drive), y de preferencia también almacenarlos en la nube (servicios como Dropbox y Google Drive).”

Temario

Los temas que se abordarán en este curso de certificado son los siguientes:

- Tema 1** Introducción a la administración de proyectos de software
- Tema 2** Ciclo de vida del software
- Tema 3** Artefactos de administración de software
- Tema 4** Diseño y construcción del software
- Tema 5** Introducción a la verificación y validación de software(V&V)
- Tema 6** Implementación y mantenimiento de software
- Tema 7** Fundamentos de Maven
- Tema 8** Pruebas con JUnit
- Tema 9** Calidad de software
- Tema 10** Identificación de riesgos
- Tema 11** Introducción al Personal Software Process (PSP)
- Tema 12** Técnicas de gestión de software

GUÍA PARA EL PROFESOR

Tema 13 Introducción a Team Software Process (TSP)

Tema 14 Trabajando en equipo

Bibliografía y recursos especiales

El libro de texto que deberán adquirir los participantes es el siguiente:

Humphrey, W. (1999). Introduction to the Team Software Process. Estados Unidos: Addison-Wesley Professional.

ISBN: 978-0201477191

Humphrey, W. (2005). PSP(sm): A Self-Improvement Process for Software Engineers. Estados Unidos: Addison-Wesley Professional.

ISBN: 978-0321305497

Ahmed, A. (2012). Software Project Management: A process-driven approach. Estados Unidos: CRC Press.

ISBN: 978-1-4398-4656-8

Evaluación y agendas

La evaluación del curso se estructura de la siguiente manera:

Unidades	Instrumento Evaluador	Puntaje
4	Actividades	40
1	Evidencia	60
Total		100 puntos

Dichos productos se entregarán de acuerdo a la siguiente agenda, definida una vez que se hayan **validado fechas y valores con la información disponible en Servicios en Línea**:

Agenda	
Actividades	Puntaje
Actividad 1	10
Actividad 2	10
Avance 1 de evidencia	30
Actividad 3	10
Actividad 4	10
Entrega final de evidencia	30
Total	100

Banner

Estimado profesor, no olvide capturar las calificaciones de su grupo en las fechas indicadas.

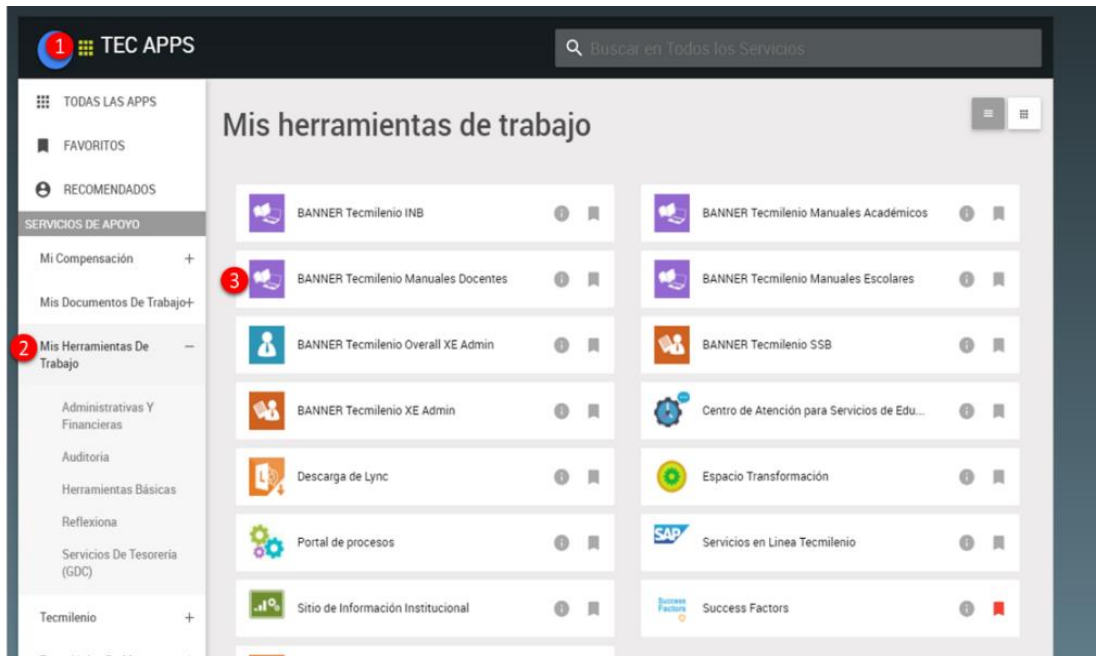
Puede ver un manual para capturar calificaciones siguiendo esta ruta en Mi espacio:

- Mi espacio → TEC Apps → Mis Herramientas de Trabajo → BANNER Tecmilenio Manuales Docentes

GUÍA PARA EL PROFESOR

Si imparte clase de manera presencial, puedes ver un manual para capturar inasistencias siguiendo esta ruta en Mi espacio:

- Mi espacio → TEC Apps → Mis Herramientas de Trabajo → BANNER Tecmilenio Manuales



Notas de enseñanza

Antes de impartir el curso, por favor revise de manera general los datos y conceptos proporcionados en el mismo, con el fin de detectar y, en su caso, poder actualizar y/o enriquecer previamente la información específica al tiempo en que se está impartiendo el curso.

Un aspecto de gran importancia en el desarrollo de los temas es su involucramiento como Facilitador para propiciar que la competencia del curso se cumpla. Además, debe preparar a los participantes para que vayan desarrollando propuestas de soluciones innovadoras a problemas actuales propios del área de estudio.

Enseguida puede revisar las notas de enseñanza generales y por tema para este curso.

Generalidades

Para la impartición de este curso se sugiere:

1. Revisar con tiempo la lista de entregables y la agenda en Banner para saber en qué temas y semanas se deben realizar las actividades.
2. Revisar el manual de Blackboard para conocer las mejores formas de mantener una comunicación constante y efectiva con los estudiantes, despejar dudas y motivarlos. Puedes ver un tutorial de la plataforma en esta liga: <https://drive.google.com/file/d/1PbQgW47gdmob1YQ42NhMXkrwcy9Ys4dl/view?usp=sharing>

Generalidades

3. Revisar periódicamente el foro de dudas en Blackboard para resolver las preguntas e inquietudes de los participantes acerca de las actividades y la evidencia.
4. Motivar al alumno a participar y realizar sus actividades a tiempo.
5. Proveer retroalimentación constante de las actividades que realizan los participantes.
6. Elaborar una Agenda y subirla a la plataforma para que los participantes puedan visualizar de manera esquemática los temas y actividades que deberán revisar cada semana.
7. Recordar (opcionalmente) a los participantes acerca de las entregas de sus actividades por medio de la sección de Avisos de Blackboard.
8. Enriquecer el curso con videos o lecturas adicionales.

Tema 1

Objetivo:

Este tema nos permitirá conocer los conceptos básicos para una correcta administración de proyectos de software.

Notas para la enseñanza del tema:

- Se recomienda al instructor leer el primer capítulo de la referencia bibliográfica.
- Se recomienda al instructor tener varios ejemplos de diferentes metodologías de desarrollo de software.
- Es importante que el instructor pueda descargar la plantilla del chárter del proyecto y pueda comprender su uso y las diferentes secciones del documento.

Tema 2

Objetivo:

Identificar las diferentes etapas del ciclo de vida del software.

Este tema nos permitirá conocer las diferentes fases por las que atraviesa un proyecto de desarrollo de sistemas.

Notas para la enseñanza del tema:

- Se recomienda al instructor que pueda investigar sobre la metodología RUP (*Rational Unified Process*) y el software que lo acompaña.
- Se recomienda al instructor identificar diferentes modelos de desarrollo de software y cuál es el ciclo de vida de dichos modelos. Por ejemplo: iterativo, ágil, entre otros.

Tema 2

Notas para la actividad:

- Existen muchas herramientas para la gestión de requerimientos de software. Como ejemplo, el instructor puede utilizar la llamada “*Open Source Requirements Management Tool*”
- Existen muchas plantillas para la generación de requerimientos de software. La más utilizada es la plantilla Especificación de Requerimientos de Software. Dentro de ella, podemos identificar algunos atributos necesarios para la documentación de los requerimientos de software.

<id>-999	<nombre descriptivo>
[Versión]	<nº versión> (<fecha versión>)
[Dependencias]	<ul style="list-style-type: none">• <requisitos generales de los que depende>• <otros requisitos de los que depende>• ...
Descripción	El sistema deberá almacenar la información correspondiente a <concepto relevante>. En concreto:
Datos específicos	<ul style="list-style-type: none">• <datos específicos sobre el concepto relevante>• ...
[Importancia]	<importancia del requisito para el cliente>.
[Prioridad]	<prioridad del requisito para la dirección del proyecto>
[Estado]	<estado del requisito según el ciclo de vida adoptado por el proyecto >
Comentarios	<comentarios adicionales sobre el requisito>

Tema 3

Objetivo:

Identificar los diferentes artefactos requeridos a través del ciclo de vida del desarrollo del software.

Notas para la enseñanza del tema:

- Se recomienda al instructor que pueda identificar cuál es la funcionalidad de cada uno de los artefactos descritos en el tema.
- Se recomienda también al instructor que pueda documentarse acerca de cada uno de los artefactos que no han sido descritos en el tema.
- Se recomienda al instructor documentarse en la metodología RUP, la cual permite conocer a fondo cada uno de los artefactos y su relación entre sí.

Tema 4

Objetivo:

Identificar la importancia de un correcto diseño del software. Identificar cuál es la importancia de generar código, utilizando métodos y estándares de codificación.

Tema 4

Notas para la enseñanza del tema:

- Es importante que el instructor pueda conocer de antemano cuál es la diferencia entre el modelo top-down y el modelo bottom-up.
- Se recomienda al instructor que pueda identificar diferentes estándares de codificación para diferentes lenguajes de programación.

Tema 5

Objetivo:

Conocer los fundamentos de validación y verificación del software.

Notas para la enseñanza del tema:

- Se sugiere que el tutor en línea provea algunas ligas de información sobre los defectos a través de un anuncio, pues la verificación y validación está orientada a encontrar fallas y defectos a lo largo del ciclo de vida de desarrollo, y el proceso de V&V permite que en cada fase exista un porcentaje de remoción de defectos que disminuye la cantidad de defectos inyectados que migran de una fase a otra.
- Es importante que se haga reflexionar al alumno que la disciplina de V&V minimiza el costo de un proyecto de software por cada defecto encontrado en cada una de las fases, de manera que mientras más defectos se encuentren en fases tempranas, el costo de arreglar el producto será menor; es decir, un defecto encontrado en la fase de requerimientos puede consumir 1 hora hombre de trabajo para poder arreglar el problema, no obstante, si este defecto se encontrara cuando la aplicación se encuentra en producción, podría costar de 50 a 100 horas hombre corregirlo. Se sugiere que el tutor elabore un mensaje de 2 a 3 párrafos con un ejemplo real de este concepto para lograr alcanzar el objetivo de la reflexión en el alumno en línea.
- Se recomienda al profesor presencial hacer énfasis en el trabajo conjunto de la calidad de software con el proceso de verificación y validación (V&V), el alumno debe de entender que es esta disciplina el factor principal que permite generar proyectos de software funcionales con la más alta calidad.
- Se sugiere que el profesor explique brevemente información sobre los defectos, pues la verificación y validación están orientadas a encontrar fallas y defectos a lo largo del ciclo de vida de desarrollo, y el proceso de V&V permite que en cada fase exista un porcentaje de remoción de defectos que disminuye la cantidad de defectos inyectados que migran de una fase a otra.
- También es muy importante que se haga reflexionar al alumno que la disciplina de V&V minimiza el costo de un proyecto de software por cada defecto encontrado en cada una de las fases, de manera que mientras más defectos se encuentren en fases tempranas, el costo de arreglar el producto será menor; es decir, un defecto encontrado en la fase de requerimientos puede consumir 1 hora hombre de trabajo para poder arreglar el problema, no obstante, si este defecto se encontrara cuando la aplicación se encuentra en producción, podría costar de 50 a 100 horas hombre corregirlo.

Notas para la actividad:

- Es importante que el instructor pueda conocer el uso de las pruebas de aceptación de usuario.

Tema 5

- El instructor puede documentarse utilizando la siguiente bibliografía:
Bordo, V. (2010). Overview of User Acceptance Testing (UAT) for Business Analysts (BAs). Recuperado de <https://www.develop.com/useracceptancetests>
- Se recomienda al instructor que pueda verificar con anticipación algunas herramientas de gestión de incidentes. Entre algunos ejemplos podemos encontrar Jira, Mantis, Bugzilla, Trac, IBM Rational ClearQuest.
- El software de gestión de incidencias nos permite realizar un monitoreo de todos aquellos errores identificados durante las actividades de V&V. A través de este software, los testers pueden documentar los errores identificados, permitiendo llevar un control de la prioridad con la que debe ser solucionado y de cuándo ha sido solucionado.

Tema 6

Objetivo:

Identificar la importancia de las actividades de implementación y mantenimiento de sistemas.

Notas para la enseñanza del tema:

- Se recomienda al instructor que pueda documentarse acerca del plan de despliegue.
- Es recomendable que el instructor tenga experiencia previa (no es indispensable) con algún software de construcción de software: Jenkins, Bamboo, Maven. Esto permitirá al instructor dar ejemplos de cómo un software puede ser construido de forma automática y desplegado a ambientes de desarrollo y producción.

Tema 7

Objetivo:

Identificar la importancia de las herramientas de integración continua en los proyectos de software.

Notas para el avance 1 de evidencia:

- Se recomienda al instructor leer y conocer cada una de las plantillas solicitadas.
- Dentro de cada una de las plantillas anexas, se encuentran las instrucciones de cómo se debe realizar el documento. Se recomienda que el instructor pueda leer cada una de las instrucciones antes de comenzar el tema.

Tema 8

Objetivo:

Identificar la importancia de JUnit para la realización de pruebas automáticas.

Notas para la enseñanza del tema:

- Se recomienda que el profesor presencial lleve preparados ejemplos distintos a los del curso para complementar la explicación y que los alumnos entiendan mejor el uso de JUnit y JUnitPerf.
- Se recomienda al profesor tener preparado el software necesario para realizar los ejemplos del tema. (Eclipse, Java JDK).

Tema 9

Objetivo:

Este tema nos permitirá conocer qué es la calidad de software, la diferencia entre validación y verificación de software, así también, las formas de prevenir y reducir riesgos en el software.

Notas para la enseñanza del tema:

- Se sugiere leer el capítulo 1 del libro de la bibliografía del tema.
- Se sugiere que se pueda leer acerca del estándar ISO-9126 antes de comenzar el tema.
- Se sugiere que se pueda identificar previamente la diferencia entre validación y verificación del software. Existen diferentes técnicas para realizar cada una de estas acciones, por lo que se recomienda identificar algunos ejemplos previamente.

Tema 10

Objetivo:

Este tema nos permitirá identificar estándares y metodologías que sirven como apoyo en el desarrollo de software y permiten establecer métricas de calidad de software.

Notas para la enseñanza del tema:

- Se recomienda identificar ejemplos de metodologías y estándares de calidad de software.
- Se recomienda leer sobre el ISO 9126.
- Se recomienda leer y documentar sobre el modelo CMMI.

Tema 11

Objetivo:

A través de este tema, analizaremos las bases del Proceso Personal de Software. Durante el tema, analizaremos cuales son los factores clave para el éxito de un proyecto de desarrollo de software.

Notas para la enseñanza del tema:

- Es muy importante que el instructor se documente acerca de cada uno de las fases de PSP (PSP0, PSP0.1, etc.), y cuáles son los entregables básicos de cada una de las fases.
- Se recomienda que el instructor pueda leer los primeros seis capítulos de la referencia bibliográfica.
- Se recomienda al instructor identificar los scripts necesarios para la fase PSP0, el plan de trabajo recomendado, el registro de tiempos y el registro de defectos.

Notas para la actividad:

- Se recomienda al instructor que pueda verificar la bibliografía del tema, ya que los ejemplos se encuentran localizados en el tema 2 del libro.

Tema 12

Objetivo:

A través de este tema, continuaremos analizando los conceptos básicos de la metodología PSP.

Notas para la enseñanza del tema:

- Es importante que el instructor pueda leer el material marcado como lecturas obligatorias, ya que en este documento se da una explicación detallada del uso del método de estimación PROBE.
- Se recomienda al instructor identificar el script de estimación PROBE.

Tema 13

Objetivo:

A través de este tema descubrirás las bases para poner en práctica el Proceso Personal de Software (PSP) en un ambiente de colaboración en equipo.

Notas para la enseñanza del tema:

- Es importante que el instructor pueda leer el documento marcado como lectura obligatoria, ya que en este se da una breve introducción a la metodología.
- Es también importante que el instructor pueda comprender con anterioridad las ocho etapas del proceso TSP y cuáles son los entregables de cada una de ellas. Dichos entregables se encuentran en la bibliografía del tema.

Tema 13

- Es muy importante que el instructor motive a los alumnos a desarrollar la evidencia del tema con tiempo, ya que necesitarán desarrollar diferentes entregables de las metodologías PSP y TSP.

Notas para la actividad:

- Se recomienda leer los temas 4 al 12 de la referencia bibliográfica, ya que en dicha bibliografía se encuentra el material solicitado por la actividad.

Tema 14

Objetivo:

A través de este tema, continuaremos analizando los conceptos básicos de la metodología PSP.

Notas para la enseñanza del tema:

- Es importante que el instructor pueda leer el material marcado como lecturas obligatorias, ya que en este documento se da una explicación detallada del uso del método de estimación PROBE.
- Se recomienda al instructor identificar el script de estimación PROBE.