

# Soluciones a los ejercicios de autocomprobación

## 1. Significado e importancia de la gestión de proyectos

### Actividad práctica

Las metodologías ágiles mejoran la comunicación entre las partes del equipo. Dado que en el ámbito de las tecnologías todo cambia muy rápido, el alcance de los proyectos también. Por tanto, solo con las metodologías ágiles podremos actuar dinámicamente respecto a estos cambios.

1. a) Planificación, Coste, Alcance.
2. c) La segunda mitad del siglo XX.
3. b) Dar valor a cada entregable.
4. b) La gestión ágil.
5. **Falso.** Todo proyecto por pequeño que sea necesita de una gestión y no solo una mera ejecución. Siempre necesitaremos planificar, gestionar riesgos, hablar con proveedores, clientes, etc.
6. **Verdadero.**
7. **Verdadero.**
8. **Verdadero.**
9. Con un equipo multidisciplinar rompemos con resistencias y criterios preconcebidos y ampliamos el punto de vista en los análisis y decisiones. Los integrantes aportan más valor y aprenden unos de otros.
10. **Procesos de inicio.** Prepararemos el enunciado preliminar del proyecto y el plan de gestión.  
**Procesos de planificación.** Estudiaremos el alcance del mismo y las actividades que deberemos llevar a cabo, elaboraremos los cronogramas (calendarios con la planificación temporal) y calcularemos los costes y riesgos asociados.  
**Procesos de ejecución.** Empezaremos a convertir en acciones las ideas y la planificación que tenemos sobre el papel.

**Procesos de supervisión y control.** Paralelamente a los procesos de ejecución, monitorizaremos la ejecución del proyecto.

**Procesos de cierre.** Las actividades relacionadas con estos procesos son necesarias para comprobar que los objetivos han sido alcanzados y así proceder al cierre administrativo del proyecto.

## 2. Características

### Actividad práctica

La respuesta depende totalmente del alumno y de su creatividad, pero podemos indicar cinco posibles riesgos:

- A. La plataforma de desarrollo puede darnos dificultades.  
Causa: utilizamos una plataforma de desarrollo nueva.  
Impacto: retrasos.  
Probabilidad: media.  
Importancia del impacto: media.
- B. El interfaz táctil nos puede quedar poco usable.  
Causa: nunca hemos desarrollado aplicaciones con interfaz táctil.  
Impacto: en el alcance (uno de los requisitos de realizar una aplicación usable).  
Probabilidad: baja.  
Importancia del impacto: alta.
- C. Nuestra aplicación puede acabar siendo ilegal.  
Causa: la legislación sobre el ámbito de nuestra aplicación está en proceso de cambio.  
Impacto: alcance (todo el proyecto se puede ver cancelado).  
Probabilidad: baja.  
Importancia del impacto: alta.
- D. A lo mejor necesitamos contratar un experto.  
Causa: un módulo de la aplicación necesita algoritmos basados en redes neuronales.  
Impacto: costes.  
Probabilidad: alta.  
Importancia del impacto: media.
- E. Uno de los requisitos puede que deba ser aplazado para posteriores fases.  
Causa: la planificación ha sido muy optimista en este aspecto.  
Impacto: retrasos, alcance.  
Probabilidad: alta.  
Importancia del impacto: alta.

1. b) La probabilidad de producirse y la importancia del impacto.

2. c) La planificación del proyecto.
3. a) Las características y funcionalidades del *software*, los criterios de aceptación, los entregables y las exclusiones del producto.
4. b) Riesgos, planificación y alcance.
5. **Falso.** Debemos adaptar continuamente la planificación para adecuarla a los acontecimientos.
6. **Verdadero.**
7. **Verdadero.**
8. **Falso.** El presupuesto forma parte de otras áreas de conocimiento; sin embargo, en el alcance hablamos de los objetivos y de todo lo que cuelga de los mismos, sin hablar de recursos o presupuestos.
9. Primero es necesario identificar qué tipo de recursos vamos a necesitar para llevar a cabo nuestro proyecto, cuántos y cuánto tiempo vamos a utilizarlos.  
El siguiente paso sería la obtención de los recursos materiales, la asignación pertinente al proyecto y la retirada en la finalización de su utilidad.  
  
Y, por último, realizar un seguimiento de los recursos, comprobando que cada uno de ellos se está utilizando o está trabajando en la tarea asignada durante el tiempo establecido y que los recursos disponibles realmente son suficientes para poder cumplir con los objetivos.
10. ¿Cuánto tiempo se requerirá? ¿Cuántas personas serán necesarias? ¿Cuánto costará la tarea? ¿Qué tareas deben terminarse antes de empezar otras? ¿Pueden solaparse algunas de ellas?

### 3. Partes interesadas

#### Actividad práctica

Aunque la respuesta depende totalmente del alumno y de su creatividad, estas podrían ser diez características de nuestro *target*:

- Es de mediana edad
- Le interesan los deportes
- Es hombre
- Vive en la gran ciudad
- Es soltero
- Tiene carrera universitaria
- No utiliza el transporte público

- Le gusta la música
- No mira la televisión
- Es crítico en cuestiones políticas

1. c) Las dos respuestas anteriores son correctas.
2. b) Documentar y fijar el funcionamiento de la aplicación que desarrollaremos.
3. c) Ninguna de las respuestas anteriores es correcta.
4. c) El equipo de gestión.

**5. Verdadero.**

- 6. Falso.** El equipo de desarrollo debería reunirse semanalmente para hacer seguimiento de las tareas realizadas y planificar las actividades de la semana.

**7. Verdadero.**

- 8. Falso.** En proyectos pequeños, con recursos limitados, los desarrolladores suelen ser los gestores; los mismos que han impulsado el proyecto.

- 9.** El equipo de gestión debe saber en todo momento el estado de las tareas: las que se han ejecutado, las que se han retrasado, los problemas que ha habido, las dudas que tiene el equipo de desarrollo, etc. Todo ello es básico para poder realizar la gestión del proyecto.

- 10.** Cuanta más formación técnica tenga el cliente, más conocerá las limitaciones y posibilidades de la tecnología que se vaya a utilizar para desarrollar el proyecto. Además, su lenguaje será más preciso, de manera que el entendimiento será mayor.

## 4. Gestión de código

### Actividad práctica

Una división de la parte de usuario podría ser la siguiente:

- **Módulo 1:** se encarga de registrar a un usuario nuevo.
- **Módulo 2:** se encarga de comprobar que un usuario existe.
- **Módulo 3:** si el usuario existe, se encarga de mostrarle un panel privado con las opciones que estén disponibles para el usuario.

- **Módulo 4:** permite la modificación de los datos personales del usuario.
- **Módulo 5:** desconecta al usuario de la aplicación.

1. b) El volumen de líneas de código que puede ocupar el proyecto.

2. a) Miles de líneas de programación objeto.

3. c) Tiene aproximadamente cinco errores por KLOC.

4. c) Ninguna de las dos respuestas anteriores es correcta.

5. **Verdadero.**

6. **Falso.** Una práctica utilizada es la de tomar como referencia proyectos ya realizados que sean similares al nuestro.

7. **Verdadero.**

8. **Verdadero.**

9. Es el hecho del incremento del coste del producto durante la ejecución del proyecto.

10. Los miembros del equipo de trabajo deben interactuar entre sí para poder realizar la función de todo el código en conjunto. Si en el momento de la planificación no se han tenido en cuenta posibles causas o situaciones que debían estar contempladas, aparece trabajo extra. Por diferentes causas, se ha sobrepasado el margen de errores aceptable para entregar el proyecto con una calidad mínima.

## 5. Gestión de incidencias y errores

### Actividad práctica

Debido a que la incidencia de mayor impacto implica tener recursos asignados durante más tiempo, por su dificultad sería recomendable asignar un mínimo de recursos a la incidencia fácil y el resto de recursos disponibles a estudiar y solucionar la crítica. Con esto cerraríamos una incidencia y podríamos tener todos los recursos centrados en una. De esta manera no tenemos dos trabajos pendientes.

1. c) Poder asociar una prioridad a una incidencia.

2. c) Las dos respuestas anteriores son correctas.

- 3. b) Encontrar soluciones para el error.
- 4. c) Cualquier evento que no forma parte de la operación estándar de un servicio y que causa, o puede causar, una interrupción o una reducción de calidad del mismo.
- 5. **Verdadero.**
- 6. **Falso.** Es la primera fase que se debe realizar.
- 7. **Falso.** Un error conocido es la siguiente fase de un problema.
- 8. **Verdadero.**
- 9. Es la fase en la que, después de solucionar la incidencia, se ha de cumplimentar la base de datos con toda la información en la que se hable de la misma. Una vez implementada la solución, esta fase también nos indica que debe realizarse una verificación para comprobar que no sigue reproduciéndose el error y no se han causado efectos secundarios.
- 10. Es importante porque estos informes nos pueden revelar en cualquier momento si el proyecto cumple con los requisitos de calidad establecidos en la planificación del mismo; nos pueden aportar una idea del grado de satisfacción del cliente; nos facilitan la tarea de documentación estadística, etc.

## 6. Documentación de proyecto y código

### Actividad práctica

La documentación interna es muy útil tanto para el programador que la crea como para el resto de miembros del equipo, ya que les facilita la lectura y comprensión del código. La principal característica de la documentación interna es que tiene que ser capaz de hacer que el código sea autodocumentado. Esto significa que, a la hora de programar, se han de tener en cuenta aspectos como, por ejemplo, asignar nombres a nuestras funciones que sean representativos de aquello a lo que se corresponden y que sean autoexplicativos y no nombres sin sentido.

- 1. c) Lo más clara posible para que la comprenda cualquier tipo de lector.
- 2. b) Documentos de viabilidad económica, rentabilidad, etc.
- 3. a) Explicar partes de código confusas.

4. c) Ninguna de las respuestas anteriores es correcta.

5. Verdadero.

6. Falso. Cada lenguaje puede usar una manera diferente para comentar.

7. Falso. Los comentarios de código no pueden ocupar más de una o dos líneas.

8. Verdadero.

9. Son frases de texto normales, que soportan cualquier carácter, en las que podemos escribir lo que queramos y que no son interpretadas como una línea de código.

10. Este informe especifica todos los puntos relacionados con el proyecto para su implementación. Puede contemplar datos como el plan estructurado del proyecto, la gestión del personal, etc.

## 7. Introducción a las metodologías ágiles: Scrum como marco de desarrollo

### Actividad práctica

El primer paso sería crear y organizar el equipo de trabajo. Para ello vamos a necesitar definir quién será el *product owner*. Es posible que el cliente elija a una persona para adoptar este rol. Elegiremos a las personas que van a formar parte del equipo de desarrollo, entre 3 y 9 personas, y quién será el *Scrum master*.

1. b) Una planificación flexible para adaptarse a los cambios.

2. c) Herramientas y roles para realizar un seguimiento del proyecto.

3. c) Entre 7 y 11 para facilitar la comunicación y colaboración.

4. a) Se realizan reuniones diarias y breves para poner en común todas las tareas.

5. Falso. Uno de los valores fundamentales de las metodologías ágiles es que la colaboración del cliente esté por encima de la negociación del contrato.

**6. Falso.** Las metodologías ágiles apuestan por una planificación flexible que pueda adaptarse a los cambios.

**7. Verdadero.**

**8. Verdadero.**

**9.** Los individuos e interacciones están por encima de los procesos y herramientas, el funcionamiento del *software* está por encima de la documentación, la colaboración está por encima del contrato y la respuesta ante cambios está por encima de un plan.

**10.** Es responsable de que se adopten los valores, prácticas y principios de *Scrum*, además de guiar la colaboración entre los integrantes del equipo.

## 8. Diseño de interfaces para aplicaciones móviles con Pencil

### Actividad práctica





1. b) Un boceto donde se representa visualmente la estructura y funcionalidad de una aplicación web.
2. c) Las dos anteriores son correctas.
3. c) ser una aplicación *open source*.
4. a) Windows, Linux y macOS.
5. **Falso.** un *wireframe* representa una aplicación web o móvil.
6. **Falso.** Se deben usar en las primeras fases del proceso de desarrollo para definir el contenido y la posición de los diversos elementos de la aplicación.
7. **Verdadero.**
8. **Verdadero.**
9. Puede ralentizar el desarrollo e improvisar sobre la marcha la estructura y funcionalidad.
10. Es de código abierto (*open source*), plantillas y formas predefinidas, exportación en múltiples formatos, agregar nuevas colecciones y actualizaciones constantes.

# Índice

<b>Introducción.....</b>	<b>5</b>
 <b>1. Significado e importancia de la gestión de proyectos....</b>	<b>6</b>
1.1. La gestión clásica.....	7
1.2. La gestión ágil.....	9
<b>Resumen.....</b>	<b>12</b>
<b>Ejercicios de autocomprobación.....</b>	<b>13</b>
 <b>2. Elementos de la gestión de proyectos.....</b>	<b>15</b>
2.1. El alcance del proyecto.....	15
2.1.1. Recopilación de requerimientos .....	15
2.1.2. Definición del alcance .....	16
2.1.3. Descomposición del alcance en tareas .....	16
2.2. La planificación del proyecto.....	17
2.2.1. Definición de las actividades .....	18
2.2.2. Secuenciar las actividades .....	18
2.2.3. Determinar la duración de las actividades .....	18
2.2.4. Desarrollar el cronograma .....	19
2.3. Los riesgos del proyecto.....	20
2.3.1. Identificar los riesgos .....	20
2.3.2. Cuantificar los riesgos .....	21
2.3.3. Respuesta planificada a los riesgos .....	21
<b>Resumen.....</b>	<b>23</b>
<b>Ejercicios de autocomprobación.....</b>	<b>24</b>
 <b>3. Partes interesadas de un proyecto.....</b>	<b>26</b>
3.1. El cliente o <i>target</i> .....	26
3.1.1. Importancia y significado del cliente y el <i>target</i> .....	26
3.1.2. Relaciones con el resto de partes.....	27
3.2. El equipo de gestión.....	28
3.2.1. Importancia y significado del equipo de gestión.....	28
3.2.2. Relaciones con el resto de partes.....	29
3.3. El equipo de desarrollo.....	30
3.3.1. Importancia y significado del equipo de desarrollo.....	30
3.3.2. Relaciones con el resto de partes .....	31
<b>Resumen.....</b>	<b>32</b>
<b>Ejercicios de autocomprobación.....</b>	<b>33</b>
 <b>4. Gestión de código.....</b>	<b>35</b>
4.1. Introducción.....	35
4.2. Métrica de líneas de código.....	35
4.3. Uso de las líneas de código.....	36
4.4. Tamaño de los proyectos.....	37

4.5. Errores de código y líneas de programa.....	38
4.6. Proyectos nuevos: cantidad de líneas.....	38
4.7. Factores a tener en cuenta en la gestión de código.....	39
<b>Resumen.....</b>	<b>41</b>
<b>Ejercicios de autocomprobación.....</b>	<b>42</b>
 <b>5. Gestión de incidencias y errores.....</b>	 <b>44</b>
5.1. Gestión de incidencias.....	44
5.2. Gestión de errores o problemas.....	48
<b>Resumen.....</b>	<b>53</b>
<b>Ejercicios de autocomprobación.....</b>	<b>53</b>
 <b>6. Documentación de proyecto y código.....</b>	 <b>55</b>
6.1. Introducción.....	55
6.2. Documentación de proyecto.....	55
6.3. Documentación de código.....	56
<b>Resumen.....</b>	<b>59</b>
<b>Ejercicios de autocomprobación.....</b>	<b>60</b>
 <b>7. Introducción a las metodologías ágiles: Scrum como marco de desarrollo.....</b>	 <b>62</b>
7.1. Qué son las metodologías ágiles.....	62
7.2. Fundamentos de Scrum.....	63
7.2.1. Desarrollo iterativo .....	63
7.2.2. Prioridades .....	64
7.2.3. Control del proceso .....	64
7.2.4. Colaboración .....	64
7.3. Personas y roles.....	64
7.4. Planificación.....	65
7.5. <i>Sprints</i> .....	67
<b>Resumen.....</b>	<b>69</b>
<b>Ejercicios de autocomprobación.....</b>	<b>70</b>
 <b>8. Diseño de interfaces para aplicaciones móviles con Pencil.....</b>	 <b>72</b>
8.1. Pencil.....	73
8.2. Creación de un <i>wireframe</i> .....	74
<b>Resumen.....</b>	<b>78</b>
<b>Ejercicios de autocomprobación.....</b>	<b>79</b>
 <b>Soluciones a los ejercicios de autocomprobación.....</b>	 <b>81</b>
 <b>Índice.....</b>	 <b>90</b>