

# Taller de Proyectos I en Ingeniería de Sistemas e Informática

---

Guía de Trabajo

---



**Universidad Continental**

Material publicado con fines de estudio

Código: ASUC01584



## **Presentación**

Taller de Proyectos I, es una asignatura que permite al estudiante obtener la experiencia de identificar un problema y proponer consecuentemente una solución que requiera de Tecnologías de la Información, de ahí su importancia ya que incentiva en el estudiante a proponer ideas innovadoras que hagan uso de los conocimientos y habilidades adquiridos en semestres previos.

Por ese motivo los contenidos diseñados se enfocan en la Iniciación y planificación del proyecto, el diseño de la solución y su correspondiente construcción, pasando por la debida gestión del proyecto, culminando en las Pruebas y planes de aseguramiento de calidad.

Al finalizar la asignatura el estudiante será capaz de identificar un problema u oportunidad, diseñar la solución, plantear el proyecto relacionado y llevar a cabo el proceso de gestión y construcción respectivo.

Dada la naturaleza de la asignatura es importante que tengas tempranamente un panorama de la secuencia de la asignatura, complementado con la aplicación adecuada de la metodología propuesta,

*El autor / La autora*



## Primera unidad Semana 1 – Sesión 2

### Evaluación del Impacto e importancia

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 1	Fecha: ...../...../..... Duración: 60 min

I. **Propósito:** Usar la plantilla de conveniencia para evaluar del Impacto de tu proyecto y su respectiva importancia.

#### II. Descripción de la actividad a realizar (caso.)

Efectúa la lectura del artículo de Robby Ralston , link: <https://elcomercio.pe/especial/zona-ejecutiva/tendencias/proceso-creativo-y-innovacion-segun-roby-ralston-noticia-1992048>

Considerando las ideas de proyecto que ha planteado el equipo completen la matriz de conveniencia-

#### III. Procedimientos

Apellidos y Nombres	Código

Fecha	
-------	--

#### Instrucciones:

1. Completa con una breve descripción cada una de las ideas a evaluar.
2. El equipo debe Valorar cada idea en cada uno de los aspectos indicados en la escala de 1 a 5. Se recomienda la mayor honestidad posible.



	<b>IDEA 1:</b>	<b>IDEA 2:</b>	<b>IDEA 3:</b>
Conocimientos teóricos y técnicos referentes a lo planteado.			
Impacto de la solución a implementar			
Preferencia personal por el tema			
Factibilidad técnica de la solución			
Valoración de la innovación de la idea*			
Descripción del Sustento de la valoración de innovación			
<b>Total obtenido</b>			

\* Considere para la evaluación de creatividad lo fundamentado en el artículo de Robby Ralston con referencia a ¿Cuál es el proceso? Y la distinción de innovación.



## Semana 2 – Sesión 2

### Especificación de requisitos de software

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 1	Fecha: ...../...../2022 Duración: 60 min

I. **Propósito:** Obtener el listado formal de requerimientos funcionales de su proyecto.

#### II. Descripción de la actividad a realizar (caso.)

La Pila de Producto (Product Backlog) es el instrumento metodológico del marco de trabajo Scrum, que se usa para listar las características (Features) o funcionalidades del software a desarrollar, para priorizarlas de acuerdo a las necesidades del área de negocio. Su contenido se desarrolla a partir de las historias de usuario identificadas por el dueño de producto (Product Owner).

La pila de producto permitirá tener visualización de las funcionalidades a desarrollar, priorizar las características del software según las necesidades del negocio, dejar registrado el esfuerzo necesario para desarrollar la historia y asignarla a una iteración (Sprint). Para ello, se deben seguir las **reglas de administración de la pila de producto**.

Visualizar el siguiente recurso: 1 **Scrum Ejemplo Práctico Product Backlog**, en <https://www.youtube.com/watch?v=573jB8DQPEU> ( ConsultoraPMO, 3)

#### III. Procedimientos

- Descarga el archivo plantilla\_productBackLog desde:  
<https://github.com/y Marquez/tallerProyectos/>

#### Instrucciones:

1. Efectúa la descarga del archivo plantilla y de acuerdo a las instrucciones completa ÚNICAMENTE las columnas: Identificador (ID) de la Historia, Enunciado de la Historia y Alias.
2. Sigue la estructura del enunciado que tiene la forma de :  
Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado], los elementos dentro de los corchetes son los que deben ser reemplazados.



3. Si la naturaleza del proyecto lo permite listar los requerimientos mínimamente considerando un rol de USUARIO FINAL y un rol de USUARIO ADMINISTRADOR
4. Evitar el uso de la palabra PODER en el enunciado de la tarea.

	A	B	C	D
1	<b>Pila de Producto (Product Backlog)</b>			
2				
3				
4		<b>Identificador (ID) de la Historia</b>	<b>Enunciado de la Historia</b>	<b>Alias</b>
5		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
6		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
7		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
8		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
9		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
10		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
11		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
12		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
13		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
14		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
15		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
16		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
17		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	
18		XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	



## Semana 3 – Sesión 2

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 1	Fecha: ...../...../2022 Duración: 60 min

### Características y criterios de evaluación de un PMV

I. **Propósito:** Obtener el listado formal de requerimientos funcionales de su proyecto.

#### II. Descripción de la actividad a realizar (caso.)

La hoja de ruta del producto (Agile Product RoadMap) describe como se visiona la evolución del producto a lo largo de varias salidas a producción, similar a un plan de producto, ve más allá de un proyecto o Release individual, describiendo la ruta que seguirá el producto en los próximos 12 meses o mas.

A diferencia de la lista de producto (Product Backlog) el cual es un documento ideal para capturar ideas y requerimientos, la hoja de ruta es útil para describir como se desarrollará el producto en el futuro.

En la Plantilla Scrum para la hoja de ruta de producto, podrás ver una hoja de ruta que divide los flujos de trabajo en las funcionalidades que se van a desarrollar y presenta la línea de tiempo de las fases del ciclo de desarrollo y pase a producción.

#### III. Procedimientos

- Descarga el archivo [Hoja de ruta del producto.pptx](https://github.com/y Marquez/tallerProyectos) desde: <https://github.com/y Marquez/tallerProyectos>

#### Instrucciones:

1. Efectúa la descarga del archivo plantilla completa ÚNICAMENTE las funcionalidades logradas en el transcurso de los meses.



Proyecto:

<Nombre de la empresa>	1er. mes	2do. mes	3er. mes	4to. mes
<Flujo de trabajo 1>	Funcionalidad 1	Funcionalidad 2	Funcionalidad 3	
<Flujo de trabajo 2>	Funcionalidad 4		Funcionalidad 5	
<Flujo de trabajo 3>	Funcionalidad 7	Funcionalidad 8	Funcionalidad 9	
<Flujo de trabajo 4>	Funcionalidad 10		Funcionalidad 11	Funcionalidad 12



## Semana 4 – Sesión 2

### Estimación del esfuerzo

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 1	Fecha: ...../...../2022 Duración: 60 min

I.

I. **Propósito:** Estimar formalmente el desarrollo de cada tarea de su proyecto .

#### II. Descripción de la actividad a realizar (caso.)

Sin entrar en el tema de si se debe estimar o no en el desarrollo de software, es cierto que tener una estimación, aunque sea a alto nivel nos será útil para tomar mejores decisiones a la hora de ordenar y priorizar el backlog. Por ejemplo si tenemos historia A y B, y la historia A aporta valor 100 con esfuerzo 100 y la historia B aporta valor 90 pero esfuerzo 10, seguramente sea más interesante hacer primero B antes que A. (casanova, 2022)

#### La estimación ágil

Objetivo	Descripción
Tener diferentes puntos de vista	Está claro que 7 pares de ojos ven más que uno. En la estimación ágil se busca la democracia, que todo el mundo participe y diga la suya. Si hay discusión mejor, ya que de lo que se trata es de avanzar todo lo posible los problemas y tenerlos en cuenta desde el inicio. No debería haber personas con una voz y voto más fuertes que los demás.
Detectar posibles tareas ocultas y posibles obstáculos	La sesión de estimación es una de las primeras oportunidades de detectar riesgos que pueden comenzar a tratarse para que no se conviertan en impedimentos.
Tener una visión compartida de la que se viene encima	Es muy útil haber participado en las estimaciones para después hacer las planificaciones. Si conoces el tamaño de las historias y tareas es mucho más sencillo comprometerte con un plan de trabajo que si las estimaciones te vienen "impuestas".
Tener estimaciones más realistas (no más precisas)	La idea es que entre todos los numeritos que salga sea lo más cercano a la realidad posible. Para ello necesitas eliminar la presión contractual, esto es, dejar margen para equivocarse y evitar así introducir buffers "inconscientes" por si acaso. Lo que buscamos es realismo, no precisión, es decir, quiero saber si una historia serán 3 o 5 días, si me dices que tardarás 26,5 horas dudaré de que hayas hecho un buen ejercicio de estimación.



### Planning Poker:

Se trata de una dinámica ágil en la que se reúne el equipo con una baraja de Poker modificada y se hacen rondas de estimación con ayuda de estas cartas.

### Material necesario

Será necesario que cada participante tenga una baraja de Planning Poker. Puedes descargar e imprimir tu propia baraja en este enlace: [https://www.scrumdesk.com/agile-estimation-principles/planning-poker-cards/#lightbox\[postimages\]/0](https://www.scrumdesk.com/agile-estimation-principles/planning-poker-cards/#lightbox[postimages]/0)

En cada baraja hay una pseudo-secuencia de Fibonacci modificada. Se recomienda quedarse con solo las primeras cartas para evitar estimar tareas demasiado grandes. Así cada participante tendrá las siguientes cartas: 0,1/2, 1, 2, 3, 5, ? e infinito. El cero significa que la historia ya está hecha o no requiere ningún esfuerzo, el interrogante significa que nos falta información para estimar esa historia o tarea y el infinito es que es demasiado grande y hay que dividirla en tareas más pequeñas.

### Trabajo previo

El equipo debe tener claro en qué unidades va a estimar. Es posible que para estimaciones de alto nivel o de épicas o historias prefiera estimar en semanas ideales, mientras que para historias más pequeñas y tareas de bajo nivel prefiera estimar en días ideales. Un día ideal es el trabajo que consigue una persona en un día en el que no tiene interrupciones de ningún tipo y todo le sale a la primera. También es posible que decidas estimar en una medida relativa como puntos historia. En cualquier caso, lo importante es que todos sepan en qué medida se estima.

También será importante que todos sepan qué se incluye y qué no en la estimación, si se incluye alguna documentación, tests unitarios, tests de integración y cualquier otra cosa que forme parte del desarrollo. Yo personalmente recomiendo incluirlo todo en la estimación de las historias y separar solo la documentación (no los tests) al desglosarlo en tareas.

## III. Procedimientos

- Descarga el archivo Requerimientos Funcionales.xlsx desde:  
<https://github.com/ymarquezs/tallerProyectosI>

### Instrucciones:

2. Completa la hoja ScrumSprint, primero copia cada uno de los requerimientos del productbacklog, y de inmediato su respectivo diseño, a continuación se debe aplicar la técnica de estimación de esfuerzo de cada tarea, lo cual debe registrarse en la columna **Duración**



Identificador (ID) de la Historia	Requerimientos	PRIORIDAD	DURACIÓN
2021-Hu-001	Como cliente quiero registrar mi ...	Alta	10 días
2021-Hu-002	Como cliente quiero contar con ...	Baja	2 días

3. Considere que normalmente a cada requerimiento funcional le corresponde una interface, la cual se debe especificar en una hoja codificada con el número del requerimiento. Por lo que desde la hoja GANT se debe vincular el requerimiento funcional y agregar la **descripción**, así como las **Restricciones** necesarias detectadas.

The screenshot shows a GANT chart interface with a requirement card for 'Req 1'. The card contains the following information:

- Descripción:** Como cliente quiero registrar mi ...
- Restricciones:**
  - 1 En los campos de nombres y apellidos solo se pueden ingresar letras
  - 2 La información de la sección "Plan de suscripción" solo podrá ser modificada por el administrador
  - 3 Solo para usuarios de entre 15 a 55 años

On the right side, there is a preview of a web page titled 'Perfil' with the following sections:

- Datos personales:** Nombres, Apellidos, Fecha de nacimiento, and Genero (Masculino/Femenino).
- Información de contacto:** Celular and Correo.
- Plan de suscripción:** Fecha de suscripción: 15/09/2021 and Fecha de caducidad: 15/11/2021.

The GANT chart at the bottom shows a task for 'Req 1' with a red circle around it.



## Segunda unidad

### Semana 5 – Sesión 2

#### Planificación del proyecto

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 2	Fecha: ...../...../2022 Duración: 60 min

I. **Propósito:** Planificar formalmente el desarrollo de su proyecto.

#### II. Descripción de la actividad a realizar (caso.)

La gestión ágil de proyectos (Agile Project Management o APM) es un enfoque iterativo para planificar y guiar los procesos del proyecto, que lo divide en ciclos más pequeños llamados sprints o iteraciones.

Al igual que en el Desarrollo de Software Ágil (Agile), un proyecto ágil se completa en pequeñas secciones. En el desarrollo de software ágil, por ejemplo, una iteración se refiere a un solo ciclo de desarrollo. Cada sección o iteración es revisada y criticada por el equipo del proyecto, que debe incluir representantes de las diversas partes interesadas del proyecto. Los conocimientos adquiridos a partir de la crítica de una iteración se utilizan para determinar cuál debería ser el siguiente paso en el proyecto.

En esta sesión deberá planificar la ejecución de su proyecto, pero considerando:

- Que se tiene una fecha límite de entrega, la cual se considera una semana antes de la evaluación Final de la asignatura.
- Las tareas ya están estimadas desde la semana 4.

#### III. Procedimientos

- Retomar el archivo Requerimientos Funcionales.xlsx el que fue descargado desde:  
<https://github.com/y Marquez/tallerProyectosI>

#### Instrucciones:

1. Completa la hoja Gantt, para ello el equipo debe asignar al responsable de cada actividad, considere una actividad culminada a aquella que se ha implementado y además probado.



Verifique que no se está generando sobrecarga de recursos. Pero también debe en la medida de lo posible definir actividades que se desarrollen en paralelo siempre en cuando no se asignen al mismo responsable.

- Cada participante del equipo debe tener asignado una abreviatura en base a su apellido, además de un color que permita reconocer fácilmente a quien se asignó la tarea.
- Es importante que el equipo acuerde los días de trabajo durante los cuales desarrollaran el proyecto, tratando que esta dedicación sea la misma para todos los integrantes del equipo de desarrollo.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Mes 1																				
2	Actividad	Requerimiento	Prioridad	Duración	Predecesor	17	20	21	22	23	24	27	28	29	30	1					
3		Diseño de base de datos	Alta	3 días	-																
4	<a href="#">2021-Hu-015</a>	15	Como administrador quiero	Alta	2 día	-															
5	<a href="#">2021-Hu-016</a>	16	Como administrador quiero	Alta	2 días	-															



## Semana 6 – Sesión 2

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 2	Fecha: ...../...../..... Duración: 60 min

### Preparación del entorno de desarrollo

I. **Propósito:** El equipo de desarrollo posea las mismas herramientas.

#### II. Descripción de la actividad a realizar (caso.)

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.

#### III. Procedimientos

##### Instrucciones:

1. Emplear un cuadro comparativo para mostrar por qué se ha elegido el entorno de desarrollo.

Criterio de evaluación	Nombre IDE		
Facilidad de uso			
Pertinencia con el proyecto/lenguaje			
Autocompletado			
Resaltado de sintaxis			
Depurador			
Capacidad de testing			
Total logrado			



2. El equipo debe crear un proyecto en Github y planificar las ramas y condiciones de los commits, una vez completado completar la siguiente tabla:

<b>URL github del proyecto</b>	
<b>Captura de pantalla (1 commit inicial por cada integrante)</b>	



## Semana 7 – Sesión 2

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 2	Fecha: ...../...../..... Duración: 60 min

### Gestión del Proyecto

#### I. Propósito: Controlar el buen desarrollo del proyecto

#### II. Descripción de la actividad a realizar (caso.)

En la gestión de proyectos durante la etapa de desarrollo o construcción se debe tener muy claro que características se deben tener en cuenta para valorar y validar el cumplimiento de las actividades planificadas.

Una forma de realizar lo indicado es mediante los criterios de aceptación, en el glosario del PMI ([www.pmi.org](http://www.pmi.org)) los criterios de aceptación (Acceptance Criteria) están definidos como: **“Conjunto de condiciones que debe cumplir antes de que se acepten los entregables”**. Una explicación muy escueta para un concepto de suma trascendencia en los proyectos, ya que es clave para completar la entrega, confirmando que el proyecto cumplió con su compromiso de entrega al cliente. Cuando el criterio de aceptación no es bueno, o no existe, o no se lo acordó a tiempo, la entrega suele complicarse.

User story	Acceptance Criteria
As a signed-in user,	<input type="checkbox"/> Display 'comments' icon below the post.
I want to be able to comment on a feed post	<input type="checkbox"/> Display list of comments from other users
So that I can get feedback from others	<input type="checkbox"/> Display keypad when the user clicks on the comments icon
	<input type="checkbox"/> User should be able to type a comment in the text box after clicking on comments icon
	<input type="checkbox"/> User can click on 'Post' button to save the comment
	<input type="checkbox"/> 'Post' button should be inactive when nothing is typed
	<input type="checkbox"/> 'Post' button becomes active after user starts typing
	<input type="checkbox"/> Display users comment in the list of comments
	<input type="checkbox"/> Display 'profile picture' and 'username' to the left of the comment
	<input type="checkbox"/> Display 'remove' and 'edit' icon opposite to the comment
	<input type="checkbox"/> Display 'back' button
	<input type="checkbox"/> 'Back' button takes the user back to the post

Ejemplo: (tomado del SWEBOKv3 @ <http://www4.ncsu.edu/~tjmenzie/cs510/pdf/SWEBOKv3.pdf>)

Contexto



Estamos construyendo una calculadora básica, la cual solo soporta las cuatro operaciones básicas: Adición, Sustracción, Multiplicación y División. Vamos a tomar cada una de las operaciones aritméticas básicas como un único requerimiento, así que tendremos:

- Req-1: La calculadora debe soportar la operación Adición.
- Req-2: La calculadora debe soportar la operación Sustracción.
- Req-3: La calculadora debe soportar la operación Multiplicación.
- Req-4: La calculadora debe soportar la operación División.

### Criterios de Aceptación

Conjunto de sentencias las cuales indican como sera juzgado (evaluado) un componente de software dado para ver si satisface o no un requerimiento. Cada elemento, criterio, es una sentencia especifica.

### Ejemplos

Nosotros, como expertos en matemáticas, sabemos que las operaciones aritméticas básicas tienen ciertas "propiedades". Podemos pensar en esas propiedades como reglas o sentencias las cuales definen en mayor detalle algún aspecto del requerimiento (operación aritmética). Así que ahora tomaremos la operación (requerimiento) Adición como el sujeto de nuestros ejemplos. Tendremos un Criterio por cada una de las propiedades de la operación Adición.

- Cri-1-1: La calculadora debe cumplir con la propiedad Conmutativa para la operación Adición.
- Cri-1-2: La calculadora debe cumplir con la propiedad Asociativa para la operación Adición.
- Cri-1-3: La calculadora debe cumplir con la propiedad Elemento Neutro para la operación Adición.
- Cri-1-4: La calculadora debe cumplir con la propiedad Distributiva, respecto a la multiplicación, para la operación Adición.

## III. Procedimientos

### Instrucciones:

1. Emplear la tabla de criterios de evaluación para validar la calidad de cada uno de las tareas. El equipo puede cambiar o agregar el criterio de evaluación de acuerdo a la naturaleza del proyecto.

Criterio de evaluación	Id requerimiento: Descripción: Criterios de aceptación: [Ejemplo] Registrar el 99.99% de ventas con valores normales		
	Test 1: valores normales	Test 2 – valores por defecto	Test 3- Valores grandes
Funcionalidad			



<b>Usabilidad</b>			
<b>Autocompletado</b>			
<b>Tiempo de respuesta</b>			
<b>Total logrado</b>			



## Semana 8 – Sesión 1

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 2	Fecha: ...../...../..... Duración: 60 min

### Gestión del Proyecto – Evaluación Parcial

I. **Propósito:** Controlar el buen desarrollo del proyecto

II. **Descripción de la actividad a realizar (caso.)**

En la gestión de proyectos no sólo se evalúa el avance individual de las tareas sino también el avance global del proyecto como un todo, por ello es importante definir Hitos, que se refieren a revisiones y validaciones importantes del avance logrado hasta la fecha

III. **Procedimientos**

**Instrucciones:**

Descarga desde <https://github.com/y Marquezz/tallerProyectosI> el archivo:

**presentacion\_\_avance\_\_parcial\_\_Proyecto.pptx**, que es la plantilla para que presentes el avance parcial de tu proyecto. Ten en cuenta que se evaluará considerando la Lista de cotejo siguiente:

Indicadores	Cumplió	
	SÍ	NO
1. La Formulación del problema es correcta y se basa en fuentes objetivas.		
2. Se han validado los requerimientos del sistema aplicando la metodología del Product Backlog		
3. Los requerimientos del sistema consideran restricciones realistas.		
4. La solución se orienta específicamente a reducir un problema social, económico, ambiental o político.		
5. La solución planteada ha considerado los posibles impactos económicos, sociales y ambientales, que genera la solución de Ingeniería.		
6. Está claramente identificado el sustento teórico central de su proyecto.		
7. La presentación empleada argumenta la solución planteada exitosamente empleando elementos gráficos adecuados		
8. El Plan de Gestión del proyecto considera criterios de responsabilidades, paralelismo y restricciones de tiempo.		
9. El Plan de Gestión del proyecto está actualizado de acuerdo al avance real del proyecto.		
10. El avance del proyecto está acorde a lo programado, o se han corregido los riesgos que impidan el avance correcto		
	SUB TOTAL	
	NOTA VIGESIMAL	



## Tercera unidad

### Semana 9- Sesión 2

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 3	Fecha: ...../...../..... Duración: 60 min

### Control de hitos y línea base aplicación al proyecto

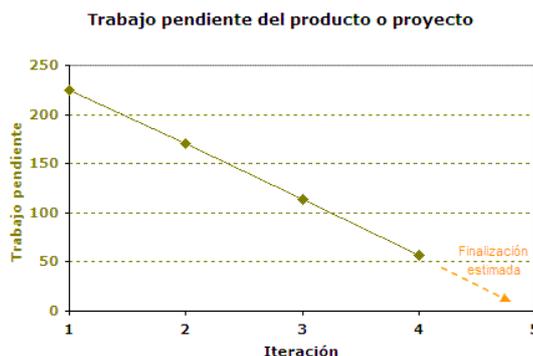
I. **Propósito:** Controlar el buen desarrollo del proyecto

II. **Descripción de la actividad a realizar (caso.)**

Un gráfico de trabajo pendiente es una representación gráfica del trabajo que queda por hacer frente al tiempo. A menudo se utiliza en metodologías de desarrollo de software ágiles como Scrum. Sin embargo, los gráficos de trabajo pendiente se pueden aplicar a cualquier proyecto que contenga un progreso medible a lo largo del tiempo.

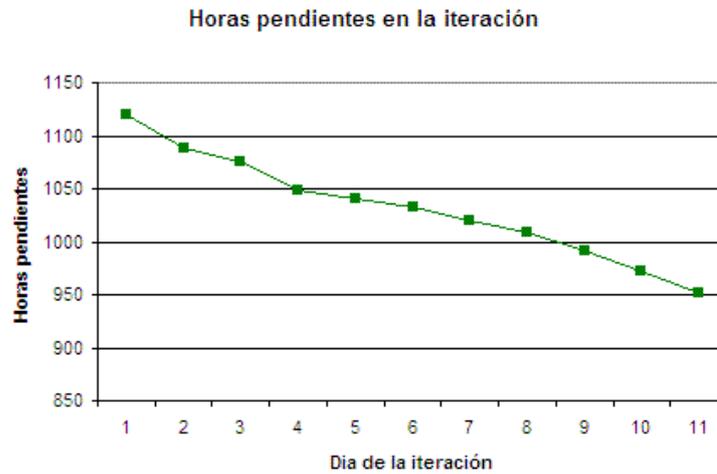
Por lo general, en un gráfico de trabajo pendiente, el trabajo sobresaliente suele estar en el eje vertical, con el tiempo a lo largo del horizontal. Es útil para predecir cuándo se completará todo el trabajo. En el Daily Scrum, el equipo de desarrollo actualiza el Sprint Burn Down y traza el trabajo restante del día. Un gráfico de evolución es casi una herramienta "imprescindible" para un equipo Scrum por las siguientes razones principales:

- monitorear el avance del alcance del proyecto
- Mantener al equipo funcionando según lo programado
- Comparar el trabajo planificado con la progresión del equipo





Fuente: (<https://proyectosagiles.org>, 2022)



Fuente: Fuente: (<https://proyectosagiles.org>, 2022)

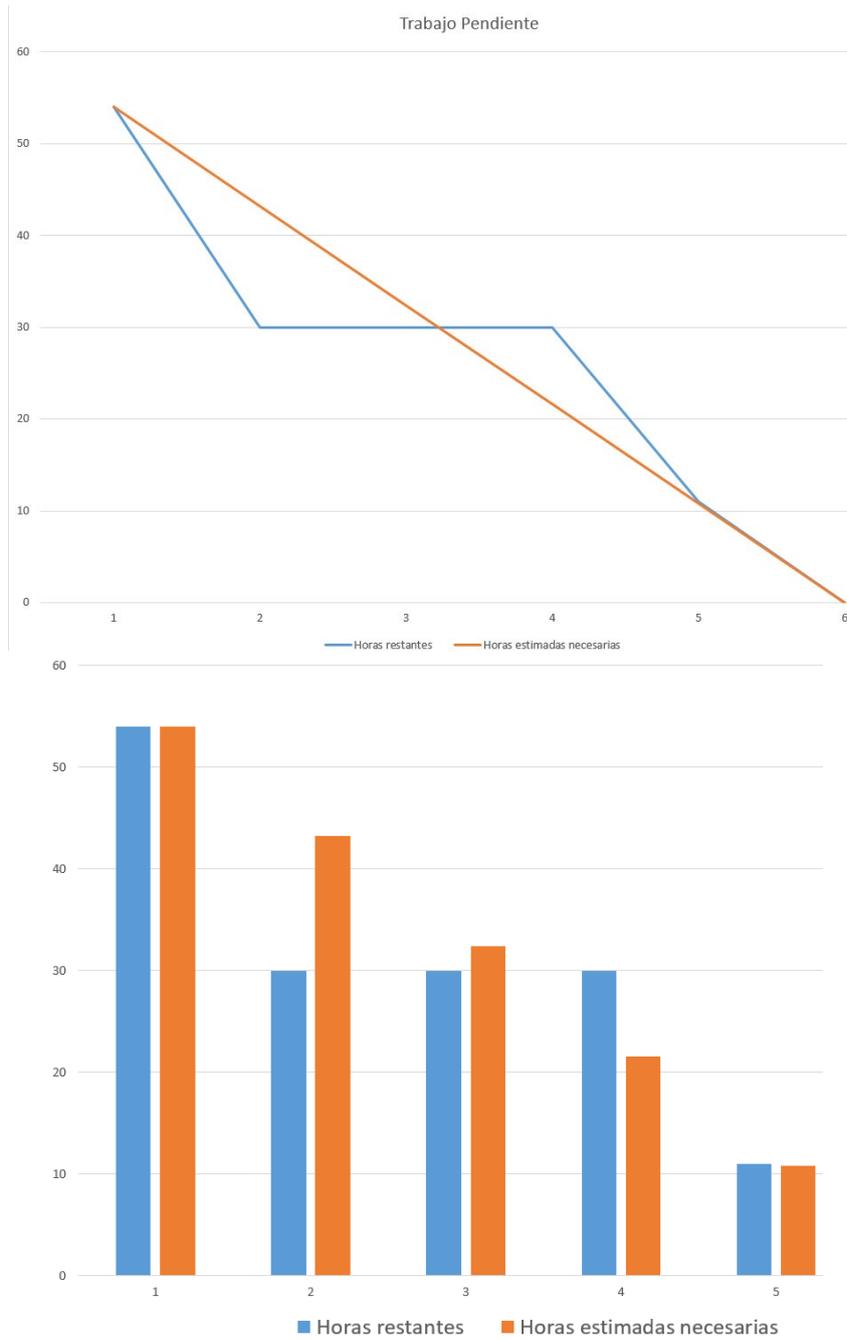
### III. Procedimientos

#### Instrucciones:

Descarga desde <https://github.com/y Marquezz/tallerProyectosI> el archivo: [burndownchart .xlsx](#), que es la plantilla para que puedas monitorizar el trabajo pendiente, para ello empleamos el concepto de BurnDownChart.:

En Hoja1 debes ingresar cada una de las tareas que se han realizado así como la duración que ha requerido cada una.

Tanto en gráfico 1 y gráfico 2, observarás los gráficos que mostrarán el trabajo que es necesario realizar para completar el proyecto de acuerdo al registro de las tareas de tu proyecto.





## Semana 10- Sesión 2

Sección: .....
Docente :
Unidad : Unidad 3

Apellidos :
Nombres :
Fecha: ...../...../..... Duración: 60 min

### Gestión de cambios

I. **Propósito:** Incorporar adecuadamente los cambios en el proyecto.

#### II. Descripción de la actividad a realizar (caso.)

Uno de los motivos para realizar un seguimiento de cambios es para frenar al Dueño del Producto de hacer agregados innecesarios al Backlog del Producto. Y esto resulta en una mayor eficiencia para el Dueño del Producto. A veces los elementos del backlog no están bien pensados, y por lo tanto tienden a cambiar muy seguido.

Recordemos que el backlog pertenece al Dueño del Producto, y puede cambiar tanto como lo crea necesario. De hecho, tomar medidas para impedir que surjan cambios en el backlog puede indicar problemas más serios en el proceso. Sin embargo, tampoco es bueno tener un Backlog de Producto que cambia constantemente.

El objetivo de cualquier Backlog inicial es ser muy acotado y de alto nivel, de manera que no debería sufrir cambios. Los pedidos de cambio en un backlog suelen indicar que el backlog era muy extenso o demasiado detallado; se dedicó demasiado tiempo a adivinar el futuro.

#### III. Procedimientos

##### Instrucciones:

Descarga desde <https://github.com/y Marquezz/tallerProyectosI> el archivo: Plantilla de solicitud de cambios.doc, que es la plantilla para que el equipo pueda generar las solicitudes de cambio.

El docente puede generar una solicitud de cambio sino hubiera una generada dentro del equipo de trabajo del proyecto.



**Solicitud de cambio**  
*[Nombre del Proyecto]*  
**Fecha:** [dd/mm/aaa]

**Datos de la solicitud de cambio**

Nro control de solicitud de cambio	
Solicitante del cambio	
Área del solicitante	
Lugar	

**Descripción de la propuesta de cambio**

**Justificación de la propuesta de cambio**

**Impacto del cambio en la línea base**

Alcance:

Cronograma:

Costo:

Calidad:

**Riesgos**



## Semana 11- Sesión 2

Sección: .....
Docente : .....
Unidad : Unidad 3

Apellidos : .....
Nombres : .....
Fecha: ...../...../..... Duración: 60 min

### Producto mínimo viable

- I. **Propósito:** Definir un primer Producto mínimo viable.
- II. **Descripción de la actividad a realizar (caso.)**

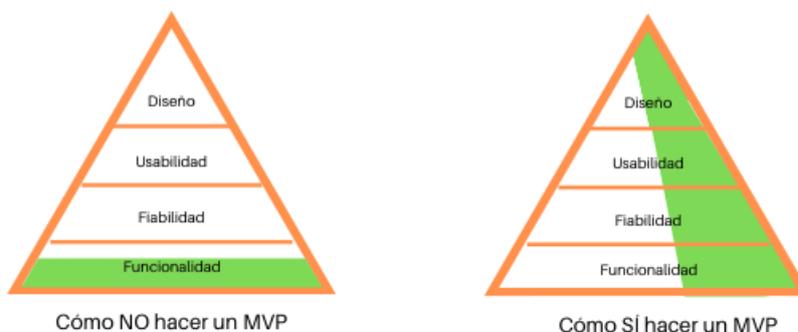
Fuente: (Benitez, 2022)

El lanzamiento de una idea al mercado representa un momento crítico que puede definir tu éxito o fracaso. Basado en el modelo lean startup, el design thinking y la metodología ágil, un producto mínimo viable es una propuesta inicial que sirve para probar la funcionalidad y recepción de un proyecto.

Una fuente adicional a revisar es : <https://youtu.be/F78byq2Ye50> , fuente (Ágil, 2022).

### Qué es un producto mínimo viable

El lanzamiento al mercado de un producto o servicio nuevo puede ser un riesgo calculado aplicando el método Lean Startup. Indicado no sólo para startups, sino para diferentes tipos de emprendimientos; su proceso te lleva a la creación de un producto mínimo viable (PMV), que es una primera versión funcional de tu idea y la cual te ayuda a entender su viabilidad.



### Para qué sirve hacer un MVP



Como ya te he comentado, el mvp sirve para desarrollar un producto mínimamente funcional que te proporciona valor inmediato sin disparar los costes. Empezar con este tipo de prototipo te permite aprender más sobre el usuario objetivo, el mercado en el que piensas establecerte y comprobar si tus hipótesis son ciertas.

Podemos resumir lo que puedes lograr con esta metodología ágil de la siguiente manera:

- Obtener mejor feedback de tus usuarios (clientes reales).
- Construir un prototipo, medir su viabilidad, aprender e iterar, en un reducido espacio de tiempo.
- Reducir los riesgos de validación mediante la retroalimentación constante y diferentes pruebas.
- Usar mejor tu presupuesto acelerando el desarrollo de un producto funcional e invirtiendo en técnicas de bajo coste.
- Validar tus hipótesis sobre el problema, la solución y el mercado con tu producto o servicio de forma eficiente.

### **III. Procedimientos**

#### **Instrucciones:**

#### **1 ) Lee detenidamente el siguiente caso.**

#### **Ejemplo: Como Grammarly llegó a la idea de desarrollar su extensión para Chrome**

La herramienta de corrección de textos en inglés Grammarly ha acumulado casi 20 millones de usuarios desde 2008. La clave de su éxito es que lograron convertir su producto de una herramienta que las personas usan ocasionalmente en un compañero cotidiano.

El producto original de Grammarly permitía corregir las faltas de ortografía al copiar y pegar texto en su editor de texto.

Vamos a desarrollar este punto un poco más con un ejemplo, para que veas todo el proceso:

- Contexto: el usuario está en su ordenador en el trabajo y necesita enviar un correo importante en inglés.
- Motivación: este usuario fue contratado recientemente y está ansioso por causar una buena impresión en su jefe. En primer lugar, está nervioso de enviarle un correo electrónico con errores ortográficos o tipográficos a un cliente extranjero importante, lo que le haga parecer descuidado o perezoso.
- Puntos de dolor : «Es fácil pasar por alto un error tonto cuando estás corrigiendo tu propia escritura».
- Modelos mentales: el usuario quiere sentirse seguro de que el borrador final de su correo electrónico está bien escrito y sin errores, a pesar de que se siente nervioso y propenso a errores. Este no quiere pedirle a nadie en su equipo que revise su correo electrónico a su jefe, porque se siente incómodo y pierde demasiado tiempo.



El mapa de viaje del usuario para este usuario podría ser el siguiente:

Línea de tiempo	Fase 1	Fase 2	Fase 3
<b>Punto de preocupación</b>	El usuario no quiere corregir su texto ni que otro de su equipo lo haga.	El usuario quiere asegurarse de que todos los errores son corregidos para su texto final.	El usuario necesita pasar el texto corregido a un email.
<b>Punto de contacto</b>	El usuario descubre que Grammarly puede hallar los errores del texto por él.	El usuario puede ver notas sobre ortografía contextual y errores de estilo a medida que Grammarly los identifica. Esto le asegura que sus faltas van más allá de una simple errata.	Cuando copia y pega el escrito no captura algunas palabras del final. Lo envía así sin percatarse y se siente frustrado por la omisión.
<b>Canal</b>	Aplicación web	Aplicación web	Aplicación web
<b>Experiencia</b>	Positiva.  El usuario encontró lo que necesitaba.	Positiva.  El proceso es bastante exhaustivo.	Negativa.  El proceso de copiar y pegar anuló en parte los beneficios de la corrección.

Este tipo de experiencia negativa podría disuadir a los usuarios de dejar de utilizar Grammarly, ya que agrega mucha fricción. Cada vez que el usuario quiere corregir algo, tiene que abrir Grammarly y copiar y pegar, dando lugar a un gran margen de error y pérdida de tiempo.

Grammarly solucionó esto lanzando una extensión web de Chrome. Activar la extensión permitiría a Grammarly corregir el texto en cualquier lugar en el que el usuario estuviera escribiendo dentro de este navegador. (Emails, formularios de contacto, google docs,...)

Ahora su producto puede ayudar a un usuario a escribir bien en inglés en cualquier situación. Así es como Grammarly cambió de un producto que la gente usaba ocasionalmente a uno que era constantemente necesario.

**2 ) Utilizando como plantilla el caso anterior, utiliza tu proyecto como caso y adecua para que podamos visualizar las 3 fases de necesidad del producto.**



## Semana 12- Sesión 2

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 3	Fecha: ...../...../..... Duración: 60 min

### Criterios de calidad para evaluar el Producto mínimo viable

- I. **Propósito:** Definir los indicadores que permitan evaluar el PMV.
- II. **Descripción de la actividad a realizar (caso.)**

Fuente: (Benitez, 2022)

El lanzamiento de una idea al mercado representa un momento crítico que puede definir tu éxito o fracaso. Basado en el modelo lean startup, el design thinking y la metodología ágil, un producto mínimo viable es una propuesta inicial que sirve para probar la funcionalidad y recepción de un proyecto.

Una fuente adicional a revisar es : <https://youtu.be/F78byq2Ye50> , fuente (Ágil, 2022).

### III. Procedimientos

Instrucciones:

Lee con atención el siguiente artículo:

#### ¿Qué necesita el PMV?

Como probar el MVP	
<b>Identificar a nuestro cliente</b>	Saber cuál es nuestro cliente potencial para definir su perfil buyer persona es el primer paso que debemos dar. Debemos conocer sus características geográficas, demográficas, qué le interesa...
<b>Investigar</b>	Debemos informarnos sobre las necesidades de las personas, salir a la calle, sondear las redes sociales y comprobar de primera mano los lugares físicos y virtuales por donde se mueven.
<b>Conocer a nuestra competencia</b>	Probar sus servicios y productos, tanto en el estudio de la competencia directa como de posibles productos que puedan sustituir al nuestro. Comprobar sus precios, canales y



	el valor que ofrecen, entre otra variedad de aspectos que dependen de cada caso.
<b>Crear nuestro PMV</b>	Elaborar nuestra primera versión del producto con los datos recogidos.
<b>Volver a nuestros clientes</b>	Con una propuesta firme, debemos ponernos en contacto con nuestros clientes de forma directa o incluso haciendo email marketing. Hay que averiguar si estarían interesados en pagar una determinada cantidad de dinero por nuestro producto mínimo viable.
<b>Invitar a probarlo</b>	De esta manera podremos contrastar nuestra idea con lo que piensa el cliente.

<b>Acción</b>	<b>Descripción</b>	<b>SI</b>	<b>NO</b>
Campañas de publicidad	Las campañas publicitarias son un método realmente efectivo para detectar el interés y hacer validación de mercado, sobre todo en Facebook y Google Ads donde podemos aprovecharnos de la segmentación que nos permiten hacer sobre nuestros clientes potenciales. Podremos comprobar cuáles son las características y funcionalidades más valoradas por el público.		
Landing page con el producto mínimo viable	Si nuestros clientes potenciales visitan nuestra web tendremos una gran oportunidad para nuestro negocio. En la página de aterrizaje podemos destacar las particularidades de nuestro producto logrando captar el interés de los visitantes. Es fundamental realizar una propuesta de valor eficaz, con una llamada a la acción efectiva y llamativa.		
Infografías o fotografías de producto	Elegir entre infografía o fotografía dependerá de las características de la iniciativa y de lo que se desea mostrar. Las infografías, gráficos y diseños son un estupendo recurso para mostrar imágenes cuando aún no se dispone de productos o servicios terminados. Por otro lado, las fotografías pueden ayudar a la audiencia a hacerse una idea clara de cómo será cualquier producto físico.		
Vídeo explicativo del producto mínimo viable	No hay nada que comunique mejor que un vídeo de producto en el que se pueda observar la experiencia de algún usuario con nuestro producto. Con estos vídeos podemos explicar brevemente qué hace nuestro producto y las razones por las que deberían comprarlo.		
Crowdfunding	Estas campañas basadas en modelos de negocio son una excelente oportunidad para efectuar pruebas de tu PMV ya que nos ofrecen la posibilidad de estudiar el impacto final que tienen los productos en el mercado según las contribuciones que hayan tenido las acciones. Además, puede que obtendremos la financiación necesaria para llevarlo a cabo a cambio de recompensas.		
Eventos y gamificación	Haciendo participar a la comunidad a través de sistemas de gamificación como juegos, sorteos o eventos, estaremos en contacto con un primer grupo realmente implicado en nuestro proyecto. También aquí es posible incluir incentivos para quienes nos apoyen, vídeos explicativos y más cosas.		



## Cuarta unidad

### Semana 13- Sesión 2

Sección: .....
Docente : .....
Unidad : Unidad 4

Apellidos : .....
Nombres : .....
Fecha: ...../...../..... Duración: 60 min

### Criterios de calidad de software

I. **Propósito:** Definir los criterios de calidad de software.

II. **Descripción de la actividad a realizar (caso.)**

Se requiere definir los criterios de calidad de software, para ello es de suma importancia tener muy presente los objetivos del proyecto y el alcance.

III. **Procedimientos**

**Instrucciones:**

1) Completa la tabla con respecto a los objetivos del proyecto:

Nro	Objetivo
1.	
2.	
3.	

2) Elabora un organizador de conocimiento comparando los estándares de calidad de McCall, Norma ISO/IEC 9126, e ISO 25000.



## Semana 14- Sesión 2

Sección: .....

Docente : .....

Unidad : Unidad 4

Apellidos : .....

Nombres : .....

Fecha: ...../...../..... Duración: 60 min

### Criterios de calidad de software

I. **Propósito:** Definir los criterios de calidad de software.

II. **Descripción de la actividad a realizar (caso.)**

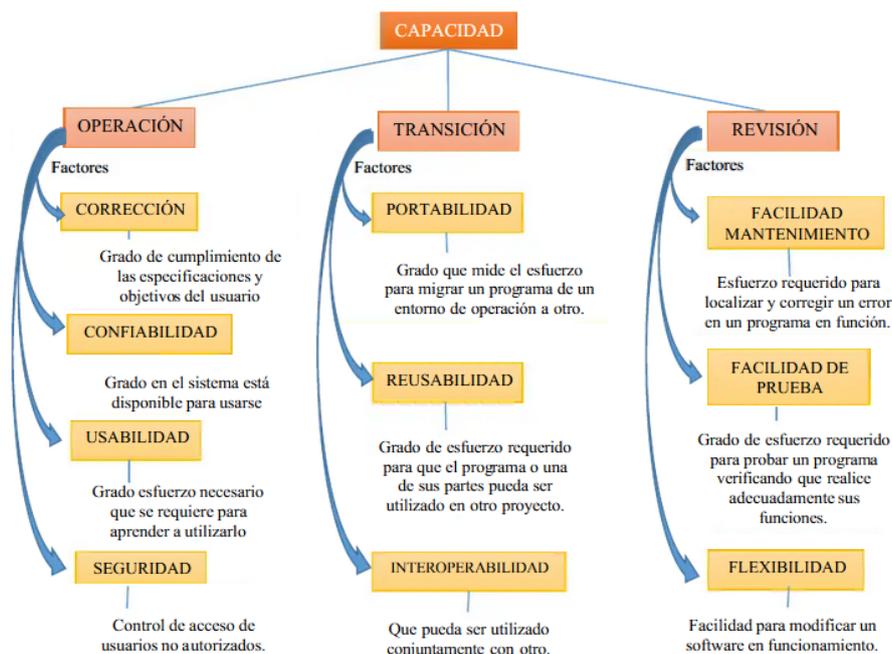
Se requiere definir los criterios de calidad de software que se medirán en la solución planteada, para ello puede basarse en distintos estándares, en este caso se propone emplear los del modelo de McCall.

III. **Procedimientos**

**Instrucciones:**

- 1) Efectúa la lectura e interpretación del siguiente organizador:

El modelo de Calidad Mc Call está planteado desde la perspectiva del usuario presentándose cuestionamientos en cada una de sus capacidades, en la siguiente figura se resumen las capacidades.





2) Emplea la siguiente tabla para determinar las capacidades y métricas con las cuales se evaluará la calidad de la solución de software que ha desarrollado. Elija entre 2 a 4 criterios.

CAPACIDAD	FACTOR	MÉTRICA	ELEGIDO
<b>Operación</b>	Corrección: Grado de cumplimiento de las especificaciones y objetivos del usuario  Confiabilidad: Grado en el sistema está disponible para usarse.	● Compleción	
		● Consistencia	
		● Trazabilidad	
		● Complejidad	
		● Consistencia	
		● Exactitud	
	Usabilidad: Grado de esfuerzo necesario que se requiere para aprender a utilizarlo.  Integridad o Seguridad: Grado en el que se controla el acceso al programa o los datos por usuarios no autorizados.  Eficiencia o Performance: Cantidad de recursos y código requeridos por un programa para realizar una función.	● Modularidad	
		● Simplicidad	
		● Tolerancia a errores	
		● Facilidad de formación	
		● Operatividad	
		● Facilidad de auditoria	
<b>Transición</b>	Portabilidad: Grado que mide el esfuerzo para migrar un programa de un entorno de operación a otro.  Reusabilidad: Grado de esfuerzo requerido para que el programa o una de sus partes pueda ser utilizado en otro proyecto.	● Instrumentación	
		● Seguridad	
		● Concisión	
		● Eficiencia de ejecución.	
		● Operatividad	
		● Auto documentación	
	Interoperabilidad: Grado de esfuerzo dedicado para que un sistema o programa pueda operar conjuntamente con otro.	● Generalidad	
		● Modularidad	
		● Auto documentación	
		● Generalidad	
		● Independencia hardware	
		● Independencia del sistema	
<b>Revisión</b>	Facilidad Mantenimiento: Esfuerzo requerido para localizar y corregir un error en un programa en funcionamiento.	● Modularidad	
		● Estd. Comunicaciones	
		● Estandarización de datos	
		● Auto documentación	
		● Concisión	
		● Consistencia	
		● Instrumentación	



	<p>Flexibilidad: Esfuerzo requerido para modificar un software en funcionamiento.</p>	<ul style="list-style-type: none"> <li>● Modularidad</li> <li>● Simplicidad</li> <li>● Auto documentación</li> <li>● Capacidad de expansión</li> <li>● Complejidad</li> <li>● Concisión</li> <li>● Consistencia</li> <li>● Generalidad</li> <li>● Modularidad</li> <li>● Simplicidad</li> </ul>	
	<p>Facilidad de Prueba: Grado de esfuerzo requerido para probar un programa verificando que realice adecuadamente sus funciones.</p>	<ul style="list-style-type: none"> <li>● Auto documentación</li> <li>● Complejidad</li> <li>● Facilidad de auditoria</li> <li>● Instrumentación</li> <li>● Modularidad</li> <li>● Simplicidad</li> </ul>	

3) Por cada uno de los criterios elegidos sustente su elección.

Criterio	Sustento
1.	
2.	
3.	
4.	



## Semana 15- Sesión 2

Sección: .....	Apellidos : .....
Docente :	Nombres : .....
Unidad : Unidad 4	Fecha: ...../...../..... Duración: 60 min

### Definición de las pruebas de calidad de software

I. **Propósito:** Definir las pruebas de calidad de software.

II. **Descripción de la actividad a realizar (caso.)**

Se tienen definidos los criterios de calidad de software, por lo que ahora toca especificar las pruebas a llevar a cabo.

III. **Procedimientos**

**Instrucciones:**

- 1) Por cada uno de los criterios elegidos especifique el indicador y el instrumento o método de medición.

Criterio	Indicador	Instrumento/Método de medición
1.		
2.		
3.		
4.		



## Lista de referencias

### Básica:

Sommerville, L. (2011). Ingeniería de software. (9.ª ed.). Pearson Educación.

### Complementaria:

Ken Schwaber & Jeff Sutherland (2020). La Guía Scrum.

Jeff Sutherland (2020) Scrum. Manual de Campo. Editorial Oceano

### Recursos digitales:

Ágil, I. d. (03 de 01 de 2022). <https://blog.hubspot.es/sales/producto-minimo-viable>. Obtenido de <https://youtu.be/F78byq2Ye50>

Benitez, P. R. (03 de 01 de 2022). <https://economyatic.com/>. Obtenido de <https://economyatic.com/producto-minimo-viable/>

Casanova, s. (03 de 01 de 2022). [samuelcasanova.com](http://samuelcasanova.com). Obtenido de [samuelcasanova](http://samuelcasanova.com)

ConsultoraPMO. (2022 de 01 de 3). Scrum Ejemplo Práctico Product Backlog. Obtenido de en <https://www.youtube.com/watch?v=573jB8DQPEU>

<https://proyectosagiles.org>. (03 de 01 de 2022). <https://proyectosagiles.org>. Obtenido de <https://proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts/>

Huarcaya Almeida Remigio. (25 de septiembre 2021). Ejemplo práctico de scrum y pmbok [video]. YouTube. [https://youtu.be/\\_BGGIBbTKg8](https://youtu.be/_BGGIBbTKg8)

Render2wrb (2020) Como Trabajar con Git, Github en Visual Studio Code. [software]. [www.Github.com](http://www.Github.com)

ScrumIO (2021, 25 de Setiembre ) Scrum: la guía extendida. <https://www.scrumio.com/scrum/>

Yo Androide. (2020). Android Studio: Github como clonar un repositorio, agregar cambios (commit) y subir cambios (push). [software]. [www.Github.com](http://www.Github.com)