**Ingeniería en Sistemas de Información**



**Universidad Tecnológica Nacional**

**Facultad Regional Córdoba**

**Ingeniería de Software**

Docentes: Meles, Silvia Judith (Adjunto)

Massano, María Cecilia (JTP)

Abdala, Valeria

Curso: 4k1

Grupo Nro: 2

Trabajo Práctico Nro: 5. Métricas

Integrantes:

Ciceri Leonardo Legajo: 52649

Fraschina Lucas Legajo: 57543

Liberal Rodrigo Legajo: 51658

Peker Julian Legajo: 51395

Perez Villar Eric Legajo: 51337

Sanhueza Emanuel Legajo:

Año del 2011

Índice

Tema Pag

## Enunciado del trabajo práctico 1

## Consigna 1

## Instrucciones 1

## Definición del Producto 1

## Métricas de Software 6

## Métricas Ágiles 15

## Conclusión 21

# Enunciado del trabajo práctico

## Consigna

Definir un conjunto de métricas de software a aplicar durante la ejecución de proyectos de desarrollo de software.  
Instrucciones

1. Dividirse en grupos
2. Realizar la identificación de métricas según cada enfoque.
3. Describir las métricas respetando el template.
4. Realizar un análisis comparativo de los dos enfoques y elaborar una conclusión.
5. Seleccionar un representante del grupo para presentar el trabajo realizado.

Es necesario desarrollar un producto de software a medida para la Administración de Campeonatos de Fútbol, a pedido de la Asociación Cordobesa de Fútbol, que todos los años organiza campeonatos en los que participan clubes de la región.

## Definición del Producto:

Objetivo:

Gestionar los campeonatos que organiza la LIGA, registrando los clubes participantes con todos sus jugadores y obteniendo información resultante de la realización de los campeonatos.

Requerimientos Funcionales (Alcances):

* Administración de campeonatos y sistemas de campeonatos.
* Administración de clubes
* Diagramación de Campeonatos y publicación de fixtures (campeonatos diagramados)
* Gestión de Inscripción de los clubes y sus jugadores
* Administración de la información de cada partido (goles, sanciones, desempeño de los jugadores).
* Administración de usuarios
* Emisión de estadísticas e informes vinculados a: partidos, jugadores, sanciones, goles.

Requerimientos No Funcionales:

| Nro. | Nombre | Descripción |
| --- | --- | --- |
| 1 | Navegador WEB | El producto debe ser compatible con Internet Explorer 7.0. o superior |
| 2 | Usuarios concurrentes | El producto debe dar soporte hasta 50 usuarios concurrentes sin afectar la performance de la aplicación. |
| 3 | Motor de Base de Datos | El motor de base de datos debe ser open source y no tener costo de licenciamiento. El motor de base de datos deberá soportar como mínimo la creación de storeprocedures, triggers y funciones, subconsultas y las consultas básicas (select, update, delete, create, drop) y de resumen (sum, avg, may, min). |
| 4 | Configuración de Monitores | Debe asumirse una configuración de monitores de 1024 x 768. |
| 5 | Sistemas de Ayuda | Debe tener un sistema de ayuda integrado y sensible al contexto. |
| 6 | Requerimientos de Hardware para usuarios | El producto no debe demandar instalaciones de ningún tipo en las máquinas de los usuarios. Debe ser suficiente tener acceso a internet y el navegador web. |
| 7 | Lenguaje de Programación | El lenguaje de programación debe permitir desarrollos de productos utilizables en Internet. |
| 8 | Interfaz para los reportes y estadísticas | Para todos los reportes y estadísticas se ha requerido que los mismos se muestren tanto en formato de cuadros como en formato gráfico. |
| 9 | Tiempo de Respuesta | El tiempo de respuesta para transacciones debe ser de hasta 5 segundos y para consultas debe ser de hasta 7 segundos en el 95 % de los casos. |
| 10 | Formatos de archivos para publicación de fixtures | El sistema deberá generar la información de los fixtures diagramados en diferentes formatos para Envío de Mails, Envío de mensajes a celulares, Envío a diarios y revistas para su publicación |

Consideraciones para el Proyecto

La Asociación Cordobesa de Fútbol realizó un llamado a licitación para desarrollar un producto que le ayude con la administración de los campeonatos y los clubes que participan en los mismos, que organiza a través de las Ligas Regionales de Fútbol. El pliego contiene las especificaciones técnicas generales del producto, que son las que se describieron en el apartado anterior y las consideraciones generales vinculadas al proyecto, las cuales se exponen a continuación:

El alcance del proyecto implica la especificación de los requerimientos, el desarrollo del producto, las pruebas, el despliegue del mismo y la capacitación a un representante de cada una de las Ligas Regionales. Es fundamental para el éxito del proyecto la aceptación del producto no sólo por parte de los referentes de la Liga Regional sino también por los usuarios de cada uno de los clubes, dado que se ha decido que el software que permita que cada club pueda gestionar un usuario y una clave de acceso y pueda registrar su propia inscripción y la inscripción de la lista de jugadores que participará, como consecuencia el producto de software no solo deberá desarrollarse con tecnología web, si no que es indispensable que sea un sistema fácil de usar y una ayuda integrada y sensible al contexto.

El plazo expuesto para la finalización del proyecto en la licitación es de 40 semanas y se requiere que además se contemple un período de “garantía” de 15 días hábiles. En el pliego están estipuladas multas que la empresa contratada deberá pagar por incumplimiento de plazos. Se ha definido como restricción que el sistema puede empezar a usarse para nuevos campeonatos, no para aquellos que se encuentren en curso, información que es importante tener en cuenta para definir las fechas de capacitación de usuarios referentes.

Las empresas que se presenten a la licitación debieron entregar como parte de la documentación la descripción del

proceso de desarrollo que utilizarán que incluya actividades de aseguramiento de calidad.

En el pliego se establece que las empresas que se presenten a la licitación no podrán subcontratar actividades del proyecto, lo que es un condicionante con respecto a la disponibilidad necesaria del personal de la empresa y de la evaluación de necesitar contratar y capacitar personal nuevo y los tiempos que esto implique.

# Métricas de Software

|  |
| --- |
| **Nombre de la Métrica**: Tiempo que lleva la ejecución de cada etapa de desarrollo del software. |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?)** : Calcular una estimación aproximada de duración de cada etapa de desarrollo del software. |
| **Preguntas que se responden con esta métrica:** ¿Cuántas semanas se estiman para el desarrollo de cada etapa del software? |
| **Nivel de Audiencia y de uso (Para quién?):** Líder del proyecto, para que el mismo pueda controlar los tiempos de cada una de las etapas de desarrollo. |
| **Base de Medición (Qué?):**  Semana: Semana de trabajo.  Etapa: etapa de desarrollo del software. |
| **Asunciones a cerca de los datos (Cómo?)** : Es la cantidad de semana que tiene cada etapa del desarrollo, exceptuando los feriados y días no laborables. |
| **Tendencia Esperada:**  La tendencia esperada del proyecto que se cumpla aproximadamente en 5 semanas por cada etapa del proceso de desarrollo. |
| **Recomendaciones:**  Tener en cuenta feriados, jornadas laborales de 6 horas, fines de semanas, actividades de cada etapa del proceso de desarrollo que se pueden ejecutar en paralelo. |
| **Dominio de la métrica:** Es del proceso. |
| **Frecuencia (Cada cuánto?)**  Debe emplearse en la etapa de planificación. |

|  |
| --- |
| **Nombre de la Métrica:**  Completitud de Funcionalidad Entregada |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Para evaluar la completitud del conjunto de funcionalidades entregadas al cliente |
| **Preguntas que se responden con esta métrica:**  ¿Qué tan completa está la implementación funcional? |
| **Nivel de Audiencia y de uso (Para quién?):**  Analista Funcional, ya que él se encarga de evaluar los requerimientos funcionales. |
| **Base de Medición (Qué?):**  F = 1 – X/Y  F = función que representa la métrica de la funcionalidad entregada del software. X = número de funciones faltantes Y = número de funciones descritas en la especificación de requerimientos. |
| **Asunciones a cerca de los datos (Cómo?):**  F oscila entre los valores de 0 y 1. Matemáticamente expresado a continuación.  0 <= F <= 1. Más cercano a 1, más completa es la implementación funcional del software. |
| **Tendencia Esperada:**  La tendencia esperada es que la implementación funcional del software se aproxime a 1. |
| **Recomendaciones:**  Cada función entregada tiene que estar al 100% completa, por ejemplo si una función está al 40%  Debe ser añadida como función faltante, solo cuando la función esta 100% completa no se añade como función faltante. |
| **Dominio de la métrica:** Es del producto. |
| **Frecuencia (Cada cuánto?)**  Debe emplearse cuando se entrega el producto de software desarrollado al cliente. |

|  |
| --- |
| **Nombre de la Métrica:**  Desvío en costo, diferencia entre el costo estimado y el costo real |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?)**  Para evaluar el porcentaje sobreestimado o subestimado del costo del proyecto. |
| **Preguntas que se responden con esta métrica:**  ¿Cuál es el porcentaje de variación entre el costo estimado y el costo real? |
| **Nivel de Audiencia y de uso (Para quién?):**  Lider del proyecto . |
| **Base de Medición (Qué?)** |
| **Asunciones a cerca de los datos (Cómo?):**  Si el valor de es negativo subestime el proyecto , si el valor es positivo sobrestime el proyecto. |
| **Tendencia Esperada:**  Cuando se aproxima a 0 indica que los costos estimado con los reales fueron razonablemente calculados |
| **Recomendaciones:**  No se deberían considerar los costos fijos por que implícitamente están considerados  Se debe incluir aquello variables como la cantidad de horas hombres, los costos requeridos para la aceptación del producto con los usuarios del sistema. |
| **Dominio de la métrica**: Es del proyecto. |
| **Frecuencia (Cada cuánto?):**  Al finalizar el proyecto de software. |

|  |
| --- |
| **Nombre de la Métrica:**  Media de Defectos detectados por el usuario |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?)**  Para mejorar la calidad del software que se entrega al usuario. |
| **Preguntas que se responden con esta métrica:**  ¿Cuántos defectos son detectados por los usuarios? |
| **Nivel de Audiencia y de uso (Para quién?):**  Desarrolladores, Analistas Funcionales. |
| **Base de Medición (Qué?)**  D=Total de defectos detectados por los usuarios. |
| **Asunciones acerca de los datos (Cómo?):**  Identificar cuantos defectos son encontrados en la funcionalidad del software durante la etapa de aceptación del mismo. |
| **Tendencia Esperada:**  La tendencia esperada es disminuir la cantidad de defectos, por cada prueba de aceptación realizada. |
| **Recomendaciones:**  Deben considerarse tanto los defectos encontrados dentro de los requerimientos funcionales y no funcionales. |
| **Dominio de la métrica:** Es del producto. |
| **Frecuencia (Cada cuánto?):**  En el momento de realizar la prueba de aceptación de los usuarios. |

|  |
| --- |
| **Nombre de la Métrica:**  Líneas de código empleadas en el software. |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Métrica empleada para determinar el tamaño del software en función de sus líneas de código. |
| **Preguntas que se responden con esta métrica:**  ¿Cuántas líneas de código se implementaron para desarrollar el software? |
| **Nivel de Audiencia y de uso (Para quién?):**  Destinado a los desarrolladores y líder del proyecto, para que puedan determinar el tamaño del software. |
| **Base de Medición (Qué?):**  KLOC es miles de líneas de código.  Aquí se realiza la sumatoria de cantidad de líneas de código que se emplean. |
| **Asunciones a cerca de los datos (Cómo?):**  Tener en cuenta que el resultado está expresado en miles de líneas de código. |
| **Tendencia Esperada :**  Se esperan 200 KLOC. |
| **Recomendaciones:**  Solo deberán considerarse las líneas de códigos eficientes. Los comentarios no serán considerados como líneas de código.  Las líneas de código serán sumadas de 1000 ya que la unidad está expresada es el KLOC. |
| **Dominio de la métrica:** Es del producto. |
| **Frecuencia (Cada cuánto?):**  Al fin de la etapa de implementación. |

|  |
| --- |
| **Nombre de la Métrica:**  Desviación del esfuerzo real respecto al esfuerzo estimado en cada etapa. |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Para poder cumplir con los tiempos de cada etapa del proceso de desarrollo. |
| **Preguntas que se responden con esta métrica:**  ¿En cuales etapas de desarrollo se producen desvíos? |
| **Nivel de Audiencia y de uso (Para quién?):**  Para el Líder Proyecto y el Responsable de Calidad para determinar el desvió de cada etapa del proyecto y tomar medidas correctivas. |
| **Base de Medición (Qué?):**  -  = Desvió total de la etapa i(cada etapa i es una etapa del proceso de desarrollo. |
| **Asunciones a cerca de los datos (Cómo?):**  Si la desviación es negativa, el proyecto presenta un atraso en la etapa que está siendo evaluada. |
| **Tendencia Esperada:**  Se espera que la desviación se aproxime a cero, es decir que no exista un atraso en la etapa. |
| **Recomendaciones:**  Solamente puede emplearse por cada etapa del proyecto, no se puede emplear por el proceso desarrollo entero. |
| **Dominio de la métrica:** Es del proceso. |
| **Frecuencia (Cada cuánto?):**  Durante todo el proceso de desarrollo del software. |

|  |
| --- |
| **Nombre de la Métrica:**  Defectos en un caso CU. |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Para evitar el retrabajo que es generado por defectos presentes un caso de uso. |
| **Preguntas que se responden con esta métrica:**  ¿En cuales casos de uso se producen la mayor cantidad de defectos? |
| **Nivel de Audiencia y de uso (Para quién?):**  Para el Analista Funcional y el Desarrollador para determinar la cantidad de defectos que se presentan en un caso de uso. |
| **Base de Medición (Qué?)** |
| **Asunciones a cerca de los datos (Cómo?):**  Es la sumatoria de defectos presentes en un caso de uso. |
| **Tendencia Esperada :**  Se espera que la cantidad de defectos sea 25 en un caso de uso. |
| **Recomendaciones:**  Solamente puede emplearse por cada etapa del proyecto, no se puede emplear por el proceso desarrollo entero. |
| **Dominio de la métrica:** Es del proyecto. |
| **Frecuencia (Cada cuánto?):**  Puede emplearse cuando se realiza las pruebas en cada caso de uso. |

|  |
| --- |
| **Nombre de la Métrica:**  Numero de Requerimientos no funcionales no entregados. |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Para conocer la cantidad de requerimientos que no se cumplen en el plazo estimulado |
| **Preguntas que se responden con esta métrica:**  ¿Cuánto Requerimientos no funcionales que no se cumplieron dentro el plazo de entrega? |
| **Nivel de Audiencia y de uso (Para quién?):**  Para el Analista Funcional y QA para que tome medidas correctivas. |
| **Base de Medición (Qué?):** |
| **Asunciones a cerca de los datos (Cómo?):**  Representa a la cantidad de requerimientos no funcionales no entregados al finalizar la entrega del proyecto |
| **Tendencia Esperada :**  Se espera se aproxime a cero, la cantidad de requerimientos no funcionales no entregados. |
| **Recomendaciones:**  La métrica es necesaria reformularla si se detecta un nuevo requerimiento no funcional |
| **Dominio de la métrica:** Es del producto. |
| **Frecuencia (Cada cuánto?):**  Al finalizar el proyecto de software. |

|  |
| --- |
| **Nombre de la Métrica:**  Desvío de calendario en semanas. |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Para determinar las si se subestimó o se sobreestimó el tiempo de desarrollo del proyecto. |
| **Preguntas que se responden con esta métrica:**  ¿Cuál es la desviación que se presentó en el proyecto? |
| **Nivel de Audiencia y de uso (Para quién?):**  Para el Líder Proyecto para monitorear y controlar el desvió que se presenta en el proyecto. |
| **Base de Medición (Qué?):**  Diferencia entre la sumatoria de semanas estimadas y la sumatoria de semanas total reales. |
| **Asunciones a cerca de los datos (Cómo?):**  Si el valor de la desviación es negativo se subestimó el proyecto.  Si el valor de la desviación es positivo se sobreestimó el proyecto. |
| **Tendencia Esperada:**  En general, la tendencia de está métrica es de valor negativo, ya que siempre existen semanas de retraso. |
| **Recomendaciones:**  No deben medirse otra unidad que no sean semanas.  Las semanas son de 5 días. |
| **Dominio de la métrica:** Es del proceso. |
| **Frecuencia (Cada cuánto?):**  Durante el proceso de desarrollo del software. |

|  |
| --- |
| **Nombre de la Métrica:**  Cantidad de Trabajadores Disponible por Etapa. |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Para planificación del proyecto, asignando la cantidad de trabajadores necesarios en cada etapa para evitar subcontratar actividades. |
| **Preguntas que se responden con esta métrica:**  ¿Cuántos Trabajadores disponibles hay en cada etapa de desarrollo? |
| **Nivel de Audiencia y de uso (Para quién?):**  Líder de Proyecto la utiliza en la planificación, para conocer a los trabajadores disponible que posee en cada etapa del desarrollo del software. |
| **Base de Medición (Qué?)**  CantidadTrabajdoresDisponiblexEtapai= Cantidad de trabajadores disponibles que hay en la etapa i(las etapas de desarrollo del software) |
| **Asunciones a cerca de los datos (Cómo?):**  Se Representa la cantidad total de trabajadores disponibles de cada etapa de desarrollo de software |
| **Tendencia Esperada:**  Se espera tener los trabajadores necesarios para cada etapa de desarrollo. |
| **Recomendaciones:**  Esta métrica solamente puede ser utilizado en la etapa de planificación, si se presentan desviaciones no puede ser utilizada la métrica. |
| **Dominio de la métrica:** Es del proyecto. |
| **Frecuencia (Cada cuánto?):**  En la etapa de planificación del proyecto. |

# Métricas Ágiles

|  |
| --- |
| **Nombre de la Métrica:**  Pruebas de aceptación de Sprint |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  No solo le permiten al Líder de Proyecto asegurarse de compilar correctamente lo que el cliente necesita a lo largo del ciclo de vida del proyecto, sino también de consolidar la confianza del cliente y de mostrar las responsabilidades que ha aceptado |
| **Preguntas que se responden con esta métrica:**  ¿Cuántos Sprint son aceptados durante el ciclo de vida del proyecto? |
| **Nivel de Audiencia y de uso (Para quién?):**  El equipo debe ejecutarlos con tanta frecuencia como sea posible a fin de proporcionar una línea base en el conjunto de pruebas que es preciso superar en un sprint. El propietario del producto puede ejecutar también las pruebas de aceptación y exigir la comprobación para completar correctamente el sprint. |
| **Base de Medición (Qué?)**  Donde:   * CPU: valor de la métrica para el control de pruebas de unidad. * NCP: número de componentes probados individualmente antes de integrarlos. * CImp: número de componentes implementados. |
| **Asunciones a cerca de los datos (Cómo?):**  Ofrece una medida de los componentes que se han probado individualmente antes de la integración con respecto al total de componentes que fueron implementados. |
| **Tendencia Esperada:**  Esta métrica ofrece valores entre 0 y 1. Si CPU = 1 significa que todos los componentes implementados fueron probados individualmente antes de la integración. |
| **Recomendaciones:**  No debe utilizarse esta métrica para presionar al equipo de trabajo en cuanto a los componentes implementados en el transcurso del proyecto |
| **Dominio de la métrica:** Es del proyecto |
| **Frecuencia (Cada cuánto?):**  Conforme se entrega cada función |

|  |
| --- |
| **Nombre de la Métrica:**  Funcionalidades a terminar por mes |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Mide las funcionalidades que quedan por terminar en periodos mensuales |
| **Preguntas que se responden con esta métrica:**  ¿Cuánto trabajo queda por terminar? |
| **Nivel de Audiencia y de uso (Para quién?):**  Miembro del equipo, dueño del producto, scrum master y el administrador de proyecto |
| **Base de Medición (Qué?)**  Donde:   * CFF = Cantidad de funcionalidad por terminar en el mes * CFT = Cantidad de funcionalidad asignado a ese mes * CFH = Cantinda de funcionalidad hecha |
| **Asunciones a cerca de los datos (Cómo?):**  Cantidad de funcionalidades que el equipo puede realizar |
| **Tendencia Esperada:**  Se espera que esta métrica ronde valores que se aproximen a 1 de manera que indique poca funcionalidad por terminar |
| **Recomendaciones:**  Al saber cuántas funcionalidades faltan por terminar no se deben utilizar esta métrica para ejercer presión sobre el rendimiento del equipo |
| **Dominio de la métrica:** Es del proyecto. |
| **Frecuencia (Cada cuánto?):**  En intervalos de frecuencia fija. Mensualmente |

|  |
| --- |
| **Nombre de la Métrica:**  Esfuerzo real por Sprint |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Útil para proyectar la fecha probable de un alcance dado y útil para estimar la cantidad de alcance que puede ser entregado para una fecha dada. |
| **Preguntas que se responden con esta métrica:**  ¿Cuánto horas se trabajaron para finalizar el Sprint? |
| **Nivel de Audiencia y de uso (Para quién?):**  El Scrum Master lo utiliza para cuantificar la entera capacidad del equipo. |
| **Base de Medición (Qué?)**  Donde:   * ERS = Esfuerzo Real por Sprint * WH = Horas de Trabajo Disponible por Día * DA = Días Disponibles Iteración |
| **Asunciones a cerca de los datos (Cómo?):**  Se representa el esfuerzo de todo el equipo para cada iteración multiplicando las horas de trabajo disponible por día por los días disponibles de iteración. |
| **Tendencia Esperada:**  Se representa la capacidad de esfuerzo de todo el equipo de trabajo en cada iteración. |
| **Recomendaciones:**  Tener en cuenta feriados, jornadas laborales de 6 horas, fines de semanas, actividades de cada etapa del proceso de desarrollo que se pueden ejecutar en paralelo. |
| **Dominio de la métrica:** Es del proceso. |
| **Frecuencia (Cada cuánto?):**  Al final de cada Iteración |

|  |
| --- |
| **Nombre de la Métrica:**   Tasa de propagación de defectos |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?):**  Indicar el comportamiento de la propagación de defectos en la realización de las pruebas |
| **Preguntas que se responden con esta métrica:**  ¿Cuál es la tasa con la cual se propagan los defectos en ciclo de vida del Proyecto? |
| **Nivel de Audiencia y de uso (Para quién?):**  Líder de Proyecto la utiliza para analizar con que envergadura se van a propagar los defectos luego de realizado los correspondientes casos de prueba. |
| **Base de Medición (Qué?)**  Donde:   * TDP = tasa de propagación de defectos * Dn: número de defectos ocasionados al corregir defectos anteriores. * Dc: número de defectos corregidos. |
| **Asunciones a cerca de los datos (Cómo?):**  Se representa el número de defectos ocasionados al corregir un defecto dividido por el número total de defectos corregidos. |
| **Tendencia Esperada:**  El valor de TPD ofrece mejores resultados cuando está más cerca de 1. |
| **Recomendaciones:**  Cuando el número de defectos ocasionados al corregir defectos es mayor que el número de defectos corregidos, en este caso dichos valores no se tendrán en cuenta al trabajar con el resultado de la métrica, sino que se considerarán como cero |
| **Dominio de la métrica:** Es del proyecto. |
| **Frecuencia (Cada cuánto?):**  Conforme se entrega cada función |

|  |
| --- |
| **Nombre de la Métrica**:  Velocidad con que se aporta valor al negocio |
| **Razón para usarla (cómo ayuda a crear valor) (Por qué?)** :  Permite medir le aprendizaje del equipo, ya que es una métrica que debería aumentar con el tiempo. Suma de los "story points" realmente completados en el sprint. |
| **Preguntas que se responden con esta métrica:**  ¿Cantidad de story points terminados en tiempo y forma en el Sprint por el equipo? |
| **Nivel de Audiencia y de uso (Para quién?):**  Miembros del equipo, el Scrum Master, el dueño del producto y el administrador del proyecto. |
| **Base de Medición (Qué?):**  Donde:   * SPC = Story Points Compleatos * SPE = Story Points Esperados * SPT = Story Points Terminados en tiempo y forma |
| **Asunciones a cerca de los datos (Cómo?)** :  Se representa la velocidad del equipo mediante los Story Points Terminados de todo el equipo para cada iteración dividiendo los Story Points Compleatos por los Story Points Compleatos. |
| **Tendencia Esperada:**  El valor de SPT (Story Points Terminados en tiempo y forma) cuando está más cerca de 1 indica que quedaron pocos Story Points por completar |
| **Recomendaciones:**  No se debe usar la métrica para castigar al equipo si no se consigue buenos resultados. Se deben tener en cuenta los días en los cuales no se presentó todo el equipo a trabajar. |
| **Dominio de la métrica:** Es del proceso. |
| **Frecuencia (Cada cuánto?)**  Debe emplearse al termino de cada sprint |

# Conclusión

En este trabajo práctico se definió una guía de posibles métricas a aplicar durante el ciclo de vida del proyecto utilizando dos enfoques, por un lado las Métricas de Software y por el otro las Métricas Ágiles.

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento.

Hay varias razones que justifican el uso de las métricas en el proceso de desarrollo de software. Por un lado se dice que cuando se puede medir aquello de lo cual se está hablando y se puede expresar en números, se sabe realmente acerca de ello; pero cuando no puede medirse, y no puede expresarse en números, el conocimiento que se tiene de ello es escaso e insatisfactorio.

En general, la medición persigue tres objetivos fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en nuestros proyectos y poder mejorar nuestros procesos y nuestros productos.

El contar con datos estadísticos de métricas de software, da un panorama de situaciones reales que ayudan aplicar y dar seguimiento a las diferentes formas de evaluar y determinar métricas de calidad para un mejor desempeño en la calidad de software.