



**UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL DE CÓRDOBA**

INGENIERÍA DE SOFTWARE
TRABAJO PRÁCTICO 1: DINÁMICA DE MANIFIESTO ÁGIL

CURSO: 4K2

PROFESORES:

- Ing. Cecilia Massano
- Ing. Gerardo Boiero

GRUPO: 3

INTEGRANTES:

- | | |
|-------------------------------|-------|
| ● Calderon, Jonathan | 71984 |
| ● Gonzalez, Marcos | 71997 |
| ● Palacios, Javier Guerino | 55624 |
| ● Salaberri, Marcelo | 57659 |
| ● Vela Rodriguez, Mariel Azul | 75865 |

Fecha: 10/08/21

ENUNCIADO

Unidad:	Unidad Nro. 2: Gestión Ágil de Productos de Software
Consigna:	Asocie cada uno de los valores del manifiesto ágil con el o los principios ágiles que considere que están relacionados.
Objetivo:	Comprender valores y principios del Manifiesto Ágil, para aplicarlos a ejemplos y prácticas concretas y cotidianas de trabajo.
Propósito:	Familiarizarse con los valores y principios del Manifiesto Ágil.
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Manifiesto Ágil. Bibliografía referenciada sobre el tema.
Salida:	Comparación de los valores del manifiesto ágil con los principios del manifiesto ágil, y justificación asociada.
Instrucciones:	<ol style="list-style-type: none"> 1. Cada grupo discute individualmente qué valor del manifiesto ágil está asociado con qué principio ágil. 2. Cada grupo presenta al resto del curso sus conclusiones, justificando las relaciones presentadas.

Manifiesto Ágil

VALORES

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

PRINCIPIOS

1. La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes
2. Recibir cambios de requerimientos, aun en etapas finales
3. Releases frecuentes (2 semanas a un mes)
4. Técnicos y no técnicos trabajando juntos TODO el proyecto
5. Hacer proyectos con individuos motivados
6. El medio de comunicación por excelencia es cara a cara
7. La mejor métrica de progreso es la cantidad de software funcionando
8. El ritmo de desarrollo es sostenible en el tiempo
9. Atención continua a la excelencia técnica
10. Simplicidad - Maximización del trabajo no hecho
11. Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados
12. A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar

DESARROLLO

Valor	Principio	Justificación
Individuos e interacciones por sobre procesos y herramientas	<p>4. Técnicos y no técnicos trabajando juntos TODO el proyecto.</p> <p>5. Hacer proyectos con individuos motivados.</p> <p>6. El medio de comunicación por excelencia es el cara a cara.</p> <p>9. Atención continua a la excelencia técnica.</p> <p>11. Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados.</p> <p>12. A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar.</p>	Si los equipos se auto organizan y se mantienen motivados traen consigo, como ventaja, la optimización de tiempos y una comunicación fluida. Esto permite que los requerimientos solicitados por el cliente le aporten valor real ya que ellos son los que saben que se necesita para avanzar en el proyecto y si el cliente quiere realizar cambios sabe que tanto impacta en el mismo.
Software funcionando sobre documentación extensiva	<p>1. La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes</p> <p>3. Releases frecuentes</p> <p>7. Software funcionando es la medida principal de progreso</p> <p>8. El ritmo de desarrollo es sostenible en el tiempo</p> <p>10. Simplicidad: Maximización del trabajo no hecho</p>	Validar rápidamente lo requerido sin exceso de documentación permite entregar un software funcional, una medida de logro, lo que posibilita comprobar la realización del trabajo y sobre todo, la retroalimentación sobre algo real.
Colaboración por sobre negociación con el cliente	<p>3. Releases frecuentes</p> <p>1. La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes</p>	Permite tener feedback y realizar modificaciones con menor impacto en el desarrollo del software, logrando tener un producto con los requisitos que el

	8. El ritmo de desarrollo es sostenible en el tiempo	cliente realmente necesita agregando valor con cada entrega.
Responder a cambios por sobre seguir un plan	2. Recibir cambios de requerimientos, aun en etapas finales. 8. El ritmo de desarrollo es sostenible en el tiempo. 3. Releases frecuentes.	Se prefiere lograr el software que el cliente necesita aunque sea necesario realizar cambios de último momento, para darle ventaja competitiva. Se debe poder mantener las entregas continuas considerando los cambios que el cliente pida, a través de plazos cortos de entrega del software.