

## M7 - Taller 2

### Matías Espinoza

#### Contexto:

En este taller, el enfoque será la creación y priorización de Historias de Usuario utilizando los modelos INVEST y SMART. Los estudiantes también realizarán estimaciones ágiles mediante puntos de historia y aplicarán técnicas de priorización como Moscow para asegurar que las historias estén bien estructuradas y refinadas para un sprint.

#### 1. Escritura de Historias de Usuario (2 puntos)

- Redacta al menos 3 Historias de Usuario relacionadas con el desarrollo de la web del hospital.

- Aplica correctamente los modelos INVEST y SMART para asegurar que las historias estén claras y enfocadas en la entrega de valor.

- Cada historia debe incluir la descripción del usuario, la acción a realizar, y el valor esperado.

Ejemplo: "Como administrador, quiero gestionar el listado de pacientes, para tener acceso rápido a los historiales médicos."

A continuación, se presentan 3 Historias de Usuario relacionadas con el desarrollo de la web del hospital, aplicando los modelos INVEST y SMART:

- **Como paciente, quiero poder ver la lista de médicos disponibles por especialidad, para poder seleccionar el médico adecuado para mi necesidad.**
  - **I (Independiente):** Esta historia puede desarrollarse sin depender de otras historias.
  - **N (Negociable):** Los detalles de la lista y las especialidades pueden discutirse.
  - **V (Valiosa):** Permite a los pacientes encontrar y elegir médicos, añadiendo valor a la web.
  - **E (Estimable):** El esfuerzo para mostrar una lista de médicos puede ser estimado.
  - **S (Pequeña):** Es una tarea que se puede completar en un sprint.
  - **T (Testable):** Se puede verificar si la lista de médicos se muestra correctamente.
  - **S (Específica):** El usuario (paciente) quiere ver una lista de médicos por especialidad para seleccionar uno.
  - **M (Medible):** Se puede medir si la lista se muestra y si se puede filtrar por especialidad.
  - **A (Alcanzable):** Es factible desarrollar esta funcionalidad en un sprint.
  - **R (Relevante):** Es importante para los pacientes poder encontrar médicos.
  - **T (Temporal):** Se puede completar dentro de un límite de tiempo (sprint).
- **Como médico, quiero poder ver mi horario de citas, para poder gestionar mi agenda de manera eficiente.**
  - **I (Independiente):** Ver el horario puede desarrollarse independientemente de otras funcionalidades.
  - **N (Negociable):** El formato y los detalles del horario pueden discutirse.
  - **V (Valiosa):** Ayuda a los médicos a organizar su trabajo.

- **E (Estimable):** Se puede estimar el esfuerzo de mostrar el horario.
- **S (Pequeña):** Es una funcionalidad que se puede completar en un sprint.
- **T (Testeable):** Se puede verificar si el horario del médico se muestra correctamente.
- **S (Específica):** El usuario (médico) quiere ver su horario de citas para gestionar su agenda.
- **M (Medible):** Se puede verificar si el horario se muestra con las citas correspondientes.
- **A (Alcanzable):** Desarrollar la visualización del horario es factible.
- **R (Relevante):** Es crucial para los médicos gestionar sus citas.
- **T (Temporal):** Se puede completar dentro de un sprint.
- **Como administrador, quiero gestionar el listado de pacientes, para tener acceso rápido a los historiales médicos.**
  - **I (Independiente):** La gestión del listado de pacientes puede desarrollarse de forma aislada.
  - **N (Negociable):** Las funcionalidades de gestión (crear, editar, eliminar) pueden discutirse.
  - **V (Valiosa):** Permite al administrador mantener la información de los pacientes.
  - **E (Estimable):** El esfuerzo de implementar la gestión del listado puede ser estimado.
  - **S (Pequeña):** Se puede dividir en tareas más pequeñas para un sprint.
  - **T (Testeable):** Se puede verificar si las operaciones de gestión funcionan correctamente.
  - **S (Específica):** El usuario (administrador) quiere gestionar el listado de pacientes para acceder a historiales.
  - **M (Medible):** Se puede verificar si se pueden realizar las operaciones de gestión.
  - **A (Alcanzable):** Es factible desarrollar la gestión de pacientes.
  - **R (Relevante):** Es esencial para la administración del hospital.
  - **T (Temporal):** Se puede completar en un periodo determinado.

## **2. Estimación Ágil de Historias de Usuario (1.5 puntos)**

- Realiza la estimación de las historias de usuario utilizando Puntos de Historia o modelos de tallas (XS, S, M, L, XL).

- Justifica la lógica detrás de cada estimación, explicando cómo la complejidad técnica y el esfuerzo influyen en la asignación de puntos o tallas.

A continuación, se realiza la estimación de las historias de usuario utilizando Puntos de Historia:

- **Como paciente, quiero poder ver la lista de médicos disponibles por especialidad, para poder seleccionar el médico adecuado para mi necesidad.**
  - **Estimación: 3 puntos de historia.**
  - **Justificación:** Se considera una complejidad media, ya que implica consultar una base de datos de médicos y filtrarla por especialidad, además de diseñar la interfaz para mostrar la información.

- **Como médico, quiero poder ver mi horario de citas, para poder gestionar mi agenda de manera eficiente.**
  - **Estimación: 5 puntos de historia.**
  - **Justificación:** Se estima una complejidad mayor debido a la necesidad de acceder a la agenda específica del médico, considerando diferentes tipos de citas y posiblemente integraciones con otros sistemas de gestión de citas.
- **Como administrador, quiero gestionar el listado de pacientes, para tener acceso rápido a los historiales médicos.**
  - **Estimación: 8 puntos de historia.**
  - **Justificación:** Se considera la historia más compleja, ya que implica la creación, lectura, actualización y eliminación (CRUD) de registros de pacientes, validación de datos y consideraciones de seguridad y acceso a historiales médicos (que podrían ser una funcionalidad separada pero relacionada).

### 3. Priorización de Historias de Usuario (1.5 puntos)

- *Utiliza el modelo de priorización Moscow para clasificar las historias en Must Have, Should Have, Could Have, o Won't Have.*
- *Explica las razones detrás de la priorización, considerando las necesidades del negocio y la visión del producto.*

A continuación, se aplica el modelo de priorización MoSCoW a las historias de usuario:

- **Must Have (Debe Tener):**
  - **Como paciente, quiero poder ver la lista de médicos disponibles por especialidad, para poder seleccionar el médico adecuado para mi necesidad.**
    - **Razón:** Esta funcionalidad es **esencial** para que los pacientes puedan utilizar la web para encontrar atención médica. Sin esta capacidad, el sitio web carece de una de sus funciones principales.
- **Should Have (Debería Tener):**
  - **Como administrador, quiero gestionar el listado de pacientes, para tener acceso rápido a los historiales médicos.**
    - **Razón:** Esta funcionalidad es **importante** para la gestión del sistema y el acceso a la información de los pacientes. Aunque los pacientes podrían interactuar con otras partes del sitio, la administración del listado es crucial para el funcionamiento a largo plazo.
- **Could Have (Podría Tener):**
  - **Como médico, quiero poder ver mi horario de citas, para poder gestionar mi agenda de manera eficiente.**
    - **Razón:** Si bien es **deseable** que los médicos puedan ver su horario en la web, inicialmente se podrían utilizar otros métodos para la gestión de citas. Esta funcionalidad podría implementarse en una fase posterior.
- **Won't Have (No Tendrá):**
  - En este contexto inicial, no se identifica ninguna historia que explícitamente no se vaya a incluir. Sin embargo, funcionalidades más complejas como la

reserva de citas online por parte de los pacientes podrían entrar en esta categoría para el primer sprint.

#### **4. Refinamiento del Backlog (2 puntos)**

- Lleva a cabo una sesión de Refinamiento del Backlog para asegurar que las historias de usuario estén listas para el sprint.
- Ajusta las historias según sea necesario para reducir la ambigüedad y mejorar su claridad y completitud.

*Herramientas a Utilizar:*

- Plantillas para Historias de Usuario basadas en los modelos INVEST y SMART.
- Herramientas de estimación como puntos de historia o modelos de tallas.
- Modelo de priorización Moscow para organizar las historias por urgencia e importancia.

Una sesión de **Refinamiento del Backlog** tiene como objetivo asegurar que las Historias de Usuario estén claras, sean entendibles y listas para ser trabajadas en futuros sprints. Durante esta sesión, el **Equipo de Desarrollo**, el **Product Owner** y, a menudo, el **Scrum Master** se reúnen para revisar el **Product Backlog**.

Los pasos clave en una sesión de Refinamiento del Backlog incluirían:

- **Revisar las historias de usuario:** El Product Owner presenta las historias priorizadas del backlog. El equipo discute cada historia para asegurar una comprensión común del **objetivo** y el **beneficio** que aporta. Por ejemplo, para la historia de "Como paciente, quiero poder ver la lista de médicos disponibles por especialidad...", se discutirían detalles como qué información del médico se mostrará (nombre, especialidad, información de contacto), cómo se agruparán las especialidades, y si habrá opciones de búsqueda o filtrado adicionales.
- **Dividir historias grandes:** Si alguna historia se considera demasiado grande o compleja para ser completada en un sprint, se **descompone en tareas más pequeñas y manejables**. Por ejemplo, la historia de "Como administrador, quiero gestionar el listado de pacientes..." podría dividirse en tareas como "Crear interfaz para listar pacientes", "Implementar funcionalidad para agregar un nuevo paciente", "Implementar funcionalidad para editar la información de un paciente", y "Implementar funcionalidad para eliminar un paciente".
- **Estimar el esfuerzo:** Se revisan o se realizan las **estimaciones de esfuerzo** (utilizando puntos de historia o tallas) para las historias. El equipo puede utilizar técnicas como **Planning Poker** para llegar a un consenso sobre la complejidad relativa de cada historia o tarea. Si la discusión revela una falta de claridad o un nuevo entendimiento, las estimaciones pueden ajustarse.
- **Añadir criterios de aceptación:** Se definen claramente los **criterios de aceptación** para cada historia de usuario. Estos criterios describen las condiciones que deben cumplirse para considerar que la historia está completa y funcional. Por ejemplo, para la historia de la lista de médicos, un criterio de aceptación podría ser: "La lista de médicos debe estar ordenada alfabéticamente por apellido" o "El usuario debe poder filtrar la lista por más de cinco especialidades diferentes".
- **Resolver dudas:** El equipo tiene la oportunidad de hacer preguntas y **clarificar cualquier duda** sobre los requisitos, el diseño o los aspectos técnicos de las

historias. Esto ayuda a reducir la ambigüedad y asegura que todos tengan una comprensión clara del trabajo a realizar.

- **Repriorizar si es necesario:** Basándose en la nueva información obtenida durante el refinamiento, el **Product Owner puede reconsiderar la priorización** de algunas historias en el Product Backlog.

El resultado de una sesión de Refinamiento del Backlog es un Product Backlog más claro, detallado y priorizado, con historias de usuario que están mejor definidas y listas para ser incluidas en la planificación del próximo sprint.