

Problem Statements in UX Discovery

<https://www.nngroup.com/articles/problem-statements/>

- ¿A **quién** afecta el problema? (Who?)
- ¿**Cuál** es el problema? (What?)
- ¿**Dónde** ocurre este problema? (Where?)
- ¿**Cuándo** ocurre el problema? (When)
- ¿**Por qué** ocurre el problema? ¿Por qué es importante el problema? (Why?)

Ejemplo: Docente Universitario - Toma de Asistencia

1. Necesidad o Problema del Usuario:

Un docente universitario necesita registrar la asistencia de sus alumnos a clases.

Ejemplo: Docente Universitario - Toma de Asistencia

1. Necesidad del Usuario:

Un docente universitario necesita conocer quiénes y cuántos alumnos asisten a sus clases en los diferentes horarios de dictado para poder determinar la nota de participación en clase, la cual considera el porcentaje de asistencia a las sesiones programadas.

Desarrollo de Técnica 5Ws

Who: *"El docente universitario, responsable de evaluar la participación de los alumnos,"* – Identifica a la persona con el problema.

- **What:** *"necesita conocer en tiempo real la asistencia en cada una de sus clases programadas"* – Define el problema específico.
- **When:** *"en cada una de sus clases programadas"* – Indica cuándo ocurre el problema, es decir, durante las clases programadas.
- **Where:** *"dictado de clases programadas"* – Refleja que el problema ocurre en el contexto de las clases dictadas.
- **Why:** *"para calcular con precisión la nota de participación, dado que esta nota depende directamente del porcentaje de asistencia registrado a lo largo del curso."* – Explica por qué es importante resolver el problema.

Declaración del problema:

"El docente universitario, responsable de evaluar la participación de los alumnos, necesita conocer en tiempo real la asistencia en cada una de sus clases programadas para calcular con precisión la nota de participación, dado que esta nota depende directamente del porcentaje de asistencia registrado a lo largo del curso."

Ejemplo: Docente Universitario - Toma de Asistencia

1. Declaración de Problemas:

El docente universitario, responsable de evaluar la participación de los alumnos, necesita conocer en tiempo real la asistencia en cada una de sus clases programadas para calcular con precisión la nota de participación, dado que esta nota depende directamente del porcentaje de asistencia registrado a lo largo del curso.

2. Requisitos Funcionales (Preliminar):

1. El producto de software deberá permitir registrar la asistencia de los alumnos para cada sesión de clase programada, indicando si asistieron o no.

Ejemplo: Docente Universitario - Toma de Asistencia

1. Declaración de Problemas:

El docente universitario, responsable de evaluar la participación de los alumnos, necesita conocer en tiempo real la asistencia en cada una de sus clases programadas para calcular con precisión la nota de participación, dado que esta nota depende directamente del porcentaje de asistencia registrado a lo largo del curso.

2. Requisitos Funcionales (Detallado):

1. El producto de software deberá permitir que el docente pueda visualizar la lista de horarios de clase en los que imparte docencia.
2. El producto de software deberá listar las fechas y horas de las sesiones de clase programadas para cada uno de los horarios de dictado del docente.
3. El producto de software deberá mostrar la nómina de alumnos matriculados en cada horario de clase, incluyendo su código de estudiante y nombres completos.
4. **El producto de software deberá permitir registrar la asistencia de los alumnos para cada sesión de clase programada, indicando si asistieron o no.**

Ejemplo: Docente Universitario - Toma de Asistencia

1. Declaración de Problemas:

El docente universitario, responsable de evaluar la participación de los alumnos, necesita conocer en tiempo real la asistencia en cada una de sus clases programadas para calcular con precisión la nota de participación, dado que esta nota depende directamente del porcentaje de asistencia registrado a lo largo del curso.

2. Requisitos Funcionales (Detallado):

1. El producto de software deberá permitir que el docente pueda visualizar la lista de horarios de clase en los que imparte docencia.
2. El producto de software deberá listar las fechas y horas de las sesiones de clase programadas para cada uno de los horarios de dictado del docente.
3. El producto de software deberá mostrar la nómina de alumnos matriculados en cada horario de clase, incluyendo su código de estudiante y nombres completos.
4. **El producto de software deberá permitir registrar la asistencia de los alumnos para cada sesión de clase programada, indicando si asistieron o no.**

3. Requisitos No Funcionales:

1. El sistema deberá ser accesible desde cualquier dispositivo con conexión a internet, incluyendo computadoras, tablets y smartphones.
2. El sistema deberá actualizar la lista de asistencia en tiempo real, con un retraso máximo de 2 segundos después de que se registre la asistencia.
3. La interfaz de usuario deberá ser intuitiva y fácil de usar.
4. El sistema deberá garantizar la privacidad de los datos de los alumnos.
5. El sistema deberá estar disponible el 99.9% del tiempo.

Ejemplo: Docente Universitario - Toma de Asistencia

1. Declaración de Problemas:

El docente universitario, responsable de evaluar la participación de los alumnos, necesita conocer en tiempo real la asistencia en cada una de sus clases programadas para calcular con precisión la nota de participación, dado que esta nota depende directamente del porcentaje de asistencia registrado a lo largo del curso.

2. Requisitos Funcionales (Detallado + Innovación):

1. El producto de software deberá permitir que el docente pueda visualizar la lista de horarios de clase en los que imparte docencia.
2. El producto de software deberá listar las fechas y horas de las sesiones de clase programadas para cada uno de los horarios de dictado del docente.
3. El producto de software deberá mostrar la nómina de alumnos matriculados en cada horario de clase, incluyendo su código de estudiante y nombres completos.
4. El producto de software deberá permitir registrar la asistencia de los alumnos para cada sesión de clase programada, indicando si asistieron o no.
5. **El producto de software deberá calcular automáticamente el porcentaje de asistencia para cada alumno, basado en las sesiones programadas y asistidas.**

3. Requisitos No Funcionales (Sin ambigüedades):

6. El sistema deberá ser accesible desde cualquier dispositivo con conexión a internet, incluyendo computadoras, tablets y smartphones.
7. El sistema deberá actualizar la lista de asistencia en tiempo real, con un retraso máximo de 2 segundos después de que se registre la asistencia.
8. La interfaz de usuario deberá ser intuitiva y fácil de usar, **permitiendo al docente registrar la asistencia de una clase completa en menos de 3 minutos.**
9. El sistema deberá garantizar la privacidad de los datos de los alumnos, **utilizando cifrado SSL para todas las comunicaciones y almacenamiento de datos.**
10. El sistema deberá estar disponible el 99.9% del tiempo, **especialmente durante los horarios de clase.**

Análisis de requerimientos de Software:

1. **Claridad en la Necesidad del Usuario:** Asegúrate de que la necesidad del usuario esté claramente conectada con un objetivo específico, como la evaluación de la participación, para dar contexto al porqué de los requisitos.
2. **Detallado en los Requisitos Funcionales:** Se ha agregado un requisito funcional que considera el cálculo automático del porcentaje de asistencia. Esto ayuda a vincular directamente la necesidad del usuario con la funcionalidad del software.
3. **Enfoque en la Usabilidad:** Los requisitos no funcionales incluyen detalles sobre la accesibilidad y la facilidad de uso, lo cual es esencial para un sistema que será utilizado por un docente en situaciones posiblemente estresantes (como pasar lista rápidamente al inicio de una clase).
4. **Privacidad y Seguridad:** En los requisitos no funcionales, se añadió un punto sobre la seguridad de los datos, que es crucial cuando se trata de información personal de estudiantes.
5. **Disponibilidad y Rendimiento:** Se hizo énfasis en la disponibilidad y la capacidad de respuesta del sistema, aspectos clave para que el software sea confiable y eficiente en un entorno académico.

Actividad:

Nueva Necesidad: ¿Cómo puedo asegurarme que un alumno es quien dice que es?
(Suplantación)

Se solicita:

- Defina una nueva problemática.
- Actualice los requerimientos funcionales y no funcionales.