

Requisitos de Software

Proyecto: <nombre>

Integrantes:

|  |  |  |
| --- | --- | --- |
| Nombres y Apellidos | Email | ROL USM |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Elimine esta página y las secciones en marrón antes de entregar el informe.**

# Pauta del Informe correspondiente al Entregable I

## Objetivo

Establecer las bases de compromiso entre cliente y grupo de proyecto con el objetivo de asumir los requisitos necesarios para la construcción de su solución.

Esta versión inicial del Entregable I será refinada en iteraciones posteriores, por lo anterior,el objetivo es proponer una visión, conseguir el “vamos” del cliente, y permitir que el equipo empiece a explorar caminos para el desarrollo.

## Contenido del Informe a Entregar

Contexto del proyecto (máximo 1 página) 3

Objetivo de proyecto (máximo 140 caracteres) 3

Resumen del Proyecto (máximo 1/2 página) 3

Modelo de Dominio (Inicial) 4

Actores y tareas clave 5

Requisitos funcionales y no-funcionales 6

Casos de Uso iniciales 7

Creación de softgoals 8

Creación del árbol de utilidad de la solución 9

Identificación preliminar de situaciones potencialmente negativas para el proyecto 10

## Administrativo

1. El informe deberá ser firmado o validado por el cliente
2. El plazo de entrega del presente informe es hasta el **16/abril/2017 a las 23:55** horas y debe ser subido a Github (en formato digital PDF). Además se debn incluir las versiones de los modelos realizados para este informe. Por cada día de atraso existe una penalización de 20 puntos.

# Contexto del proyecto (máximo 1 página)

## Objetivo de proyecto (máximo 140 caracteres)

Objetivo central del sistema que se construirá; ES UNA SOLA COSA, de la que se desprenden (y a la que apoyan) todas las demás.

## Resumen del Proyecto (máximo 1/2 página)

Resumen del proyecto. Describa el problema abordado, los criterios de éxito, las alternativas consideradas, el enfoque propuesto (a grandes rasgos), y cómo se piensa (de momento) validar que se haya logrado el objetivo.

# Modelo de Dominio (Inicial)

Toda aplicación existe para informar a alguien sobre la realidad y/o para permitirle manipularla; por ejemplo, una aplicación clínica podría dirigir hacer una ecografía (le informa al doctor), guardarla en una BD (algo interno a la aplicación), y enviarla por e-mail (actúa sobre el exterior). Por lo tanto, toda aplicación tiene una “imagen del mundo”, una visión restringida de la realidad, que incluye las entidades sobre las que debe mantener información actualizada y/o a las que debe manipular; la aplicación debe proveer interfaces con usuarios u otros sistemas para que los cambios en el estado real se reflejen en el modelo, y viceversa.

En la figura ejemplo sobre prestamos de equipos, hay 9 tipos de entidades y 15 tipos de relaciones.

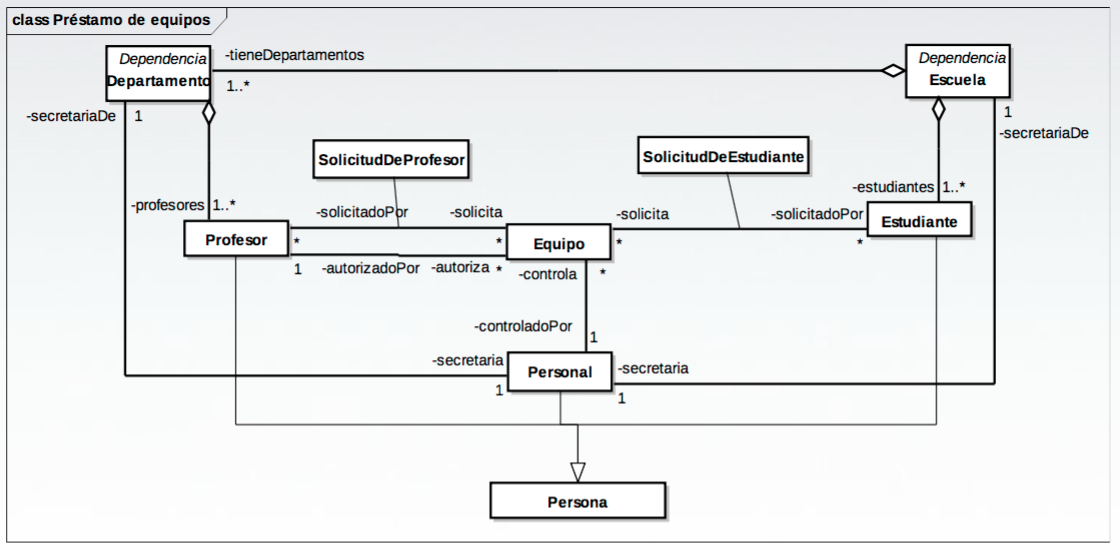


Figura 1: Ejemplo de modelo de dominio

Tabla 1: Entidades del Dominio

|  |  |
| --- | --- |
| **Entidad** | **Descripción (1 línea)** |
|  |  |
|  |  |

# Actores y tareas clave

Identifique los actores que interactúan con el sistema. Estos son los tipos de usuarios (no clientes) que su aplicación tendrá cuando exista.

|  |  |
| --- | --- |
| **Actor** | **Descripción (1 línea)** |
|  |  |
|  |  |

Las tareas clave son aquellas cosas que la aplicación debe poder hacer cuando se entregue. Son parte de sus criterios de éxito, y se pueden derivar de la proposición de valor de la aplicación. Algunos ejemplos:

* Crear un perfil para todo “amigo” referenciado por un usuario.
* Detectar automáticamente usuarios duplicados.

|  |  |
| --- | --- |
| **Tarea clave** | **Descripción (máximo 3 líneas)** |
|  |  |
|  |  |

Note que los actores deberían cuadrar con sus tareas clave: alguien realiza estas tareas, y si son clave, es porque serán esenciales para algún actor importante.

# Requisitos clave funcionales y extra-funcionales

|  |  |
| --- | --- |
| **Req. funcional** | **Descripción y medición (máximo 2 líneas)** |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Req. extra-funcional** | **Descripción y medición (máximo 2 líneas)** |
|  |  |
|  |  |

Los requisitos funcionales y no-funcionales deben ser ESPECÍFICOS y VERIFICABLES (a la entrega final del ejecutable). Para los extr-funcionales, no haga afirmaciones genéricas o inverificables como “el sistema debe ser rápido y amigable”. Ejemplos aceptables:

* [Disponibilidad] Podrá atender a 100 usuarios simultáneos.
* [Escalabilidad] Podrá crecer (agregando recursos pero sin cambiar el código) hasta 10.000 usuarios simultáneos.
* [Desempeño] El tiempo promedio de creación de un perfil será <= 30s.
* [Usabilidad] La satisfacción de los usuarios en el cuestionario SUS será >= 50%.
* [Disponibilidad] El uptime del servidor será de 99,9%, y en caso de caída el tiempo de recuperación será de 10min.

Importante: NO INVENTE REQUISITOS; si no parece haber exigencias específicas, DIGA ESO (y que su cliente lo firme).

# Casos de Uso iniciales

Los casos de uso son una técnica para diseño preliminar, y permiten documentar en detalle las acciones que los usuarios podrán realizar con la aplicación. Por lo tanto, esta parte del informe es por naturaleza la más detallada, y por lo mismo la que probablemente más cambie durante el proyecto.

Diagrama ejemplo:

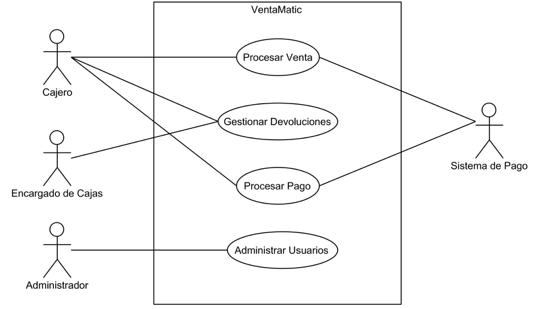


Figura 2: Ejemplo de casos de uso

Definir al menos **3 casos de uso no-triviales** e incluir el flujo principal y alternativo de acción.

|  |  |
| --- | --- |
| Nombre |  |
| Descripción  (max: 5 líneas) |  |
| Pre-condiciones |  |
| Post-condiciones |  |
| Flujo principal |  |
| Flujo alternativo |  |

# Identificación de softgoals

A partir de la información obtenida de los requisitos extra-funcionales, identifique y *softgoals* (metas blandas), descomponiéndolos en sub-metas más específicos para este proyecto. Describa meta y sub-metas en un diagrama (que sea legible y quepa en una página).

Diagrama ejemplo:

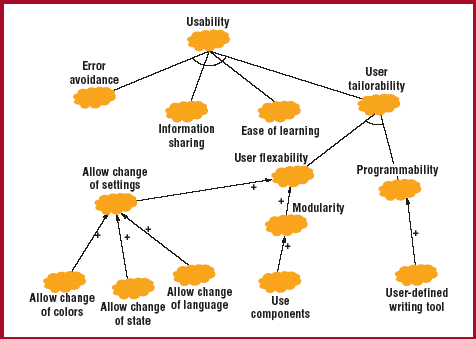


Figura 3. Ejemplo de softgoal

# Creación del árbol de utilidad de la solución

Los árboles de utilidad representan gráficamente los *atributos de calidad* de un sistema de software y su refinamiento hasta nivel de *escenarios,* especificando nivel de *prioridad* (según el cliente) y de *dificultad* (según el equipo).

Especifique los atributos de calidad más importante de su proyecto; obtenga los escenarios correspondientes; y para cada escenario registre el nivel de prioridad (H= High, M= Medium, y L=Low) dado por el cliente y la *dificultad* estimada por el equipo .

Árbol ejemplo (legible, máx. 1 página, entre 4 y 12 escenarios):

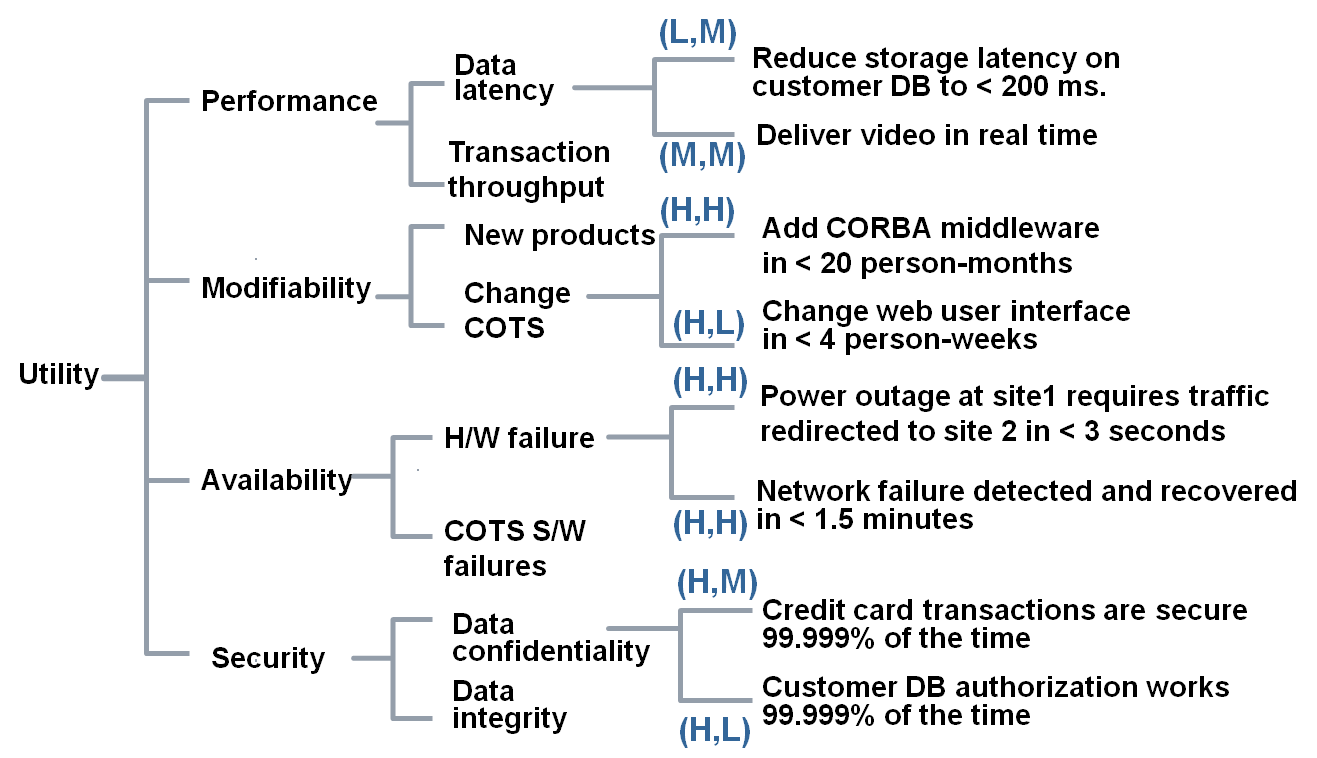


Figura 4: Ejemplo de árbol de utilidad

# Identificación preliminar de riesgos para el proyecto

Identifique y describa (someramente) los 3 riesgos principales asociados al desarrollo del sistema; seleccione y justifique el mayor riesgo que el equipo determinó; y proponga medidas de mitigación para él.