Índice

1. Entregables

Entregables

Todos los entregables serán subidos a Git, con lenguaje Markdown. Se entregarán tres tipos de ficheros:

1. Un documento de extracción de requisitos en el que aparezcan las partes interesadas, los datos que debe almacenar la aplicación, los requisitos funcionales y no funcionales, y finalmente, la priorización de los mismos.

2. Casos de uso: un archivo Markdown por cada caso de uso, así como un diagrama de casos de uso final.

3. Historias de usuario: un archivo Markdown con todas las historias de usuario.

- El representante del grupo se encargará de enviar por Moodle el **enlace al repositorio de GitHub** con todos los elementos de la entrega
- La entrega es **obligatoria**
- La evaluación que se hará de la misma es **orientativa**: se indicará la valoración general y puntos a corregir, de cara a la entrega final

2. Desarrollo de las clases

Desarrollo de las clases

- Se realizará una entrevista con cada uno de los grupos durante las sesiones de prácticas.
- Los alumnos deben identificar los requisitos, refinarlos, especificarlos y priorizarlos.
- Una vez estén terminadas todas las entrevistas, se podrán hacer preguntas sueltas al profesor de la asignatura para sugerir y/o preguntar nuevas funcionalidades de la aplicación.

Desarrollo de las clases

- Primera sesión: Entrevistas e identificación de requisitos
- Segunda sesión: Elaboración de historias de usuario y casos de uso

3. Problema a resolver

Problema a resolver

Es necesario el desarrollo de un software de gestión para una clínica médica. Este software será utilizado por la secretaría de la clínica y los doctores para registrar los datos relativos a los pacientes, las citas, el tratamiento y el historial médico pasado.

Restricciones

- El lenguaje de programación es C++.
- El lenguaje de documentación será Markdown.
- El proyecto se realizará haciendo uso del sistema de control de versiones Git y la plataforma GitHub para el almacenamiento del repositorio de forma remota.
- El historial de cambios queda guardado en las cuentas de Git por lo que la evaluación será incremental y no servirá subir los ficheros de un día para otro en Git.

4. Requisitos

Requisitos

Describen los **servicios que debe proporcionar** el sistema y sus **restricciones operativas**

Un requisito puede ser una simple declaración abstracta de alto nivel o bien una **definición detallada y formal** de una función del sistema

Requisitos funcionales y no funcionales

Requisitos funcionales

Expresa lo que debería hacer el sistema. Las funcionalidades que este debe proporcionar y especifican la manera en que el sistema debe reaccionar (o no debe reaccionar) antes determinadas entradas y/o el comportamiento del mismo en determinadas situaciones.

Qué debería hacer el sistema

Requisitos no funcionales

No se refieren a funciones específicas que proporciona el sistema: son restricciones de los servicios o funciones ofrecidas por el sistema.

Cómo debería ser el sistema

Actividades de análisis de requisitos

Se distinguen cuatro actividades:

- **Extracción** de requisitos: Proceso de descubrimiento, por parte de los clientes, de los requisitos que desean.
- **Análisis** de requisitos: razonamiento sobre los requisitos obtenidos en la la extracción. Detectar y resolver posibles inconsistencias o conflictos.
- **Especificación** de requisitos: redacción o registro de los requisitos, haciendo uso de lenguaje natural, formal, modelos, gráficos etc.
- **Validación** de requisitos: proceso de confirmación por parte de los usuarios/clientes de los requisitos especificados en el que se comprueba su validez, consistencia y completitud.

4.1. Casos de uso

Casos de uso

- Describen una **actividad o tarea** que involucra al sistema
- Comunmente usados para **describir los requisitos funcionales**
- Son desarrollados por uno o más **actores**
- Tienen un **flujo principal** y, opcionalmente, **flujos alternativos**
- Pueden **relacionarse** entre ellos: inclusión, extensión o generalización
- Son una buena base para elaborar **pruebas del software**: de regresión, del sistema, de integración...

Ejemplo de caso de uso

Buscar usuario por nombre y apellidos

ID: 04 **Descripción**: Se introduce el nombre y apellidos del usuario que se busca y el sistema lo muestra.

Actores principales: Administrador **Actores secundarios**: Usuario

Precondiciones:

- Ninguna

Flujo principal:

1. El administrador desea consultar los datos de un usuario
2. El administrador abre el cuadro de diálogo de búsqueda en el menú principal
3. El administrador introduce el nombre y apellidos del usuario
4. El sistema muestra por pantalla los datos del usuario

Postcondiciones:

- Se muestran al administrador las distintas operaciones relativas al usuario

Flujos alternativos:

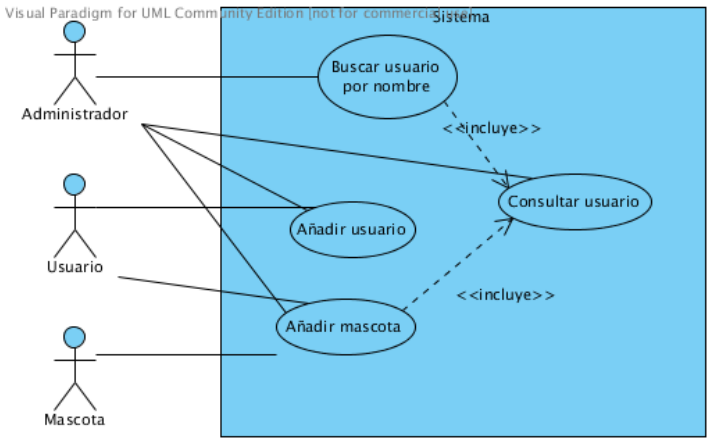
4.a. Si no existe el usuario, se muestra un mensaje de error

Diagrama de casos de uso

Muestra de un único vistazo:

- Los sistemas
- Los actores
- Los casos de uso
- Qué actores están involucrados en qué casos de uso
- Qué relaciones existen entre los actores y casos de uso

Ejemplo de diagrama de casos de uso



1

4.2. Historias de usuario

Historias de usuario

- Describe de una **funcionalidad** que debe incorporar el sistema, desde el punto de vista del usuario
- Su implementación aporta **valor** al cliente

«Como (usuario), quiero conseguir (meta) para (objetivo)»

Contiene:

- Identificador
- Nombre descriptivo
- Descripción de la funcionalidad
- Criterios de validación
- Opcionalmente:
  - Prioridad
  - Estimación de tiempo
  - Desarrollador responsable

Ejemplo de historia de usuario

[1] Más sobre casos de uso y diagramas de casos de uso:
«Aprendiendo UML en 24 horas» `https://www.u-cursos.cl/ingenieria/2008/1/CC51H/1/material_docente/bajar?id_material=160144`

ID: 06 **Nombre**: Buscar usuario por nombre y apellidos

Prioridad (de 1 a 10): 7 **Puntos estimado**: 3 **Iteración**: 1

Responsable: Javier Barbero

Descripción

Como *administrador* quiero *localizar los datos de un usuario* para *consultar su información y las operaciones posibles sobre el mismo*

Validación

- Se debe poder localizar cualquier usuario
- Los resultados de la búsqueda pueden mostrar cero o más resultados
- No es necesario utilizar el nombre completo
- Se deben mostrar todos los datos del usuario
- Se deben mostrar la lista de mascotas del usuario
- Se deben mostrar las operaciones posibles sobre el usuario