

Proyecto Final

Nombre del Proyecto:

☐

Club Equinoccio

Integrantes:

☐

Benjamín Cortés
Helena Anwandter
Noelia Muñoz
Tamar Rojas
Juan Riquelme
Benjamín Ríos

Líder del Proyecto. (Elijan a un referente del equipo, la persona que va a organizar el proyecto y va a ser el nexo entre los tutores y el equipo)

☐

Benjamín Cortés

Breve descripción del proyecto. Definir en no mas de 20 palabras de que se trata el sistema a realizar

Pagina Web; Club De Montaña, form inscripcion, Salidas a terreno, act club, Cursos, dudas, etc.

Ciclo de vida de un proyecto de Software

El ciclo de vida de un proyecto hace referencia a las etapas por las cuales pasa el proyecto desde su concepción hasta su implementación y puesta en marcha final.

A continuación se muestran las etapas por las cuales va a ir pasando nuestro proyecto.

El ciclo de vida básico de un *software* consta de los siguientes procedimientos:

- 1- Definición de objetivos:** define la finalidad del proyecto.
- 2- Análisis de los requisitos y su viabilidad:** recopila, examina y formula los requisitos del cliente y examina cualquier restricción que se pueda aplicar.
- 3- Diseño general:** requisitos generales de la arquitectura de la aplicación.
- 4- Diseño en detalle:** definición precisa de cada subconjunto de la aplicación.

5- Programación (programación e implementación): implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.

6- Prueba de unidad: prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.

7- Integración: garantiza que los diferentes módulos se integren con la aplicación. Este es el propósito de la prueba de integración que está cuidadosamente documentada.

8- Prueba beta (o validación): garantiza que el *software* cumple con las especificaciones originales.

9- Documentación: sirve para documentar información necesaria para los usuarios del *software* y para desarrollos futuros.

10- Implementación

11- Mantenimiento: comprende todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del *software* (mantenimiento continuo).

El orden y la presencia de cada uno de estos procedimientos en el ciclo de vida de una aplicación dependen del tipo de modelo de ciclo de vida acordado entre el cliente y el equipo de desarrolladores.

Durante la ejecución del proyecto final vamos a atravesar por varias de estas etapas, para poder llevar a cabo la mejor solución para nuestro problema a resolver.

Etapas 1. Definición del objetivo general y los objetivos específicos del proyecto.

Identificar los objetivos del proyecto

El primer paso para definir los objetivos del proyecto es identificar qué aspectos del proyecto son realmente importantes; y aquí debemos separar dos ámbitos:

- Los objetivos del proyecto a nivel del producto que este debe entregar a la organización, cliente, u otros involucrados.
- Aquellos aspectos internos del proyecto que son más importantes, y por tanto van a requerir mayor atención, para la consecución de los objetivos.

Para el primer punto deberemos empezar identificando y priorizando los objetivos a través de la información que nos den los involucrados (cliente, sponsor, usuarios, etc.),

y de los documentos contractuales que puedan dar origen al proyecto (pedido, oferta, etc.). El tema de priorizar es muy importante, ya que aunque nuestro objetivo será cumplir con todo, la realidad es que nos podemos encontrar con situaciones que nos impidan poder hacerlo, por lo que debemos ser capaces de saber cuáles objetivos pueden ser más flexibles.

Por otro lado, una vez hayamos planificado el proyecto, veremos que no todas las tareas o recursos involucrados van a tener el mismo efecto a la hora de conseguir los objetivos. Por lo que nuestra atención se deberá centrar en aquellos que sean más relevantes. Esto se entiende bien cuando planificamos con Camino Crítico, ya que las tareas dentro de este camino son las que marcan la duración total del proyecto, por lo que pasan a ser más relevantes que el resto.

Definir correctamente los objetivos del proyecto

“No se puede controlar lo que no se puede medir”, por tanto para controlar si estamos consiguiendo o no los objetivos del proyecto, debemos definir estos de forma que sean medibles.

Objetivo general

Los objetivos generales son aquellos que se formulan para completar una meta o señalar el fin de un proyecto o investigación.

El objetivo general en el caso de un proyecto siempre va relacionado con el título del mismo, es decir, es el mismo título con la diferencia que se encuentra redactado en infinitivo.

Es por ello que el objetivo general es el paso más sencillo de formular al momento de realizar una investigación a diferencia del planteamiento del problema y los marcos legales, metodológicos y conceptuales.

Incluso es más sencillo de redactar que los objetivos específicos.

Al formular el objetivo general es conveniente tomar en cuenta que debe ser:

- **Cualitativo:** Debe indicar calidad.
- **Integral:** Debe ser capaz de integrar los objetivos específicos y para ello debe estar encabezado por un verbo de mayor nivel.
- **Terminal:** Debe definir un plazo.

Objetivos específicos

Se trata de objetivos que provienen del objetivo general, son objetivos redactados con un verbo de menor nivel al verbo en el que fue redactado el objetivo general del que provienen.

Los objetivos específicos son una serie de fines que llevan a cumplir el objetivo general.

Los objetivos específicos deben tener:

Claridad: El lenguaje en el que se encuentran redactados debe ser claro y preciso.

Factibilidad: Los objetivos deben poder ser cumplidos a través de la metodología seleccionada.

Pertinencia: Deben relacionarse lógicamente con el problema a solucionar.

Diferencia entre objetivo general y específico

- Un objetivo general precisa la finalidad de un estudio, investigación, organización o empresa.
- Los objetivos generales son actividades a cumplir para llevar a cabo el objetivo general.

En nuestro caso vamos a pensar que nuestro sistema va a quedar completamente operativo e instalado, entonces por ejemplo para el tinder de mascotas un objetivo sería:

Tener 1500 usuarios de la aplicación al finalizar diciembre del 2020

Y un objetivo específico sería:

Invertir 100 pesos en campañas de Marketing para fidelizar la aplicación.

Etapas 2. Análisis de requisitos y Diseño

En esta etapa vamos a “entrevistar” al ideador del proyecto para poder determinar qué es lo que debe hacer el Sistema, una vez relevados los datos debemos validar con el ideador las funcionalidades detectadas.

La toma de requerimientos es de las etapas mas importantes, ya que todo nuestro sistema se fundamentará en el relevamiento que hagamos en esta etapa.

Una vez definida las funcionalidades, estas se documentan mediante Casos de Uso.
Qué son los casos de uso?

Casos de Uso:

En ingeniería del software, un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final. En ocasiones, se utiliza a usuarios sin experiencia junto a los analistas para el desarrollo de casos de uso.

En otras palabras, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo

Características

Los casos de uso evitan típicamente la jerga técnica, prefiriendo la lengua del usuario final o del experto del campo del saber al que se va a aplicar. Los casos del uso son a menudo elaborados en colaboración por los analistas de requerimientos y los clientes.

Cada caso de uso se centra en describir cómo alcanzar una única meta o tarea de negocio. Desde una perspectiva tradicional de la ingeniería de software, un caso de uso describe una característica del sistema. Para la mayoría de proyectos de software, esto significa que quizás a veces es necesario especificar diez o centenares de casos de uso para definir completamente el nuevo sistema. El grado de la formalidad de un proyecto particular del software y de la etapa del proyecto influenciará el nivel del detalle requerido en cada caso de uso.

Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Un caso de uso contiene una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema. Los casos de uso no describen ninguna funcionalidad interna (oculta al exterior) del sistema, ni explican cómo se implementará. Simplemente muestran los pasos que el actor sigue para realizar una tarea.

Un caso de uso debe:

- describir una tarea del negocio que sirva a una meta de negocio
- tener un nivel apropiado del detalle

- ser bastante sencillo como que un desarrollador lo elabore en un único lanzamiento
- Situaciones que pueden darse:
- Un actor se comunica con un caso de uso (si se trata de un actor primario la comunicación la iniciará el actor, en cambio si es secundario, el sistema será el que inicie la comunicación).
 - Un caso de uso extiende otro caso de uso.
 - Un caso de uso utiliza otro caso de uso.

Ventajas

La técnica de caso de uso tiene éxito en sistemas interactivos, ya que expresa la intención que tiene el actor (su usuario) al hacer uso del sistema.

Como técnica de extracción de requerimiento permite que el analista se centre en las necesidades del usuario, qué espera éste lograr al utilizar el sistema, evitando que la gente especializada en informática dirija la funcionalidad del nuevo sistema basándose solamente en criterios tecnológicos.

A su vez, durante la extracción (elicitation en inglés), el analista se concentra en las tareas centrales del usuario describiendo por lo tanto los casos de uso que mayor valor aportan al negocio. Esto facilita luego la priorización del requerimiento.

Limitaciones

Los casos de uso pueden ser útiles para establecer requisitos de comportamiento, pero no establecen completamente los requisitos funcionales ni permiten determinar los requisitos no funcionales. Los casos de uso deben complementarse con información adicional como reglas de negocio, requisitos no funcionales, diccionario de datos que complementen los requerimientos del sistema. Sin embargo la ingeniería del funcionamiento especifica que cada caso crítico del uso debe tener un requisito no funcional centrado en el funcionamiento asociado.

Buenas prácticas y recomendaciones de uso

Los casos de uso, como el resto de los requisitos, deben tener una redacción cuidada para evitar problemas de interpretación. En general, algunas recomendaciones a tener en cuenta son:

- El caso de uso debe describir qué debe hacer el sistema a desarrollar en su interacción con los actores y no cómo debe hacerlo. Es decir, debe describir sólo comportamiento observable externamente, sin entrar en la funcionalidad interna del sistema.
- El nombre del caso de uso debe ilustrar el objetivo que pretende alcanzar el actor al realizarlo.
- El caso de uso debe describir interacciones con los actores sin hacer referencias explícitas a elementos concretos de la interfaz de usuario del sistema a desarrollar.
- La invocación de unos casos de uso desde otros casos de uso (lo que se conoce como inclusión, o extensión si es condicional, en UML), sólo debe usarse como un mecanismo para evitar repetir una determinada secuencia de pasos que se repite en varios casos de uso. Nunca debe usarse para expresar posibles menús de la interfaz de usuario.

- Se debe ser cuidadoso al usar estructuras condicionales en la descripción del caso de uso, ya que los clientes y usuarios no suelen estar familiarizados con este tipo de estructuras, especialmente si son complejas.
- Se debe intentar que todos los casos de uso de una misma ERS estén descritos al mismo nivel de detalle.
- En los diagramas de casos de uso, debe evitarse que se crucen las líneas que unen los actores a los casos de uso.
- En caso de usar github, discutir el nombre de las variables que se van a usar y dejarlas guardadas en un pdf para evitar problemas y conflictos en el repositorio.

A continuación les dejo unos ejemplos de un caso de uso del tinder de mascotas.

Recuerden identificar TODAS las funcionalidades que quieran incluir en el Sistema y diagramarlo como caso de uso. (Para el tinder de mascotas por ejemplo: Login de Usuario, Registración Usuario, Crear mascota, Buscar mascota cercana, Like mascota, etc) La idea es hacer un caso de uso de manera sencilla.

CU 1. Login del usuario

CU 1. Login del usuario	
Descripción	El usuario ingresa al sistema mediante usuario y clave
Precondición	(Acá va algo sólo si debe existir algo antes que esta acción) Por ejemplo: El usuario debe estar dado de alta.
Acciones (Siempre se describe el camino feliz, osea la situación exitosa)	1- El usuario ingresa nombre y clave 2- El usuario presiona Aceptar 3- El sistema valida que exista un usuario con esas credenciales. 4- El sistema muestra la pantalla principal de la aplicación. (Debemos colocar paso a paso la interacción entre el/los actores y el sistema)
Postcondición	El usuario se encuentra logueado y listo para usar el sistema ¿Qué pasa cuando termina el caso de uso?
Nota	Acá va alguna aclaración
Actores	Usuario Personas que interactúan con el Sistema
Alternativas	4.1- El sistema muestra pantalla de error con el mensaje "Sus credenciales son incorrectas, intente nuevamente." Acá van las acciones en el caso de que el camino no sea feliz, por ejemplo, que pasa si el usuario o la clave con incorrectas?

CU 2. Registrar usuario

CU 2. Registrar usuario	
Descripción	El usuario nuevo desea registrarse para poder usar el sistema
Precondición	-

Acciones	1- El usuario ingresa a la pantalla de registración. 2- El sistema le solicita los datos personales. 3- El usuario ingresa los datos solicitados y presiona “Registrar”. 4- El sistema envía un mail confirmando la registración exitosa.
Postcondición	El usuario se registrado y listo para loguearse.
Nota	-
Actores	Usuario
Alternativas	-