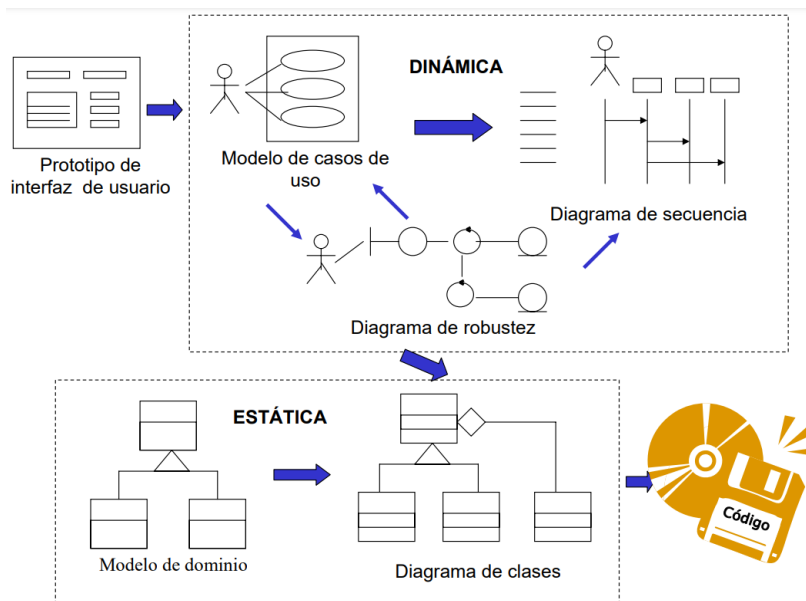


Guía de la fase de Ejecución

3. Ejecución

a. Como se lleva a cabo el proyecto de acuerdo a la metodología de desarrollo seleccionada

La metodología de desarrollo que se seleccionó para el proyecto de software se denomina ICONIX es una metodología pesada-ligera de Desarrollo del Software que se halla a medio camino entre un RUP (Rational Unified Process) y un XP (eXtreme Programming).



Fases

Iconix se estructura en cuatro fases. La primera de ellas es el análisis de requisitos, seguida del análisis y diseño preliminar, a continuación viene el diseño y finaliza con su implementación.

1. Análisis de Requisitos

En esta primera fase se realiza un Modelo de Dominio, que no es más que un Diagrama de Clases extremadamente simplificado. Este modelo contiene únicamente aquellos objetos de la vida real cuyo comportamiento o datos deben ser almacenados en el sistema.

Requisitos nuevos o modificados Sistema nuevo o modificado Proceso de Desarrollo de Software Requisitos nuevos o modificados Sistema nuevo o modificado Proceso de Desarrollo de Software A partir de este pequeño modelo, se realiza un pequeño prototipo basándose en la storyboard de la interfaz gráfica obtenida previamente, el cual se mostrará al cliente y se refinó en sucesivas reuniones. Normalmente este prototipo suele converger en dos o tres iteraciones.

Una vez el prototipo ya es final y se han obtenido todos los requisitos del sistema por parte del cliente, se procede a realizar los casos de uso. Estos diagramas de casos de uso se agrupan en diagramas de

paquetes (es decir, utilizan referencias entre diagramas de casos de uso para simplificar su lectura) y se asocia cada requisito a un caso de uso para obtener la ya mencionada anteriormente trazabilidad.

2. Análisis y Diseño Preliminar

A partir de cada caso de uso se obtienen sus correspondientes fichas de caso de uso. Cabe destacar que estas fichas no pertenecen al UML. He aquí un ejemplo de ficha para que se entienda mejor:

Nombre:	Crear mensaje foro
Autor:	Joaquin Gracia
Fecha:	24/08/2003
Descripción:	Permite crear un mensaje en el foro de discusión.
Actores:	Usuario de Internet logeado.
Precondiciones:	El usuario debe haberse logeado en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El actor pulsa sobre el botón para crear un nuevo mensaje. 2. El sistema muestra una caja de texto para introducir el título del mensaje y una zona de mayor tamaño para introducir el cuerpo del mensaje. 3. El actor introduce el título del mensaje y el cuerpo del mismo. 4. El sistema comprueba la validez de los datos y los almacena.
Flujo Alternativo:	<ol style="list-style-type: none"> 4. El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija
Poscondiciones:	El mensaje ha sido almacenado en el sistema.

La ficha está formada por un nombre, que suele ser el del caso de uso, posee una breve descripción (generalmente en vista usuario, es decir, que hace de forma intuitiva, no como), una precondición que debe cumplir antes de iniciarse, una postcondición que debe cumplir al terminar si termina correctamente, un flujo normal que sigue el sistema en caso de que todo vaya correctamente y un flujo alternativo en caso de que haya cualquier problema. El resto de campos son opcionales.

Después será necesario realizar lo que se conoce como Diagrama de Robustez, el cual pertenece al proceso Iconix y tampoco forma parte del UML. A continuación describiremos cómo se realiza un diagrama de este tipo.

El diagrama de Robustez se divide en:

- **Objetos fronterizos:** usado por los actores para comunicarse con el sistema.
- **Objetos entidad:** son objetos del modelo del dominio.
- **Objetos de Control:** es la unión entre la interfaz y los objetos de entidad.

El objetivo del diagrama de robustez es añadir nuevas relaciones a los diagramas de clase, de forma que ya tendremos un esqueleto aceptable de la arquitectura y del diseño a partir del cual podremos proseguir nuestro proceso. Con esto y las fichas, refinamos el



diagrama de clases tanto como sea necesario y obtenemos una nueva versión preparada para la siguiente fase.

3. Diseño

En esta fase se procede a realizar los diagramas de secuencia, los cuales derivan directamente de las fichas de caso de uso. Obsérvese como, los diagramas de secuencia se relacionan con fichas de caso de uso que se relacionan con casos de uso que se relacionan con requisitos. Esto implica que una vez finalizado el diseño, tras refinar nuevamente el diagrama de clases, podremos verificarlo directamente gracias a este factor de trazabilidad, y prepararnos para la siguiente fase. En caso de que no estemos satisfechos con el resultado, será necesario repasar todo el proceso hasta que éste sea correcto. Es vital que los requisitos se satisfagan correctamente para el éxito del proyecto.

4. Implementación

De cara a poder distribuir el software correctamente, puede ser adecuado realizar un diagrama de componentes en algunos casos, pero no siempre es necesario. En cualquier caso, aquí es donde se escribe el código tal y como fue especificado en las fases anteriores y se planean las pruebas basándonos en los requisitos iniciales, al nivel que fuese necesario. Aquí es donde hacemos uso real de la trazabilidad y donde realmente ponemos en práctica esa garantía de calidad que tanto hemos mencionado. Después de tener un buen diseño, es cuestión de crear un buen software a partir de ese diseño, y mediante los testeos y pruebas adecuados podemos garantizar que el sistema final cumple con los requisitos iniciales y por tanto proceder a su entrega.



Caso Práctico del Proyecto Agenda

En el transcurso de la unidad 2 y 3 hemos desarrollado las fases anteriormente mencionadas, pero adaptándolas a nuestras necesidades y conocimientos.

1. Análisis de requisitos.

para llevar a cabo esta fase nuestro grupo de trabajo obtuvo un listado de requerimientos funcionales y no funcionales:

a. Requerimientos Funcionales

Código	Nombre	Descripción
RF001	Gestionar evento - Lourdes	<p>El sistema permitirá realizar la gestión de un evento.</p> <ul style="list-style-type: none">• El usuario podrá ingresar eventos especificando cuándo se realizará, especificando rangos de tiempo, tendrá un título, una categoría y por defecto el evento quedará con estado "no realizado".• El sistema le permitirá al usuario poder eliminar un evento si así lo estima conveniente.• El usuario tiene la opción de editar los eventos, ya sea fechas, título, estado o categoría.• El sistema listará todos los eventos con su respectivo detalle que han sido registrados por el usuario.
RF002	Gestionar contacto	<p>El sistema permitirá al usuario:</p> <ul style="list-style-type: none">• Añadir contactos: Cada usuario podrá agregar nuevos contactos en su agenda personal.• Modificar contactos: Los usuarios podrán cambiar la información acerca de los contactos de su agenda personal.• Eliminar contactos: Los usuarios podrán eliminar contactos de su agenda personal.• Buscar contactos: Los usuarios podrán buscar entre los contactos agregados a su agenda personal.• Compartir Contactos: Cada usuario podrá compartir contactos de su agenda personal a otro(s) contactos de su misma agenda.
RF003	Gestionar calendario	<p>El sistema permitirá al usuario:</p>



		<ul style="list-style-type: none">● Modificar calendario: Los usuarios podrán cambiar la información acerca del calendario de su cuenta.● Visualizar calendario: Los usuarios podrán acceder al calendario y visualizar toda su información.● Compartir calendario: Cada usuario podrá compartir su calendario, permitiendo el acceso a otros usuarios y que puedan modificarlo.
RF004	Gestionar archivo	<p>El sistema debe permitir gestionar los archivos:</p> <ul style="list-style-type: none">● Cargar archivos: los usuarios pueden subir archivos, como documentos, imágenes y hojas de cálculo, a la plataforma y asociarlos con eventos o tareas específicas en la agenda empresarial.● Compartir archivos: los usuarios pueden compartir archivos con otros miembros del equipo o con clientes externos, y establecer permisos de lectura y edición para controlar quién tiene acceso a cada archivo.● Buscar archivos: los usuarios pueden buscar archivos por nombre, fecha, tipo de archivo, y otros criterios relevantes.● Descargar archivos: los usuarios pueden descargar archivos desde la plataforma y guardarlos en su dispositivo para su uso fuera de línea.● Eliminar archivos: los usuarios pueden eliminar archivos de la plataforma si ya no son necesarios.
RF005	Gestionar Notificación	<p>El sistema deberá permitir que los usuarios reciban, visualicen y eliminen notificaciones automáticas y personalizadas sobre eventos y tareas próximas en su agenda empresarial.</p> <ul style="list-style-type: none">● Recibir notificación: Los usuarios recibirán notificaciones sobre algún evento próximo en el calendario.● Visualizar notificaciones: Los usuarios podrán visualizar un buzón con las notificaciones que ha recibido.



		<ul style="list-style-type: none"> • Eliminar notificaciones: Los usuarios podrán seleccionar las notificaciones que desea eliminar. • Personalizar notificaciones: Los usuarios podrán planificar cómo se van a recibir las notificaciones para los eventos próximos, especificando hora y una breve descripción. • Generar Recomendaciones de forma autónoma: El sistema deberá crear recomendaciones cada cierto tiempo según desee el usuario (cada 7 días por defecto), basándose en las tareas y eventos creados por el usuario (Esto se presentará en formato de notificaciones, al igual que el usuario va a poder configurar la frecuencia con la que recibe las notificaciones o si desea activar o no la función).
RF006	Sincronizar Agenda	El usuario tiene la opción de sincronizar su agenda de escritorio con la aplicación de la agenda en su celular u otros dispositivos, teniendo una actualización en tiempo real, con el menor desfase de tiempo posible.
RF007	Gestionar Categoría	<ul style="list-style-type: none"> • El sistema deberá contar con categorías estándar (Tarea, Trabajo, nutrición, pasatiempos), pero además deberá permitir generar categorías personalizadas. • El sistema permite eliminar una categoría • El sistema le va a permitir listar las categorías existentes. • El usuario tiene la opción de editar las categorías
RF008	Gestionar Cuenta	<p>El sistema permitirá realizar la gestión de una cuenta.</p> <ul style="list-style-type: none"> • El usuario podrá registrar una cuenta personal en el sistema especificando su nombre de usuario, clave, cédula, celular y correo electrónico. • El sistema le permitirá al usuario poder eliminar su cuenta si así lo estima conveniente. • El usuario tiene la opción de editar los datos que se registraron al crear la cuenta.



		<ul style="list-style-type: none"> Una vez el usuario esté registrado, podrá iniciar sesión y realizar cambios dentro del entorno de su agenda.
--	--	--

b. Requerimientos no funcionales

Código	Nombre	Descripción
RNF001	Interfaz Gráfica de Usuario	El sistema debe disponer de una interfaz lo más simple posible (sin perder funcionalidades), permitirá el poder mostrar en pantalla los elementos esenciales del sistema de agendas. Así mismo, el mantener las diferentes interfaces minimalistas contribuirá a un mejor rendimiento.
RNF002	Facilidad de uso	El sistema debe poseer una facilidad de aprendizaje , seguridad al usar (implementando verificaciones de las acciones a realizar), interfaz intuitiva , presentación de la información (funciones, tareas, instrucciones, ayuda) clara y organizada , para obtener una mayor aceptación por los diferentes usuarios.
RNF003	Seguridad	El sistema debe garantizar la protección de la información confidencial de los usuarios, mediante el uso de técnicas de autenticación , autorización y la encriptación de datos almacenados
RNF004	Portabilidad	La aplicación debe tener la capacidad de funcionar en dispositivos (iOS, Android) y sistemas operativos (Windows, Linux, MacOS) diferentes , sin tener la necesidad de realizar cambios significativos en el código o realizar un sistema específico para cada plataforma.
RNF005	Disponibilidad	El sistema debe estar disponible para su uso la mayor parte del tiempo por año , teniendo en consideración los lapsos de tiempo planificados para mantenimiento y los mantenimientos emergentes que se presenten.

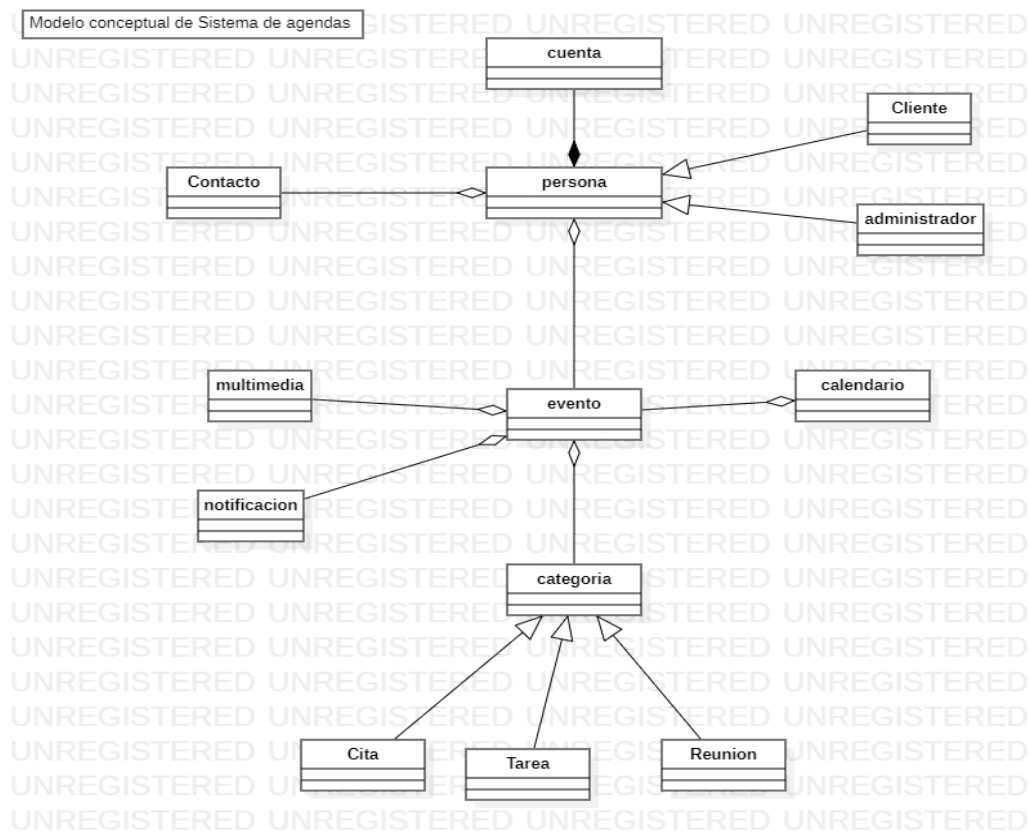


RNF006	Rendimiento	El sistema debe ser capaz de manejar un alto número de peticiones y eventos programados de forma simultánea y responder lo más veloz posible a las solicitudes del usuario (se debe tener en cuenta las características del equipo que usa el usuario, tener en cuenta cuán estable es la conectividad a internet, punto geográfico desde donde hace la petición el usuario y el saturamiento de la red en uso).
RNF007	Facilidad de Mantenimiento	La aplicación a desarrollar debe contar con la documentación detallada, y la simplificación de la arquitectura del sistema, haciendo al sistema fácil de mantener y actualizar.
RNF008	Accesibilidad	El sistema debe cumplir con los estándares de accesibilidad establecidos por la normativa WCAG 2.1 para garantizar que personas con discapacidades puedan acceder y utilizar el sistema de manera eficiente.
RNF009	Escalabilidad	El sistema debe tener la capacidad para manejar un posible aumento en el número de usuarios , accesos al sistema en diferentes puntos por el mismo usuario (el usuario ha accedido y ha hecho cambios en su cuenta por medio de diferentes dispositivos) y eventos planificados (eventos como : citas, actividad, tarea).
RNF010	Fiabilidad	El sistema debe operar de una forma consistente y predecible (esperar que haga para lo que fue programado) en base al software desarrollado y la implementación en el hardware adecuado manteniéndose por encima de las características mínimas que requiera el sistema.



En esta fase también desarrollamos un diagrama de dominio y un diagrama de clases en base a las ideas de cada uno de los compañeros de aula y con la guía de nuestro docente, a partir de los requerimientos analizados.

Diagrama de dominio

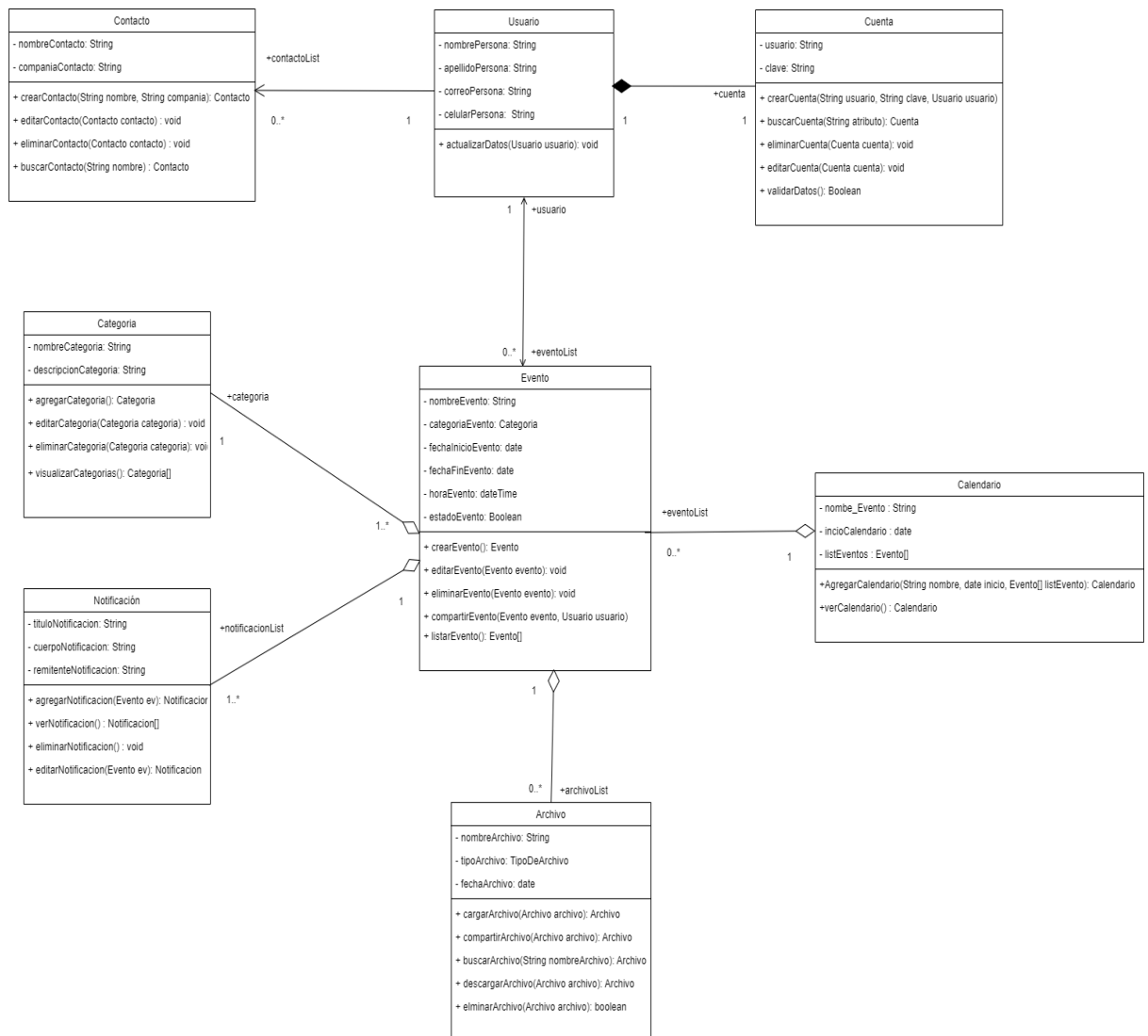




1859

Universidad
Nacional
de Loja

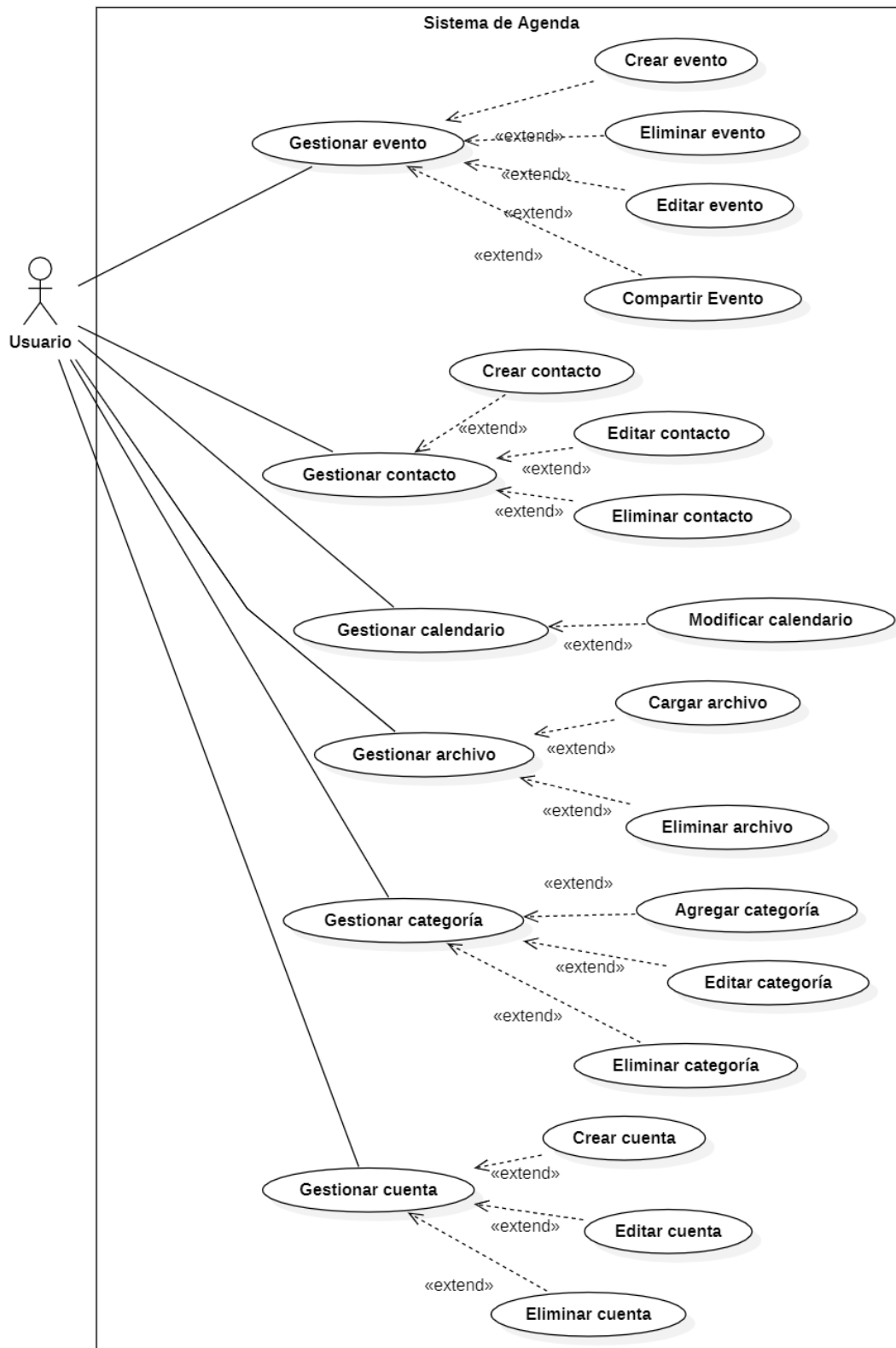
Diagrama de Clases



2. Análisis y diseño preliminar.

En esta fase se desarrolló un diagrama de casos de uso para así poder desarrollar las fichas de caso de uso, es decir la especificación de cada caso de uso.

Diagrama de Casos de Uso





Especificación de Casos de Uso

Caso de uso	CU01: Gestionar evento
Actor	Usuario
Descripción	El sistema permitirá al usuario poder crear, modificar, listar y borrar un evento en la agenda.
Flujo básico	<p>Crear</p> <ul style="list-style-type: none">• El usuario ingresara a la sección principal.• El sistema presentará el calendario de inicio (calendario creado en blanco) en conjunto con diversas funcionalidades en la parte izquierda de la sección principal (agregar contacto, agregar categoría, agregar evento, agregar cuenta, personalizar notificación, configuración).• El usuario seleccionara la opción de “agregar evento”.• El sistema presentará la sección para agregar evento, donde se tendrá un formulario (los siguientes datos a llenar: nombre, fecha, hora, descripción, tipo, estado).• El usuario ingresará los datos en todos los campos del formulario para registrar el evento de forma obligatoria.• El usuario seleccionara la opción de “agregar evento”• El sistema validará que todos los campos dentro del formulario estén completos• El sistema confirmará la creación del evento,



	<p>mediante un mensaje de aceptación.</p> <ul style="list-style-type: none">• El sistema creará el evento en base a la información proporcionada dentro del formulario
Flujo alternos	<p>Modificar</p> <ul style="list-style-type: none">• El usuario ingresara a la sección principal.• El sistema presentará el calendario de inicio.• El usuario seleccionará el evento a modificar.• El sistema presentará un despliegue de opciones para el evento seleccionado (modificar, eliminar, cambiar color, añadir invitados, listar)• El usuario seleccionara la opción de “modificar”• El sistema presentará una sección con los datos del evento para modificar.• El usuario realizará las modificaciones necesarias.• El usuario seleccionará la opción de “guardar nuevos cambios”• El sistema validará si se realizó cambios en el evento• El sistema validará que todos los campos del formularios estén completos (no dejar campos vacíos).• El sistema confirmará la modificación del evento, mediante un mensaje de aceptación.• El sistema modificará el evento en base a la información proporcionada dentro del formulario



	<p>Listar</p> <ul style="list-style-type: none">• El usuario ingresara a la sección principal.• El sistema presentará el calendario de inicio.• El usuario seleccionará el evento a modificar.• El sistema presentará un despliegue de opciones para el evento seleccionado (modificar, eliminar, cambiar color, añadir invitados, listar)• El usuario seleccionara la opción de “listar”.• El sistema presentará la sección donde se pueda listar los eventos por diferentes apartados (categoría, fecha, coincidencias).• El usuario seleccionará la opción de listar por “categoría”.• El usuario seleccionará la categoría por la que desea editar.• El sistema buscará coincidencias por categoría.• El sistema presentará las coincidencias encontradas por la categoría ingresada. <p>Eliminar</p> <ul style="list-style-type: none">• El usuario ingresara a la sección principal.• El sistema presentará el calendario de inicio.• El usuario seleccionará el evento a modificar.• El sistema presentará un despliegue de opciones para el evento seleccionado (modificar, eliminar, cambiar color, añadir invitados, listar por categoría)• El usuario seleccionara la opción de “eliminar”.• El sistema confirmará la
--	--



	<p>eliminación del evento, mediante un mensaje de aceptación.</p> <ul style="list-style-type: none"> • El sistema eliminará el evento en base al id propio del evento. • El sistema presentará la sección inicial con los cambios guardados.
Pre-condiciones (crear evento)	<ul style="list-style-type: none"> - El usuario debe iniciar sesión en el sistema. - El usuario debe tener creado un calendario, donde asignar los eventos - Presentar opción para crear el evento
Post-condiciones	<ul style="list-style-type: none"> - El sistema debe presentar el calendario creado con los diferentes eventos creados o espacios vacíos en caso de que no tenga creado en una fecha específica. - Buscar el evento creado por coincidencias (nombre, fecha, categoría) - Editar el evento creado en caso de ser necesario (cambiar categoría, fecha del evento, nombre del evento, invitados). - Eliminar el evento creado en caso de ser necesario (mala asignación de demasiados campos) - Crear nuevos eventos - Personalizar notificaciones

Caso de uso	CU02: Gestionar Calendario
Actor	Usuario
Descripción	El sistema permitirá al usuario modificar, visualizar y compartir calendario.



Flujo básico	<p>Modificar El usuario seleccionará un día o un intervalo de días. El sistema le presentará que desea añadir sea un evento, o una tarea. El usuario agrega el evento o tarea. El sistema guardará los datos.</p> <p>Visualizar El sistema presentará en pantalla el calendario, y los días que contengan cualquier tipo de actividad estarán marcados de un color verde, además cuando el usuario pase el cursor por encima de un día, se desplegará la lista de eventos y tareas que contenga ese día.</p> <p>Compartir El usuario presionará la opción compartir calendario. El sistema presentará la agenda personal con checks y búsqueda. El usuario seleccionará el/los contacto(s) que desee. El usuario seleccionará enviar calendario. El sistema presentará un diálogo de confirmación “Está seguro de compartir su calendario con (n) contactos?”. El usuario presionará en “Sí”. El sistema compartirá el calendario con los contactos del usuario seleccionados.</p>
Flujo alternos	Ninguno
Pre-condiciones	Para poder eliminar una cuenta es necesario que la cuenta esté previamente registrada en la base de datos.
Post-condiciones	Ninguno



Caso de uso	CU03: Gestionar Contacto
Actor	Usuario
Descripción	El sistema permitirá al usuario poder añadir, eliminar y modificar un contacto
Flujo básico	<p>1. Añadir</p> <ol style="list-style-type: none">El sistema le presenta al usuario una opción de añadir contacto .El sistema le presenta al usuario un formulario para que llene con la información pertinente del contacto.<ul style="list-style-type: none">■ Nombres.■ Apellidos■ Celular■ Correo Electrónico■ CompañíaEl usuario confirma la creación del contacto.El sistema valida que todos los campos del formulario estén llenos. <p>2. Eliminar</p> <ol style="list-style-type: none">El sistema le permite al usuario eliminar un contacto.El usuario ingresa a la ventana de editar contacto, donde existirá la opción de eliminar contacto.El sistema le muestra una ventana de confirmación de la eliminación del contacto.El usuario puede confirmar o cancelar la eliminación del contacto. <p>3. Modificar</p> <ol style="list-style-type: none">El usuario ingresa a la ventana de editar contacto.El sistema le mostrará el formulario con las



	<p>opciones que tiene permitido editar como:</p> <ul style="list-style-type: none"> ■ Nombres. ■ Apellidos ■ Celular ■ Correo Electrónico ■ Compañía <p>c. El usuario guarda los cambios</p>
Flujo alternos	<p>Campos incompletos</p> <ol style="list-style-type: none"> 1. El usuario ingresa a la sección de añadir contacto. 2. El usuario llena alguno o ninguno de los campos de información del contacto. 3. El sistema presentará una ventana emergente para alertar que todos los campos no están completos.
Pre-condiciones	Para poder añadir, eliminar o modificar un contacto el usuario tiene que estar previamente registrado en el sistema.
Post-condiciones	Actualizar los datos del sistema

Caso de uso	CU04: Gestionar cuenta
Actor	Usuario
Descripción	El sistema permitirá al usuario poder crear, editar y eliminar su cuenta de usuario.
Flujo básico	<p>Crear cuenta</p> <ul style="list-style-type: none"> ● El usuario elige la opción [Registrarse]. ● El sistema presenta la interfaz [Registrar cuenta de usuario]. ● El usuario ingresa sus datos personales. <ul style="list-style-type: none"> ○ Nombres ○ Apellidos ○ Dirección ○ Teléfono ○ Correo electrónico



	<ul style="list-style-type: none">○ Clave● El usuario selecciona la opción [Registrarse].● El sistema verifica que se hayan ingresado datos válidos en los campos de texto.● El sistema guarda datos en la entidad [Usuario] (el correo electrónico y clave serán las credenciales de ingreso).● El sistema crea una cuenta de usuario en la entidad [Usuario] de la base de datos.● El sistema refresca la página.● Finaliza el use case.
Flujo alternos	<p>Editar cuenta</p> <ul style="list-style-type: none">● El usuario selecciona la opción [Editar cuenta de usuario].● El sistema presenta la vista [Editar cuenta].● El sistema habilita la edición de los datos, excepto el correo electrónico.● El usuario modifica sus datos personales.<ul style="list-style-type: none">○ Nombres○ Apellidos○ Dirección○ Teléfono○ Clave● El usuario selecciona la opción [Guardar cambios].● El sistema actualiza los datos en la entidad [Usuario].● Finaliza el use case. <p>Eliminar cuenta</p> <ul style="list-style-type: none">● El usuario selecciona la opción [Eliminar cuenta de usuario].● El sistema presentará la interfaz [Eliminar cuenta].



	<ul style="list-style-type: none"> • El usuario seleccionara la opción [Eliminar]. • El sistema presentará un mensaje de confirmación diciendo “¿<i>Está seguro de que desea eliminar su cuenta de usuario?</i>” • El usuario selecciona la opción [Si, estoy seguro]. • El sistema desactiva la cuenta de usuario en la base de datos. • Finaliza el use case.
Pre-condiciones (crear evento)	<ul style="list-style-type: none"> - El usuario debe iniciar sesión en el sistema.
Post-condiciones	<ul style="list-style-type: none"> - Registrar datos personales del usuario en el sistema. - Creación de una cuenta de usuario. - Modificar datos personales. - Dar de baja su cuenta de usuario. - Perder el acceso al sistema de agenda.

3. Fase Final: Diseño e implementación

En nuestro caso lo más cercano a los diagrama que se implementan en esta sección es la ruta crítica que se desarrolló en la fase de planificación ya que en la ruta crítica se describen varias actividades que se desarrollan en el proyecto y en la ruta crítica podemos visualizar el flujo que puede tener nuestro sistema.



Guía de la fase de Monitoreo

4. Monitoreo

a. Cómo se controlan los avances de un proyecto de software

Para controlar los avances de un proyecto de software, es necesario establecer un conjunto de herramientas y técnicas para medir el progreso y la calidad del trabajo realizado. A continuación, se detallan algunos métodos comunes para controlar los avances de un proyecto de software:

- **Planificación del proyecto:** se establece un plan detallado para el proyecto que incluye objetivos, entregables, hitos y un calendario de trabajo.
- **Gestión del alcance:** se establecen límites claros sobre el alcance del proyecto y se identifican los requisitos que se deben cumplir.
- **Gestión de riesgos:** se identifican y se evalúan los riesgos asociados al proyecto y se establecen planes de mitigación para minimizar su impacto.
- **Seguimiento y control:** se establecen métricas para medir el progreso del proyecto y se realiza un seguimiento regular de las actividades para asegurarse de que se están cumpliendo los objetivos y los plazos.
- **Comunicación y colaboración:** se fomenta una comunicación y colaboración efectiva entre los miembros del equipo y los stakeholders del proyecto para asegurar que todos estén informados y comprometidos con el progreso del proyecto.
- **Uso de herramientas y software:** existen herramientas y software especializados que permiten el seguimiento y la gestión de proyectos de software, como Jira, Trello, Asana, entre otros.

En general, la gestión de proyectos de software requiere un enfoque estructurado y organizado para asegurar que el proyecto se realice dentro de los plazos establecidos y con la calidad requerida. La aplicación de técnicas y herramientas para el control de avances es fundamental para asegurar el éxito del proyecto.

b. Cómo se controlan los gastos del proyecto

El control de gastos en un proyecto de software es una parte crítica de la gestión del proyecto. Para lograr un control efectivo de los gastos, se pueden seguir los siguientes pasos:

1. **Establecer un presupuesto:** Es necesario establecer un presupuesto detallado para el proyecto desde el principio. Esto ayuda a tener una idea clara de cuánto se puede gastar y en qué áreas se pueden hacer recortes.
2. **Identificar y estimar los costos:** Identificar y estimar los costos de todos los elementos del proyecto, como hardware, software, personal, licencias, infraestructura, etc.
3. **Monitorear los costos:** Monitorear los costos durante todo el proyecto, comparando los gastos reales con los presupuestados. Se



pueden utilizar herramientas como hojas de cálculo o software de gestión de proyectos para hacer un seguimiento preciso de los gastos.

4. **Controlar los cambios en el alcance del proyecto:** Controlar los cambios en el alcance del proyecto para asegurarse de que cualquier aumento en los costos esté justificado y esté dentro del presupuesto establecido.
5. **Establecer un proceso de aprobación:** Establecer un proceso de aprobación para todos los gastos del proyecto. Esto ayuda a garantizar que todos los gastos sean necesarios y estén dentro del presupuesto.
6. **Realizar informes de gastos:** Realizar informes de gastos periódicos para informar al equipo de gestión del proyecto sobre el estado del presupuesto y cualquier desviación significativa.
7. **Ajustar el presupuesto:** Si se produce una desviación significativa del presupuesto, se debe considerar ajustarlo. En algunos casos, puede ser necesario reducir el alcance del proyecto o buscar nuevas fuentes de financiamiento.