Ingeniería del Software Grado en Ingeniería Informática

Universidad de Zaragoza, curso 2023/2024

Práctica 1: Requisitos y análisis

1. Introducción y objetivos

A lo largo de las prácticas de la asignatura se va a abordar el análisis, diseño, implementación y pruebas de una aplicación móvil que permita que el propietario o propietaria de una tienda de preparación de comidas pueda gestionar los pedidos que le solicitan sus clientes por teléfono.

El primer objetivo de esta sesión de prácticas es realizar la captura de los requisitos de la aplicación y su representación mediante un catálogo de requisitos y la definición de casos de uso asociados a esos requisitos. Otro objetivo de esta sesión de prácticas es realizar el análisis de la aplicación, estableciendo el modelado estático mediante diagramas de clase y el modelado dinámico mediante diagramas de secuencia. Los resultados de estas fases de requisitos y análisis formarán parte de la documentación de análisis, diseño y pruebas que se irá elaborando durante las prácticas de la asignatura. En el apéndice A se incluye una guía para elaborar esta documentación.

Para abordar estas fases de definición de requisitos y análisis os apoyaréis en el uso de la herramienta CASE Visual Paradigm, introducida en la sección 2. La sección 3 describe las actividades a realizar en esta práctica. Por último, la sección 4 describe la entrega requerida para esta sesión de prácticas.

2. La herramienta CASE Visual Paradigm

CASE, del inglés Computer Aided Software Engineering, significa Ingeniería de Software asistida por ordenador. Una herramienta CASE es un programa software que proporciona una ayuda, más o menos automática, en cualquiera de las etapas del proceso de desarrollo del software:

Construcción de modelos de Análisis y Diseño.



- Verificación y corrección de modelos.
- Prototipado de la interfaz de usuario.
- Utilidades para almacenar y obtener informes, así como consultar información de desarrollo.
- Generación de código.

Por tanto, el objetivo primordial de las herramientas CASE es ayudar al ingeniero de software automatizando en la medida de lo posible su trabajo. Ello se materializa al menos en:

- Automatizar etapas del ciclo de vida: Requisitos, Análisis, Diseño, Codificación, Pruebas.
- Proporcionar un acceso eficaz y consistente a la información que se va generando en esas etapas.
- Crear y mantener de forma gráfica las especificaciones del sistema.
- Controlar y analizar automáticamente la calidad de la información.
- Construir maquetas, prototipos o simulaciones del sistema.
- Generar automáticamente código y/o documentación.
- Llevar a cabo ingeniería inversa en sistemas ya desarrollados y difíciles de mantener.

En esta asignatura vas a hacer uso de la herramienta CASE Visual Paradigm, que facilita las tareas del análisis y diseño de proyectos software. Esta herramienta también permite la planificación y gestión de proyectos, aunque este ámbito está fuera del alcance de esta asignatura.

El Departamento de Informática e Ingeniería de Sistemas dispone de una licencia Standard del software "Visual Paradigm" que se puede utilizar tanto en los laboratorios de docencia como en los ordenadores personales de los alumnos. Aparte de la funcionalidad básica de edición de diagramas de análisis y diseño conformes con la notación de UML, la licencia Standard permite también utilizar herramientas de diseño de experiencia de usuario (wireframes y wireflows) o modelado de bases de datos. Se puede encontrar más información sobre todo lo que incluye la licencia Standard en https://www.visual-paradigm.com/editions/.

Esta herramienta funciona sobre Java, lo que facilita que sea multiplataforma. Aunque en los laboratorios del Edificio Ada Byron está disponible la versión 17.1 de herramienta Visual Paradigm arrancando tanto con en el sistema operativo Windows como con el sistema operativo Cent OS,¹ en el enlace https://ap.visual-paradigm.com/

¹Se debe inciar sesión con el usuario guest y la contraseña diis7d5

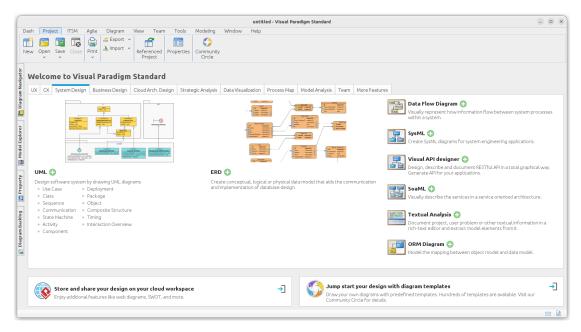


Figura 1: Ventana principal de la herramienta CASE Visual Paradigm.

university-of-zaragoza2 puedes descargar, entre otros, los siguientes recursos:

1. Instaladores del producto Visual Paradigm para distintos sistemas operativos. Si en macOS os aparece un mensaje de que el fichero de instalación (o al ejecutar la app) está dañado, ejecutad estos comandos previamente en la terminal:

```
sudo spctl --master-disable
xattr -cr /Applications/Visual\ Paradigm.app
```

2. El código de activación de la licencia. Para la activación podéis seguir la ayuda en https://circle.visual-paradigm.com/docs/installing-visual-paradigm/activating-de-activating-visual-paradigm/activating-visual-paradigm-with-academic-license/, aunque es relativamente intuitivo. Además del código de activación, pide un nombre y un correo electrónico. El correo electrónico ha de ser corporativo (@unizar.es). Envían un código de confirmación al correo para completar la activación.

La herramienta Visual Paradigm se organiza en base a proyectos. En la Figura 1 se puede ver la pantalla principal de la herramienta cuando se inicia por primera vez o se crea un proyecto nuevo. Un proyecto (fichero con extensión .vpp) contendrá todos los diagramas y elementos que se hayan definido dentro de él. Este fichero es el que puede compartirse entre los participantes de un equipo de desarrollo. [Importante! Visual Paradigm incorpora una política de actualizaciones peculiar: cuando un proyecto de Visual Paradigm desarrollado con una versión de la herramienta se abre con una

versión más actual, el proyecto se actualiza, dejando de poder abrirse con la versión vieja. Esto implica que si se trabaja en equipo, todos los participantes del mismo han de disponer de la misma versión de Visual Paradigm.

Por defecto, en la parte izquierda de la pantalla principal tienes acceso a tres paneles diferentes:

- Diagram Navigator es el panel al que tenemos que acceder para crear diferentes diagramas en nuestro proyecto, o bien para acceder a los diagramas que se hayan creado.
- Model Explorer es el panel que nos da acceso a los elementos del modelo de nuestro proyecto. Aquí nos encontraremos una vista en forma de árbol de todos los elementos que se encuentren definidos dentro de nuestro modelo. Conforme vayamos creando diagramas en nuestro proyecto, el contenido de este panel irá también aumentando.
- Diagram Backlog es un panel que se usa para la creación de elementos del modelo desde un análisis textual. No vamos a hacer uso de esta parte de la herramienta.

Desde el menú de View Panes pueden accederse a otros paneles como Class Repository (muestra una lista con todas las clases que se encuentran definidas en nuestro proyecto) o Property (seleccionado un elemento o un diagrama, permite definir ciertas propiedades como el nombre, color, fuente y tamaño, etc), entre otros. Puedes consultar el capítulo 5 de la guía de usuario [2] para conocer más detalles acerca de cómo crear distintos tipos de diagramas UML.

3. Actividades a realizar

3.1. Definición de un catálogo de requisitos

Se desea construir una aplicación que debe funcionar en dispositivos móviles con sistemas operativos Android para que el propietario o propietaria de una tienda de preparación de comidas pueda gestionar los pedidos que le comunican informalmente sus clientes. La aplicación exigida para este trabajo deberá permitir lo siguiente:

- La creación de platos. Cada plato consta de un nombre, una descripción en texto libre de los ingredientes, la categoría del plato (a elegir entre PRIMERO, SEGUN-DO, o POSTRE) y un precio en euros por ración.
- La consulta de un listado de los platos previamente creados, pudiéndolos ordenar por categoría, nombre o ambos criterios.
- La modificación de los platos previamente creados.

- La eliminación de los platos previamente creados.
- La creación de un pedido. El pedido constará de un nombre de cliente, el número móvil del cliente, la fecha y hora a la que se recogerá el pedido, y una selección del número de raciones que se necesitan de algunos de los platos previamente creados. Para la fecha y hora de recogida se debe tener en cuenta que la fecha y hora deben ser posteriores a la fecha y hora actual y que la tienda de preparación de comidas solo permite recogida de pedidos de Martes a Domingo entre las 19:30 y las 23:00. Además, los pedidos tienen un estado asociado que inicialmente será SOLICITADO por defecto, pero que puede cambiar a PREPARADO o RECOGIDO.
- La consulta de un listado de los pedidos previamente creados, pudiéndolos ordenar por nombre de cliente, número de móvil, o fecha y hora de recogida. También se podrá filtrar el listado para visualizar solo pedidos con un estado concreto (SOLI-CITADO, PREPARADO o RECOGIDO).
- La modificación de los pedidos previamente creados.
- El cálculo del precio total del pedido. El precio total se debería calcular de forma automática en la creación del pedido, o en la modificación del pedido si se realiza una selección distinta de platos o número de raciones. Por respecto a los clientes, el precio total se debería mantener aunque se modifique el precio por ración después de la última actualización de platos y raciones en el pedido.
- El envío al móvil del cliente de la información del pedido (incluido precio).²
- La eliminación de los pedidos previamente creados.

Por último, se debe tener en cuenta que no se espera que haya más de 100 platos y 2000 pedidos almacenados en la aplicación.

La primera tarea que debes realizar en esta sesión de prácticas es la elaboración de un catálogo de requisitos que incluya la descripción de los requisitos funcionales, los requisitos no funcionales y las restricciones de esta aplicación.

La notación utilizada para este catálogo será la presentada para las tablas de requisitos en las clases de teoría. Este catálogo de requisitos formará parte de la documentación de análisis, diseño y pruebas que se irá elaborando durante las prácticas de la asignatura. Se recuerda que en el apéndice A se incluye una guía para elaborar esta documentación. Se recomienda además que este documento se comparta entre los miembros del equipo dentro de una plataforma como Google Drive o One Drive que permita la edición colaborativa.

²Para abordar este requisito se deberá utilizar el patrón de diseño propuesto en la *Práctica 5*.

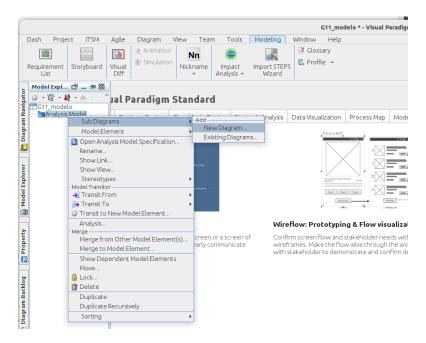


Figura 2: Creación de un nuevo diagrama UML dentro del modelo de análisis.

3.2. Definición de casos de uso

Partiendo del catálogo de requisitos de la aplicación de gestión de pedidos para la tienda de preparación de comidas, construye el diagrama de casos de uso que representa los requisitos de dicha aplicación.

En primer lugar, una vez que hayas abierto la herramienta Visual Paradigm, deberás dar un nombre a tu proyecto la primera vez que lo guardes con la opción de menú Project Save. Para homogeneizar el nombre del proyecto de cada equipo este debería ser «<código equipo>_modelo» donde el código se corresponde con el número de equipo asignado, por ejemplo M11 (equipo 1 dentro del grupo de prácticas 1 del turno de mañana) o T201 (equipo 1 dentro del grupo de prácticas 20 del turno de tarde).

A continuación también es recomendable acceder al panel *Model Explorer* y crear con el menú contextual un modelo de análisis con la opción de menú Model Analysis Model. Así podrás ir creando y localizando facilmente todos los diagramas y elementos del modelo correspondientes al análisis durante esta sesión de prácticas.

Para crear un diagrama UML de casos de uso dentro del modelo de análisis puedes utilizar la opción del menú contextual Sub Diagrams New Diagram, y posteriormente seleccionar Use Case Diagram como tipo de diagrama (véase Figura 2). Alternativamente, también puede utilizar el panel de Diagram Navigator y en el apartado de UML Diagrams Use Case Diagrams hacer clic derecho sobre él y seleccionar la opción New Use Case Diagram (puedes hacerlo también desde el menú Diagram New), como se muestra en la Figura 3.

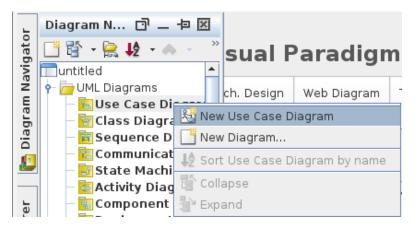


Figura 3: Creación de nuevo diagrama UML de casos de uso.

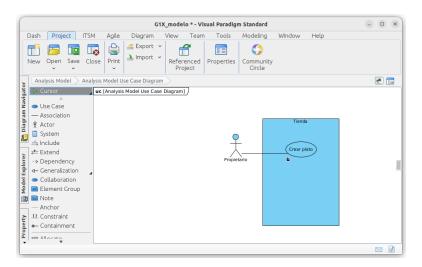


Figura 4: Elementos disponibles para diagramas UML de casos de uso.

Independientemente de la opción elegida, se creará un nuevo diagrama, que aparecerá en la parte derecha de la aplicación.

En la parte izquierda de la aplicación aparecerá ahora todos los elementos que podemos insertar en el diagrama, según el estándar UML (véase la Figura 4). Aprenderás el significado de estos elementos en mayor profundidad en las clases de teoría de la asignatura. Para añadir elementos como casos de uso o actores, simplemente podemos arrastrar desde el panel izquierdo y soltar en el panel derecho, o también seleccionar el elemento deseado en el panel y hacer clic sobre el diagrama. Para añadir asociaciones, tras seleccionarla en el panel izquierdo hay que posteriormente seleccionar ambos extremos de la asociación en el diagrama. Algunas veces es probable que en la parte inferior de algunos elementos te aparezca un icono como el de la Figura 5 (una especie de cubo verde). Este icono permite ajustar el tamaño de un elemento al tamaño de aquellos elementos que



Figura 5: Ajuste automático del tamaño de elementos.

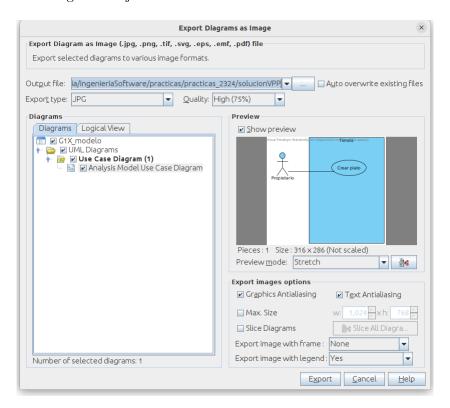


Figura 6: Exportación de diagramas UML como imagen.

tenga dentro. Por ejemplo, en el caso del elemento ovalado correspondiente a un caso de uso, éste se extendería hasta que el nombre del caso de uso cupiera en una única línea de texto.

Puedes incorporar una imagen del diagrama de casos de uso que acabas de crear dentro del documento de análisis, diseño y pruebas. Visual Paradigm permite exportar todos los diagramas que realicemos en formato de imagen. Esto se consigue mediante el menú Project Export Diagrams as Image... o haciendo clic derecho en el panel de un diagrama y seleccionando la opción Export Export as Image... (véase la Figura 6). Verás que Visual Paradigm añade un pequeño texto con el nombre y el logotipo de la herramienta a todas las figuras que se exporten. Si deseas eliminarlo, puedes elegir entre los formatos de exportación algún formato que sea editable como SVG o PDF.

A continuación, incorpora en tu documento la descripción de cada caso de uso es-

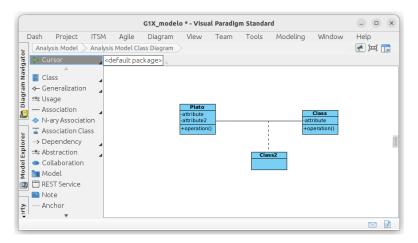


Figura 7: Construcción de diagramas de clases.

pecificando los escenarios principales y alternativos de los casos de uso mediante flujos de eventos. Estos flujos de eventos también se podrían documentar en Visual Paradigm seleccionando un caso de uso en un diagrama y utilizando la opción del menú contextual Open Use Case Details Flow of Events. Sin embargo, para no tener que ir copiando y pegando los flujos de eventos entre Visual Paradigm y el documento, resulta más práctico definirlos directamente en el documento.

3.3. Modelado estático de la aplicación

Teniendo en cuenta los requisitos y su representación mediante casos de uso, construye un modelo estático a nivel de análisis mediante un diagrama de clases que refleje el dominio del problema sin preocuparte de los elementos de la interfaz de usuario, que se abordarán en prácticas posteriores. Lo importante es que este modelo de clases permita representar conceptualmente la información necesaria para gestionar los pedidos de la tienda de preparación de comidas con los atributos que sean más adecuados y las relaciones entre clases que necesitas.

Para crear un diagrama de clases dentro del modelo de análisis, puedes utilizar el mismo procedimiento que en la actividad anterior. En el panel *Model Explorer* puedes seleccionar el modelo de análisis y utilizar la opción del menú contextual Sub Diagrams New Diagram. Posteriormente, selecciona *Class Diagram* como tipo de diagrama (véase Figura 7).

Si no tienes claro cómo debería ser el diagrama de clases, puedes construir como paso previo un diagrama de objetos para entender mejor el contexto de la aplicación (véase Figura 8). Este procedimiento es el que se siguió en la clase de problemas para el modelado de objetos de una aplicación que pretende gestionar los vuelos disponibles de una compañía aérea y que está disponible en el Anillo Digital Docente (ADD) de la asignatura

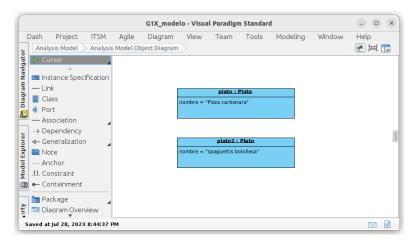


Figura 8: Construcción de diagramas de objetos.

(ejercicio de modelado de objetos).

Recuerda también que debes añadir las operaciones (métodos) que consideres necesarias para dar soporte a los casos de uso. No olvides que puedes definir operaciones de clase (métodos estáticos) para devolver arrays de objetos que den soporte a las funcionalidades relacionadas con ofrecer listados de objetos.

3.4. Modelado dinámico de la aplicación

Teniendo en cuenta los casos de uso y el diagrama de clases que se ha realizado previamente, construye ahora los diagramas de secuencia que permiten describir los flujos de eventos asociados a cada caso de uso. Ten en cuenta que para construir estos diagramas de secuencia se deben utilizar exclusivamente los actores y las clases definidas en las actividades previas. Sin embargo, es posible que necesites incorporar nuevas operaciones en el modelo de clases para representar los flujos de eventos de manera apropiada.

Puedes crear un diagrama UML de secuencia de forma independiente utilizando el panel Diagram Navigator, pero dado que vamos a crear un diagrama UML de secuencia que está vinculado a un caso de uso, podemos marcar esa referencia implícita en nuestro modelo creando el diagrama de secuencia mediante el clic derecho sobre un caso de uso y seleccionando Sub Diagrams New Diagram... (véase la Figura 9 donde se selecciona el caso de uso "Crear plato" para vincular un diagrama de secuencia). En la ventana para la creación de nuevo diagrama, selecciona Sequence Diagram. También se puede crear de manera independiente y asociarlo al caso de uso posteriormente, con la opción Sub Diagrams Existing Diagrams....

Verás que Visual Paradigm incorpora multitud de diagramas de ejemplo, pero en nues-

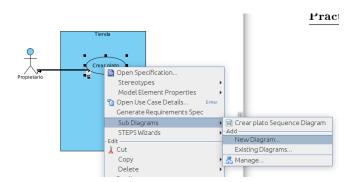


Figura 9: Creación de subdiagrama.

tro caso selecciona la opción *Blank* y dale el nombre que desees (te recomendamos que visualices los diagramas de ejemplo para que veas cómo son los diagramas UML que puedes obtener con Visual Paradigm). De nuevo, el panel de la izquierda cambia para mostrar los elementos que se pueden dibujar en un diagrama de secuencia.

Si ahora vas al panel *Model Explorer*, verás que aparecen todos los elementos que hemos insertado en todos los diagramas hasta el momento. Puedes seleccionar un actor y arrastrarlo hasta la parte derecha. Verás que se crea automáticamente una línea de vida asociada a este actor. Esta es la ventaja de las herramientas CASE: una vez que definimos los elementos de nuestro modelo, hemos de usar siempre estos elementos para garantizar que se guarda la coherencia entre todos los diagramas del sistema. De este modo, si se produce una actualización en los elementos (e.g., se cambia el nombre, se elimina), automáticamente se refleja la actualización en todos los diagramas donde se use el elemento actualizado.

Al igual que has arrastracto actores al diagrama de casos de uso, también podemos arrastrar una clase de nuestro modelo para poder tener una línea de vida de un objeto de esta clase en nuestro diagrama de secuencia (véase un ejemplo en la Figura 10). La ventaja de definir métodos en una clase, junto con sus parámetros y tipo de dato devuelto, es que permite seleccionar cuál es el método a invocar del objeto cuando se envían mensajes en el diagrama de secuencia.

4. Entrega de la práctica

Respecto a los **entregables exigidos para esta práctica**, debes enviar los siguientes elementos a través de una tarea accesible en el ADD:

 El documento en formato PDF correspondiente al análisis, diseño y pruebas de la aplicación con la estructura correspondiente a la guía disponible en el apéndice, y donde de momento solo completarás la parte correspondiente a las actividades

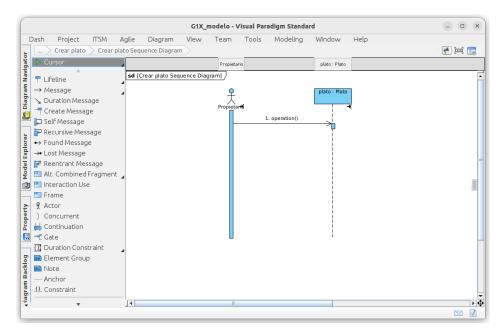


Figura 10: Ejemplo de diagrama UML de secuencia.

solicitadas en esta práctica:

- catálogo de requisitos;
- casos de uso; y
- modelado estático y dinámico a nivel de análisis.

El nombrado del archivo PDF seguirá el patrón «<código equipo>_documentacion.pdf», donde código equipo se corresponde con el número de equipo asignado, por ejemplo G11 (equipo 1 dentro del grupo de prácticas 1 del turno de mañana) o T201 (equipo 1 dentro del grupo de prácticas 20 del turno de tarde).

■ Un fichero denominado «<código equipo>_modelo.vpp» con la exportación del proyecto de Visual Paradigm. Se puede usar la opción de menú Project Export > Visual Paradigm Project seleccionando todos los elementos del modelo y todos los diagramas, o simplemente copiar el archivo con extensión «.vpp» de la carpeta que se ha utilizado para guardar el proyecto de Visual Paradigm.

La fecha límite para subir este documento en el ADD será el día anterior a la siguiente sesión de prácticas de cada equipo. Durante la sesión de la siguiente práctica (*Práctica* 2) se presentará al profesor el resultado de esta práctica. Recuerda que para superar la asignatura hay que presentar todas las prácticas.

A. Guía de la documentación a entregar

El documento donde se describa el análisis, diseño y pruebas realizadas para la aplicación exigida a lo largo de las prácticas de la asignatura deberá incluir, al menos, los siguientes apartados:³

- Resumen. Síntesis muy breve de todo el trabajo realizado: en qué ha consistido, cómo se ha realizado, por qué se ha realizado de cierta manera y qué objetivos se pretendían alcanzar.
- Índice.
- Introducción y objetivos. Descripción breve del contexto y de lo que se pretende conseguir a través de este documento.
- Requisitos. Como parte de la documentación de requisitos, se deberá aportar lo siguiente:
 - Catálogo de requisitos. Se incluirá la descripción de los requisitos funcionales, no funcionales y restricciones sobre el sistema a desarrollar que se hayan tenido en consideración.
 - Diagramas de casos de uso y descripción textual de los mismos a través de flujos de eventos.
- Análisis. Dentro de la documentación de análisis, se deberán incluir los siguientes elementos:
 - Modelos estáticos utilizados, incluyendo al menos el diagrama de clases correspondiente al análisis.
 - Modelos dinámicos utilizados, incluyendo al menos diagramas de secuencia (correspondientes al análisis) que describan los casos de uso.
 - Descripción de la interfaz de usuario, incluyendo prototipos de pantallas y el mapa de navegación.
- **Diseño del sistema.** Dentro de la documentación del diseño del sistema, se deberán describir los siguientes elementos:
 - Descomposición en subsistemas mediante la utilización de diagramas de paquetes.
 - Distribución en componentes y asignación a nodos de procesamiento mediante diagramas de componentes y de despliegue.

³Esta guía es una adaptación de las recomendaciones elaboradas por Villarrolla Gaudó et al [1] para elaborar una memoria técnica .

- Diseño de objetos. Dentro de la documentación del diseño de objetos, se incluirán los siguientes elementos:
 - Descripción de decisiones de diseño adoptadas como por ejemplo, selección de patrones de diseño, utilización de bibliotecas especiales, etc.
 - Diagramas correspondientes a los modelos estáticos y dinámicos con un nivel de detalle adecuado al nivel de diseño de objetos. Al menos, debe incluirse el diagrama de clases de diseño y diagramas de secuencia de diseño.
 - Modelo lógico de la base de datos relacional que se ha diseñado para dar soporte a los requisitos de la aplicación.
- Pruebas. Descripción de las pruebas unitarias y pruebas de sistema que se han realizado. Esta descripción deberá incluir tanto los casos de prueba planteados utilizando técnicas de diseño de pruebas, como la descripción de la implementación de esos casos de prueba y los resultados de su ejecución.
- Resultados y conclusiones. Breve presentación y explicación de los resultados obtenidos y un análisis de los mismos, remarcando aspectos como puntos fuertes y débiles de los resultados obtenidos, y señalando las posibilidades de mejora.
- Bibliografía. Se incluirá al final de la memoria un listado con las referencias bibliográficas y de otro tipo que hayan sido utilizadas. Estas referencias incluirán también algún tipo de identificador (por ejemplo, numérico) que permita citar estas referencias desde alguno de los apartados anteriores.

Referencias

- [1] María Villarroya Gaudó, Jesús Alastruey Benedé, Javier Resano Ezcaray, Enrique Torres Moreno, and Darío Suárez Gracia. Cómo redactar una memoria técnica. Technical report, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, https://webdiis.unizar.es/~silarri/coordinadorGrado/DOCS/RECOMENDACIONES/Ayuda_elaboracion_memoria_tecnica-proyHWsistDistrib.pdf, 2017.
- [2] Visual Paradigm Community Circle. Visual Paradigm User's Guide. https://circle.visual-paradigm.com/docs/, 2023.