Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Curso Inteligencia Artificial aplicada a la Ingeniería de Software

Curso 2025-26

Proyecto Final

Sudoku

ENTREGA: **1 - FINAL**

EQUIPO: Equipo 6

ALUMNOS: Paula Alcarás / Bastián González / Hugo Serón / Fernanda Vásquez / Benjamín Zúñiga

FECHA: **xx/08/2025**

PÁGINA DE CONTROL DE DOCUMENTO

Registro de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Autor | Cambios en la versión |
| 0 | 21/08/2025 | Paula Alcarás | Plantilla inicial |
| 1 | 25/08/2025 | Benjamín Zúñiga | Código Inicial y relleno capitulo 1 y 2 |
| 2 | 25/08/2025 | Bastián González  Hugo Serón | Redacción de Requerimientos |
| 3 | 25/08/2025 | Paula Alcarás | Redacción Capitulo 1 y 2 |
| 4 | 26/08/2025 | Benjamín Zuñiga | Relleno capítulo 2 |
| 5 | 27/08/2025 | Paula Alcarás | Relleno capítulo 2 |
| 6 | 27/08/2025 | Bastián González  Hugo Serón | Finalización de redacción de requerimientos |
| 7 | 27/08/2025 | Paula Alcarás  Bastián González  Hugo Serón | Redacción capítulo 3 y finalización del capítulo 3 |
| 8 | 28/08/2025 | Benjamin zuñia | Relleno capitulo 2 (estado del arte)+ |
|  |  |  |  |
|  |  |  |  |

Horas invertidas en el proyecto

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE | INDIVIDUAL | EQUIPO | TOTAL |
| Paula Alcarás | 6 | 4 |  |
| Bastián González | 6 | 4 |  |
| Hugo Serón | 6 | 4 |  |
| Fernanda Vásquez | 1 | 4 |  |
| Benjamín Zúñiga | 5 | 4 |  |
| TOTAL |  |  |  |

Contribución a la redacción del documento

|  |  |
| --- | --- |
| NOMBRE | Descripción de sus principales contribuciones a la redacción del documento |
| Paula Alcarás | Relleno página control y redacción capítulos 1,2 y 3 |
| Bastián González | Redacción requisitos y redacción de capítulo 3 |
| Hugo Serón | Redacción requisitos y redacción de capítulo 3 |
| Fernanda Vásquez |  |
| Benjamín Zúñiga |  |

TABLA DE CONTENIDO

1 INTRODUCCIÓN Y OBJETIVOS GENERALES 5

1.1 Objetivos generales del proyecto 5

1.2 Visión general del documento 5

2 DESCRIPCIÓN DEL PRODUCTO 6

2.1 Descripción del producto a desarrollar 6

2.2 Situación actual y estado del arte 6

2.3 Alcance y Funcionalidades generales 6

2.4 Restricciones y elementos fuera del alcance 6

2.5 Diagrama de Contexto 6

3 REQUISTOS 9

3.1 Clasificación de los requisitos 9

3.2 Plantilla para la redacción de requisitos 9

3.3 Requisitos funcionales 10

3.4 Requisitos no funcionales 10

3.5 Vocabulario del dominio 10

4 ALGORITMO GENÉTICO 11

4.1 Población / individuo / cromosoma / gen 11

4.2 Creación de población 11

4.3 Algoritmo de selección 11

4.4 Algoritmos de cruzamiento 11

4.5 Algoritmo de mutación 11

4.6 Algoritmo de reemplazo 11

5 MODELADO CONCEPTUAL 12

5.1 Diagrama de Casos de Uso 12

5.2 Diagramas de Clases 13

6 ARQUITECTURA 15

6.1 Modelo de implementación 15

6.2 Diagrama de componentes 16

7 LISTAS Y TABLAS 18

7.1 Consistencia entre requisitos: conflictos, redundancias, acoplamientos 18

7.2 Trazabilidad requisitos – modelo conceptual (clases) 18

7.3 Trazabilidad requisitos – modelo de implementación (componentes) 18

8 ANEXOS 19

8.1 Anexo I – Bibliografía 19

8.2 Anexo II – Uso de IA en el proyecto 19

CONSIDERACIONES SOBRE LAS ENTREGAS

* + La **primera entrega** incluye los capítulos 1, 2, 3 y 4
  + La **entrega final** incluye el resto de capítulos y revisión de capítulos previos

Eliminar esta página del trabajo final

# INTRODUCCIÓN Y OBJETIVOS GENERALES

Este informe detalla la creación del proyecto Sudokus Web, que tiene como finalidad concebir y construir una plataforma web interactiva donde los usuarios puedan disfrutar de una versión diferente del Sudoku de una forma ágil, sencilla y a su alcance.

## Objetivos generales del proyecto

* Desarrollar una página web funcional que genere y resuelva Sudokus de diferentes tamaños y de Multidokus.
* Proporcionar una interfaz de usuario clara, atractiva y fácil de usar.
* Elaborar la documentación técnica y metodológica que respalde el desarrollo del sistema.

## Visión general del documento

El presente documento se conforma de diferentes capítulos, los cuales abordan cada uno de los aspectos específico del proyecto “Sudoku Web”, desde la definición inicial hasta la entrega final del sistema implementado.

En el capítulo 1, se presenta el contexto del proyecto, su importancia, los objetivos planteados y la justificación que respalda la elección del tema. En el capítulo 2 se expone los fundamentos conceptuales relacionados con el Sudoku y sus variantes, además de los principios de desarrollo de aplicaciones web. También se incluyen antecedentes de proyectos similares y el valor agregado que ofrece la propuesta de Multidoku. En el capítulo 3 se detalla los requerimientos funcionales y no funcionales del sistema, los diagramas de casos de uso, los perfiles de usuario y las principales restricciones. Este capítulo establece la base técnica que guiará el diseño y posterior implementación. En el capítulo 4 se describirá la arquitectura planteada para el sistema, tanto a nivel de software como de infraestructura. Se incluyen diagramas de clases, diagramas de secuencia, estructura de la base de datos y componentes principales de la aplicación. En el capítulo 5 se explica el proceso de construcción del sistema, las herramientas empleadas, los módulos desarrollados y la integración de las diferentes funcionalidades, como la lógica del juego, el generador de tableros y la interfaz gráfica. En el capítulo 6 se expone los métodos de prueba utilizados, los casos de prueba aplicados y los resultados obtenidos. Asimismo, se analiza el desempeño del sistema, posibles limitaciones y el grado de cumplimiento de los objetivos establecidos. Finalmente, en el capítulo 7 se presenta una síntesis del proyecto, destacando los logros alcanzados, las lecciones aprendidas y las recomendaciones para futuras mejoras o posibles líneas de trabajo relacionadas con el Sudoku Web.

En conjunto, esta estructura busca proporcionar un recorrido claro y ordenado desde la concepción inicial del proyecto hasta la presentación de los resultados finales, facilitando la comprensión integral del proceso de desarrollo y su justificación técnica.

# DESCRIPCIÓN DEL PRODUCTO

## Descripción del producto a desarrollar

Nos hemos embarcado en la creación de una aplicación web, a la que hemos bautizado como Sudoku Web. La idea central es brindar a los usuarios un espacio interactivo donde puedan disfrutar de Sudokus tradicionales, como 4x4, 6x6 y una versión renovada del tradicional Sudoku, y Multidokus. El sistema tendrá la capacidad de crear tableros de forma automática, verificar el cumplimiento de las reglas al instante, y proporcionar una experiencia a medida con distintos niveles de desafío y un cronómetro.

Los problemas que se abordan es la escasez de plataformas de variantes de sudoku interactivas con este tipo de sudoku. También está la falta de aplicaciones accesibles desde la web que no requieran de instalación en dispositivos.

Nuestros objetivos son ofrecer un sistema web intuitivo y accesible para jugar esta variable de sudoku. Integrar una interfaz intuitiva con diseño responsivo. Inclusión de un generador de tableros y de validación para una buena experiencia de usuario.

El Sudoku es un juego lógico con alta aceptación a nivel mundial. Sin embargo, la mayoría de las aplicaciones existentes se enfocan únicamente en lo básico y no aprovechan las posibilidades que brinda la web para mejorar la experiencia del usuario. Este proyecto surge con la motivación de combinar entretenimiento, desarrollo cognitivo y accesibilidad digital en un solo producto, brindando una alternativa atractiva tanto a jugadores casuales como a entusiastas de los rompecabezas.

El Sudoku es un juego matemático, el cual trata de un rompecabezas matemático cuyo objetivo es rellenar una cuadricula de 9x9 celdas, dividida en subcuadrículas de 3x3 con las cifras de 1 al 9, partiendo de algunos números ya dispuestos en algunas de las celdas.

Las reglas del sudoku son: en una cuadrícula de 9x9, cada fila, cada columna y cada caja de 3x3 deben contener todos los números del 1 al 9, sin repetición. El objetivo es rellenar los espacios vacíos con estos números siguiendo estas tres reglas fundamentales para completar el rompecabezas.

**Cómo jugar:**

* Utiliza el razonamiento deductivo: A diferencia de los juegos de azar, el Sudoku requiere lógica. Puedes deducir dónde va un número observando las otras casillas de la fila, columna y bloque.
* Proceso de eliminación: Si sabes dónde están ciertos números, puedes eliminar esos mismos números de las posibilidades en otras casillas de la misma fila, columna o bloque.
* Busca "únicos obvios": A veces, solo habrá un número que pueda ir en una casilla específica dentro de su fila, columna y bloque.
* Verifica la solución: El juego está bien planteado si tiene una solución única

Existen diferentes tipos de Sudokus, donde se pueden clasificar de diferentes maneras, por tamaño, por el tipo de elementos, o por reglas adicionales que se añaden al sudoku clásico también se pueden clasificar por dificultad, desde el nivel fácil hasta el experto, para adaptarse a diferentes niveles de habilidad.

**Por tamaño de la cuadrícula:**

* [Sub-Doku](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifOI4Rl_ZRZy6aN9BkIX8LasT5yuhQ%3A1756141202693&q=Sub-Doku&sa=X&ved=2ahUKEwj7nb_kt6aPAxXMrpUCHbAXHtAQxccNegQIEBAB&mstk=AUtExfDhghjRBzpyYJodFS6XDYV3j1uiiSOWJ0LmisC6C_2e_DYatytI5o2UzLC-E4Hor3Qk6QXKLCIpvi3HjHFXdt0RjctyzciaGS7GBfL9ASd9OQnCBKGY4PYeVWSzG1QVCAm6YNdLTWx4-U4rRiywZCGRhkeKpoEdjTPMPRxu7UhEgeFslVobhm-QLRF5z-F4CtU_2CZ3Qm_TbXPvBeXBFwp-vCiUAiDr-8ojasl7o3nsVAv3QUpfFuDsi4pYPI_qoUnBpsgQ8qw7L7FFyU4rt0my&csui=3): Cuadrículas más pequeñas que las 9x9, a menudo utilizadas para niños, como el 4x4.
* [Sudoku Clásico](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifOI4Rl_ZRZy6aN9BkIX8LasT5yuhQ%3A1756141202693&q=Sudoku+Cl%C3%A1sico&sa=X&ved=2ahUKEwj7nb_kt6aPAxXMrpUCHbAXHtAQxccNegQIERAB&mstk=AUtExfDhghjRBzpyYJodFS6XDYV3j1uiiSOWJ0LmisC6C_2e_DYatytI5o2UzLC-E4Hor3Qk6QXKLCIpvi3HjHFXdt0RjctyzciaGS7GBfL9ASd9OQnCBKGY4PYeVWSzG1QVCAm6YNdLTWx4-U4rRiywZCGRhkeKpoEdjTPMPRxu7UhEgeFslVobhm-QLRF5z-F4CtU_2CZ3Qm_TbXPvBeXBFwp-vCiUAiDr-8ojasl7o3nsVAv3QUpfFuDsi4pYPI_qoUnBpsgQ8qw7L7FFyU4rt0my&csui=3): El Sudoku estándar de 9x9.
* [Super-Doku](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifOI4Rl_ZRZy6aN9BkIX8LasT5yuhQ%3A1756141202693&q=Super-Doku&sa=X&ved=2ahUKEwj7nb_kt6aPAxXMrpUCHbAXHtAQxccNegUIqQEQAQ&mstk=AUtExfDhghjRBzpyYJodFS6XDYV3j1uiiSOWJ0LmisC6C_2e_DYatytI5o2UzLC-E4Hor3Qk6QXKLCIpvi3HjHFXdt0RjctyzciaGS7GBfL9ASd9OQnCBKGY4PYeVWSzG1QVCAm6YNdLTWx4-U4rRiywZCGRhkeKpoEdjTPMPRxu7UhEgeFslVobhm-QLRF5z-F4CtU_2CZ3Qm_TbXPvBeXBFwp-vCiUAiDr-8ojasl7o3nsVAv3QUpfFuDsi4pYPI_qoUnBpsgQ8qw7L7FFyU4rt0my&csui=3): Cuadrículas más grandes que el 9x9.
* [Sudoku-Zilla](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifOI4Rl_ZRZy6aN9BkIX8LasT5yuhQ%3A1756141202693&q=Sudoku-Zilla&sa=X&ved=2ahUKEwj7nb_kt6aPAxXMrpUCHbAXHtAQxccNegUImwEQAQ&mstk=AUtExfDhghjRBzpyYJodFS6XDYV3j1uiiSOWJ0LmisC6C_2e_DYatytI5o2UzLC-E4Hor3Qk6QXKLCIpvi3HjHFXdt0RjctyzciaGS7GBfL9ASd9OQnCBKGY4PYeVWSzG1QVCAm6YNdLTWx4-U4rRiywZCGRhkeKpoEdjTPMPRxu7UhEgeFslVobhm-QLRF5z-F4CtU_2CZ3Qm_TbXPvBeXBFwp-vCiUAiDr-8ojasl7o3nsVAv3QUpfFuDsi4pYPI_qoUnBpsgQ8qw7L7FFyU4rt0my&csui=3): Una cuadrícula con 100 regiones de 10x10, para los más experimentados.
* Por elementos utilizados Sudoku de números: El tipo más común, donde se usan los dígitos del 1 al 9.
* Sudoku de letras: Usa letras en lugar de números. ¿¿¿¿
* Sudoku de símbolos: Utiliza otros símbolos o figuras de colores. ¿¿¿¿

**Por reglas adicionales:**

* Killer Sudoku (Sudoku Asesino): Además de las reglas del Sudoku clásico, hay que rellenar casillas de forma que sumen una cantidad determinada indicada en cada bloque.
* Samurai Sudoku: Consiste en cinco Sudokus interconectados, compartiendo regiones.
* Sudoku X: Se añaden las diagonales principales, que tampoco deben tener números repetidos.
* Greater Than Sudoku: Se usan signos de desigualdad (mayor que > o menor que <) entre casillas, indicando cuál debe ser mayor.
* Consecutive Sudoku: Especifica casillas donde los números deben ser consecutivos.
* Even-Odd Sudoku: Algunas casillas están marcadas para indicar si el número debe ser par o impar.
* Jigsaw Sudoku (Sudoku de Rompecabezas): Las regiones no son cuadradas, sino de formas irregulares.

Los beneficios esperados son un espacio de entretenimiento que se puede generar con esta página. Generación de un producto escalable que pueda extenderse a futuras mejoras como implementación de otras variantes del Sudoku. Y, por último, facilitar el acceso a una aplicación sin necesidad de instalación en dispositivos.

## Situación actual y estado del arte (Fefe)

Investigación inicial sobre trabajos relacionados

En el ámbito de la resolución del Sudoku mediante algoritmos genéticos, se han identificado distintos esfuerzos tanto académicos como provenientes de comunidades de desarrolladores. Estos trabajos constituyen un referente para el presente proyecto, ya que permiten reconocer enfoques previos y contrastar diferentes estrategias de implementación.

Entre los proyectos encontrados destacan los siguientes:

* KarlHeitmann – “Algoritmo Genético Simple: Solucionando SUDOKU”  
   Propuesta que implementa un algoritmo genético básico para abordar la resolución del Sudoku, enfocándose en la simplicidad del diseño y la claridad de la lógica evolutiva.
* Bxtto – “algoritmoGenetico\_sudoku”  
   Desarrollo experimental que aplica operadores genéticos en la resolución del Sudoku, compartido de manera abierta en la comunidad de GitHub como ejemplo práctico del uso de algoritmos evolutivos.
* Fixx910 – “Sudoku-con-Algoritmo-Genético”  
   Implementación orientada a la construcción de un sistema capaz de resolver tableros de Sudoku mediante la técnica de algoritmos genéticos, destacando el interés de la comunidad en explorar variaciones de este enfoque.

Estos proyectos, aunque de carácter principalmente práctico y comunitario, reflejan la relevancia que ha adquirido el Sudoku como banco de pruebas para algoritmos genéticos. Su existencia confirma que este tipo de problemas representa una oportunidad de aprendizaje y experimentación para desarrolladores e investigadores, al mismo tiempo que proporciona una base de comparación para el trabajo que se desarrolla en el presente proyecto.

**Estado del arte**

Un algoritmo genético se inspira en los procesos evolutivos de la naturaleza y se ha consolidado como una técnica versátil para resolver problemas de optimización y búsqueda. Existen diversos problemas clásicos que pueden abordarse mediante esta técnica; en el presente proyecto se selecciona el Sudoku, uno de los desafíos matemáticos más conocidos y estudiados.

Este problema ha sido explorado por distintos investigadores y entusiastas. Por ejemplo, usuarios como Karl Heitmann y Bxtto, pertenecientes a la comunidad de GitHub, han compartido implementaciones experimentales, destacando que los algoritmos genéticos son una “excusa perfecta para que los programadores principiantes se ensucien las manos en un proyecto real”. Más allá de estas aproximaciones prácticas, el Sudoku también ha sido objeto de investigación formal. Norvig (2009) propuso una solución basada en métodos exactos, mientras que Mantere & Koljonen (2007) implementaron un algoritmo genético especializado, convirtiéndose en referentes dentro de la literatura académica.

En el panorama actual, al 2025, los avances en algoritmos genéticos para Sudoku incluyen enfoques híbridos que combinan GA con técnicas de búsqueda local y aprendizaje profundo para mejorar la eficiencia en rompecabezas de alta dificultad.

## Alcance y Funcionalidades generales

El presente proyecto tiene como objetivo principal el desarrollo de una plataforma web orientada a la resolución del Sudoku mediante el uso de algoritmos genéticos. El sistema se plantea como una aplicación interactiva que, además de mostrar los resultados obtenidos por el algoritmo, permitirá al usuario participar activamente en la dinámica del juego. En este contexto, el alcance del producto se define en términos de las siguientes funcionalidades principales:

* **Plataforma web accesible desde cualquier navegador**  
  El sistema estará disponible en línea, garantizando que los usuarios puedan acceder a él sin necesidad de instalaciones adicionales, únicamente mediante un navegador web.
* **Interfaz de usuario con tablero de Sudoku responsivo**  
  Se dispondrá de una interfaz interactiva que representará el tablero de Sudoku en una cuadrícula 9×9. Esta será adaptable a diferentes dispositivos, de modo que la experiencia de uso se mantenga fluida tanto en equipos de escritorio como en dispositivos móviles.
* **Generador de tableros con distintos niveles de dificultad**  
  El sistema incorporará un generador capaz de producir tableros con diferentes grados de complejidad (fácil, medio y difícil), lo que permitirá evaluar el desempeño del algoritmo en contextos de distinta exigencia.
* **Validación de jugadas**  
  El tablero contará con un mecanismo de validación que verificará en tiempo real la corrección de las jugadas realizadas por el usuario, respetando las reglas fundamentales del Sudoku en filas, columnas y subcuadros.
* **Cronómetro**  
  Se incluirá un cronómetro que registre el tiempo utilizado en la resolución de cada partida, lo que permitirá tanto medir la eficiencia del usuario como establecer una referencia para evaluar el rendimiento del algoritmo.

## Restricciones y elementos fuera del alcance

No se desarrollará aplicación móvil nativa (el sistema será únicamente web).

## Diagrama de Contexto

En el siguiente diagrama de contexto, posiciona a la aplicación web “Sudoku Web” ilustrando las interacciones entre el sistema y las entidades externas que interactúan con él, incluyendo usuarios tecnologías y stakeholders. El diagrama muestra la generación y resolución de sudokus accesibles a través de navegador web. Se proporciona una visión clara y general del alane del sistema, destacando accesibilidad y usabilidad.

<https://www.canva.com/design/DAGxTeiBSBk/IBfp7oWeBewEr-jQz11ung/edit?utm_content=DAGxTeiBSBk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton> -> en proceso diagrama contexto

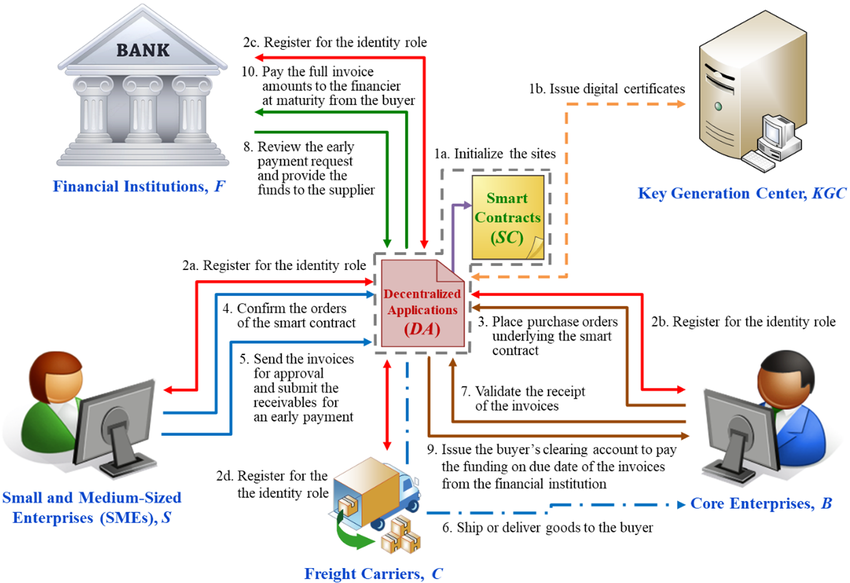


Diagrama de contexto

Fuente: Creada por los estudiantes respecto al proyecto

# REQUISITOS

Este capítulo contiene la especificación completa de los requisitos del sistema software que se va a desarrollar.

Se clasifican en requisitos funcionales y no funcionales, se redactan siguiendo una plantilla estandarizada que facilita su comprensión, trazabilidad y verificación.

## Clasificación de los requisitos

Para organizar de manera clara y estructurada los requisitos del sistema Sudoku web, se propone clasificarlos en dos grandes categorías: requisitos funcionales y requisitos no funcionales.

* + Requisitos Funcionales (RF):

Representan las funcionalidades que el sistema debe ofrecer a los usuarios finales. Esos requisitos describen las operaciones esenciales del software, tales como la generación de tableros, la validación de jugadas y el registro de métricas de rendimiento. A su vez, se han agrupado en bloques funcionales que reflejen el alcance del sistema:

1.- Gestión del juego: Generación de tableros válidos, validación de movimientos en vivo, verificación de soluciones, pistas y tutoriales.

2.- Interacción con el Usuario: Interfaz responsiva, accesibilidad, soporte multilenguaje, personalización de temas y ajuste de accesibilidad.

3.- Progreso y Estadísticas: Guardado/carga de partidas, registro de tiempos, estadísticas de rendimiento y métricas opcionales.

4.- Documentación y Aprendizaje: Inclusión de técnicas de resolución, documentación técnica y metodológica, tutorial interactivo y ayuda contextual.

* + Requisitos No Funcionales (RNF):

Se refieren a las características de calidad que el sistema debe cumplir, más allá de la funcionalidad directa. Entre ellos destacan aspectos de usabilidad, rendimiento, seguridad, compatibilidad y mantenibilidad. Estos requisitos se clasifican en:

1.- Usabilidad y Accesibilidad: Cumplimiento de estándares WCAG 2.1 AA, interfaz clara y soporte de accesibilidad.

2. Rendimiento: tiempo de respuesta en generación y resolución de tableros de inferior a los limites definidos.

3. Compatibilidad: Funcionamiento en navegadores modernos y despliegue en entornos web estáticos.

4. Mantenibilidad y Calidad del Código: Diseño modular, pruebas unitarias e integración con al menos un 70% de cobertura.

5. Seguridad y Privacidad: Restricción de datos personales, manejo transparente de cookies y registro controlado de errores.

Con esta clasificación permite estructurar el conjunto de requisitos de manera lógica, garantizando que tantos las funcionalidades como las propiedades de calidad del sistema queden correctamente representadas.

## Plantilla para la redacción de requisitos

Para garantizar uniformidad y claridad en la especificación, cada requisito se redactará utilizando a una plantilla básica que facilite su lectura, trazabilidad y validación. Los atributos seleccionados son:

1.- ID: Código único que identifica el requisito (ejemplo: RF01, RNF3).

2.- Requisito: Nombre breve y representativo de la funcionalidad o característica

3.- Descripción: Explicación detalla del requisito, indicando su propósito y alcance.

4 y 5 nose si van dependencias o pruebas de verificación (preguntar al profe) \*\*\*\*

A continuación, se muestra una tabla de cómo se encontrarán los requerimientos:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF01 | Requisito | **GENERANDO TABLEROS SUDOKU VALIDOS** |
| Descripción | Permitiendo la generación de tableros de Sudokus y Multidokus, asegurando que cada uno tenga solución única y diferentes niveles de dificultad (fácil, medio, difícil, experto) | | |
| Pruebas de verificación | Pruebas a realizar para verificar / validar el requisito | | |
| Dependencias | Lista de otros requisitos dependientes o de los que depende | | |

## Requisitos funcionales

Los requisitos funcionales describen las acciones, servicios y comportamientos concretos que el sistema deba ofrecer para cumplir los objetivos planteados. Están orientados a detallar las funcionalidades visibles por los usuarios, así como la interacción que estos mantendrán con la plataforma. En este proyecto, los requisitos funcionales se centran en la gestión del juego de Sudoku y sus variantes, La validación de jugadas, el seguimiento de estadísticas, la personalización de la experiencia de usuario y el soporte de accesibilidad y multilenguaje. Estos constituyen la base operativa del sistema, ya que determinan cómo debe responder ante las diferentes entradas del usuario y que servicios debe garantizar durante la ejecución del juego. A continuación, se dejarán todos los requisitos funcionales realizados.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF01 | Requisito | **GENERANDO TABLEROS SUDOKU VALIDOS** |
| Descripción | Permitiendo la generación de tableros de Sudokus y Multidokus, asegurando que cada uno tenga solución única y diferentes niveles de dificultad (fácil, medio, difícil, experto) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF02 | Requisito | **VALIDANDO EN TIEMPO REAL LOS MOVIMIENTOS** |
| Descripción | Comprobando al escribir si el valor viola reglas de fila/columna/caja y las reglas extra de la variante (diagonales, regiones irregulares, sumas Killer, solapes Samurai). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF03 | Requisito | **COMPROBANDO SOLUCIÓN** |
| Descripción | Verificando si el tablero completado por el usuario cumple con todas las restricciones del juego y mostrando un mensaje de confirmación o error. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF04 | Requisito | **OFRECIENDO PISTAS PARA LA RESOLUCIÓN DEL TABLERO** |
| Descripción | Mostrando posibles jugadas o candidatos válidos para una celda especifica, ayudando al usuario a continuar cuando se encuentre bloqueado. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF05 | Requisito | **EXPLICANDO TECNICAS DE RESOLUCION APLICADAS\*** |
| Descripción | Mostrando al usuario que técnica lógica se puede aplicar en el tablero (como Single, Hidden single, Locked Candidates), para reforzar el aprendizaje del juego. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF06 | Requisito | **PERMITIENDO DESHACER Y REHACER MOVIMIENTOS** |
| Descripción | Manteniendo un historial de acciones del jugador para regresar a un estado anterior o recuperar jugadas eliminadas. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF07 | Requisito | **GUARDANDO Y CARGANDO PARTIDAS** |
| Descripción | Almacenando automáticamente el progreso del usuario y permitiendo recuperar partidas guardadas para continuar desde el mismo punto. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF08 | Requisito | **MOSTRANDO TEMPORIZADOR Y ESTADISTICAS DE RENDIMIENTO DEL USUARIO** |
| Descripción | Registrando el tiempo de resolución, la dificultad seleccionada, la cantidad de pistas utilizadas y otros indicadores de rendimiento del jugador. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF09 | Requisito | **PROPORCIONANDO INTERFAZ CLARA, RESPONSIVA Y ACCESIBLE** |
| Descripción | Diseñando la interfaz de forma intuitiva, responsiva y adaptable a distintos dispositivos (móvil y escritorio). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF10 | Requisito | **OFRECIENDO PERSONALIZACIÓN DE TEMAS Y AJUSTES DE ACCESIBILIDAD** |
| Descripción | Brindando al usuario la opción de elegir entre modo oscuro, claro, alto contraste y ajustar tamaños de letra. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF11 | Requisito | **SOPORTANDO MULTIPLES IDIOMAS** |
| Descripción | Permitiendo que el sistema muestre su interfaz en distintos idiomas. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RF12 | Requisito | **INCLUYENDO TUTORIAL INTERACTIVO Y PAGINA DE AYUDA PARA EXPLICAR LAS REGLAS DEL JUEGO** |
| Descripción | Guiando al usuario en su primera interacción con la aplicación y ofreciendo una sección de ayuda con reglas del juego y consejos de uso. | | |

## Requisitos no funcionales

Los requisitos no funcionales especifican las propiedades de calidad y las restricciones técnicas que el sistema debe cumplir, independientemente de las funcionalidades que ofrezca. Se relacionan con aspectos de rendimiento, seguridad, accesibilidad, compatibilidad, usabilidad, privacidad y mantenibilidad, asegurando que la experiencia de uso sea fluida, confiable y segura.

Para este proyecto, los requisitos no funcionales establecen parámetros como tiempos máximos de respuesta, estándares de accesibilidad, compatibilidad con navegadores modernos y lineamientos de privacidad de datos. De esta manera, completan los requisitos funcionales, garantizando que el sistema no solo funcione correctamente, sino que además cumpla con criterios de calidad que lo hagan sostenible y escalable con el tiempo. A continuación, se dejarán los requisitos no funcionales realizados.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF01 | Requisito | **ASEGURANDO LA USABILIDAD CON FORME A LAS PAUTAS DE ACCESIBILIDAD WCAG 2.1 AA.** |
| Descripción | Cumpliendo con estándares WCAG 2.1 AA, garantizando que la aplicación sea usable por personas con distintas capacidades. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF02 | Requisito | **GARANTIZANDO TIEMPO DE RESPUESTAS MENOR A 1,5 SEGUNDOS EN LA GENERACIÓN DE TABLEROS** |
| Descripción | Manteniendo el tiempo de generación por debajo de 1,5 segundos para asegurar fluidez en el uso. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF03 | Requisito | **EJECUTANDO LA RESOLUCIÓN AUTOMATICA DE TABLEROS EN MENOS DE 1 SEGUNDO** |
| Descripción | Optimizando el algoritmo de resolución para responder rápidamente a tableros de cualquier dificultad. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF04 | Requisito | **MANTENIENDO COMPATIBILIDAD CON NAVEGADORES MODERNOS** |
| Descripción | Asegurando el correcto funcionamiento en las últimas versiones de Chrome, Firefox, Edge y Safari. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF05 | Requisito | **DISEÑADO EL SISTEMA CON CODIGO MODULAR** |
| Descripción | Organizando el código en módulos independientes para facilitar su mantenimiento y futuras mejoras. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF06 | Requisito | **IMPLEMENTANDO PRUEBAS UNITARIAS Y DE INTEGRACIÓN PARA ALCANZAR UNA COBERTURA MINIMA DEL 70%** |
| Descripción | Alcanzando un nivel mínimo del 70% de cobertura en el núcleo del sistema para garantizar calidad y confiabilidad. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF07 | Requisito | **REESTRINGIENDO LA RECOPILACIÓN DE DATOS PERSONALES Y APLICANDO MEDIDAS BASICAS DE SEGURIDAD** |
| Descripción | Limitando la recolección de información a lo estrictamente necesario y aplicando medidas básicas de seguridad. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF08 | Requisito | **REGISTRANDO ERRORES DE FORMA CONTROLADA MINIMIZANDO FALLOS EN LA EXPERIENCIA DEL USUARIO\*** |
| Descripción | Capturando fallos y excepciones en el sistema para prevenir interrupciones en la experiencia de uso. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF09 | Requisito | **DESPLEGANDO EL SISTEMA EN ENTORNOS WEB ESTATICOS\*** |
| Descripción | |  | | --- | |  |  |  | | --- | | Construyendo la app para hosting estático y/o serverless, sin dependencias de servidor persistente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF10 | Requisito | **CENTRALIZANDO LOS TEXTOS DE LA INTERFAZ PARA SOPORTAR TRADUCCIONES** |
| Descripción | Separando las cadenas de texto de la lógica para facilitar la traducción y localización del sistema. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF11 | Requisito | **RESPETANDO LA PRIVACIDAD DEL USUARIO SIN EMPLEAR COOKIES NO ESENCIALES POR DEFECTO** |
| Descripción | Evitando el uso de cookies no esenciales y manteniendo la transparencia en el manejo de datos. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RNF12 | Requisito | **DOCUMENTANDO BUILD, TEST Y DEPLOY REPRODUCIBLES** |
| Descripción | |  | | --- | | Proveyendo scripts y guía paso a paso que permitan levantar el proyecto con pocos comandos. |  |  | | --- | |  | | | |

## Vocabulario del dominio

El vocabulario del dominio reúne los conceptos fundamentales asociados al mundo de los juegos de lógica tipo Sudoku y sus variantes, los cuales constituyen la base temática sobre la cual se desarrolla el proyecto. Estos términos permiten delimitar el alcance del sistema, describiendo las reglas, estructuras y particularidades de cada modalidad de Sudoku y Multidoku. Su correcta definición garantiza una comprensión compartida entre los desarrolladores y los usuarios, facilitando tanto el análisis como el diseño del sistema.

### Vocabulario del modelo conceptual

El vocabulario del modelo conceptual reúne los términos que representan las entidades, relaciones y componentes internos del sistema, los cuales sirven de base para el diseño del modelo de clases y diagramas conceptuales. Estos conceptos permiten describir cómo se organiza la información del dominio del Sudoku dentro del sistema, asegurando una representación clara y coherente entre la realidad del problema y su implementación en software.

|  |  |
| --- | --- |
| TÉRMINO | DESCRIPCIÓN |
| Sudoku | Un rompecabezas lógico-matemático jugado en una cuadrícula de 9x9, dividida en 9 subcuadrículas de 3x3. |
| Multidoku | Una variante avanzada del Sudoku que involucra múltiples cuadrículas de 9x9 interconectadas, comúnmente en formaciones como una cruz. |
| Tablero | Cuadrícula de 9x9 donde se colocan números del 1 al 9, con celdas iniciales prellenadas y celdas vacías para completar. |
| Celda | Espacio individual en el tablero donde se coloca un número (1–9) o permanece vacío. |
| Dificultad | Nivel de complejidad de un tablero (fácil, medio, difícil, experto), determinado por el número de celdas prellenadas y la lógica requerida para resolverlo. El sistema genera tableros con distintos niveles de dificultad. |
| Validación en vivo | Proceso que verifica en tiempo real si un número ingresado en una celda cumple las reglas del Sudoku (sin repeticiones en fila, columna o subcuadrícula). |
| Pista | Sugerencia proporcionada al usuario, como un número válido para una celda específica o una lista de candidatos posibles, para ayudar cuando está bloqueado en la resolución del tablero. |
|  |  |
|  |  |
|  |  |

Tabla de vocabulario del dominio. Modelo Conceptual

### Vocabulario técnico

El vocabulario técnico en este proyecto está orientado a los conceptos propios de la computación evolutiva y, en particular, de los algoritmos genéticos (GA), los cuales se utilizan como parte de la estrategia de generación y resolución de Sudokus y Multidokus. Estos términos permiten describir la manera en que el sistema modela posibles soluciones como poblaciones de individuos, representados por cromosomas y genes, los cuales evolucionan mediante operadores como la selección y la mutación. Además, se incluyen nociones específicas relacionadas con la dificultad de los tableros, que constituyen un parámetro clave en el proceso de evaluación y validación de soluciones. Su definición asegura un entendimiento común del enfoque técnico utilizado en el desarrollo del sistema.

|  |  |
| --- | --- |
| TÉRMINO | DESCRIPCIÓN |
| Algoritmo genético (GA) | Técnica computacional basada en evolución natural, usada para generar y resolver tableros de Cross Sudoku, optimizando soluciones mediante población, selección, cruce y mutación. |
| Población | Conjunto de posibles soluciones en el algoritmo genético, cada una representada como un individuo. |
| Individuo | Una solución candidata en el algoritmo genético, representada como un tablero de Sudoku con celdas llenas. |
| Cromosoma | Representación codificada de un individuo en el GA, que contiene genes. |
| Gen | Unidad básica del cromosoma, representando el valor de una celda específica en el tablero. |
| Selección | Proceso del algoritmo genético que elige individuos con mejor fitness para reproducirse, utilizando métodos como selección por ruleta o torneo, para generar la siguiente población. |
| Cruce (Crossover) | Operador del algoritmo genético que combina dos individuos (tableros) para crear nuevos individuos, intercambiando partes de sus cromosomas (e.g., intercambiar filas o bloques) para explorar nuevas soluciones. |
| Mutación | Operador del algoritmo genético que altera aleatoriamente uno o más genes en un cromosoma para introducir diversidad en la población. |
| Interfaz Responsiva | Diseño de la interfaz de usuario que se adapta automáticamente a diferentes dispositivos y tamaños de pantalla, garantizando una experiencia de usuario consistente y fluida. |
| Pruebas Unitarias | Método de verificación del código que evalúa unidades individuales. |
| * + WCAG 2.1 AA |  |
|  |  |

Tabla de vocabulario del dominio. Vocabulario técnico

# ALGORITMO GENÉTICO

Definición de los principales artefactos del algoritmo genético a utilizar. Modelado de poblaciones e individuos, algoritmos principales (fitness, cruzamiento, mutación, reemplazo…).

Se puede programar con más de una función y parametrizar, o cambiar de función según circunstancias. En este caso se debe explicar aquí, en cada apartado del documento.

Utilizar diagramas visuales que faciliten el entendimiento del AG.

## Población / individuo / cromosoma / gen

Modelar aquí, qué es un individuo, cómo se organizan los cromosomas, qué son los genes. Tipo de codificación y significado (binaria, etc…)

## Creación de población

## Algoritmo de selección – Función fitness

## Algoritmo de creación de pool

## Algoritmos de cruzamiento

## Algoritmo de mutación

## Algoritmo de reemplazo

# MODELADO CONCEPTUAL

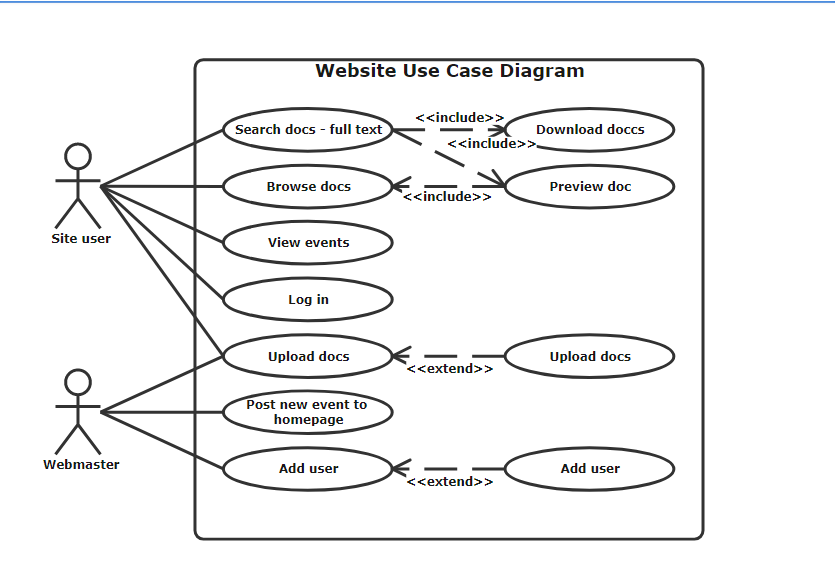
Esta sección debe contener un modelo cuyo propósito es especificar la *modelo de información* del sistema que se desea construir, mediante un conjunto de diagramas de clases adecuadamente explicados

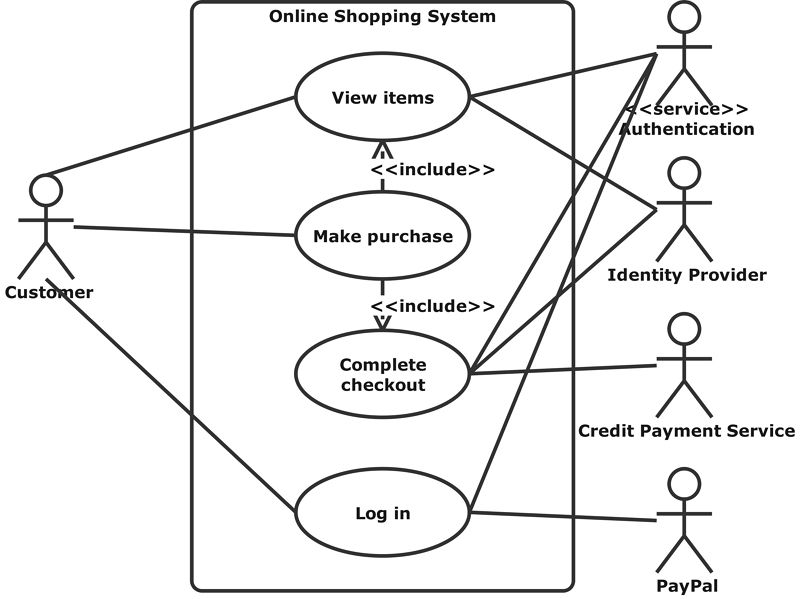
El modelo de información, o modelo conceptual, debe estar adecuadamente *justificado* a partir de los requisitos. No tiene sentido que en él aparezcan clases, atributos, operaciones y otros elementos que no hayan aparecido anteriormente en los requisitos. Igualmente, no tiene sentido que en los requisitos se mencionen conceptos importantes que no aparezcan reflejados de ninguna manera en el modelo conceptual. La referencia al *vocabulario del dominio* es nuevamente un puente importante que vincula los requisitos con el modelo conceptual.

## Diagrama de Casos de Uso

Seguir el estándar UML. Identificar todos los actores y representar sus funciones. Hay una buena explicación en: <https://www.edrawsoft.com/es/uml-use-case.html>

Mencionar la herramienta de modelado que se utiliza en la práctica para la elaboración de los distintos diagramas exigidos. Ejemplo de herramienta: edraw permite elaboración de diagramas on-line UML. <https://www.edrawsoft.com/es/edraw-uml.html>





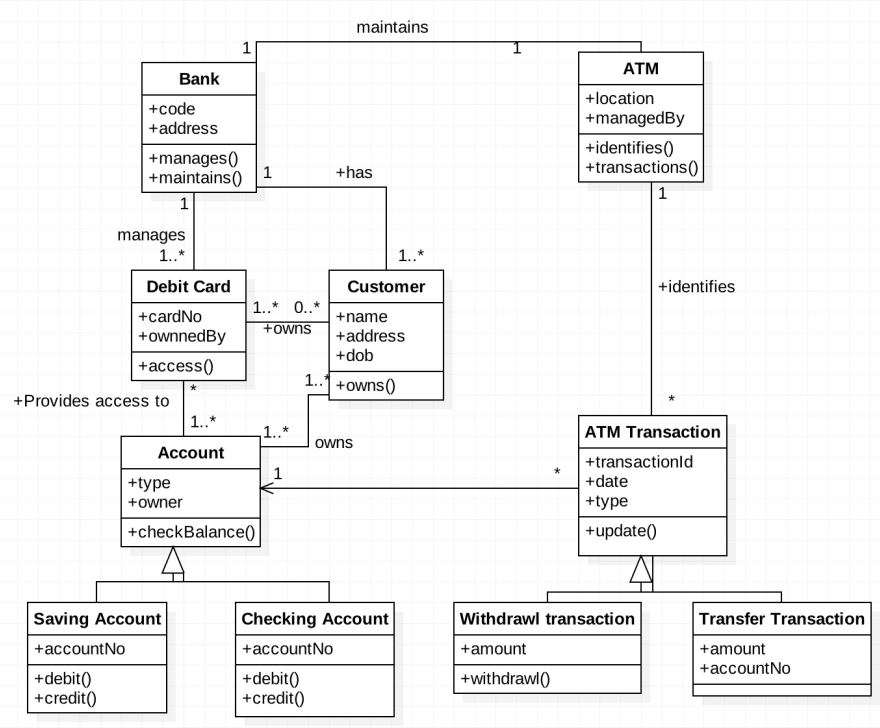
Ejempls de diagrama de Casos de uso. Fuente: <https://www.edrawsoft.com/es/uml-use-case.html>

## Diagramas de Clases

Los diagramas de clases deben ayudar a entender el sistema. En consecuencia, a estos diagramas no se les debe añadir tanta información que sean difíciles de comprender, ni tan poca información que no añadan nada a lo que ya se dice en los requisitos. Estos diagramas deben tener el grado de detalle suficiente para conservar su carácter comunicativo, en tanto que constituyen una vista gráfica de los requisitos. Por tanto, deberán contener todas las clases conceptuales (es decir, las que aparecen mencionadas en los requisitos), pero no las que sean exigencia exclusivamente del diseño posterior. Así mismo, estas clases aparecerán con los atributos que sean relevantes desde el punto de vista conceptual. Para aumentar la claridad y potencia expresiva, puede ser conveniente mostrar un diagrama global en el que sólo aparezcan las clases, y varios diagramas parciales en los que aparezcan clases con atributos.

No incluir las operaciones en las clases, para simplificar y disminuir el trabajo. Se trata de hacer un modelo de información.

Mencionar la herramienta de modelado que se utiliza en la prácica para la elaboración de los distintos diagramas exigidos. Ejemplo de herramienta: edraw permite elaboración de diagramas on-line UML. <https://www.edrawsoft.com/es/edraw-uml.html>



Ejemplo diagrama de clases (nota.- no incluir métodos en este proyecto) Fuente: <https://www.edrawsoft.com/es/example-uml-class-diagram.html>

# ARQUITECTURA

## Modelo de implementación

En esta sección se presenta y justifica la *arquitectura de desarrollo* elegida (descomposición del sistema en subsistemas y componentes), y se especifican las dependencias entre los distintos componentes que hayan resultado de la descomposición. La descomposición se hace habitualmente a varios niveles (componentes que a su vez se descomponen en subcomponentes, etc.), pero en esta práctica sólo se trata la descomposición de primer nivel, centrada en la vista externa (interfaces) de los componentes, y las relaciones entre ellos.

Se deben utilizar los diagramas y vistas que sean necesarios y las oportunas explicaciones textuales. Vale lo dicho en la sección 4.1 sobre la adecuada combinación de imágenes y palabras

Mencionar la herramienta de modelado que se utiliza en la prácica para la elaboración de los distintos diagramas exigidos. Ejemplo de herramienta: edraw permite elaboración de diagramas on-line UML. <https://www.edrawsoft.com/es/edraw-uml.html>

Interfaz de usuario gráfica

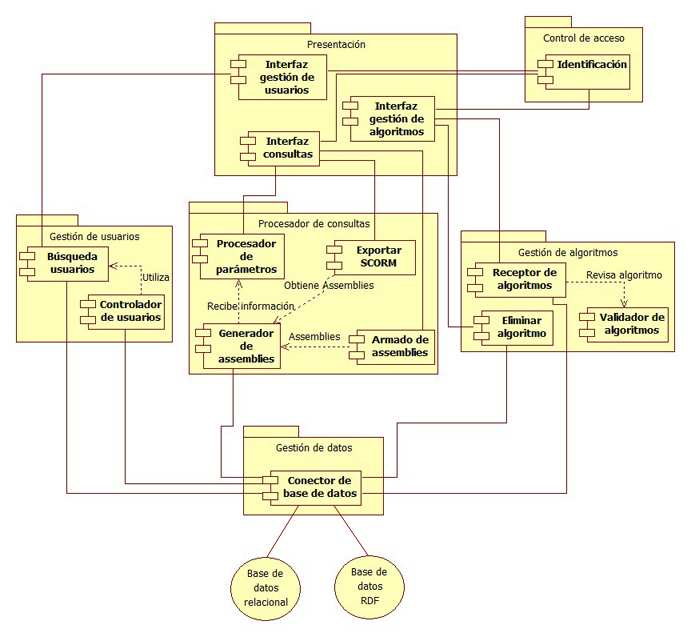
El contenido generado por IA puede ser incorrecto.

Ejemplo de modelo arquitectural en capas. Fuente: https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay

## Diagrama de componentes

Diagrama

El contenido generado por IA puede ser incorrecto.



Ejhemplos diagramas de componentes. Fuente: <https://www.researchgate.net/figure/Figura-2-Diagramas-de-componentes-Con-el-proposito-de-ampliar-lo-descrito-por-el_fig2_255751413>

### Componente A (repetir para cada componente)

Por cada componente se repite la misma estructura con los siguientes apartados:

Tipo

Debe indicarse esencialmente si es ejecutable o no ejecutable. Componentes no ejecutables serían archivos de configuración, plantillas de páginas web, etc. El contenido de los demás apartados de la descripción del componente depende del tipo de componente: si es ejecutable, habrá que hacer énfasis en el Procesamiento (por ejemplo, métodos ofrecidos por las clases de implementación); si no es ejecutable, en los Datos (por ejemplo, información contenida en una página web). Los apartados de la descripción no son fijos, sino que deben adaptarse al tipo de componente.

Propósito [Función, Procesamiento, Datos, Recursos]

El propósito del componente se define esencialmente trazándolo contra los requisitos que el componente implementa. Esta trazabilidad hacia atrás justifica la existencia del componente y explica su propósito. Este apartado contiene una breve descripción textual del propósito del componente, basada en esos requisitos. El detalle de la trazabilidad se especifica en la sección 5.3.

Dependencias (interfaces requeridas)

Se definen las dependencias respecto a interfaces proporcionadas por otros componentes del sistema, o por sistemas externos. En el primer caso, dado que toda *interfaz requerida* tiene su contrapartida en una *interfaz proporcionada*, no es necesario repetir la especificación de la interfaz en ambos lugares; basta con especificarla como interfaz proporcionada y referenciarla adecuadamente en el otro lugar donde es requerida. En el segundo caso, sí es necesario especificar completamente la interfaz requerida, puesto que la interfaz proporcionada queda fuera del sistema.

Interfaces (Interfaces proporcionadas)

De modo general una interfaz es un conjunto de operaciones que ofrecen un servicio coherente. En diseño orientado a objetos se va más allá: una interfaz *define un tipo*, que proporciona un conjunto coherente de operaciones sobre las instancias compatibles con ese tipo.

La definición de interfaces permite aislar componentes y lograr un diseño modular y mantenible mediante interacciones bien definidas. Para definir correctamente el tipo especificado por la interfaz, se enumeran las operaciones pertenecientes a la interfaz, así como los contratos que deben satisfacer cada una de las operaciones. Para especificar los contratos debe utilizarse la técnica de *diseño por contratos* vista en el curso.

# LISTAS Y TABLAS

## Consistencia entre requisitos: conflictos, redundancias, acoplamientos

## Trazabilidad requisitos – modelo conceptual (clases)

## Trazabilidad requisitos – modelo de implementación (componentes)

En estas tres secciones pueden utilizarse tablas de doble entrada o tablas de 2-3 columnas. La tabla de doble entrada tiene el peligro de ser excesivamente dispersa, por lo que a menudo las tablas simplificadas de pocas columnas son una solución más conveniente.

# ANEXOS

## Anexo I – Bibliografía

Lista completa de *todos los documentos citados* (estándares, manuales, páginas web, herramientas utilizadas, etc.), identificados por su autor, título y fecha, u otros datos análogos que permitan su localización unívoca (título de la revista u organización que lo ha publicado, URL y fecha de descarga, etc.).

Nota.- Usar algún estándar. Ver información sobre estándares bibliográficos en: <https://uc3m.libguides.com/guias_tematicas/citas_bibliograficas/inicio>. MS-Word permite el uso de estos estándares (ver pestaña Referencias – Citas y bibliografía)

## Anexo II – Uso de IA en el proyecto

Resumen de herramientas IA usadas en el proyecto y para qué se han usado. Incorporar también principales prompts y comentarios a los resultados

### Herramientas y uso

### Prompts relevantes utilizados

### Comentarios y valoración de las herramientas