



**BUAP**

**Benemérita Universidad Autónoma de Puebla**

Facultad de Ciencias de la Computación

Tesis presentada para obtener el Título de Licenciatura en Ingeniería en  
Ciencias de la Computación

Implementación de un sistema de juego para desarrollar habilidades de  
programación básica basados en el paradigma de la programación  
orientada a objetos en clases

Presenta

**Alma Delia Gasparino Tlatelpa**

Director y Asesor

**Dra. Guillermina Sánchez Roman**

Agosto 2022

# Índice

<b>Introducción</b>	<b>4</b>
<b>Capítulo 1. Planteamiento del problema</b>	<b>7</b>
1.1 Antecedentes	7
1.2 Descripción del problema	7
1.3 Objetivo general y específico	9
1.3.1 Objetivo general	9
1.3.2 Objetivos específicos	9
1.4 Hipótesis	10
1.5 Alcances y limitaciones	10
1.5.1 Alcances	10
1.5.2. Limitaciones	10
1.6 Justificación	10
<b>Capítulo 2 Marco Teórico</b>	<b>11</b>
2.1 Gamificación	11
2.1.1 Juegos serios	11
2.1.2 Elementos básicos que aseguran la efectividad de un juego serio	13
2.2 Gamificación para el aprendizaje	17
2.3 Discapacidad	18
2.3.1 Clasificación de discapacidades	18
2.3.2 Discapacidad motriz	20
2.3.2.1 Métodos de acceso para personas con discapacidad motriz	21
2.4 Herramientas de desarrollo utilizadas	23
2.5 Metodología de desarrollo	31
2.5.1 Metodología SCRUM	31
<b>Capítulo 3 Estado del arte</b>	<b>35</b>
3.1 Sistemas y herramientas desarrolladas para aprender programación empleando la gamificación	35
3.2 Videojuegos enfocados al aprendizaje de la Programación Orientada a Objetos	37
3.3 Herramienta de aprendizaje para personas con capacidades diferentes	39
<b>Capítulo 4 Diseño y desarrollo del Sistema</b>	<b>41</b>
4.1 Requerimientos generales	41
4.2 Infraestructura del sistema	48
4.3 Diseño de juegos	50
4.4 Implementación del sistema	54
4.4.1 Inicio al sistema de aprendizaje propuesto	55
4.4.2 Juego atributos	63

4.4.3 Características de la aplicación de la gamificación	67
4.5 Características de accesibilidad en el juego serio desarrollado	71
<b>Capítulo 5 Pruebas y resultados</b>	<b>77</b>
5.1 Pruebas internas de desarrollo	77
5.2 Pruebas de lanzamiento del sistema	77
<b>Capítulo 6 Conclusiones y trabajo futuro</b>	<b>85</b>
<b>Referencias bibliográficas</b>	<b>87</b>
<b>Anexos</b>	<b>90</b>

# Introducción

Conforme se avanza en las asignaturas de las carreras del área de computación, los alumnos se suelen encontrar con la dificultad para aprender los principales conceptos de la materia y enfrentarse al uso del razonamiento algorítmico. A todo esto se añade un nivel de dificultad adicional cuando es necesario resolver problemas empleando un lenguaje de programación [1]. “La mayoría de los cursos introductorios de ciencias de computación se basan en un lenguaje comercial, como JAVA o C++ (...) que no están diseñados para la educación, sino que están destinados a ser utilizados para el desarrollo de software a gran escala. La complejidad de estos lenguajes(...) y las altas tasas de abandono de los cursos introductorios han contribuido a la percepción de la ciencia de computación como una disciplina aburrida y técnicamente difícil entre los estudiantes”[2].

Para resolver este problema, muchas universidades han implementado diversas estrategias lúdicas, que en general han dado buenos resultados en el proceso de aprendizaje de la programación [3]. Es un buen momento para considerar proveer de a los usuarios la habilidad de acceder a información, servicios y entretenimiento de manera independiente por medio de recursos computacionales que tengan características de adaptación [4], es decir, que sean accesibles para alumnos con alguna discapacidad motriz, pues al igual que el resto de los estudiantes, desean aprender, pero no cuentan con la habilidad física para hacer uso de estos.

Para el desarrollo de este proyecto se consultaron estudios donde se han gamificado plataformas y desarrollado juegos serios para el aprendizaje de la programación orientada a objetos; sin embargo, en estos trabajos no se encontró la consideración de un diseño de accesibilidad para alumnos con discapacidad motriz ni las dificultades que podrían tener estos estudiantes al momento de interactuar con los recursos, ya que este grupo de usuarios tienen poca coordinación con el puntero y, por lo tanto, se

inclinan a hacer mayor uso del teclado [4], o bien, hacen uso de tecnologías especiales como mousekey y teclados numéricos [5].

Dado que, en general, se han obtenido buenos resultados con el uso de la gamificación para el aprendizaje y se carece de recursos para este grupo de estudiantes, este trabajo tiene como objetivo ofrecer una opción para satisfacer esta necesidad desarrollando una herramienta de aprendizaje que haga uso de la gamificación para explicar conceptos relacionados a la programación orientada a objetos como atributos, métodos, clases y, además, sea accesible para personas con discapacidad motriz.

Este trabajo de tesis se encuentra organizado y conformado por seis capítulos, en el primer capítulo, *Planteamiento del problema*, se identifican los obstáculos del proceso de aprendizaje de la programación orientado a objetos que enfrentan los usuarios con discapacidad motriz y cómo en este trabajo de investigación se pretende dar solución a este asunto.

En el capítulo 2, *Estado del arte*, se aborda el tema de programación orientada a objetos y las herramientas que se han creado para el aprendizaje de esta asignatura hasta el momento y cómo la gamificación es aplicada en estos recursos de aprendizaje; del mismo modo, se revisará si existen herramientas para el aprendizaje de programación orientado a objetos enfocado a usuarios con discapacidad motriz.

En el *Marco Teórico*, capítulo 3, se definen los temas y/o conceptos de gamificación, discapacidad motriz, los métodos de acceso en la web para usuarios con discapacidad motriz en los cuales se enfoca este trabajo, herramientas necesarias para desarrollar el recurso que se propone para dar solución al problema planteado.

Seguidamente, en el capítulo 4, *Diseño y desarrollo del sistema*, se describe el diseño de sistema tomando en cuenta las investigaciones de los capítulos anteriores, adicionalmente se mencionan las tecnologías utilizadas para el desarrollo de la herramienta que propone este proyecto

Posteriormente, en el capítulo 5 *Pruebas y resultados*, se describen las pruebas realizadas durante el desarrollo de la herramienta propuesta y las respuestas de los cuestionarios aplicados a los usuarios que probaron el sistema desarrollado.

Finalmente, en el capítulo 6, se puntualiza con las *conclusiones* de los resultados obtenidos y las mejoras que se podrían hacer como trabajo futuro.

## Capítulo 1. Planteamiento del problema

### 1.1 Antecedentes

La complejidad del aprendizaje de la programación orientada a objetos es una cuestión a la que se ha tratado de dar solución en diversas universidades como las mencionadas por Campaña et al. [1] y Corsi [3]; sin embargo, estos juegos serios no hacen mención para conocer qué tan usables son para estudiantes con discapacidad motriz fina, es decir, qué tanto puede un usuario con limitación de movimientos finos de las manos hacer uso de éstos para su aprendizaje. Durante la investigación de este trabajo se consultaron varios trabajos de investigación lo más cercanos posible al objetivo de este proyecto, entre los cuales se encuentran [1, 3, 5, 6, 7, 8, 9, 10, 11, 12]. López y Sánchez [5] encontraron un recurso llamado *scratch* que, por medio del uso de rampas digitales, por ejemplo, agregando un teclado numérico, este software podría ser utilizado por usuarios con discapacidad motriz. Considerando los trabajos consultados para esta investigación se hace evidente que existe una gran carencia de recursos para este grupo de personas.

### 1.2 Descripción del problema

Los alumnos de diferentes universidades presentan dificultades y desafíos para

aprender los conceptos de la programación pues necesitan desarrollar varias habilidades a la vez como el aprendizaje de un nuevo lenguaje de programación o la lógica de la programación, como mencionan Campaña et al. [1] y Corsi [3]. De acuerdo con las investigaciones consultadas para este trabajo, este problema se ha tratado de solucionar mediante el uso de diferentes estrategias, medios, recursos y dinámicas dentro del aula, de los cuales el uso de la gamificación ha sobresalido, arrojando mejores resultados, específicamente, cuando se ha hecho uso de los juegos serios donde se ha desarrollado diversos juegos que tratan de ser un medio de apoyo para el aprendizaje de la programación orientado a objetos por ejemplo “scratch”, “Alice”, “Robocode” que son utilizados en algunas instituciones. A pesar que se han desarrollado varios medios para el aprendizaje, solo se tiene registro que el juego scratch puede ser utilizado para usuarios con discapacidad motriz, esto es posible, adaptando tecnología, por ejemplo, el uso de un “mouse key” y un teclado numérico. El uso de estas rampas digitales puede ayudar a hacer que el juego sea accesible para este grupo de usuarios y, en este caso registrado, se puede demostrar como la interfaz y la dinámica del juego no son aptas por sí solas para personas con discapacidad motriz [5].

En la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla, se imparten tres licenciaturas relacionadas a esta área. En la Licenciatura en Ciencias de la Computación y la Ingeniería en Ciencias de computación se imparte la asignatura de Programación II, donde se explica exclusivamente el tema de programación orientada a objetos, mientras que en la carrera de Ingeniería en Tecnologías de la información este tema se imparte en dos materias que son Programación I y Programación II.

Es importante mencionar que, en las materias de cada plan de estudios posteriores a las mencionadas en el párrafo anterior, los conocimientos y habilidades desarrolladas en la programación orientada a objetos se retoman para desarrollar algún programa o proyecto. Por este motivo, es importante tener bien cimentadas las bases del tema, pues como se puede notar, el aprendizaje de la programación es un conocimiento

acumulativo.

Los alumnos durante sus estudios de programación presentan dificultades al comprender los conceptos, la lógica de programación y la solución de problemas empleando un lenguaje de programación, por lo que se han desarrollado sistemas gamificados como medio motivador y así facilitar el aprendizaje, sin embargo, estos recursos han dejado de lado la usabilidad, una característica importante pues hay que mencionar que, en algunas ocasiones, en las aulas también se encuentran alumnos con alguna discapacidad motriz y, al hacer uso de estos recursos, puede ser complejo si no tiene las características necesarias de acuerdo a sus necesidades, esto se debe a que al momento de desarrollar el software no se define al público que va hacer uso de este tampoco se investigan los conocimientos ni habilidades que tiene.

## 1.3 Objetivo general y específico

### 1.3.1 Objetivo general

Desarrollar un juego centrado en el paradigma de Programación Orientada a objetos que integre componentes de usabilidad adecuados para personas que tengan discapacidad motriz en las manos.

### 1.3.2 Objetivos específicos

Definir el conjunto de conceptos que se abordarán en el juego y que están relacionados directamente con el tema de clases (atributos, métodos e instancia) del paradigma Orientado a Objetos.

Diseñar, con base en las técnicas de gamificación, cada uno de los minijuegos que conformarán el juego principal para que se entienda un concepto a la vez en cada minijuego.

Diseñar la interfaz de usuario de cada minijuego, tomando en cuenta la usabilidad que

necesitan las personas con discapacidad motriz fina.

Implementar el juego con herramientas y librerías de desarrollo necesarias para que cumpla con la funcionalidad esperada.

Probar la efectividad de la posibilidad de aplicar características de accesibilidad a herramientas de aprendizaje de la programación orientada a objetos para usuarios con discapacidad motriz con limitaciones en extremidades superiores.

## 1.4 Hipótesis

Si se desarrolla un juego serio y, además tomando en cuenta y aplicando las características necesarias que mejore la usabilidad para usuarios con discapacidad motriz, este se convertirá en un recurso viable para el aprendizaje de este grupo de estudiantes.

## 1.5 Alcances y limitaciones

### 1.5.1 Alcances

- Se pretende contribuir con un recurso para el aprendizaje de la programación orientado a objetos apto para el uso de estudiantes con discapacidad motriz con limitaciones en las extremidades superiores, además éste podrá ser utilizado por usuarios sin capacidad diferente.
- Con las características del juego propuesto se busca facilitar el aprendizaje de los conceptos abstractos de atributo, método y clase de la programación orientada a objetos .

### 1.5.2. Limitaciones

- La interfaz del juego sólo se adapta a pantallas de una computadora de escritorio o laptop.

## 1.6 Justificación

Los juegos para el aprendizaje de la programación orientada a objetos normalmente utilizan los lenguajes de Java o C que son difíciles de comprender, más aún si se está recién aprendiendo a programar [2], también muchos de los recursos disponibles considerados en este trabajo no son aptos para usuarios con discapacidad motriz especialmente en donde se tiene poca movilidad de las manos.

## Capítulo 2 Marco Teórico

Antes de hablar de los términos de descripciones y definiciones en gamificación, es necesario definir este último término desde su raíz; ¿qué es un juego? A lo que Kapp [13] lo define como cuando un jugador se va atrapado para seguir jugando ese juego. Se ve atrapado porque la retroalimentación instantánea y la interacción constante están relacionados a los desafíos del juego, el cual está definido por las reglas, que funcionan dentro del sistema para provocar una reacción emocional y, finalmente, como efecto da un resultado cuantificable dentro de una versión abstracta de un sistema completo.

### 2.1 Gamificación

Campaña et al. [1] definen a la gamificación, también denominada *ludificación*, como el uso de mecanismos de juego como la obtención de puntos, insignias y rankings o niveles en contextos no lúdicos, con el objetivo de motivar y captar la atención del jugador. Lo atractivo de la gamificación es que toma la capacidad motivadora de los juegos y la aplica a la vida real [1].

Kapp [13] define a la gamificación como el uso de mecanismos basado en juegos, estética y el pensamiento del juego para captar la atención de las personas, motivar a la acción, promover el aprendizaje y resolver problemas [13].

Para hacer uso genuino de la gamificación, la propuesta debe estar basada en

elementos del juego los cuales son: *la participación, narración de historias, visualización de personajes, resolución de problemas*. También es importante incluir PBL, (Points, Badges, Leaderboards) es decir, incluir *puntos, recompensas, medallas y/o niveles* [13].

### 2.1.1 Juegos serios

Los objetivos de los *juegos serios* y la *gamificación* son relativamente parecidos. Los juegos serios y la gamificación tratan de resolver un problema, motivar a las personas y promover el aprendizaje mediante las técnicas y el uso de pensamiento basado en juegos.

Un juego serio es una experiencia diseñada haciendo uso del mecanismo del juego y el pensamiento del juego, específicamente para educar a las personas en un dominio de contenido específico. Hay varios juegos serios para el liderazgo, las técnicas de venta, y otros temas de negocios, al igual que muchos juegos serios en el campo de la salud. Estas personas abordan los juegos serios como un uso noble de los mecanismos del juego y una manera de involucrarse e interactuar con los estudiantes.

Los juegos serios tienden a adoptar el enfoque de usar al juego dentro de un espacio de juego bien definido como un tablero de juego o dentro de un navegador de computadora, mientras que la gamificación tiende a llevar el uso de un juego fuera de un espacio definido y aplicar el concepto a elementos como subir escalones, el seguimiento del número de millas que se han corrido, o haciendo una llamada de ventas, pero incluso estas son distinciones artificiales.

Kapp [13] considera que el uso de juegos serios es una forma de juego gamificación, porque los juegos serios son un subconjunto específico del metaconcepto de gamificación. La gamificación abarca la idea de agregar elementos del juego, pensamiento del juego y mecanismos del juego al contenido de aprendizaje. El objetivo de la gamificación es tomar el contenido que es típicamente presentado como una clase (lección) o un curso e-learning y agregar elementos basado en juegos (historia,

desafíos, retroalimentación, recompensas, etc.) y crear una oportunidad de aprendizaje gamificada, ya sea en forma de un juego educativo con elementos que parecieran ser de juego aplicado en tareas normales como se puede notar en recursos motivadores para hacer ejercicios como correr, o bien en forma de una experiencia interesante en el salón de clases en donde los estudiantes participan en un desafío basado en una historia para dominar el contenido presentado.

Entender cómo funcionan los juegos y cómo influyen en los estudiantes ayudará a los profesionales a comprender cómo crear futuras experiencias de aprendizaje que sean motivadoras, capaces de captar la atención, y que conduzcan a una mayor retención y aplicación del conocimiento [13].

A los juegos que hacen uso de la tecnología se les conoce como juegos digitales, los cuales pueden clasificarse de acuerdo a su objetivo: entretenimiento (como algún juego de consola como los juegos de *super mario bros*) o juegos serios (como *Robomind*) [7]. Los juegos serios o serious games promueven el aprendizaje, la formación y práctica de habilidades ya que le permite experimentar al usuario en un ambiente simulado que no genera consecuencias[11].

Los autores Centeno, Pisabarro y Vivaracho (2018) coinciden “... un juego serio es aquel cuyo objetivo no es el de divertir sino otro distinto como el de formar, educar o establecer un comportamiento en el jugador. Típicamente el medio para crear un juego serio es la gamificación” [12].

### 2.1.2 Elementos básicos que aseguran la efectividad de un juego serio

Norero anota cinco elementos básicos que aseguran la efectividad de un juego serio: *historia, dinámica de interacción, retroalimentación, simulación y aprendizaje* [11].

La *historia* consigue que el jugador se identifique y se involucre con mayor facilidad a la temática y se vea más motivado a progresar en el juego hasta terminarlo.

La *dinámica de interacción*, establece cómo se llevará a cabo el juego, tener reglas, objetivos, sistema de recompensas, clasificación de los jugadores entre otros.

La *retroalimentación* o *feedback* le permite conocer al usuario el motivo si las decisiones que toma dentro del juego son acertadas o no, e indica cuán cerca se está de conseguir el objetivo. Se puede aplicar de distintas formas como por ejemplo: el uso de PBL *Points, Badges and Leaderboards*, una barra de progreso que muestre el cumplimiento de las tareas o un mensaje.

La *simulación* brinda la oportunidad de contar con ambientes seguros para tomar decisiones además de representar un ambiente virtual donde el usuario se sienta identificado y así establecer una conexión que permita enlazar los conocimientos que se van adquiriendo en el juego con la vida real.

El *aprendizaje* es una característica que define a un juego serio, por lo tanto, el objetivo principal es que el jugador aprenda.

Mihaly Csikszentmihalyi [14] sugiere que debe de haber un equilibrio entre las competencias, metas conseguidas y tiempo de aprendizaje, de esta forma evitamos la ansiedad o frustración y conseguimos que se perciba que se está avanzando como se aprecia en la Figura 1 .

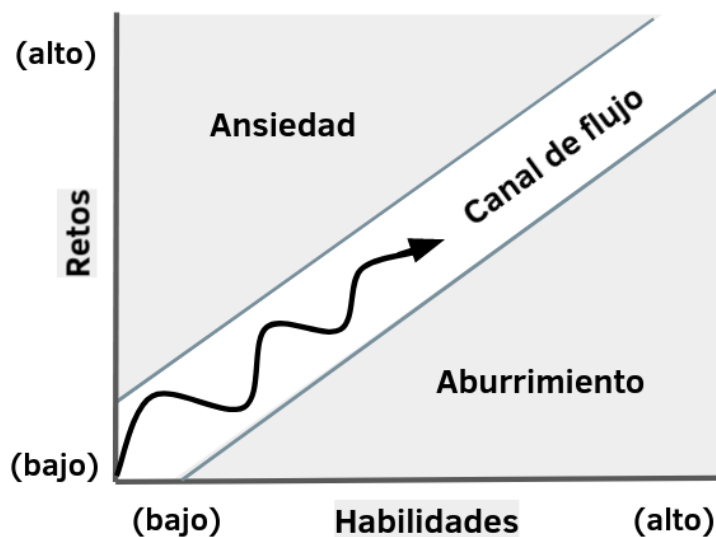


Fig. 1. Canal de flujo (adaptado) [14]

Este equilibrio es un *flujo de estado mental* de operación en el cual una persona está completamente sumergido y enfocado en lo que se encuentra haciendo. Esto implica un completo involucramiento y continuo compromiso en el proceso de la actividad. Es el estado ideal entre la ansiedad y el aburrimiento.

El *flujo* es impreciso y no siempre o fácilmente puede ser diseñado en un juego. Incluso es aún más difícil de probar el flujo porque no es sencillo para un diseñador de un juego entrar en un estado de flujo de su propio juego. Pero el flujo es algo que los diseñadores de juegos quieren que logren sus jugadores. Y, aunque un diseñador de juegos no pueda garantizar un estado de flujo para un jugador, el diseñador puede crear condiciones bajo las cuales el estado del flujo pueda ocurrir.

El flujo es experimentado cuando el desafío que enfrenta una persona está en casi perfecto equilibrio con las habilidades y las capacidades de la persona, así que puede lograr la tarea, pero necesitaría de concentración, bloquear las distracciones y un alto nivel de esfuerzo. Mihaly Csikszentmihalyi indica ocho componentes que hacen posible el flujo.

- *Tarea alcanzable*

La persona involucrada en la tarea debe creer que puede lograrla con un poco de esfuerzo; esto no significa que la tarea no sea difícil o desafiante; es justo lo contrario. Es difícil y, por lo tanto, requiere de un enorme esfuerzo de compromiso. Si la tarea es demasiado simple la persona se aburrirá y no entra en un estado de flujo; si la tarea es muy difícil la persona se frustrará y se dará por vencida. El término medio ideal es una tarea que parece realizable, pero requiere un gran esfuerzo para lograrlo.

- *Concentración*

Para que una persona entre en un estado de flujo, debe de aplicar energía mental y física con un enfoque intenso. Las distracciones desaparecen y las

acciones y los pensamientos de la persona juntos a la perfección para lograr la tarea.

- *Objetivos claros*

Significa que la persona conoce exactamente qué necesita hacer. Aquí no hay ambigüedad acerca de qué es lo que hay que cumplir o alcanzar. La única cuestión es *cómo* cumplir con la tarea, no *qué* es lo que se necesita cumplir.

- *Retroalimentación*

A medida que la persona se involucra en la actividad, la retroalimentación es proporcionada inmediatamente; toda acción provoca una reacción y la persona sabe si está haciendo las acciones correctas o las acciones equivocadas. La retroalimentación inmediata y continua ayuda a la persona a permanecer en el estado de flujo.

- *Participación sin esfuerzo*

A causa del alto nivel de concentración, el nivel de retroalimentación, y la habilidad para lograr los objetivos, la persona percibe que se involucra sin esfuerzo. Este caso sucede cuando la tarea está en un nivel de dificultad que desafía a la persona, pero el nivel de desafío está en la misma proporción que las habilidades y el esfuerzo que la persona es capaz de realizar. Cuando ésta se encuentra en el estado de flujo, no hay pensamientos externos que entran en la mente.

- *Control sobre la acción*

La persona se siente en completo control sobre lo que está haciendo y cree que las acciones tienen un efecto inmediato y resultados con propósito.

- *Preocupación por la desaparición de uno mismo*

Cuando la persona se encuentra en el estado de flujo está tan absorta de las actividades que está realizando que es en lo único que está pensando tanto que olvida otras actividades.

- *Pérdida del sentido del tiempo*

Cuando se está en el estado de flujo, el tiempo no tiene importancia. La persona se sienta para participar en la actividad y lo que pareciera que sólo pasan unos minutos son en realidad varias horas.

## 2.2 Gamificación para el aprendizaje

Un número de organizaciones está usando la gamificación para entrenar trabajadores, educar a estudiantes, resolver problemas y generar nuevas ideas y conceptos. Esos sectores de organizaciones van desde escuelas de negocios, compañías de software, compañías farmacéuticas, organizaciones gubernamentales y más. En casi cada industria es posible encontrar un ejemplo de la gamificación y el aprendizaje, motivación y la resolución de problemas.

Los profesionales del aprendizaje deben entender el crecimiento de la tendencia de aplicar sensibilidades basadas en juegos para el desarrollo de instrucciones a través de la creación de actividades basadas en el tiempo, incremento del nivel de experiencias de aprendizaje, narración, avatares y otras técnicas.

El crecimiento del uso de avatares, el incremento de popularidad de los juegos de rol masivo de multijugador en línea y la adición de sistema de puntos, insignias, y tablas de clasificación como en el ámbito económico, ventas al menudeo y finanzas son lo que lleva a una proliferación de técnicas de aprendizaje y colaboración gamificadas.

Esto no es una técnica menguante, más bien está ganando impulso y aceptación en más y más campos. Los profesionales del aprendizaje y el desarrollo deben seguir esa tendencia o quedarse atrás. Esto es especialmente cierto cuando se ha aplicado a áreas no típicamente consideradas como material apropiado para juegos.

Esto también es crucial porque los métodos tradicionales de aprendizaje están perdiendo preferencia. El tiempo y la atención de los alumnos es limitado y los profesionales del aprendizaje deben enfocarse en proveer una solución interesante y orientada a objetivos para el dilema de formación y enseñanza. Un enfoque en la gamificación incrementa el compromiso, la relevancia, la inmersión y asistencia con la transferencia de conocimiento a la situación actual. Educadores y profesionales del aprendizaje y el desarrollo están en una posición única para aprovechar la oportunidad de crear experiencias interactivas para empleados internos, estudiantes y clientes haciendo uso de la gamificación.

Establecer *conceptos claros*, *metas específicas* y un *conjunto de reglas* son áreas con las que muchos profesionales del aprendizaje se sienten cómodos y familiarizados. Adicionalmente, la idea de usar el tiempo como un motivador y proveer tanta retroalimentación como sea posible, también son técnicas comúnmente utilizadas. Las insignias, las recompensas y los niveles, así como la narrativa de historias, son áreas con las que los profesionales no están familiarizados y son buenas técnicas de las que se puede tomar prestado y ser aplicado en el área del aprendizaje.

El uso de estas técnicas junto con la narración de historias y permitiendo el fracaso o error puede crear poderosas herramientas de aprendizaje y aprovechar los elementos de la gamificación para crear resultados de aprendizaje eficaces, lo que da como resultado el incremento de retención y la aplicación del aprendizaje.

## 2.3 Discapacidad

La discapacidad se refiere a las dificultades que se presentan en cualquiera de las siguientes tres áreas de funcionamiento: *Deficiencias*: son problemas en la función corporal o alteraciones en la estructura corporal por ejemplo parálisis o ceguera; *Limitaciones de la actividad*: son dificultades para realizar actividades por ejemplo caminar o comer; *Restricciones de participación*: son problemas para participar en

cualquier ámbito de la vida diaria, por ejemplo, a la hora de conseguir empleo o transporte [15]

### 2.3.1 Clasificación de discapacidades

A continuación se muestra cómo la discapacidad se ha clasificado y los subtipos que la conforman con base en características de limitación [16]:

1 **Discapacidad motriz:** Personas que presentan discapacidades para caminar, manipular objetos y de coordinación de movimientos para realizar actividades de la vida cotidiana. También se incluye la pérdida total o parcial de uno o más dedos de las manos y los pies .

1.1 *Discapacidades de las extremidades inferiores, tronco, cuello y cabeza:* Personas que tienen limitaciones para moverse o caminar debido a la falta total o parcial de sus piernas. También aquellas que aún teniendo sus piernas no tienen movimientos o sus movimientos tienen limitaciones que provocan que no puedan desplazarse por sí mismos lo cual requiere ayuda ya sea de otra persona o de algún instrumento como silla de ruedas, andadera o prótesis.

1.2 *Discapacidades de las extremidades superiores:* Personas que, tienen limitaciones para utilizar sus brazos y manos por la pérdida total o parcial de ellos, o bien aun teniendo sus brazos y manos han perdido el movimiento, por lo que no pueden realizar actividades propias de la vida cotidiana tales como agarrar, abrir, cerrar, empujar, tirar o jalar (etc.) objetos con sus brazos y manos. Algunos ejemplos son: atrofia muscular de brazos y/o manos, defectos de huesos en las manos, acortamiento óseo entre otros.

1.3 *Insuficientemente especificadas del grupo de discapacidades motrices:* En esta clasificación discapacidades motrices que no se han logrado clasificar por su descripción con términos ambiguos por ejemplo parálisis.

**2 Discapacidad sensorial y de comunicación:** Discapacidad para ver, oír, hablar (mudez).

*2.1 Discapacidad para ver:* Cuando afecta a uno o los dos ojos la pérdida total de la visión, a la debilidad visual y a limitaciones que no pueden ser superadas con el uso de lentes. Algunos ejemplos de esta discapacidad son: ablepsia, afecciones de la retina entre otros.

*2.2 Discapacidad para oír:* Pérdida total de la audición en uno o ambos oídos, o con la pérdida parcial pero intensa, grave o severa en ambos oídos. En esta clasificación se incluyen personas sordomudas.

*2.3 Discapacidades de la comunicación y comprensión del lenguaje:* Incluye las discapacidades que se refieren a la incapacidad para generar, emitir y comprender mensajes del habla, limitaciones graves y severas del lenguaje que impiden la producción de mensajes claros y comprensibles.

**3 Discapacidad intelectual (retraso mental):** Discapacidades intelectuales que se manifiestan como retraso o deficiencia mental y pérdida de la memoria. Personas que presentan una capacidad intelectual inferior al promedio de las que tienen su edad, grado de estudios y nivel sociocultural. También se les dificulta realizar actividades cotidianas como asearse, desplazarse a sitios públicos, leer anuncios o instrucciones, sumar, escribir etc.

*3.1 Discapacidades conductuales y otras mentales:* Combinación de la pérdida de la memoria y depresión severa: En este grupo se encuentran las discapacidades moderadas a severas que se manifiestan en el comportamiento de las personas tanto en las actividades de la vida diaria como en la relación con otros. Estas personas pueden tener una interpretación y respuesta inadecuada a acontecimientos extremos. También pueden experimentar perturbación de la capacidad para identificar debidamente objetos y personas o a las dimensiones de espacio y tiempo.

**4 Discapacidades múltiples:** Este grupo agrupa a la población que manifiesta en forma conjunta más de una discapacidad

### 2.3.2 Discapacidad motriz

La discapacidad motriz constituye una alteración de la capacidad de movimiento que afecta en distinto nivel las funciones de desplazamiento, manipulación o respiración, y que limita a las personas en su desarrollo personal y social. Ocurre cuando hay una alteración en los músculos, huesos o articulaciones, o bien, cuando hay daño en el cerebro que afecta el área motriz y que le permite a la persona moverse de forma adecuada o realizar movimientos finos con precisión [17]. La discapacidad motriz se ha clasificado en los siguientes trastornos, respecto al origen del daño.

1. *Trastornos físicos periféricos.* Afectan huesos, articulaciones, extremidades y músculos. Se presentan desde el nacimiento (por ejemplo, algunas malformaciones de los huesos), o bien, son consecuencias de enfermedades en la infancia (como la tuberculosis ósea pulmonar). Algunos accidentes o lesiones en la espalda dañan la médula espinal e interrumpen la comunicación de las extremidades (brazos y piernas) hacia el cerebro y viceversa.
2. *Trastornos neurológicos.* Se denomina al daño originado en el área del cerebro en la corteza motora cerebral el cual se encarga de procesar y enviar la información de movimiento al resto del cuerpo. Origina dificultades en el movimiento, y en el uso, sensaciones y control de ciertas partes del cuerpo. Los más comunes son la parálisis cerebral, los traumatismos craneoencefálicos y los tumores localizados en el cerebro.

### 2.3.2.1 Métodos de acceso para personas con discapacidad motriz

Hay un número de soluciones que pueden mejorar de diferentes formas la experiencia en línea para un rango de usuarios con discapacidad motriz, como las que se mencionan a continuación [4]:

- **Navegación por teclado:** Algunos usuarios con discapacidad motriz frecuentemente sólo usan un teclado para interactuar con una computadora.
- **Estilos de enfoque:** Un elemento será enfocado cuando se haya tabulado en él o se interactúe con él.
- **Tabindex:** Si a un usuario no le es posible navegar e interactuar con el contenido solamente usando el teclado, usar el *tabindex* solucionará este problema. Se puede usar *tabindex* para asegurar que el contenido importante sea enfocado y ayudar al usuario a rápidamente a pasar el contenido que no requiere de interacción.
- **Capas accesibles:** Las capas como los elementos llamados modales, son características comunes en línea. Para los usuarios que navegan sólo usando el teclado pueden chocar con esta barrera cuando el contenido de la página es alterado por una acción del usuario, como usar un botón, pero es imposible de interactuar con el contenido que no es aparece. Se pueden hacer más accesibles: cambiando el enfoque cuando la capa aparece, haciendo uso de atajos de teclado y manteniendo el enfoque dentro del nuevo contenido. Es decir, permitir al usuario cerrar el nuevo contenido que está encima: con un botón de cerrar o cliqueando fuera del contenido del modal.
- **Gestos basados en punteros:** Algunos usuarios con discapacidad motriz prefieren navegar e interactuar con el contenido en la web usando el puntero,

Esto involucra apuntar y clicar con el mouse, pero a menudo se cansan rápidamente, son menos precisos o son propensos a cometer errores.

Es importante recordar que esos usuarios pueden no ser capaces de hacer movimientos certeros, como arrastrar y soltar o hacer zoom en una pantalla táctil. Se necesita evitar la suposición de que las interacciones siempre son transmitidas por la combinación del ratón y el teclado. La clave es reducir la necesidad de precisión en las acciones que se piden a los usuarios. Esto se puede lograr con:

- **Áreas grandes de interacción:** Revisar que las áreas donde los usuarios realizan acciones, como botones o casillas (checkboxes) sean suficientemente grandes y no requieren de un alto grado de destreza para operar.
- **Cancelación de puntero:** El usuario debería de poder cancelar la acción realizada dando click en algún elemento disponible a la vista.
- **Deshacer acciones:** El usuario debería de poder deshacer las acciones realizadas.
- **Voz a texto:** Voz a texto es otra forma común de navegación que es cada vez más usada, especialmente entre usuarios con dolor crónico o artritis.

Como se puede apreciar existen varios métodos que se pueden aplicar para mejorar la experiencia de usuarios con discapacidad motriz, algunos de estos métodos se consideraron para el diseño de la propuesta del software.

## 2.4 Herramientas de desarrollo utilizadas

### JavaScript

Javascript o también conocido como JS es distinguido por ser un lenguaje de scripting (secuencia de comandos) fácil de aprender, ligero, multi paradigma, basado en prototipos y acepta diferentes estilos de programación como lo son la programación orientada a objetos, imperativos y funcionales. Es ampliamente conocido y empleado para programar el comportamiento de las páginas web pero también es utilizado en entornos fuera del navegador como Node.js entre otros [18]. Javascript y Java son usualmente confundidos, ambos son marcas registradas de Oracle pero difieren en sintaxis, semántica y son empleados en diferentes situaciones [19].

## **React**

React hace que sea fácil crear interfaces de usuario interactivas. Al diseñar vistas simples para cada estado en la aplicación, React actualizará eficientemente y mostrará los componentes correctos cuando los datos cambien. Permite crear componentes encapsulados que son gestionados por su propio estado para luego ser integrados para crear interfaces de usuarios complejas. Ya que la lógica de los componentes son escritos en Javascript en lugar de plantillas, fácilmente se puede pasar datos a través de una aplicación y mantener el estado fuera del *DOM (Document Object Model* en español Modelo de Objetos del Documento). React no deja fuera el resto de tecnologías, así que se puede desarrollar nuevas características sin necesidad de volver a reescribir código existente [20].

## **AntDesign**

Librería de interfaz de usuario para React que contiene una colección de componentes de alta calidad y demos para construir interfaces de usuario interactivas [21].

## **Linux**

Linux es un sistema operativo de código abierto que se basa en los principios de Unix, diseñado y creado por Linus Torvalds en 1991. Algo que Linus empezó como un pasatiempo, se ha convertido en el sistema operativo más utilizado en los servidores de Internet disponibles públicamente, además de ser el único usado en las 500 supercomputadoras más rápidas a nivel mundial. Linux fue lanzado al público bajo la licencia GNU General Public License, esto quiere decir que cualquier persona puede ejecutar, estudiar, compartir y modificar el software. Asimismo, el código que se haya modificado se puede redistribuir, incluso vender, siempre y cuando sea bajo la misma licencia [22].

## **Python**

Es un lenguaje de programación orientado a objetos, con una sintaxis simple y fácil de aprender, esto hace a Python ideal para el desarrollo de prototipos y fácil de mantener. Cuenta con una amplia librería estándar que simplifica la integración contra diversos sistemas como servicios web, bases de datos, etc. Además de que funciona bien en los principales sistemas operativos del mercado [23], Python viene con una gran variedad de frameworks para aplicaciones web dependiendo de las necesidades de la aplicación web a desarrollar, por mencionar algunas son: Django, Flask, Bottle, Tornado CherryPy Hug [24].

## **Flask**

“Es un framework escrito en python que permite desarrollar fácilmente aplicaciones web. Fue diseñado para proveer la mínima funcionalidad para este tipo de aplicaciones y se permite extender mediante plugins” [25].

Flask es un framework web que provee de varias herramientas para el desarrollo de aplicaciones web. Flask es considerado como un micro-framework, lo que significa que no tiene dependencias de librerías externas. Este framework es ligero, flexible y fácil de aprender [24].

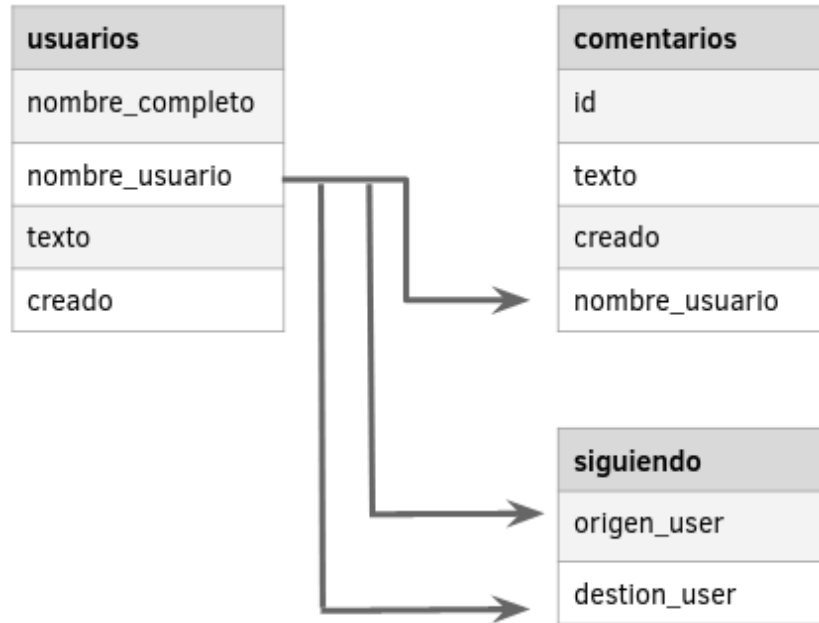
## **Base de datos**

Una base de datos es una colección organizada de información o datos estructurados. Una base de datos es usualmente controlada por un sistema de administración de base de datos o *database management system* (o por sus siglas en inglés DBMS) [26]. Los sistemas de administración permiten el acceso a la base de datos para almacenar, organizar, obtener y modificar datos de muchos usuarios, sin necesidad de preocuparse por su representación interna.

Hay diversos tipos de base de datos. La mejor base de datos para una organización en específico depende de cómo dicha organización pretende utilizar los datos. Para este proyecto se seleccionó el tipo de base de datos relacional.

- **Base de datos Relacional:**

Los elementos en una base de datos relacional están organizados como un conjunto de tablas con columnas y filas[26]. Las columnas tienen una entrada para datos de una categoría en específico y las filas contienen una instancia para ese dato definido de acuerdo a la categoría. En las bases de datos relacionales se utiliza el lenguaje SQL (Structured Query Language) para realizar (consultas) *queries* para solicitar información y/o manipular información que satisfaga ciertos criterios [27]. La tecnología de base de datos relacional provee de forma más eficiente y flexible de acceder a información estructurada [26].



**Fig. 2.** Ejemplo de un diagrama de base de datos relacional (adaptado) [28].

Para conocer más tipos de base de datos véase anexo 1

## POSTGRES

Es un sistema de bases de datos objetos-relacional de código abierto con más de treinta años de desarrollo activo. Utiliza y amplía el lenguaje SQL combinado con muchas características que de forma segura almacenan y escalan las cargas de trabajo de datos más complicados.

PostgreSQL ha ganado una sólida reputación por su arquitectura, integridad de datos, su conjunto sólido de funciones, extensibilidad y la dedicación de la comunidad de código abierto para ofrecer soluciones innovadoras y de alto rendimiento. PostgreSQL corre en la gran mayoría de los sistemas operativos. Cumple con *ACID* desde 2001 y

tiene potentes complementos, como el conocido extensor de base de datos geo-espacial PostGIS. PostgreSQL se ha convertido en la base de datos relacional de código abierto elegida por muchas personas y organizaciones [29].

El acrónimo *ACID* significa *atomic, consistent, isolated y durable* o su traducción al español sería atómica, consistente, aislado y durable respectivamente. Para garantizar un comportamiento predecible, todas las transacciones deben poseer estas propiedades básicas, lo que refuerza el rol de las transacciones críticas como propuestas de todas o ninguna [30].

*Atómica:* Una transacción debe ejecutarse exactamente una vez, ya sea que todo el trabajo se haya realizado o que ninguna de ellas. Las operaciones implicadas en una transacción suelen compartir una intención común y son interdependientes.

*Consistente:* Una transacción debe conservar la coherencia de los datos, transformando un estado coherente de los datos en otro estado coherente de los datos.

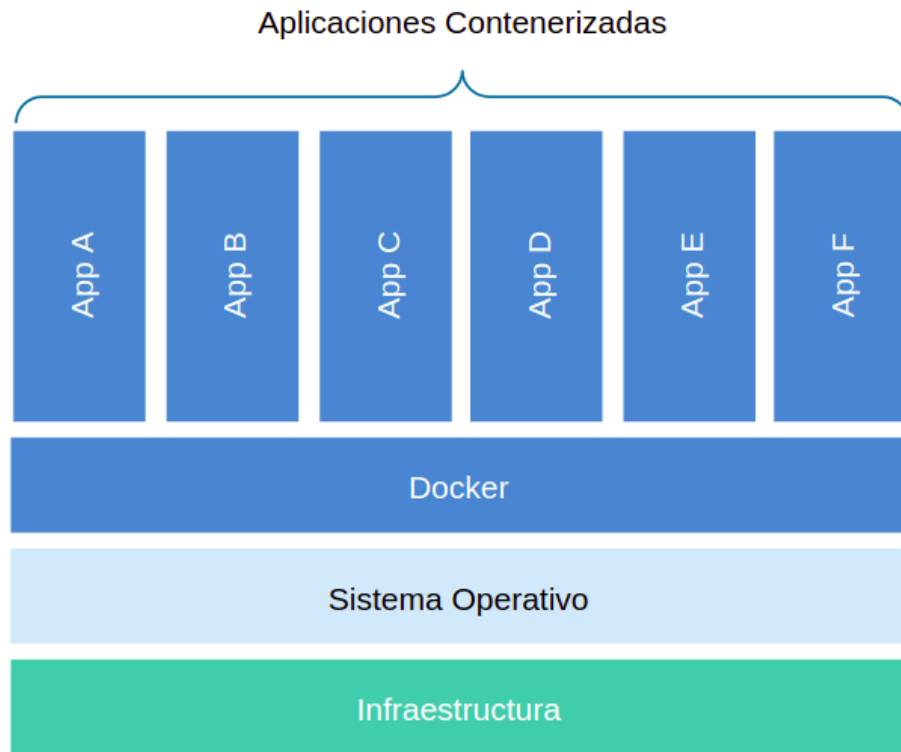
*Aislado*: Una transacción debe ser una unidad de aislamiento, lo que significa que las transacciones simultáneas deben comportarse como si cada una fuera la única transacción que se ejecuta en el sistema. Dado que un alto grado de aislamiento puede limitar el número de transacciones simultáneas, algunas aplicaciones reducen el nivel de aislamiento a cambio de un mejor rendimiento.

*Durable*: Una transacción debe ser recuperable y, por tanto, debe tener durabilidad. Si se confirma una transacción, el sistema garantiza que sus actualizaciones pueden persistir incluso si el equipo se bloquea inmediatamente después de la confirmación. El registro especializado permite que el procedimiento de reinicio del sistema complete las operaciones sin terminar que requiere la transacción, lo que hace que la transacción sea duradera.

## **Docker**

Desarrollar aplicaciones hoy en día requiere mucho más que escribir código. Múltiples lenguajes, frameworks, arquitecturas, y múltiples interfaces discontinuas entre herramientas para cada etapa del ciclo de vida crean enorme complejidad. Docker simplifica y acelera el flujo de trabajo, mientras provee a los desarrolladores la libertad de innovar con su elección de herramientas, pila de aplicaciones y entornos de aplicación para cada proyecto [31].

En 2013, Docker presentó lo que se convertiría en el estándar de la industria para contenedores. Los contenedores son una unidad estandarizada de software que permite a los desarrolladores aislar su aplicación de su entorno, resolviendo así el dolor de cabeza de “funciona en mi maquina”. Para millones de desarrolladores en la actualidad, Docker es el estándar por defecto para construir y compartir aplicaciones en contenedores, desde el escritorio hasta la nube [32].



**Fig. 3.** Ejemplo ilustrativo de la Ingeniería de funcionamiento de Docker (adaptado) [32]

## UML

Las grandes empresas deben estar estructuradas de una manera que permita la escalabilidad, seguridad y la ejecución robusta en condiciones de alta demanda y su estructura (o arquitectura) debe definirse con suficiente claridad para que los programadores de mantenimiento puedan rápidamente encontrar y corregir un error que aparece mucho después de que los autores originales hayan pasado a otros proyectos. Es decir, estos programas deben estar diseñados para funcionar a la perfección en muchas áreas [33].

El modelado es el diseño de aplicaciones de software previo a la codificación. El modelado es una parte esencial y útil de proyectos de software grandes, medianos y pequeños. Un modelo juega un rol de papel análogo en el desarrollo de software, como lo hacen los planos y/o maquetas en la construcción de un rascacielos. Al usar un

modelo, los responsables del proyecto pueden asegurarse que la funcionalidad del negocio sea completa y correcta, que satisfaga las necesidades del usuario final y que el diseño del programa admite los requisitos de estabilidad, robustez, seguridad, extensibilidad entre otras características previo a la implementación [33].

Algunas ventajas de UML son [34]:

- Simplificar la complejidad
- Ayuda a mantener abierta la comunicación
- Automatiza los procesos y la producción de software
- Ayuda a resolver problemas de arquitectura
- Incrementa la calidad del trabajo
- Disminuye los costos de desarrollo y los tiempos de comercialización.

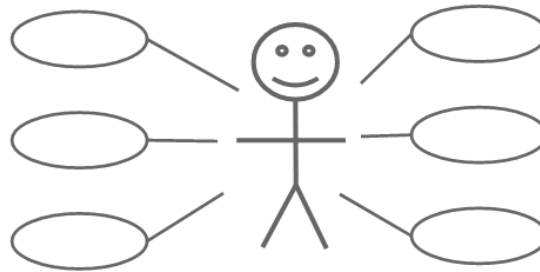
El Lenguaje Unificado de Modelado o UML desempeña un rol importante tanto en el desarrollo de software como también en muchas industrias que no tienen software, es una forma de visualizar el comportamiento y la estructura de un sistema o proceso [34].

El objetivo del UML es que exprese visualmente diagramas que sean fáciles de entender para todos. Es por eso que desde clientes, director de proyectos, técnicos, diseñadores, codificadores, encargados de pruebas y control de calidad utilizan un diagrama específico que se adapte a sus necesidades dependiendo del enfoque y nivel de detalle que se requiera [34]. Para este proyecto se utilizó el diagrama casos de uso con el fin de definir las actividades a realizar por el usuario, (para conocer otros tipos de diagramas véase anexó 2)

- **Diagrama de casos de uso**

“Éste describe las funciones del sistema, pero no la forma en que las hace. Un caso de uso es un conjunto de eventos que ocurren cuando un “actor” usa un sistema para completar un proceso. Un actor se define como cualquier persona

o cualquier cosa que interactúa con el sistema (persona, organización o aplicación) desde fuera del sistema. Por lo tanto, un diagrama de casos de uso describe visualmente ese conjunto de secuencias y representa los requisitos funcionales del sistema.”



**Fig. 4.** Ejemplo de diagrama UML básico de casos de uso (adaptado) [34]

## 2.5 Metodología de desarrollo

### 2.5.1 Metodología SCRUM

Scrum es un marco que permite el trabajo colaborativo entre equipos. Aunque son los equipos de desarrollo de software los que utilizan con mayor frecuencia este tipo de metodología, sus principios y lecciones se pueden aplicar a todo tipo de trabajo en equipo. Se basa en el aprendizaje continuo y en la adaptación a los factores cambiantes. Reconoce que el equipo no lo sabe todo al inicio de un proyecto y evolucionará a través de la experiencia. Scrum está estructurado para ayudar a los equipos a adaptarse de forma natural a las condiciones cambiantes y a los requisitos de los usuarios, con el cambio de prioridades integrado en el proceso y ciclos de lanzamiento breves para que tu equipo pueda aprender y mejorar constantemente [35].

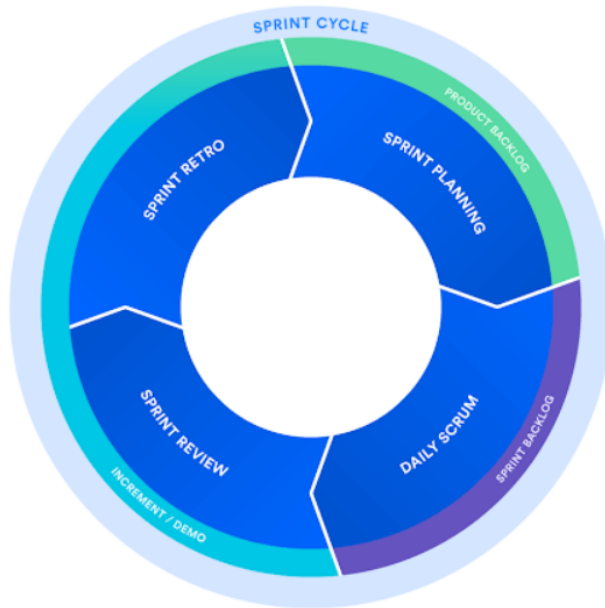


Fig. 5. Estructura SCRUM [35]

Aunque el scrum está estructurado, su ejecución se puede adaptar a las necesidades de cualquier organización. En scrum, se identifican tres artefactos para dar solución a un problema los cuales son un backlog del producto, un backlog de sprint y un incremento con tu definición de "hecho".

**Backlog del producto:** lista principal del trabajo de cosas por hacer del equipo que debe realizar el propietario del producto o el gestor de productos. El propietario del producto está constantemente revisando, cambiando las prioridades y realizando el mantenimiento del backlog del producto puesto que el mercado va cambiando y el conocimiento que se tiene del producto va incrementando, es posible que los problemas se solucionen de otras maneras o algunos elementos ya no sean relevantes.

**Backlog de sprint:** lista de elementos, historias de usuario, o corrección de errores seleccionadas por el equipo de desarrollo, para su implementación en el ciclo actual de sprint

**Incremento:** Es el objetivo o bien el producto final utilizable de un sprint.

## **Protocolos o eventos de SCRUM**

**Organización del backlog:** se conoce como limpieza del backlog. Es responsabilidad del propietario del producto mantener al día la lista utilizando los comentarios de los usuarios y del equipo de desarrollo para priorizar tareas y mantenerla limpia y preparada para trabajar sobre ella.

**Planificación de sprint:** reunión en la cual todo el equipo de desarrollo planifica el trabajo que se va a realizar (alcance) durante el sprint actual. Al final de la reunión de planificación, cada miembro del scrum debe tener claro qué se puede entregar en el sprint y cómo se puede entregar el incremento.

**Sprint:** un sprint es el periodo real en que el equipo de scrum trabaja de forma conjunta para finalizar un incremento. La duración de un sprint suele ser de dos semanas, aunque algunos equipos manifiestan que les resulta más fácil una semana para el alcance o un mes para entregar un incremento valioso. Dave West, de Scrum.org, recomienda que cuanto más complejo sea el trabajo y más incógnitas haya, más corto debería ser el sprint.

**Scrum diario o reunión rápida:** reunión diaria de muy corta duración que tiene lugar siempre a la misma hora normalmente, por las mañanas y en el mismo sitio para facilitar las cosas. La reunión rápida es el momento de expresar cualquier inquietud que se tenga con el cumplimiento del objetivo del sprint o de notificar los impedimentos existentes.

Una forma habitual de realizar la reunión rápida es que cada miembro del equipo responda tres preguntas en el contexto de alcanzar el objetivo del sprint:

¿Qué hice ayer?, ¿Qué tengo planificado para hoy?, ¿Hay algún obstáculo?

**Revisión de sprint:** al final del sprint, el equipo se reúne en una sesión informal para ver una demostración o inspeccionar el incremento. El equipo de desarrollo muestra los elementos del backlog que ahora están "finalizados" a las partes interesadas y a los compañeros de equipo para recibir comentarios.

**Retrospectiva de sprint:** la retrospectiva es donde el equipo se reúne para documentar y analizar qué ha funcionado y qué no ha funcionado en un sprint, un proyecto, en las personas o relaciones, herramientas o incluso para determinados protocolos. La idea es crear un lugar donde el equipo pueda centrarse primordialmente en lo que salió bien y en lo que debe mejorarse para la próxima vez, y menos en lo que salió mal.

El éxito de scrum en el desarrollo de productos de software y hardware en distintos sectores y mercados lo convierten en un marco de trabajo conveniente para ser adoptado a la organización [35]. Los equipos de desarrollo de software han demostrado que la implementación de marcos ágiles, como el scrum permite ofrecer soluciones a los clientes de forma más rápida y con mayor previsibilidad; además, les da la capacidad de reaccionar con rapidez gracias a la información nueva. La implementación de una metodología ágil a nivel de equipo individual resulta relativamente sencilla. No obstante, el verdadero desafío se encuentra en querer llevar la implementación de scrum a varios equipos de una organización grande [36].

Así mismo Horquin [37] ha demostrado en su investigación la posibilidad de la aplicación de scrum en equipos unipersonales. Dicho autor ha desarrollado un software de manera individual que concluyó de manera satisfactoria. Horquin con el uso de scrum que ha adaptado de acuerdo a las necesidades del proyecto, le ofreció un proceso flexible, adaptable y con retroalimentación constante, por ser la única persona a cargo del proyecto ha logrado entregar el software de manera continua al finalizar cada sprint y por lo tanto ha conseguido concluir con el software.

# Capítulo 3 Estado del arte

## 3.1 Sistemas y herramientas desarrolladas para aprender programación empleando la gamificación

El objetivo del trabajo de Campaña et al. [1] fue desarrollar una metodología para la enseñanza de iniciación a la programación basada en el uso de metodologías activas como actividades de programación en parejas, introducción entre pares, computación multimedia, problemas resueltos, aprendizajes basados en proyectos, ejercicios de trazas de código, pensamiento computacional y programación y programación visual, todo esto englobado con el uso de la gamificación donde al ir completando dichas actividades los alumnos van obteniendo puntos con el fin de aumentar la participación y la motivación. Esta implantación se aplicó en la titulación de Ingeniería Informática en la asignatura de Fundamentos de programación de la Universidad de Granada, España. Para observar la evolución de la percepción de la programación por parte del alumnado a lo largo del curso y sondear las actividades planteadas, se desarrolló una encuesta tipo Likert. Concluyen que las actividades mejor valoradas son los ejercicios resueltos, la programación en parejas, trazas de código, programación lúdica (videojuegos) y los proyectos prácticos, así como también se vio una mejora en la motivación de los alumnos. Los resultados de este trabajo nos dan la pauta para observar al uso de videojuegos como camino viable para el aprendizaje de la programación [1]

Para la asignatura Programación I en unidad de fundamentos de Java Universidad de Extremadura, España, Beltran [6] trabajó sobre la plataforma e-learning de aprendizaje Moodle, donde estructuró y empleó estrategias de gamificación para la realización de tareas que el alumnado deberá resolver de manera autónoma. Esto con el fin de proponer una solución al problema que enfrentan los catedráticos para conseguir que los estudiantes se motiven y se sientan comprometidos durante el proceso de aprendizaje de la materia y conjuntamente promover la sana competencia entre ellos.

El contenido del curso fue distribuido en cuatro capítulos: Introducción a la Programación Orientado a Objetos, fundamentos de Java, Arreglos y matrices y por último métodos.

Beltran [6] menciona que utilizar la plataforma Moodle que ha sido gamificado por él es una herramienta efectiva para mejorar el proceso de aprendizaje de la asignatura de Programación I, de igual manera, es una herramienta para los profesores donde podrán controlar y dar seguimiento a los alumnos de una forma efectiva [6].

Astudillo, Bast y Willging [7] diseñaron un taller aplicando la gamificación sobre la plataforma Moodle donde se pusieron a disposición juegos serios enfocados en la programación para la introducción de conceptos de algoritmo, variables y estructuras de control. Dicho taller diseñado por Astudillo, Bast y Willging [7] se encuentra compuesto por cuatro niveles. En el primer nivel emplearon el juego llamado *Lightbot 2.0* ; el objetivo de este nivel es hacer que los alumnos comprendan el concepto de algoritmo e instrucción. Para el segundo nivel, eligieron Code.org con el fin de tener una primera idea de estructuras de control. En los niveles tres y cuatro emplearon “Scratch” para brindar nociones de variable, estructuras de control y resolver problemas a través de la programación.

En conclusión, se pudo observar una mayor participación de los estudiantes en comparación con las versiones previas del taller de Introducción a la Programación, conjuntamente el 70% de los estudiantes que participaron en este taller de introducción en su calificación final obtuvieron un promedio arriba de siete y, por lo tanto, prueba un dominio aceptable de los conceptos [7].

El objetivo de la investigación realizada por el autor Corsi [3] fue evaluar el impacto que el uso de estrategias lúdicas en la enseñanza y aprendizaje de la programación tiene en los siguientes puntos principales: adquisición de conocimientos, relaciones interpersonales y la inteligencia emocional de los estudiantes. El método fue el siguiente: Se implementaron tres tratamientos.

Primer tratamiento, *Aprendizaje mediante el desarrollo de videojuegos*: a los estudiantes del curso se les pidió organizarse en ocho equipos y desarrollar un

videojuego para aprender conceptos de programación orientado a objetos.

Segundo tratamiento, *Aprendizaje mediante la utilización de videojuegos*: al grupo del siguiente periodo se le entregó los juegos desarrollados por el grupo del año previo.

Y el tercer tratamiento, *Ludificación del aprendizaje* o bien *Gamificación en el aula*: un año después los estudiantes del curso compitieron durante cuatro semanas, donde resolvían ejercicios (del libro OCA Java SE 8 Programmer 1 Exam Guide publicado en 2017) dentro y fuera del horario de clases y se otorgaban insignias y puntos.

Al analizar los pre y post test que se aplicaron en cada uno de los tratamientos el autor concluye que la estrategia de utilización de videojuegos produjo resultados que destacaron sobre los otros tratamientos [3].

## 3.2 Videojuegos enfocados al aprendizaje de la Programación Orientada a Objetos

Existe una diversidad de herramientas para facilitar el aprendizaje de manera lúdica de la programación orientada a objetos, algunas de ellas se describen a continuación [8].

**Scratch**: tiene la dinámica de arrastrar y soltar elementos con el ratón y modificar valores numéricos, textos con el teclado con la intención de animar objetos que se muestran en pantalla.

**GreenFoot**: Es una herramienta didáctica para el aprendizaje de programación utilizando el lenguaje Java. Básicamente es un editor para desarrollar código con un visor de clases.

**Alice**: Es un ambiente para programar juegos interactivos o animaciones de objetos 3D como humanos, animales y objetos de fantasía que se encuentran en diferentes escenarios o mundos.

**Robocode**: Es un entorno de simulación de guerras, donde hay que combatir contra los tanques de otros jugadores programando en lenguaje Java un tanque propio.

**Jeroo**: Herramienta educativa para la introducción a la programación orientada a objetos provee de una interfaz con resaltado de código fuente, animaciones básicas y un panel de status.

**RoboMind:** Entorno de desarrollo el cual mediante la escritura de programas donde se podrá ingresar órdenes para hacer que el robot que se encuentra en el escenario para que se mueva o realice alguna acción, conjuntamente este entorno provee de un panel donde se muestran mensajes de error [8].

En su trabajo, Tabares [9] propone un método para el aprendizaje de los fundamentos de programación orientada a objetos ligado al uso del juego digital, con el fin de aumentar la motivación de aprendizaje en personas que inician sus estudios en ingeniería en sistemas y programas similares.

Tabares [9] desarrolló un prototipo de juego para el aprendizaje de conceptos fundamentales del paradigma de la programación orientada a objetos: clase, objeto, herencia y polimorfismo. El juego se conforma de tres niveles. Nivel 1 llamado *Jugando con clases* se abordan los métodos constructores, modificadores y analizadores. El nivel 2 está compuesto con actividades para que el jugador comprenda los conceptos de herencia y polimorfismo. El último nivel lleva por nombre *Resolviendo laberintos*, consiste en ejecutar los métodos avanzar y girar para salir del laberinto. Además, se muestra un mensaje en cada ocasión que el jugador se equivoca o acierta.

Los resultados se realizaron por medio de rúbricas diligenciadas por profesores y por encuestas aplicadas a estudiantes.

Las pruebas se aplicaron a cuatro grupos de estudiantes, dos del programa de Ingeniería en Software de la Institución universitaria y dos del décimo grado de secundaria media técnica en informática. A cada grupo de experimental (un grupo de cada nivel de estudios) se aplicó el método llamado *MIPSOO* (Método Integrado de Programación Secuencial y Programación Orientado a Objetos) para la enseñanza y el juego *CoquitoDobleO* como motivador.

Al finalizar el curso, se aplicó a los grupos experimentales (los que hicieron uso del juego) y de control (los que prescindieron del juego) una encuesta con ocho incisos con escala tipo Likert y dos preguntas abiertas. Al analizar los resultados, el autor concluye lo positivo que es incluir juegos digitales de forma gradual y permanente durante los cursos y evaluación de asignaturas referentes al aprendizaje de programación. Además, los profesores encargados de los grupos evaluaron los

conocimientos de los participantes, donde se mostró que los promedios de los alumnos de los grupos experimentales fueron más altos [9].

### 3.3 Herramienta de aprendizaje para personas con capacidades diferentes

Se han desarrollado e implementado diversos juegos serios para el aprendizaje de los fundamentos de la programación; sin embargo, no se menciona que estos recursos sean accesibles para usuarios con discapacidad motriz, por lo que López y Sánchez [5] sugieren una herramienta llamada Scratch que ayude al aprendizaje de los fundamentos de la programación de forma visual pues no hay que escribir líneas de código; ésta puede ser usada para personas con diferentes capacidades, entre ellas, la discapacidad motriz. Scratch puede ser accesible por medio de *rampas digitales*, es decir, haciendo uso de un *mouse-key* y un *teclado numérico* [5], por ejemplo.

En su artículo, Lancheros, Carrillo y Lara [10] diseñaron un sistema que incorpora una interfaz adaptativa personalizada, utilizando el perfil de los estudiantes para brindar recursos educativos, que permiten proporcionar información de contenido ajustado al estilo de aprendizaje y a la característica de la discapacidad. Los cambios que se aplicaron a la interfaz personalizada fueron aspectos como: color y tamaño de texto, idioma, nivel de volumen auditivo y distribución del contenido. Las características recopiladas en el perfil del estudiante (nombre y fecha de nacimiento, aspectos académicos ligados a su historia académica, datos relacionados con preferencias visuales y de contenido, y aspectos psicopedagógicos, como el estilo de aprendizaje) permiten describir satisfactoriamente los gustos y necesidades, lo cual facilita la adaptación de la información contenida en la interfaz gráfica. Con el objetivo de facilitar el proceso de aprendizaje dentro de la misma aula clases para alumnos con y sin discapacidad auditiva. A pesar de que aún no se tienen resultados de la investigación realizada del trabajo de [10] sugieren también validar el modelo con otro tipo de discapacidad para determinar la aceptación del sistema.

Con base en los trabajos de los autores Campaña et al. [1], Beltran [6] y Astudillo y Willging [7], se implementó el desarrollo de herramientas para la enseñanza de fundamentos de la programación. Los autores Botero [8] y Botero [9] se enfocaron en el desarrollo de herramientas para el aprendizaje de la programación orientado a objetos; sin embargo, en ninguno de estos trabajos se menciona que puedan ser usables para alumnos con discapacidad motriz fina. Por lo que López y Sánchez [5] sugieren la herramienta *Scratch* como un recurso para aprender conceptos básicos de programación de forma visual, que puede ser de utilidad para usuarios con esta característica por medio de rampas digitales por ejemplo un mouse-key y un teclado numérico. Al hacer la revisión de los trabajos previos y las herramientas desarrolladas para el aprendizaje básico y especialización de la programación orientada a objetos, nos da la certeza que existe una escasez de recursos, para el aprendizaje de este tema, que integren características de usabilidad para usuarios con discapacidad motriz. Según los resultados obtenidos de la investigación de comparación de Corsi [3], el impacto del uso de estrategias lúdicas en la enseñanza y aprendizaje de la programación, el autor concluye que la estrategia de utilización de videojuegos produjo resultados que destacaron sobre los otros tratamientos que puso a prueba. Por lo tanto, este trabajo de estudio tiene el objetivo de implementar un recurso de apoyo para el aprendizaje de programación orientado a objetos haciendo uso de la gamificación (debido a los resultados contenidos de Corsi [3]) con una interfaz que tenga en cuenta las necesidades de los usuarios con discapacidad motriz con limitaciones en las extremidades superiores. Lo cual nos hace suponer que este trabajo puede tener un aporte significativo.

## Capítulo 4 Diseño y desarrollo del Sistema

## 4.1 Requerimientos generales

Con base en el objetivo del proyecto, se definieron los requerimientos y se generaron diagramas para establecer prioridades, definir las actividades con el fin de establecer el orden de desarrollo.

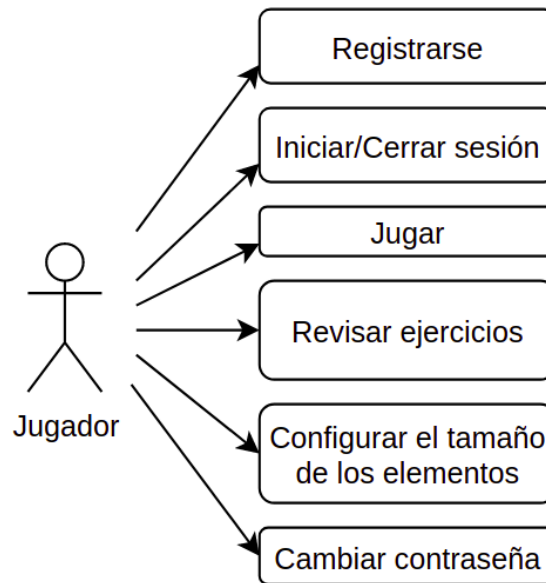
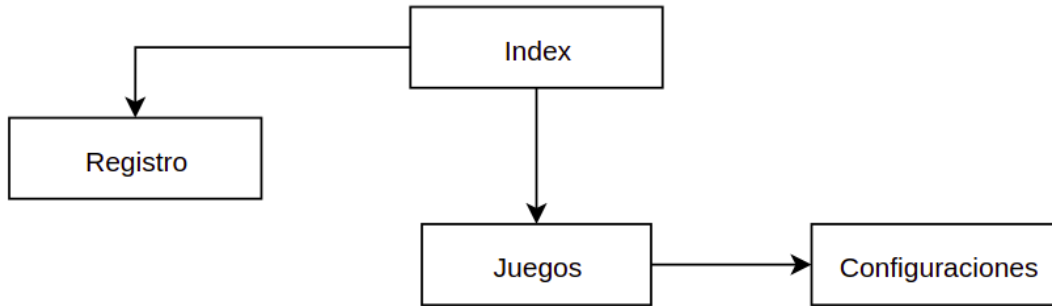


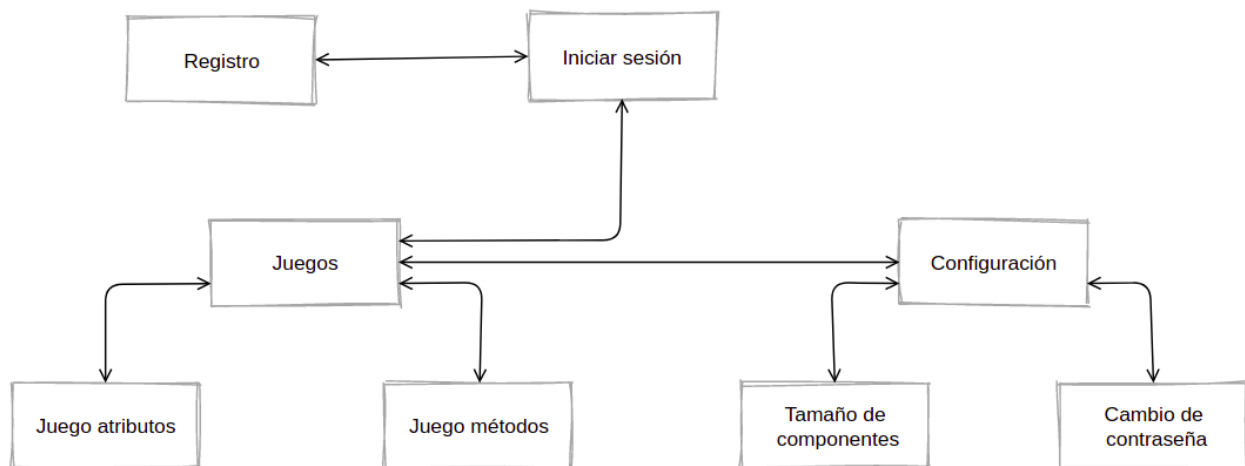
Fig. 6. Diagrama de casos de uso del sistema propuesto

Previo a cualquier tipo de desarrollo, es importante primero conocer las actividades que realizará el usuario dentro del sistema web [33]. Entre las acciones que podrá realizar el jugador será: (ver figura 6) poder registrarse para poder tener una cuenta y poder iniciar y cerrar sesión en el sitio. También, podrá interactuar con las actividades propuestas en el sitio y, a su vez, revisar si acertó o no en dichas actividades. Debido a que se pretende que este sitio sea accesible, adicionalmente podrá cambiar el tamaño de ciertos elementos que se encuentran en el sitio para su fácil interacción y, por último, también podrá cambiar su contraseña con la cual inicia sesión.



**Fig. 7.** Mapa del sitio web del sistema propuesto

Una vez conociendo las actividades que el usuario podrá realizar en el sitio web, se propone tener la siguiente estructura como se muestra en la figura 7. En la página index, los usuarios nuevos podrán crear su cuenta, o bien, iniciar sesión en caso de que ya cuente con una; una vez dentro serán redirigidos a la página de juegos, o en su caso, podrán ir a la página de configuraciones para cambiar su contraseña o el tamaño de los componentes.



**Fig. 8.** Diseño navegacional del sistema web propuesto

El usuario primero tendrá que registrarse en el sistema; una vez creada la cuenta, podrá iniciar sesión y entrar al sitio como se muestra en la figura 8. Una vez dentro, será directamente redirigido a la página donde podrá seleccionar uno de los dos juegos disponibles. Es importante mencionar que al inicio solo se encuentra disponible el juego de atributos, ya que para poder comprender los conceptos de “métodos” se necesita

primero conocer las definiciones del primer apartado.

En caso de que el jugador decida no jugar, podrá ir a la sección de configuraciones en donde se muestran las opciones de cambiar el tamaño de algunos elementos con los que interactuar frecuentemente como los botones, asimismo, cambiar la contraseña que proporcionó en el momento de su registro en la plataforma si así lo desea.



The image shows a web browser window with a single tab titled 'login'. The address bar contains the URL 'https://www.gamepoo.duckdns.org'. The page content is centered and includes a login form. The form has two input fields: the first is labeled 'matricula:' and contains the number '123456789'; the second is labeled 'contraseña:' and contains seven asterisks. Below these fields is a prominent blue button with the text 'Iniciar sesion'. Underneath the button is a blue text link that says 'Registrate'.

**Fig. 9.** Diseño de presentación: Inicio de sesión

La página de inicio de sesión se compondrá de un formulario en donde el usuario que ya tiene una cuenta creada tendrá que ingresar su matrícula de nueve dígitos numéricos, así como su contraseña. Si el usuario aún no tiene cuenta, podrá darse de alta dando clic en un link *Registrate*, como se muestra en la figura 9.

Una vez de haber dado clic en el link de *Registrate*, se abrirá un modal con un formulario donde al usuario se le pedirá que ingrese los datos de nombre, matrícula, correo y la contraseña en dos ocasiones para validar que ésta se ha escrito correctamente como se muestra en la figura 10.

The image shows a web browser window with a single tab titled 'login'. The address bar contains the URL 'https://www.gamepoo.duckdns.org'. The main content area displays a registration form titled 'Registrarse'. The form includes the following fields and controls:

- Nombre:** Input field containing 'lorem'.
- Matricula:** Input field containing '123456789'.
- Correo:** Input field containing 'lorem@correo.com'.
- Contraseña:** Input field containing '\*\*\*\*\*'.
- Repetir Contraseña:** Input field containing '\*\*\*\*\*'.
- Buttons:** A 'Cancelar' button and a blue 'Enviar' button.

**Fig. 10.** Diseño de presentación: Registro de un nuevo usuario

Cuando el usuario ingrese su matrícula y contraseña de forma correcta, se iniciará la sesión y será redirigido a la página de juegos para que empiece a realizar las actividades propuestas que se encuentran listadas dentro de esta sección como se muestra en la página de juegos (figura 11). Cada juego se compondrá de una carátula con el nombre del juego y su rango de calificación.



Fig. 11. Diseño de presentación: Contenido de la página de juegos.

Para visualizar la página de *Configuración*, el usuario tendrá que dar clic en esta opción en la barra lateral. El contenido de esta página se compone de dos secciones: la primera de ellas (y además se visualiza por defecto al visualizar la página de configuración) es la sección para cambiar el tamaño de algunos elementos; aquí, el usuario podrá dar clic en las opciones disponibles: pequeño, mediano y grande y los elementos como se muestra en la figura 12, en este caso, botones, cambian de inmediato al tamaño elegido; una vez que se ha elegido el tamaño deseado procederá a dar clic en el botón *guardar* para mantener esta configuración.

Para actualizar la contraseña, el usuario tendrá que dar clic en la barra superior en la opción *contraseña* como se muestra en la figura 13. De inmediato cambiará el contenido por un formulario en donde se ingresará la nueva contraseña y luego dar clic en el botón *guardar*. Cuando el usuario intente iniciar sesión nuevamente en el sitio web, deberá ingresar la nueva contraseña.



Fig. 12. Diseño de presentación: Contenido de la página de Configuración, Sección de cambio de tamaño



Fig 13. Diseño de presentación: Contenido de la página de Configuración, Sección de cambio de contraseña

La posición y el diseño de la barra lateral se inspiró tomando en cuenta el diseño del componente *sidebar* de *react-rainbow* que tiene una colección de varios componentes enfocados a mejorar la accesibilidad para usuarios con discapacidad [38]. El tamaño de cada opción del menú es amplio en donde se puede dar un click. Para acceder a los juegos, se eligió un componente tipo tarjeta que tiene una gran área de interacción. Hay que recordar que incrementar las áreas en donde el usuario tendrá que dar click es una mejora de usabilidad [4]. Por último, se diseñó una interfaz basada en un reconocedor de voz que el usuario podrá usar para navegar por el sitio y contestar los juegos propuestos. En la figura 14, se muestra la barra lateral con los menús con un tamaño amplio, la elección de tarjetas y en la parte superior se encuentran los botones para activar o desactivar el reconocedor de voz que ayudará al usuario a navegar por el sitio.



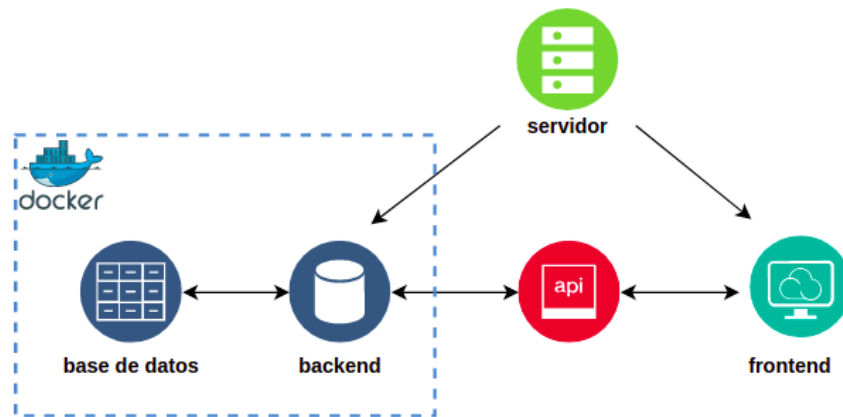
Fig. 14. Esquema para la vista de juegos.

Adicional se creó una sección de configuración donde se puede cambiar el tamaño de algunos componentes como los botones. Las opciones disponibles son pequeño,

mediano y grande este último es el pre definido por el sitio como se muestra en la figura 12.

## 4.2 Infraestructura del sistema

Desarrollar aplicaciones hoy en día requiere mucho más que escribir código. Múltiples lenguajes, frameworks, arquitecturas, y múltiples interfaces discontinuas entre herramientas para cada etapa del ciclo de vida crean enorme complejidad [31]. Con base en estos requerimientos se seleccionaron las tecnologías más adecuadas a emplear.



**Fig. 15.** Infraestructura del sistema propuesto en este trabajo.

Todo el proyecto fue desarrollado en sistema Linux distribución Solus 4.2, en éste se desarrolló tanto el *frontend* como el *backend* y se implementó con Postgres [29] una base de datos para guardar las configuraciones e información de los usuarios.

Para la implementación del backend se decidió utilizar el lenguaje de programación multipropósito de Python [23] en su versión 3.6. Para la integración del frontend con el backend se desarrolló una API con el framework Flask debido a que no requiere de grandes recursos de cómputo y el desarrollo es ágil. Se utilizó la extensión de Flask-SQLAlchemy para definir la estructura y la relación entre tablas de la base de datos, adicionalmente nos permite abstraer las consultas que se hacen a la base de

datos y nos ofrece compatibilidad con los principales manejadores de bases de datos, en este proyecto el que se empleó fue Postgres [29].

En el caso del frontend se optó por la librería React [20], una biblioteca de Javascript para construir interfaces de usuario y, para la implementación de la interfaz y la experiencia de usuario, se decidió por el kit de interfaz de usuario Ant design; una librería que contiene componentes para React.

Durante la investigación y desarrollo del sistema se consultaron diversas librerías de desarrollo que proveyeron de las funcionalidades que se adaptan con los objetivos del proyecto. Al final de la investigación se optó por hacer uso de la librería ant design puesto que es una de las librerías que cuenta con mayor soporte dentro de la comunidad web además de proveer múltiples widgets y a la vez éstos tienen características de accesibilidad.

Con el objetivo de realizar pruebas con los usuarios, el proyecto se puso en producción en un servidor con el sistema operativo Linux con distribución Ubuntu 18.04. Y para la puesta en marcha del proyecto los componentes fueron puestos en un contenedor Docker [31], [32] para su fácil manejo y desarrollo como se muestra en la figura 10.

El software fue desarrollado aplicando la metodología scrum pues la implementación es viable tanto para equipos individuales como para equipos de organizaciones grandes [36] , se decidió utilizar esta metodología donde se tienen múltiples etapas de diseño, desarrollo, testeo, liberación de las características de la etapa actual, el lanzamiento de ésta y su revisión de este pequeño entregable, y una vez que se hace el último ciclo o sprint lo más probable es que el proyecto sea funcional.

Otro motivo por el cual se utilizó esta metodología además de poder ser adaptada a las necesidades de la organización [36][37] es porque en cada iteración se hace una revisión del entregable funcional, de esa forma se puede detectar si las características desarrolladas cumplen con los objetivos del proyecto.

Los entregables tenían un tiempo de desarrollo de dos semanas. Se hacía una revisión del entregable con el fin de obtener retroalimentación y mejorar la calidad del producto en las siguientes iteraciones. Al final de cada sprint se definían los objetivos a entregar del sprint siguiente.

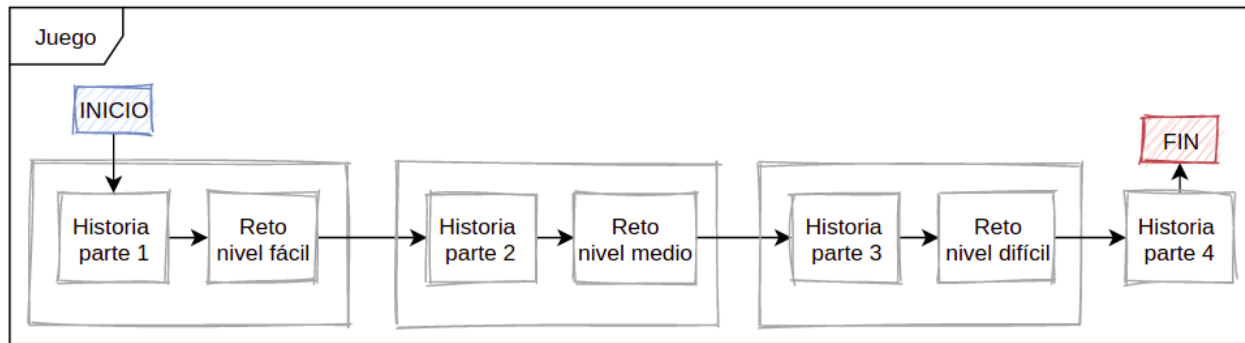
### 4.3 Diseño de juegos

Como se puede observar en la figura 6 una de las actividades del usuario es jugar un juego con el fin de aprender conceptos de la programación orientada a objetos. Para asegurar la efectividad de un juego serio, Norero define cinco elementos básicos: historia, dinámica de interacción, retroalimentación, simulación y aprendizaje [11].

Para este proyecto se elaboró una historieta que relaciona cómo la programación puede resolver situaciones de la vida cotidiana. Al mismo tiempo, se dan definiciones de los conceptos de la programación orientada a objetos. De esta forma, se aplicaron los elementos de *historia, simulación y aprendizaje*.

Un juego está compuesto de historias y retos que se van intercalando como se muestra en la figura 3. La *dinámica de interacción* es la siguiente: se inicia con una historia para introducir al jugador con los personajes, entorno y concepto; a aprender, seguido de un reto con el cual se pone a prueba el conocimiento adquirido en la historia.

Como *retroalimentación*, se implementaron notificaciones con mensajes alusivos a las respuestas que el jugador haya respondido en los retos, asimismo, una barra de progreso con el fin de mostrar la etapa en la se encuentra actualmente.



**Fig. 16.** Estructura de todos los juegos propuestos en este proyecto.

Tomando en cuenta que el juego es para usuarios que se introducen en el área de programación, la dificultad va ascendiendo, pues Mihaly Csikszentmihalyi sugiere que debe de haber un equilibrio entre las competencias, metas conseguidas y tiempo de aprendizaje [14].

Cuando el usuario da clic en uno de los juegos disponibles como se muestra en la figura 17, será visible el contenido de éste. Es importante mencionar que se muestran todos los juegos disponibles esto con el fin de alimentar la curiosidad del jugador y despertar su motivación a concluir con todos retos disponibles. Podemos notar esta característica en otros juegos como los juegos de consola como el famoso *Mario bros* incluso en los juegos para dispositivos móviles como *Candy crush* entre otros.

Como se mencionó previamente (ver figura 16), el juego inicia con el inicio de la historia (ver figura 18), con el fin de que el jugador se identifique con el personaje principal. En la historia también se tiene el objetivo de proveer de conceptos de la programación orientada a objetos.



**Fig. 17.** Contenido de la página de juegos. Representación de selección de un juego

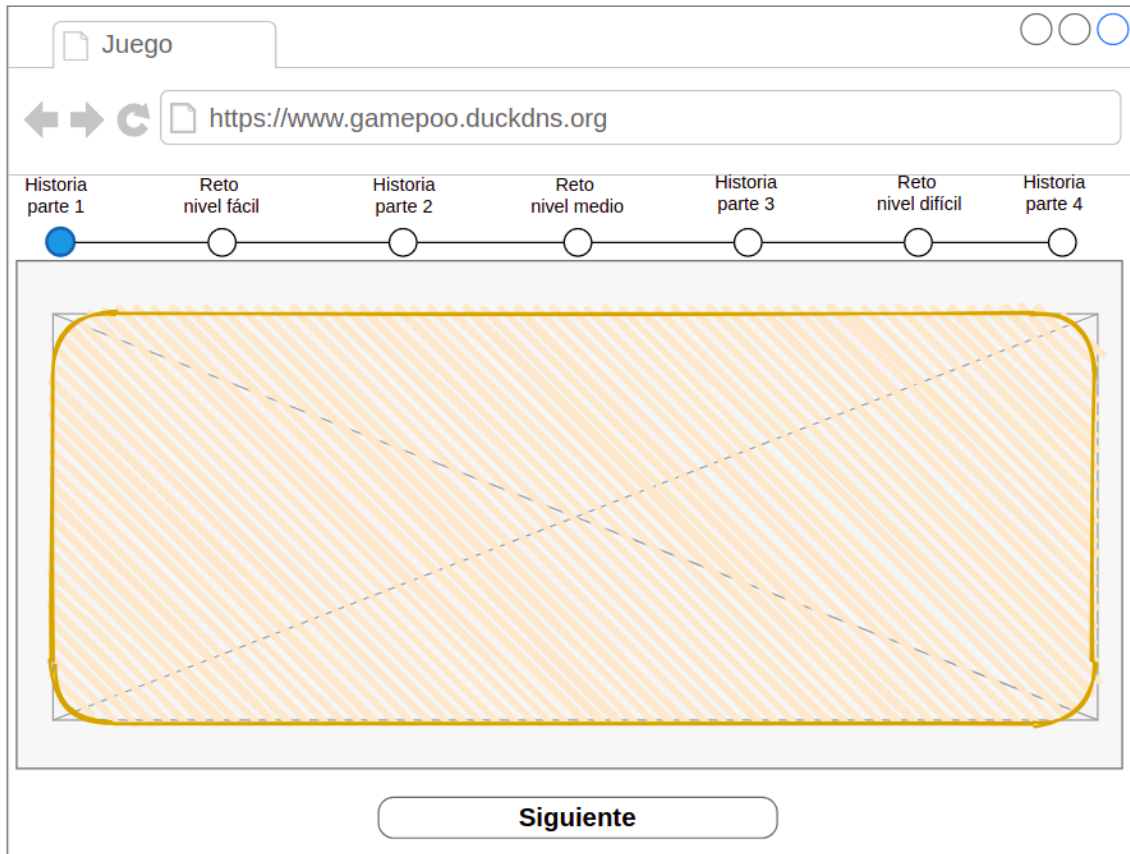


Fig. 18. Contenido de un juego parte inicial, presentación de historia.

En la figura 19, se muestra la estructura básica del contenido del juego; en la parte superior se muestra una barra de progreso para que el jugador visualice en qué parte del juego se encuentra. En la parte inferior se encuentran dos botones: uno para regresar a la sección anterior y otro para continuar. En la parte inferior derecha se podrán encontrar dos botones; el botón *calificar* es para enviar las respuestas del jugador y el botón con el signo de interrogación hará visible un modal donde se muestran los comandos de voz programados, entre los cuales se encuentran: *Revisar respuesta*, *Calificar ahora*, *Siguiete*.

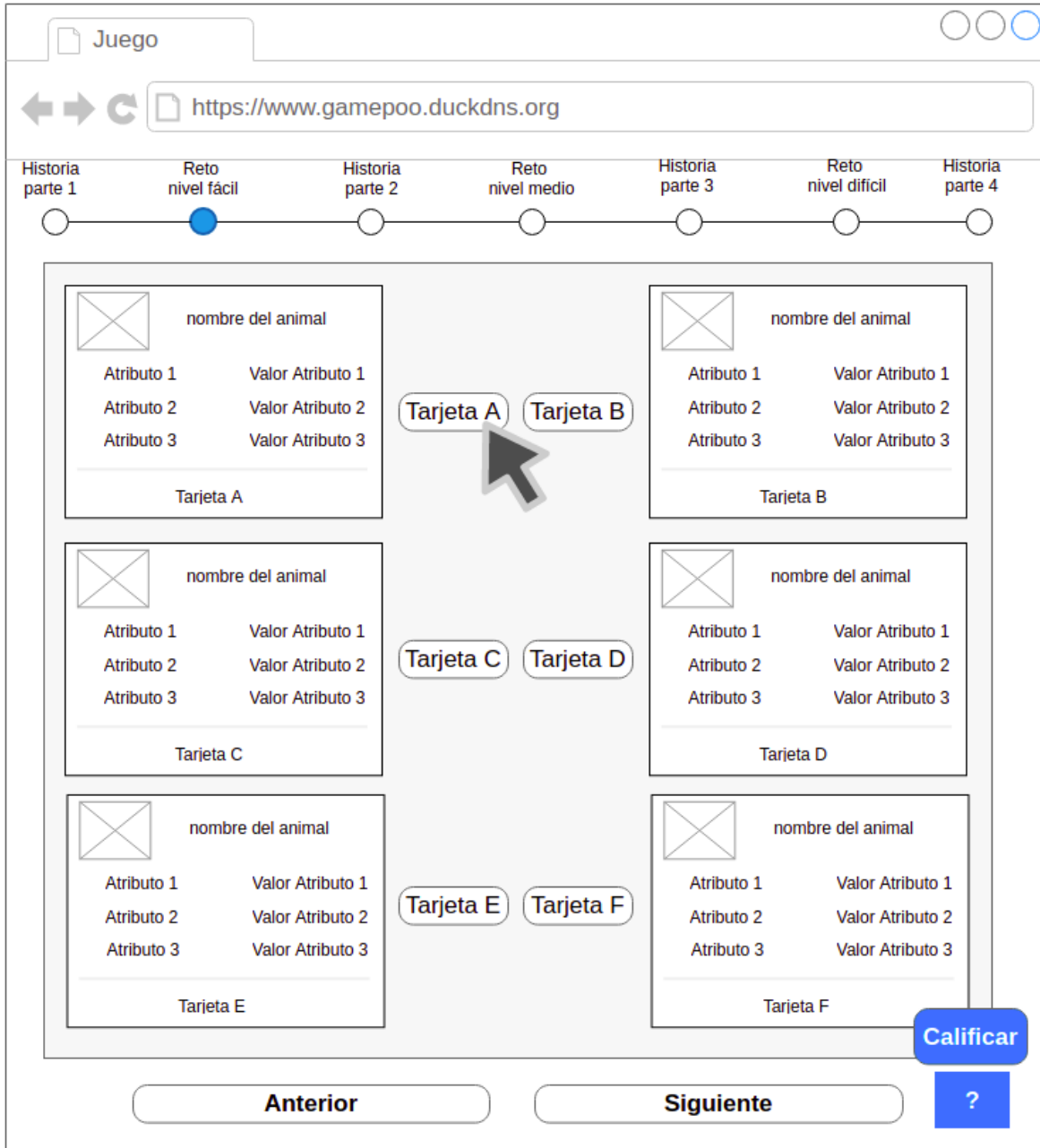


Fig. 19. Esquema para la vista de juego.

## 4.4 Implementación del sistema

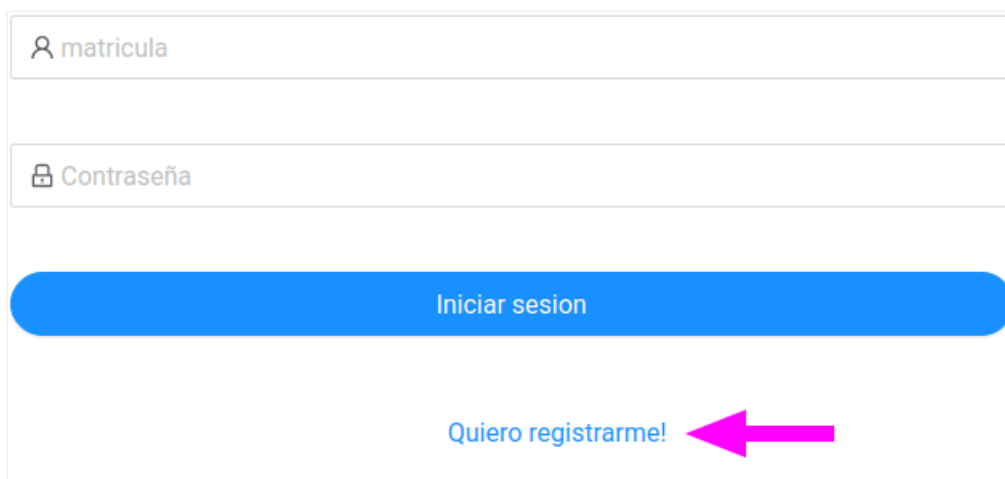
Previo a dar a conocer los métodos y características que mejoran la experiencia, es importante primero definir que este trabajo se enfoca en brindar una herramienta de

aprendizaje usable para personas con discapacidad motriz con enfoque a aquellos con limitaciones de control en las extremidades superiores. Una vez teniendo en claro al público principal al cual va dirigida esta herramienta así como conocer las limitaciones de este grupo de personas, se realizó una investigación de accesibilidad y se aterrizaron algunas características en el sistema como se mencionará a continuación.

#### 4.4.1 Inicio al sistema de aprendizaje propuesto

En este apartado explica y se muestra como desde la creación de la cuenta para el inicio de la sesión se ha diseñado la interfaz y usado elementos que sean accesibles para usuarios con discapacidad motriz. Además se muestra la sección de configuración que se ha creado para adaptar el tamaño de ciertos componentes como los botones para que usuarios con este mismo caso puedan utilizar y adaptar de acuerdo a su criterio.

En la figura 20 se muestra cómo se implementó la página principal o index del sistema; el objetivo de esta vista es para que el usuario que tenga su cuenta en el sistema pueda iniciar sesión.



The image shows a login form with two input fields. The first field is labeled 'matricula' and has a magnifying glass icon. The second field is labeled 'Contraseña' and has a lock icon. Below the fields is a large blue button with the text 'Iniciar sesion'. At the bottom of the form, there is a blue link that says 'Quiero registrarme!' with a pink arrow pointing to it from the right.

Fig. 20. Pagina index

Si el usuario no ingresa los datos correspondientes para iniciar sesión, se mostrarán mensajes indicando los errores correspondientes como se muestra en la figura 21.

The image shows a login form with two input fields. The first field is labeled 'matricula' and contains the text 'La matricula debe de ser de 9 digitos'. The second field is labeled 'Contraseña' and contains the text 'Ingresa tu contraseña!' and 'La contraseña debe ser de minimo 8 caracteres'. Below the fields is a blue button labeled 'Iniciar sesion' and a link labeled 'Quiero registrarme!'.

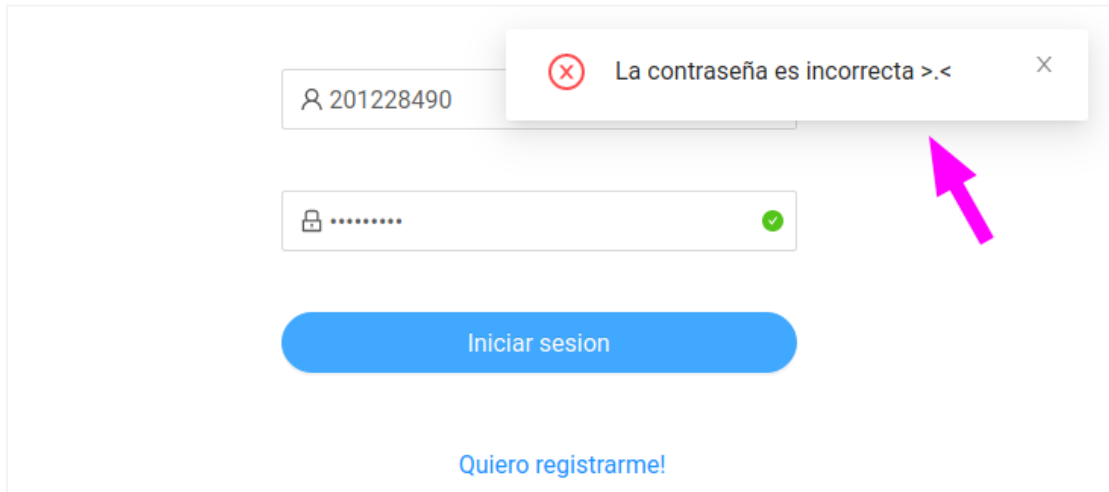
**Fig. 21.** Página index: mensajes de error que se muestran en el formulario cuando el usuario ha intentado iniciar sesión sin llenar los campos necesarios.

Si un usuario intenta iniciar sesión sin previamente registrarse al sistema o en caso de que el usuario ingrese su número de matrícula que no exista se desplegará un mensaje indicando este error, ver la figura 22.

The image shows a login form with two input fields. The first field contains the number '000000000' and has a red error message 'El usuario no existe :( ' displayed above it. The second field contains a password and has a green checkmark. Below the fields is a blue button labeled 'Iniciar sesion' and a link labeled 'Quiero registrarme!'.

**Fig. 22.** Página index: mensaje de error que se muestra cuando el usuario ha intentado iniciar sesión y ha proporcionado una matrícula que no ha sido registrada en el sistema.

Si el usuario tiene una cuenta, pero ha escrito de forma incorrecta su contraseña, se desplegará el mensaje de error que se muestra en la figura 24.



**Fig. 23.** Página index: mensaje de error que se muestra cuando el usuario ha intentado iniciar sesión y ha proporcionado la contraseña que no ha sido correcta.

En caso de no contar con una cuenta, el usuario tendrá que dar clic en el link *Quiero registrarme* disponible en la página index como se muestra en la figura 20 y se abrirá un modal con un formulario como se muestra en la figura 24 para ingresar los siguientes datos:

*Nombre de usuario:* nombre del estudiante o de la persona que quiera crear la cuenta.

*Matrícula:* matrícula de nueve dígitos del estudiante o de la persona que quiera crear la cuenta.

*Contraseña:* cadena de longitud mínima de 8 caracteres alfanuméricos.

*Contraseña otra vez:* cadena de longitud mínima de 8 caracteres alfanuméricos. Se pide la contraseña en dos ocasiones para que el usuario valide que ha escrito su contraseña correctamente y asegurar que es la palabra que quiere usar como contraseña.

The image shows a registration form titled "Title" with a close button (X) in the top right corner. The form contains four input fields: "Nombre de usuario:" (required, marked with a red asterisk), "Matrícula:", "\* Contraseña:" (required, marked with a red asterisk), and "\* Repite tu contraseña:" (required, marked with a red asterisk). The password fields have an eye icon to toggle visibility. Below the fields are two buttons: a blue "Registrarme" button and a white "Cancelar" button. A pink arrow points to the "Registrarme" button.

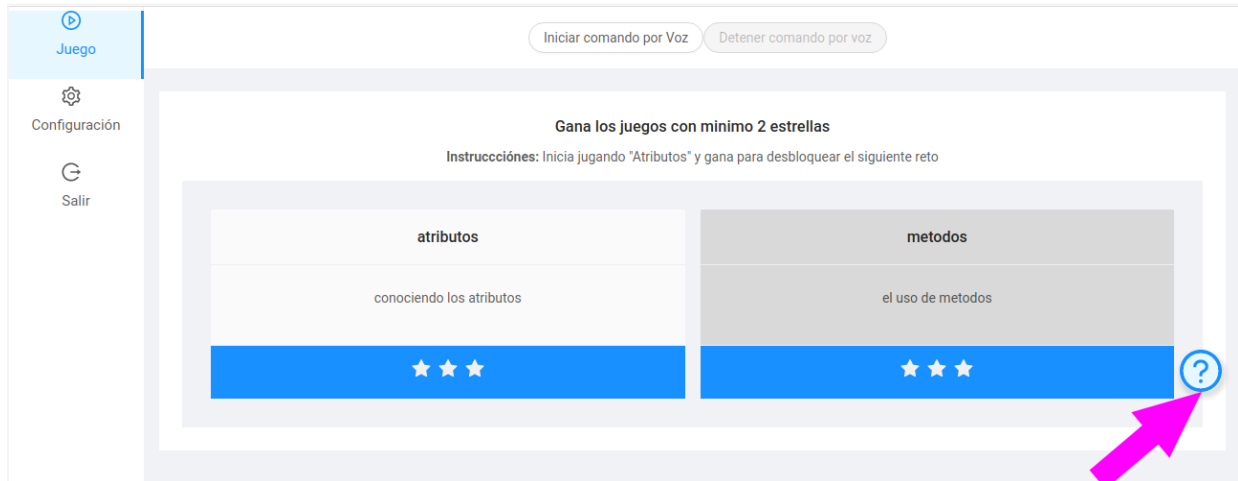
**Fig. 24.** Página index: Formulario de registro.

Si el usuario no ingresa ningún valor o algunos o en todos los campos del formulario de registro, por consecuencia, se mostrarán mensajes que indicarán al usuario los datos necesarios para realizar su registro de forma exitosa. Un ejemplo de este caso se muestra el cambio comparando la figura 24 y la figura 25. Es importante mostrar e informar al usuario el motivo del error pues de ese modo sabrá que dato ingresar o qué valor era el esperado por el sistema

The image shows a registration form with four input fields, each with a red border and a red error message below it. The fields are: 1. 'Nombre de usuario:' with the message 'Ingresa tu nombre de usuario'. 2. 'Matricula:' with the message 'La matricula debe de ser de 9 digitos'. 3. '\* Contraseña:' with the message 'Ingresa tu contraseña!' and 'La contraseña debe ser de minimo 8 caracteres'. 4. '\* Repite tu contraseña:' with the message 'Repite tu contraseña!'. Below the fields is a blue 'Registrarme' button and a grey 'Cancelar' button. On the left side, there are partial labels 'matric' and 'Contra' next to icons of a person and a lock respectively.

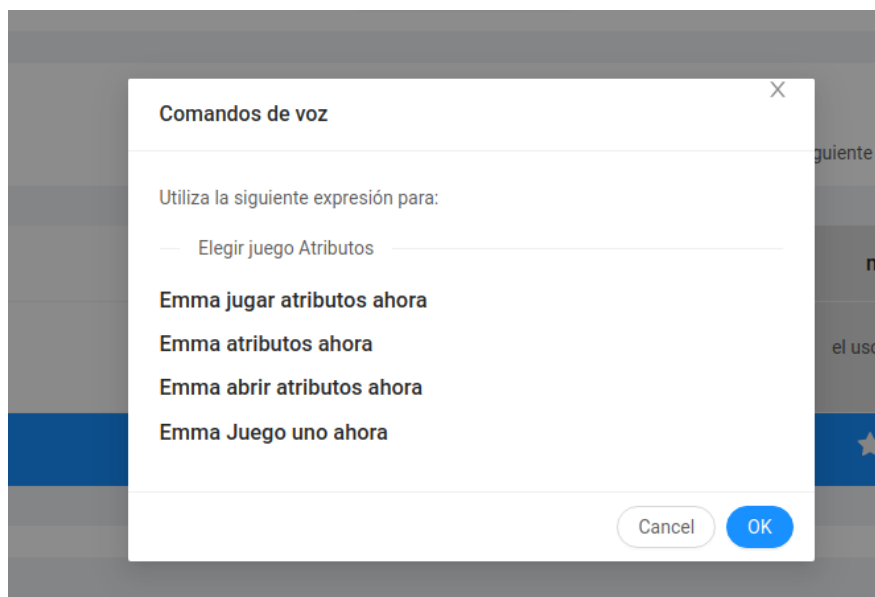
Fig. 25. Página index: mensajes de error que se muestran en el formulario de registro.

Una vez iniciada la sesión, el usuario será redirigido dentro del sistema en la página de juegos ver la figura 26. Esta página está compuesta por la barra lateral de navegación del lado izquierdo, en la parte central superior se encuentran los botones para activar o desactivar el controlador de voz, por defecto éste se encuentra desactivado. En el cuerpo de página se encuentran instrucciones de cómo empezar a jugar y debajo de éstas se encuentran los juegos disponibles, por el momento son: el juego de atributos que se encuentra habilitado y el juego de métodos que se encuentra bloqueado. Y, finalmente, en la parte inferior izquierda se encuentra un botón de ayuda representado por un signo de interrogación



**Fig. 26.** Página de juegos. Página que ve el usuario de inmediato una vez que ha iniciado sesión.

Si el usuario da clic en el botón de ayuda que se encuentra en la parte inferior izquierdo con el signo de interrogación (como se muestra en la figura 26), se abrirá un modal con todos los comandos disponibles a realizar en la página actual.



**Fig. 27.** Página de juegos. Modal que se puede visualizar al clic en el botón de ayuda.

Cuando el usuario dé clic en la opción de configuración, será redirigido a la página de configuraciones, por defecto se mostrará el contenido para cambiar el tamaño de algunos componentes con los que el usuarios se encontrará interactuando

frecuentemente. El tamaño de los componentes que se encuentran por defecto es mediano, pero puedes ser configurado a otras dos opciones de tamaño disponibles: pequeño y grande, como se muestra en la figura 30.

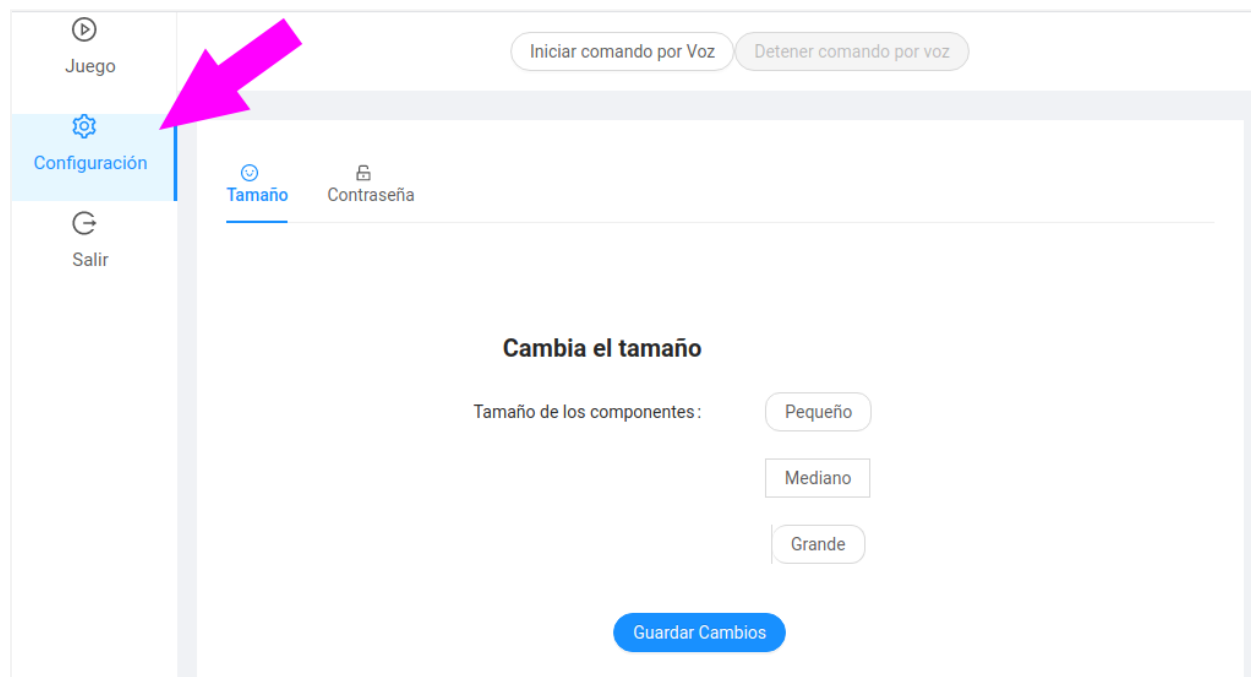
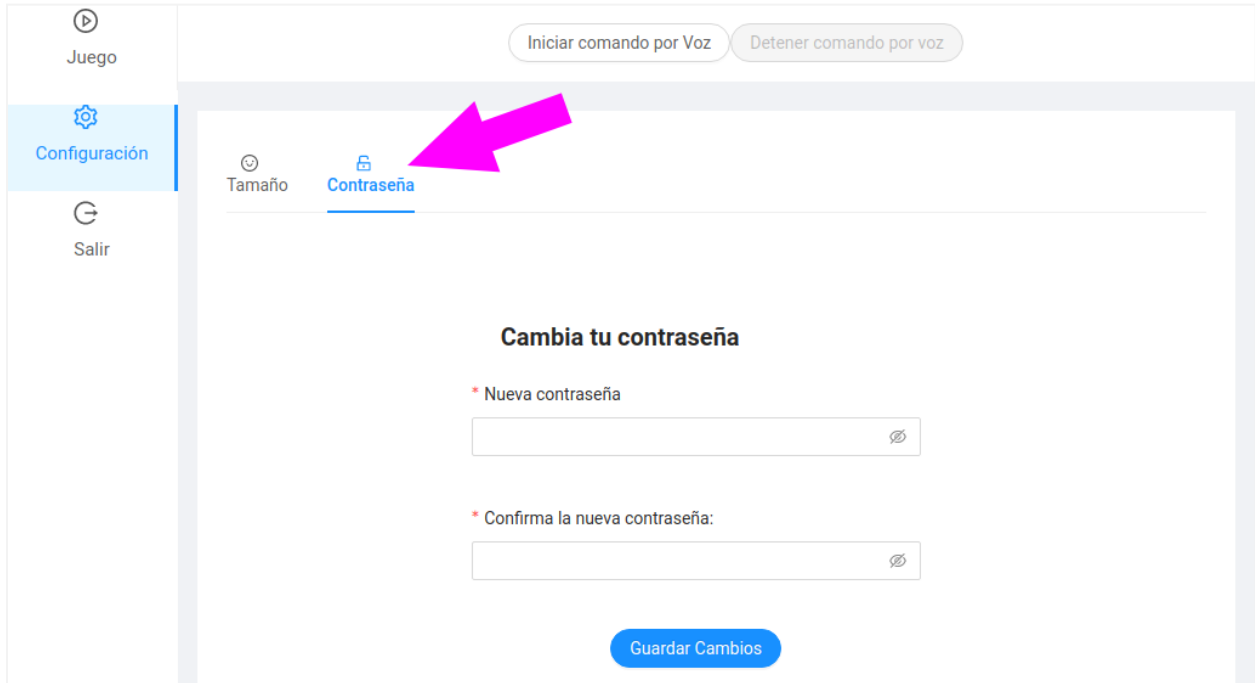
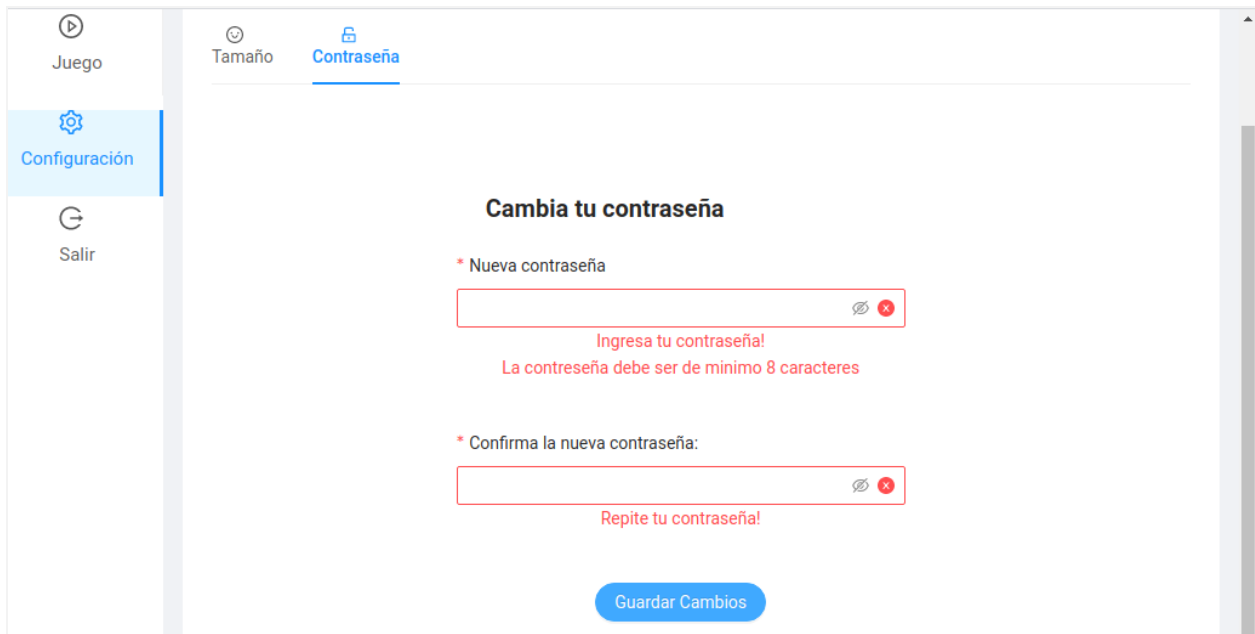


Fig. 28. Página de configuración. Vista que ve el usuario de inmediato una vez que ha cambiado a esta página.

En la página de *configuración*, se muestran en la parte superior dos pestañas, la primera de ellas es *tamaño*, esta opción muestra el contenido para cambiar el tamaño de los componentes como se ha descrito previamente (ver figura 28) y, la otra opción es *contraseña*, como su nombre lo indica es para cambiar la contraseña actual que utiliza el usuario para iniciar sesión. Como se muestra en la figura 29, el contenido de esta sección está conformada por un formulario en donde únicamente se pide que el usuario ingrese el valor de la nueva contraseña. En caso de que solo dé clic en el botón *Guardar cambios* que se encuentra visible en esta vista, se desplegarán unos mensajes de error, indicando que es necesario ingresar valores para realizar dicha actualización del nuevo valor de la contraseña, como se visualiza en la figura 30.



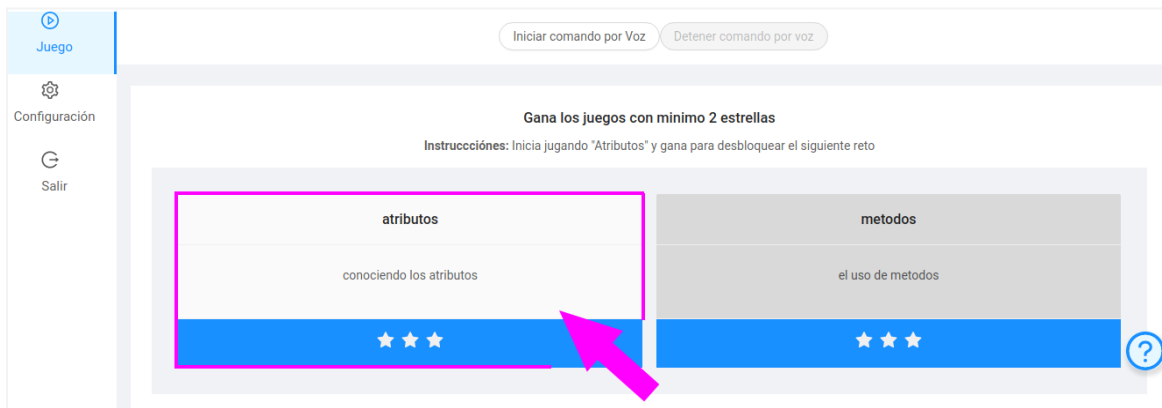
**Fig. 29.** Página de configuración. Vista de cambio de contraseña, esta sección se muestra de inmediato al entrar a la página de configuración.



**Fig. 30.** Página de configuración. Vista de la sección para el cambio de contraseña. Cuando el usuario ha dado clic al botón "Guardar cambios" sin haber llenado los cambios necesarios para cambiar la contraseña se muestran mensajes de error.

#### 4.4.2 Juego atributos

Al iniciar sesión por primera vez el único juego disponible es el de atributo, como se muestra en la figura 31. Al entrar a este juego y como ya se describe en la figura 16, éste inicia con una historia (Ver figura 32).



**Fig. 31.** Página de juegos. Haciendo énfasis al juego atributos.



Siguiente

Fig 32. Parte 1 del juego atributos se muestra el inicio de la historia.

El primer objetivo del juego es que el usuario aprenda lo que son los atributos. Como ya se mencionó, se inicia con una historia en donde se explica este concepto (ver figura 32) y se enfrenta al primer reto que es identificar cuál de las dos opciones disponibles tiene sus atributos con el tipo de valor correcto, este ejercicio se puede apreciar en la figura 33.

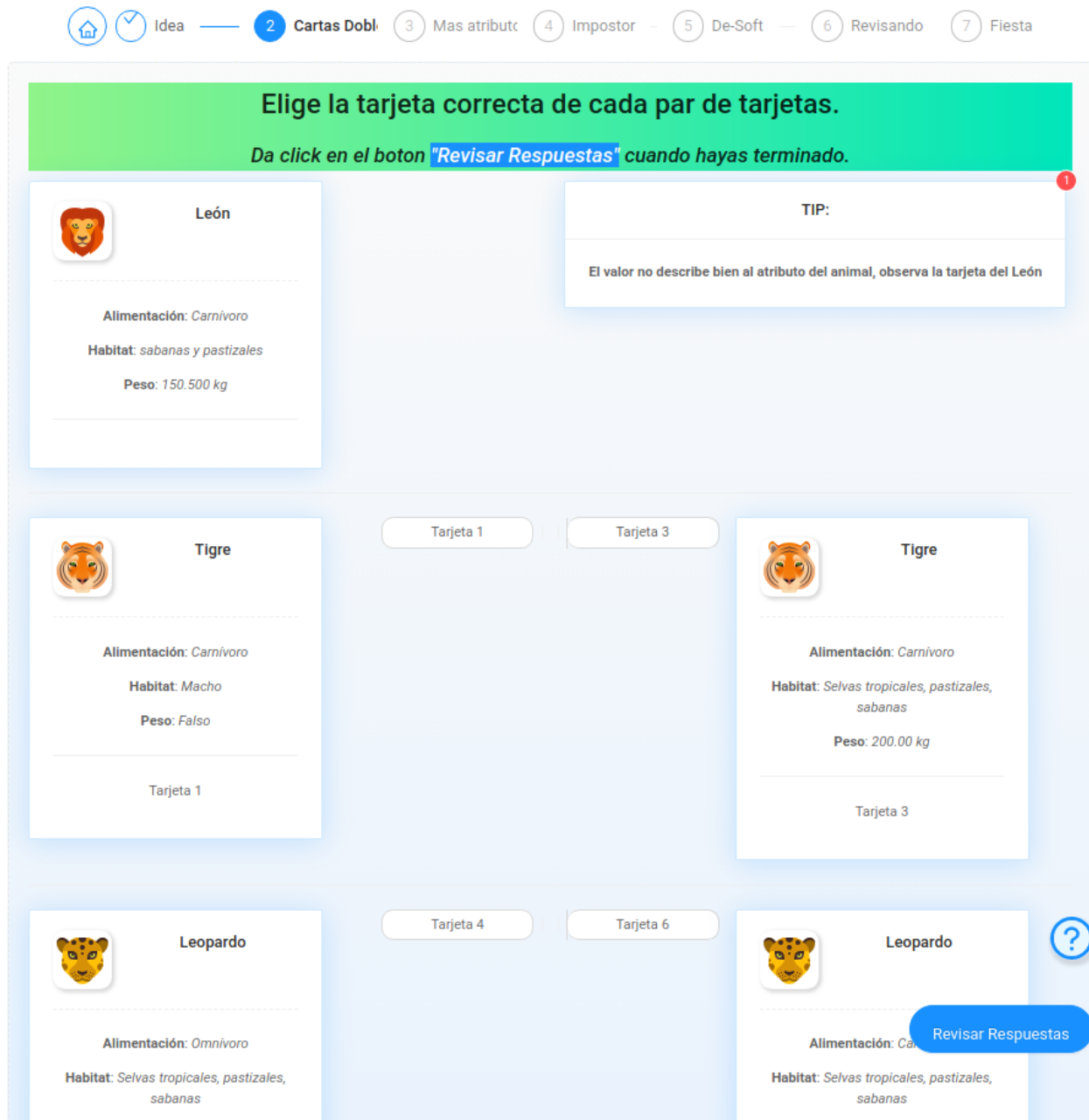


Fig. 33. Captura de pantalla parcial del primer reto nivel fácil del juego de atributos.

Una vez que se hayan contestado los ejercicios, se continuará con la segunda parte de la historia y el segundo reto con nivel medio. Ahora, el par de tarjetas son diferentes y se tendrá que elegir la que contenga atributos que son importantes según la historia, en la figura 34 se muestra este juego.

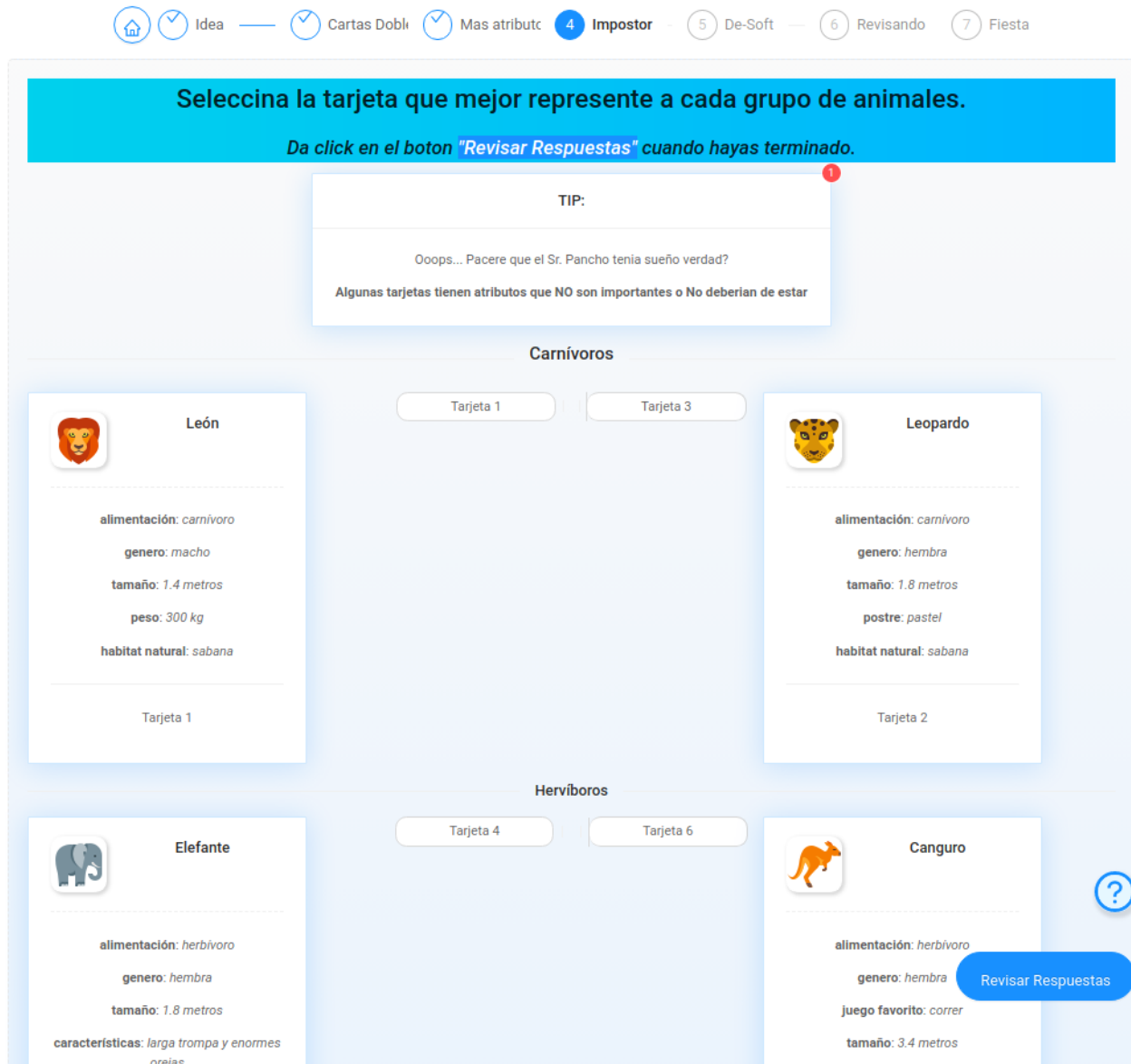


Fig. 34. Captura de pantalla parcial del segundo reto nivel medio del juego de atributos.

En la última historia, se menciona que ahora tendrá que ayudar con la revisión del código. En el juego se provee un texto con los atributos que debe contener la clase y el usuario deberá de indicar si la clase tiene o no todos los atributos que se piden, ver figura 35.

Indica si el código cumple o no con los requerimientos que se indican  
Da click en el botón "Revisar Respuestas" cuando hayas terminado.

**TIP**

\* Lee los requerimientos luego identifica los atributos que se necesitan para cada clase de animal.  
\* Revisa si el código de la derecha tiene los atributos que se requieren.

<p style="text-align: center;"><b>Carnívoros</b></p> <p>Del animal se necesita conocer: la cantidad de carne que come al día, su fecha de nacimiento, peso y su estado de salud</p>	<p>Correcto [1]</p> <p>Incorrecto [3]</p>	<pre> 1 class Carnivoro = { 2     constructor(cantidad, peso, salud){ 3         this.cantidad_carne = cantidad; 4         this.peso = peso; 5         this.salud = salud; 6     } 7 } </pre>
<p style="text-align: center;"><b>Herbívoros</b></p> <p>De este tipo de animales se necesita conocer los siguientes datos: la cantidad de alimento que consume al día, su dieta, estado de salud, y estado de su dentadura</p>	<p>Correcto [4]</p> <p>Incorrecto [6]</p>	<pre> 1 class Herbivoro = { 2     constructor ( cantidad, dieta, salud, dentadura) { 3         this.cantidad_alimento = cantidad; 4         this.dieta = dieta; 5         this.salud = salud; 6         this.estado_dentadura = dentadura; 7     } 8 } </pre>
<p style="text-align: center;"><b>Aves</b></p>	<p>Correcto [7]</p>	<pre> 1 class Ave = { </pre>

[Revisar Respuestas](#)

**Fig. 35.** Captura de pantalla parcial del tercer reto nivel difícil del juego de atributos.

El juego para aprender el concepto de métodos es similar al juego de atributos pero enfocado a los métodos *get* y *set* y a su funcionamiento.

#### 4.4.3 Características de la aplicación de la gamificación

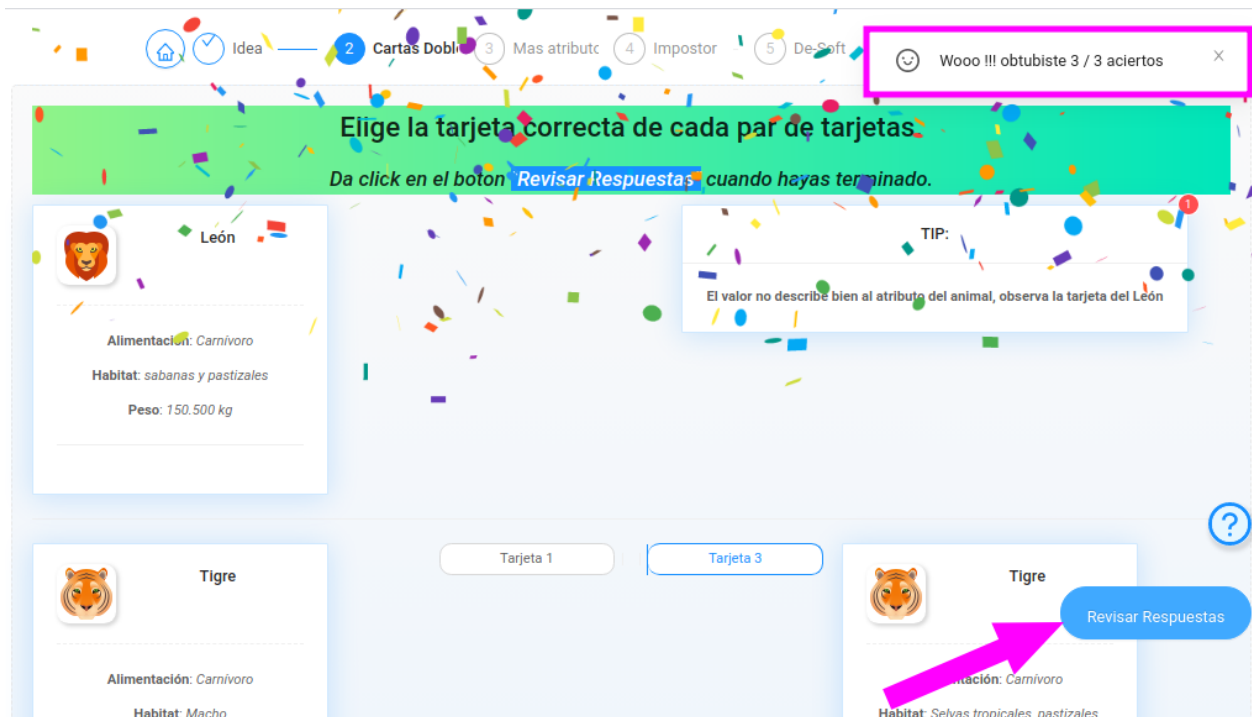
Para asegurar la efectividad del juego serio, Norero menciona cinco elementos básicos: [11]: *historia*, *dinámica de interacción*, *retroalimentación*, *simulación* y *aprendizaje*. A continuación se menciona de qué forma se ha tratado de implementar estos elementos de juego en el sistema propuesto.

La historia, el aprendizaje y la simulación se han tomado juntas y se ha aplicado en la creación de una *historia* que se muestra durante los juegos como se indica en la figura 32. Ésta tiene como personaje principal a un estudiante que está aprendiendo a programar y, durante el desarrollo de la historia, va conociendo a personajes secundarios que le explican al personaje principal conceptos referentes a la programación orientada a objetos, de este modo se aborda el elemento de *aprendizaje* que menciona Norero. Además, en la historia se han creado situaciones en donde el jugador nota de dónde surge la necesidad de crear programas, así como su beneficio en la sociedad y éste, poco a poco, va siendo sumergido en el ámbito del desarrollo de software y, de este modo, *simular* una situación de la vida real.

Hay que recordar que la dinámica de interacción se refiere a cómo se llevará a cabo el juego, cómo tener reglas, objetivos, sistemas de recompensas, clasificación de jugadores, etcétera [11]. Como se muestra en la figura 31, se implementó la regla donde inicialmente el juego disponible es el de atributos, una vez que el jugador haya pasado este nivel podrá habilitar el siguiente juego que lleva por nombre *métodos*.

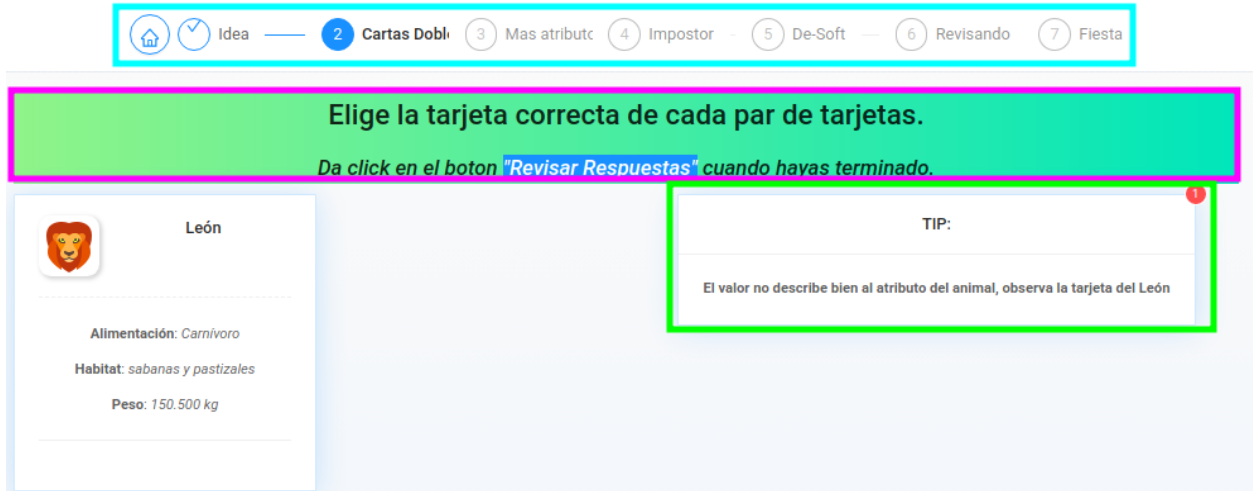
Finalmente, como *retroalimentación* o *feedback* se ha implementado de las siguientes formas.

- Cuando el jugador ha dado clic en el botón de *Revisar respuesta*, se despliega un mensaje con el número de aciertos obtenido en el nivel actual, en caso de haber acertado en todos los incisos en la vista corriente, se mostrará confeti cayendo por toda la página, como se muestra en la figura 36. En caso de no haber acertado en alguna, simplemente se desplegará el mensaje de puntuación en la parte superior derecha.



**Fig. 36.** Captura de pantalla parcial del juego de atributos nivel fácil, representación de retroalimentación al jugador al haber dado clic en el botón de “Revisar respuestas”.

- En la figura 37, se muestra cómo dentro de la estructura de los juegos se ha agregado una barra de progreso (resaltado de color azul claro), también se puede notar en los colores donde se encuentran las instrucciones de cada nivel (resaltado de color rosa) y se han agregado cartas de ayuda o tips que pueden servir cuando el jugador se enfrenta a cada reto.



**Fig. 37.** Captura de pantalla parcial del juego de atributos nivel fácil, representación de retroalimentación al jugador dentro de la estructura del juego.

- Una vez que el jugador termina con el juego, es redirigido a la página de juego y aquí se puede ver otros efectos de retroalimentación y reglas. Se muestra un mensaje en la parte superior central con el contenido que indica que el juego ha terminado, en la tarjeta del juego de atributos las estrellas que está en el pie de la tarjeta, están coloreadas en tono amarillo indicando el valor de calificación se obtuvo en el juego (ver figura 38). También, se puede ver cómo la tarjeta del juego de método se ha desbloqueado, pues se ve un tono más claro (este cambio de color se puede notar en comparacación con la figura 31) y, ahora, se puede dar clic en él.



**Fig. 38.** Captura de pantalla de la página de juegos, una vez que el usuario ha terminado el juego de tributos.

## 4.5 Características de accesibilidad en el juego serio desarrollado

Algunos usuarios con discapacidad motriz regularmente usan el teclado, así que se diseñaron juegos en donde la interacción del usuario con el uso del teclado fuera mayor. Además, se implementaron funciones para que los juegos fueran contestados por medio de comandos de voz, lo que mejora la usabilidad.

Para hacer el sistema accesible para personas con discapacidad motriz con limitaciones de movimientos finos, se implementaron múltiples formas de interacción que pueden mejorar la experiencia en el uso del juego, entre las que se aplicaron fueron las siguientes.

*Estilos de enfoque*, algunos componentes como botones, *input*, *radio button*, *checkbox* por nombrar algunos, son resaltados por los bordes indicando así que es el elemento con el cual se está interactuando actualmente, como se muestra en las figuras 40 y 41. Esta característica la provee la librería Ant design [18], que se utilizó para la implementación de la herramienta propuesta.

*Tabindex*, un ejemplo es cuando se tiene un formulario, el usuario se puede desplazar a través de este utilizando la tecla tabulador. Esta característica se implementó cuando un usuario crea su cuenta ingresando sus datos en un formulario, como se puede apreciar en la figura 24.

*Capas accesibles*, esta característica se puede notar al mostrar un modal, o también conocido como ventana flotante, cuando se hace visible, el contenido que tiene de fondo es sombreado, esta característica puede notarse en las figuras 24, 25 y 27.

Con este mismo componente se puede notar la cualidad de cancelación por punto, es decir, el modal puede ser cerrando dando click fuera de la venta, y ésta se cerrará de inmediato; de esta forma, el usuario no tiene que dar click en un botón para cerrar lo cual es complicado para el grupo de usuarios al que va dirigido el proyecto.

Una característica que se tuvo presente durante todo el desarrollo del proyecto fue

tener *áreas grandes de interacción*: esta propiedad se puede ver reflejada al seleccionar el juego y en la barra lateral para la navegación de los menús disponibles en la página.

Otro rasgo que se implementó en el proyecto es contestar los juegos por medio del teclado numérico : como menciona Firth [4] algunos usuarios por conveniencia hacen uso del teclado o bien es una característica que puede mejorar la usabilidad.

Finalmente se implementó *comando por voz* para navegar, por el sitio, a través de los juegos y para responderlos, el usuario da algún comando de acuerdo a lo que quiera hacer en la página actual.

En resumen, los juegos se pueden contestar de tres formas distintas:

1) *Dando click sobre los botones*, esto está dirigido a usuarios que hacen uso del mouse o touchpad

2) *Haciendo uso del teclado*, presionando la tecla del número que corresponda con el del botón de la interfaz; y,

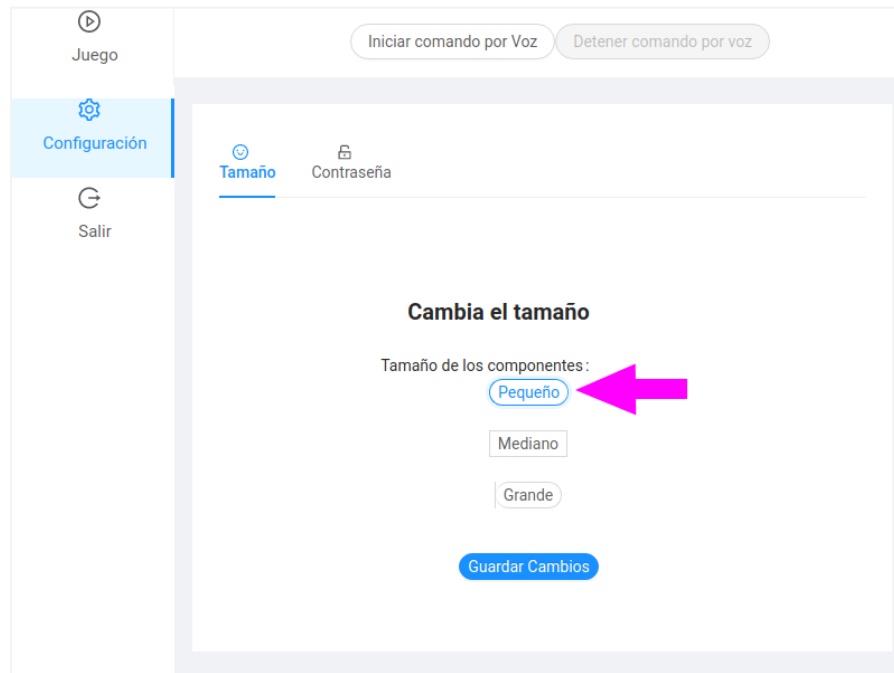
3) *Hablando y diciendo el texto* que corresponda con el botón que se desea dar clic. También, en la parte inferior derecha de cada juego se encuentra un botón flotante que califica las respuestas del alumno, este puede ser activado por medio de voz o bien dando click como se puede apreciar en la Figura 8.

En la figura 39, se puede visualizar cómo las opciones de la barra lateral son amplias, así como las tarjetas para acceder a los juegos de atributos y métodos y, en la parte superior central, se encuentran los botones para activar o desactivar el asistente de voz.



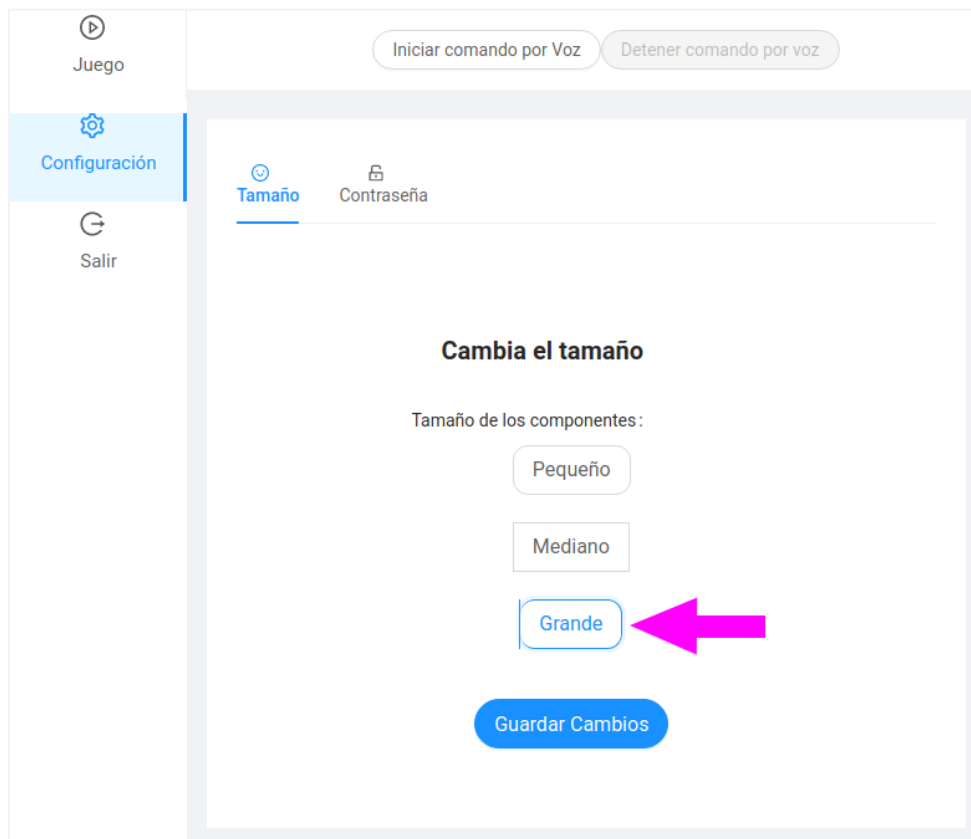
**Fig. 39.** Captura de pantalla de la página de juegos haciendo énfasis en las características implementadas de accesibilidad en la interfaz.

En la figura 40, se muestra el apartado donde el usuario podrá cambiar el tamaño de los componentes que se encuentran en el software. Las opciones disponibles son pequeño, mediano o grande, según sea la necesidad; hay que tener en cuenta la posibilidad que este recurso puede ser usado por personas sin alguna discapacidad.



**Fig. 40.** Captura de pantalla de la página de configuraciones para el cambio de tamaño. Tamaño pequeño seccionado.

Se puede notar el cambio de tamaño en la sección de *contraseña*. Al comparar la figura 40 y 41, se puede observar este efecto en los botones para el cambio de tamaño. En la figura 40, se nota que los bordes internos de los botones son más angostos en comparación con el espacio de los bordes de la figura 41.



**Fig. 41.** Captura de pantalla de la página de configuraciones para el cambio de tamaño. Tamaño grande seccionado.

Como se ha mencionado y mostrado en la figura 26, existe un botón en la parte inferior izquierda con un signo de interrogación, al dar clic, se despliega un modal con los comandos disponibles en esa página, como se puede observar en la figura 28, este peculiar botón de ayuda aparece en varias vistas del sistema de juegos, se nota en las figuras 32, 33, 34 y 35 que son el contenido del juego y en la página de juegos como se muestra en la figura 39. En la figura 42, se muestra una tabla con los comandos disponibles hasta el momento para la navegación a través del sistema, así como comandos de voz que pueden ser utilizados, como se ha dicho, para responder los juegos creados en este sistema de aprendizaje. Es importante mencionar que, como se muestra en la figura 26, se tiene que decir la palabra *Emma* seguido de los comandos de la figura 42.

Comando de voz	Descripción	Respuesta y/o acción a realizar
Al activar controlador de voz		
<ul style="list-style-type: none"> <li>Hola</li> </ul>	Una vez que se ha activado el botón de comando de voz este responderá en automático con el saludo "hola"	No aplica
<ul style="list-style-type: none"> <li>Como estas</li> </ul>	Para validar que está prendido el micrófono para poder detectar la voz del usuario.	Estoy bien
Página de juegos		
<ul style="list-style-type: none"> <li>Jugar atributos</li> <li>Atributos</li> <li>Abrir atributos</li> <li>Juego uno</li> </ul>	Estos comando tienen el objetivo de iniciar el juego de atributos, es decir, en lugar de dar clic a la tarjeta de atributos para iniciar a jugar, podría decir estos comandos	Una vez que se detecta el comando abrirá de inmediato el juego. En caso de estar en una sección diferente a la página de juegos el controlador de voz no podrá realizar la acción y responderá con la siguiente frase "Lo siento no se puede hacer esta acción por ahora"
En cada sección del juego		
<ul style="list-style-type: none"> <li>Opción uno</li> <li>Opción tres</li> <li>Opción cuatro</li> <li>Opción seis</li> <li>Opción siete</li> <li>Opción nueve</li> </ul>	Estos comandos tienen el fin de seleccionar por comando de voz alguna opción disponible, estas opciones es el indicador que tiene cada una de las tarjetas de los juegos.	El controlador de voz responderá con la frase "NumeroDeOpcion seleccionado", es decir si el usuario dice "Opción uno" el controlador responderá "uno seleccionado" y se activará el botón correspondiente, como si el usuario hubiera dado clic en el botón.
<ul style="list-style-type: none"> <li>Revisar respuestas</li> </ul>	El usuario podrá decir este comando cuando el usuario tenga que revisar cuantos aciertos obtuvo de sus respuestas previamente seleccionadas	Como respuesta se escuchará "Revisando" y se mostrará un mensaje del número de aciertos obtenidos.
<ul style="list-style-type: none"> <li>Siguiente</li> </ul>	Comando para continuar a la siguiente sección del juego	Redireccionará al usuario a la siguiente sección más próxima.
<ul style="list-style-type: none"> <li>Fin</li> </ul>	Comando para finalizar el juego	Terminará con el juego y el usuario será redireccionado a la página de juegos.
Navegación entre las páginas del <a href="#">sitio</a> (página de juegos y página de configuración)		
<ul style="list-style-type: none"> <li>Juego</li> <li>Juegos</li> <li>Ir a juegos</li> </ul>	Cuando el usuario tenga o necesite ir a la página de juegos podrá decir estos comandos en voz alta	El usuario será redirigido de enmedio a la página de juegos.
<ul style="list-style-type: none"> <li>Configuración</li> <li>Ir a configuración</li> </ul>	Cuando el usuario tenga o necesite ir a la página de configuración podrá decir estos comandos en voz alta	El usuario será redirigido de enmedio a la página de configuración.

Fig. 42. Comandos disponibles para la navegación a través del sistema de juegos propuestos.

# Capítulo 5 Pruebas y resultados

## 5.1 Pruebas internas de desarrollo

Se utilizó la herramienta *Funkify* con el fin de experimentar las dificultades que un usuario con discapacidad motriz en las extremidades superiores tiene que lidiar al utilizar un sitio web. *Funkify* es una herramienta la cual ayuda a experimentar interfaces y la web desde una perspectiva de usuarios con capacidades y discapacidades diferentes, como son una visión borrosa, poco control en el ratón, filtros de colores y dislexia. Para poder usarla, se instala como una extensión en el navegador *Google Chrome*, que permite emular los movimientos de puntero de un usuario con poco control del mismo, estas pruebas ayudaron para definir la dinámica de interacción que tendría el usuario al responder a los juegos.

Durante el desarrollo, se realizaron pruebas internas de usabilidad de manera empírica para detectar áreas en donde los usuarios con discapacidad podrían tener mayor problema al tratar de utilizar el sistema web, ya sea navegando e interactuando en él cuando el usuario intenta responder a los juegos.

## 5.2 Pruebas de lanzamiento del sistema

Debido a la pandemia mundial causada por el virus coronario COVID-19, las pruebas se aplicaron de manera remota. Por lo tanto el sitio web se desplegó el proyecto a internet bajo el dominio <http://gamepoo.duckdns.org/> para que los usuarios de prueba pudieran acceder.

Debido a la contingencia de salud que se presentó durante la fase de evaluación de este trabajo, se aplicó un test vía remota, por lo tanto, se puso en producción la página web utilizando un servidor casero y se utilizaron las herramientas google meet para reunir a los alumnos y google forms para crear un test y recolectar la información.

Se aplicó un test a 11 participantes con edades entre 19 y 21 años, de los cuales

81.8% son hombres y el 18.2% son mujeres. Los participantes pertenecen a las carreras de Ing. en Ciencias de la computación. En el momento en que se aplicó dicho test, todos los participantes se encontraban cursando el primer semestre.

Para la evaluación del programa, se aplicó el test CSUP en la versión adaptada al español con escala de Likert con 7 niveles de izquierda a derecha va de totalmente de desacuerdo a totalmente de acuerdo como se puede apreciar en el anexo 3.

*CSUP* es una herramienta que tiene un buen rendimiento para evaluar las respuestas de satisfacción de los usuarios a una interfaz de un sitio web. Este test es apto utilizarlo en pruebas de campo (ambiente no controlado) para realizar evaluaciones veraces con grupos reducidos de usuarios y seguir manteniendo altos niveles de confiabilidad (ver anexo 3).

Este cuestionario se encuentra compuesto por 16 preguntas las cuales se encuentran en tres categorías para la evaluación de:

*calidad del sistema*: conformado por las preguntas de 1 a 6;

*calidad de la información*: conformado por las preguntas de 7 a 12;

y *calidad de la interfaz*: conformado por las preguntas de 13 a 16.

El test se encuentra conformado por dos secciones. La primera parte está conformada por un test *CSUP* versión 3 (ver anexo 3) de 16 preguntas utilizando una escala Likert de siete niveles, de izquierda a derecha donde (1) iban de totalmente en desacuerdo a (7) totalmente de acuerdo (ver anexo 3) y se consideró agregar cinco preguntas adicionales al test para conocer los datos personales de los participantes como su nombre completo, edad, sexo, carrera, y el grado o semestre que se encuentran estudiando. La segunda parte (ver anexo 4), se encuentra conformada por ocho preguntas, las primeras siete utilizando una escala Likert de siete niveles de respuesta y la octava pregunta es abierta para que los alumnos pudieran dejar algún comentario referente al sistema (ver anexo 4).

### Calidad del sistema: pregunta 1 a la pregunta 6

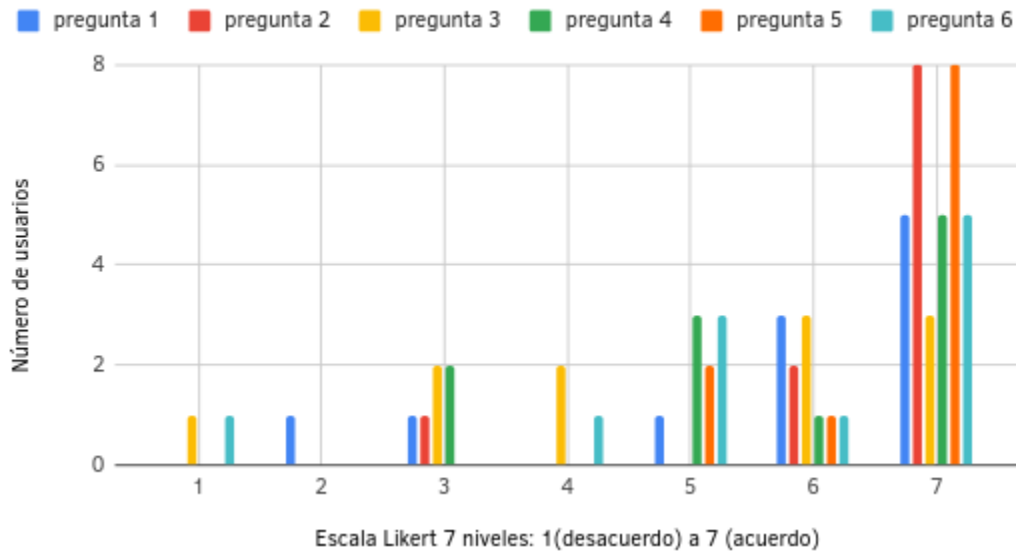


Fig. 43. Calidad del sistema.

En la gráfica de Calidad del sistema. El eje  $x$  representa las posibles respuestas en escala de Likert representadas del 1 al 7 donde 1 equivale a totalmente en desacuerdo y 7 es totalmente de acuerdo. Esto se aplica para las figuras 43, 44, 45 y 46 (gráficas).

En la figura 43, el 72.7% de los participantes estuvo totalmente de acuerdo en las preguntas 2 (Fue simple utilizar este sitio) y 5 (Fue fácil aprender a utilizar este sitio web), lo que indica que el sitio es simple y fácil de utilizar.

En el caso de las preguntas 1(En general, estoy satisfecho con lo fácil que es utilizar este sitio web), 4 (Me siento cómodo utilizando este sitio web) y 6 (Creo que me volví experto rápidamente utilizando este sitio web), los resultados se encuentran distribuidos por la tabla, pues el 45.4% participantes estuvo totalmente de acuerdo, por lo tanto, en general utilizar el sitio web puede tener características que no sea agradable de utilizar o se necesite un tiempo de aprendizaje para utilizar el sitio.

Para la pregunta 3 (Soy capaz de completar mi trabajo rápidamente utilizando este sitio web) los estudiantes consideran que se requiere más tiempo de lo necesario para concluir con las actividades. Un participante compartió su experiencia “...como retroalimentación quisiera comentar que las historias tardan mucho tiempo en cargar,

tal vez en otro formato cargarían más rápido porque se pierde un poco el hilo de lo que estás haciendo.”

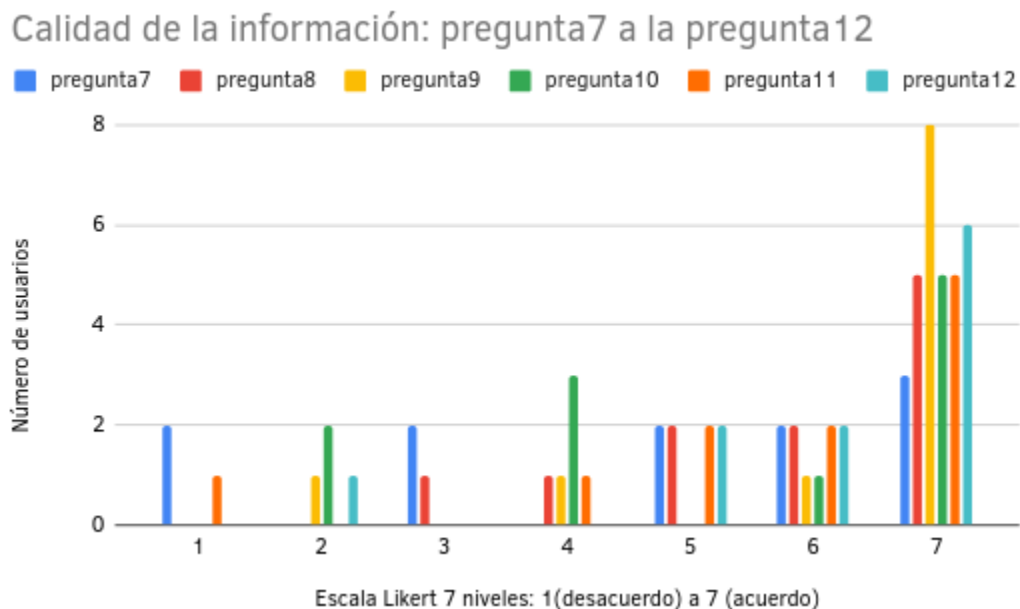


Fig. 44. Gráfica de calidad de la información.

Para la pregunta 9 (La información (como ayuda en línea, mensajes en pantalla y otra documentación) que provee este sitio web es clara), el 72.7% de los usuarios coinciden, el sitio proporciona información clara (ver figura 44). En el caso de la pregunta 12 (La organización de la información del sitio web en la pantalla fue clara), el 54.5% de los participantes consideran que la organización de la información del sitio web es clara. Sólo 45.5% participantes estuvieron totalmente de acuerdo en las preguntas 8 (Cada vez que cometo un error utilizando el sitio web, lo resuelvo fácil y rápidamente), 10 (Es fácil encontrar en el sitio web la información que necesito) y 11 (La información que proporciona el sitio web fue efectiva ayudándome a completar las tareas) y únicamente el 27.2% participantes para la pregunta 7 (El sitio web muestra mensajes de error que me dicen claramente cómo resolver los problemas), estos puntajes nos hace inferir lo siguiente: El sitio web muestra información clara de cómo resolver los problemas que se presentan, sin embargo, la información proporcionada puede que no sea suficiente para resolver los problemas presentados y, por lo tanto,

pueden llegar a ser complicados de resolver; la organización de la información en el sitio es concisa, pero puede mejorar pues los mensajes de retroalimentación se limitan a marcar errores o instrucciones y éstas últimas pueden ser confusas y complicadas. El siguiente comentario de un participante nos hace una observación que tiene relación con este apartado. *“En lo personal está genial el sitio web solo que sí recomiendo que en las indicaciones sean un poco más claras porque al inicio es un poco complicado de comprender qué se tiene que hacer, pero de ahí en adelante todo bien”*

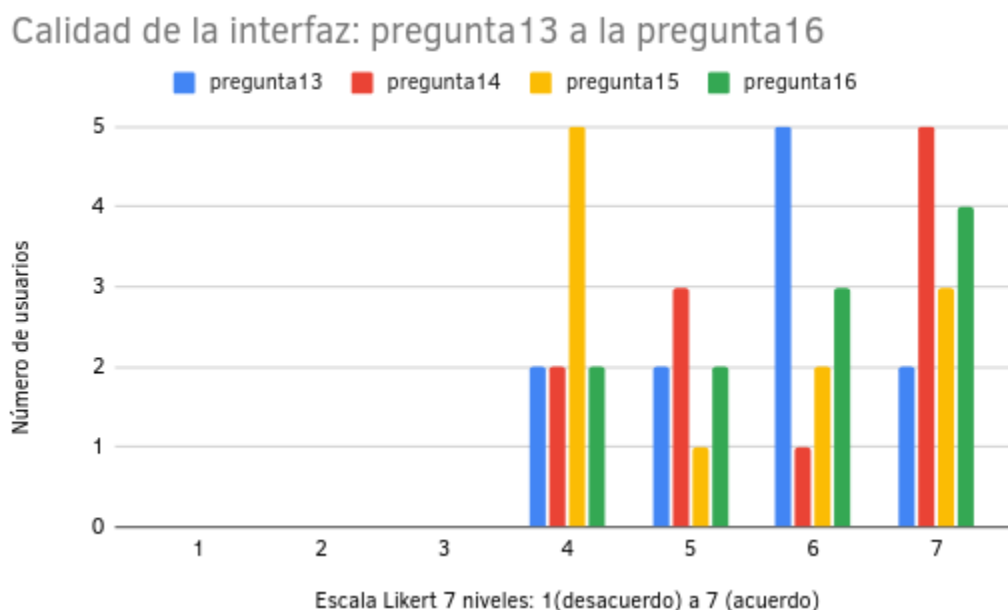


Fig. 45. Gráfica de calidad de interfaz

La interfaz del sitio web tiene ciertos elementos visuales que no son tan agradables a la vista, también se esperaba tener algunas herramientas que podrían ser de utilidad, a falta de ellas provocó que los participante no estuvieran tan satisfechos con el sitio. A pesar de todos estos detalles, los participantes se llevaron una buena experiencia al interactuar con el sitio.

Esta gráfica tiene una característica especial (ver figura 45), pues todas las respuestas de las preguntas 13 (La interfaz del sitio web fue placentera), 14 (Me gustó utilizar el sitio web), 15 (El sitio web tuvo las herramientas que esperaba que tuviera) y 16 (En general, estuve satisfecho con el sitio web) se encuentran del nivel cuatro en adelante lo cual indica que para todos los alumnos la interfaz es aceptable en general. A pesar

de esto, al no tener todas las respuestas en el nivel 7, es decir, totalmente de acuerdo, lo cual nos indica que la interfaz va por buen camino, pero existen algunas características que podrían hacer mejor al sitio. El siguiente comentario nos indica una característica que puede ser implementada con el fin de mejorar la interfaz del sitio. *“podría ser posible el moverte mejor en las historias ya que hay veces necesitaba la información de las historias y no me podía mover a través de ellas”*

De acuerdo a los resultados obtenidos y a los comentarios de algunos participantes, podemos concluir que el sitio es bueno, pero puede ser mucho mejor reduciendo el tiempo de carga de las historias, así como optimizar la dinámica para navegar entre éstas, hacer que visualmente sean más distintivos los mensajes de retroalimentación, revisar la redacción de las instrucciones para evitar posible confusión

En la segunda parte del test, se realizaron preguntas referentes al contenido y la accesibilidad, está compuesto por 7 preguntas las cuales pueden ser contestadas por medio de la escala de Likert con 7 niveles. El cuestionario se puede encontrar en la sección de anexos al final de este trabajo.

## Contenido y Accesibilidad: pregunta 17 a la pregunta 23

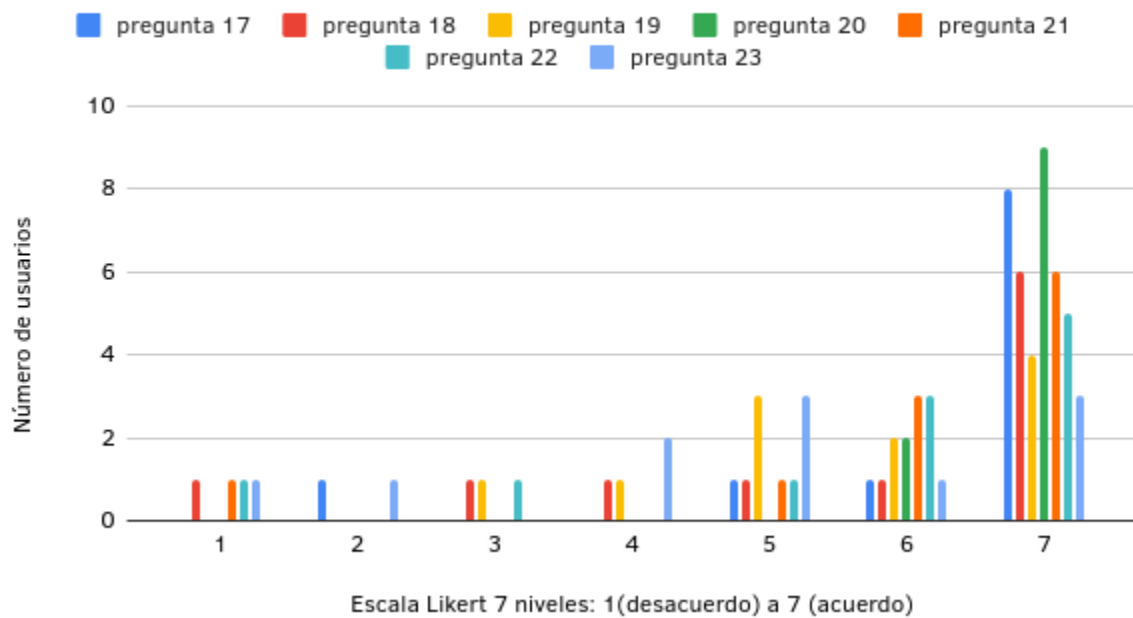


Fig. 46. Gráfica de evaluación de contenido y accesibilidad.

Como se puede notar la gran mayoría de los resultados se encuentran en los niveles 5 en adelante, así, el contenido como las opciones de accesibilidad fueron bien valorados. Para el análisis de la tabla se observa con detenimiento la columna con el nivel 7 (ver figura 46). La pregunta 20 (¿Comprendiste el concepto de atributo?) fue la mejor calificada donde se obtuvo un 81.9 % en el nivel 7 (totalmente de acuerdo), esto quiere decir que los alumnos entendieron bastante bien el concepto de atributos, aunque parece que entender el concepto de objeto que corresponde a la pregunta 21 (¿Comprendiste el concepto de Objeto?) les fue un poco complicado pues se obtuvo un 54.5% en el nivel 7. A pesar de que se obtuvo un porcentaje bajo, se puede notar como el 90.9 % de las respuestas se encuentran dentro de los niveles de aceptación de la escala de likert (entre los niveles 5 y 7 ).

La pregunta 17 (¿El contenido fue claro?) fue la siguiente mejor valorada con un 72.7%, por lo tanto, en general, el contenido del sistema es bastante concreto y entendible.

Para la pregunta 18 (¿La historia fue entretenida?), el 54.5% de los encuestados

estuvieron totalmente de acuerdo que la historia fue entretenida, pese a que se obtuvo un porcentaje apenas rebasando el 50%, se puede notar en la gráfica como la mayoría de las respuesta se encuentran en el cuadrante de aceptación, esto nos indica que la historia es aceptable, sin embargo, tiene potencial para mejorarse.

En la pregunta 17 (¿El contenido fue claro?), el 36.6% de los participante estuvo totalmente de acuerdo; también, se puede apreciar como los resultados se encuentran desde el nivel 3 al nivel 7, a pesar que de el porcentaje es bajo, es una buena señal, pues los ejercicios están hechos para que incrementen de complejidad de manera gradual, ya que si los ejercicios son complejos, los usuarios dejarán de jugar y perderán el interés. Lo mismo sucede para la pregunta 22 (¿Consideran agregar más ejercicios para comprender el tema y alcanzar el objetivo?), donde en el nivel 7 se obtuvo un 45.4%, un porcentaje por debajo de la mitad, de modo que los ejercicios son suficientes para comprender los temas que se pretenden explicar. Finalmente, en la pregunta 23 (Las opciones de usabilidad para personas con capacidades diferentes te ayudaron a “moverte” en el contenido) se puede observar cómo el 63.6% de las respuestas se encuentra en el rango de 5 a 7, dicho de otra manera, la mayoría de los participantes consideran que las opciones implementadas pueden ayudar a usuarios con discapacidad motriz a utilizar este sitio.

## Capítulo 6 Conclusiones y trabajo futuro

Tomando en cuenta los objetivos iniciales de este proyecto, los cuales son crear un sistema de aprendizaje de la programación orientada a objetos, que dicho sistema sea accesible para personas con discapacidad motriz con limitación en la extremidades superiores y que, además, este sistema haga uso de las características de la gamificación y, de acuerdo los datos obtenidos de las pruebas realizadas, podemos concluir que la herramienta desarrollada sí logró los objetivos de ser un medio para el aprendizaje de conceptos de iniciación a la programación orientada a objetos. Aunque, respecto a la accesibilidad en el sitio web implementado, los usuarios aprueban tener diferentes medios para navegar a través del sitio.

De acuerdo al objetivo *Definir los conceptos que están relacionados directamente con el tema de clases (atributos, métodos e instancia) del paradigma Orientado a Objetos*, se probó que al generar una historia por concepto se logró transmitir el conocimiento como se puede notar en los resultados obtenidos en la parte 2 de la encuesta (donde se realizaron preguntas referentes al contenido y la accesibilidad) pues según las respuestas obtenidas, aproximadamente el 90% de los participantes se encuentran dentro del rango de aceptación de la escala de Likert.

En base en las investigaciones realizadas, los juegos fueron desarrollados tomando en cuenta las técnicas de la gamificación, por lo tanto, dio como resultado tener un concepto explicado por mini juegos e ir incrementando la complejidad paulatinamente. De este modo, se pretende cumplir con el objetivo *“Diseñar con base en las técnicas de gamificación cada uno de los minijuegos que conformarán el juego principal para que se entienda un concepto a la vez en cada minijuego.”*

Uno de los objetivos iniciales es “Diseñar la interfaz de usuario de cada minijuego tomando en cuenta la usabilidad que necesitan las personas con discapacidad motriz fina.” Es por esto que este proyecto, además de ser un recurso para el aprendizaje de iniciación a la programación orientada a objetos, tiene como principal característica una interfaz diseñada para personas con discapacidad motriz fina además que el sistema provee de una interfaz navegable por comando de voz, lo cual da más alternativas de uso y navegación.

En resumen, podemos concluir que, de acuerdo a los resultados obtenidos de las encuestas, al haber creado una historia para dar a conocer los conceptos de la programación orientada a objetos se tuvo una gran aceptación por parte de los participantes, además de que la gran mayoría de ellos comentan haber comprendido los conceptos que se explicaron, que el sistema es fácil de comprender en qué parte de entre se encuentra y las acciones que puede realizar y asimismo la navegación a través del sistema web de aprendizaje desarrollo es bastante simple y puede considerarse que es fácil de usar para usuarios con discapacidad motriz con limitaciones en las extremidades superiores que es al grupo de usuarios al cual principalmente va dirigido.

Como trabajo futuro para mejorar la experiencia de los usuarios se pretende montar el sistema en un servidor con un servicio en la nube autoescalable, además de buscar alternativas de formato, que permitan optimizar la calidad y tamaño de los archivos de las historietas. Se pretende mejorar la presentación de las historias, así como agregar la característica de poder consultarlas en cualquier momento durante el juego. Así mismo continuar con el desarrollo de contenido para explicar conceptos más avanzados de la programación orientada a objetos.

# Referencias bibliográficas

1. J.R. Campaña, A.E. Marín, M. Ros, D. Sánchez, J.M. Medina, M.A. Vila, M.D. Ruiz, M.P. Cuéllar, M.J. Martín-Bautista: Metodologías activas y gamificación en las asignaturas de iniciación a la programación. Presentada en XXII Jornadas de Enseñanza Universitaria de la Informática en la Universidad de Almería., 6-8 de Junio 2016. Disponible en: <http://hdl.handle.net/2117/90277>
2. A. Radenski: Python First: A Lab-Based Digital Introduction to Computer Science. *Proceedings of the Eleventh Annual Conference on Innovation and Technology in Computer Science Education*, ITICSE 06, University of Bologna, Italy, Junio 26-28, 2006, ACM Press, 197-201. Disponible en [https://digitalcommons.chapman.edu/scs\\_articles/206/](https://digitalcommons.chapman.edu/scs_articles/206/)
3. D.P. Corsi, “Estrategias lúdicas para la enseñanza de la programación: un análisis comparativo de su eficiencia en la Educación Superior”. Tesis doctoral, Universidad de Extremadura, España, 2019. Disponible en: <http://dehesa.unex.es/handle/10662/9274>
4. A. Firth, “Motor Disabilities”, *Practical Web Inclusion and Accessibility*, Joyce, K. London UK: Apress, 2019 cap. 4, pp. 93-134
5. C. López Escribano y R. Sánchez Montoya: “Scratch y Necesidades Educativas Especiales: Programación para todos”, *RED Revista de Educación a Distancia*, no. 34, Noviembre 2012. Disponible en <http://www.um.es/ead/red/34>
6. J.T. Beltrán Morales, “E-learning y gamificación como apoyo al aprendizaje de programación”. Tesis doctoral, TICN, Universidad de Extremadura, España, 2017. Disponible en: <http://dehesa.unex.es/handle/10662/6429>
7. G.J. Astudillo, S.G. Bast y P.A. Willging, “Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación”, *VEC Virtualidad, Educación y Ciencia*, vol. 7, no. 12, pp. 125-142, Julio 2016. Disponible en: <https://revistas.unc.edu.ar/index.php/vesc/article/view/14739>
8. R. J. Botero Tabares, “Revisión Bibliográfica de los Juegos Digitales para el aprendizaje de la Programación Orientada a Objetos”, *Cuaderno Activa*, vol.4 no. 2, pp. 91-106, Noviembre 2013, Disponible en: <http://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/37>
9. R. J. Botero Tabares, “La lúdica de juegos en el aprendizaje de la programación orientada a objetos: un prototipo en C#”, Tesis de grado, Magíster en ingeniería, Área de sistemas y computación, Universidad EAFIT, Colombia, 2012. Recuperado de: <https://repository.eafit.edu.co/handle/10784/1175>
10. D.J. Lancheros C., A. Carrillo Ramos y J. L. Lara, “Modelo de adaptación en ambientes virtuales de aprendizaje para personas con discapacidad”, *Avances en Sistemas en Informática*, Universidad Nacional de Colombia, vol. 8, no. 2, pp. 17-30, Mayo 2011. Disponible en: <https://revistas.unal.edu.co/index.php/avances/article/view/26722>
11. G. Norero Ochoa, “Aprendizaje basado en juegos para la enseñanza de la programación orientada a objetos y el diseño multimedia”, *DAYA diseño, arte y arquitectura*, vol.1 no. 3, pp. 21-32, Diciembre 2017. Disponible en: <http://revistas.uazuay.edu.ec/index.php/daya/article/view/96>

12. L.A. Centeno Bragado, A.M. Pisabarro Marrón, C.E. Vivaracho Pascual, "Virtualización de un juego serio para el aprendizaje de la programación", Trabajo de fin de grado, Ing. Informática, Universidad de Valladolid, Valladolid, España, 2018. Disponible en: <https://uvadoc.uva.es/handle/10324/32917>
13. K.M. Kapp, *The Gamification of learning and instruction: Games-based methods and strategies for training and education*. San Francisco CA: Pfeiffer 2012
14. M. Csikszentmihalyi, *Flow the psychology of optimal experience*. HarperCollins Publishers 2008
15. Organización Mundial de la Salud & Banco Mundial. 2011. Informe mundial sobre la discapacidad 2011. Disponible en: <https://apps.who.int/iris/handle/10665/75356>
16. INEGI. Clasificación de tipo de discapacidad - Historica, Disponible en: [https://www.inegi.org.mx/contenidos/clasificadoresycatalogos/doc/clasificacion\\_de\\_tipo\\_de\\_discapacidad.pdf](https://www.inegi.org.mx/contenidos/clasificadoresycatalogos/doc/clasificacion_de_tipo_de_discapacidad.pdf)
17. Secretaría de Educación Pública. Discapacidad motriz, guía didáctica para la inclusión en educación inicial y básica. Gobierno de México, SEP Secretaría de Educación Pública. Disponible en: [https://www.educacionespecial.sep.gob.mx/2016/pdf/discapacidad/Documentos/Atencion\\_educativa/Motriz/2discapacidad\\_motriz.pdf](https://www.educacionespecial.sep.gob.mx/2016/pdf/discapacidad/Documentos/Atencion_educativa/Motriz/2discapacidad_motriz.pdf).
18. MDN contributors, "JavaScript", MDN Web Docs Mozilla. Disponible en: <https://developer.mozilla.org/es/docs/Web/JavaScript>
19. MDN contributors, "Acerca de JavaScript ¿Qué es JavaScript?", MDN Web Docs Mozilla. Disponible en: [https://developer.mozilla.org/es/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/es/docs/Web/JavaScript/About_JavaScript)
20. Facebook Open Source, "React", Meta Platforms Inc. Disponible en: <https://reactjs.org/>
21. XTech, "Ant Design of React", Ant Design 2022. Disponible en: <https://ant.design/docs/react/introduce>
22. Red Hat, Inc., "El concepto de Linux", Red Hat, 2018. Disponible en: <https://www.redhat.com/es/topics/linux>
23. ChrisM, BeginnersGuide/Overview, Python™, Septiembre 2019. Disponible en: <https://wiki.python.org/moin/BeginnersGuide/Overview>
24. P. Giampedraglia, "Why we choose Python as a backend language", *Asap*, Noviembre 2019. Disponible en: <https://www.asapdevelopers.com/python-backend-language/>
25. Python Basics, "What is Flask Python", Python Basics. Disponible en: <https://pythonbasics.org/what-is-flask-python/>
26. OCI, "What is a Database?", Oracle. Disponible en: <https://www.oracle.com/database/what-is-database/>
27. P. Deitel, H. Deitel, A. Deitel, *Cómo programar Internet & World Wide Web*. Quinta edición, México: Pearson Education, 2014
28. S. Sam, "Types of databases", Tutorials Point, Julio 2018, Disponible en: <https://www.tutorialspoint.com/Types-of-databases>
29. The PostgreSQL, "About", PostgreSQL. Disponible en: <https://www.postgresql.org/about/>
30. S. White, K. Sharkey, G. Tapizquent, M. Satran, "Propiedades ACID", Microsoft, Septiembre 2021. Disponible en: <https://docs.microsoft.com/es-es/windows/win32/cosxdk/acid-properties>
31. Docker Inc., "Developers bring their ideas to life with Docker", Docker Inc, Disponible en: <https://www.docker.com/why-docker/>
32. Docker Inc., "Use containers to Build, Share and Run your applications", Docker Inc, Disponible en: <https://www.docker.com/resources/what-container/>

33. J Milutinovich y K. Bondareva, "*What is UML*". UML Unified Modeling Language, Julio 2005. Disponible en: <https://www.uml.org/what-is-uml.htm>
34. Microsoft 365 Team, "La guía sencilla para la diagramación de UML y el modelado de la base de datos", Microsoft, Septiembre 2019. Disponible en: <https://www.microsoft.com/es-mx/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling>
35. C. Drummond, "Scrum, Aprende a utilizar scrum con lo mejor de él". Atlassian, Disponible en: <https://www.atlassian.com/es/agile/scrum>
36. J. Berman, "Metodología ágil a escala, un paso más allá: Ampliar la metodología ágil en las grandes organizaciones", Atlassian. Disponible en: <https://www.atlassian.com/es/agile/agile-at-scale>
37. N. Horquin Enzo, "Aplicación de Scrum en equipos unipersonales", Trabajo fin de carrera de Ingeniería de Sistemas, Departamento de Computación y Sistemas, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, 2020. Disponible en: <https://www.ridaa.unicen.edu.ar/xmlui/bitstream/handle/123456789/2379/Tesis%20Horquin.pdf?sequence=1&isAllowed=y>
38. nexxtway, "react-rainbow". Github. Disponible en: <https://github.com/nexxtway/react-rainbow>

# Anexos

## Anexo 1

### Tipos de base de datos

Hay diversos tipos de base de datos. A continuación se mencionan algunos de los tipos de base de datos más conocidos [26].

- **Base de datos Relacional:**

Los elementos en una base de datos relacional están organizados como un conjunto de tablas con columnas y filas [26]. Las columnas tienen una entrada para datos de una categoría en específico y las filas contienen una instancia para ese dato definido de acuerdo a la categoría. En las bases de datos relacionales se utiliza el lenguaje SQL (Structured Query Language) para realizar (consultas) *queries* para solicitar información y/o manipular información que satisfaga ciertos criterios [27]. La tecnología de base de datos relacional provee de forma más eficiente y flexible de acceder a información estructurada [26].

- **Bases de datos Orientado a objetos**

La información en una base de datos orientada a objetos es presentada en forma de objetos, como en programación orientada a objetos [26]. Una base de datos orientada a objetos se encuentra organizada en torno a objetos en lugar de acciones, y en datos en lugar de lógica. Por ejemplo, un registro multimedia en una base de datos relacional puede ser definido como un dato objeto, en lugar de un valor alfanumérico [28].

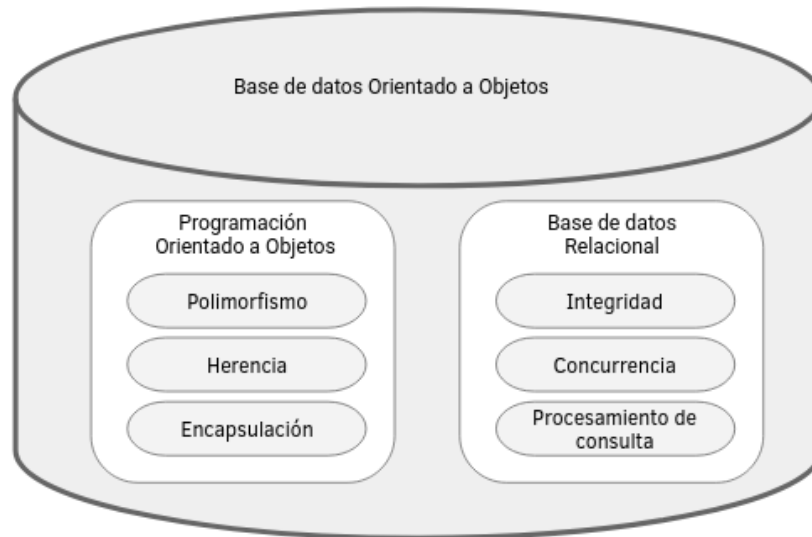


Fig. 47. Ejemplo de un diagrama de base de datos orientado a objetos (adaptado) [28].

- **Base de datos objetos-relacional**

Este modelo es una combinación de la base de datos relacional y la base de datos orientada a objetos. Por lo tanto acepta objetos, clases, herencia entre otros elementos orientados a objetos, de igual modo admite tipos de datos, estructuras tabulares y más como en el modelos de datos relacionales [34].

- **Bases de datos distribuidas**

Una base de datos distribuida consiste de dos o más archivos alojados en diferentes sitios. La base de datos podría ser almacenada en múltiples computadoras, ubicadas en la misma ubicación física o esparcidas en diferentes redes. Se puede pensar en una base de datos distribuida como tener varias partes de una base de datos que se encuentran almacenados en múltiples ubicaciones (físicas) diferentes junto con los procedimientos de aplicación que se replican y distribuyen entre varios puntos en una red. Existen dos tipos de base de datos distribuidos: homogéneo y heterogéneo. La base de datos que tiene el mismo hardware y corre el mismo sistema operativo y procedimientos de aplicación es conocida como DDB (Base de datos Distribuida) homogénea. Mientras que si el sistema operativo, el hardware al igual que los procedimientos de aplicación son ser diferentes se trata de una base de datos distribuida (DDB) heterogénea.

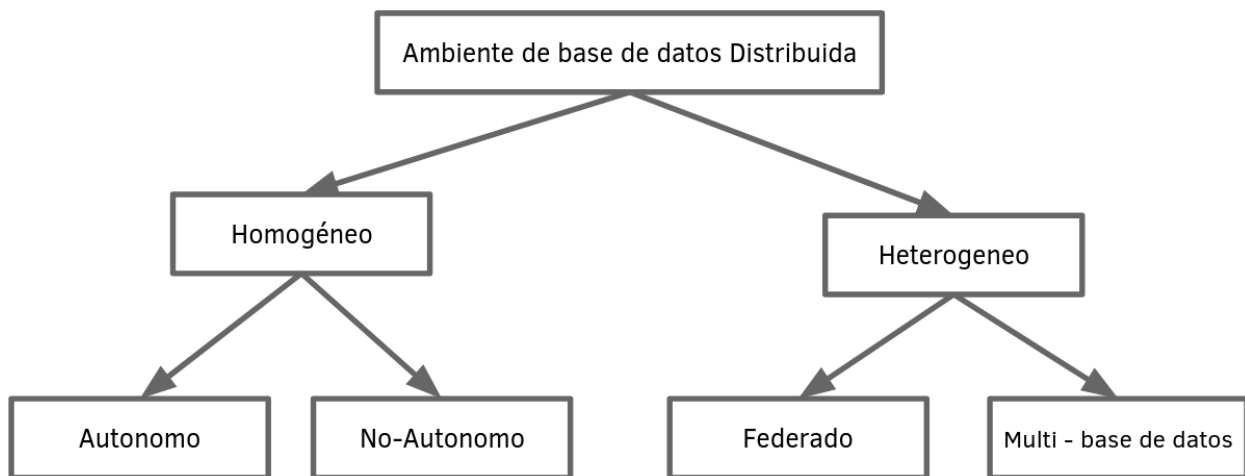


Fig. 48. Clasificación de tipo de base de datos distribuidas (adaptado) [28]

- **Data warehouse:**

Un repositorio central de datos, es un tipo de base de datos específicamente diseñado para la rápida consulta y análisis.

- **Base de datos NoSQL:**

Un NoSql o una base de datos no relacional, permite datos desestructurados o semiestructurados para ser almacenada y manipulada [26] . Se utilizan para grandes conjuntos de datos distribuidos. Son eficientes en el análisis de datos no estructurados de gran tamaño que pueden almacenarse en múltiples servidores. [28]

- **Base de datos de grafos:**

Una base de datos de grafos almacena datos en términos de entidades y las relaciones entre entidades. El grafo es una colección de nodos y bordes donde cada nodo es usado para representar una entidad y cada borde describe la relación entre entidades. Una base de datos orientada a grafos o una base de datos de grafos es un tipo de base de datos NoSQL que usa la teoría de grafos para guardar, mapear y consultar relaciones. Básicamente una base de datos de grafos es usada para analizar interconexiones. Por ejemplo, una compañía podría usar este tipo de base de datos para extraer datos acerca de los clientes de las redes sociales.



Fig. 49. Ejemplo de una base de datos de grafos (adaptado) [28]

## Anexo 2

### Tipos de diagramas UML

Los diagramas UML se pueden dividir en los siguientes dos tipos: *diagramas estructurales* y *diagramas de comportamiento* [34].

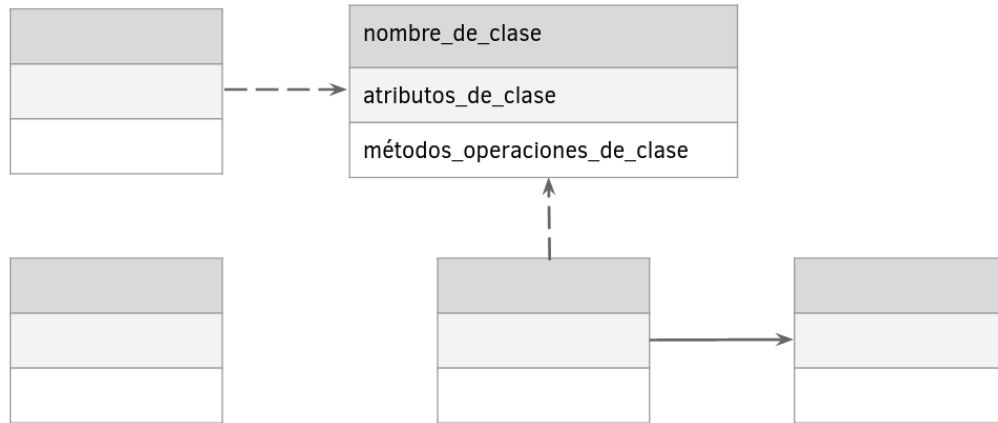
El objetivo del UML es que exprese visualmente diagramas que sean fáciles de entender para todos. Es por eso que desde clientes, director de proyectos, técnicos, diseñadores, codificadores, encargados de pruebas y control de calidad utilizan un diagrama específico que se adapte a sus necesidades dependiendo del enfoque y nivel de detalle que se requiera.

#### Diagramas estructurales

Representan la estructura estática de un software o sistema, asimismo muestran diferentes niveles de abstracción e implementación. “Se usan para ayudar a visualizar las diferentes estructuras que conforman un sistema por ejemplo una base de datos y/o aplicación. Muestran la jerarquía de componentes o módulos y cómo se conectan e interactúan entre sí. Estas herramientas ofrecen orientación y garantizan que todas las partes de un sistema funcionen según lo previsto en relación con todas las demás partes” [34]. A continuación se describen brevemente algunos diagramas de este tipo.

- **Diagrama de clases**

Es el más común en el desarrollo de software, se emplea para representar el diseño lógico y físico de un sistema. “Tiene un aspecto similar al del diagrama de flujo porque las clases se representan con cuadrados. Este diagrama ofrece una imagen de las diferentes clases y la forma en la que se interrelacionan, y cada clase posee tres comportamientos, empezando por la parte superior a inferior, son: nombre de la clase, atributos de clase y por último métodos u operaciones de clase.



**Fig. 50.** Ejemplo de diagrama UML de clases (adaptado) [34]

- **Diagrama de objetos**

“A menudo, este diagrama se usa como una forma de comprobar la revisión de un diagrama de clases para fines de precisión. En otras palabras, ¿funcionará en la práctica? Muestra los objetos de un sistema y sus relaciones, y ofrece una mejor visión de los potenciales defectos de diseño que necesitan reparación.”

- **Diagrama de componentes**

“También conocido como diagrama de flujo de componentes, muestra agrupaciones lógicas de elementos y sus relaciones. En otras palabras, ofrece una vista más simplificada de un sistema complejo al desglosarlo en componentes más pequeños. Cada una de las piezas se muestra con una caja rectangular, que tiene su nombre escrito dentro. Los conectores definen la relación/las dependencias entre los diferentes componentes.”

- **Diagrama de estructura compuesta**

“Este lo utilizan rara vez las personas externas al campo de desarrollo de software. ¿Por qué? Aunque es similar a un diagrama de clases, adopta un enfoque más profundo, que describe la estructura interna de múltiples clases y muestra las interacciones entre ellas. Salvo que usted sea desarrollador, la vista de nivel superior probablemente le entregará información suficiente.”

- **Diagrama de despliegue**

“Este diagrama muestra los componentes de hardware (nodos) y software (artefactos) y sus relaciones. Ofrece una representación visual exacta del lugar donde se implementa cada componente de software.”

- **Diagrama de paquetes**

“Este se utiliza para representar las dependencias entre los paquetes que componen un modelo. Su objetivo principal es mostrar la relación entre los diversos componentes grandes que forman un sistema complejo.”

- **Diagrama de perfiles**

“Este es más similar a un lenguaje que a un diagrama. Un diagrama de perfil ayuda a crear nuevas propiedades y semántica para los diagramas UML al definir estereotipos personalizados, valores marcados y restricciones. Estos perfiles le permiten personalizar un metamodelo de UML para diferentes plataformas (por ejemplo, Java Platform, Enterprise Edition (Java EE) o Microsoft .NET Framework) y dominios (por ejemplo modelado de proceso empresarial, arquitectura orientada a servicios, aplicaciones médicas y más).”

## **Diagramas de comportamiento**

Se centra en aspectos dinámicos de sistema de software o proceso. Estos diagramas tienen la finalidad de mostrar la funcionalidad de un sistema y hace énfasis en lo que debe ocurrir en el sistema que se está modelando.

## **Diagramas UML de comportamiento**

- **Diagrama de actividades**

“Representa un proceso paso a paso con un inicio y final claros. Es un conjunto de actividades que deben realizarse para lograr un objetivo. Muestra cómo cada actividad conduce a la siguiente y cómo todas éstas se conectan. Además del desarrollo de software, éstas se pueden utilizar en casi cualquier entorno empresarial. También se denominan asignación o modelado de proceso empresarial.”

- **Diagrama de casos de uso**

“Éste describe las funciones del sistema, pero no la forma en que las hace. Un caso de uso es un conjunto de eventos que ocurren cuando un “actor” usa un sistema para completar un proceso. Un actor se define como cualquier persona o cualquier cosa que interactúa con el sistema (persona, organización o aplicación) desde fuera del sistema. Por lo tanto, un diagrama de casos de uso describe visualmente ese conjunto de secuencias y representa los requisitos funcionales del sistema.”

- **Diagrama de descripción general de interacción**

“Este diagrama, a menudo complejo, es similar al diagrama de actividad, ya que ambos muestran una secuencia paso a paso de las actividades. Sin embargo, un diagrama de descripción general de interacción es un diagrama de actividad que se compone de diferentes diagramas de interacción. Usan la misma

composición que un diagrama de actividad (nodos inicial, final, decisión, unión) e incorpora elementos como la interacción, el uso de la interacción, restricción de tiempo y restricción de la duración.”

- **Diagrama de tiempo**

Cuando el tiempo ocupa un lugar central, se usa este diagrama de UML. También conocido como un diagrama de secuencia o eventos, no muestra la forma en que los objetos interactúan o cambian entre sí. Funcionalmente, muestra cómo los objetos y actores se desempeñan en una línea de tiempo. El enfoque aquí está en la duración de los eventos y los cambios que se producen en función de las restricciones de duración. Las principales partes de un diagrama de tiempos incluye:

- **Línea de vida:** participante individual.
- **Línea de tiempo de estado:** estados diferentes por los que pasa la línea de vida dentro de una canalización.
- **Restricción de duración:** tiempo necesario para que se cumpla una restricción.
- **Restricción de tiempo:** un periodo en el que el participante debe completar una acción.
- **Destrucción:** cuando finaliza la línea de vida de un objeto. Después de que se realiza la destrucción en una línea de tiempo, no se produce otra ocurrencia.

- **Diagrama de máquina de estados**

“También denominado gráfico de estados, este diagrama se aplica cuando el comportamiento de un objeto es complejo y el detalle es esencial. Ayuda a describir el comportamiento de un objeto (o a veces de un operador) y la forma en que cambia según los eventos internos y externos.”[34].

- **Diagrama de secuencia**

“Popular más allá de la comunidad de diseño, este diagrama visualmente atractivo es bueno para mostrar todo tipo de procesos empresariales. Simplemente revela la estructura de un sistema, mostrando la secuencia de mensajes e interacciones entre actores y objetos cronológicamente. Los diagramas de secuencia muestran iteraciones y ramificaciones simples. Es favorable al realizar múltiples tareas.”[34]

- **Diagrama de comunicación**

“Un diagrama de comunicación o colaboración es similar a un diagrama de secuencia. Sin embargo, enfatiza la comunicación entre objetos. Muestra la organización de los objetos que participan en una interacción y presenta iteraciones y ramificaciones más complejas.”[34]

### Anexo 3

**Cuestionario CSUQ de usabilidad en sistemas informáticos adaptado al español, rescatado de Revista Iberoamericana de las Ciencias Computacionales e Informática Adaptación al español del Cuestionario de Usabilidad de Sistemas Informáticos CSUQ**

	Totalmente en desacuerdo				Totalmente de acuerdo		
	1	2	3	4	5	6	7
1 En general, estoy satisfecho con lo fácil que es utilizar este sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2 Fue simple usar este sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3 Soy capaz de completar mi trabajo rápidamente utilizando este sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4 Me siento cómodo utilizando este sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5 Fue fácil aprender a utilizar este sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6 Creo que me volví experto rápidamente utilizando este sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7 El sitio web muestra mensajes de error que me dicen claramente cómo resolver los problemas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8 Cada vez que cometo un error utilizando el sitio web, lo resuelvo fácil y rápidamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9 La información (como ayuda en línea, mensajes en pantalla y otra documentación) que provee este sitio web es clara.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10 Es fácil encontrar en el sitio web la información que necesito.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11 La información que proporciona el sitio web fue efectiva ayudándome a completar las tareas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12 La organización de la información del sitio web en la pantalla fue clara.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13 La interfaz del sitio web fue placentera.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14 Me gustó utilizar el sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15 El sitio web tuvo todas las herramientas que esperaba que tuviera.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16 En general, estuve satisfecho con el sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Anexo 4

**Cuestionario creado para la segunda parte del test aplicado a los participantes que utilizaron el software desarrollado en este trabajo. Creación propia**

	Totalmente en desacuerdo	en	Totalmente de acuerdo				
	1	2	3	4	5	6	7
17 ¿El contenido fue claro?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18 ¿La historia fue entretenida?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19 ¿Consideras que los ejercicios fueron incrementando de complejidad?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20 ¿Comprendiste el concepto de atributo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21 ¿Comprendiste el concepto de Objeto?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22 ¿Consideran agregan más ejercicios para comprender el tema y alcanzar el objetivo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23 Las opciones de usabilidad para personas con capacidades diferentes te ayudaron a "moverte" en el contenido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>