

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**TÍTULO: “ODISEA: JUEZ VIRTUAL EN LÍNEA PARA
COMPETENCIAS ALGORÍTMICAS ORIENTADO A NIÑOS Y
ADOLESCENTES”**

PARA OPTAR AL TÍTULO EN LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: FABIOLA VANESSA ALIAGA SALVATIERRA

TUTOR METODOLÓGICO: M. Sc. ROSA FLORES MORALES

ASESOR: LIC. JORGE HUMBERTO TERÁN POMIER

LA PAZ - BOLIVIA
2020



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria

A mi hija, el amor de mi vida, Vanessa Castro por ser mi eterna inspiración en la vida, por ayudarme a forzarme a mí misma a ser mejor como persona en todos los ámbitos posibles gracias a todo el amor que siento hacia ella.

A mi mamá Maria Elena Salvatierra Vargas por motivarme a salir siempre adelante a pesar de las circunstancias, por inculcarme todos los valores que tengo como persona y brindarme su apoyo incondicional en todo momento.

A mi papá Miguel Angel Hipólito Aliaga Muñoz por todas los consejos y apoyo incondicional que me brinda en la vida.

A mis hermanos Ángela y Mario por todo el cariño y apoyo que me dan en el momento que más lo necesito.

A Jhonatan Castro por los consejos y el apoyo brindado.

Los adoro.

Agradecimientos

A mis papás y a mis hermanos por todo el apoyo brindado, comprensión y confianza que tienen hacia mí. A mamá María Elena Salvatierra por tomarse el tiempo para revisar y hacer seguimiento del presente proyecto de tesis de grado. Sin ellos este logro no sería posible.

A Jhonatan Castro por todos los consejos a nivel académico y profesional, también a su mamá Rosario Rocabado por todo el apoyo brindado sobretodo al momento de culminar mi carrera universitaria.

A la M. Sc. Rosa Flores Morales como mi tutora metodológica, por todo su apoyo, sus enseñanzas, su exigencia y su paciencia, mismos que fueron indispensables para el desarrollo del presente proyecto de Tesis.

Al Lic. Jorge Terán Pomier como mi asesor, por todo el conocimiento impartido y los consejos que ayudaron a llevar a cabo el presente proyecto de tesis.

A mis compañeros de universidad por todo el compañerismo y por la linda amistad a lo largo de la carrera universitaria.

¡GRACIAS!

Índice de contenidos

Capítulo 1 Introducción.....	1
1.1. Problemática	2
1.2. Objetivo general	3
1.2.1 Objetivos específicos	3
1.3. Metodología.....	4
1.3.1. Tipo de Investigación.....	4
1.3.2 Metodología de desarrollo	4
1.3.3. Población de evaluadora	4
1.4. Límites y alcances	4
1.4.1 Límites	4
1.4.2 Alcances	5
1.5. Justificación	5
1.5.1 Justificación Social	5
1.5.2 Justificación Tecnológica.....	5
1.5.3 Justificación Económica	5
Capítulo 2 Marco teórico	6
2.1 Competencias de programación.....	6
2.1.1 Inicios de las competencias de programación.....	6
2.1.2 Olimpiada Internacional de Informática	7
2.1.3 Competencias en Bolivia	7
2.2 Sistema de juez en línea.....	8
2.3 Jueces en línea con más popularidad.....	9
2.4 La importancia que tienen los jueces en línea en la educación	13
2.5 Los niños, los videojuegos y la programación	14
2.6 El uso de los jueces en línea en el ámbito profesional	16
2.7 Método de evaluación del motor de un sistema de juez virtual.....	17
2.8 Metodología de desarrollo ágil SCRUM	23
2.8.1 Scrum Team	23
2.8.2 Flujo de la metodología SCRUM	23

2.8.3 Tecnologías utilizadas para el desarrollo del sistema de juez en línea Odisea	24
Capítulo 3 Desarrollo del sistema “Odisea”	26
3.1 Requisitos	26
3.1.1 Requisitos funcionales	26
3.2 Diagrama de casos de uso.....	30
3.3 Diagrama de secuencia	30
3.4 Metodología ágil SCRUM.....	31
3.4.1 Roles de usuarios dentro de la metodología Scrum.	31
3.4.2 Descripción de privilegios de usuario.....	32
3.4.3 Pila de productos (Product Backlog)	33
3.4.4 Primer sprint.....	35
3.4.5 Segundo sprint	46
3.4.6 Tercer Sprint	56
Capítulo 4 Pruebas y Resultados	63
4.1. Estudio de caso	63
4.2. Inventario de los resultados	65
4.2.1. Encuestas preparadas para la población niños antes de realizar el software.....	65
4.2.2. Encuestas preparadas a la población niños.	68
4.2.3. Encuestas preparadas para usuarios expertos.....	74
4.3. Resultados alcanzados	78
Capítulo 5 Conclusiones y Recomendaciones	79
5.1 Conclusiones.....	79
5.2 Recomendaciones	80
Bibliografía.....	80

Índice de figuras

Figura 1: Proceso Scrum Fuente: Codejavu	24
Figura 2: Diagrama de casos de uso.....	30
Figura 3: Diagrama de secuencia	31
Figura 4: Colección de datos para Usuario	39
Figura 5: Colección de datos para Problemas	39
Figura 6: Colección de datos para Problema.....	39
Figura 7: Colección de datos para proceso de envío de problemas	40
Figura 8: Tablero Kanban primera iteración de sprint	40
Figura 9: Tarjeta de tareas de historia de usuario N°1	41
Figura 10: Tarjeta de tareas de historia de usuario N°2	42
Figura 11 : Tarjeta de tareas de historia de usuario N°3	43
Figura 12: Tarjeta de tareas de historia de usuario N°4	44
Figura 13: Tarjeta de tareas de historia de usuario N°5	45
Figura 14: Tarjeta de tareas de historia de usuario N°6	46
Figura 15. Colección de datos para módulo categoría	50
Figura 16. Colección de datos para módulo tutorial	50
Figura 17. Colección de datos de módulo concursos	51
Figura 18. Colección de datos para la sección de ranking	51
Figura 19: Tablero Kanban segunda iteración	52
Figura 20: Tarjeta de tareas de historia de usuario N°7	53
Figura 21: Tarjeta de tareas de historia de usuario N°8	54
Figura 22: Tarjeta de tareas de historia de usuario N°9	55
Figura 23: Tarjeta de tareas de historia de usuario N°10	56
Figura 24. Colección de datos para Usuario	59
Figura 25. Colección de datos para la sección de Jugabilidad.....	59
Figura 26: Tablero kanban de la tercera iteración	60
Figura 27: Tarjeta de desarrollo de historia de usuario N° 11	61
Figura 28: Tarjeta de desarrollo tarjeta de usuario N°12	62
Figura 29. Histograma de la edad de los encuestados.....	65

Figura 30. Gráfico pastel de proporciones de jueces virtuales.....	66
Figura 31. Respuestas de los encuestados	66
Figura 32. Gráfica de pastel de proporciones de respuestas de encuesta realizada.	67
Figura 33. Gráfica de pastel de proporciones sobre lenguaje de programación.	67
Figura 34. Gráfica de pastel de proporciones sobre edades de los encuestados.	68
Figura 35. Gráfica de barras de resultados de la encuesta.	69
Figura 36. Gráfica de barras de resultados de la encuesta	69
Figura 37. Gráfica de barras de resultados de las encuestas	70
Figura 38. Gráfica de barras resultados de las encuestas	71
Figura 39. Gráfico de barras de los resultados de las encuestas	72
Figura 40. Gráfica de barras de resultado de encuestas.	72
Figura 41. Gráfico de barras de resultados de encuesta.	75
Figura 42. Gráfico de barras de respuestas de encuesta	74
Figura 43. Gráfico de barras de respuestas de encuesta	74
Figura 44. Gráfico de barras de respuestas de encuesta	75
Figura 45. Gráfico de barras de resultados de encuesta	75
Figura 46. Gráfico de barras de resultados de encuesta	76
Figura 47. Gráfico de barras de respuestas de encuesta	77

Índice de tablas

Tabla 1. Matriz de trazabilidad de requerimientos.....	26
Tabla 2. Roles de usuario	31
Tabla 3. Descripción de privilegios de usuario	32
Tabla 4. Product backlog.....	33
Tabla 5. Sprint Backlog primera iteración	35
Tabla 6. Historia de usuario número uno	36
Tabla 7. Historia de usuario número dos.....	36
Tabla 8. Historia de usuario número tres	37
Tabla 9. Historia de usuario número cuatro	37
Tabla 10. Historia de usuario número cinco	37
Tabla 11. Historia de usuario número seis	38
Tabla 12. Sprint Backlog segunda iteración.....	47
Tabla 13. Historia de usuario número siete.....	48
Tabla 14. Historia de usuario número ocho	48
Tabla 15. Historia de usuario número nueve.....	49
Tabla 16. Historia de usuario número diez	49
Tabla 17. Sprint Backlog tercera iteración.....	56
Tabla 18. Historia de usuario número once.....	57
Tabla 19. Historia de usuario número doce.....	58

Resumen

El presente proyecto corresponde al estudio y desarrollo del juez en línea llamado “Odisea” que tiene como objetivo servir como material de enseñanza con ayuda de elementos de jugabilidad para incentivar el autoaprendizaje o bien para que los tutores puedan guiar a los niños y adolescentes en el área de la programación básica y competitiva, en el caso de los usuarios experimentados, el sistema tiene como objetivo entrenar para competencias o simplemente aumentar sus habilidades en el desarrollo de algoritmos.

Al concluir este proyecto se realizó un inventario de resultados en el que se observó que en la prueba de calidad del sistema un 80% de la población de niños y adolescentes afirman que el sistema es funcional respecto al envío de problemas, un 50% dice que el minijuego realiza correctamente su función, un 80% afirma que el sistema es confiable en el aspecto de consideración de errores de usuario, un 90% afirma que el sistema es usable en cuestión de facilidad de uso, un 60% afirma que el sistema es eficiente en aspectos de velocidad al momento de realizar alguna de sus funciones. Mientras tanto, los usuarios expertos afirman en un 100% que el sistema cumple con los aspectos de usabilidad en cuestión de facilidad de uso, un 70% afirma que el sistema es eficiente en aspectos de velocidad al momento de realizar alguna de sus funciones, un 100% afirma que el sistema es confiable en aspectos de corrección de errores de los usuarios.

Para tener una mejor organización al momento de realizar el desarrollo del sistema se hizo un análisis de requerimientos con la ayuda de una matriz de trazabilidad, posteriormente se realizaron diagramas de casos de uso y de secuencia, y finalmente, se realizaron tres iteraciones dentro de la metodología ágil Scrum conjuntamente con la implementación de una base de datos no relacional. Finalmente, se concluyó que los objetivos fueron alcanzados satisfactoriamente ya que el producto final cumplió los requerimientos iniciales del proyecto.

Palabras clave: juez en línea, programación competitiva, aprendizaje con elementos de jugabilidad, tutoriales de programación.

Abstract

This project is about the study and development of the online judge called "Odisea". The goal of this project is teaching material with help of gameplay elements to encourage self-learning or for tutors to guide children and adolescents in the area of basic and competitive programming, in the case of experienced users, the goal of this project is to train for competitions or simply to increase their skills in algorithm development.

At the end of this project, an inventory of results was carried out in which it was observed that in the quality test of the system, 80% of the population of children and adolescents affirm that the system is functional respect to sending problems, 50% say that The minigame performs its function correctly, 80% affirm that the system is reliable in the aspect of consideration of user errors, 90% affirm that the system is usable in terms of ease of use, 60% affirm that the system is efficient in speed aspects when performing any of its functions. Meanwhile, expert users affirm 100% that the system complies with the usability aspects in terms of ease of use, 70% affirm that the system is efficient in terms of speed when performing any of its functions, a 100% affirm that the system is reliable in terms of correcting user errors.

To have a better organization at the time of developing the system, a requirements analysis was made with the help of a traceability matrix, later use case and sequence diagrams were made, and finally, three iterations were carried out within the Agile Scrum methodology together with the implementation of a non-relational database.

Finally, it was concluded that the objectives were satisfactorily achieved since the final product met the initial requirements of the project.

Keywords: online judge, competitive programming, learning with gameplay elements, programming tutorials.

Capítulo 1

Introducción

El presente proyecto trata sobre el desarrollo de un juez en línea llamado “Odisea” fuera de lo convencional; es decir, un sistema web que aparte de cumplir su función principal que es la de evaluar automáticamente programas algorítmicos, también ofrece elementos de jugabilidad y material de enseñanza orientado a niños y adolescentes.

El interés de este proyecto responde a una experiencia personal obtenida mediante la organización de las Olimpiadas Bolivianas en Informática (OBI) y las Olimpiadas Científicas Estudiantiles (OCE) en las cuales se observó que existía falta de información respecto a lo que son los concursos de programación y su modalidad entre otras cosas. Así fue que surgió una inspiración para realizar este proyecto de Tesis.

La programación no está en la malla curricular escolar, es por eso que los niños que estén interesados en programar deben hacerlo con el mismo material provisto para los universitarios.

Aprender a programar con los métodos universitarios no es sencillo, pues es necesario tener algunos conocimientos previos, como también apoyarse de una bibliografía extensa. Es por eso que el sistema provee una sección de juegos con implementos sobre programación que se enfoca únicamente en usuarios que nunca hayan realizado ni una línea de código.

En estas últimas décadas, la tecnología fue creciendo de forma exponencial, los sistemas informáticos y las aplicaciones web tienen gran importancia y uso en el mundo; sin embargo, el retraso tecnológico y la brecha digital es evidente en Bolivia. Esta situación puede mejorar educando a los niños en el área de programación, pues son el futuro del país y del mundo entero.

“La explosión más grande de demanda de cualquier otra carrera en la historia es la de programación, esto no significa que todo el mundo deba ser un profesional en ella, pero considero que todos deben saber hacerlo” (Vega, 2018).

A lo largo del proyecto se describen los inicios de las competencias de programación, como también se menciona la lista de jueces en línea más conocidos y utilizados por los usuarios como

forma de entrenamiento de las competencias de programación; asimismo, se explican las plataformas virtuales sobre videojuegos para que el usuario pueda programar y jugar, también se explica otros usos que tienen las plataformas en línea. Otro tema interesante que se aborda es la explicación del funcionamiento del juez en línea, como la explicación de sus métodos de evaluación y la explicación matemática sobre el funcionamiento del motor de evaluación. Una vez explicada la teoría del proyecto, se describe el desarrollo de este mediante la metodología ágil Scrum. Concluido el proyecto, se presenta un estudio de resultados y pruebas de calidad. Finalmente se presentarán las conclusiones y recomendaciones para el uso adecuado del sistema.

1.1. Problemática

Durante los últimos años se vió un incremento exponencial en el número de participantes para los concursos de programación en general, tanto para universitarios como para colegiales; sin embargo, comparado con otros países latinoamericanos, Bolivia tiene menos participantes y menos nivel competitivo de programación. Según Codeforces Standings (Codeforces, s.f.), el mejor competidor de Bolivia se encuentra en el puesto número 3838 en la lista mundial.

La falta de nivel competitivo en el área de programación también se refleja en el concurso IOI (International Olympiad in informatics), este concurso está orientado a participantes que se encuentran en la etapa escolar, según las estadísticas de la página oficial de la IOI, en Bolivia nunca se ganó una medalla a comparación de Brasil que es el país latinoamericano con más medallas ganadas con un total de: 3 medallas de oro, 12 medallas de plata y 29 medallas de bronce. Mundialmente, el país con más medallas ganadas es China con un total de 84 medallas de oro, 27 medallas de plata, y 12 medallas de bronce (International Olympiad in informatics - Statistics, s.f.).

Si bien las competencias de programación en Bolivia tienen cada vez mayor cantidad de participantes, aún así no es suficiente la calidad de preparación que estos tienen.

La OBI hizo conocer de alguna forma las competencias de programación a más colegios y tutores externos a los que les interesaba enseñar programación competitiva a niños, es así que se incrementó la participación de varios niños.

Como evaluadora y parte del equipo de organización vi muchas falencias en la OBI ya que la falta de preparación de parte de los tutores era evidente, y tampoco contaban con el material necesario para brindar a sus estudiantes. Algunos niños querían participar de la competencia pero no sabían cómo manejar el sistema de juez virtual, pues no se tenía un manual de usuario del juez en línea que estaba provisto para la olimpiada, incluso no sabían cómo ingresar a la sección de los problemas propuestos. Aún así los niños tenían muchas ganas de participar y quedar bien en las olimpiadas.

Los niños que pasan clases particulares en institutos privados tienen más ventajas y privilegios que los otros niños que no pueden costear aquellas clases. En los colegios, la materia de computación no lleva programación, solo llevan alfabetización digital y no es obligatoria. Varios niños no tienen la ventaja de pasar materias de programación en el colegio.

El incentivo a la programación competitiva puede traer grandes cambios en un futuro, Bolivia podría convertirse en un país tecnológico y moderno.

1.2. Objetivo general

Desarrollar un sistema web de juez virtual en línea, orientado a niños y adolescentes con tutoriales y unidades didácticas para que el usuario tenga el material suficiente para aprender de forma autodidacta sobre programación básica y competitiva.

1.2.1 Objetivos específicos

- Establecer un sistema didáctico con interfaz de usuario amigable.
- Evaluar y calificar programas algorítmicos de forma automática.
- Designar aulas especializadas y sugerir un material de enseñanza para que maestros y tutores puedan proveer a sus alumnos.
- Sugerir un material de aprendizaje desde principiante hasta experto para el usuario que esté interesado sobre programación competitiva.
- Establecer un espacio de jugabilidad para los más pequeños que sirva a la vez como material de enseñanza sobre programación.

1.3. Metodología

1.3.1. Tipo de Investigación

Para llevar a cabo la investigación del presente proyecto se tomó en cuenta la búsqueda de artículos científicos, libros y páginas web que están referidos al tema y al desarrollo del sistema. Estos estarán relacionados con los siguientes temas:

- Programación
- Programación competitiva
- Juez virtual en línea
- Concurso de programación
- ICPC
- IOI

1.3.2 Metodología de desarrollo

Como metodología de desarrollo se utilizará SCRUM ya que tiene las siguientes ventajas: es un método de desarrollo ágil, tiene alta calidad en software, los equipos en el trabajo son integrados, tiene colaboradores satisfechos y, finalmente, es de fácil escalabilidad (Platzi, 2015).

1.3.3. Población de evaluadora

Nuestro público de evaluación son estudiantes de colegio que tengan edad a partir de los ocho años hasta los dieciocho, pueden ser estudiantes principiantes o avanzados en las competencias de programación.

Otro público de evaluación son los tutores o ex competidores que tienen experiencia previa en la programación competitiva.

1.4. Límites y alcances

1.4.1 Límites

- En el área de tutoriales no se contará con un tutorial por cada problema del juez.

- El Juez en línea contará únicamente con soporte para evaluar los siguientes lenguajes: C, C++, Java, Python.
- El Juez en línea tendrá como objetivo enfocarse a la programación competitiva y no a otro tipo de concursos de software.

1.4.2 Alcances

- Una vez concluido el desarrollo del sistema, se deberá subir el juez en línea a un dominio y que esté disponible en todo momento.

1.5. Justificación

1.5.1 Justificación Social

El juez en línea “Odisea”, aportará herramientas necesarias de manera libre y gratuita hacia toda aquella persona que tenga ganas de aprender desde programación básica hasta programación competitiva. El sistema estará activo todo el día durante todos los días, también enseñará a la población de forma didáctica ya que proveerá un pequeño videojuego de programación y tutoriales desde cómo utilizar el juez en línea hasta como crear programas informáticos.

1.5.2 Justificación Tecnológica

No será necesario tener un instalador o algún elemento que necesite ser compatible para comenzar a entrar en la aventura del juez en línea “Odisea”. Este está adaptado para cualquier navegador web.

1.5.3 Justificación Económica

La necesidad de tener un juez en línea con material didáctico de aprendizaje apoyará a los estudiantes con menos recursos a prepararse en las competencias de programación ya que este no necesita de ningún tipo de remuneración económica.

Capítulo 2

Marco teórico

En este capítulo se explicará en qué consisten los concursos de programación para tener una idea mucho más clara sobre la importancia de los jueces en línea en ese campo, después se mencionan los jueces en línea con más popularidad a nivel internacional.

Uno de los temas principales a tratar será la importancia que tienen los videojuegos en el aprendizaje de los niños y adolescentes, y a partir de ahí se hará mención a las plataformas más populares que hacen uso de la gamificación. Como complemento, se hará una pequeña referencia a los jueces en línea que son utilizados por las empresas de desarrollo software como forma de reclutamiento de empleados.

Es importante también conocer las herramientas de desarrollo que se utilizará para el desarrollo del juez en línea Odisea y la metodología que se aplica para el desarrollo de este sistema.

2.1 Competencias de programación

2.1.1 Inicios de las competencias de programación

La Competencia Internacional de Programación Universitaria (ICPC por sus siglas en inglés) se inicia en 1970, cuando los pioneros de Alpha Chapter de la “Sociedad de honor de ciencias en computación” (UPE) presentaron la primera competencia de este estilo. La iniciativa se propagó rápidamente hacia los Estados Unidos y Canadá como un programa de innovación para incrementar ambición, aptitud de resolución de problemas, y la oportunidad para estudiantes fuertes en el campo de la computación (ICPC , s.f.).

Con el tiempo, el concurso se convirtió en una competencia de varios niveles con la primera ronda de campeonato realizada en 1977; desde entonces el concurso se ha expandido a una colaboración mundial de universidades que albergan competencias regionales que hacen avanzar a los equipos a la ronda anual de campeonato mundial, el ICPC World Finals (ICPC , s.f.).

El concurso fomenta la creatividad, el trabajo en equipo y la innovación en la creación de nuevos programas de software y permite a los estudiantes evaluar su capacidad para desempeñarse

bajo presión. El concurso ha elevado las aspiraciones y el rendimiento de generaciones de solucionadores de problemas del mundo en ciencias de la computación e ingeniería.

Los participantes conforman equipos de tres personas, cada equipo representa a su universidad y país respectivamente. El concurso dura cinco horas, tiempo en que los estudiantes deben resolver de ocho a trece problemas algorítmicos. Un equipo es ganador cuando éste resuelve la mayor cantidad de problemas en el menor tiempo de penalización (Wasik et al. 2018).

Desde entonces se crearon varias competencias de programación de ese mismo estilo, la más popular es la Olimpiada Internacional de Informática (IOI).

2.1.2 Olimpiada Internacional de Informática

El inicio de la Olimpiada Internacional de Informática (IOI) corresponde al mes de octubre de 1987 propuesta por el delegado búlgaro Profesor Blagovest Sendov. En mayo de 1989, la UNESCO inició y patrocinó la primera Olimpiada Internacional de Informática (IOI) que se celebró en Bulgaria (International Olympiad in Informatics, s.f.).

La IOI se organiza anualmente, la ubicación va cambiando y se la realiza en uno de los países participantes; cada país envía una delegación de cuatro concursantes y dos adultos acompañantes. Los estudiantes compiten individualmente y tratan de maximizar su puntaje resolviendo un conjunto de problemas algorítmicos durante dos días de competencia, los días restantes se organizan eventos culturales y recreativos.

Los desafíos dentro de la competencia son de naturaleza algorítmica; sin embargo, los concursantes tienen que mostrar habilidades básicas de TI como análisis de problemas, diseño de algoritmos y estructuras de datos, programación y pruebas (International Olympiad in Informatics, s.f.).

2.1.3 Competencias en Bolivia

En Bolivia también se llevan a cabo las competencias de programación en línea, por ejemplo:

- **Olimpiada Boliviana de Informática:** Se fundó el año 2011, estas Olimpiadas están orientadas a niños de tercero hasta sexto de secundaria donde los primeros veinte clasificados pasan a formar un equipo que después de un año de capacitación compiten para los cuatro puestos para representar a Bolivia en la competencia IOI mencionada anteriormente (Olimpiada Boliviana de Informática, s.f.).
- **Red de programación competitiva:** Es un espacio de entrenamiento a través de un concurso. Se fundó hace siete años en Colombia, en el cual, cualquier persona o grupos de tres personas pueden inscribirse libremente y participar, la evaluación es de la misma modalidad de la ICPC. Cada una de las universidades bolivianas tienen una sede dentro del concurso, suele llevarse a cabo en tiempo real junto con otras universidades latinoamericanas (Red de Programación Competitiva, s.f.).

2.2 Sistema de juez en línea

La clave principal para este tipo de concursos anteriormente mencionados es el sistema que verifica automáticamente la corrección de las soluciones presentadas por los participantes, también verifica que la solución no exceda los límites de utilización de recursos (como el tiempo y memoria). Según la evaluación realizada, se calcula la clasificación en línea de todos los participantes presentada en tiempo real durante el desarrollo de un concurso.

El término juez en línea fue introducido por primera vez el año 2001 por Kurnia, Lim y Cheang como una plataforma en línea que está inspirada tanto para las competencias de programación como para la educación. Kurnia, Lim y Cheang le dan forma a esta definición como una plataforma que puede evaluar automáticamente cualquier programa escrito en cierto lenguaje de programación en tiempo real (Kurnia et al. 2001).

En general, el objetivo de los sistemas de jueces en línea es evaluar los programas algorítmicos de manera segura, confiable y continua.

A continuación se describe el proceso de evaluación según (Wasik et al. 2018): 1) Envío, 2) Evaluación y 3) Puntuación.

Para cada ejecución de caso de prueba de la presentación particular, se verifica si:

- 1) El proceso de ejecución se realizó sin errores.
- 2) No se han excedido las limitaciones de recursos específicos del problema.
- 3) El resultado obtenido cumple con las reglas descritas en la definición del problema.

2.3 Jueces en línea con más popularidad

En la actualidad, existen plataformas en línea que comparten desafíos similares a los que se utilizan para las competencias de programación, muchas universidades proveen este tipo de sistemas para apoyar a sus estudiantes para la preparación de los concursos de programación.

- **UVa Online Judge:** Este juez en línea tiene sus inicios en noviembre de 1995, fue desarrollado inicialmente por un estudiante de informática para una competencia de programación que se iba a realizar en la Universidad de Valladolid en España; con el tiempo ese juez llegó a ser perfeccionado y ya no sólo se le utilizaba para las competencias de programación, sino también para el entrenamiento constante de los competidores del mundo ya que éste tenía una gran cantidad de desafíos de programación competitiva. Es el repositorio oficial para los problemas del ICPC , competencias regionales y la de los mundiales de programación (Revilla et al, 2018).

Fue así que el juez en línea UVa ganó la máxima popularidad en todo el mundo, de la mano de Miguel Ángel Revilla que imparte clases sobre algoritmos en la Universidad de Valladolid en España.

- **Codeforces:** Tiene sus inicios en el año 2010, su fundador es Mike Mirzayanov. Este juez en línea es uno de los más populares a nivel mundial, en un inicio fue preparado para los concursantes de nacionalidad rusa. Su forma de evaluación consiste en dos divisiones según su calificación Elo, que se actualiza para cada competencia realizada (Codeforces, s.f.).

Cada concurso interno tiene cinco problemas por cada división, su duración es de dos horas y el ganador es el competidor que realizó la máxima cantidad de problemas en el rango de tiempo dentro del concurso.

- **SPOJ:** Sphere Online Judge, es un juez en línea que tiene diversos desafíos de programación de todo tipo de nivel y se los puede clasificar mediante etiquetas (Sphere Online Judge (SPOJ , s.f.).
- **TopCoder:** Es una empresa con comunidad global abierta de diseñadores, desarrolladores, científicos de datos , y programadores competitivos.

En la programación competitiva, topcoder gira alrededor de una ronda de competencias en la arena SRM (plataforma de juzgamiento) cronometrados de 1,5 horas en el que todos los participantes compiten en línea que tratan de resolver el mismo conjunto de problemas tan rápidamente como sea posible. Estos fueron el primer tipo de desafíos en Topcoder (Topcoder, 2020).

- **Codechef:** Tiene sus inicios en septiembre de 2009 en Mumbai India. Se creó como una plataforma para ayudar a los programadores a triunfar en el mundo de los algoritmos, la programación de computadoras y los concursos de programación.

En CodeChef se organiza un concurso de programación a principios de mes y dos desafíos de programación más pequeños a mediados y finales de mes. Esta plataforma también tiene sesiones de capacitación y discusiones relacionadas con algoritmos, búsqueda binaria, tecnicismos como el tamaño de la matriz y los gustos. Y como complemento final, la plataforma tiene varios tutoriales de algoritmos y debates en foros para ayudar a aquellos que son nuevos en el mundo de la programación de computadoras (CodeChef , s.f).

- **E-Olymp:** En este sitio se puede aprender a resolver problemas de programación, participar en concursos de programación, así como estudiar y realizar lecciones de informática.

El sitio proporciona una base de datos de problemas, materiales educativos y un sistema de prueba automático. Los usuarios pueden resolver problemas escribiendo un pequeño programa de solución en uno de los lenguajes de programación y enviándolo al sistema de prueba automatizado. El sistema automático verificará la solución ejecutándola en varios conjuntos de pruebas y le dará un puntaje. Puede encontrar más información en la sección de Ayuda.

E-Olymp fue creado en 2007 para preparar a los estudiantes para las olimpiadas en informática. Desde entonces, hemos podido reunir una base de datos única de problemas y materiales educativos, crear un sistema de evaluación automática, crear una comunidad de maestros y estudiantes y hacer que el aprendizaje sea más fácil, más divertido y accesible para todos.

El sitio existe gracias a un grupo de voluntarios que trabajan en la creación y publicación de problemas y materiales educativos, apoyan a la comunidad y brindan apoyo técnico y financiero para el sitio web. El sitio web no es propiedad de ninguna organización comercial o gubernamental. El sitio cuenta con el apoyo financiero de Eolymp Charitable Found (About project - E-Olymp , s.f).

- **ACM-ICPC Live Archive:** Este sitio web tiene la misma estructura de los ya mencionados anteriormente, es decir, cuenta con múltiples desafíos para que los estudiantes puedan entrar libremente a resolverlos y así darles un apoyo. Lo que hace distinto a este sistema es que los desafíos son únicamente problemas de competencias pasadas de la ICPC en todas sus categorías desde que se comenzó a tener registro de esta (ICPC Live Archive , s.f.).
- **USA Computing Olympiad (USACO):** Se fundó en 1992 por el Dr. Don Piele, profesor de matemáticas en la Universidad de Wisconsin Parkside, en este concurso participan los estudiantes de colegio, los ganadores de este concurso, son posteriormente llevados a campamentos para su entrenamiento a la IOI (USACO , s.f.).
- **Project Euler:** Es una plataforma de juez virtual que tiene varios desafíos orientados principalmente a las matemáticas, las que ayudarán a resolver estos problemas con métodos más elegantes y eficientes.

“La motivación para comenzar el Proyecto Euler, y su continuación, es proporcionar una plataforma para que la mente inquisitiva profundice en áreas desconocidas y aprenda nuevos conceptos en un contexto divertido y recreativo” (Project Euler, s.f.).

- **Juez en línea de Herbert:** Esta plataforma proporciona más de mil desafíos para resolver en lenguaje H.

El lenguaje H es un lenguaje de programación muy simple propuesto para controlar un robot llamado Herbert que fue desarrollado con el propósito del desafío, llamado

Competencia de Algoritmo, realizado durante el torneo Imagine Cup en 2008; le permite al participante fortalecer sus habilidades algorítmicas, especialmente con respecto a la búsqueda de patrones y la implementación de la recursividad (Wasik et al. 2018).

- **Juez en línea URI:** Debido a que los jueces en línea que publican desafíos recopilados de concursos de programación competitivos se han vuelto muy populares en estos días, también existen adaptaciones sofisticadas de dichos sistemas con fines educativos, por ejemplo, el Juez en línea de URI es un sistema donde los desafíos proporcionados se distinguen en ocho categorías, lo que permite a los usuarios encontrar fácilmente ejercicios de un tema determinado (Wasik et al. 2018).

Si bien estos son los jueces en línea más populares reconocidos a nivel mundial, vale la pena destacar varios jueces en línea que fueron creados exclusivamente para el entrenamiento de estudiantes de distintas universidades pero que a la vez cualquier persona que desee resolver los distintos desafíos puede ingresar de manera libre, tal como el juez en línea UVa ya mencionado anteriormente .

Por ejemplo: Juez en línea de la Universidad de Tianjin (TJU), Pekín University (PKU) Judge Online (POJ) y Timus Online Judge, AC 2333 es el sistema de jueces en línea de la Universidad Tecnológica de Ningbo(está solamente en chino), el sistema Adjule organizado por la Universidad Adam Mickiewicz, Universidad de Fuzhou (China), la Universidad de Tecnología de Harbin (China), la Universidad de Ciencia y Tecnología Electrónica de China, la Universidad del Norte de China, la Universidad de Ciencias de la Información en Cuba (sede del Caribe Juez en línea) y la Universidad de Ciencia y Tecnología de Huazhong en China (sede de HUSTOJ).

Otro evento que vale la pena mencionar es el **Concurso de resolución de problemas de Internet (IPSC)**.

En los últimos años se crearon dos grandes competencias de programación a nivel mundial que ganaron mucha fama:

- **Facebook Hacker Cup:** Esta competencia se inició el año 2011, organizada por Facebook. Esta competencia se realizaba como un medio para identificar a los mejores talentos de

desarrollo para posibles empleos en Facebook. Se lleva a cabo anualmente y tiene distintas fases de clasificación (Facebook Hacker Cup , s.f).

- **Google Code Jam:** Esta competencia se inició el año 2008, organizada por Google. Consta de cuatro rondas con desafíos nuevos en cada una de ellas, al final veinticinco participantes a mejor oportunidad de poner a prueba sus habilidades, compitiendo por premios en efectivo y el codiciado título del campeonato en las Finales Mundiales (Code Jam - Google's Coding Competitions , s.f.).

2.4 La importancia que tienen los jueces en línea en la educación

Los sistemas de juez en línea han apoyado a muchos estudiantes universitarios como también a los estudiantes que se encuentran en los últimos años del colegio. Estos sistemas están a disposición completa de los estudiantes o maestros todos los días durante todo el día.

El juez en línea funciona exitosamente para la evaluación de programas algorítmicos y no solamente en competencias de programación sino también para alguna asignatura relacionada con algoritmos de la universidad o del colegio. Se ahorra tiempo de revisión y los resultados son más exactos y precisos, es por ello que las calificaciones tienen más consistencia. Al recibir una respuesta casi instantánea, los alumnos tienen más tiempo para realizar otros problemas algorítmicos (Kurnia et al. 2001).

“La principal conclusión a la que hemos llegado es que estamos administrando una herramienta con un gran potencial no solo para la capacitación sino también para la enseñanza de la informática, una combinación de aprendizaje competitivo y cooperativo, digamos una especie de competencia colaborativa. Además, puede tener éxito donde los sistemas convencionales a menudo fallan: hacer que los estudiantes sientan curiosidad, hacer que realicen trabajos no académicos que a menudo parecen más interesantes o mejorar su creatividad. Como todos pueden acceder a estos jueces en línea y comenzar a practicar con problemas fáciles y divertidos, podría ser una buena manera de atraer a los recién llegados (por ejemplo, estudiantes de secundaria) al mundo de la programación y a las ciencias de la computación en general. Es muy importante en un mundo donde muchos de nosotros, incluidos algunos gigantes del software, tenemos la percepción de que la generación joven está perdiendo interés en la informática. En los pasos posteriores, desde la perspectiva del diseño de algoritmos, el concurso de programación es un tesoro. Parece que hay numerosas formas

de resolver el mismo problema, pero también para los ingenieros de confiabilidad de software este es el caso: hay aún más formas de no resolver el problema. La primera presentación de la mayoría de los autores es incorrecta. Toman algunas pruebas para (en la mayoría de los casos) finalmente llegar a la solución correcta.”

(Revilla et al , 2008, p.5)

2.5 Los niños, los videojuegos y la programación

Las plataformas de aprendizaje son más atractivas para los usuarios cuando estas contienen un tipo de gamificación.

Los niños del siglo XXI son nativos digitales, lamentablemente en Bolivia debido al retraso tecnológico sólo algunos niños en ciertas regiones son afortunados por tener recursos tecnológicos. Todos los niños deberían tener acceso a la tecnología. Tanto desde la escuela como desde la familia es recomendable adaptar juegos y herramientas al nuevo entorno generado por las nuevas tecnologías (Revista, s.f.).

Aprender a programar desde pequeños aporta las herramientas necesarias para el razonamiento lógico que sirve incluso para diversas situaciones en la vida. A la vez al aprender a programar se tiene una gran ventaja para triunfar en el mundo digital.

“Everybody should learn how program a computer because it teaches you how to think” (Jobs , 1996) .

Existen algunos jueces en línea que introdujeron elementos de juego adicionales para fortalecer el compromiso y la motivación de los estudiantes, por ejemplo:

- **Code.org:** Es una organización sin fines de lucro dedicada a ampliar el acceso a las ciencias de computación en las escuelas y aumentar la participación de las mujeres y las minorías que no están suficientemente representadas. El objetivo de esta organización es que cada estudiante, en cada escuela, tenga la oportunidad de aprender ciencias de la computación de la misma manera que aprende biología, química o álgebra. Se ofrece el plan de estudios más utilizado para enseñar ciencias de la computación en escuelas primarias y secundarias,

y también se organiza la campaña anual **La Hora del Código**, en la que participa el 10% de los estudiantes de todo el mundo (Code.org , s.f.).

En esta plataforma existen varias salas de actividades, por ejemplo: laboratorio de baile, en el cual mediante un programa un bot baila de acuerdo a como lo programes; Laboratorio de juegos, en el cual se desarrollan videojuegos que los niños programan; Laboratorio de aplicaciones, en el cual los niños programan una aplicación; etc.

En esta plataforma los niños interactúan directamente con el lenguaje de programación visual llamado scratch, el cual se manejan los elementos de un algoritmos de forma gráfica con el uso de bloques.

- **La hora del código:** La Hora del Código comenzó como una introducción de una hora a las ciencias de la computación, diseñada para desmitificar el "código", demostrar que todos pueden aprender los conceptos básicos y ampliar la participación en el campo de las ciencias de la computación. Desde entonces, se ha convertido en una iniciativa mundial que celebra las ciencias de la computación: comenzó con actividades de programación de una hora y se amplió a todo tipo de iniciativas comunitarias (Hora del Código: únete al movimiento, s.f.).
- **CheckiO:** Es una plataforma dedicada a la experiencia de jugar videojuegos aprendiendo programación, la codificación viene como una habilidad necesaria para progresar en un juego divertido (CheckiO - coding games and programming, s.f).

Los desafíos de CheckIo están dirigidos a Python y TypeScript, además, permite a los usuarios definir "aulas" y asignarles estudiantes particulares para apoyar el proceso de monitoreo de su progreso de aprendizaje

- **Empire of Code:** Esta plataforma viene de los desarrolladores de CheckIO como un juego de estrategia en la que también ayuda a desarrollar habilidades de programación en Python.
- **CodeHunt:** Code Hunt es el juego que acaban de lanzar en Redmond desarrollado de Microsoft en el que se nos presentan diferentes rompecabezas o puzzles dentro de algunas líneas de código que tendremos que resolver. Code Hunt nos plantea dos lenguajes de programación a elegir: Java o C#. Una vez seleccionado el lenguaje que deseemos y después de realizar un pequeño tutorial empezará el auténtico reto. Ir cazando las líneas de

código erróneas o las que son necesarias para completar el código y que la secuencia tenga sentido. De esta manera iremos superando un nivel tras otro en el que la dificultad también irá en ascenso (Code Hunt - Microsoft Research , s.f.).

- **Leek Wars:** Leek Wars es un juego de lucha de leeks solo para navegador en el que se debe programar la inteligencia artificial para derrotar a los oponentes (Leek Wars, s.f.).
- **Python Challenge:** Python Challenge es un juego en el que cada nivel se puede resolver con un poco de programación (Python). Esta plataforma de Python fue fundada por Nadav Samet. Todos los niveles pueden resolverse mediante scripts sencillos y muy cortos (About The Python Challenge , s.f.).
- **CodinGame:** Es una plataforma que cuenta con múltiples juegos en línea, los usuarios escogen uno de los juegos, posteriormente debe elegir uno de los veinte lenguajes de programación para realizar la codificación y así lograr la victoria (Coding Games and Programming Challenge, s.f.).
- **Codecademy:** Es una plataforma de aprendizaje que no tiene la misma estructura de los jueces en línea convencionales, pues tiene un entorno de práctica para los usuarios en todos los campos de la informática; sin embargo, cuenta con un entorno didáctico de aprendizaje para los usuarios que sean principiantes en esos temas, por ejemplo: Ciencias de la computación, Data Science, Desarrollo Web, también cuenta con diversas áreas para mejorar aptitudes en programación, por ejemplo: Análisis de datos con Python, Análisis de datos en R, Desarrollo de modelo de Machine Learning con python, etc (Codecademy , s.f.).
- **Exercism.io:** Esta plataforma es exclusivamente de aprendizaje en el cual el usuario puede elegir entre cincuenta lenguajes de programación que no solamente abarca algoritmos sino también aplica para otros campos de la informática. También se cuenta con mentores que apoyarán a la explicación y evaluación del usuario (Exercism, s.f.).

2.6 El uso de los jueces en línea en el ámbito profesional

Como se estudió anteriormente, los jueces en línea tienen gran importancia en las competencias de programación como también en la educación. Pero también se observan muchos otros usos que estos sistemas pueden llegar a tener.

Se centra principalmente en apoyar a los reclutadores con la entrega de desafíos de codificación totalmente personalizados dirigidos directamente a los posibles candidatos y la integración perfecta de evaluaciones en tiempo real en el proceso de reclutamiento.

Muchas empresas de desarrollo con gran reconocimiento a nivel mundial toman en cuenta las habilidades de solución de problemas algorítmicos para reclutar a sus futuros empleados desarrollo de software. Es por ello que las pruebas de ingreso son en su mayoría la solución de problemas al estilo de las competencias de programación y para que estas pruebas sean llevadas a cabo, se desarrollaron algunos jueces en línea personalizadas para este tipo de pruebas son por ejemplo, Codility.

Existen algunos jueces en línea que están enfocados en la contratación de desarrolladores. Por ejemplo: HackerRank es un juez en línea que tiene una variedad de desafíos abiertos y también realiza concursos, de los cuales, los ganadores son buscados para su contratación en cualquier empresa (HackerRank , s.f.).

Esta modalidad de contratación mediante concursos también la tiene la empresa de Google y de Facebook mencionados anteriormente, los ganadores de dichos concursos tendrán una entrevista directa para su posterior contratación.

2.7 Método de evaluación del motor de un sistema de juez virtual.

A parte de la estructura que tiene un juez en línea, el motor de evaluación es el componente más importante del sistema.

Comienza con la ejecución del compilador apropiado, éste no debe devolver ningún error de compilación y debe verificar que el tiempo y tamaño binario de salida no se exceda de las características especificadas.

En la siguiente sección se presentará un estudio formal sobre el funcionamiento del motor de un juez en línea por los autores Wasik, Antczak, Badura, Laskowski, & Sternal (Wasik et al. 2018).

La fase de evaluación suele ser la más costosa desde el punto de vista computacional. Requiere la ejecución de la solución compilada en cada instancia de prueba considerada. Cada una de estas

ejecuciones puede tomar varias docenas de minutos para problemas complejos inspirados en la industria. Aquí, se proporciona la definición formal de una instancia de prueba:

Definición de Instancia de prueba. Tenemos a Σ que denota un alfabeto usado para codificar datos de entrada y salida. La instancia de prueba $t_i \in \mathbb{T}$, donde \mathbb{T} es un conjunto de todos los considerados para las instancias de prueba de problemas particulares, se define como un triple $t_i = (d_i, o_i, p_i)$, donde $d_i = \Sigma^*$ es un dato de entrada, $o_i = \Sigma^*$ es un dato de salida de referencia, y p_i es un conjunto de parámetros pasados al motor de evaluación.

En la gran mayoría de los problemas, el alfabeto Σ que se usa para codificar datos de entrada y salida consta de dígitos, espacios y nuevos caracteres de línea $\Sigma = \{0, 1, \dots, \backslash n\}$. A veces, se extiende con letras mayúsculas y minúsculas y hasta varios caracteres especiales; sin embargo, el formato utilizado para codificar ambos tipos de datos suele ser lo más simple posible para fines de análisis. Por lo general, se representa mediante una lista de números enteros cuya estructura se define en la descripción del problema o un simple archivo de valores separados por comas que se usa a menudo en aplicaciones de minería de datos. Comúnmente, cada entrada y salida de datos está representada por un solo archivo redirigido a la entrada estándar y desde la salida estándar del programa de solución, respectivamente. Sin embargo, en caso de problemas específicos, podría suceder que se proporcionen varios archivos almacenados en el archivo único.

Cuando, de acuerdo con la definición del problema, para cada instancia de entrada d_i existe exactamente una solución correcta, esta solución puede calcularse previamente y almacenarse en el archivo de datos de salida de referencia o_i . De esta manera, el motor de evaluación calcula la solución de referencia de inmediato y luego la usa durante la verificación de cada envío de usuario. Siempre que haya muchas soluciones factibles para una sola instancia de entrada, el archivo de datos de salida de referencia puede almacenar ciertos valores precalculados que permiten reducir la complejidad computacional del proceso de evaluación. Cuando ningún dato adicional puede simplificar el proceso de evaluación, los datos de salida de referencia o_i pueden estar vacíos (es decir, $o_i = \emptyset$).

El conjunto de parámetros p_i , representa las limitaciones de recursos específicos (por ejemplo, tiempo de CPU, utilización máxima de RAM) que no se pueden exceder durante la evaluación en función de esta instancia en particular. Sin embargo, también se pueden pasar parámetros adicionales si es necesario, como un generador de números aleatorios cuando el motor permite soluciones aleatorias o un límite de tamaño máximo para los datos de salida generados como resultado de la ejecución de la solución para esta instancia en particular. Cuando el motor de evaluación está configurado para usar los límites de recursos predeterminados y no se necesitan otros parámetros, el conjunto p_i puede estar vacío (es decir, $p_i = \emptyset$).

Durante la fase de evaluación, se utiliza la solución compilada como resultado de la fase de envío. Aquí, proporcionamos la definición formal de la solución:

Definición de Solución. Una solución es una función, $b(d_i, p'_i) \rightarrow o'_i$, que representa la forma binaria de la presentación que, en base a los datos de entrada d_i , calcula los datos de salida o'_i , teniendo en cuenta los parámetros de ejecución p'_i proporcionados por el motor de evaluación.

El conjunto de parámetros de ejecución proporcionados por el motor de evaluación p'_i puede ser igual al conjunto de parámetros definidos como parte de la instancia de prueba particular (es decir, $p'_i = p_i$). Sin embargo, en particular, p'_i puede ser el subconjunto de p_i (es decir, $p'_i \subset p_i$), p'_i puede ser un conjunto de parámetros completamente diferente o p_i puede estar vacío (es decir, $p'_i = \emptyset$). A diferencia de p_i , el conjunto p'_i generalmente está vacío porque los parámetros que influyen en el proceso de evaluación están ocultos de la solución. Los parámetros de ejecución más utilizados a menudo pasados a la solución del usuario son la semilla del generador de números aleatorios y el límite de tiempo máximo que la solución no puede exceder durante su ejecución para la instancia de prueba en particular.

Definición de Motor de evaluación. Un motor de evaluación es una función, $E(b, t_i) \rightarrow (s_i, v_i, e_i)$, que ejecuta el archivo binario b , dando como entrada la instancia de prueba t_i , y devolviendo el estado de ejecución s_i , un puntaje de evaluación calculado para el resultado de la solución, $v_i \in R$ y la lista de estadísticas recopiladas para el proceso de ejecución e_i .

El estado s_i de la ejecución de envío puede tomar uno de los siguientes valores:

- Aceptado (AC) cuando la ejecución de envío finaliza con éxito sin ningún error de tiempo de ejecución, sin exceder los límites de recursos, y devuelve datos de salida factibles coherentes con el formato descrito en la descripción del problema;
- Límite de tiempo excedido (TLE) significa una ejecución incorrecta del envío debido a que excede el límite de tiempo máximo de procesamiento;
- Límite de memoria excedido (MLE) significa una ejecución incorrecta del envío debido a que excede el límite máximo de utilización de RAM;
- Respuesta incorrecta (WA) significa que el programa generó una salida de formato desconocido (es decir, el formato existente no es coherente con el formato solicitado en la descripción del problema) o no se cumplieron algunas restricciones adicionales formuladas en la descripción del problema;
- Error de tiempo de ejecución (RE) significa que se produjo un error de tiempo de ejecución durante la ejecución del envío en particular;
- El límite de salida excedido (OLE) significa que el envío excedió el límite máximo para el tamaño de los datos de salida.

Muchos sistemas de jueces en línea siguen las reglas de ICPC y evalúan los envíos en cada instancia de prueba de forma binaria, sólo como una solución correcta o incorrecta. En tal caso, el puntaje de evaluación siempre es igual a 0 (es decir, $v_i = 0$). Hay dos casos principales cuando se usa este valor. Primero, en problemas de optimización, cuando almacena el valor de una función objetivo calculada para los datos de salida obtenidos como resultado de una ejecución de envío particular en una instancia de prueba particular. Puede ser tanto un problema clásico de optimización como un problema de código de golf cuando el objetivo es optimizar el tamaño del código fuente resolviendo alguna tarea. Segundo, en las competiciones que siguen las reglas de IOI, v_i caracteriza el puntaje que el usuario obtiene por recibir el estado "Aceptado" para esta instancia en particular (es decir, $s_i = AC$). En general, diferentes instancias de prueba se pueden caracterizar por varios valores de puntaje, o incluso se pueden aplicar procedimientos de puntaje más complejos. Por ejemplo, la Olimpiada polaca en informática penaliza las soluciones que utilizan más de la mitad

del límite de tiempo al disminuir la puntuación proporcionalmente, de acuerdo con la siguiente fórmula:

$$v_i = V_i \cdot \min\left(1.0, 2.0 \cdot \frac{\Gamma_i - \tau_i}{\Gamma_i}\right)$$

Aquí, V_i denota el número máximo de puntos que se pueden otorgar para alguna instancia de prueba, Γ_i denota el límite de tiempo máximo para la solución, y τ_i denota el tiempo de procesamiento utilizado por la solución para generar la salida para la instancia particular.

Finalmente, las estadísticas e_i , recopiladas para el proceso de ejecución de la solución, generalmente incluyen información sobre los valores máximos de utilización de recursos observados durante la ejecución de envío particular (por ejemplo, consumo de tiempo y memoria). En particular, cuando el juez en línea no comparte dicha información con el usuario, la lista de estadísticas de ejecución puede estar vacía (es decir, $e_i = \emptyset$).

Puntuación

El objetivo de la fase de puntuación es calcular un estado s agregado y una puntuación de evaluación agregada v de cada envío de usuario. Estos valores se utilizan para mostrar los resultados del procedimiento de evaluación al usuario y clasificar las soluciones al problema. Si y sólo si, para todas las instancias de prueba, la solución recibió el estado AC, entonces el estado agregado también es AC. De lo contrario, el procedimiento más común es seleccionar el primer estado diferente de AC:

$$s = AC \Leftrightarrow \forall_i s_i = AC$$

$$s = s_j \Leftrightarrow (\forall_{i < j} s_i = AC) \wedge (s_j \neq AC)$$

Sin embargo, a veces se utilizan algunas variaciones, por ejemplo, devolver un estado RE cuando se produce cualquier "Error de tiempo de ejecución" antes de cualquier otro estado.

Calcular el puntaje de evaluación agregado v es relativamente simple cuando el sistema no trata con problemas de optimización. En tal caso, para sistemas que evalúan envíos de forma binaria y que siguen las reglas de ICPC, este puntaje siempre es igual a 0 (es decir, $v = 0$). De lo contrario,

generalmente es el puntaje de evaluación total calculado para todas las instancias resueltas correctamente:

$$v = \sum_{i=1}^{|T|} \left\{ \begin{array}{l} v_i, \text{ si } s_i = AC; \\ 0, \text{ de otra manera} \end{array} \right.$$

En el caso de problemas de optimización, el sistema debe considerar la puntuación calculada en función de la mejor solución entre todos los envíos o una solución de referencia calculada por el autor del problema. En la mayoría de los casos, la siguiente fórmula se usa para clasificar las soluciones presentadas para problemas en los que la función objetivo se maximiza:

$$v = \frac{100}{|T|} \sum_{i=1}^{|T|} \left\{ \begin{array}{l} \frac{v_i}{b_i} \text{ si } s_i = AC; \\ 0, \text{ de otra manera} \end{array} \right.$$

donde b_i denota el mejor puntaje de solución para la i -ésima instancia (es decir, la solución entre todas las presentaciones para las cuales el puntaje de función objetivo calculado para esta instancia particular tiene el valor óptimo). También se pueden proponer tipos similares de fórmulas para problemas de minimización, teniendo en cuenta un procedimiento de puntuación extendido y tiempos de ejecución o incluso la utilización de otros recursos del entorno informático.

Juez en línea

En base a las definiciones presentadas en esta sección, podemos definir un sistema de jueces en línea de la siguiente manera:

Definición Sistema de juez en línea. Un sistema de juez en línea es un servicio en línea que realiza cualquiera de los pasos del procedimiento de evaluación en una nube; es decir:

- (1) recopilar, compilar fuentes si es necesario y verifica la ejecutabilidad de los binarios resultantes;
- (2) evalúa las soluciones $b(d_i, p'_i)$ basándose en un conjunto de instancias de prueba, T , definidas para un problema combinatorio particular Π en un entorno de evaluación confiable y homogéneo;
- (3) calcula los estados agregados y el puntaje de evaluación v en función de los estados y puntajes para instancias particulares (es decir, s_i y v_i , donde $1 \leq i \leq |T|$) (Wasik et al. 2018, p.23).

2.8 Metodología de desarrollo ágil SCRUM

El desarrollo del juez virtual Odisea se realizará con la metodología Scrum, éste es un marco de trabajo para el desarrollo de un producto que en este caso viene a ser un sistema web. Esta es una de las metodologías ágiles más populares y usadas en proyectos de software.

Scrum se basa en la teoría de control de procesos empírica o empirismo. Esta metodología emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. (Schwaber & Sutherland, 2017).

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación.

Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación:

- Planificación del Sprint (Sprint Planning)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

2.8.1 Scrum Team

- **Product Owner:** Se encarga de transmitir las necesidades del cliente y reproducirlo en el product backlog.
- **SCRUM master:** Es el encargado de organizar los sprints y planificación de la iteración.
- **Equipo de desarrollo:** Es el equipo que está suficientemente capacitado para dar cumplimiento a los Sprints.

2.8.2 Flujo de la metodología SCRUM

Todo comienza con la reunión del cliente con el product owner, de esta reunión se obtiene el Product Backlog, después el Scrum master junto con el equipo de desarrollo se encarga de planificar el sprint y el sprint backlog para así comenzar con el corazón del Scrum que es el

desarrollo del Sprint. Mientras se lleva a cabo este proceso las reuniones diarias del equipo Scrum serán necesarias.

Cuando se acaba la fase del sprint, el equipo Scrum tendrá una reunión en la que se definirá el Sprint Review (Revisión del Sprint) y finalmente, tendrán una nueva reunión donde se evaluarán los temas a mejorar, esta reunión se llama Sprint Retrospective (Retrospectiva del Sprint).

Este proceso puede tomar la cantidad de iteraciones que sean necesarias hasta lograr el mayor objetivo. Así se ilustra el proceso de Scrum en la Figura 1.

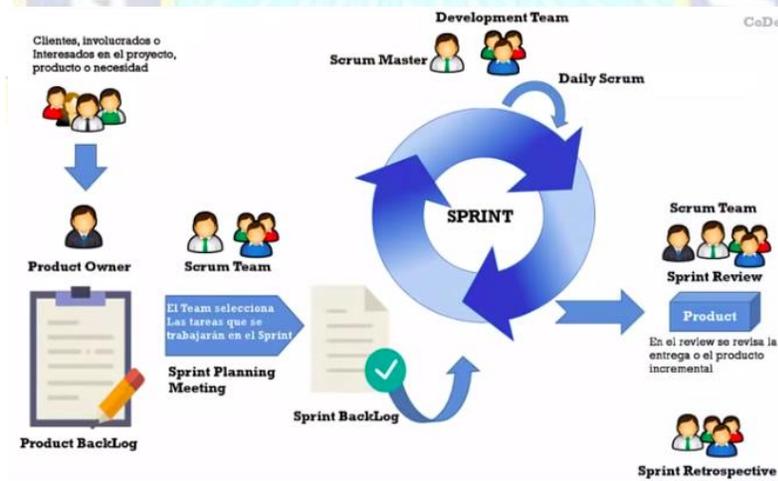


Figura 1: Proceso Scrum Fuente: Codejavu

Es de suma importancia que mientras se van desarrollando las iteraciones, sea necesario que en cada final de sprint se muestre la navegación del sistema y su correcta funcionalidad para así solucionar posibles problemas en el siguiente sprint.

Para ello se realizará un prototipo del sistema en cada final de sprint.

2.8.3 Tecnologías utilizadas para el desarrollo del sistema de juez en línea Odisea.

Las tecnologías que se utilizaron para el desarrollo del sistema Odisea son los siguientes:

- Para el desarrollo front end: Javascript, React, React Hooks, React Router, Redux, Redux Think.

- Para el desarrollo backend: Node js, Express js.
- Para el motor de evaluación: Contest Management System (CMS)
- Para la base de datos: MongoDB (base de datos no relacional).
- Herramientas adicionales: Github, Heroku, Bootstrap .
- Para el diseño de gráficos: draw.io



Capítulo 3

Desarrollo del sistema “Odisea”

En este capítulo se presentará la aplicación de la metodología de desarrollo SCRUM para el desarrollo del sistema de juez en línea Odisea.

Para una mejor organización del sistema se presentarán requisitos funcionales, requisitos no funcionales, diagrama de casos de uso y diagrama de secuencia. luego se realizará el proceso del desarrollo de la metodología ágil SCRUM, para ello se utilizaron tres iteraciones del sprint.

3.1 Requisitos

3.1.1 Requisitos funcionales

Los requisitos funcionales se desarrollaron en una matriz de trazabilidad como se muestra en la Tabla 1. ya que así se permitirá monitorear cada uno de los requisitos durante el ciclo de vida del proyecto para asegurar que se están cumpliendo de manera eficaz.

Tabla 1

Matriz de trazabilidad de requerimientos

Id	Descripción	Criterios de aceptación	Objetivo	Desarrollo o Planificación
1	El sistema debe tener registro de usuario y login del mismo.	Usuario apto para crear cuenta e iniciar en ella dentro del sistema. Inicialmente cada usuario tiene un rol de “ <i>user logged</i> ”. El usuario administrador (<i>admin</i>) será responsable de asignar privilegios.	Personalizar el sistema para todo tipo de usuario	Este requisito deberá inicializar desarrollo MERN del sistema
2	El sistema deberá tener una página principal	Al momento de ingresar al sistema el usuario deberá estar en la página de inicio de forma automática.	El objetivo de crear una página inicial será de guiar al usuario al uso del sistema.	Esta página de inicio deberá estar contenido de noticias en general, de información sobre concurso y de blogs que hayan postado algunos usuarios.

3	El sistema deberá proveer un área de tutoriales	Los tutoriales deberán abarcar varios temas respectivos a concursos de programación. Por ejemplo, tutoriales sobre diferentes algoritmos, ejercicios propuestos y soluciones	El objetivo de tener tutoriales en el sistema será para el aprendizaje de los usuarios interesados en los concursos de programación.	Será necesario que el sistema tenga una sección de tutoriales donde al ingresar a uno en particular se deberá llevar a otra página con el tutorial en texto plano.
4	El sistema deberá tener un motor de evaluación.	El sistema deberá evaluar los programas presentados por los usuarios.	El objetivo principal de todo el sistema en sí es la evaluación automática de los programas presentados por los usuarios.	El motor de búsqueda será la parte más importante en el sistema, será necesario implementar un motor ya existente que esté correctamente comprobado en su éxito respecto a la evaluación.
5	El sistema deberá tener una lista de problemas propuestos.	El sistema deberá desplegar una lista con el nombre y puntuación de cada problema.	El objetivo de desplegar una lista de problemas en el sistema es para que el usuario pueda identificar y resolver los problemas.	Se deberá desplegar una nueva página proveniente de cada sub-categoría, la cual presentará una lista de problemas. Será importante que el sistema tenga la opción de importar archivos comprimidos para la adición de nuevos problemas planteados por usuarios administradores.
6	El sistema proveerá mecanismo para clasificar los problemas de acuerdo a su tipo. Por ejemplo: Principiante, cadenas, matemáticas, etc.	El sistema deberá mostrar una sección especial de categorías en forma de recuadros con imágenes.	El objetivo de esta sección será facilitar al usuario que busca solucionar un tipo de problema específico para cualquiera sea su propósito.	Se deberá desplegar una nueva página proveniente de la sección problemas.
7	El sistema tendrá un sistema de puntuación de cada usuario que permitirá colocarlo en una lista que se denominará Ranking.	El sistema deberá desplegar una lista con los nombres de los usuarios desplegados descendientemente de acuerdo a su puntaje.	El objetivo de esta sección será de motivar al usuario y sacar el lado competitivo de este.	Se deberá desplegar una sección de Ranking en texto plano con las siguientes características: Id usuario, nombre, puntaje.
8	El sistema tendrá una sección para crear concursos, como también será necesario poder crear una sala de	El sistema deberá tener la opción de crear concursos, una vez dentro, seleccionar problemas.	El objetivo de esta sección será para que docentes o tutores puedan crear concursos de programación para el monitoreo de sus	El método de selección de problemas será la autocompletación de palabras y selección automática.

	clases con el tutor y los alumnos.		estudiantes.	
9	El sistema tendrá la opción de personalizar el perfil del usuario	El sistema deberá proveer un espacio para que el usuario pueda editar su perfil.	El objetivo de esta sección es introducir gamificación al sistema. Para ello el usuario con ayuda de sus puntajes podrá personalizar su perfil y avanzar de niveles en el juez en línea.	Para ello será necesario el diseño de personajes y formas de mejorar el avatar del usuario.
10	El sistema contará con una sección de control de usuarios.	El sistema deberá tener una sección especial para que los usuarios que tengan privilegios de administrador en distintos niveles puedan: añadir o quitar problemas, añadir concursos, añadir aula, Cambiar contraseñas, y datos de los usuarios	El objetivo de esta sección es de suma importancia ya que varios docentes o tutores pueden aportar al sistema con sus conocimientos o nuevos retos para los nuevos y antiguos usuarios.	Se deberá crear un entorno gráfico especial para que el usuario tenga privilegios de administrador.
11	Se dispondrá de una sección de noticias.	Esta sección estará únicamente incluida en la sección de la página principal.	El objetivo de esta sección es informar alguna noticia respecto a las competencias de programación, también informar respecto algún aspecto importante referente a las aulas, nuevos cursos externos, etc.	Se creará un componente especial para cada noticia con foto incluida. Si la publicación original no incluye una fotografía entonces tendrá otro diseño.

3.1.2 Requisitos no funcionales

Luego del análisis del sistema proyectado se identificaron los siguientes requerimientos no funcionales:

- No será posible estar autenticado en más de una máquina.

- La contraseña de usuario deberá tener un mínimo de 6 caracteres.
- Los avatares en el perfil de usuario se irán desbloqueando de acuerdo a su nivel y puntaje.

Seguridad

- El sistema utilizará un protocolo SSL para realizar conexiones seguras entre clientes .
- El sistema registrará al usuario con una contraseña segura JWT.

Confiabilidad

- El sistema permitirá almacenar datos de manera correcta y completa en la base de datos, además de ello ofrece seguridad y confidencialidad.

Portabilidad

- El sistema será portable ya que deberá ser un sistema web que debe funcionar únicamente en la web y no requiere de instalación, sino de una conexión activa a internet.
- El sistema deberá ser portable en cualquier navegador.

Disponibilidad

- El sistema deberá estar disponible en un 100% para todo el público, cuando sea el momento de realizar un mantenimiento el sistema mostrará un mensaje y es en ese momento que dejará de estar disponible para el público la cantidad de tiempo que sea requerido.

Mantenibilidad

- Se deberá realizar un estudio que determine cuál será el tiempo necesario para realizar un mantenimiento necesario al sistema.
- Se deberán realizar cambios y actualizaciones necesarias cada vez que el programador vea por necesario, estas se deberán realizar a parte del tiempo de mantenibilidad.

3.2 Diagrama de casos de uso

En el diagrama de casos de uso representado por la Figura 2 se especificó la comunicación y el comportamiento del sistema de juez en línea “Odisea” mediante su interacción con los usuarios y el sistema de motor de evaluación.

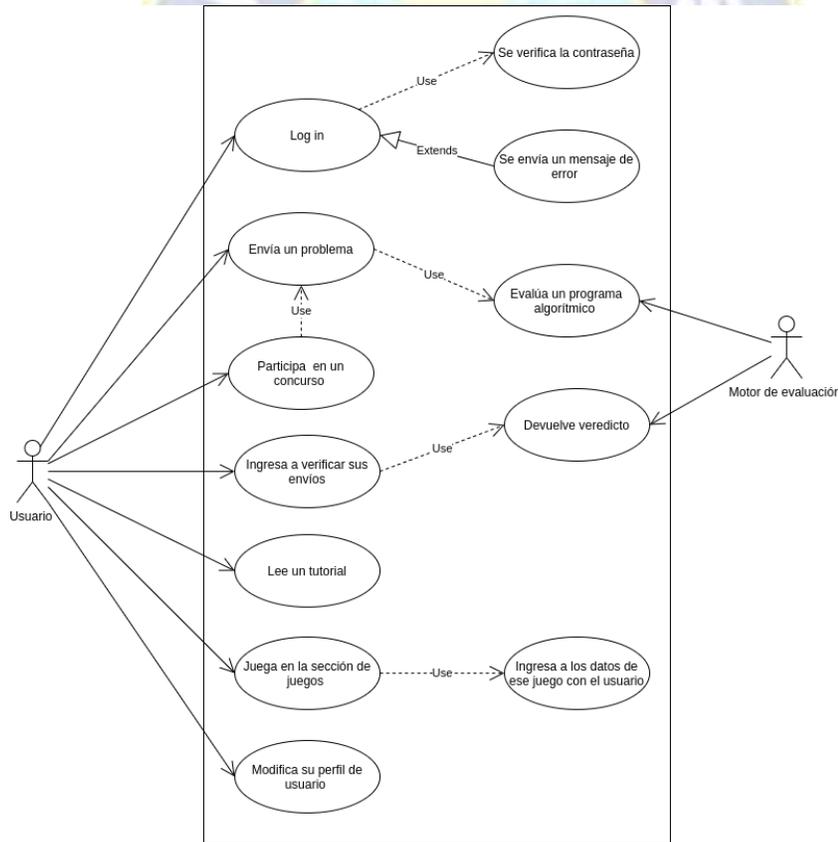


Figura 2: Diagrama de casos de uso

3.3 Diagrama de secuencia

Para el modelado de interacción entre objetos dentro del sistema de juez en línea “Odisea” se toman en cuenta los objetos que tienen relación entre sí, estos serían: el sistema de juez en línea por parte del cliente (front end), el sistema de juez en línea por parte del servidor

(back end), el motor externo de evaluación y la base de datos. Este proceso se describe en la Figura 3.

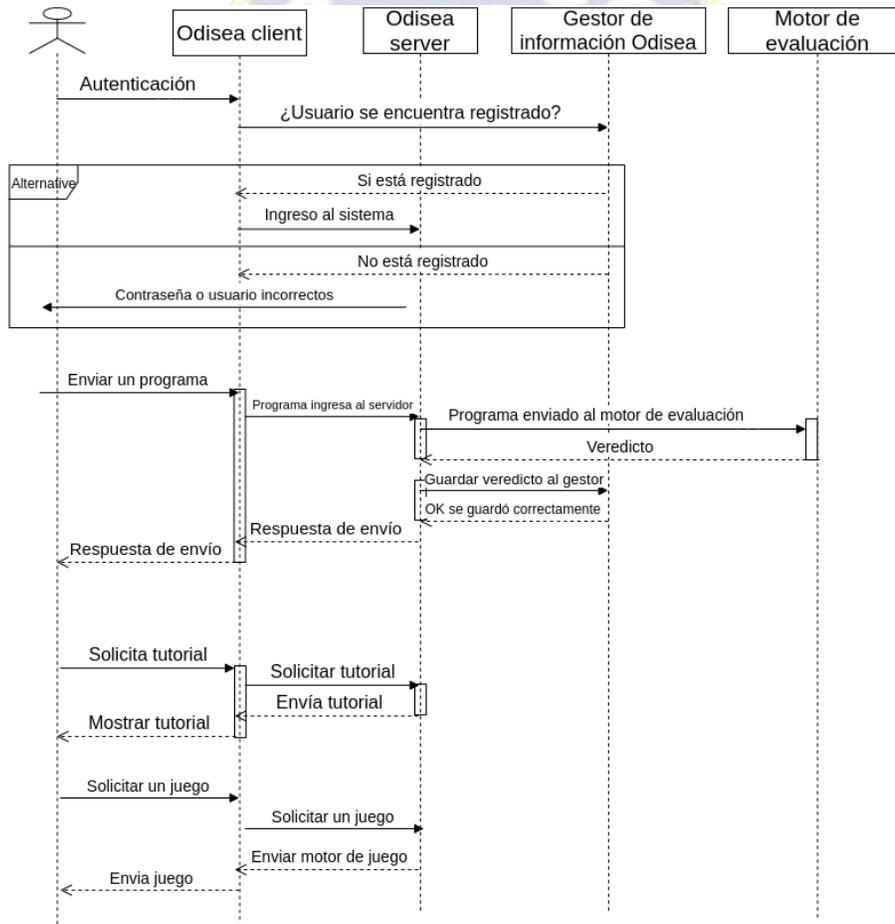


Figura 3: Diagrama de secuencia

3.4 Metodología ágil SCRUM

3.4.1 Roles de usuarios dentro de la metodología Scrum.

Para iniciar con el desarrollo de la metodología Scrum, se necesita definir los integrantes del equipo Scrum que se reflejan en la Tabla 2.

Tabla 2.

Roles de usuario

Rol	Descripción
Dueño del producto (Product Owner):	El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. Estará representado por: Univ. Fabiola Vanessa Aliaga Salvatierra y Lic. Jorge Terán
Scrum Master:	El Scrum Master es el responsable de que el equipo Scrum que trabaja ajustándose a la teoría, prácticas y reglas de Scrum. Este estará representado por : Lic. Rosa Flores
Equipo de desarrollo:	Será el equipo responsable de diseñar y desarrollar el producto. Estará representado por: Univ. Fabiola Vanessa Aliaga Salvatierra.

3.4.2 Descripción de privilegios de usuario

Los usuarios podrán ingresar al sistema con ciertos permisos de acuerdo al rol asignado, la Tabla 3 describe las opciones asignadas.

Tabla 3

Descripción de privilegios de usuario

Tipo de usuario	Descripción
Administrador:	El usuario Administrador podrá realizar las siguientes acciones dentro del sistema: Acceso a la información y edición de todo el contenido dentro del sistema, asignar privilegios de usuario, crear aulas, agregar y editar noticias, agregar problemas, agregar concursos y rejuzgar problemas.

Tutor:	El tutor tendrá los siguientes privilegios: Crear aulas, agregar problemas, agregar concursos públicos y privados, rejuzar problemas.
Usuario:	Los usuarios podrán ingresar a las funcionalidades básicas del sistema, como ser: Crear y editar su perfil de usuario, crear contenido para su blog para que esté publicado en las noticias del sistema, inscribirse a concursos públicos, acceder y enviar cualquier problema, ingresar al área de jugabilidad e ingresar a los tutoriales y poder descargarlos.

3.4.3 Pila de productos (Product Backlog)

Según el proceso Scrum, será necesario crear la pila de productos donde todas las necesidades del sistema estén desplegadas en forma de pila como se muestra en la Tabla 4.

Tabla 4

Product Backlog

ID	Nombre	Prioridad	Estado	Dependencia
R1	Desarrollo de módulo de autenticación	Alta	Terminado	-
R2	Desarrollo de módulo de página de inicio: noticias y menú de opciones.	Alta	Terminado	-
R3	Desarrollo de módulo de lista de	Alta	Terminado	R8

	problemas.			
R4	Desarrollo de módulo de problema. Estructura de cuerpo de descripción de problema.	Alta	Terminado	R3
R5	Desarrollo de módulo de envío. Estructura de cuerpo de envío.	Alta	Terminado	R4
R6	Desarrollo de módulo de conexión con el motor de evaluación .	Alta	Terminado	R5
R7	Desarrollo de módulo de despliegue de la lista de envíos.	Alta	Terminado	R6
R8	Desarrollo de módulo de despliegue de categorías.	Media	Terminado	R2
R9	Desarrollo de módulo de área de tutoriales.	Alta	Terminado	R2
R10	Desarrollo de despliegue de lista de concursos.	Alta	Terminado	R2, R3
R11	Desarrollo de módulo de creación de aulas.	Media	Terminado	R2, R3
R12	Desarrollo de módulos de despliegue de lista de Ranking.	Alta	Terminado	R1
R13	Desarrollo de módulo de jugabilidad.	Alta	Terminado	R2
R14	Desarrollo de módulo de perfil de usuario.	Media	Terminado	R1, R2

3.4.4 Primer sprint

- **Sprint Backlog**

Los elementos seleccionados desde el product backlog para el primer sprint se observa en la tabla 5:

Tabla 5

Sprint Backlog primera iteración

Sprint 1			Inicio: 01/12/2019	Duración: 23 días
ID	Tarea del sprint	Tipo	Est. (Días)	Estado
1	Planificación de Sprint	Análisis	1	Terminado
2	Creación de proyecto en node js, conexión con Mongodb.	Desarrollo (back end)	1	Terminado
3	Desarrollo de módulo de autenticación	Diseño y desarrollo (front end, back end)	1	Terminado
3	Desarrollo de módulo de página de inicio: noticias y menú de opciones.	Diseño y desarrollo (front end)	1	Terminado
4	Desarrollo de módulo de lista de problemas.	Desarrollo (front end, back end y base de datos)	1	Terminado
5	Desarrollo de módulo de problema. Estructura de cuerpo de descripción de problema.	Diseño y desarrollo (front end)	1	Terminado

6	Desarrollo de módulo de envío. Estructura de cuerpo de envío.	Diseño y desarrollo (front end)	1	Terminado
7	Desarrollo de módulo de conexión con el motor de evaluación .	Desarrollo (back end)	14	Terminado
8	Desarrollo de módulo de despliegue de la lista de envíos.	Diseño y desarrollo (back end, front end)	2	Terminado

- **Historias de usuario:** Una vez obtenido el Sprint Backlog, este fue representado por historias de usuario (ver tablas 6, 7, 8, 9, 10, 11).

Tabla 6

Historia de usuario número uno.

Historia de usuario	
Numero: 1	
Nombre de historia: Registro y autenticación de usuario	
Puntos estimados: 5	Iteración asignada: 1
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
Descripción: Se debe realizar un sector de registro de usuario y autenticación del mismo.	
Validación: Se deberá tomar en cuenta una autenticación de forma segura. Se deberá tomar en cuenta contraseña minimamente de seis caracteres.	

Tabla 7

Historia de usuario número dos.

Historia de usuario	
Numero: 2	
Nombre de historia: Página de inicio y noticias	
Puntos estimados: 2	Iteración asignada: 1
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta

Descripción: Realizar una página de inicio en el que se deberá mostrar noticias y novedades. Tanto en texto plano como en fotografía.

Validación: Para cada sección se deberá tomar en cuenta que todas las imágenes tendrán el mismo tamaño.

Tabla 8

Historia de usuario número tres.

Historia de usuario	
Numero: 3	
Nombre de historia: Lista de problemas.	
Puntos estimados: 5	Iteración asignada: 1
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
Descripción: Se debe realizar una sección de problemas enlistados en una tabla.	

Tabla 9

Historia de usuario número cuatro.

Historia de usuario	
Numero: 4	
Nombre de historia: Cuerpo del problema	
Puntos estimados: 5	Iteración asignada: 1
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
Descripción: Al momento de ingresar a la lista de problemas y seleccionar un problema se deberá generar la estructura de cada problema de la siguiente forma: descripción del problema, descripción de entrada, descripción de salida, ejemplo de entrada y de salida.	

Tabla 10

Historia de usuario número cinco.

Historia de usuario
Numero: 5

Nombre de historia: Envío del problema (submit)	
Puntos estimados: 5	Iteración asignada: 1
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
Descripción: Al momento de ingresar a la sección de envío será necesario implementar una sección de editor de texto y a la vez una sección subir archivos.	
Validación: Los lenguajes de programación que aceptará el sistema serán los siguientes:C, C++, JAVA y Python	

Tabla 11

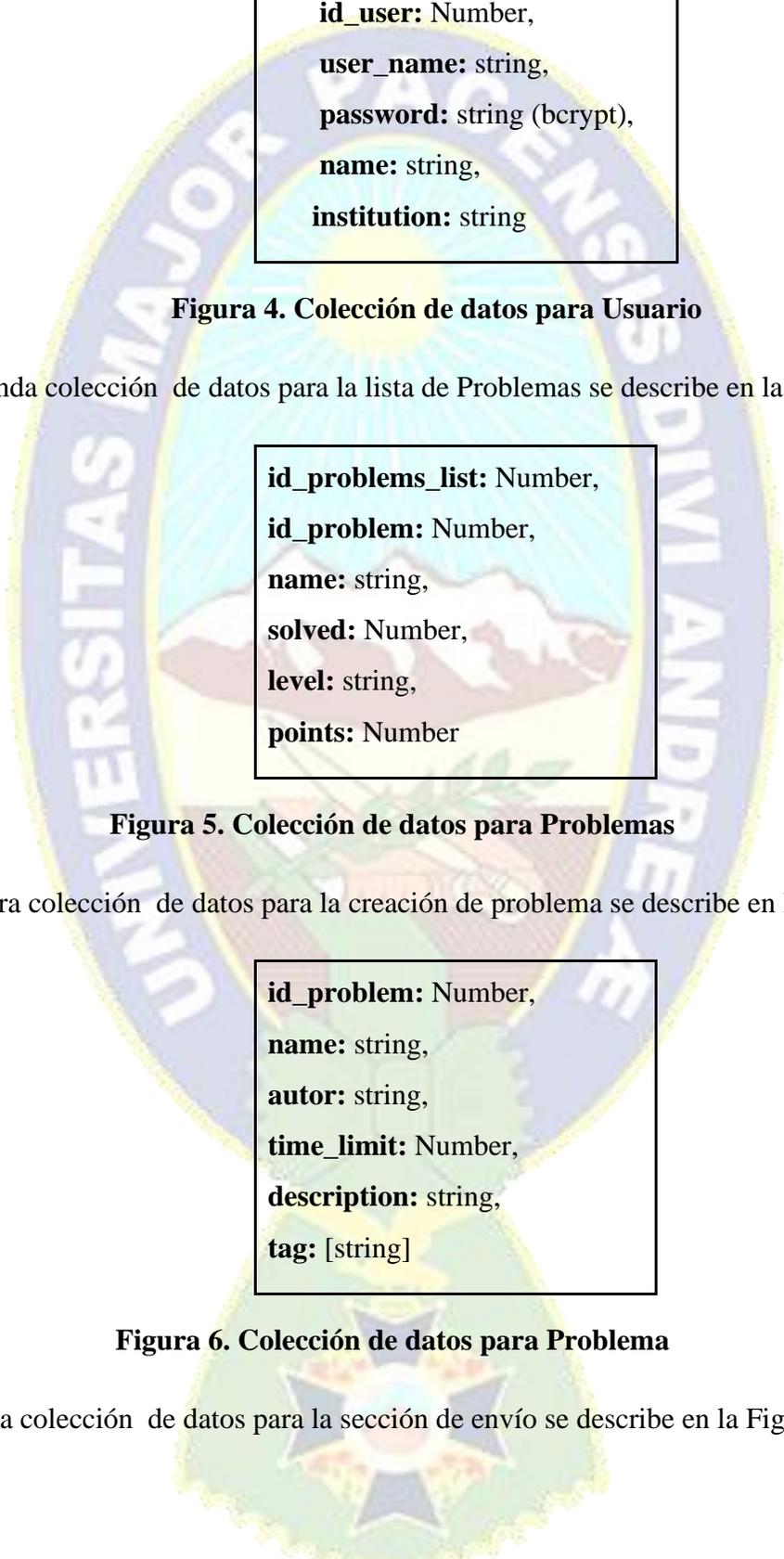
Historia de usuario número seis.

Historia de usuario	
Numero: 6	
Nombre de historia: Sección de mis envíos	
Puntos estimados: 5	Iteración asignada: 1
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
Descripción: Al momento de realizar el envío del problema se deberá ver una lista de envíos con las siguientes características: Id del problema, resultado, memoria, tiempo, lenguaje de programación, fecha y hora de envío.	

- **Modelado de base de datos primera iteración**

La base de datos del sistema de juez en línea “Odisea” será desarrollado en MongoDB, un gestor de base de datos no relacional.

En base a los módulos de la primera iteración vistos anteriormente, se mostrará las colecciones que se crearon a partir de ellas. La primera colección de datos para creación de usuario se describe en la Figura 4.



```
id_user: Number,  
user_name: string,  
password: string (bcrypt),  
name: string,  
institution: string
```

Figura 4. Colección de datos para Usuario

La segunda colección de datos para la lista de Problemas se describe en la Figura 5.

```
id_problems_list: Number,  
id_problem: Number,  
name: string,  
solved: Number,  
level: string,  
points: Number
```

Figura 5. Colección de datos para Problemas

La tercera colección de datos para la creación de problema se describe en la Figura 6.

```
id_problem: Number,  
name: string,  
autor: string,  
time_limit: Number,  
description: string,  
tag: [string]
```

Figura 6. Colección de datos para Problema

La cuarta colección de datos para la sección de envío se describe en la Figura 7.

id_submission: Number,
code: string,
veredict: string,
language: string,
memory: Number,
time: Number,
date: Timestamps

Figura 7. Colección de datos para proceso de envío de problemas

- **Scrum diario**

El equipo de desarrollo formuló diariamente las siguientes preguntas: ¿Qué se hizo ayer para lograr el objetivo del scrum?, ¿Qué haré hoy para el objetivo del scrum?, ¿Veo algún impedimento para que se evite el objetivo del sprint?

Para tener una mejor organización de sprint se utilizaron tableros Kanban. Estos tableros estarán desarrollados en la plataforma en línea “Trello” como se muestra en la Figura 8.

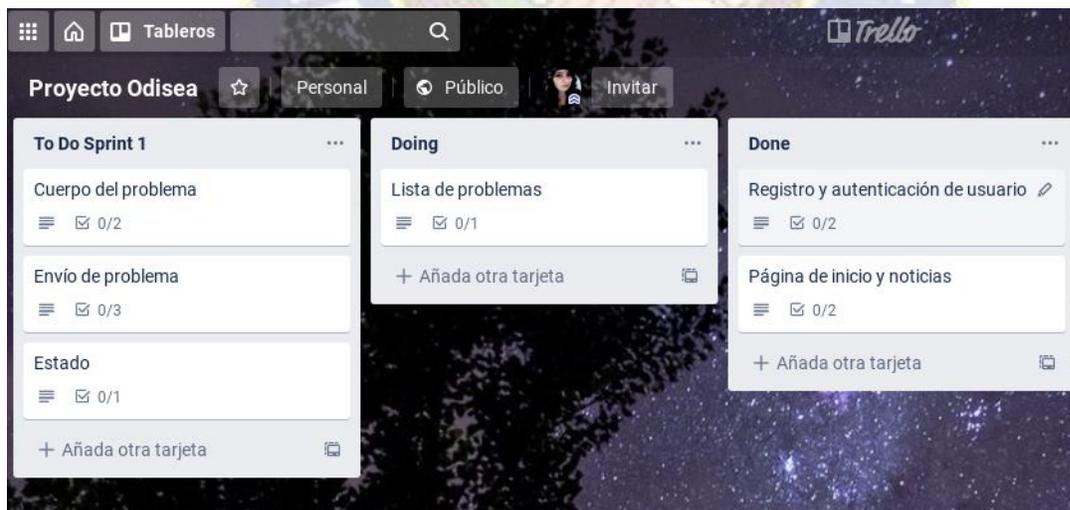
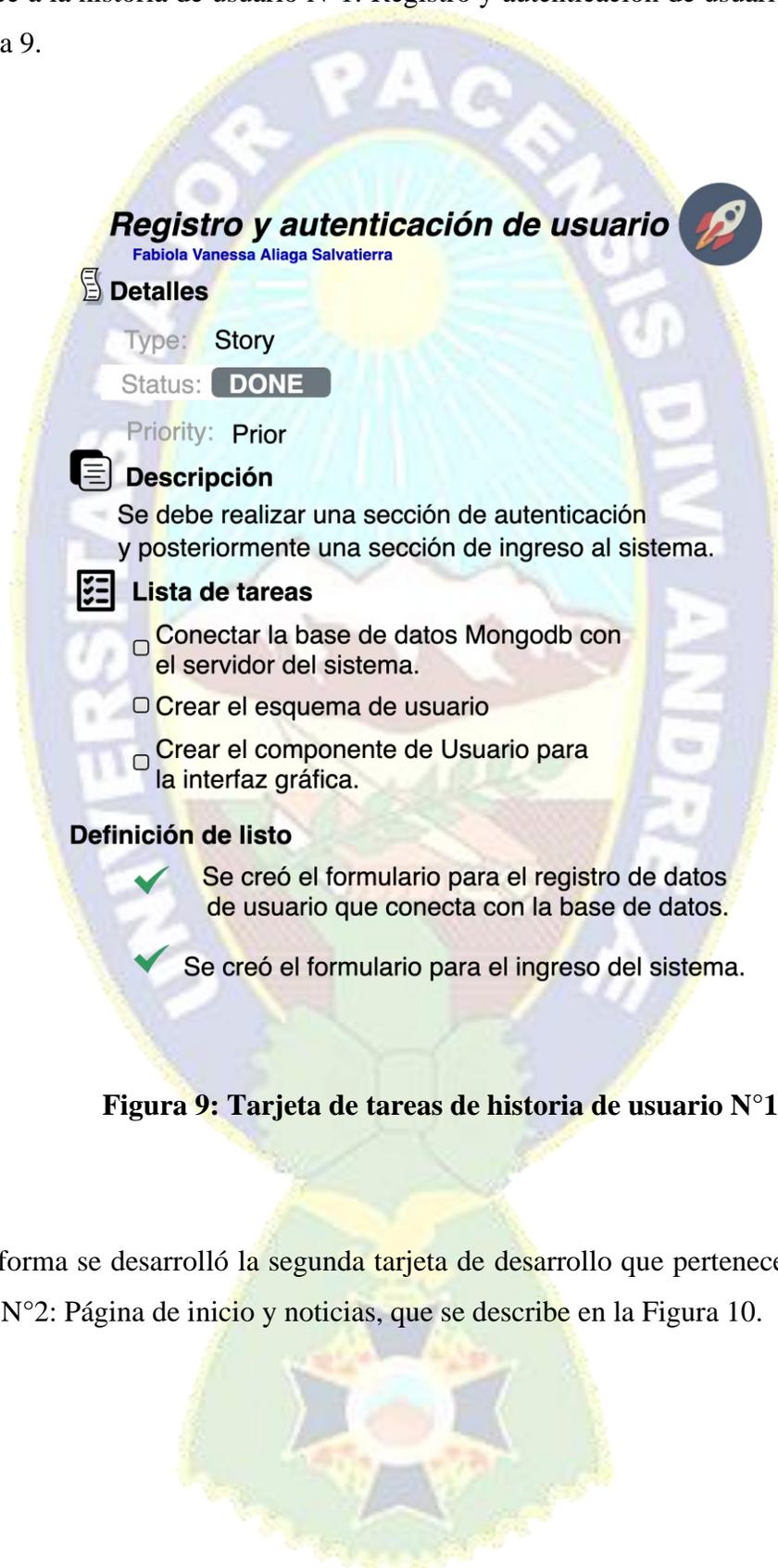


Figura 8: Tablero Kanban primera iteración de sprint

Para tener una mejor organización en la realización de cada tarea asignada, se realizaron tarjetas de tarea que ayudó al equipo de desarrollo. La primera tarjeta de desarrollo que

pertenece a la historia de usuario N°1: Registro y autenticación de usuario, se describe en la Figura 9.



Registro y autenticación de usuario 

Fabiola Vanessa Aliaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

Se debe realizar una sección de autenticación y posteriormente una sección de ingreso al sistema.

Lista de tareas

- Conectar la base de datos Mongoddb con el servidor del sistema.
- Crear el esquema de usuario
- Crear el componente de Usuario para la interfaz gráfica.

Definición de listo

- ✓ Se creó el formulario para el registro de datos de usuario que conecta con la base de datos.
- ✓ Se creó el formulario para el ingreso del sistema.

Figura 9: Tarjeta de tareas de historia de usuario N°1

De esa forma se desarrolló la segunda tarjeta de desarrollo que pertenece a la historia de usuario N°2: Página de inicio y noticias, que se describe en la Figura 10.

Página de inicio y noticias

Fabiola Vanessa Aliaga Salvatierra



Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

Se debe realizar una página de inicio en la que se mostrará: noticias, novedades del sistema, blogs de usuario.

Lista de tareas

- Realizar una sección de menú de opciones como nuevo componente.
- Crear tres secciones: noticias, concursos y perfil de usuario.
- Cada vez que un usuario realice un post ingresar a la página de inicio.

Definición de listo

- ✓ Se creó la sección de usuario en la que se observa una pequeña foto con características del usuario .
- ✓ Se creó la sección de noticias en la que se compone de una foto con una pequeña descripción. La misma modalidad se aplicará a la sección de post.
- ✓ Se creó el área de próximos concursos en los que se observa el nombre de concurso y la fecha de inicio y duración de este.

Figura 10: Tarjeta de tareas de historia de usuario N°2

Después, se desarrolló la tercera tarjeta de desarrollo que pertenece a la historia de usuario N°3: Lista de problemas, que se describe en la Figura 11.

Lista de problemas



Fabiola Vanessa Aliaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

Todos los problemas deben estar enlistados en una tabla.

Lista de tareas

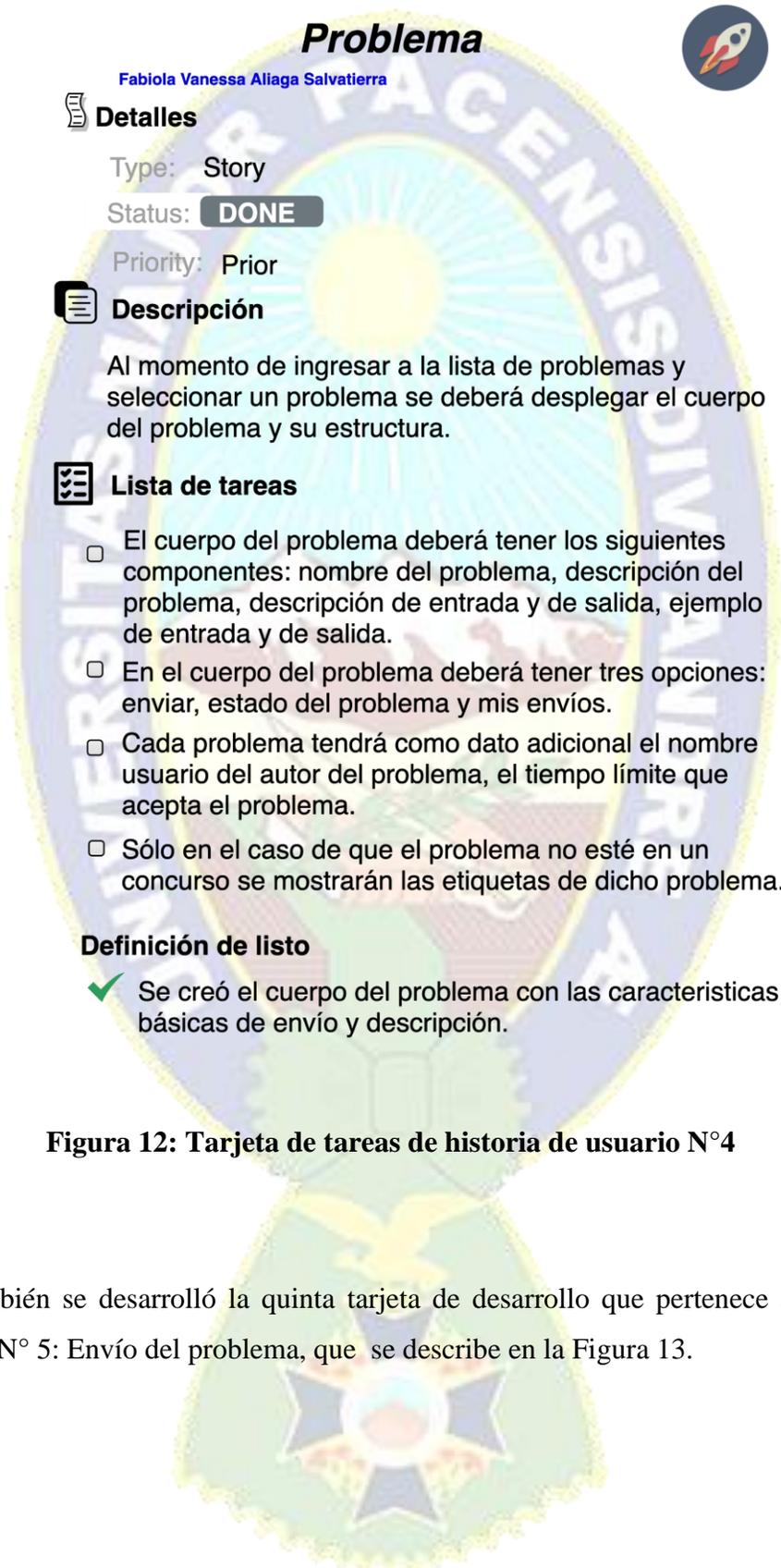
- Realizar una tabla con las características: id_problema, nombre, cantidad resueltos, nivel de dificultad, cantidad de resueltos y puntaje.

Definición de listo

- Se creó la sección de problemas que viene desde el menú de opciones. Los problemas están en forma de lista.

Figura 11 : Tarjeta de tareas de historia de usuario N°3

Así también se desarrolló la cuarta tarjeta de desarrollo que pertenece a la historia de usuario N°4: Cuerpo de problema, que se describe en la Figura 12.



Problema

Fabiola Vanessa Aliaga Salvatierra



Detalles

Type: Story
Status: **DONE**
Priority: Prior

Descripción

Al momento de ingresar a la lista de problemas y seleccionar un problema se deberá desplegar el cuerpo del problema y su estructura.

Lista de tareas

- El cuerpo del problema deberá tener los siguientes componentes: nombre del problema, descripción del problema, descripción de entrada y de salida, ejemplo de entrada y de salida.
- En el cuerpo del problema deberá tener tres opciones: enviar, estado del problema y mis envíos.
- Cada problema tendrá como dato adicional el nombre usuario del autor del problema, el tiempo límite que acepta el problema.
- Sólo en el caso de que el problema no esté en un concurso se mostrarán las etiquetas de dicho problema.

Definición de listo

- Se creó el cuerpo del problema con las características básicas de envío y descripción.

Figura 12: Tarjeta de tareas de historia de usuario N°4

Así también se desarrolló la quinta tarjeta de desarrollo que pertenece a la historia de usuario N° 5: Envío del problema, que se describe en la Figura 13.

Envío del problema



Fabiola Vanessa Allaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

Este componente será el encargado de enviar la solución del problema directamente al motor de evaluación.

Lista de tareas

- Se deberá implementar un editor de texto de código. Una vez que se guarden los cambios del texto que se haya escrito, será necesario que se guarde en un archivo para que sea compilado y evaluado por el motor de juez en línea.
- Será necesaria la opción de seleccionar archivos desde el escritorio por si el usuario no hace uso del editor de texto.
- Colocar un botón de selección múltiple de lenguajes de programación para que una vez asignado el motor pueda compilar el archivo.

Definición de listo

- Al escribir código fuente en la sección de editor de código, el sistema lo envía para su posterior evaluación.
- Al seleccionar un archivo directamente de nuestro escritorio, el sistema lo envía para su posterior evaluación.

Figura 13: Tarjeta de tareas de historia de usuario N°5

De la misma forma, se desarrolló la sexta tarjeta de desarrollo que pertenece a la historia de usuario N° 6: Mis envíos, que se describe en la Figura 14.

Mis envíos

Fabiola Vanessa Allaga Salvatierra

Detalles

Type: Story
 Status: **DONE**
 Priority: Prior

Descripción

Este módulo se desplegará en el momento que el usuario realice un envío o simplemente ingrese a la sección de "Mis envíos". Será una tabla con las características del envío.

Lista de tareas

- Se deberá desplegar una tabla con las siguientes características: id_envío, Nombre de usuario, Nombre de problema, Veredicto, Lenguaje de compilación, Memoria, Tiempo y Fecha de envío.

Definición de listo

- Se logró observar el veredicto del envío del problema junto con el historial de envíos registrados en una tabla.

Figura 14: Tarjeta de tareas de historia de usuario N°6

- **Revisión del sprint**

En la revisión del primer sprint se evaluó el desarrollo de cada tarea y se concluyó que hubieron varios desbordamientos en el tiempo estimado de cada tarea y que el diseño no fue tomado a consideración dentro de la pila de backlog.

- **Retrospectiva del sprint**

Mejoras para el siguiente sprint:

- Mejorar el tiempo de conclusión de tareas.
- Estimar las tareas de una manera más realista.

3.4.5 Segundo sprint

- **Sprint Backlog**

Los elementos seleccionados desde el product backlog para el segundo sprint se observa en la tabla 12:

Tabla 12.

Sprint Backlog segunda iteración

Sprint 2			Inicio: 01/01/2020	Duración: 20 días
ID	Tarea del sprint	Tipo	Est. (Días)	Estado
1	Planificación de Sprint	Análisis	1	Terminado
2	Desarrollo de módulo de componente categoría.	Desarrollo (back end)	1	Terminado
3	Desarrollo módulo de sección de categorías.	Diseño y desarrollo (front end, back end)	1	Terminado
4	Desarrollo del módulo componente de tutorial.	Diseño y desarrollo (front end, back end)	3	Terminado
5	Desarrollo módulo sección de tutoriales.	Desarrollo (front end, back end)	7	Terminado
6	Desarrollo módulo sección de ranking mejores participantes.	Diseño y desarrollo (front end, back end)	1	Terminado
7	Desarrollo módulo de componente de concurso e inscripción.	Diseño y desarrollo (front end, back end)	3	Terminado
8	Desarrollo de sección de concursos.	Diseño y desarrollo (front end, back end)	3	Terminado

- **Historias de usuario:** Una vez obtenido el Sprint Backlog de la segunda iteración, este fue representado por historias de usuario (ver tablas 13, 14, 15, 16).

Tabla 13

Historia de usuario número siete

Historia de usuario	
Numero: 7	
Nombre de historia: Categorías de problemas	
Puntos estimados: 5	Iteración asignada: 2
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
<p>Descripción: El sistema deberá mostrar una sección especial de categorías para que el usuario pueda clasificar los problemas propuestos para resolver con mayor facilidad.</p>	
<p>Validación: Esta sección se mostrará en forma de pequeños recuadros desplegados en la pantalla. Cada recuadro llevará una imagen y su texto.</p>	

Tabla 14

Historia de usuario número ocho

Historia de usuario	
Numero: 8	
Nombre de historia: Tutoriales	
Puntos estimados: 5	Iteración asignada: 2
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Media
<p>Descripción: El sistema deberá mostrar una sección especial de categorías para que el usuario pueda clasificar los problemas propuestos para resolver con mayor facilidad.</p>	
<p>Validación: La sección de tutoriales estará en texto plano y tendrá el siguiente formato: a) Título de tutorial, b) Cuerpo o texto principal, En el caso de ser un tema de estudio contendrá lo siguiente a parte</p>	

de los dos primeros incisos:
 c) Problema de ejemplo,
 d) Resolución de ese problema propuesto.

Tabla 15

Historia de usuario número nueve

Historia de usuario	
Numero: 9	
Nombre de historia: Lista de Concursos	
Puntos estimados: 5	Iteración asignada: 2
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
<p>Descripción: En el sistema se deberá desplegar una lista de concursos para que el usuario pueda ingresar a participar del mismo. Si el concurso ya no está "corriendo" el usuario podrá entrar a revisar los problemas y también podrá observar el ranking del mismo.</p>	
<p>Validación: Para esta sección el usuario necesitará inscribirse de forma personal o mediante invitación al concurso al que se desea participar.</p>	

Tabla 16

Historia de usuario número diez

Historia de usuario	
Numero: 10	
Nombre de historia: Sección de ranking	
Puntos estimados: 5	Iteración asignada: 2
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
<p>Descripción: En el sistema se deberá desplegar una lista de usuarios ordenados de forma descendente respecto a sus puntajes.</p>	

Validación: Para esta sección la lista será una tabla con las siguientes referencias:

- a) Id Usuario
- b) Nombre de usuario,
- c) Puntaje

- **Modelado de base de datos segunda iteración**

En base a los módulos de la segunda iteración vistos anteriormente, se mostrará las colecciones que se crearon a partir de ellas. La primera colección de categorías de problemas se describe en la Figura 15.

```
{  
  id_category: Number,  
  nombre: string,  
  description: string  
}
```

Figura 15. Colección de datos para módulo categoría

La segunda colección de datos para el módulo tutorial se describe en la Figura 16.

```
id_tutorial: Number,  
name: string,  
description: string,  
date: Timestamps
```

Figura 16. Colección de datos para módulo tutorial

La tercera colección de datos para la creación de concursos se describe en la Figura 17.

```
{
  id_contest: Number,
  name: string,
  description: string,
  type: string,
  duration: Number,
  init: Timestamps,
  id_problems: [Number],
  id_users: [Number]
}
```

Figura 17. Colección de datos de módulo concursos

La cuarta colección de datos para la sección de ranking se describe en la Figura 18.

```
{
  id_users: [Number],
  user_name: string,
  user_points: Number,
  user_solved: Number
}
```

Figura 18. Colección de datos para la sección de ranking

- **Scrum diario**

En el equipo de desarrollo se formularán estas preguntas diariamente mientras dure el sprint. ¿Qué se hizo ayer para lograr el objetivo del scrum? ¿Qué haré hoy para el objetivo del scrum? ¿Veó algún impedimento para que se evite el objetivo del sprint?

El desarrollo del segundo sprint está actualizado en el tablero Kanban, que se encuentra en la plataforma en línea Trello, de la segunda iteración como se muestra en la Figura 19.

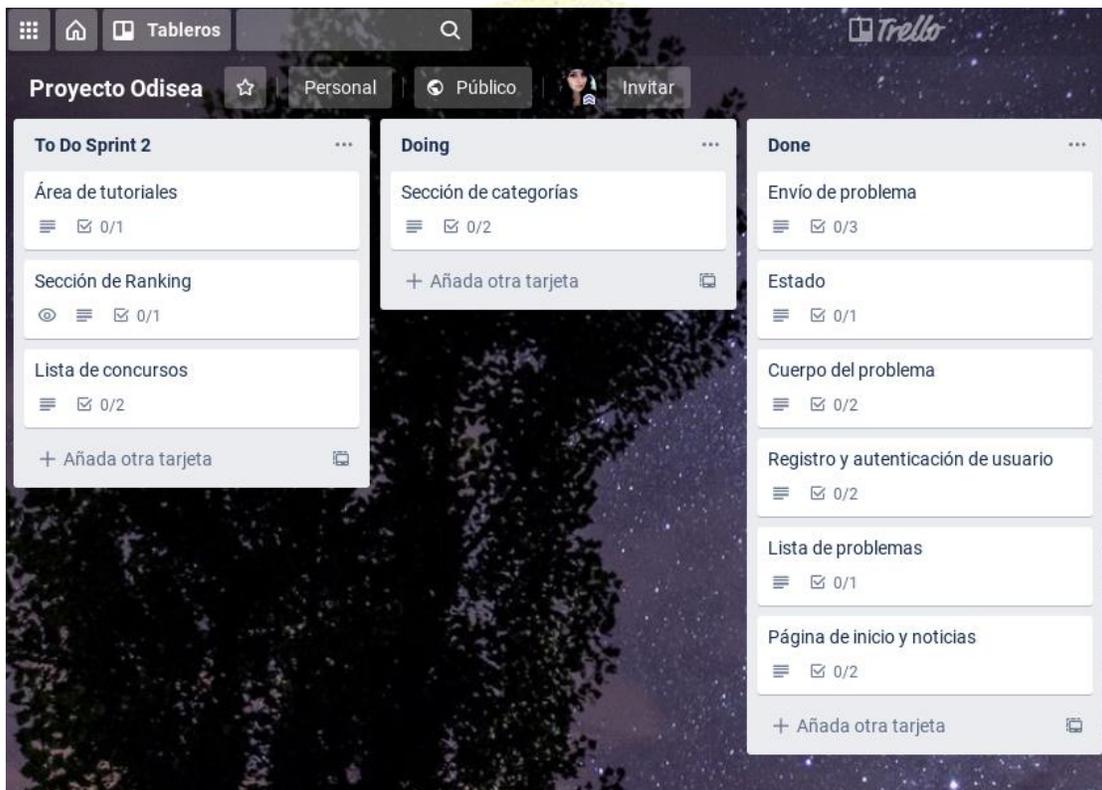


Figura 19: Tablero Kanban segunda iteración

La séptima tarjeta de desarrollo que pertenece a la historia de usuario N°7: Sección de categorías, se describe en la Figura 20.

Categorías



Fabiola Vanessa Allaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

El sistema deberá tener una sección de categorías en la cual, se clasificarán de acuerdo a sus etiquetas o tags.

Lista de tareas

- La sección de categoría estará organizada en forma de grid (Recuadros de filas y columnas).
- Cada categoría tendrá un título y una descripción.

Definición de listo

- ✓ Al ingresar al menú a la sección de problemas, se logró desplegar la sección de categorías.
- ✓ Al ingresar a cada categoría, se logró acceder a la lista de problemas clasificados por etiquetas.

Figura 20: Tarjeta de tareas de historia de usuario N°7

La octava tarjeta de desarrollo que pertenece a la historia de usuario N°8: Área de tutoriales, se describe en la Figura 21.

Tutoriales



Fabiola Vanessa Aliaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

El sistema deberá tener una sección de tutoriales que servirán como guía sobre distintos temas para el usuario.

Lista de tareas

- La sección de tutoriales estarán organizados en forma de grid (Filas y columnas)
- Realizar el cuerpo del tutorial de la siguiente forma:
 - a) Título tutorial
 - b) Descripción
 - c) Problema propuesto
 - d) Sugerencia de resolución del problema.

Definición de listo

- Se logra ingresar a los tutoriales desde la sección de tutoriales.

Figura 21: Tarjeta de tareas de historia de usuario N°8

La descripción y subtareas de la tercera historia de usuario dentro del segundo Sprint llamada: Lista de concursos, estará descrito en la Figura 22.

Concursos



Fabiola Vanessa Allaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

En el sistema se deberá desplegar una lista de concursos.
El usuario podrá inscribirse a un concurso cuando este sea público.

Lista de tareas

- La lista de concursos estará desplegada en una tabla.
- Se creará dos botones: Ingresar e inscribirse
- Una vez que el concurso no esté activo, el usuario igual podrá ingresar a ver la lista de problemas, pero ya no aparecerá en el ranking del concurso.

Definición de listo

- Se logra ingresar al concurso como participante y como observador.

Figura 22: Tarjeta de tareas de historia de usuario N°9

La descripción y subtareas de la tercera historia de usuario dentro del segundo Sprint llamada: Sección de ranking, estará descrito en la Figura 23.

Ranking

Fabiola Vanessa Allaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

En el sistema se deberá desplegar una lista de usuarios ordenados de forma descendente respecto a sus puntajes.

Lista de tareas

- La lista de usuarios estará desplegada en una tabla con las siguientes opciones: id_registro, nombre de usuario, puntaje de usuario y cantidad de ejercicios resueltos.

Definición de listo

- Se logró desplegar una tabla con los mejores usuarios según su puntuación.

Figura 23: Tarjeta de tareas de historia de usuario N°10

En este sprint se ve mucha mejoría en la estimación del tiempo comparada con el sprint anterior, la tarea sobre la realización de perfil de usuario no fue concluida por lo que ingresará en el siguiente sprint.

3.4.6 Tercer Sprint

- **Sprint Backlog**

Los elementos seleccionados desde el product backlog para el tercer sprint se observa en la tabla 17:

Tabla 17.

Sprint Backlog tercera iteración

Sprint 3			Inicio: 01/02/2019	Duración: 24 días
ID	Tarea del sprint	Tipo	Est. (Días)	Estado
1	Planificación de Sprint	Análisis	1	Terminado
2	Desarrollo de interfaz gráfica para todos los componentes	Diseño y desarrollo (front end)	7	Terminado
3	Desarrollo módulo de perfil de usuario.	Diseño y desarrollo (front end, back end)	5	Terminado
4	Desarrollo módulo de jugabilidad.	Diseño y desarrollo (front end, back end)	10	Terminado
5	Desarrollo elementos de interfaz para usuario con privilegios.	Diseño y desarrollo (front end, back end)	1	Terminado

- **Historias de usuario:** Una vez obtenido el Sprint Backlog, este fue representado por historias de usuario (ver tablas 18 y 19).

Tabla 18

Historia de usuario número once

Historia de usuario	
Numero: 11	
Nombre de historia: Perfil de usuario	
Puntos estimados: 5	Iteración asignada: 3
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
Descripción: En el sistema se tendrá un área especial de perfil de usuario en el cual se podrá personalizar de acuerdo al gusto del mismo y obtener objetos especiales y escenarios distintos de acuerdo al nivel.	

Validación: Para este sector será necesario implementar una opción para invertir unos puntos en diamantes para que el usuario pueda comprar a su avatar implementos especiales.

Tabla 19

Historia de usuario número doce

Historia de usuario	
Numero: 12	
Nombre de historia: Sección de jugabilidad (niños)	
Puntos estimados: 5	Iteración asignada: 3
Responsable: Fabiola Vanessa Aliaga Salvatierra	Prioridad: Alta
Descripción: En esta sección se deberá desarrollar un minijuego interactivo para que el niño pueda aprender conceptos básicos de programación mientras juega.	
Validación: En esta sección se implementará niveles distintos en el juego.	

- **Modelado de base de datos primera iteración**

En base a los módulos de la tercera iteración vistos anteriormente, se mostrará las colecciones que se crearon a partir de ellas. La primera colección de datos para definir el perfil de usuario se describe en la Figura 24. Cabe recalcar que esta colección ya se definió en la colección de datos de la primera iteración, MongoDB nos permite escalar las colecciones de datos para una mayor eficacia.

```
{
  id_user: Number,
  user_name: string,
  password: string (bcrypt),
  name: string,
  institution: string,
  picture: string(url),
  solved: Number,
```

```
points: Number,  
post: string  
}
```

Figura 24. Colección de datos para Usuario

La segunda colección de datos para la sección de jugabilidad se describe en la Figura 25.

```
id_game: Number,  
name: string,  
level: Number
```

Figura 25. Colección de datos para la sección de Jugabilidad

- **Scrum diario**

El equipo de desarrollo formulará estas preguntas diariamente mientras dure el sprint y mientras se realizan cambios en el tablero tipo kanban.

¿Qué se hizo ayer para lograr el objetivo del scrum?, ¿Qué haré hoy para objetivo del scrum?, ¿Veo algún impedimento para que se evite el objetivo del sprint?

Las actividades de la tercera iteración se muestran en el tablero Kanban actualizado en la Figura 26. Las actividades pasadas están en la sección de Realizado (Done).

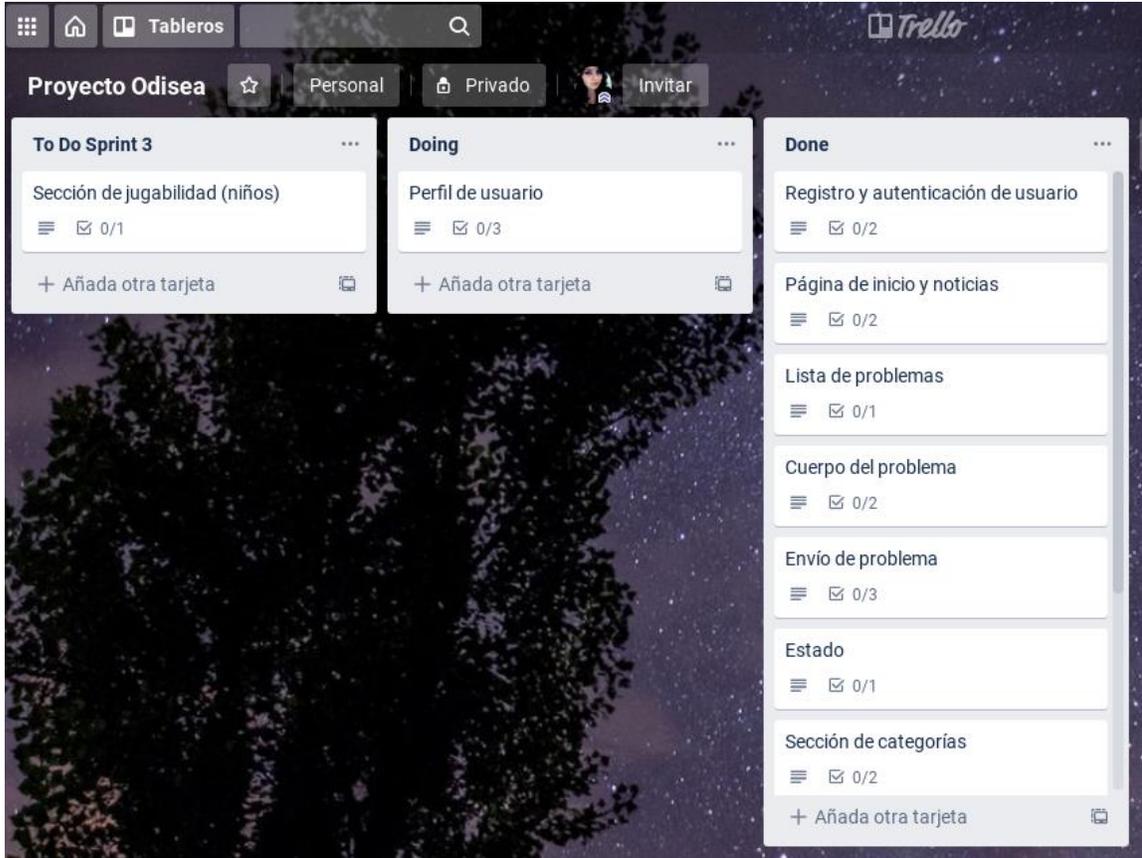
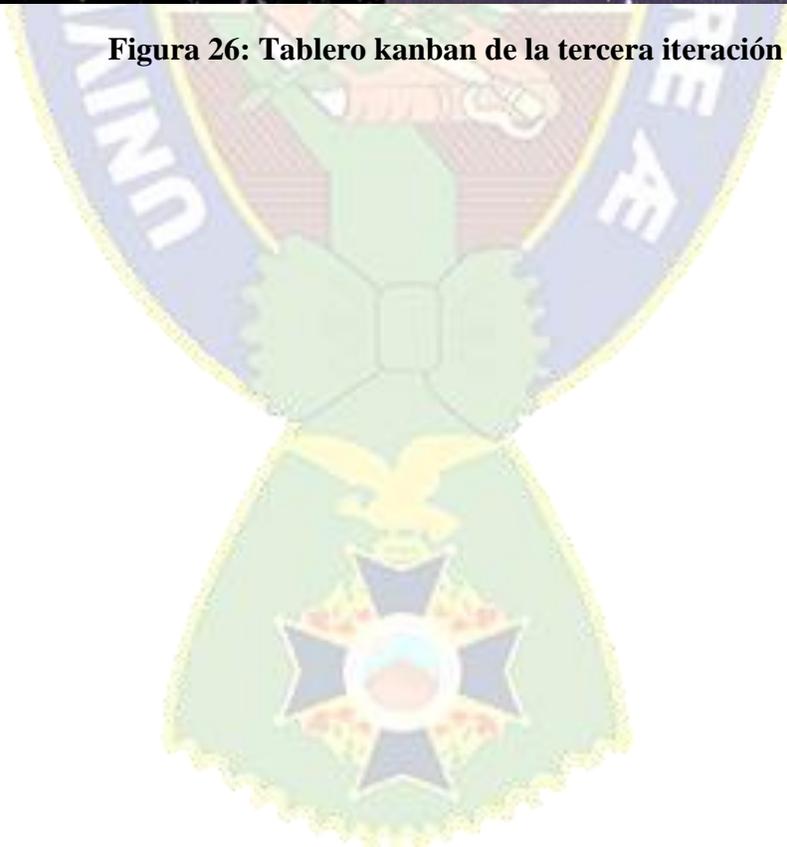
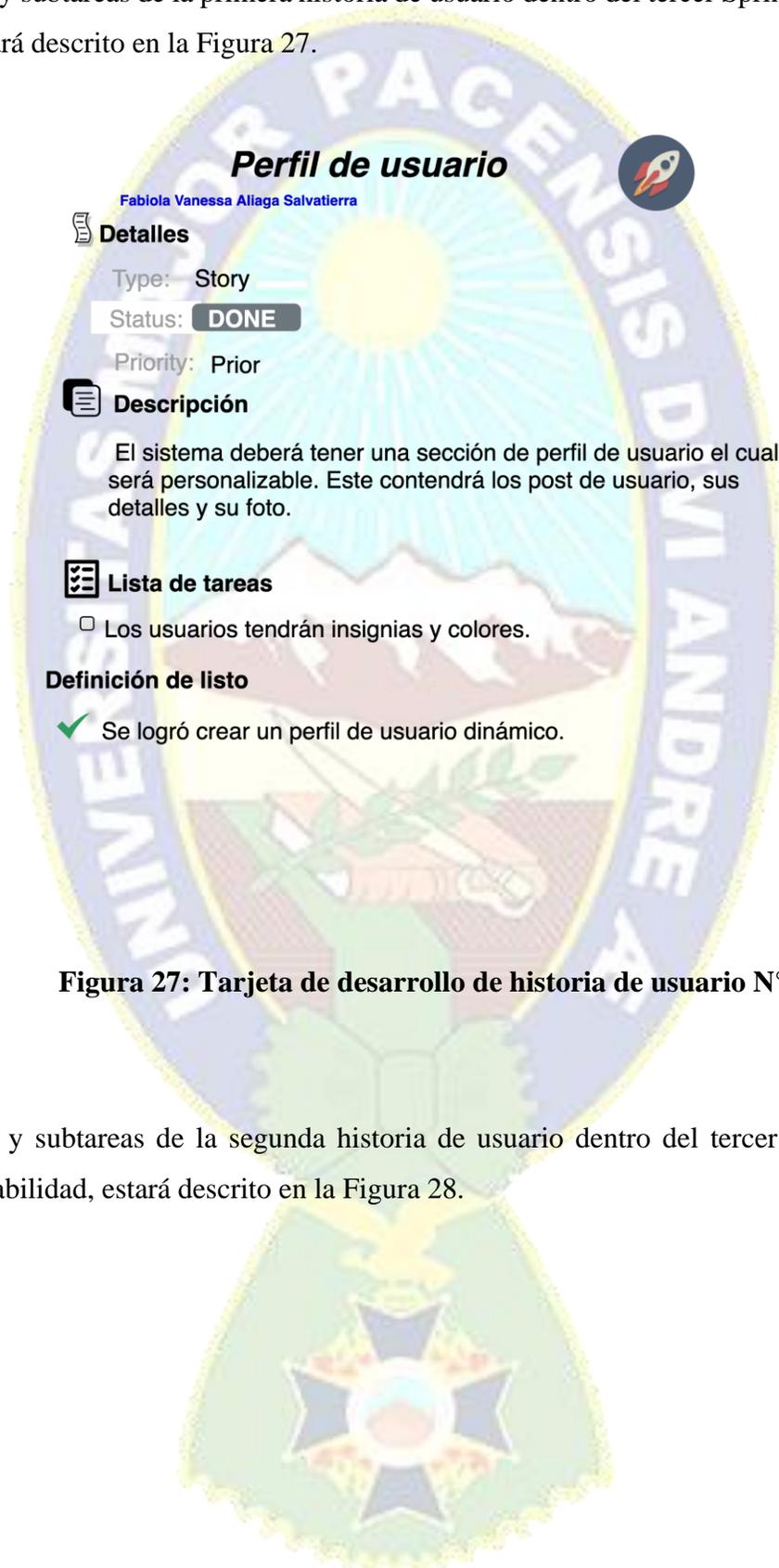


Figura 26: Tablero kanban de la tercera iteración



La descripción y subtareas de la primera historia de usuario dentro del tercer Sprint llamada: Perfil de usuario, estará descrito en la Figura 27.



Perfil de usuario 

Fabiola Vanessa Allaga Salvatierra

Detalles

Type: Story
Status: **DONE**
Priority: Prior

Descripción

El sistema deberá tener una sección de perfil de usuario el cual será personalizable. Este contendrá los post de usuario, sus detalles y su foto.

Lista de tareas

- Los usuarios tendrán insignias y colores.

Definición de listo

- Se logró crear un perfil de usuario dinámico.

Figura 27: Tarjeta de desarrollo de historia de usuario N° 11

La descripción y subtareas de la segunda historia de usuario dentro del tercer Sprint llamada: Sección de jugabilidad, estará descrito en la Figura 28.

Sección de jugabilidad



Fabiola Vanessa Allaga Salvatierra

Detalles

Type: Story

Status: **DONE**

Priority: Prior

Descripción

En esta sección se deberá desarrollar un minijuego interactivo para que el niño pueda aprender conceptos básicos de programación mientras juega.

Lista de tareas

- Se deberá implementar una sección de juego en 2D.

Definición de listo

- Al ingresar al juego, se logra interactuar con el motor del mismo.

Figura 28: Tarjeta de desarrollo tarjeta de usuario N°12

- **Revisión del sprint**

Para la realización de este sprint se tomaron en cuenta nuevas tarjetas que son consideradas en esta iteración, la realización del mini juego fue un gran reto para el equipo de desarrollo.

- **Retrospectiva del sprint**

Finalmente se logró el desarrollo de la pila de productos. A futuro, el equipo de desarrollo realizará el mantenimiento correspondiente al sistema y mejoras e implementaciones

Capítulo 4

Pruebas y Resultados

Luego de haber concluido el desarrollo del sistema de juez en línea “Odisea”, será necesario realizarle pruebas de calidad que determinen el correcto funcionamiento de este sistema, de la misma manera medir la satisfacción de los usuarios que utilizarán este sistema con distintos fines.

Para iniciar este proceso, se realizará un estudio de caso donde se evaluará la metodología que se utilizará para la recolección de datos, asimismo se detectará a la población evaluadora y se determinará el lugar donde se realizará la evaluación.

También se presentarán las encuestas que se aplicaron a la población evaluadora, y finalmente, se registrarán los resultados obtenidos del estudio de caso.

4.1. Estudio de caso

La metodología de recolección de datos que se utilizará serán las encuestas. La población a evaluar se divide en dos grupos; en el primer grupo se realizarán encuestas a la población infantil y adolescente que no tiene mucha experiencia en el campo de las competencias de programación, posteriormente a la población infantil adolescente que tienen experiencia en el campo de las competencias de programación, por ejemplo: niños y adolescentes que aprenden programación en sus colegios o en institutos privados. El segundo grupo de evaluación será la población experimentada en el campo de las competencias de programación, por ejemplo: tutores, docentes, ex-competicionistas, competidores de ICPC.

Para ambos grupos de evaluación, las redes sociales servirán de apoyo para tener un contacto directo. Se realizarán búsquedas en la sección de grupos o páginas dentro de las redes sociales que estén relacionadas con los concursos de programación y se invitará a que participen en las encuestas, para ello se les dará acceso al juez en línea “Odisea” para que posteriormente realicen

la encuesta; asimismo, se tendrá contacto con tutores y competidores registrados en el concurso de programación OBI 2019 y OCE 2019 mediante correo electrónico.

El cuestionario estará elaborado con la herramienta de Formularios de Google y las preguntas están basadas en la metodología de calidad de software con la norma ISO/IEC 9126 (del Sastre, 2010).

- **Funcionalidad:** La capacidad del producto software para proporcionar los resultados o efectos correctos o acordados con el grado necesario de precisión.
- **Confiabilidad:** La capacidad del producto software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificadas.
- **Usabilidad:** La norma iso 9241-11 define la usabilidad como la capacidad que tiene un producto para ser usado por determinados usuarios con el fin de alcanzar unos objetivos concretos con efectividad, eficiencia, y satisfacción dentro de un contexto de uso específico (Sánchez, 2011).

Según Jakob Nielsen, la usabilidad pueden medirse con cinco parámetros los cuales son: la facilidad de aprendizaje (tiempo necesario para cumplir ciertas tareas en el sistema), la eficiencia de uso (tiempo necesarios que lleva a usuarios expertos para cumplir tareas típicas del sistema), recuerdo en el tiempo (tiempo necesario por usuarios expertos que no han utilizado el sistema y ejecutan tareas de forma repetitiva) , tasa de errores (cantidad de errores cometidos por usuarios al realizar tareas típicas), y finalmente, satisfacción del usuario.

- **Eficiencia:** La capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados bajo condiciones determinadas
- **Mantenibilidad:** La capacidad del producto software para ser modificado. Las modificaciones podrían incluir correcciones, mejoras o adaptación del software a cambios en el entorno, requisitos y especificaciones funcionales

4.2. Inventario de los resultados

Antes de medir los resultados de las encuestas sobre la calidad del sistema de juez en línea “Odisea” según la norma 9126, se realizará una encuesta a la población adolescente-infantil antes de presentar el sistema para así tener un estudio más profundo sobre las necesidades de la población.

4.2.1. Encuestas preparadas para la población niños antes de realizar el software.

Número de personas que realizaron la encuesta: Once personas.

- ¿Cuál es tu edad?

Según la encuesta, en el mundo de programación competitiva se observa mayor población para los adolescentes que tienen edades entre quince y dieciocho años como se observa en la Figura 29.

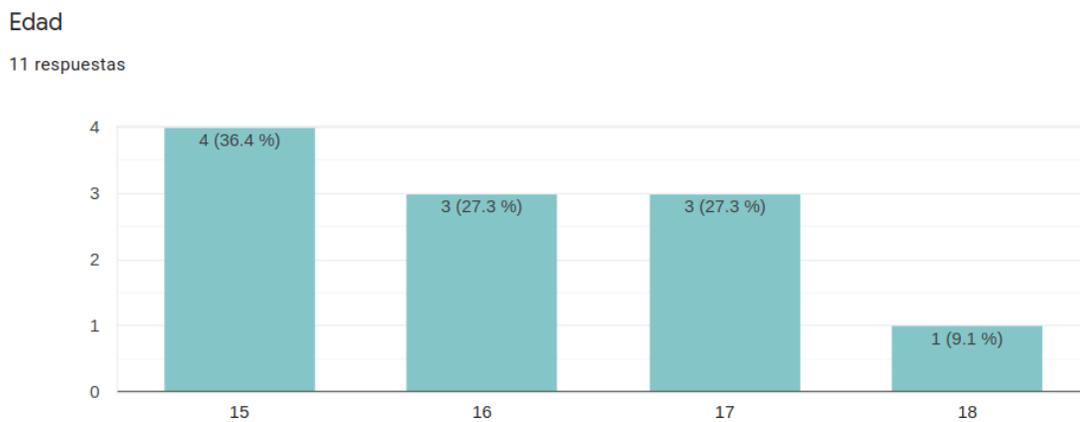


Figura 29. Histograma de la edad de los encuestados

- ¿Cual es el juez virtual que utilizas frecuentemente?

Según los encuestados, la plataforma de juez en línea más común para practicar programación competitiva es Codeforces, como nos indica la Figura 30.



Figura 30. Gráfico pastel de proporciones de jueces virtuales.

- ¿Qué opciones te gustaría que estén incluidas en el juez virtual “Odisea”?

Algunas de las respuestas literales se muestran en la Figura 31.

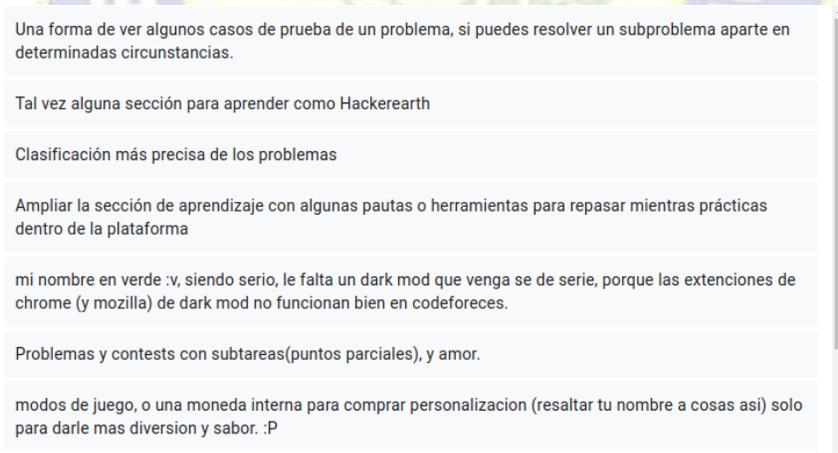


Figura 31. Respuestas de los encuestados

- ¿Te gustaría que en algún juez virtual estén incluidos algunos juegos interactivos?

A la mayoría de los encuestados les parece una gran idea la implementación de juegos interactivos, si es que este es interesante. Como se muestra en la Figura 32

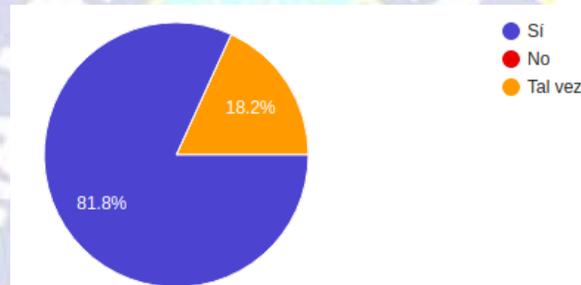


Figura 32. Gráfica de pastel de proporciones de respuestas de encuesta realizada.

- ¿En qué lenguaje te gusta programar?

Según nuestros encuestados el lenguaje de programación C++ es el más común y frecuente para practicar programación competitiva como se muestra en la Figura 33.

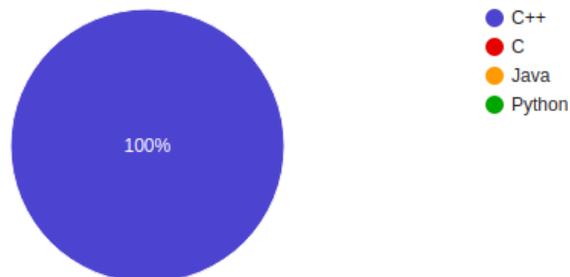


Figura 33. Gráfica de pastel de proporciones sobre lenguaje de programación.

- ¿Desde que edad aprendiste a programar?

La mayoría de los encuestados insinúan haber aprendido programación a los catorce años en institutos privados o en sus colegios; sin embargo, dos encuestados mencionaron que ex competidores les habían enseñado programación de forma particular. Solo un encuestado insinuó que aprendió de manera autodidacta. Toda esta información se aprecia en la Figura 34.

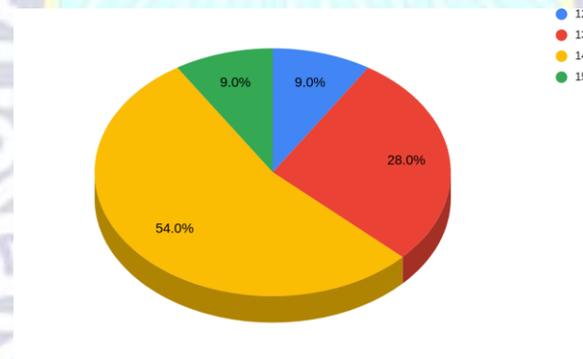


Figura 34. Gráfica de pastel de proporciones sobre edades de los encuestados.

Estas preguntas fueron realizadas en base a las normas de calidad de las normas ISO/IEC 9126.

4.2.2. Encuestas preparadas a la población niños.

Estas encuestas fueron preparadas para el momento que se haya presentado el juez en línea “Odisea”.

Número de personas que realizaron la encuesta: Diez personas.

Funcionalidad

- ¿Cómo calificarías al juez virtual “Odisea” al momento de enviar problemas, tuviste algún inconveniente?

Según las respuestas, un 80% de los encuestados afirman que no tuvieron ningún inconveniente al momento de enviar problemas, sin embargo el restante 20% tuvo inconvenientes para entender el método de envío de problemas. Así se refleja en el gráfico de barras de la Figura 35.

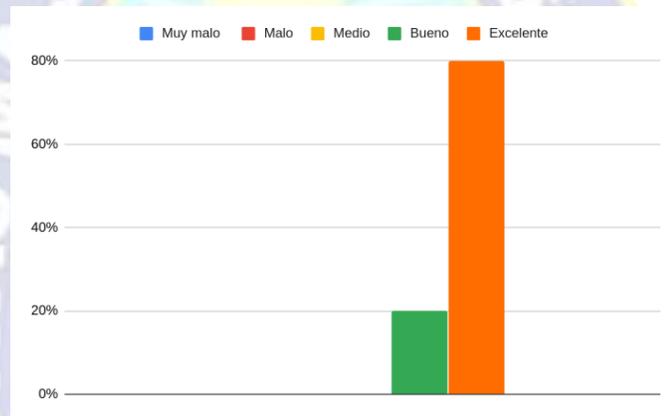


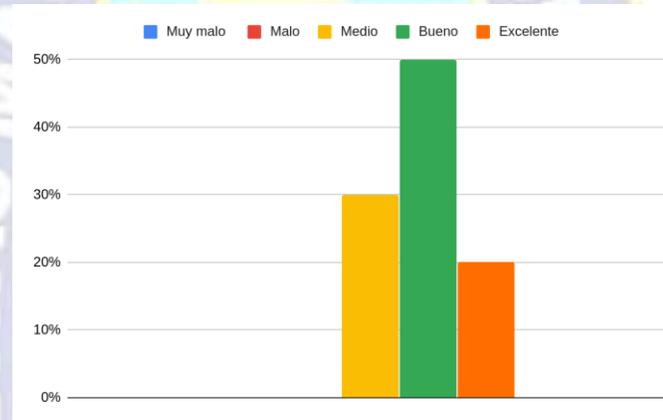
Figura 35. Gráfica de barras de resultados de la encuesta.

- ¿Cómo calificarías el minijuego implementado en juez virtual “Odisea”, crees que funciona bien?

Según las respuestas un 20% de los encuestados afirman que no tuvieron ningún inconveniente con el minijuego implementado; sin embargo, a un 50% les gustó la

idea de ese videojuego pero no les pareció perfecto, un 30% indicó que se puede mejorar la idea del juego. Así se refleja en el gráfico de barras de la Figura 36.

Figura 36. Gráfica de barras de resultados de la encuesta



Confiabilidad

- ¿Cómo calificarías el siguiente proceso? Cuando envío algún problema y olvidé definir cuál es mi lenguaje de programación o simplemente olvidé subir el archivo de código fuente, aparece en mi pantalla algún mensaje de alerta que indique alguno de esos aspectos.

Según las respuestas un 80% de los encuestados afirman que los mensajes de alerta aparecen en el momento adecuado; sin embargo, a un 20% le parece que los mensajes pueden ser más explícitos. Así se refleja en el gráfico de barras de la Figura 37.

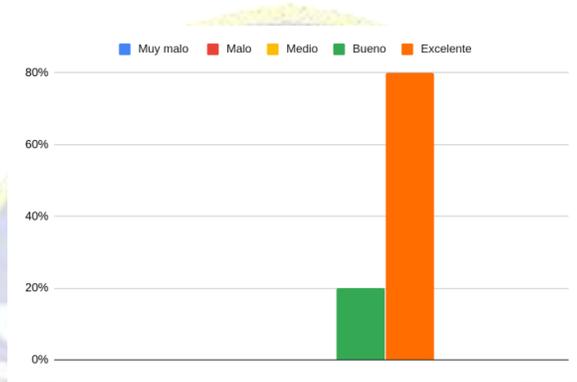


Figura 37. Gráfica de barras de resultados de las encuestas

Usabilidad

- ¿Cómo calificarías la navegación por el sistema del juez virtual “Odisea”, te pareció fácil de manejar?

Según las respuestas un 90% de los encuestados afirman que el sistema es fácil de manejar; sin embargo, a un 10% le parece que la navegación puede ser aún mejor. Así se refleja en el gráfico de barras de la Figura 38.

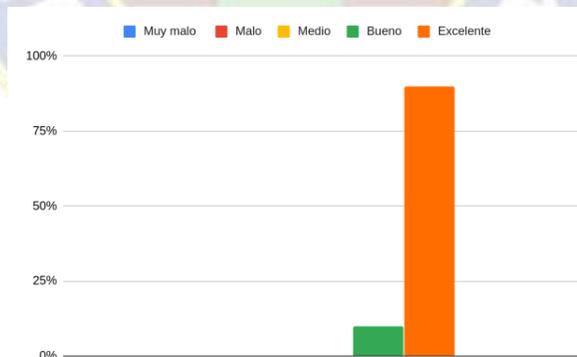


Figura 38. Gráfica de barras resultados de las encuestas

- ¿Cómo calificarías el acceso a las opciones y menú que tiene el sistema de juez en línea “Odisea”, en algún momento te costó encontrar alguna opción que buscabas?

Según las respuestas un 90% de los encuestados afirman que no tuvo problemas para encontrar alguna opción del menú dentro del sistema; sin embargo, a un 10% le parece que la navegación por el menú del sistema puede ser aún mejor. Así se refleja en el gráfico de barras de la Figura 39.

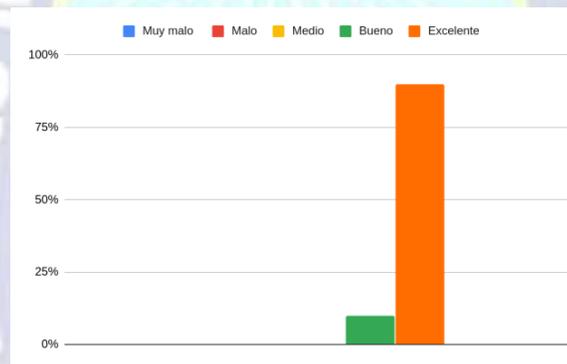


Figura 39. Gráfico de barras de los resultados de las encuestas

Eficiencia

- ¿Cómo calificarías el tiempo de espera al momento de enviar problemas al juez virtual “Odisea”, la respuesta de evaluación fue inmediata?

Según las respuestas un 60% de los encuestados afirman que el tiempo de espera de envío estuvo excelente; sin embargo, a un 40% le parece que podría ser un poco más rápido. Así se refleja en el gráfico de barras de la Figura 40.

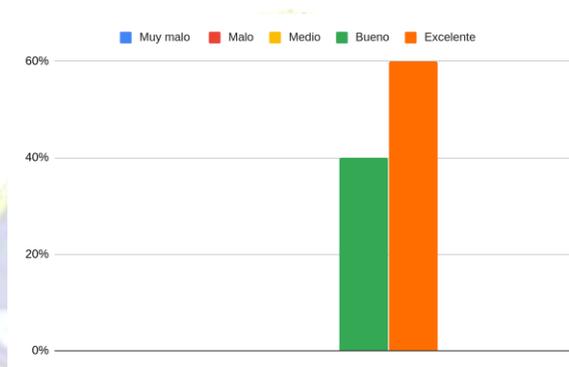


Figura 40. Gráfica de barras de resultado de encuestas.

- En general, ¿Cómo calificarías el comportamiento del juez en línea “Odisea”, te pareció lento en algún momento?

Según las respuestas un 90% de los encuestados afirman que el juez no tiene comportamientos retrasados en ningún momento; sin embargo, a un 10% le parece que el sistema podría ser un poco más rápido. Así se refleja en el gráfico de barras de la Figura 41.

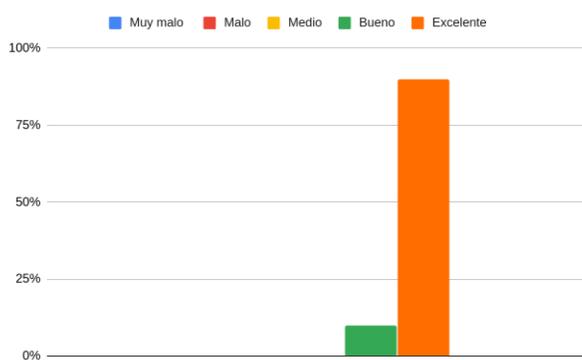


Figura 41. Gráfico de barras de resultados de encuesta.

4.2.3. Encuestas preparadas para usuarios expertos

Usabilidad

- ¿Considera que se puede ingresar el juez virtual “Odisea” de forma intuitiva a la sección de problemas y envíos?

Según las respuestas un 100% de los encuestados afirman que el juez ingresa a la sección de problemas y envíos de manera intuitiva. Así se refleja en el gráfico de barras de la Figura 42.

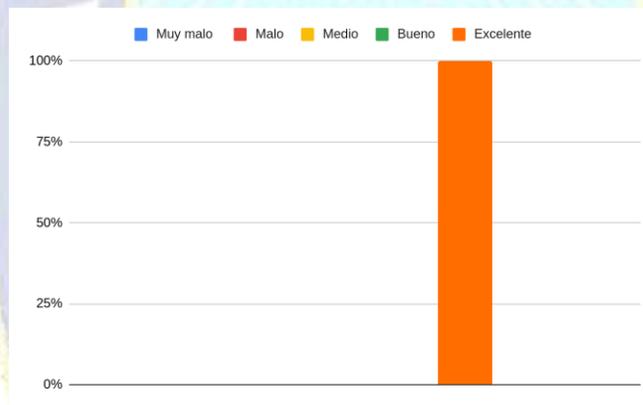


Figura 42. Gráfico de barras de respuestas de encuesta

- ¿Considera que se puede ingresar el juez virtual “Odisea” de forma intuitiva hacia la sección de jugabilidad?

Según las respuestas un 100% de los encuestados afirman que el juez ingresa a la sección de jugabilidad de manera intuitiva. Así se refleja en el gráfico de barras de la Figura 43.

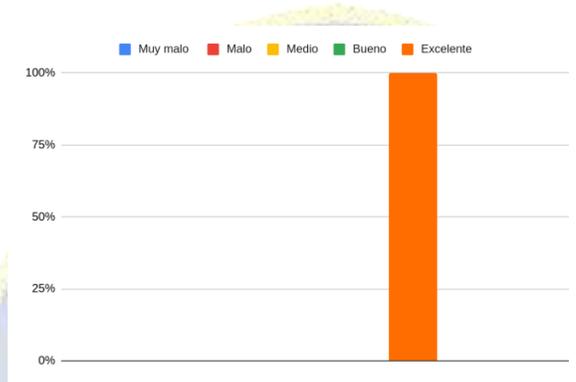


Figura 43. Gráfico de barras de respuestas de encuesta

Confiabilidad

- ¿Cómo calificaría el siguiente proceso? Cuando envío algún problema que olvidé definir el lenguaje de programación o simplemente olvidé subir el archivo de código fuente, aparece en mi pantalla algún mensaje de alerta que indique alguno de esos aspectos.

Según las respuestas un 100% de los encuestados afirman que el juez en línea despliega mensajes de alerta adecuados en el momento en el que el usuario realiza un acto erróneo. Así se refleja en el gráfico de barras de la Figura 44.

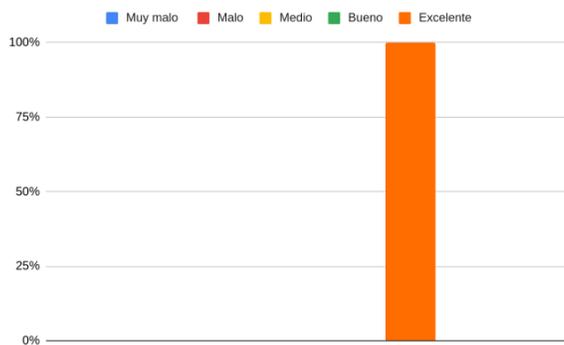


Figura 44. Gráfico de barras de respuestas de encuesta

Eficiencia

- ¿Cómo calificaría el tiempo de espera al momento de enviar problemas al juez virtual “Odisea”, la respuesta de evaluación fue inmediata?

Según las respuestas un 70% de los encuestados afirman que el tiempo de espera del envío de algún problema en el juez en línea es el adecuado; sin embargo el 30% afirma que el tiempo de espera de envío puede mejorar. Así se refleja en el gráfico de barras de la Figura 45.

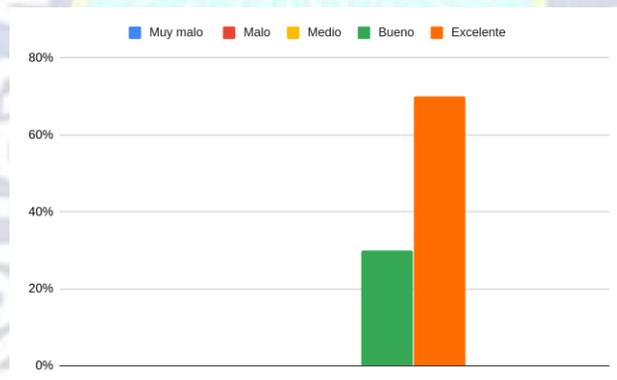


Figura 45. Gráfico de barras de resultados de encuesta

- En general, ¿Cómo calificaría el comportamiento del juez en línea “Odisea”, le pareció lento en algún momento?

Según las respuestas un 80% de los encuestados afirman que el proceso temporal del sistema es el adecuado; sin embargo, el 20% afirma que el tiempo de espera para algunos componentes del sistema puede mejorar. Así se refleja en el gráfico de barras de la Figura 46.

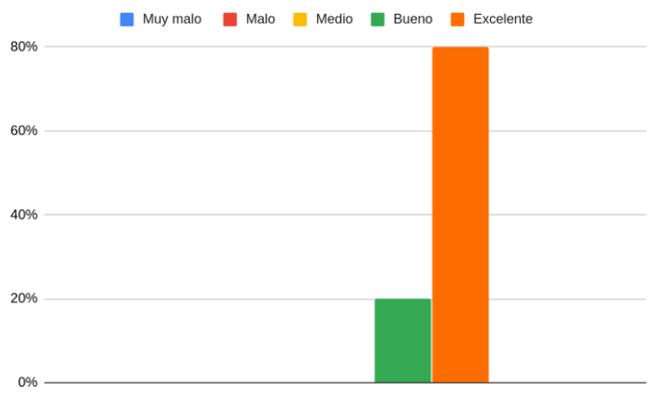


Figura 46. Gráfico de barras de resultados de encuesta

Funcionalidad

- ¿Cómo calificarías al juez virtual “Odisea” al momento de enviar problemas, tuviste algún inconveniente?

Según las respuestas un 100% de los encuestados afirman que el juez no tuvo ningún inconveniente al momento de enviar problemas. Así se refleja en el gráfico de barras de la Figura 47.

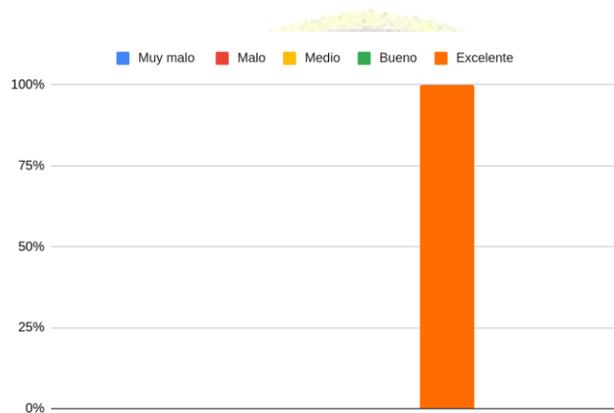


Figura 47. Gráfico de barras de respuestas de encuesta

4.3. Resultados alcanzados

En general, el sistema cumple la función principal que tiene un juez en línea, que es la de evaluar programas informáticos de forma automática.

Pero también, según la Figura 31 en las respuestas de la encuesta presentada, algunos niños piden que se implemente una sección que provea materiales de aprendizaje dentro de la plataforma. El sistema de juez en línea “Odisea” provee una sección de tutoriales donde éstos tendrán toda la información necesaria y suficiente para aprender programación básica y competitiva, se podrán descargar en formato pdf, y una sección de mini juego interactivo para los más pequeños. Según la Figura 32 la implementación del mini juego fue una gran idea, pues los niños aprenden jugando.

También se observa en la Figura 31 que la idea de tener un juez virtual con insignias digitales como ser: creación de avatar, nombres de usuario con colores e insignias, puede motivar al usuario para que ingrese al sistema de juez en línea de forma recurrente y así el usuario se convierta en un competidor con más nivel gracias a la práctica que tendrá dentro del juez en línea “Odisea”. Así Bolivia puede aumentar de puesto en los rankings mundiales de los concursos más prestigiosos de programación competitiva.

Capítulo 5

Conclusiones y Recomendaciones

A lo largo del capítulo se mencionan las conclusiones a las que se llegó de acuerdo a los objetivos y resultados planteados en los anteriores capítulos. Por otro lado, se mencionan las recomendaciones dirigidas a los usuarios que estén interesados en utilizar el sistema de juez en línea “Odisea” como medio de enseñanza y entrenamiento, también se mencionan recomendaciones a los usuarios con privilegios de docente para que puedan hacer un uso correcto del sistema. Finalmente se mencionan sugerencias para la persona que esté interesada en realizar un proyecto similar en un futuro y así poder mejorarlo.

5.1 Conclusiones

Tomando en cuenta los objetivos planteados anteriormente, se llegaron a las siguientes conclusiones:

- Se logró diseñar el juez en línea “Odisea” con una interfaz amigable e intuitiva, es decir que el sistema cumple con aspectos de usabilidad en un 90% según las pruebas de calidad.
- El sistema logra evaluar de forma precisa y de forma instantánea los problemas algoritmos presentados en lenguaje C++, C, Java, Python.
- Se logró implementar el espacio llamado “Aulas” dedicado a una mejor interactividad entre tutor y estudiante.
- Se logró implementar un material didáctico dentro de la sección de tutoriales. En cada tutorial se explica el tema de forma didáctica, se sugiere un problema propuesto y se explica la resolución del mismo.
- Se logró crear un espacio de jugabilidad, para que el usuario pueda aprender conceptos básicos de programación mientras juega.

En general, se logró que el juez virtual “Odisea” esté desarrollado con elementos de jugabilidad para apoyar a la enseñanza y a la práctica constante de algoritmos y programación a niños y adolescentes.

5.2 Recomendaciones

Se proponen las siguientes recomendaciones para el sistema:

- Los usuarios pueden realizar sugerencias al sistema para mejorar el mismo en un futuro.
- Los usuarios con privilegio de docente deben tener una redacción coherente y una ortografía correcta al momento de subir problemas al sistema o también al momento de publicar post de usuario para que se mantenga la buena imagen del sistema.
- Instalar los paquetes de nodejs llamado node package manager (npm) al proyecto una vez clonado o descargado desde github para poder compilarlo correctamente.
- Proporcionar más juegos educativos respecto a programación dentro del sistema.
- Realizar una API que conecte el motor de evaluación con otras aplicaciones externas, para que pueda evaluar archivos de forma externa.
- Realizar backup constantemente de la base de datos para que no hayan pérdidas de datos en caso de sufrir inyecciones sql.

Bibliografía

About Code.org. (s.f.). Recuperado en Febrero 22, 2020, de
<http://code.org/international/about>

About: CodeChef. (s.f.). Recuperado en Febrero 22, 2020, desde
<https://www.codechef.com/aboutus/>

About IOI. (s.f.). Recuperado en Febrero 22, 2020, from <https://ioi2020.sg/about-ioi/>

About project. (s.f.). Recuperado en Febrero 22, 2020, desde
<https://www.e-olymp.com/en/pages/about>

Arango M., M. P. (2018, Octubre 10). ¿Saber programar será tan importante como aprender inglés? El Tiempo. Recuperado de:
<https://www.eltiempo.com/tecnosfera/novedades-tecnologia/vale-la-pena-aprender-a-programar-267858>

Code Hunt. (2020, March 13). Recuperado en Febrero 22, 2020, desde
<https://www.microsoft.com/en-us/research/project/code-hunt/>

Code Jam. (s.f.). Recuperado en Febrero 22, 2020, desde
<https://codingcompetitions.withgoogle.com/codejam/about>

Coding games and programming challenges for beginner and advanced. (s.f.). Recuperado en Febrero 22, 2020, desde <https://checkio.org/>

Coding Games and Programming Challenges to Code Better. (s.f.). Recuperado en Febrero 22, 2020, desde <https://www.codingame.com/start>

Contest Instructions and Rules.(s.f.). Recuperado en Febrero 22, 2020, from
<http://www.usaco.org/index.php?page=instructions>

Countries. (s.f.). International Olympiad in Informatics - Statistics. Recuperado en

Febrero 21, 2020, de <https://stats.ioinformatics.org/countries/>

Del Sastre. B. (2010, Septiembre). *Norma ISO/IEC 9126:2001*
https://www.academia.edu/20851916/Norma_ISO_IEC_9126_2001_Inicio

Design & Build High-Quality Software with On-Demand Talent. (2020, February 22).
Recuperado en Febrero 22, 2020, desde <https://www.topcoder.com/>

Facebook Hacker Cup. (s.f.) Información. Recuperado en Febrero 22, 2020, desde
<https://www.facebook.com/hackercup>

HackerRank. (s.f.). Recuperado en Febrero 22, 2020 desde
<https://www.hackerrank.com/>

Hora del Código: únete al movimiento. (s.f.). Recuperado en Febrero 22, 2020, desde
<https://hourofcode.com/es>

Hughes, C. (s.f.). About - Project Euler. Recuperado en Febrero 22 2020, from
<https://projecteuler.net/>

ICPC. (s.f.). Recuperado en Febrero 21, 2020, de <https://icpc.global/regionals/abouticpc>

ICPC Live Archive - Home. (s.f.). Recuperado en Febrero 22, 2020, desde
<https://icpcarchive.ecs.baylor.edu/>

Kurnia, A., Lim, A., & Cheang, B. (2001). *Online Judge. Computers & Education*, 36(4), 299-315. doi:10.1016/s0360-1315(01)00018-5

Lara, W. (2015, July 31). *¿Cómo funciona la metodología Scrum?*. Recuperado en Febrero 21,2020, de: <https://platzi.com/blog/metodologia-scrum-fases>

Learn to Code - for Free.(s.f.). Recuperado en Febrero 22, 2020 desde
<https://www.codecademy.com/>

Moxie, -. (s.f.). About Exercism. Recuperado en Febrero 22, 2020 desde

<https://exercism.io/about>

Olimpiada Boliviana de Informatica. (s.f.). Recuperado en Febrero 22, 2020, desde <https://olimpiada.icpc-bolivia.edu.bo/>

Pilow. (s.f). Leek Wars. Recuperado en Febrero 22, 2020, desde <https://leekwars.com/about>

Python Challenge - About. (s.f.). Recuperado en Febrero 22, 2020, desde <http://www.pythonchallenge.com/about.php>

Rating: users participated in recent 6 months. (s.f). Codeforces. Recuperado en Febrero 21, 2020, de: <http://www.codeforces.com/ratings>

Red de Programación Competitiva. (s.f.). Recuperado en Febrero 22, 2020, de <https://www.facebook.com/RedProgramacionCompetitiva>

Revilla, M. A., Manzoor, S., & Liu, R. (2008). *Competitive learning in informatics: The UVa online judge experience*. *Olympiads in Informatics*, 2(10), 131-148.

Revista, U. (2020, January 07). *Programación para niños, ¿cuáles son sus ventajas?*. Recuperado en Febrero 22, 2020, desde <https://www.unir.net/educacion/revista/noticias/programacion-para-ninos/549204772898/>

Sánchez, W. (2011). La usabilidad en Ingeniería de Software: definición y características. *Revista Ing-Novación*, (2), pp. 7-22.

Sphere Online Judge (SPOJ) - Info. (s.f.). Recuperado en Febrero 22, 2020, de <https://www.spoj.com/info/>

Sutherland, J., & Schwaber, K. (2013). *The scrum guide. The definitive guide to scrum: The rules of the game*. Scrum. org, 268.

Wasik, S., Antczak, M., Badura, J., Laskowski, A., & Sternal, T. (2018). *A Survey on Online Judge Systems and Their Applications*. *ACM Computing Surveys*, 51(1), 1-34.
doi:10.1145/3143560

Anexos

Matriz de marco lógico

Jerarquías de objetivos	Indicadores	Medios de verificación	Supuestos
FIN: Promover o incentivar al aprendizaje de programación en niños y adolescentes.	<ul style="list-style-type: none"> • Frecuencia de uso de la plataforma. • Número de usuarios registrados 	<ul style="list-style-type: none"> • Inspección al contador de la plataforma. • Inspección a la base de datos 	Disponibilidad del servidor.
PROPÓSITO: Desarrollar una plataforma de autoaprendizaje en programación para niños.	Disponibilidad de la plataforma en línea hasta abril 2020.	Inspección técnica	<ul style="list-style-type: none"> • Continuar con financiación. • Disponibilidad en el servidor.
COMPONENTES: Sistema de evaluación y unidades didácticas.	<ul style="list-style-type: none"> • Encontrar disponible el sistema probado en el mes de diciembre. 	<ul style="list-style-type: none"> • Inspección técnica 	Contar con fuentes de financiamiento.
ACTIVIDADES: Actividades del desarrollo de unidades didácticas. <ul style="list-style-type: none"> • Estudiar didáctica. • Estudiar juegos con programación. • Diseñar e implementar los módulos. • Evaluación de módulos. 	<ul style="list-style-type: none"> • Buscar en internet. • Tener conexión de internet. • En la computadora, tener un software con alguna especificación. 	<ul style="list-style-type: none"> • Costo del internet al mes. • Costo. • Comprar computadora o licencia. 	Contar con fuentes de financiamiento

Documentación

La Paz, 5 de octubre de 2020

Señor:

Lic. Eufren Llanque

DIRECTOR A.I.

CARRERA DE INFORMÁTICA

FACULTAD DE CIENCIAS PURAS Y NATURALES

UNIVERSIDAD MAYOR DE SAN ANDRÉS

Presente

Ref.- AVAL DE CONFORMIDAD Y SOLICITUD DEFENSA TESIS DE GRADO

De mi mayor consideración:

Mediante la presente, tengo a bien dirigirme a su autoridad para informarle, en mi calidad de tutora metodológica, que la universitaria Fabiola Vanessa Aliaga Salvatierra con CI 6122353 LP finalizó la tesis de grado:

**ODISEA: JUEZ VIRTUAL EN LÍNEA PARA COMPETENCIAS ALGORÍTMICAS
ORIENTADO A NIÑOS Y ADOLESCENTES**

Cumpliendo con el contenido y alcance de estudio, por lo que solicito se dé curso a la defensa pública, para optar al título de Licenciatura en Informática con mención en Ingeniería de Sistemas Informáticos de acuerdo con el reglamento vigente de la Universidad Mayor de San Andrés.

Sin otro particular, saludo a usted con las consideraciones más distinguidas.

Atentamente:



M. Sc. Rosa Flores Morales
Tutora Metodológica

La Paz, 5 de octubre de 2020

Sra:

M.Sc. Rosa Flores Morales

TUTOR METODOLÓGICO

Presente. -

Ref.- AVAL DE CONFORMIDAD DE TESIS DE GRADO

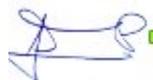
De mi mayor consideración:

Mediante la presente, tengo a bien dirigirme a su autoridad para informarle que luego de efectuar el seguimiento a la estructura y contenido de la tesis de grado titulada: **“ODISEA: JUEZ VIRTUAL EN LÍNEA PARA COMPETENCIAS ALGORÍMICAS ORIENTADO A NIÑOS Y ADOLESCENTES”**, elaborado por la universitaria Fabiola Vanessa Aliaga Salvatierra con CI 6122353 LP, habiendo realizado las respectivas correcciones a mis observaciones.

En calidad de asesor expreso mi conformidad con el contenido y la forma de trabajo, dando mi aval para que la postulante pueda realizar la defensa de la Tesis de Grado, para optar al título de Licenciatura en Informática con mención en Ingeniería de Sistemas Informáticos, de acuerdo a normas y reglamentos vigentes.

Sin otro particular, me despido de su persona con las consideraciones más distinguidas.

Atentamente:



Lic. Jorge Humberto Terán Pomier

ASESOR