

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES



PROYECTO DE GRADO

**“REGISTRO, INSCRIPCIÓN Y SEGUIMIENTO CASO: OLIMPIADA
BOLIVIANA DE INFORMÁTICA (OBI)”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: ABRAHAM VILLCA ALARCON

TUTOR METODOLOGICO: M.SC. CUEVAS QUIROZ FRANZ

ASESOR: M.SC. TERAN POMIER JORGE HUMBERTO

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.

CAPÍTULO I	1
INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. Objetivos de la entidad.....	4
1.2.2. Trabajos similares.....	4
1.3. PLANTEAMIENTO DEL PROBLEMA	5
1.4. JUSTIFICACIÓN	6
1.5. OBJETIVOS	7
1.5.1. Objetivo General	7
1.5.2. Objetivos Específicos	7
1.6. ALCANCES O LIMITACIONES.....	7
1.7. APORTES.....	8
1.8. DEFINICION DE TERMINOS	8
CAPÍTULO II.....	9
MARCO TEÓRICO.....	9
2.1. OLIMPIADAS BOLIVIANAS DE INFORMATICA	9
2.1.1. Requisitos Indispensables	10
2.1.2. Etapas y Clasificación.....	10
2.1.3. Tutores	12
2.1.4. Competencia.....	12
2.2. CMS.....	12
2.2.1. Estructura General	13
2.2.2. Servicios.....	13
2.2.3. Consideraciones de seguridad	15
2.2.4. Entorno Grafico	16
2.3. INGENIERIA DE SOFTWARE	16
2.3.1. Análisis de requerimientos.....	17
2.3.2. Especificación	17
2.3.3. Diseño y estructura	17
2.3.4. Programación.....	17
2.3.5. Prueba	17
2.3.6. Documentación.....	18
2.3.7. Mantenimiento.....	18
2.4. METODOLOGIA DE DESARROLLO DE SOFTWARE	18
2.4.1. Metodologías Tradicionales	19
2.4.2. Metodologías Agiles	19
2.4.3. Principales Metodologías Agiles	20
2.5. SCRUM PARA EL DESARROLLO DE SOFTWARE	21
2.5.1. Teoría Scrum	21
2.5.2. Roles y procesos de Scrum	22
2.5.2.1. Roles	22
2.5.2.2. Proceso.....	23
2.5.3. Fases de Scrum.....	25
2.5.3.1. Pre-juego	25

2.5.3.2.	Juego	25ix
2.5.3.3.	Post-juego.....	25
2.6.	LENGUAJES DE PROGRAMACION Y GESTORES DE BASES DE DATOS	25
2.7.	CALIDAD DE SOFTWARE – ISO/IEC 9126.....	26
2.8.	SEGURIDAD INFORMATICA	28
2.8.1.	Mecanismos de Seguridad	28
2.8.2.	Principios de Seguridad Informática	28
	CAPÍTULO III.....	35
	MARCO APLICATIVO	35
3.1.	PRE – GAME.....	35
3.1.1.	Concepción.....	35
3.1.2.	Indagación y elaboración	37
3.1.3.	Negociación.....	39
3.1.4.	Especificación	39
3.2.	GAME.....	44
3.2.1.	Sprint 0.....	44
3.2.2.	Sprint 1	46
3.2.3.	Sprint 2.....	46
3.2.4.	Sprint 3.....	46
3.3.	POST - GAME	46
3.3.1.	Validación	46
3.4.	SEGURIDAD.....	46
3.4.1.	Seguridad Administrativa	46
3.4.2.	Seguridad lógica	46
3.4.3.	Seguridad física	46
3.5.	CALIDAD DE SOFTWARE	46
3.5.1.	Usabilidad	46
3.5.2.	Eficiencia	47
3.6.	HERRAMIENTAS EMPLEADAS	47
4.	MARCO DE RESULTADOS.....	47
	BIBLIOGRAFIA	47
	ANEXOS	50
	ARBOL DE PROBLEMAS	50
	ARBOL DE OBJETIVOS	50

Lista de tablas

x

CAPITULO I

Tabla 1.1: Quienes pueden participar en dada área	3
--	---

CAPITULO II

Tabla 2.1: Lista de Servicios que componen del CMS	14
---	----

Tabla 2.3: Lenguajes de Programación y gestores de bases de datos.....	26
---	----

CAPITULO III

Tabla 3.1: Requerimientos funcionales del sistema.....	38
---	----

Tabla 3.2: Requerimientos no funcionales	38
---	----

CAPITULO II

Fig 2.1: Servicios y su s Interacciones con el CMS	15
Fig. 2.2: Entorno grafico de la Herramienta CMS	16
Fig 2.3: Metodologías Agiles	20
Fig 2.4: Componentes de la metodología Scrum	22
Fig 2.5 Roles de Screum	23
Fig 2.6: Proceso de desarrollo Scrum	24
Fig 2.7: Norma de Evaluación ISO/IEC 9126.....	27

CAPITULO III

Fig 3.1: Reuniones por diferentes plataformas digitales.....	36
Fig 3.2: Material digital del concurso año 2017.....	37
Fig 3.3: Forma de Ingreso al sistema.....	40
Fig 3.4: Arquitectura del sistema.....	40
Fig 3.5: Caso de uso Admin	41
Fig 3.6: Caso de uso Tutor	42
Fig 3.7: Caso de uso Estudiante	42
Fig 3.8: Caso de uso desarrollador	43

CAPÍTULO I

INTRODUCCIÓN

La Informática llegó a ser un pilar muy importante para la sociedad, con grandes usos en la educación, salud, ciencia. En fin, podemos definir la informática un elemento que nos rodea en el diario vivir, a la que la sociedad tiene una fuerte dependencia.

La informática es una herramienta muy útil por la crisis sanitaria que se está viviendo en el planeta. Diferentes instituciones, universidades y colegios optaron para utilizar múltiples medios de comunicación para continuar la educación.

En Bolivia como actualmente se está viviendo una crisis sanitaria se optó por tomar medidas para precautelar el año escolar utilizando diferentes medios de comunicación como las clases online por Meet o Classroom, en el cual los estudiantes podrían pasar los cursos de manera virtual.

La Olimpiada Internacional de Informática (IOI) es una más de las Olimpiadas Internacionales de Ciencias, que tiene como objetivo potenciar el aprendizaje de la informática en alumnos de secundaria y preparatoria (nivel medio y medio superior). Los problemas de esta competición están orientados a los algoritmos. No se requiere mucho conocimiento en cuanto al uso del API, pero sí habilidad en la resolución de problemas y en el diseño de algoritmos. Al concurso asisten 4 seleccionados de cada uno de los más

de 80 países que participan. Los 4 seleccionados son elegidos de entre miles de estudiantes que participan en las olimpiadas regionales y nacionales en sus países.

En Bolivia el Presidente Evo Morales Ayma Inauguro el año 2011 las Olimpiadas Científicas Estudiantiles orientada a los estudiantes de secundaria en distintas en distintas áreas como matemática, física, informática y otros, con el propósito de fomentar la ciencia y tecnología en el país, y a los ganadores se les incentivara con premios para continuar su educación y que representen en a Bolivia internacionalmente. Y también al clausurar las Olimpiadas Científicas Estudiantil Plurinacional en Cochabamba.

En Bolivia la OBI (Olimpiadas Bolivianas de Informática) es una competencia académica orientada a jóvenes bolivianos de secundaria (de tercero a sexto) que tienen gusto por la resolución de problemas prácticos usando lógica algorítmica y su programación en una computadora

La Olimpiada Boliviana de Informática seleccionara cada año a 10 estudiantes a quienes se los denominara como el equipo Bolivia y que participa en diferentes competencias para seleccionar entre ellos a los 4 estudiantes que representan a Bolivia en la competencia internacional IOI.

1.2. ANTECEDENTES

El ministerio de Educación a través del Viceministerio de Ciencia y Tecnología, en coordinación con el viceministerio de coordinación del Viceministerio de coordinación regular, Universidades Públicas y Privadas, Universidades Pedagógicas, Escuelas Superiores de Formación de Maestros, Direcciones Departamentales y Distritales de Educación, convoca todos los años a las Olimpiadas Científica Estudiantil Plurinacional Boliviana (OCEP).

En el cual pueden participar todos los estudiantes bolivianos del Subsistema de educación Regular de todas las unidades educativas fiscales, privadas y de convenio del estado Plurinacional de Bolivia; en las áreas de: MATEMARICA, FÍSICA, QUÍMICA, BIOLOÍA, GEOGRAFÍA, ASTRONOMÍA/ASTROFÍSICA, INFORMÁTICA, ROBÓTICA Y FERIA

CIENTIFICA, los estudiantes que pueden participar en las distintas áreas lo detallamos en la Tabla 1.1 dependiendo el año de escolaridad:

Tabla 1.1: Quienes pueden participar en dada área

<i>AREA</i>	<i>AÑO DE ESCOLARIDAD</i>
Física	De primero a sexto de secundaria
Matemática	De primero a sexto de secundaria
Biología	De segundo a sexto de secundaria
Química	De segundo a sexto de secundaria
Geografía	De tercero a sexto de secundaria
Astronomía/Astrofísica	De primero a sexto de secundaria
Informática	De quinto de primaria a sexto de secundaria
Robótica	De primero de primaria a sexto de secundaria
Feria Científica	Primero, segundo, quinto y sexto de secundaria

Fuente: Elaborado con base en (OCEP, 2012).

El “Bolivia Tech Hub Collaborative” y la coordinación de la ACM ICPC Bolivia convocan a estudiantes bolivianos del sistema de Educación Regular de las unidades educativas públicas, privadas y de convenio a participar de la Olimpiada Boliviana de Informática que clasificara a los representantes bolivianos a la CIIP (Competencia de Informática y Computación) online de IOI (International Olympiad in Informatics) -2021 (Singapur).

La Olimpiada Boliviana de Informática es una competencia académica, que inicio el año 2011, orientada a jóvenes de secundaria (de tercero a sexto) que tienen gusto por resolver problemas prácticos usando lógica, algoritmos y la programación en una computadora. Esta olimpiada forma parte del Sistema Internacional de Olimpiadas Científicas con su evento dominado Olimpiadas Internacionales de Informática. (IOI – International Olympiad in Informatics).

La presente gestión será desarrollada de forma virtual en todas las etapas, incluida la final, Teniendo cada una de ellas grado de control para asegurar su correcto desarrollo y que serán notificados a los participantes, tutores y padres.

1.2.1. Objetivos de la entidad

- Contribuir al mejoramiento de la enseñanza y del aprendizaje de la informática en todas las unidades educativas del país.
- Identificar y preparar a los jóvenes talentosos para que represente a sus departamentos y al país en competencias nacionales e internacionales de programación en formatos presenciales físicos e internet(online).
- Fomentar valores de solidaridad, compañerismo y amistad entre estudiantes, maestros y maestras.
- Fomentar el intercambio de conocimientos y experiencias entre maestros, maestras y estudiantes participantes en cada una de las etapas.

1.2.2. Trabajos similares

Para el Desarrollo del presente Proyecto, se vio la necesidad de revisar algunos trabajos similares que son utilizados para el entrenamiento y evaluación de las olimpiadas Internacionales de informática.

- Plataforma virtual de ‘TitanCode’ quien fue evaluador oficial en las primeras gestiones 2017 y 2018, en el cual se realizaban las prácticas y evaluaciones en sus distintas etapas y niveles de las olimpiadas.
- Plataforma virtual ‘OmegaUp’ donde los estudiantes de las Olimpiadas mexicanas en informática realizan sus entrenamientos y concursos.
- Plataforma Virtual ‘Caribbean Online Judge (COJ)’ Es un juez en línea para entrenar la programación de algoritmos en diferentes lenguajes. Su desarrollo comenzó en 2006 por programadores de la UCI y otros de la comunidad caribeña de la ACM-ICPC.

- Plataforma Virtual ‘URI Online Judge’ es un proyecto desarrollado por los departamentos de ciencias de la computación de la URI. Su objetivo principal es promover la práctica de la programación y ayuda a compartir el conocimiento.

1.3. PLANTEAMIENTO DEL PROBLEMA

Como las Olimpiadas Bolivianas de Informática se encarga de seleccionar a los mejores estudiantes de nivel secundario que nos representaran en las Olimpiadas Internacionales de Informática, su principal objetivo es seleccionar y capacitar de manera adecuada a los representantes de nuestro país, el mismo también están en competencia con otros países que prepararan a los que presentaran a los finalistas de su región. Entonces es un trabajo conjunto entre estudiantes, capacitadores, tutores y todo tipo de herramienta útil para la capacitación de los estudiantes que representaran al país.

En referencia a nuestro meta que es capacitar de manera adecuada a nuestros estudiantes encontramos las siguientes problemáticas que nos limitan las metas propuestas.

- Demora en la centralización de datos e integración con la herramienta CMS de Olimpiadas Bolivianas de Informática.
- Inscripción y control de los estudiantes en todos los diferentes niveles y etapas de la olimpiada de manera semi-manual.
- La carencia de un espacio confiable para los reportes y comunicados referentes a las Olimpiadas Bolivianas de Informática.
- Los concursos ya realizados y problemas propuestos durante cada versión y etapas de las olimpiadas se almacenan de manera física y digital en plantillas Excel.

¿Cómo podemos realizar un control adecuado de los estudiantes y capacitadores en los diferentes niveles y etapas de la OBI, dando como resultado una información confiable, coherente y que siga todas las normas institucionales para este propósito?

1.4. JUSTIFICACIÓN

Las metodologías y herramientas a emplear para el estudio y desarrollo de un Sistema de Gestión, son recursos disponibles y al alcance de cualquier institución o persona, tal es el caso de la metodología Scrum que es una metodología ágil y simple de desarrollo, en cuanto a las herramientas de desarrollo de software están disponibles en el mercado a un costo accesible, para la parte Front - End utilizaremos BOOSTRAP y Responsive Design, en la parte Back - End Utilizaremos React.js para el gestor de base de datos utilizaremos PgAdmin3, en la seguridad utilizaremos Jwt tokens.

El Sistema de Gestión tiene acceso para la plataforma Ubuntu de la versión 18.04 en el cual se está ejecutando actualmente el CMS de las olimpiadas que contenga un explorador web FireFox, Mozilla, Google Chrome.

La calidad de servicio que brinda el Sistema es tener una constante comunicación a todas las personas que pertenecen a la OBI, también ayudara mucho al tutor en el ahorro de tiempo en averiguar sobre el rendimiento de sus estudiantes. De igual forma facilitara al Administrador para realizar toda la Gestión Académica que se necesita para controlar y evaluar a los Estudiante. Toda información será en tiempo real utilizando la tecnología Web el cual será sobre plataforma mixta y fácil acceso por tener un diseño adaptable en la Web.

El Sistema de Gestión Académica de los diferentes niveles y etapas de la olimpiada, brinda un apoyo para guardar su información de manera segura en el sistema utilizando el repositorio de datos para cada uno de los integrantes del panel administrativo, tutor y estudiante.

En cuanto a las herramientas que se aplicaran en el sistema se implementara los lenguajes libres para el desarrollo web y base de datos, Para ello no tendrá ningún costo en cuanto a licencias comerciales u otras ya que son de manera gratuita.

1.5. OBJETIVOS

1.5.1. Objetivo General

Implementar en Software para la gestión y administración de los tutores, estudiantes y capacitadores en las diferentes etapas y niveles de las olimpiadas bolivianas de informática.

1.5.2. Objetivos Específicos

- Identificar los requerimientos del sistema, mediante entrevistas con el usuario.
- Determinar los atributos necesarios a registrar dentro del sistema.
- Desarrollar la interfaz del sistema de gestión.
- Diseñar formularios digitales para que los tutores puedan realizar la inscripción de los estudiantes.
- Diseñar un panel de inscripción para registrar a los capacitadores de los distintos colegios.
- Diseñar un Buscador de estudiantes por nombre, numero de carnet o RUDE
- Elaborar un espacio parar publicar los materiales didácticos, educativos para que los estudiantes puedan acceder a los mismos.

1.6. ALCANCES O LIMITACIONES

El proyecto está diseñado específicamente para la evaluación y control de las olimpiadas bolivianas de informática en sus diferentes etapas, en el cual será unas herramientas de apoyo para tutores y estudiantes en el proceso de capacitación para las evaluaciones.

El presente trabajo no realizara la evaluación de algoritmos, para ello se tiene una herramienta externa CMS-IOI, El acceso a las diferentes etapas de evaluación no depende de los tutores sino del rendimiento en las evaluaciones pasadas de las cuales solo un porcentaje aprobara a la siguiente etapa.

El presente proyecto tendrá material de ayuda para las diferentes etapas y niveles de las evaluaciones con la finalidad de mantener a los estudiantes al tanto del tipo de evaluación y el contenido de los mismos que rendirán en las distintas etapas de la olimpiada.

1.7. APORTES

Las olimpiadas bolivianas de Informática no cuentan con un sistema de gestión adecuado para sus diferentes niveles y etapas de evaluación, por el cual queremos mejorar el rendimiento y eficiencia del proceso administrativo e interacción con el Juez Virtual CMS-IOI.

1.8. DEFINICION DE TERMINOS

Jwt tokens.- es un estándar que está dentro del documento RFC 7519.

En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios.

Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro de del payload o cuerpo de un mensaje que va firmado digitalmente.

CMS-IOI.- o Contest Management System, es un sistema distribuido para ejecutar y (hasta cierto punto) organizar un concurso de programación.

CMS ha sido diseñado para ser general y manejar muchos tipos diferentes de concursos, tareas, puntuaciones, etc. No obstante, CMS se ha creado explícitamente para ser utilizado en la Olimpiada Internacional de Informática 2012, celebrada en septiembre de 2012 en Italia.

CAPÍTULO II

MARCO TEÓRICO

2.1. OLIMPIADAS BOLIVIANAS DE INFORMATICA

El sistema educativo plurinacional en Bolivia está regulado por la Ley 070 de educación Abelino Siñani –Lizardo Perez, el mismo está compuesta por las instituciones educativas fiscales (publicas), instituciones educativas privadas y de convenio. la educación regular dentro del sistema educativo comprende tres subsistemas:

- **Educación Preescolar:** Está destinada a los menores de 0 a 6 años debe orientar a los padres de familia y comunidad para lograr el desarrollo de las capacidades y vocación del niño.
- **Educación Primaria:** Comprende la formación básica, brinda condiciones y necesidades de pertenencia de las y los estudiantes; desarrolla sus conocimientos, potencialidades y capacidades. De 6 años de duración.
- **Educación Secundaria:** Articula la educación humanística y la educación técnica - tecnológica con la producción, incorporación y formación histórica, cívica y comunitaria. De 6 años de duración.

Las Olimpiadas Bolivianas de Informática están destinadas para los estudiantes que se encuentran en los niveles de primaria y secundaria los mismos podrán participar en los distintos niveles de la olimpiada. estas tendrán etapas de evaluación para seleccionar a los mejores estudiantes de cada nivel.

Para los niveles 2 y 3 se aceptan diferentes lenguajes de programación en la resolución de los problemas.

- **C:** También conocido como ANSI C, un compilador cuya principal ventaja es el manejo de apuntadores, casting, rapidez, extrema flexibilidad y muchas formas de abreviar código.
- **C++:** Una versión orientada a objetos de C, incluye muchas librerías muy útiles como son las pilas, colas, etcétera, además que ofrece una mejor flexibilidad en el área de la declaración de variables.
- **C++11:** Es una versión del lenguaje de programación C++, añade algunas cosas al núcleo como el soporte multihilo.
- **Python:** un lenguaje de programación de propósito general muy poderoso y flexible, a la vez que sencillo y fácil de aprender.
- **JAVA:** El lenguaje de programación de moda, ofrece todo lo que los demás compiladores ofrecen y muchas cosas más, se puede decir que lo tiene todo, sin embargo, debido a su naturaleza que requiere de una máquina virtual para la ejecución de sus programas, es extremadamente lento. No recomendable para muchos problemas donde se requiere velocidad. Es muy eficaz en el proceso de depuración debido al manejo de excepciones que tiene integrado.

2.1.1. Requisitos Indispensables

Participaran Estudiantes de quinto de primaria a sexto de secundaria en unidades educativas fiscales, privadas y de convenio de todo el país, que cumplen con los siguientes requisitos:

- Ser estudiante de nacionalidad boliviana.
- Haber estado inscrito legalmente en la gestión 2020, contar con un RUDE válido y el SIE de la unidad educativa a la que pertenece.

2.1.2. Etapas y Clasificación

- **Primera Etapa:** El proceso de selección e inscripción es definido por la unidad educativa. Sin embargo, pueden participar de forma individual siempre que se tenga

permiso de los padres o representante legal dadas las condiciones especiales de la presente gestión debido a la emergencia sanitaria.

- **Segunda Etapa:** Las pruebas serán elaboradas por parte del Comité de Problemsetters, profesionales expertos en Informática que ayudan en la redacción de los problemas en las diferentes etapas de la Olimpiada desde el 2011. Estas serán provistas a través de un sistema evaluador de pruebas que construirá pruebas individuales para cada estudiante que esté registrado en el sistema de inscripción (denominado Juez Virtual).
- **Tercera Etapa:** Las pruebas serán elaboradas por parte del Comité de Problemsetters, profesionales expertos en Informática que ayudan en la redacción de los problemas en las diferentes etapas de la Olimpiada desde el 2011. Estas serán provistas a través de un sistema evaluador de pruebas que construirá pruebas individuales para cada estudiante que esté registrado en el sistema de inscripción y haya aprobado la segunda etapa (denominado Juez Virtual).
- **Cuarta Etapa:** Las pruebas serán elaboradas por el Comité de Problemsetters. La evaluación será de forma virtual empleando el Juez Virtual CMS, para los niveles 2 y 3. La evaluación de Nivel 1 (videojuegos y unicodemy) se realizará de forma online y se evaluarán la presentación, funcionalidad y desarrollo de retos asignados a los equipos y competidoras, todas(os) deben proporcionar su documento de proyecto y código de su aplicación cargados en la plataforma designada.

Para los niveles 1, 2, 3, en caso de existir un empate en la nota, se dirimirá analizando la calidad del código fuente, por parte de los miembros del Comité Nacional de Informática que estén presentes en la evaluación, Para los niveles 2 y 3, en todas las etapas se utilizará el juez Virtual CMS (Contest Management System)

El entrenamiento del equipo preolímpico inicia después de la cuarta etapa y estará a cargo del Comité del ICPC Bolivia.

2.1.3. Tutores

Cada unidad educativa asignara un tutor o tutora (coach, entrenador(a)) de equipos, que se encargara del entrenamiento de sus estudiantes. Es recomendable que el tutor o tutora tenga conocimiento en programación de algoritmos en lenguajes C/C++/Java. El tutor o tutora puede ser el maestro o maestra de computación asignado a la unidad educativa o un tutor designado o avalado por el Director de la Unidad Educativa.

El tutor o tutora debe tener una dirección de correo electrónico válida para registrar al momento de inscribir a sus estudiantes.

2.1.4. Competencia

Los estudiantes competirán observando el honor Olímpico, dando estricto cumplimiento al Reglamento de Competencias para el área Informática (Ver Anexo I).

2.2. CMS

CMS, o Contest Management System, es un sistema distribuido para ejecutar y (hasta cierto punto) organizar un concurso de programación.

CMS ha sido diseñado para ser general y manejar muchos tipos diferentes de concursos, tareas, puntuaciones, etc. No obstante, CMS se ha creado explícitamente para ser utilizado en la Olimpiada Internacional de Informática 2012, celebrada en septiembre de 2012 en Italia.

La herramienta CMS es un software para organizar concursos de programación similares a concursos internacionales conocidos como el IOI (International Olympiad in Informatics). Fue escrito y recibió contribuciones de personas involucradas en la organización de concursos similares a nivel local, nacional e internacional, y se usa regularmente para tales concursos en muchos países diferentes. Está destinado a ser seguro, extensible, adaptable a diferentes situaciones y fácil de usar.

2.2.1. Estructura General

El sistema está organizado de forma modular, con diferentes servicios que se ejecutan (potencialmente) en diferentes máquinas y proporcionan capacidad de ampliación a través de réplicas de servicios en varias máquinas.

El estado del concurso se guarda en su totalidad en una base de datos PostgreSQL (no se admiten otros DBMS, ya que CMS se basa en la función Objeto grande (LO) de PostgreSQL). Es poco probable que en el futuro apuntemos a diferentes bases de datos.

Siempre que la base de datos funcione correctamente, todos los demás servicios se pueden iniciar y detener de forma independiente. Por ejemplo, el administrador puede reemplazar rápidamente una máquina rota por otra idéntica, que asumirá sus funciones (sin tener que mover información de la máquina averiada). Por supuesto, esto también significa que CMS depende completamente de la base de datos para ejecutarse. En contextos críticos, es necesario configurar la base de datos de forma redundante y estar preparado para realizar una conmutación por error rápidamente en caso de que ocurra algo malo. La elección de PostgreSQL como base de datos a utilizar debería facilitar esta parte, ya que existen muchas soluciones diferentes, maduras y bien conocidas para proporcionar tales procedimientos de redundancia y conmutación por error.

2.2.2. Servicios

Los servicios, enumerados en la Tabla 2.1, son programas de Python creados sobre AsyncLibrary, una biblioteca RPC a medida utilizando el Asyncore Framework (Sam Rushing y Python

Fundación del Software, 1996). Algunos de los servicios, llamados servidores, proporcionan también un segundo interfaz gracias al web Framework Tornado (Facebook, Inc.,

2009), para que ambos puedan interactuar con los otros servicios y servidores web a administradores o concursantes. Las interacciones entre los servicios se muestran en la Fig. 2.1.

En la Tabla 2.1 la lista de servicios que componen CMS. Replicable es "Sí" cuando varias instancias del servicio pueden cooperar Para aumentar la capacidad Los dos últimos servicios (ContestWebServer y AdminWebServer) también son servidores.

Tabla 2.1: Lista de Servicios que componen del CMS

<i>Nombre</i>	<i>Replicable</i>	<i>Deberes</i>
LogService	No	Recibir, agregar y mostrar todos los registros del sistema.
Worker	Yes	Ejecutar compilaciones y evaluaciones de envíos en un entorno seguro.
EvaluationService	No	Mantener la cola de los trabajos que se asignarán a los trabajadores.
ScoringService	No	Transformar los resultados de la evaluación en puntajes y comunicar ellos a los rankings en vivo.
Checker	No	Llamar a la función de latido de todos los servicios
ResourceService	Yes	Recopilar información de uso de recursos sobre la máquina en la que se encuentra ejecutando e iniciando todos los servicios en una máquina.
ContestWebServer	Yes	Servir páginas web a los concursantes, aceptar presentaciones y proporcionar comentarios.
AdminServer	No	Servir páginas web a los administradores, configurar y administrar los concursos.

Fuente: Elaborado con base en (OCEP, 2012).

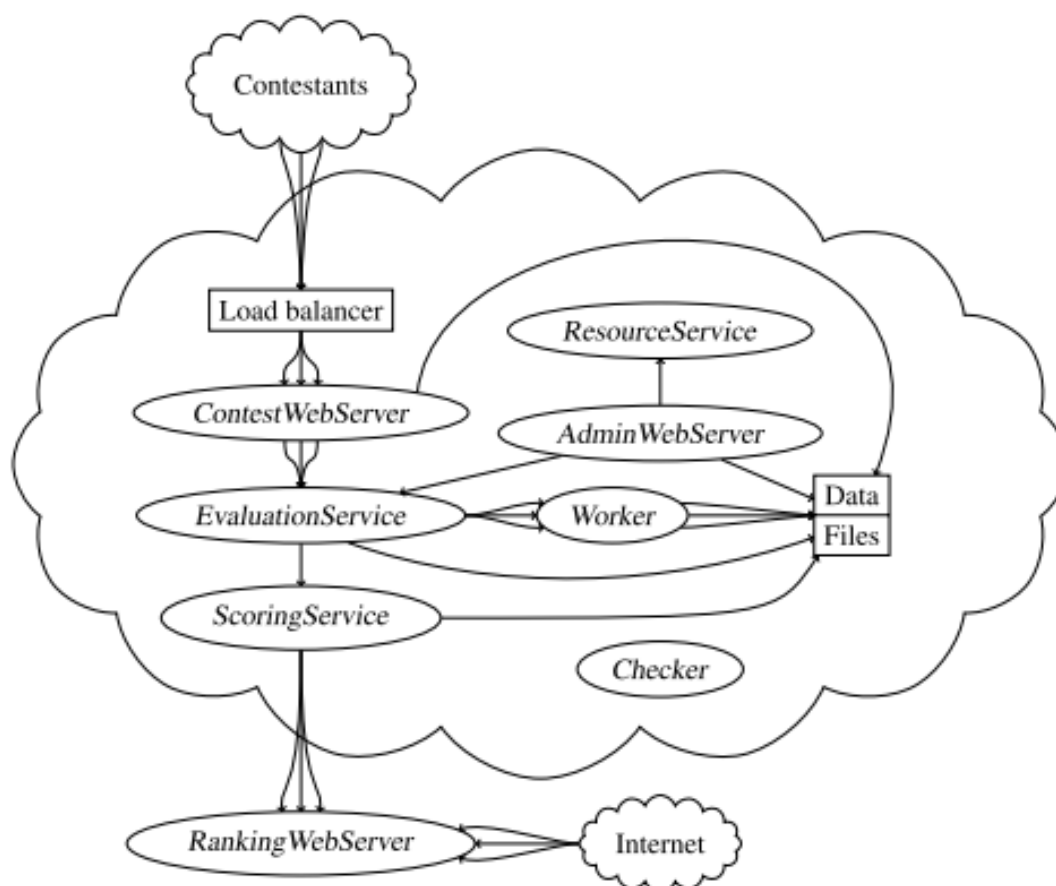


Fig 2.1: Servicios y sus Interacciones con el CMS

Fuente: Elaborado con base en (Maggiolo, Mascellani, 2012).

Las flechas representan una solicitud de algún tipo, y las puntas de flechas múltiples significan que el servicio se puede replicar. Los servicios Checker y ResourceService hablan con todos los demás servicios, y se omiten las flechas relativas.

2.2.3. Consideraciones de seguridad

Con la excepción de RWS, no existen esquemas criptográficos o de autenticación entre los distintos servicios o entre los servicios y la base de datos. Por lo tanto, es obligatorio mantener los servicios en una red dedicada, aislándolos adecuadamente a través de firewalls de los concursantes o de las computadoras de otras personas. Este tipo de operaciones, como

también evitar que los concursantes se comuniquen y hagan trampa, es responsabilidad del administrador y no es administrado por el propio CMS.

2.2.4. Entorno Grafico

Para visualizar el entorno grafico del CMS podemos iniciar el servidor con el siguiente comando “cmsAdminWebServise” y podremos acceder en el puerto localhost:8889 el cual nos pedirá el acceso al mismo con la contraseña que se proporcionó a la hora de instalar el CMS, la fig. 2.2 nos muestra la interfaz de inicio de sesión y el entorno de administración.

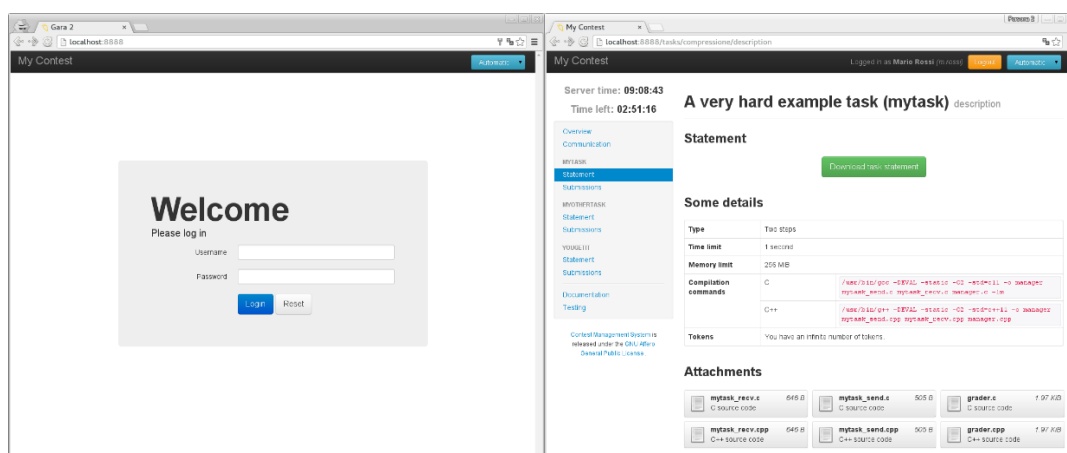


Fig. 2.2: Entorno grafico de la Herramienta CMS

Fuente: Elaborado con base en (Elaboración propia, 2020).

2.3. INGENIERIA DE SOFTWARE

La ingeniería del software es el proceso formal de desarrollo de software en el que las necesidades del usuario se traducen en requerimientos, estos se transforman en diseño que se implementa en código que se prueba, documenta y se certifica para su uso operativo. La ingeniería del software se define como “(1) la aplicación de un método sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, esto es, la aplicación de la ingeniería al software” y “(2) el estudio de los métodos de (1)” [Xavi, 2013].

El proceso requiere una metodología con las siguientes etapas:

2.3.1. Análisis de requerimientos

Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios. Usualmente el cliente/usuario tiene una visión incompleta/inexacta de lo que necesita y es necesario ayudarlo para obtener la visión completa de los requerimientos. El contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible. [Xavi, 2013].

2.3.2. Especificación

Es la tarea de describir detalladamente el software a ser escrito, de una forma rigurosa. Se describe el comportamiento esperado del software y su interacción con los usuarios y/o otros sistemas. [Xavi, 2013]

2.3.3. Diseño y estructura

Determinar cómo funcionará de forma general sin entrar en detalles incorporando consideraciones de la implementación tecnológica, como el hardware, la red, etc. Consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la segunda etapa también conocidas como las entidades de negocio. Generalmente se realiza en base a diagramas que permitan describir las interacciones entre las entidades y su secuenciado. [Xavi, 2013]

2.3.4. Programación

Se traduce el diseño a código. Es la parte más obvia del trabajo de ingeniería de software y la primera en que se obtienen resultados “tangibles”. No necesariamente es la etapa más larga ni la más compleja, aunque una especificación o diseño incompletos/ambiguos pueden exigir que, tareas propias de las etapas anteriores se tengan que realizarse en esta. [Xavi, 2013]

2.3.5. Prueba

Consiste en comprobar que el software responda/realice correctamente las tareas indicadas en la especificación. Es una buena praxis realizar pruebas a distintos niveles (por ejemplo, primero a nivel unitario y después de forma integrada de cada componente) y por equipos diferenciados del de desarrollo (pruebas cruzadas entre los programadores o realizadas por un área de test independiente). [Xavi, 2013]

2.3.6. Documentación

Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema. Las tareas de esta etapa se inician ya en la primera fase, pero sólo finalizan una vez terminadas las pruebas. [Xavi, 2013]

2.3.7. Mantenimiento

En esta etapa se realizan un mantenimiento correctivo (resolver errores) y un mantenimiento evolutivo (mejorar las funcionalidades y/o dar respuesta a nuevos requisitos). [Xavi, 2013]

2.4. METODOLOGIA DE DESARROLLO DE SOFTWARE

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado. [Maida, EG, Pacienza, 2015]

Si esto se aplica a la ingeniería del software, podemos destacar que una metodología:

- Optimiza el proceso y el producto software.
- Métodos que guían en la planificación y en el desarrollo del software.
- Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

2.4.1. Metodologías Tradicionales

También denominadas metodologías pesadas estas imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. [Maida, EG, Pacienza, 2015]

2.4.2. Metodologías Ágiles

Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo.

Un modelo de desarrollo ágil, generalmente es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento). Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para los equipos de trabajo, evitando de esta manera los caminos burocráticos de las metodologías tradicionales, generando poca documentación y no haciendo uso de métodos formales. [NGuerrero, 2015]

Todas las metodologías que se consideran ágiles cumplen con el manifiesto ágil que no es más que una serie de principios que se agrupan en 4 valores:

- **Los individuos y su interacción**, por encima de los procesos y las herramientas.

- **El software que funciona**, frente a la documentación exhaustiva.
- **La colaboración con el cliente**, por encima de la negociación contractual.
- **La respuesta al cambio**, por encima del seguimiento de un plan.

2.4.3. Principales Metodologías Ágiles

Uno de los principales focos de aplicación de las metodologías ágiles son los proyectos tecnológicos. Cada una de ellas tiene sus fortalezas y sus debilidades, pero no son excluyentes. En cada proyecto podemos adoptar una, o varias, en función de las características del propio proyecto y del equipo, Entre las metodologías ágiles más usadas se encuentran (ver Fig 2.3):

- **SCRUM:** Es un marco de trabajo que nos proporciona una serie de herramientas y roles para, de una forma iterativa, poder ver el progreso y los resultados de un proyecto.
- **KAMBAN:** Se basa en una idea muy simple. Ésta es que el trabajo en curso (Work In Progress, WIP) debería limitarse y sólo deberíamos empezar con algo nuevo cuando un bloque de trabajo anterior haya sido entregado o ha pasado a otra función posterior de la cadena.
- **XP:** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo.

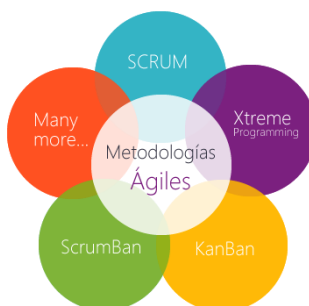


Fig 2.3: Metodologías Ágiles

Fuente: Elaborado con base en (NGuerrero, 2015).

2.5. SCRUM PARA EL DESARROLLO DE SOFTWARE

Scrum es un proceso ágil que organiza a las personas en equipos pequeños interdisciplinarios y auto-organizados, divide el trabajo en una lista de entregables pequeños y concretos llamados sprint e incrementos, asigna a cada actividad de la lista un orden de prioridad que se determina en colaboración con el cliente, en base a la revisión realizada de un entregable después de cada iteración. Consiste en un proceso iterativo no mayor a 30 días donde se realizan entregas funcionales a los clientes. Es una metodología de desarrollo simple que requiere trabajo porque no se basa en el seguimiento de un plan sino en la adaptación continua a las circunstancias que se presentan durante la evolución del proyecto. (Palacio, 2011)

2.5.1. Teoría Scrum

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control de riesgo. (Salazar, 2015). Tres pilares fundamentales soportan toda la implementación de control de procesos empírico:

- **Transparencia:** implica dar visibilidad a todo lo que está pasando, ya que los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La reunión de planificación proporciona visibilidad al equipo Scrum acerca de aquello que va a hacer el sprint.
- **Inspección:** los usuarios Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objeto, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo. La inspección tiene lugar durante: la reunión de planificación de sprint, el Scrum diario, la revisión del sprint y la retrospectiva del sprint.
- **Adaptación:** si se denomina a través de la inspección que uno o más aspectos del proceso están fuera de los límites aceptables y que el producto resultante será

inaceptable, se debe ajustar el proceso para minimizar una desviación mayor. En el ámbito de Scrum decimos que la transparencia es un valor “neutral”, ya que por sí mismo no agrega valor. Es decir, lo que genera valor es aquello que hagas con lo que descubre gracias a la transparencia.

La figura 2.4 muestra los diferentes componentes que se incluyen en la metodología Scrum:

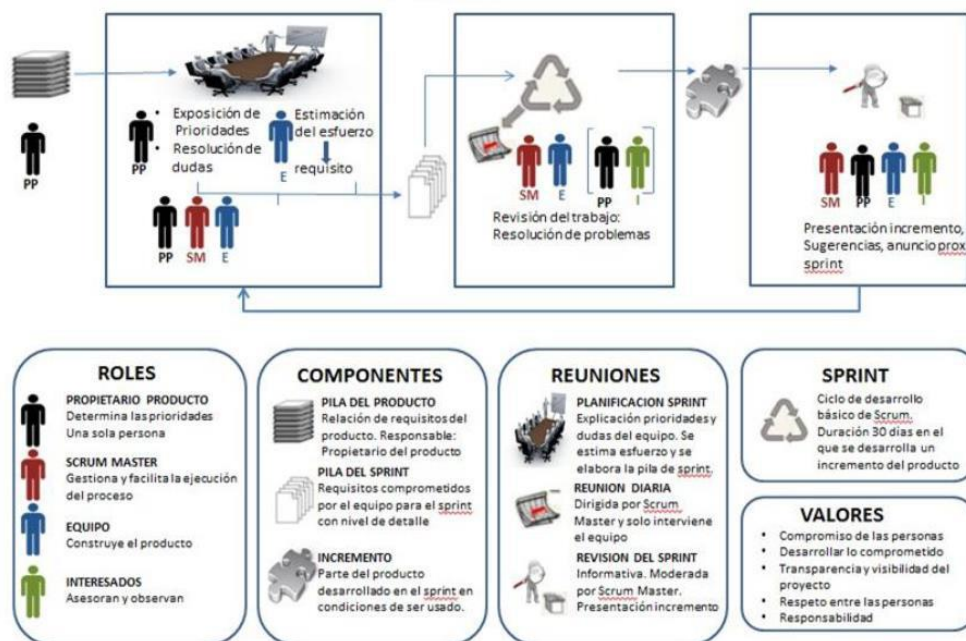


Fig 2.4: Componentes de la metodología Scrum

Fuente: Elaborado con base en (Salazar, 2015).

2.5.2. Roles y procesos de Scrum

2.5.2.1. Roles

El equipo Scrum está formado por los siguientes roles. (Salazar, 2015) (Ver figura 2.5)

- **Scrum master:** Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI.

- **Product owner (PO):** Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del ROI del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las reprioriza de forma regular.
- **Team:** Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint.

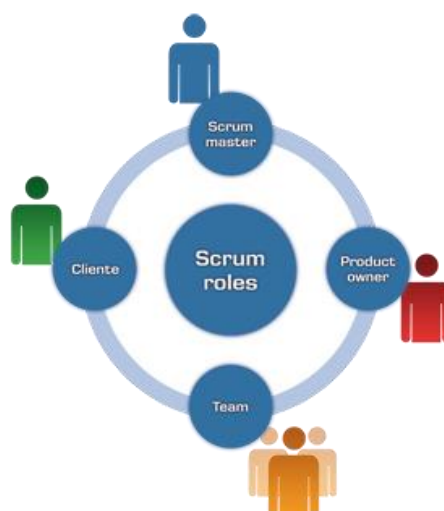


Fig 2.5 Roles de Scrum

Fuente: Elaborado con base en (Salazar, 2015).

2.5.2.2. Proceso

El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio. (ver Figura 2.6)

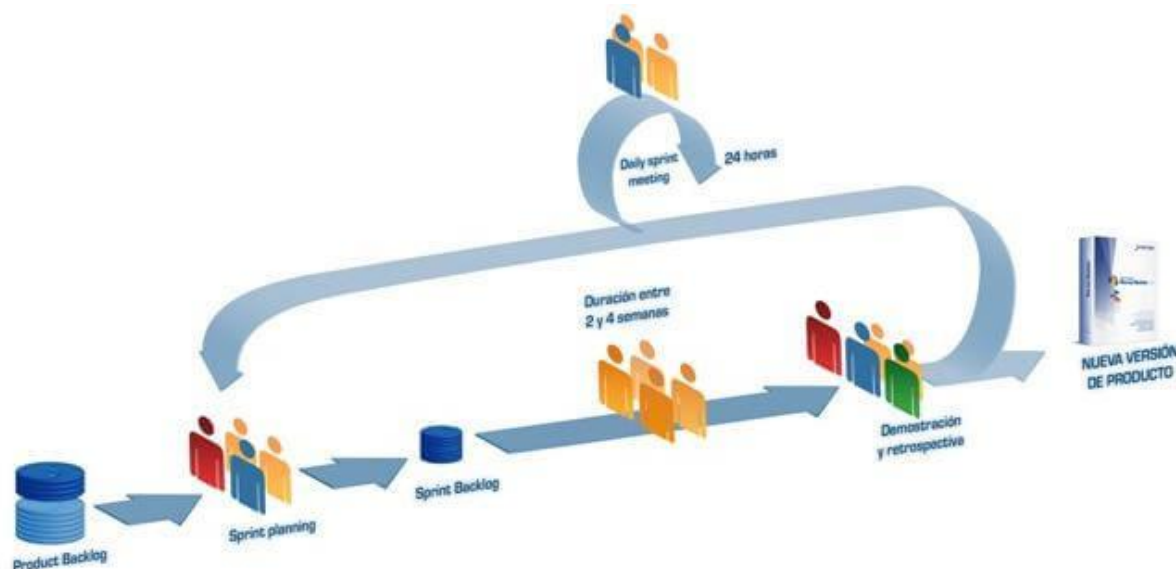


Fig 2.6: Proceso de desarrollo Scrum

Fuente: Elaborado con base en (Salazar, 2015).

- **Product Backlog:** Conjunto de requisitos demoninados historias descritos en un lenguaje no técnico y priorizados por valor de negocio, o lo que es lo mismo, por retorno de inversión considerando su beneficio y coste. Los requisitos y prioridades se revisan y ajustan durante el curso del proyecto a intervalos regulares.
- **Sprint Planning:** Reunión durante la cual el Product Owner presenta las historias del backlog por orden de prioridad. El equipo determina la cantidad de historias que puede comprometerse a completar en ese sprint, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir.
- **Sprint:** Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del Product Backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo.
- **Sprint Backlog:** Lista de las tareas necesarias para llevar a cabo las historias del sprint.
- **Daily sprint meeting:** Reunión diaria de cómo máximo 15 min. en la que el equipo se sincroniza para trabajar de forma coordinada. Cada miembro comenta que hizo el día anterior, que hará hoy y si hay impedimentos.

- **Demo y retrospectiva:** Reunión que se celebra al final del sprint y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, en la retrospectiva, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discute acerca de cómo perfeccionarlos.

2.5.3. Fases de Scrum

2.5.3.1. Pre-juego

Planificación: Definición de una nueva versión basada en la pila actual, junto con una estimación de coste y agenda. Si se trata de un nuevo sistema, esta fase abarca tanto la visión como el análisis. Si se trata de la mejora de un sistema existente comprende un análisis de alcance más limitado. Arquitectura: Diseño de la implementación de las funcionalidades de la pila. Esta fase incluye la modificación de la arquitectura y diseño generales.

2.5.3.2. Juego

Desarrollo de Sprints: Desarrollo de la funcionalidad de la nueva versión con respeto continuo a las variables de tiempo, requisitos, costo y competencia. La interacción con estas variables define el final de esta fase. El sistema va evolucionando a través de múltiples iteraciones de desarrollo o Sprints.

2.5.3.3. Post-juego

Preparación para el lanzamiento de la versión, incluyendo la documentación final y pruebas antes del lanzamiento de la versión.

2.6. LENGUAJES DE PROGRAMACION Y GESTORES DE BASES DE DATOS

El sistema de gestión trabaja de mano del CMS-IOI por lo tanto describiremos los requisitos sobre lo que se desarrolla el proyecto y las versiones necesarias para su ejecución (ver tabla 2.3).

Tabla 2.2: Lenguajes de Programación y gestores de bases de datos

Herramienta	Versión
PHP	7.2.24
PgAdmin3	1.22.2
PostgresSQL	13.0
ReactJs	16.14.0
UBUNTU	18.04.5 Lts
Python	2.7.17
OpenJDK	1.8.0_265
CMS-IOI	V1.4

Fuente: Elaboración propia.

2.7. CALIDAD DE SOFTWARE – ISO/IEC 9126

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software. Los modelos de calidad para el software se describen así:

Calidad interna y externa: Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

Calidad en uso: Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software (Ver figura 2.4). Especifica 4 características para la calidad en uso. [Borbon , 2013]

Esta norma describe 6 características:



Fig 2.7: Norma de Evaluación ISO/IEC 9126

Fuente: Elaborado con base en (Borbon, 2013).

- **Funcionabilidad:** Es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.
- **Confiabilidad:** Es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas, la confiabilidad se amplía a sostener un nivel especificado de funcionamiento y no una función requerida.
- **Usabilidad:** Es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.
- **Eficiencia:** la eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.
- **Mantenimiento:** Es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

- **Portabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reportes).

2.8. SEGURIDAD INFORMATICA

2.8.1. Mecanismos de Seguridad

La seguridad es fundamental a la hora de afrontar tareas que se realizan en sistemas informáticos ya que son las únicas medidas que pueden garantizar que éstas se realicen con una serie de garantías que se dan por sentado en el mundo físico. Por ejemplo, cuando se guardan cosas en una caja fuerte en un banco real, no se piensa que cualquier persona del mundo puede llegar a ésta de una forma inmediata como si se tratara, en lugar de un banco, de una estación de autobuses. En el mundo intangible de la informática, tan cerca de un servidor están sus usuarios legítimos como los usuarios que hacen uso de la misma red de comunicaciones. Es más, estos usuarios, en el caso de una red global, se cuentan por millones. Algunos serán “buenos vecinos” pero otros serán agentes hostiles.

2.8.2. Principios de Seguridad Informática

Para lograr sus objetivos la seguridad informática se fundamenta en tres principios, que debe cumplir todo sistema informático:

- **Confidencialidad:** Se refiere a la privacidad de los elementos de información almacenados y procesados en un sistema informático, Basándose en este principio, las herramientas de seguridad informática deben proteger el sistema de invasiones y accesos por parte de personas o programas no autorizados. Este principio es particularmente importante en sistemas distribuidos, es decir, aquellos en los que los usuarios, computadores y datos residen en localidades diferentes, pero están física y lógicamente interconectados.

- **Integridad:** Se refiere a la validez y consistencia de los elementos de información almacenados y procesados en un sistema informático. Basándose en este principio, las herramientas de seguridad informática deben asegurar que los procesos de actualización estén bien sincronizados y no se dupliquen, de forma que todos los elementos del sistema manipulen adecuadamente los mismos datos. Este principio es importante en sistemas descentralizados, es decir, aquellos en los que diferentes usuarios, computadores y procesos comparten la misma información.
- **Disponibilidad:** Se refiere a la continuidad de acceso a los elementos de información almacenados y procesados en un sistema informático. Basándose en este principio, las herramientas de seguridad informática deben reforzar la permanencia del sistema informático, en condiciones de actividad adecuadas para que los usuarios accedan a los datos con la frecuencia y dedicación que requieran, este principio es importante en sistemas informáticos cuyos compromisos con el usuario, es prestar servicio permanente.

CAPÍTULO III

MARCO APLICATIVO

Para el proyecto que se desarrolla se utilizara la metodología ágil Scrum, en el cual se realiza la siguiente actividad:

- La planificación del proyecto Sistema de Gestión de las Olimpiadas Bolivianas de Informática, inicia recolectando información de todos los requerimientos necesarios.
- Se establecerá tiempos, tecnologías, herramientas de desarrollo y la definición del número de Sprint.
- Una vez desarrollado el sistema procederemos a realizar las pruebas correspondientes para verificar calidad del sistema, con las métricas de calidad referentes a la norma ISO/IEC 9126.

En el capítulo anterior describimos la metodología Scrum para el desarrollo de software el cual está orientada a desarrollar en grupos de a dos, en el proyecto de grado se limita a utilizar todo el contenido de Scrum por el número de involucrados en el desarrollo.

3.1. PRE – GAME

En esta fase se realiza un listado de Requerimientos del sistema, para esto se aplicó la ingeniería de requerimientos, en el cual tomaremos como base algunos puntos en los cuales se trabajó y se verá a continuación.

3.1.1. Concepción

En este punto se realizó diferentes tareas para la obtención de los requerimientos del sistema, esto con el objetivo de obtener información de las actividades académicas que realiza la unidad educativa. Estas se podrán observar a continuación:

- **Observación:** se observaron los diferentes procesos que realizan los administrativos de la Olimpiada Boliviana de Informática como ser: inscripción, Participación en concursos, niveles, etapas, entre otros.
- **Entrevistas personales:** se hicieron diferentes entrevistas por plataformas digitales, con el propósito de respetar el distanciamiento y obtener los requerimientos del sistema. (ver figura 3.1)



Fig 3.1: Reuniones por diferentes plataformas digitales

- **Documentación:** fue posible acceder a copias de la documentación con la que actualmente las Olimpiadas de Versiones anteriores como ser: RUDE de los estudiantes, RDA de los profesores, convocatorias, Inscripción a niveles y etapas. Todo en material digital de Concursos anteriores. (ver figura 3.2)

DEPARTAMENTO	DISTRITO	AREA	NIVEL	GRADO	NIVEL EN LA ESCUELA	RUDE
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220023
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220023
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220011
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220025
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220025
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220031
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	82220158
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	71940013
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220013
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220023
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220025
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220022
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220017
Beni	GUAYARAMERIN	Informática	Nivel0	5P	Primaria Comunitaria Vocacional	72220023

Fig 3.2: Material digital del concurso año 2017

3.1.2. Indagación y elaboración

Primero debemos identificar los actores que interactuaran con el sistema directa e indirectamente y los mostramos a continuación con su respectivo detalle:

- **Estudiante:** Representa a todos los estudiantes de quinto de primaria a sexto de secundaria de unidades educativas fiscales, privadas y de convenio de todo el país, que cumplan con los requisitos para participar en las Olimpiadas Bolivianas de Informática.
- **Tutor:** El tutor o tutora puede ser el maestro o maestra de computación asignado a la unidad educativa o un tutor designado o avalado por el Director de la unidad educativa.
- **Administrador:** es aquel que se encarga de gestionar todo el sistema de las olimpiadas desde crear una nueva etapa de las Olimpiadas hasta deshabilitar estudiantes.

Una vez concluido las tareas para la obtención de requerimientos, se procederá a realizar el respectivo análisis de los requerimientos que se encontraron y posteriormente se realizará la clasificación de requerimientos funcionales (Product Backlog) que se muestra en la Tabla 3.1, y los requerimientos no funcionales como se puede observar en la Tabla 3.2.

Tabla 3.1: Requerimientos funcionales del sistema

ID	Requerimientos Funcionales
1	El tutor inscribe a un nuevo estudiante y el mismo deberá tener un correo electrónico para verificar la inscripción.
2	El tutor crea o modifica su cuenta personal para acceder al sistema y para el mismo deberá tener un correo electrónico valido con el propósito de verificar su autenticidad.
3	Tutor selecciona el colegio en el que fue designado para la instrucción de los estudiantes con el SIE de la unidad educativa.
4	Estudiante valida su cuenta personal mediante su correo electrónico.
5	Administrador crea una nueva versión de las Olimpiadas Bolivianas de Informática para el cual podrá agregar la convocatoria del mismo.
6	Administrador agrega los niveles que se tomara en el concurso, el cual.
7	Administrador agrega o modifica noticias referentes a las olimpiadas
8	El administrador puede agregar etapas de evaluación a las Olimpiadas.
9	El administrador inicia un concurso para que los estudiantes puedan participar.
10	El administrador puede agregar material de ayuda para los estudiantes.
11	El estudiante puede buscar material de ayuda para los diferentes niveles y etapas.
12	El tutor puede buscar un estudiante por su número de carnet o RUDE.
13	El administrador puede buscar un estudiante por el número de carnet o el RUDE para ver el rendimiento dentro de las olimpiadas.

Tabla 3.2: Requerimientos no funcionales

ID	Requerimientos no Funcionales	Descripción
1	Seguridad	El ingreso al sistema será restringido solo para usuarios autorizados, por tanto, requiere de identificación del usuario con una contraseña. El sistema usara roles para el manejo de procesos

2	Disponibilidad	El sistema estará disponible las 24 horas para su acceso
3	Usabilidad	El sistema contara con una interfaz gráfica amigable y de fácil uso.
4	Rendimiento	El sistema dará respuesta a todas las consultas que se asignara a cada nivel de acceso.
5	Multiplataforma	El sistema tendrá acceso desde cualquier plataforma que contenga acceso a internet, por cualquier explorador web.

3.1.3. Negociación

No es raro que los clientes y usuarios pidan más de lo que puede lograrse dado lo limitado de los recursos del negocio (Pressman, 2010). En este caso se aclararon ciertos requisitos con el cliente (Coordinador de las Olimpiadas), los requisitos que se trataron fueron los siguientes:

- Para el ingreso al sistema, se proporcionará al usuario (secretaria, regente, docente, estudiante y padre de familia) un nombre de usuario y una contraseña.
- Se hará uso del gestor de base de datos PostgreSQL para el almacenamiento de la información, servidor web Apache y un navegador de internet para el acceso al sistema.
- Se inscribirán a los estudiantes y se los podrá asignar un nivel en el que quieren participar.
- Se realizará reportes de los estudiantes y tutores de cada olimpiada en formato pdf.
- Se podrá habilitar o deshabilitar si un estudiante concursa a una nueva etapa dependiendo de su desempeño.
- Se creará nuevas etapas y niveles dentro de una nueva Versión de la olimpiada.

3.1.4. Especificación

Debido al uso de un diseño responsivo, el entorno de trabajo permite a los usuarios el acceso desde diferentes tipos de dispositivos con acceso a internet, como computadoras, laptop, tablets y dispositivos móviles. En la Figura 3.3 se puede observar la forma de ingreso al sistema,

seguido a esto en la Figura 3.4 se observa el procedimiento que sigue en la arquitectura propuesta para el sistema.

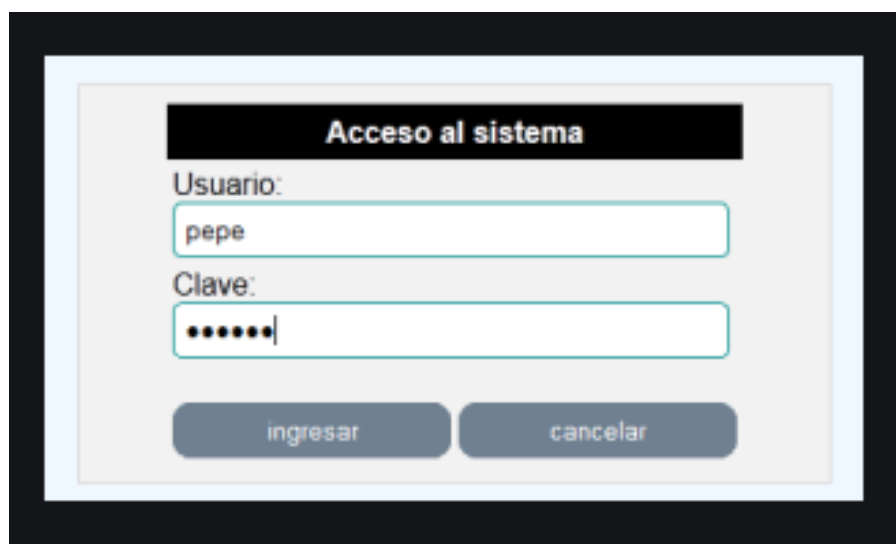
A screenshot of a web-based login form titled "Acceso al sistema". It features two input fields: "Usuario:" with the text "pepe" and "Clave:" with masked characters ".....". Below the fields are two buttons labeled "ingresar" and "cancelar". The form is set against a light gray background with a white border.

Fig 3.3: Forma de Ingreso al sistema

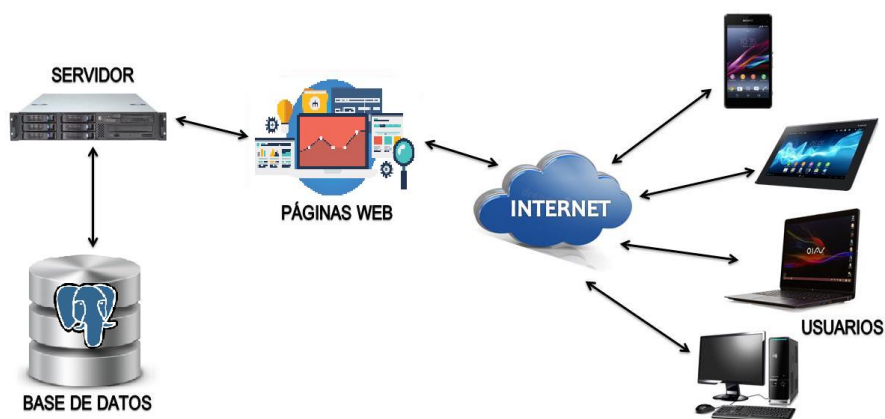


Fig 3.4: Arquitectura del sistema

A continuación, detallamos el diagrama de casos de uso obtenidos mediante los requerimientos para su mejor entendimiento lo separamos en 4, casos de uso de Admin (ver figura 3.5), Tutor (ver figura 3.6), Estudiante (ver figura 3.7), Desarrollador (ver figura 3.8).

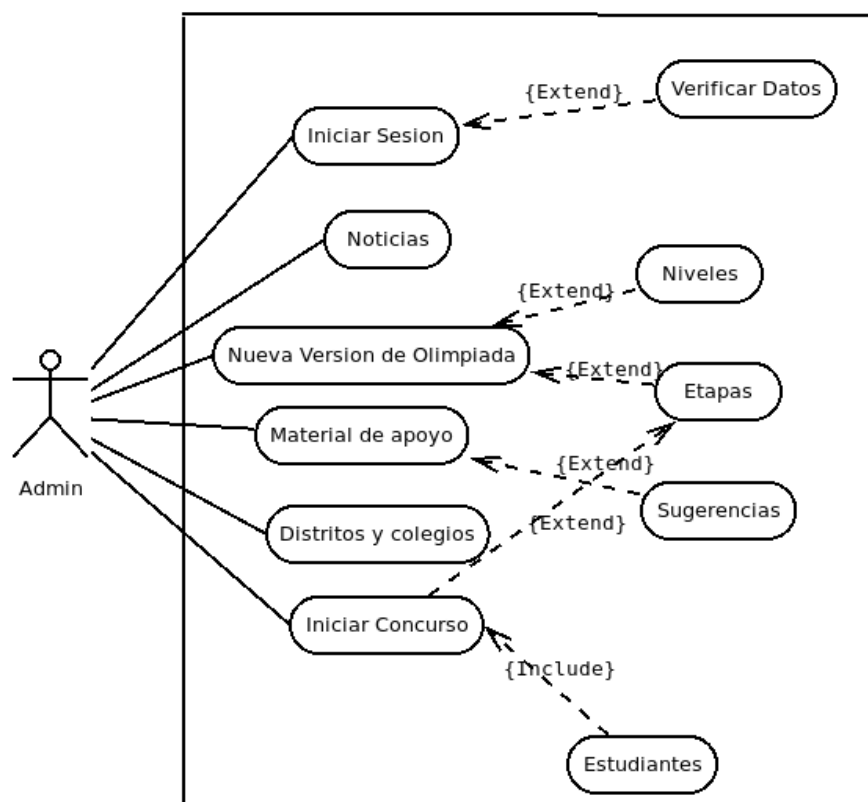


Fig 3.5: Caso de uso Admin

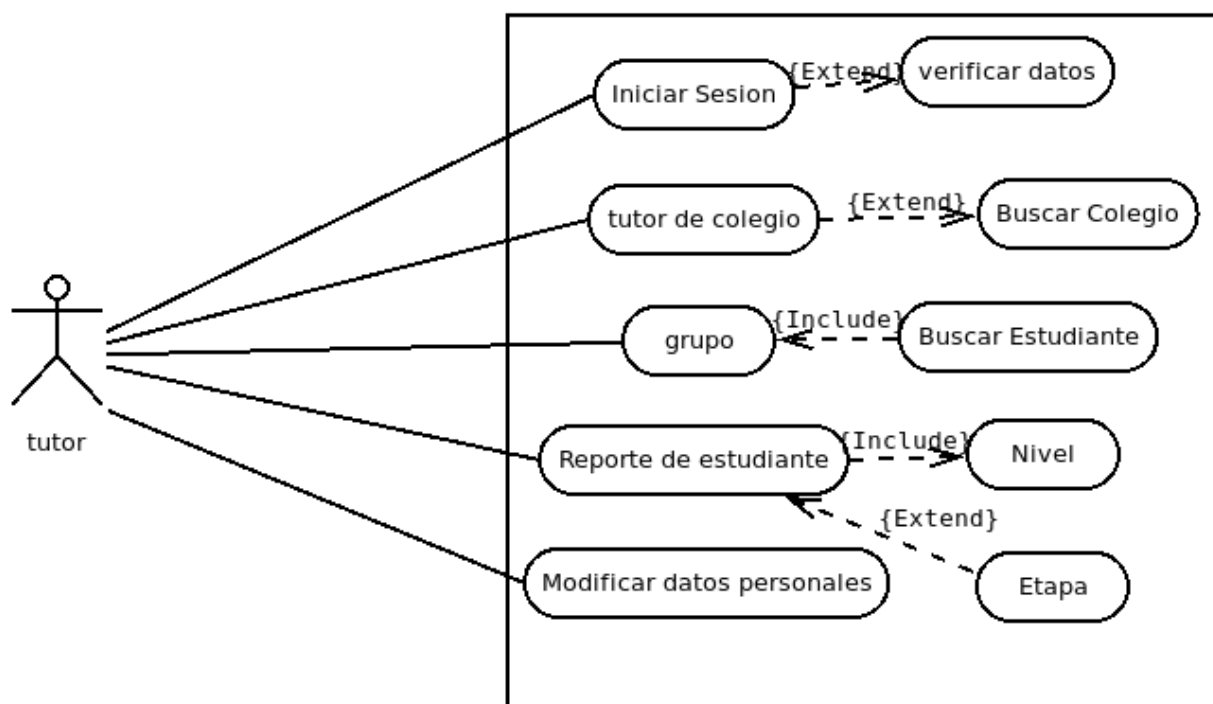


Fig 3.6: Caso de uso Tutor

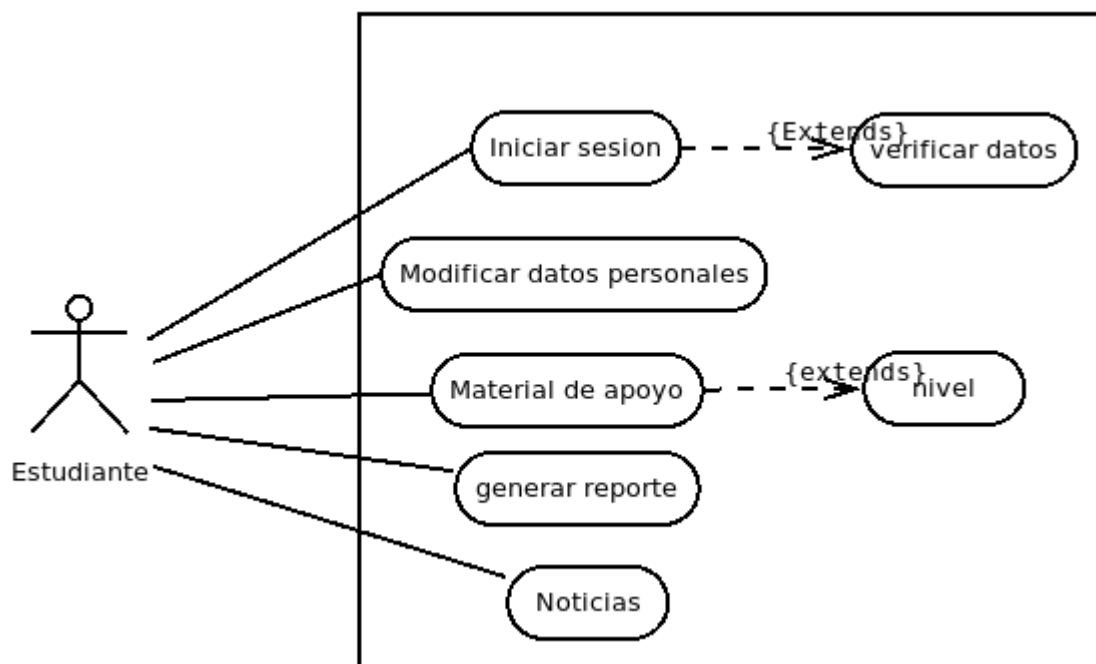


Fig 3.7: Caso de uso Estudiante

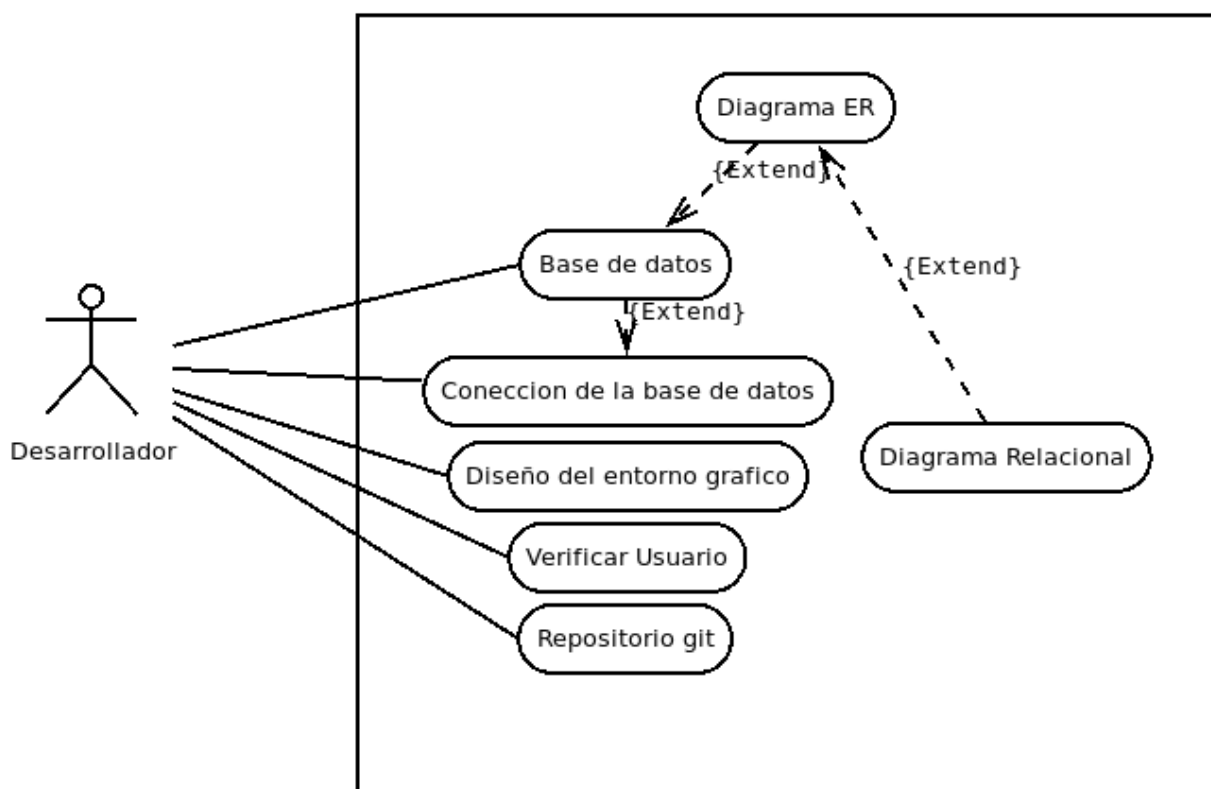


Fig 3.8: Caso de uso desarrollador

A continuación, listamos las historias de usuario con su respectiva clasificación de prioridad, estimación, y Sprint asignado. (ver tabla 3.3)

ID	Nombre	Usuarios	Prioridad	Estimación hrs	Sprint
1	Creación de la BD	Admin, Tutor, Estudiante	Alta	4	Sprint 0
2	Diseño de la plantilla	Admin, Tutor, Estudiante	Alta	8	
3	Verificar Usuario	Admin, Tutor, Estudiante	Alta	8	
4	Repositorio GIT		Alta	4	

5	Iniciar Sesión	Admin, Tutor, Estudiante	Alta	5	Sprint 1
6	Verificar datos de Inicio de sección	Admin, Tutor, Estudiante	Media	4	
7	Distritos y colegios	Amin,tutor	Media	5	
8	Material de apoyo	Admin, tutor, estudiante	Media	4	
9	Niveles	Admin, Tutor	Media	5	
10	Etapas	Admin, Tutor	Media	4	
11	Grupos	Tutor	Media	6	
12	Noticias	Admin, Tutor, Estudiantes	Media	4	Sprint 2
13	Iniciar Concurso	Admin	Alta	3	
14	Modificar datos personales	Tutor, Estudiante	Media	5	
15	Reporte	Admin, Tutor, estudiante	Baja	3	
16	Asignar Estudiante a los grupos	Tutor	Alta	6	
17	Buscar Estudiante	Admin, Tutor	Baja	4	
18	Pruebas e Integración		Alta	20	Sprint 3

3.2. GAME

3.2.1. Sprint 0

Para el caso de Uso Diseño de base de datos implementamos el diagrama entidad relación y posteriormente el diagrama relacional. (ver Figura 3.10) (ver Figura 3.11), el análisis y diseño de la Base de Datos para el Sistema Web son muy importantes para el inicio y ejecución de todos los Sprints es por ese motivo la priorización de este. En la Figura 3.9 se muestra el respectivo diagrama de caso de uso para este sprint.

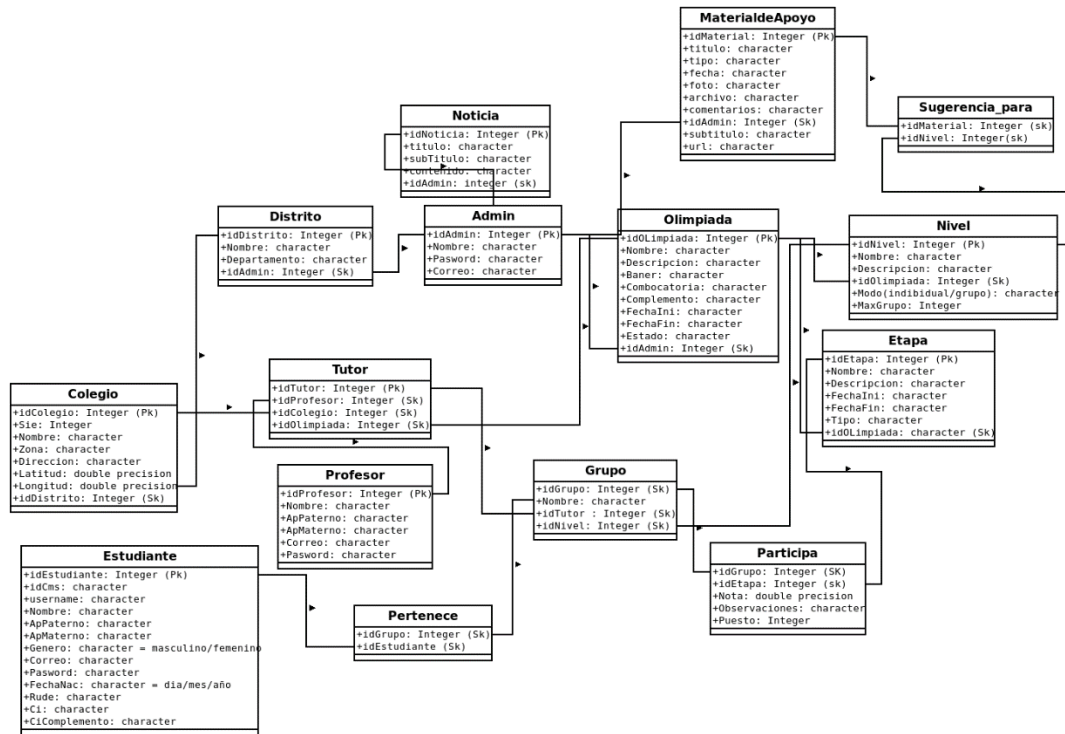


Fig 3.11: Diagrama relacional

3.2.2. Sprint 1

3.2.3. Sprint 2

3.2.4. Sprint 3

3.3. POST - GAME

3.3.1. Validación

3.4. SEGURIDAD

3.4.1. Seguridad Administrativa

3.4.2. Seguridad lógica

3.4.3. Seguridad física

3.5. CALIDAD DE SOFTWARE

3.5.1. Usabilidad

3.5.2. Eficiencia

3.6. HERRAMIENTAS EMPLEADAS

4. MARCO DE RESULTADOS

BIBLIOGRAFIA

[Ingenioempresa, 2020]

Ingenioempresa.com. 2020. *Marco Lógico: Definición, Elaboración Y Ejemplo Detallado*. [online] Available at: <<https://ingenioempresa.com/metodologia-marco-logico/>> [Accessed 25 July 2020].

[Correa y Álvarez, 2012]

Correa, Álvarez, (2012). La Gestión Educativa un nuevo paradigma, Fundación Universitaria Luis Amigo, Medellín – Colombia, [http://www. virtual.funlam.edu.co](http://www.virtual.funlam.edu.co)

[Pressman, 2002]

Pressman, (2002) Roger Pressman, “*Ingeniera de software un enfoque de practico*”, quinta Edición

[Pressman, 2010]

Pressman. (2010). *Ingeniería de software: un enfoque práctico*. Mc. Graw Hill.

[Maggiolo y mascellani, 2012]

Maggiolo, Mascellani, (2012),Introducing CMS: A Contest Management System. Olympiads in Informatics, 6, 86-99

[Maggiolo, mascellani y Wehrstedt, 2014]

Maggiolo, Mascellani, Wehrtedt (2014). CMS: A Growing Grading System. Olympiads in Informatics, 8, 123-131

[Bolivia Tech Hub Collaborative, 2020]

Bolivia Tech Hub Collaborative (2014). Convocatoria: Olimpiada Boliviana de Informática 2020

[Xavi, 2013]

Xavi (2006). Las 5 etapas de Ingeniería de Software: Consultado el 09 de octubre de 2020. Página web: <http://proyectosguerrilla.com/blog/2013/02/las-cinco-etapas-en-la-ingenieria-del-software/>

[React, 2020]

React (2020). Tecnología react js: Consultado el 17 de octubre de 2020. Página web: <http://react.org>

[Figuerola, Solís y Cabrera, 2008]

Figuerola Roberth G, Solís Camilo J, Cabrera Armando A (2008). Metodologías tradicionales vs. Metodologías Agiles, Universidad Técnica Particular de Loja, escuela de Ciencias de la Computación 9

[Maida, EG, Pacienza, 2015]

Maida, EG, Pacienza, J. Metodologías de desarrollo de (2015). Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería “Fray Rogelio Bacon”. Universidad Católica Argentina, 2015. Disponible en: <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf> [Fecha de consulta: 14/10/2020]

[Borbon, 2013]

Borbon Ardila Nuvia Ines (2013), Norma de evaluación ISO/IEC 9126.
Consultado el 25 de octubre de 2020. Página web: <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-deevaluacion-isoiec-9126.html>

[NGuerrero, 2015]

NGuerrero (2015), Metodologías Ágiles.
Consultado el 25 de octubre de 2020. Página
web: <https://www.programaenlinea.net/metodologias-agiles/>

[Littardi, 2018]

Littardi Juan Carlos (2018), Como hacer una buena toma de requerimientos y no morir en el intento.
Consultado el 28 de octubre de 2020. Página web:
<https://www.lainnovacionnecesaria.com/como-hacer-una-buena-toma-de-requerimientos-y-no-morir-en-el-intento/>

ANEXOS

Anexo I

I. Reglamentación sobre fraude

Los competidores deben utilizar únicamente la computadora y cuenta que se les asigne en la competencia, en particular:

- Los competidores no deben intentar enviar programas ilegales (programas que utilizan librerías para acceso a red, lectura de archivos que comprometen el sistema operativo o dan acceso a información de los problemas planteados y sus soluciones, código orientado a producir mal funcionamiento y/o compromiso del sistema operativo), ni intentar causar daño e interferencia o compromiso con el sistema de evaluación.
- Los competidores no deben intentar tener acceso a niveles superiores a los de competidor (Super usuario, comité de evaluación, etc.), o a cualquier otra cuenta que no se les asignó.
- Los competidores no deben intentar almacenar información en cualquier parte del sistema de archivo, salvo en el escritorio, directorio base de su cuenta, o directorio temporal (/temp)
- Los competidores no deben tocar ni acceder a otras computadoras y sus accesorios, salvo la que se les asignó.
- Los competidores no deben intentar acceder a otras máquinas/dispositivos en la red o internet, salvo lo requerido, para enviar su solución (respuesta) al sistema de evaluación.
- Los competidores no deben intentar reiniciar o alterar la secuencia de inicio y configuración del computador que se les asigna.
- Los competidores no deben comunicarse con otras personas durante la competencia, salvo personal del equipo asignado por el Comité a cargo de la competencia y/o miembros del Comité Científico.
- Los competidores no deben ni pueden ingresar dispositivos electrónicos de almacenamiento y/o transmisión de datos a la competencia, así mismo, no pueden ingresar ningún tipo de material impreso.
- Los competidores pueden ingresar papel (completamente limpio, sin escrituras), y artículos para escribir.
- El incumplimiento del presente reglamento, será considerado como fraude y resultará en la descalificación del estudiante infractor.

ARBOL DE PROBLEMAS

ARBOL DE OBJETIVOS