**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

***ESCOM***

*Trabajo Terminal*

***Repositorio de soluciones a problemas de programación competitiva***

***“Solution Book”***

*Que para cumplir con la opción de titulación curricular en la carrera de*

**“Ingeniería en Sistemas Computacionales con especialidad**

**en Sistemas”**

*Presentan*

**Pérez Feregrino Marco Atltzin**

**Rodríguez Rodríguez Luis Eduardo**

**Sánchez Ramírez Valeria**

*Director*

**Franco Martínez Edgardo Adrián**



***México D.F., a ? de Junio de 2015***

Instituto Politécnico Nacional

Escuela Superior de Cómputo



No de Registro: TT 2014-B010 Serie: Amarilla Junio 2015

Documento Técnico

**“*Repositorio de soluciones a problemas de programación competitiva*”**

Presentan:

**Pérez Feregrino Marco Atltzin1**

**Rodríguez Rodríguez Luis Eduardo2**

**Sánchez Ramírez Valeria 3**

Directores:

**M.C Franco Martínez Edgardo Adrián**

**Resumen:**

El sitio Web permitirá la publicación de problemas y la soluciones a problemas que hayan sido planteados en algún concurso en línea o presencial de programación competitiva. Las soluciones podrán publicarse adjuntando material multimedia para apoyo didáctico, permitiendo a los usuarios resolver el problema de otra forma o atacando un caso que no contemple la solución original. Finalmente se podrán calificar las soluciones de otros usuarios y así crear un sistema de rankeo.

.

**Palabras Clave** – Repositorio, Programación competitiva, Plataforma Web

1.- ferefuc@gmail.com

2.- lusirodriguez77@gmail.com

[3.-](mailto:3.-valesp_505@hotmail.com) [valesp\_505@hotmail.com](mailto:valesp_505@hotmail.com)

1. **Introducción**

Una de las actividades académicas más fuertemente relacionadas en el ámbito de programación son las competencias de programación competitivas en las que se se dan problemas a equipos de tres personas y se deben resolver un conjunto de problemas con lenguajes, comúnmente C, C++ y Java , con usa sola computadora en un tiempo de 5 horas.

Este tipo de competencias deja problemas con temas avanzados en la programación aislados de los programadores que no pertenecen al ámbito para esto reuniremos diferentes tecnologías web y técnicas de programación para una solución diferente a la de intentar difundir estos problemas hacia la comunidad de de ciencias computacionales.

El Trabajo Terminal **2014b010**  que lleva por nombre “***Repositorio de soluciones a problemas de programación competitiva***” es un sistema web que pretende implementar un repositorio de problemas de programación competitiva en los cuales se podrá adjuntar material multimedia para la explicación de los mismos así como a las soluciones a dichos problemas en los cuales también se podrá interactuar con los demás usuarios mediante comentarios.

Esta solución permite a los usuarios dentro del ámbito como a los que no , no solo compartir sus soluciones sino compartirlas con explicaciones apoyadas de material multimedia como imágenes, videos , notas de voz y diagramas incluso esto apoyará a usuarios interesados en temas especializados de programación a comprender de una forma más precisa y con soluciones ya probadas, es decir que ya funcionan para dicho problema. Éstas soluciones se probaran compilando o ejecutando el código, según sea el lenguaje de la solución, y ejecutando para comprobar la salida del programa con la salida del problema y verificar que cumpla también se obtendrán parámetros de la ejecución si el problema lo amerita para verificar que cumpla con las condiciones necesarias.. Además de que promueve la competitividad de los eventos de programación con un ranking para los usuario.

**1.1 Problemática**

Actualmente no existe un sistema que ofrezca un único lugar donde se puedan almacenar, publicar y proponer soluciones a problemas de programación competitiva orientada a la comunidad estudiantil. Los sistemas que más se asemejan a nuestro objetivo sólo permiten ver un listado simple de problemas donde no siempre se tiene una solución y en caso de que se tenga éstas no están apoyadas en material multimedia (videos, imágenes, documentos electrónicos) con lo cual se dificulta el estudio de algún problema o su temática.

Por otro lado existen diferentes sitios como foros de discusión y blogs en donde se plantean problemas y soluciones diversas donde no hay garantía de solución, tampoco estos problemas tienen una categorización para una búsqueda más fácil, por lo que obliga al usuario a navegar en más de un sitio para terminar encontrando información incompleta y poco fiable.

## 1.2 Solución Propuesta

Construcción de un sistema web que permita almacenar problemas de programación competitiva y sus soluciones con apoyo multimedia, es decir agregar una explicación de la solución, adjuntar algún video o imágenes como diagramas, documentos electrónicos y audio. También contará con un módulo de detección de duplicidad de problemas y soluciones para evitar redundancia tanto en la base de datos como en el repositorio. Se fomentará la competencia por medio de un registro de estadísticas por el cual los usuarios podrán calificar las soluciones.

## 1.3 Objetivo General

Desarrollar e implementar un sistema que almacene y proporcione acceso a problemas de programación competitiva y sus respectivas soluciones con apoyo multimedia y capacidad de vinculación a los problemas originales de algunos jueces en línea de los que provienen si es el caso.

## 1.4 Objetivos Específicos:

* Creación de soluciones didácticas por parte de los usuarios
* Proporcionar soluciones y código fuente previamente verificado y catalogado por el sistema
* ~~Ofrecer una correcta gestión de problemas competitivos.~~
* Promover la competitividad de los usuarios mediante un sistema de rankeo
* ~~Crear un foro por cada grupo de soluciones al mismo problema~~

## 1.5 Justificación

Debido a la problemática que de que no existe un sistema que reúna problemas de programación de carácter competitivo y sus posibles soluciones de forma descriptiva, organizada, rankeada y correcta para su estudio por parte de las comunidad de programación competitiva. Se propone un sistema web que permita su almacenamiento, consulta y organización de los problemas y sus soluciones mediante un repositorio que se gestionará conforme a una clasificación. También se agregara una descripción de las soluciones.

Los beneficios de nuestra propuesta son:

* Recopilación de problemas de programación competitiva
* Una fácil búsqueda de los problemas y sus soluciones
* En las soluciones se podrá adjuntar contenido multimedia como: videos, diagramas, imágenes. Lo cual refuerza la comprensión del problema o solución. Característica importante la cual no nos brindan otros sistemas.
* Links a problemas de jueces.

Esta propuesta les beneficiará a estudiantes, competidores, profesores y entrenadores que tienen relación con el área de programación competitiva para complementar y solidificar sus conocimientos.

## 1.6 Alcances

La ventaja de este sistema es que podrá ser accesible con tan sólo una computadora con internet y no será necesario estar registrado para poder ver los problemas y sus soluciones así como en otros sistemas similares que es se necesita estar registrado y subir una solución solo para ver el código fuente de las demás soluciones.

Esta solución no sólo se centra en mostrar y almacenar dichos problemas con sus soluciones sino también en que las soluciones den el resultado esperado para ese problema mediante comparación de la salida de los problemas y algunos parámetros que se podrán especificar en el problema como consumo de memoria o tiempo de ejecución, así como fomentar la competitividad mostrando estadísticas de los votos que hagan los mismos usuarios del sistema a las soluciones para poder dar un ranking al usuario que brinde sus soluciones.

# 

# **II Estado del arte**

En la actualidad existen sistemas que proveen de problemas de programación competitiva pero sin sus soluciones, hay otros que nos dan los problemas y las soluciones de los usuarios sin embargo no es posible ver las soluciones hasta que se suba una propia, otros sistemas ofrecen ranking de usuarios y soluciones pero las soluciones no vienen explicadas para su consulta de otros usuarios que no tengan el nivel de conocimientos para adquirirlo. Como ya hemos dicho existen infinidad de blogs o páginas con problemas pero con soluciones parciales y poco confiables además de que no están especializadas en problemas de nivel competitivo.

Algunos de estos sistemas son:

* Uva online judge
* COJ(Caribbean Online Judge)
* Karelotitlán
* Omegaup
* Uri online judge
* SPOJ(Sphere Online Judge)
* ACM ICPC
* RPC (Red de Programación Competitiva)

Pero como se mencionó anteriormente ninguno da una solución como la que se propone en este trabajo terminal ya que muchos se enfocan a jueces en línea o simplemente tienen otros propósitos.

## 

## 

## 

## 

## 

## 

## 

## 2.1 Marco Conceptual

## 2.1.1 Tabla Comparativa de sistemas similares

Tabla 1: Comparación de Sistemas Similares

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sistema** | **Rankeo de soluciones** | **Rankeo**  **de**  **usuarios** | **Ver soluciones** | **Subir multimedia** | **Compila**  **código** | **Lenguajes de programación** | **Funcionalidad** |
| Uva online judge [2] | No | No | Si, con previa solución del usuario aprobada | No | Si | ANSI C, Java,  C++, Pascal, | Juez en línea |
| COJ  (Caribbean Online Judge) [3] | Si | Si | Si, con previa solución del usuario aprobada | No | Si | Bash, C,  C#, C++  Java, Pascal  Perl ,PHP5, Phyton ,  Ruby | Juez en línea |
| Karelotitlán[4] | No | Si | No | No | Si | Pascal, Java | Juez en línea |
| Omegaup [5] | Si | Si | No | No | Si | C++, C, Haskell, Java, Pascal, Python, Ruby,  Karel | Juez en línea |
| Uri online judge [6] | No | Si | No | No | Si | C++, Java | Juez en línea |
| SPOJ  (Sphere Online Judge) [7] | Si | Si | No | No | Si | Assembler, Bash Basic, C, C#,C++ CommonLisp, Erlang, F#, Fortran, Haskell,Icon, Intercal, Java, JavaScript, Lua, Pascal, Perl, PHP, Python, Prolog, Ruby. | Juez en línea |
| ACM ICPC [8] | No | No | No | No | No |  | Repositorio |
| RPC  (Red de Programación Competitiva)  [9] | No | No | No | Si | No |  | Foro |
| **Solution Book** | Si | Si | Si | Si | Si | C,C++, Java, Python | Repositorio |

PONER ACRÓNIMOS EN OTRO LADO PARA SIMPLIFICAR TABLA (Luis)

**III Análisis**

**3.1 Factibilidad del Proyecto:**

El proyecto se desarrollara con tecnologías de mucho soporte y una gran comunidad además de que tenemos conocimientos sólidos. Se usarán tecnologías como un Web Socket y frameworks JavaScript que aunque no se enseñan en la escuela ya anteriormente han sido estudiados y puestos en práctica de forma particular. Al igual que todas las aplicaciones web se necesita de un alojamiento, esto y la tecnología Web Sockets que será un servicio que implementaremos requiere un costo económico que ya se ha contemplado para situaciones de implementación y prueba además de las versiones gratuitas de estos servicios para probar su funcionamiento. En conclusión el proyecto es Factible pues contamos con los recursos necesarios para desarrollarlo.

**3.1.1 Factibilidad Técnica:**

Para obtener esta factibilidad técnica, se tuvieron que contemplarlas tecnologias que implementaremos que serán como lenguaje de servidor PHP, como lenguajes de Front-End frameworks de JavaScript. Así mismo un servicio Web Socket para notificaciones con simulación en tiempo real con el usuario. Aunque se han visto algunos lenguajes en la escuela otros como algunos Frameworks como Laravel v5 para el servidor y otros Frameworks y plugins JS que nos brindan módulos y funciones comunes en el ámbito de estos sistemas que no hay que programar de nuevo y con la certeza de que funcionan esto nos ahorra tiempo de desarrollo, para ello hemos requerido tiempo de investigación y práctica que ya poseemos para desarrollarlo. Otro punto es que contamos con la experiencia del Profesor Franco Martínez Edgardo Adrián en este tema como asesorías. Por lo cual cumple con la factibilidad técnica. (no supe cómo plantear bien esta parte

**3.1.2 Factibilidad Económica:**

Uno de los inconvenientes que podrían suscitarse es la adquisición del hospedaje del sistema web y algunos servicios como el *web socket* para notificaciones sin embargo ya se habían previsto y entran dentro del presupuesto de cada integrante del equipo. En un principio se realizaron pruebas con las versiones gratuitas de estos servicios y posteriormente si cumplen con la expectativa se pagaría por el servicio para llevarlo a simulaciones de producción de forma más formal.

3.2 Análisis de Costos:

Tabla 2: Servicios de pago

|  |  |
| --- | --- |
| **Servicio** | **Costo** |
| Alojamiento del sitio(Servidor) | Rango de $ |
| Web Socket | Rango de $ |

Tabla 3: Licencias de software utilizadas

|  |  |
| --- | --- |
| **Licencia** | **Costo** |
| PHP Storm |  |
| SublimeText | :D |
|  |  |
|  |  |

Analisis y diseño

**arquitectura** descripcion de solución propuesta , aspectos tecnicos , gestor de bases de datos, internet etc con explicacion detallada.

**analisis de tecnologias** ventajas desventajas y justificación de decisión.

**3.3 Usuarios del Sistema**

**3.3.1 Administrador:**

El sistema contará con usuario administrador que se encargará de realizar operaciones básicas como registrar , borrar , editar y actualizar registros para los problemas, soluciones, noticias y usuarios. Esto debido a el registro de usuarios Problems setters que tiene que hacer.También podrá cambiar de estado suspendido a activo a un usuario.

**3.3.2 Problem Setter:**

Este tipo de usuario tendrá la capacidad de subir problemas de programación competitiva. También validará si las amonestaciones de las soluciones de los problemas que se han subido son verídicas cuando éstas entren en un lapso donde el Solver haya hecho caso omiso. Además podrá subir sus propias soluciones.

**3.3.3 Solver:**

Este usuario como su nombre lo indica su función primordial será subir sus soluciones donde podrá apoyarse de contenido multimedia. Así como también podrá votar por una solución de forma positiva o negativa y mediante estas mecánica se obtendrá un ranking con las votaciones estadísticas que obtenga de sus soluciones.

**3.3.3 Viewer:**

Aunque este usuario no figura realmente en el sistema se tiene hacer mención de él ya que sin estar registrado en el sistema podrá ver los problemas y las soluciones para consulta, pero en ningun caso podra subir soluciones o cualquier operación con la base de datos a menos que se registre en el sistema.

**3.4 Requisitos reglas de negocio**

**3.4.1 Requisitos funcionales**

**Usuario general**

RFG01.- El usuario podrá iniciar sesión en el sistema con las redes sociales Facebook y G+ o con una cuenta del propio Sistema.

RFG02.- Los usuarios podrán amonestar soluciones y/o su contenido de otros usuarios por alguna infracción según la **RN 07**

RFG03.- Los usuarios registrados podrán realizar votaciones Positivas y Negativas sobre las soluciones. **RN 11**

RFG04.- Los usuarios registrados en el sistema podrán subir soluciones a los problemas.

RFG05.- Los usuarios registrados podrán subir contenido multimedia[1] a sus problemas y/o soluciones.

RFG06.- Los usuarios podrán realizar y responder comentarios a problemas y soluciones mediante Disqus[14].

RFG07.- Los usuarios podrán hacer CRUD[9] a sus respectivas soluciones.

RFG08.- Los usuarios podrán realizar búsquedas de los problemas registrados en el sistema.

RFG09.- Los usuarios podrán suspender sus cuentas y activarlas de nuevo haciendo login.

**Problem Setter**

RFP01.- Podrá registrar problemas de programación competitiva[16].

RFP02.- Podrá alternar el tipo usuario de un usuario entre Solver o Problem Setter según la regla de negocio. **RN 08**

RFP03.- Podrá borrar la solución de otro usuario que pertenezca a su problema. **RN 09**

**Solver**

RFS01.- Podrá resolver la amonestación editando su solución según el tipo de infracción.

RFS02- Será rankeado, obteniendo una distinción numérica entre los usuarios.

**Viewer**

RFV01.- Podrá ver los problemas registrados en sistema.

RFV02.- Podrá ver las soluciones registradas en sistema.

RFV03.- Podrá ver el ranking de los usuarios y soluciones del sistema.

RFV04.- El usuario se podrá registrar en el sistema con Facebook, G+ y el propio Sistema.

**Admin**

RFA01.- Podrá hacer un CRUD de usuarios.

RFA02.- Podrá hacer un CRUD Noticias.

RFA03.- Podrá hacer CRUD de Problemas.

RFA04.- Podrá hacer CRUD de Soluciones.

RFA05.- Podrá registrar a un usuario Problem Setter. **RN 12**

**System**

RFSY01.- El sistema validará la extensión y mime type [17] del código fuente de la solución correspondiente a los lenguajes[2] y tags[3] aceptadas por el problema y el sistema.

RFSY03.- El sistema propondrá problemas semejantes durante el registro para evitar duplicidad.

RFSY04.- El sistema validará que este bien la salida obtenida de la ejecución respecto al resultado de su problema correspondiente.

RFSY05.- El sistema generará una lista de los usuarios con mejores resultados en el sistema y lo mostrará cada día.

RFSY06.- El sistema mostrará noticias relevantes sobre hechos acontecidos simulando tiempo real en el sistema. Ejemplos : eventos del ámbito, problemas con más soluciones, problemas favoritos(más número de likes).

RFSY07.- Adicionalmente, un usuario administrador tendrá la posibilidad de gestionar noticias personalizadas para todos los usuarios que tengan posibilidad de administrar.

RFSY08.- El sistema se implementará en un Servidor Linux.

RFSY09.- El sistema se desarrollara con el Framework Laravel Versión 5 (PHP)

RFSY10.- La base de datos se realizará en MySQL.

RFSY11.- La extensión permitida de los archivos de soluciones deberán ser: .c, .cpp, .java y .py.

RFSY12.- La extensión permitida de los archivos de multimedia deberán ser: .jpg, .jpeg, .png y .mp3.

RFSY13.- Se compilarán códigos fuentes de soluciones en lenguaje C, C++ y java.

RFSY14.- Se ejecutarán soluciones en código fuente .py, .class y ejecutables de C y C++.

**3.4.2 Requisitos no funcionales**

RNF01.- El sistema será mantenible implementando patrón arquitectónico *Model View Controller* y filosofía *Don’t Repeat by Yourself*.

RNF02.- El sistema deberá ser seguro implementando medidas contra *html injection*, js injection, uso de tokens en peticiones para evitar *cross-site*, *request foreign* y uso de ORM contra *sql injection* así como middlewares[12] para validar el tipo de usuario.

RNF03.- Debido a que se usará patrón arquitectónico MVC nos permitirá un grado considerable de escalabilidad debido a la separación de datos y lógica de negocio en la aplicación así como de su interfaz con el usuario.

**3.4.3 Reglas de negocio**

RN01: Habrá tres tipos de usuario principales: Problem Setter, Solver, Viewer.

RN02: Cada 3 amonestaciones a una solución de un Solver se enviará una notificación al dueño del problema.

RN 03: El tiempo límite del usuario Solver para corregir una solución amonestada será de 14 días.

RN04: El Problem Setter podrá ver el usuario que hizo el reporte, tipo de reporte, observación y fecha de expedición.

RN05: El código fuente de las soluciones a problemas deberán estar en lenguaje: Java, C, C++ o Python.

RN06: Los problemas incluirán información del formato de Problemas de Programación Competitiva de los PDF de ACM / ICPC con los siguientes campos : Título, Autor, Institución de procedencia, Límite de Tiempo, Descripción, Contenido Multimedia, Entradas de programa, Salidas de programa, ejemplos, fecha de publicación del problema, tags[2]

RN07: Se podrá amonestar a un usuario por copiar código, problema existente, contenido inadecuado, spam u otro.

RN08: Sólo el Problem Setter que haya cambiado el modo de usuario de un Solver podrá cambiarlo de nuevo el modo de usuario.

RN09: Un Problem Setter podrá borrar la solución de otro usuario si éste no atendió su amonestación [15] en el tiempo límite que estipula la RN03 y sólo para un problema propio del Problem Setter.

RN10: Un Problem setter tendrá la facultad de tomar decisiones sobre los casos de amonestaciones que se le presenten, esto incluye, borrar contenido, corroborar una falsa amonestación.

RN11: La votación incrementará o disminuirá en 1 los contadores respectivos y un usuario sólo podrá votar una vez por cada solución.

RN12: El registro de un Problem Setter sólo podrá ser llevado a cabo por un administrador.

RN13: Cuando un usuario borre un problema el usuario pierde la propiedad sobre el mismo y pasará a dominio del sistema.

RN14: Si un usuario da de baja su perfil sus soluciones se seguirán mostrando con su autoría. La visibilidad de sus datos personales quedan a decisión suya.

RN15: Un usuario al registrarse acepta las condiciones definidas en la RN 13 y 14 según sea su rol en el sistema.

**3.5 Diccionario de datos**

|  |
| --- |
| Diccionario de datos: Tabla **User**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idUser |  | Numérico (int) | Llave primaria de tabla Usuario autoincremental | si |
| username | 16 | Carácter | nombre de usuario para identificarlo en el sistema | si |
| email | 34 | Carácter | Correo electrónico del usuario | si |
| password | 32 | Carácter | Contraseña del usuario | si |
| rol | 'problem',  'solver',  'super' | Lista de opciones (ENUM) | Lista de roles según desempeñe el usuario en el sistema | si |
| ranking |  | Numérico (int) | Posición del usuario en ranking | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |
| avatar | 120 | Carácter | Enlace a la imagen de perfil | no |
| state | 'Active',  'Inactive',  'Suspended',  'Blocked' | Lista de opciones (ENUM) | Estado de la cuenta del usuario. Por default es Active | si |
| numWarnings |  | Numérico (int) | Número de amonestaciones | si |
| institucion | 50 | Carácter | Nombre de la institución a la que pertenece | no |

|  |
| --- |
| Diccionario de datos: Tabla **Problems**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idProblem |  | Numérico (int) | Llave primaria de tabla problem, autoincremental | si |
| title | 50 | Carácter | Título del problema | si |
| author | 50 | Carácter | Autor del problema | no |
| institution | 60 | Carácter | Institución del autor del problema | no |
| description | MediumText | Carácter | Descripción detallada del problema | si |
| likes |  | Numérico (int) | Número de likes que tiene el problema. Por default es 0 | si |
| numSolutions |  | Numérico (int) | Número de soluciones que tiene ese problema | si |
| limitTime |  | Time | Parámetro de evaluación para la solución | no |
| limitMemory |  | Numérico (double) | Parámetro de evaluación para la solución | no |
| numWarnings |  | Numérico (int) | Número de amonestaciones del problema. Por default es 0 | si |
| problemLink | 120 | Carácter | Enlace al problema | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |
| users\_idUser |  | Numérico (int) | Referencia del usuario al que pertenece ese problema | si |
| listaJueces\_idjueces |  | Numérico (int) | referencia a la lista , si es que pertence a un juez | no |
| tags\_idTags |  | Numérico (int) | referencia a palabras clave para ser buscado | si |

|  |
| --- |
| Diccionario de datos: Tabla **Solutions**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idSolution |  | Numérico (int) | Llave primaria de tabla solution, autoincremental | si |
| explanation |  | Carácter (LONGTEXT) | Explicación de la solución propuesta | si |
| state | 'active',  ’suspended’,  ’deleted’ | Lista de opciones (ENUM) | Estado en que se encuentra la solución, por default es Active | si |
| ranking |  | Numérico (int) | ¿?¿?¿?¿?? | si |
| solutionLink | 120 | Carácter | Enlace a solución | si |
| numWarnings |  | Numérico (int) | Número de amonestaciones de la solución. Por default es 0 | si |
| problems\_idProblem |  | Numérico (int) | Referencia del problema al que pertenece la solución | si |
| solutions\_idSolution |  | Numérico (int) | Referencia del codeSolution que tiene | si |
| users\_idUser |  | Numérico (int) | Referencia al usuario dueño de la solución | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |

|  |
| --- |
| Diccionario de datos: Tabla **codeSolutions**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idCodeSolution |  | Numérico (int) | Llave primaria de tabla codeSolution, autoincremental | si |
| path | 120 | Carácter | Directorio donde se guarda el código fuente y ejecutables | si |
| language | 'c++','java',  'python','c' | Lista de opciones (ENUM) | Lenguaje en que está programada la solución | si |
| limitTime |  | Time | Parámetro resultado de ejecución de la solución |  |
| limitMemory |  | Numérico (Double) | Parámetro resultado de ejecución de la solución |  |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |

|  |
| --- |
| Diccionario de datos: Tabla **Files**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idFiles |  | Numérico (int) | Llame primaria de tabla Files, autoincremental | si |
| name | 45 | Carácter | Nombre del archivo | si |
| path | 120 | Carácter | dirección en la que se encuentra el archivo | si |
| description |  | Carácter (TEXT) | descripción del archivo | no |
| type | ('imagenEjemplo','imagenApoyo','noteVoice','fileInput','fileOutput') | Lista de opciones (ENUM) | Tipo de archivo al que pertenece | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |
| problems\_idProblem |  | Numérico (int) | Referencia al problema al que pertenece el archivo | no |
| solutions\_idSolution |  | Numérico (int) | Referencia la solución al que pertenece el archivo | no |
| news\_idNew |  | Numérico (int) | Referencia la noticia al que pertenece el archivo | no |

|  |
| --- |
| Diccionario de datos: Tabla **Warnings**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idWarnings |  | Numérico (int) | Llave primaria de tabla Warnings, autoincremental | si |
| description |  | Carácter (TEXT) | Descripción del por qué se realiza la amonestación | si |
| reason | ‘copied Code’,  'notWorking,  'content inappropriate' | Lista de opciones (ENUM) | Razón por la cual se realiza la amonestación | si |
| estado | 'attended', 'processed', 'rejected' | Lista de opciones (ENUM) | Estado de la amonestación. Por default processed | si |
| hoursToAttend |  | Numérico (int) | Horas que tiene para ser resuelta esa amonestación | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |
| solutions\_idSolution |  | Numérico (int) | Referencia a la solución amonestada |  |
| problems\_idProblem |  | Numérico (int) | Referencia al problema amonestado |  |
| users\_idUser |  | Numérico (int) | Referencia al usuario amonestado |  |
| links\_idLink |  | Numérico (int) | Referencia a los links que pertenece, en caso de proporcionar pruebas |  |

|  |
| --- |
| Diccionario de datos: Tabla **Notifications**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idNotifications |  | Numérico (int) | Llave primaria de tabla Warnings, autoincremental | si |
| title | 45 | Carácter | Título breve de la notificación | si |
| description |  | Carácter (MEDIUMTEXT) | Descripción de la notificación | si |
| viewed |  | Numérico (tinyint) | Valor que nos indicará si fue vista la notificación, 0 si no la ha visto y 1 si ya. | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |
| users\_idUser |  | Numérico (int) | referencia al usuario que se le notifica | si |

|  |
| --- |
| Diccionario de datos: Tabla **News**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idNew |  | Numérico (int) | Llave primaria de tabla Warnings, autoincremental | si |
| title | 45 | Carácter | título de la noticia | si |
| description |  | Carácter (MEDIUMTEXT) | descripción de la noticia | si |
| finishDate |  | Date | Fecha final de la publicación de la noticia | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |
| users\_idUser |  | Numérico (int) | Referencia al usuario que la publico | si |

|  |
| --- |
| Diccionario de datos: Tabla **Tags**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idTag |  | Numérico (int) | Llave primaria de tabla Warnings, autoincremental | si |
| nombre | 20 | Carácter | nombre del tag | si |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |

|  |
| --- |
| Diccionario de datos: Tabla **judgesList**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idJueces |  | Numérico (int) | Llave primaria de tabla judgesList, autoincremental | si |
| nombre | 45 | Carácter | Nombre del juez | si |
| dirWeb | 30 | Carácter | Dirección web del juez | si |
| contacto | 45 | Carácter | Algunos campos extras de contacto | no |
| facebook | 45 | Carácter | Red social facebook del juez | no |
| twitter | 45 | Carácter | Red social twitter del juez | no |
| imagen | 120 | Carácter | Ruta de la imagen del juez | no |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |

|  |
| --- |
| Diccionario de datos: Tabla **links**  Fecha: 10 de mayo 2015  SolveBook |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tamaño** | **Tipo de dato** | **Descripción** | **Requerido** |
| idLinks |  | Numérico (int) | Llave primaria de tabla judgesList, autoincremental | si |
| link | 120 | Carácter | Dirección web |  |
| type | 'youTube','Github','Facebook','Twitter','juezOnline','amonestacion' | Lista de opciones (ENUM) | tipo de link |  |
| created\_at |  | Marca temporal | Fecha y hora para control de creación | si |
| updated\_at |  | Marca temporal | Fecha y hora para control de modificación | si |
| solutions\_idSolution |  | Numérico (int) | Referencia a la solución a la que pertenece el link | no |
| problems\_idProblem |  | Numérico (int) | Referencia al problema al que pertenece | no |

**3.6 Resultados esperados para evaluación de Trabajo Terminal 1**

* Documentación del análisis y diseño del sistema.
* Presentación de Trabajo Terminal 1.

**IX Diseño**

Esta sección mostraremos diagramas del diseño del sistema como diagramas de proceso, de estado, de secuencia y de casos de uso.

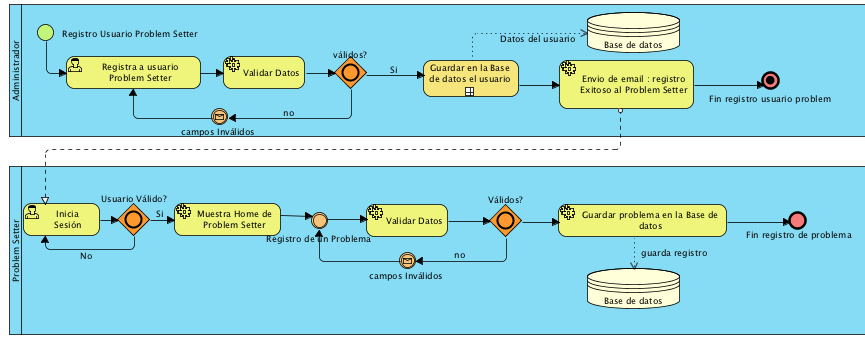
**4.1 Diagrama de proceso de negocio principal**

El diagrama de proceso de negocio se realizó en un principio para poder modelar la secuencia de los eventos intermedios entre el proceso principal que es dar de alta un usuario Problem setter que publique problemas , lo cual hará el usuario Administrador para posteriormente un usuario Solver publique soluciones a éstos problemas.

Otro motivo que nos llevó a modelar este tipo de diagrama fue que que está diseñado no sólo para la parte de planeación si no para la parte administrativa y de programación que nos ayudará a identificar estados o procesos específicos dentro de la programación y la secuencia de los eventos en el sistema haciéndolo fácil de comprender para distintas áreas y muy útil.

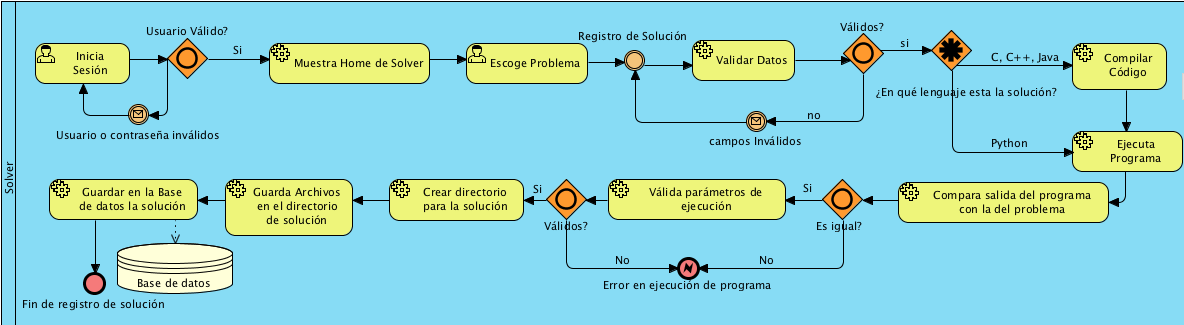
En este primera parte del diagrama observamos que el usuario administrador dará de alta en el sistema a un usuario Problem Setter que a su vez recibirá un email de corroboración de su registro en el sistema.

Después el Problem setter iniciando sesión en el sistema registrará problemas de programación competitiva.



**Ilustración 1 : Proceso Principal 1**

En esta segunda parte el Solver inicia sesión y navegara en los problemas escogiendo uno de su interés para realizar y registrar su solución. Una parte importante del proceso es el lenguaje pues de ello dependerá si se compila y ejecuta o solo se ejecuta como es el caso de python, después se comprobará si la salida del problema corresponde con la salida del programa que subió el solver y si cumple con parámetros de ejecución, en caso de que los haya, una vez esto la solución con todo y archivos multimedia se guardarán en un directorio que se crea en el proceso y se registra en la base de datos terminando con el proceso.

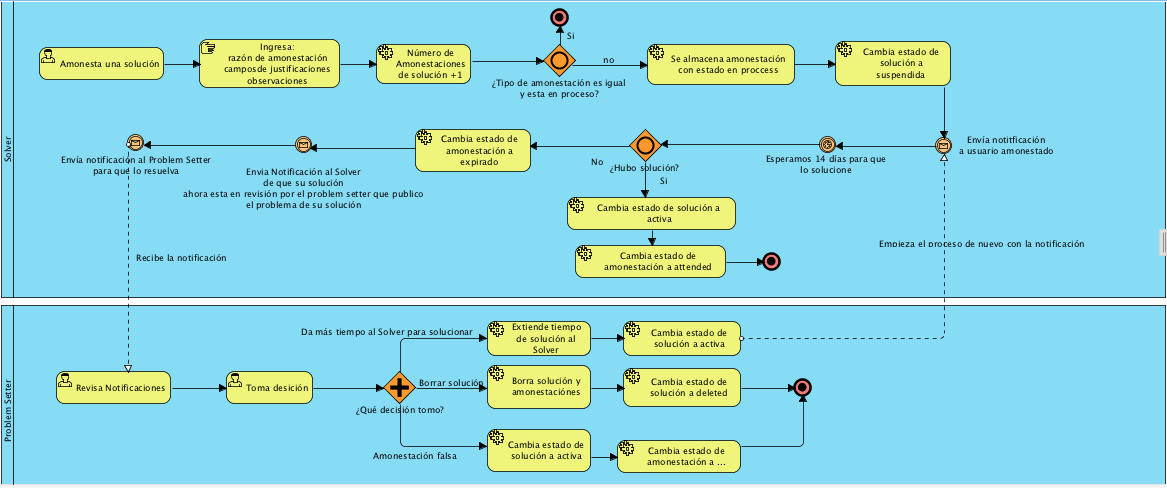


**Ilustración 2 : Proceso Principal 2**

**4.1 Diagrama de proceso de amonestaciones**

El motivo de este diagrama fue la complejidad a la hora de cambiar de estados nuestras soluciones y amonestaciones. Esta es una parte importante pues nos ayudará a controlar problemas con el contenido de las soluciones o la copia de los mismos.

Este diagrama facilita tanto al programador como para el seguimiento y mejora del proceso así como la monitorización de estados de las soluciones y las amonestaciones. Principalmente para evitar registros en la base de datos de amonestaciones repetidas y el tiempo suficiente para que el Solver corrija su solución si es necesario, también podría ser amonestación falsa y el Problem setter , el cual es dueño del problema al cual pertenece la solución, de una decisión sobre qué está sucediendo y pueda rechazarla, o darle más tiempo al Solver de atenderla.



**Conceptos**

[1] Multimedia: Imágenes formato jpg, links de videos, notas de voz.

[2] Lenguajes: C++, C, Java, Python.

[3] Tags:

[4] Solución: Código fuente, explicación en texto, material multimedia, archivo de entrada y salida. Algunos campos pueden ser opcionales según el material de apoyo o la naturaleza del problema.

[5] Material multimedia: imágenes, videos youtube y notas de voz.

[6] Problem Setter: Usuario capaz de registrar un problema y/o soluciones.

[7] Solver: Usuario capaz de registrar soluciones a problemas.

[8] Viewer: Usuario que solo podrá ver problemas y soluciones así como sus comentarios.

[9] CRUD: Operaciones con registros de la BD; Create, Read, Update y Delete.

[10] Notificación: Eventos relacionados con soluciones o problemas propios del usuario como posiciones en ranking en problema o solución, amonestaciones a las mismas, promociones del modo usuario, posiciones en ranking.

[11] Noticia: Eventos relacionados al uso del sistema y al medio de programación competitiva como problemas nuevos, competencias, resultados entre otros…

[12] Middleware: Mecanismo para filtrar petición HTTP antes de realizarla.

[13] Ranking: El ranking será el resultado del promedio de las puntuaciones a las soluciones del usuario Solver.

[14] Disqus: API que ofrece una plataforma para comentarios en aplicación web y blogs

[15] Amonestación: Advertencia o llamada de atención sobre una falta, antes de tomar una decisión negativa contra alguien.

[16] Programación competitiva: Dado un conjunto de bien conocidos problemas de las Ciencias de la Computación, resolverlos lo más rápido y eficientemente posible.

[17] MIME type: son una serie de convenciones o especificaciones dirigidas al intercambio a través de internet de todo tipo de archivos (texto, audio, vídeo, etc.)