Rojo quitar Verde agregado Anaranjado dudas

Toma de Requerimientos

**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 por nombre de los problemas registrados en el sistema.

RFG09.-

(Vale)Los usuarios podrán ordenar los resultados de una búsqueda por mayor número de votos o por menor tiempo de ejecución .

(alternativa) (Luis).- Los usuarios podrán ordenar los resultados de una búsqueda de usuarios, problemas y soluciones.

Otra alternativa (marco) Las soluciones podrán ser ordenadas ascendente o descendentemente según sus votos o sus tiempos de ejecución, si los tienen. También podrán ser filtradas según el lenguaje o la categoría :D .

hola :) esta última funcionaria en la sección de sistema creo yo

**Problem Setter**

RFP01.- Podrá subir 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] ?Falta diccionario? del código fuente de la solución correspondiente a los lenguajes[2] y categorías[3] aceptadas por el problema y el sistema.

RFSY02.- El sistema validará duplicidad de soluciones registradas en sistema.

RFSY03.- El sistema validará duplicidad de problemas registrados en sistema..

RFSY04.- El sistema validará que este bien la salida obtenidas 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 (ejemplo: Nuevo problema, trending topic, populares)

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++.

**Requisitos no funcionales**

Propuestas (by Marco)

* El sistema será mantenible implementando patrón arquitectónico *Model View Controller* y filosofía *Don’t Repeat by Yourself*.
* El sistema deberá ser seguro implementando medidas contra html injection, js injection, uso de tokens en peticiones para evitar cross-site request foreigns y uso de ORM contra sql injection así como middlewares[12] para validar el tipo de usuario.

**Reglas de negocio**

RN 01:

Habrá tres tipos de usuario principales: Problem Setter, Solver, Viewer. ?? como llamar a los usuarios en la plataforma.

RN 02:

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

RN 03:

El tiempo límite del Solver para corregir una solución amonestada será de 6 días.

RN 04:

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

RN 05:

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

RN 06:

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, categoría[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.

RN 09: 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.

RN 10: 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.

RN 11: La votación incrementará o disminuirá en 1 los contadores respectivos y un usuario sólo podrá votar una vez por cada solución.

RN 12: El registro de un Problem Setter sólo podrá ser llevado a cabo por un administrador.

**Conceptos**

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

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

[3] Categoría: ?? que tipo de categorías podríamos implementar

[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 likes 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.)