



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Jesús Yeray Ruano Folch

Tutor: Miguel Sánchez López

Co-tutor externo: Francisco Blas Izquierdo Riera

2015 – 2016

Resumen

Se ha planteado la realización de una red social para gestionar eventos de interpretación improvisada de continuidad, también conocidos como ReV (Rol en Vivo), donde los participantes introduzcan los datos requeridos por los organizadores para la realización del siguiente evento, con herramientas para gestionar a los participantes, visualizar e interpretar los datos.

Inicialmente se realizó un estudio de mercado, para saber que herramientas se requieren para dichos eventos, posteriormente se comenzó el desarrollo de una demo, donde se gestiona un único ReV, usando una red social de código libre llamada Diaspora*¹, desarrollada en Ruby on Rails (RoR), un framework que sigue el paradigma de arquitectura Modelo Vista Controlador (MVC). Sin embargo debido a la elevada curva de aprendizaje de Diaspora*, especialmente si se carece de conocimientos básicos de RoR, se optó por realizarla desde cero en RoR.

Como resultado de este proyecto se ha descubierto que es viable realizar este proyecto a mayor escala, de una manera más genérica y con la posibilidad de que distintos ReV se realicen a la vez y se comuniquen entre sí.

Palabras clave: herramienta, gestión de usuarios, red social, Ruby on Rails, modelo vista-controlador.

¹ “Diaspora*” es el nombre de un software libre para servidores web que permite implementar redes sociales.

Abstract

This project proposes the creation of a social network, purposed for the management of recurring improvisation acting events, also known as LARP (Live Action Role-playing), where the participants are enter data required by the managers in order to organize the following event, and also including helpful tools for the administration of participants, data display and interpretation.

A market study has been initially conducted in order to determine which tools are required for the events. Later the development of a demo was started, where a single LARP is managed using a free-code social network named Diaspora*², produced with Ruby on Rails (RoR), a framework that uses the architecture paradigm mode-view-controller (MVC). However, due to the problems when trying to learn Diaspora* without a basic knowledge of RoR, the option of producing the project from the very beginning has been preferred.

The conclusion obtained is that a larger-scale project is viable, can be made more generic and also include the possibility for several LARP games to be managed and communicated together at the same time.

Keywords: tool, user management, Social network, Ruby on Rails, model-view-controller.

² “Diaspora*” is name a free personal web server that implements adistributed social networking service.



Índice

| | | |
|--------|---|----|
| 1. | Introducción | 7 |
| 1.1. | Estructura de la memoria | 9 |
| 1.2. | Motivación | 10 |
| 1.3. | Objetivos | 11 |
| 1.4. | Planificación del proyecto | 12 |
| 2. | Estudio de campo | 14 |
| 3. | Planteamiento Teórico | 15 |
| 3.1. | Conectividad | 16 |
| 3.2. | Tipos de usuarios | 17 |
| 3.2.1. | Jugador | 17 |
| 3.2.2. | Organizador | 18 |
| 3.3. | Sistema de tramas | 19 |
| 3.3.1. | Módulo básico | 19 |
| 3.3.2. | Módulo de pujas | 19 |
| 3.3.3. | Módulo de puja conjunta | 19 |
| 3.3.4. | Módulo de temporizador | 19 |
| 3.3.5. | Módulo de tiradas | 20 |
| 3.3.6. | Módulo de árbol condicional | 20 |
| 3.3.7. | Módulo de múltiples resultados | 20 |
| 4. | Ruby on Rails | 21 |
| 4.1. | Arquitectura Modelo Vista Controlador | 22 |
| 4.1.1. | Modelo | 22 |
| 4.1.2. | Vista | 22 |
| 4.1.3. | Controlador | 23 |
| 4.2. | Creación de Rutas | 24 |
| 4.3. | Gemas utilizadas | 26 |
| 4.4. | Estructura de directorios | 27 |
| 5. | Implementación de la demo | 28 |
| 5.1. | Limitaciones | 29 |
| 5.2. | Problemas de desarrollo con Diaspora* | 30 |
| 5.3. | Diseño de la Base de Datos | 31 |
| 5.3.1. | Base de Datos: Red social | 31 |

| | | |
|------------|--|----|
| 5.3.2. | Base de Datos: Sistema de Tramas y Datos de partida..... | 32 |
| 5.4. | Funcionalidades básicas..... | 33 |
| 5.5. | Funcionalidades específicas | 34 |
| 5.5.1. | Ficha de jugador | 34 |
| 5.5.2. | Sistema de tramas..... | 36 |
| 5.5.2.1. | Organizador..... | 36 |
| 5.5.2.1.1. | Creación de tramas | 36 |
| 5.5.2.1.2. | Asignación de tramas..... | 38 |
| 5.5.2.2. | Jugador..... | 38 |
| 5.5.2.2.1. | Sistema de pujas de tramas | 38 |
| 6. | Conclusiones | 40 |
| 7. | Bibliografía | 41 |
| | Anexo 1 | 43 |
| 1. | Manual de usuario..... | 43 |
| 2. | Registro y login de usuario..... | 43 |
| 3. | Perfil de usuario | 45 |
| 4. | Muro de anuncios..... | 47 |
| 5. | Seguidores | 50 |
| 6. | Manual de usuario: Organizador | 51 |
| 7. | Control de usuarios | 52 |
| 8. | Creación de tramas y asignación de tramas..... | 52 |
| 9. | Manual de usuario: jugador | 54 |

Intermission System: gestión de usuarios para la simulación de líneas
argumentales en eventos de interpretación libre



1. Introducción

“Random chance plays a huge part in everybody's life”

Gary Gygax³

Para poder entender bien este proyecto, tenemos que comprender que es un juego de rol. Un juego de rol es un juego interpretativo-narrativo en el que: por un lado, gran parte de los jugadores asumen el rol de un personaje ficticio a lo largo de una historia o trama, en la que interpretan sus diálogos y describen sus acciones; y por otro lado, un jugador asume la responsabilidad de ser el director de juego. Dicho director de juego expone la situación en la que están el resto de jugadores. Pese a que se suele acordar un trasfondo para los diferentes personajes, durante la partida, estos no siguen un guión escrito previamente, pues el desarrollo de la historia es afectado por las decisiones de los jugadores, las cuales son interpretadas por el director de juego, que es el encargado de decidir qué consecuencias tienen las acciones de los jugadores, así pues en una partida de rol, la historia se desarrolla con el transcurso de la partida en base a las decisiones del director de juego.

Actualmente, los juegos de rol tienen un sistema de juego, el cual sirve para dar al director un conjunto de directrices para gestionar la partida y el entorno figurado en que esta transcurre, así como reglas que tendrán que seguir los jugadores para evitar conflictos durante la partida. Normalmente esta normativa se basa en reflejar características y habilidades del personaje ficticio de forma numérica, de tal manera que se puedan usar dichos números para saber si el personaje es capaz de realizar una acción o no, normalmente se usa la probabilidad, de modo que cuanto mejores sean ciertas cualidades del personaje, más posibilidades tiene de poder realizar bien una acción relacionada con dicha cualidad. Por ello, se puede concluir que, un juego de rol es una narración improvisada en la que los jugadores interpretan a los protagonistas de la narración, mientras que el director de juego controla la narración mediante el sistema de juego, realizando a la vez los roles de personajes secundarios, la descripción de los entornos para los jugadores, etc.

A diferencia de los juegos habituales, el objetivo no es ganar el juego, si no interpretar al personaje ficticio con la intención de colaborar con el resto de jugadores para resolver un conflicto propuesto por el director de juego.

Un evento de interpretación improvisada, más conocido como Rol en Vivo (ReV), es una modalidad de juego de rol, en el que los jugadores interpretan a un personaje en un teatro improvisado en tiempo real, donde habitualmente se usa atrezzo, y se visten como el personaje que se interpreta. Dicha actividad es controlada por los directores de juego, quienes, al igual que en los juegos de rol convencionales, son los encargados de introducir a los participantes en la situación ficticia en la que se encuentran, analizar lo transcurrido durante el evento para ver cómo afecta a la trama de la historia, y garantizar que se siguen las reglas establecidas para el evento. Dichas reglas fomentan resolver de forma aleatoria las acciones imposibles de realizar por los jugadores, o bien porque no sean realizables debido a las leyes de la

³ Uno de los principales creadores de Dungeons & Dragons



Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

física, o bien porque sean peligrosas para el jugador; mientras que el resto de acciones se realizan con el atrezo e interpretación de los participantes.

Los ReV, generalmente consisten en un único evento, tras el cual se da una conclusión, como ocurre en las obras de teatro, sin embargo, los ReV de continuidad, son eventos que se interrelacionan con aquellos realizados anteriormente en la misma ambientación. En tal caso, es imprescindible saber de antemano las acciones de los PJ (Personajes Jugadores⁴) en los eventos anteriores, para ver cómo afecta a la trama del próximo evento. En este aspecto, son similares a algunas de las series de televisión donde lo que pasó en capítulos anteriores afecta al siguiente capítulo. Si a esto añadimos que lo normal es que el número de participantes ronde entre los 30 y 60, y que por cada 5 se recomienda un organizador. Esto produce una gran cantidad de datos, lo cual hace que hacerlos de continuidad sea una tarea pesada y lenta, pues pueden trascurrir meses entre un evento y otro, hasta que los organizadores sean capaces de gestionar y analizar toda la información recolectada durante el evento anterior.

Ya que los ReV son eventos muy sociales, consideramos que la mejor manera de solucionar los problemas aparejados a hacerla de continuidad, es desarrollar una red social, donde los usuarios se registren como jugadores, pudiendo proporcionar los datos del anterior evento, e interactuar con otros jugadores entre las sesiones de un evento y otro. Por otro lado, los organizadores tendrán un papel de administradores de la red social, donde tendrán las herramientas que requieran para acelerar la recolección e interpretación de los datos, para así poder montar con más eficiencia el siguiente evento.

Dicha red social no solamente es útil para gestionar ReV, si no también es útil para solucionar uno de los principales problemas de las partidas de rol clásicas: saber que ha ocurrido entre una sesión de juego y la anterior con los personajes, ya que es complicado que durante la sesión de juego, los personajes tengan conversaciones trascendentales con las cuales puedan unir lazos. Esto es debido a que se suele disponer de un tiempo limitado, y muchas veces se fomenta que se mueva la trama de la historia por encima de la trama de los personajes.

Finalmente, dicha red social puede ser útil para fomentar las relaciones interpersonales entre las personas de intereses similares que participen en la misma.

⁴ En contraposición a PNJ o Personaje No jugador es decir, aquel que no es interpretado por un jugador del ReV.

1.1. Estructura de la memoria

La memoria de este proyecto consta de los siguientes apartados:

- Un Primer capítulo de introducción, donde se incluye: una introducción que nos explica conceptos necesarios para comprender el por qué es necesario el planteamiento de una red social para la gestión de ReV, seguido por la estructura de este documento, y posteriormente, por las motivaciones que nos han llevado a plantear este proyecto. Después explicamos cual es el objetivo que deseamos alcanzar y posteriormente explicamos cual ha sido la planificación del trabajo realizado.
- Un segundo capítulo donde se hace un planteamiento teórico en profundidad de cómo debería ser la red social para la gestión de ReV, que se tendrá en cuenta para el desarrollo de la demo.
- Un tercer capítulo donde se explican conceptos básicos de RoR, y se explican las librerías auxiliares utilizadas para el desarrollo de la red social.
- Un cuarto capítulo donde se explica el desarrollo de la demo en profundidad y se indican los problemas surgidos durante el desarrollo.
- Un quinto capítulo donde se presentan las conclusiones del proyecto.
- Un Sexto capítulo donde se expone cuál será el futuro de la red social y cuáles serán los siguientes pasos en su desarrollo.
- Por último se provee una bibliografía donde podrán consultar documentos de importancia para el proyecto.
- Se incluye un anexo compuesto de los manuales de usuario para los usuarios organizadores y otro para los usuarios jugadores.

1.2. Motivación

Llevo muchos años como aficionado a los juegos de rol, comencé como la gran mayoría, jugando en mesa con los amigos, pero por alguna razón el destino quiso que acabase siendo el principal administrador de uno de los foros de rol más importantes de Valencia, en el cual se ayudaba a la gente a encontrar una partida donde jugar. El foro acabó cerrando, pero yo continúe conociendo aficionados y participé en ReV, y a lo largo de los años incluso ayudé a montar algunos.

Al entrar en la universidad, me uní a la junta directiva del Aula de Rol y Simulación (ARS) de la Universidad Politécnica de Valencia (UPV), de la cual fui presidente los dos últimos años, igualmente en mi escaso tiempo libre jugaba alguna partida de rol y participaba en algún que otro ReV.

Con toda la experiencia que tengo en el sector, y con gran parte de esta, adquirida organizando eventos, vi la posibilidad de realizar una herramienta que facilitara el trabajo a los directores de juego, y un posible negocio en el futuro, pues no existe actualmente nada parecido en este mercado en auge en los últimos años.

1.3. Objetivos

El objetivo que se desea alcanzar con este proyecto es el diseño de una red social, con herramientas de gestión e interpretación de los datos recogidos, y la presentación de una demo técnica, en que la sólo se gestionará un único ReV, con el objetivo de simplificar el problema, y poder centrarnos en las herramientas para la gestión e interpretación de los datos recolectados de los usuarios de la red social.

Para ello se ha utilizado el lenguaje Ruby on Rails (RoR), un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, el cual sigue un paradigma de arquitectura Modelo Vista Controlador (MVC) y un conjunto de RubyGems, bibliotecas específicas para el desarrollo en RoR.

Debido a que todas las herramientas utilizadas son de código abierto, el proyecto se ha podido desarrollar en un portátil personal, sin tener que recurrir ni a licencias ni a recursos de la UPV.

1.4. Planificación del proyecto

La planificación de un proyecto es una de las partes más importantes del mismo, y en ella se detalla cuáles son las tareas que se van a realizar y cuáles son los tiempos y costes estimados para las mismas. Realizar una planificación del proyecto permite poder controlar la evolución y ejecución del proyecto y poder determinar las posibles desviaciones existentes. En base a la planificación, podremos tomar medidas correctoras (tanto en cuestión de tiempo absolutamente necesario como en cuestiones económicas) que nos eviten sorpresas el último día.

La planificación inicial del proyecto contiene 5 tareas básicas que podemos contemplar en la siguiente tabla. Su duración estimada se da en días, teniendo en cuenta que no dedicamos el tiempo completamente al proyecto, pero transcurrido ese tiempo se estima que dichas tareas estarán realizadas:

| | Tarea | Duración |
|----------------|--|----------------|
| Tarea 1 | Estudio de campo para diseñar las funcionalidades especiales de la red social | 20 días |
| Tarea 2 | Instalación de Diaspora* | 5 días |
| Tarea 3 | Aprendizaje de la plataforma Diaspora* | 15 días |
| | Documentación sobre Ruby | 1 día |
| | Documentación sobre RoR | 10 días |
| | Examinación del código de Diaspora* | 4 días |
| Tarea 4 | Añadir funcionalidades especiales de la red social para la demo | 20 días |
| | Login como admin | 2 días |
| | Pantalla de ficha de PJ | 2 días |
| | Pantalla de creación de tramas | 2 días |
| | Pantalla de interacción de tramas | 4 días |
| | Pantalla de introducción de datos de anteriores eventos | 5 días |
| | Pantalla de consulta de datos | 5 días |
| Tarea 5 | Redacción de la Memoria | 25 días |
| | Búsqueda de documentación | 5 días |
| | Realización de tablas | 3 días |
| | Redacción del proyecto | 10 días |
| | Revisión y corrección | 7 días |

En esta planificación del proyecto no se ha tenido en cuenta ningún tipo de coste económico por los siguientes motivos:

- Se va a utilizar un portátil personal, por lo que no es necesario contemplar el coste, ya que se usa de forma gratuita, (en otras condiciones se tendría que tener en cuenta el coste energético de la utilización del equipo).
- El Software utilizado es software libre, por lo que no conlleva gasto por su utilización,
- El framework RoR es de código abierto, así pues no supone ningún gasto de licencia.
- Las horas invertidas en el proyecto se deberían tener en cuenta en caso de tratarse de un trabajo para una empresa o institución pública.

Aun así, la realidad de la ejecución del proyecto ha variado por varias razones, la primera debido a que la curva de aprendizaje de RoR es elevada, al principio dificultó hasta tal punto la comprensión de Diaspora*, que al final se optó por desarrollar la red social desde el principio, por lo que se utilizó al final esta nueva planificación:

| Tarea | | Duración |
|----------------|--|----------------|
| Tarea 1 | Estudio de campo para diseñar las funcionalidades especiales de la red social | 20 días |
| Tarea 2 | Instalación de herramientas para trabajar en RoR | 5 días |
| Tarea 3 | Aprendizaje de RoR | 15 días |
| | Documentación sobre Ruby | 1 día |
| | Documentación sobre RoR | 10 días |
| | Tutoriales RoR | 4 días |
| Tarea 4 | Realizar una red social básica | 10 días |
| | Buscar documentación de las RubyGems necesarias | 1 día |
| | Crear perfil de usuario | 2 días |
| | Crear eventos | 1 día |
| | Crear posts y comentarios | 1 día |
| | Crear followers | 1 día |
| | Crear likes | 1 día |
| | Corrección de errores | 3 días |
| Tarea 5 | Añadir funcionalidades especiales de la red social para la demo | 20 días |
| | Login como admin | 2 días |
| | Pantalla de ficha de PJ | 2 días |
| | Pantalla de creación de tramas | 2 días |
| | Pantalla de interacción de tramas | 4 días |
| | Pantalla de introducción de datos de anteriores eventos | 5 días |
| | Pantalla de consulta de datos | 5 días |
| Tarea 6 | Redacción de la Memoria | 25 días |
| | Búsqueda de documentación | 5 días |
| | Realización de tablas | 3 días |
| | Redacción del proyecto | 10 días |
| | Revisión y corrección | 7 días |

Como podemos apreciar la tarea 3 es prácticamente igual que la del proyecto anterior. Únicamente cambia que se tuvo que indagar más en el funcionamiento de RoR, y se ha agregado una tarea extra, la de crear una red social nueva, lo cual sumado con el tiempo invertido en Diaspora* nos ha supuesto un retraso.

Por otro lado, se tuvo que empezar a redactar el proyecto antes de poder acabar la demo, por lo que la realización de las funcionalidades especiales se ha ido solapando con la redacción. Otra cosa a tener en cuenta es que se ha ido buscando durante todo el proyecto documentación e información requerida en esos momentos, por lo que se ha hecho de forma simultánea con otras tareas.

A pesar del retraso inesperado, dicho cambio de planificación ha sido lo mejor para el proyecto, pues ha ayudado a los examinandos a comprender el funcionamiento de cualquier proyecto desarrollado en RoR.



2. Estudio de campo

Se realizó un estudio de campo con distintos usuarios potenciales de la aplicación, a quienes se les pidió que redactasen una serie de informes.

Para el proceso de selección de usuarios potenciales se tuvieron en cuenta los siguientes factores:

- 1) Que fueran organizadores veteranos de ReV.
- 2) Que tuvieran diferentes opiniones respecto a la utilización de normativas inversivas.
- 3) Que realizasen informes de temáticas distintas.
- 4) Que los hicieran en solitario.

En dichos informes, debían exponer cuáles eran las necesidades y dificultades que tenían habitualmente para organizar un ReV de continuidad con sus temáticas favoritas. Para ello se les hicieron preguntas abiertas para que analizaran diferentes fichas de rol de sus temáticas favoritas, y explicaran con los datos que les proporciona la ficha como resolverían en un ReV cualquier situación conflictiva.

De estos informes se llegó a las siguientes conclusiones:

- 1) Existían dificultades por la gran cantidad de datos a tratar.
- 2) Habían problemas para gestionar las acciones de los jugadores entre los eventos.
- 3) Habían muchas diferencias en los informes a la hora de como comprobar si un jugador logra resolver una trama o no, debido a múltiples factores:
 - Diferencias creativas.
 - Sistemas de juego distintos.
 - Diferentes requerimientos de ambientación.

Hubo inconvenientes debido a que se produjeron retrasos a la hora de entregar los informes, e incluso se tuvieron que realizar dos entrevistas personales para acelerar el proceso.

Esto nos llevó a la conclusión de que, a pesar que cada sistema de juego requería almacenar datos distintos, muchos de ellos requerían las mismas herramientas para la gestión de ReV de continuidad, las cuales explicaremos con más detenimiento en el siguiente apartado en el punto o Sistema de tramas.

3. Planteamiento Teórico

Pretendemos diseñar una plataforma online con el propósito de servir como herramienta de ayuda a los directores de juego. Ofreciendo la posibilidad, tanto a los usuarios de rol de mesa como a los de ReV, de realizar acciones entre las partidas y comunicarse entre jugadores y los organizadores.

Este diseño está planteado para cubrir las necesidades de un ReV de continuidad con una media de entre treinta y sesenta participantes, e incluso permite gestionar partidas que estén interrelacionadas entre sí, es decir, dos o más ReV de la misma ambientación, y por ende, pueden afectar los acontecimientos ocurridos de una, a la otra. A esto lo llamamos red de ReV, porque en el fondo no dejan de ser ReV que están conectados con otros.

Otra de las peculiaridades es que las herramientas, como las bases de datos del sistema de juego, son modulares, pues distintas redes de ReV requerirán distintas herramientas, y como hemos visto en las conclusiones del Estudio de campo, la mayoría de sistemas requieren las mismas herramientas las cuales no son incompatibles entre sí. Aun así, como requisito mínimo los ReV de una misma red deben de utilizar un mismo sistema de juego y las mismas herramientas. Adicionalmente los usuarios deberían poder diseñar sus propias bases de datos de sistema para poder cubrir sus propias necesidades, ya que como se mencionó en el estudio, la principal diferencia entre sistemas son los datos a almacenar en la base de datos.

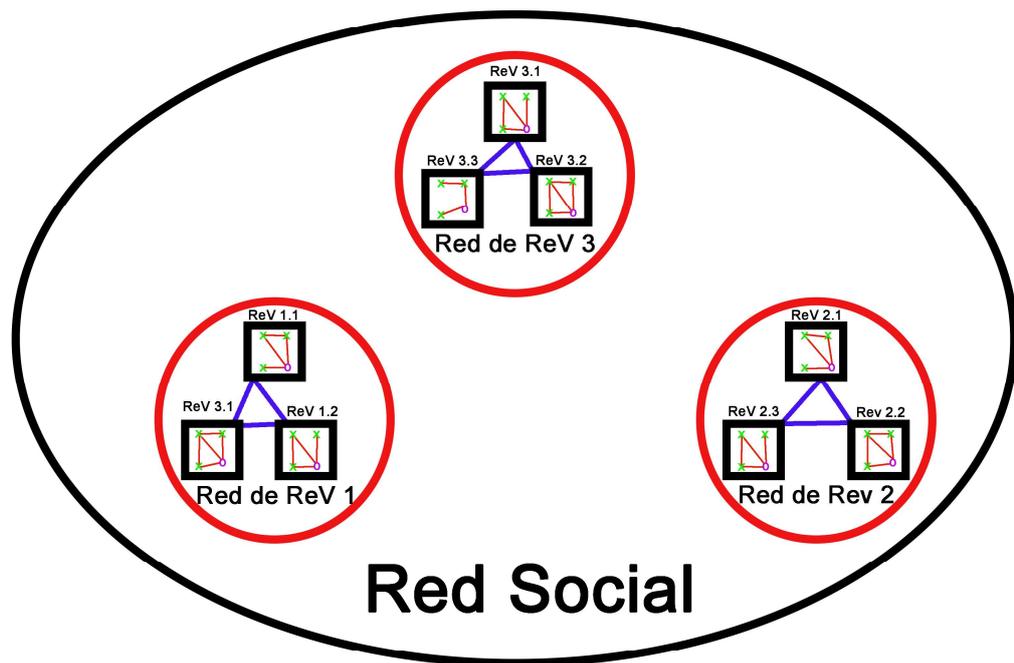
A continuación veremos tres secciones, en la primera explicaremos teóricamente la conectividad entre los ReV, en la segunda sección veremos los usuarios y las acciones que pueden realizar y por último, veremos lo más importante, el sistema de tramas y las distintas herramientas que lo componen.

3.1. Conectividad

Veamos con más detenimiento como debería estar gestionada internamente esta red social con ayuda de la figura de la derecha

Como podemos ver, dentro de la red social, pueden existir varias redes de ReV las cuales no se relacionan entre sí. Dentro de una red de ReV existen varios ReV los cuales sí pueden comunicarse entre ellos mismos, ya que las otras ReV pueden afectar al resto, dentro de la ReV hay Personajes Jugadores (PJ) y organizadores; los PJ se relacionan entre ellos, pero no tienen por qué relacionarse con todos, en cambio, los organizadores sí que tienen relación con todos los PJ de su ReV. Los PJ de una misma red de ReV se pueden comunicar entre sí, si los organizadores lo consideran oportuno. En cualquier caso los jugadores siempre pueden comunicarse entre sí siempre que lo hagan fuera de la red de ReV.

Viendo este esquema se podría pensar que cada red de ReV en realidad sería una red social en particular para ese grupo de ReV, pero la idea es que un usuario de la aplicación pueda crear una red de ReV y dentro de ella crearse los distintos ReV, al igual que el usuario podría pedir acceder como jugador a varios ReV en redes diferentes y obtener acceso por que uno de los organizadores de los ReV se lo conceda.



| Leyenda |
|-----------------------------|
| - Conexiones entre ReV |
| - Conexiones entre usuarios |
| X Personaje Jugador |
| O Organizador |

3.2. Tipos de usuarios

En esta red social hay dos perfiles bien diferenciados dentro de un ReV, el de organizador y el de jugador, tal como se plantea, un usuario de la red social podría tener ambos perfiles, para simplificarlo, un usuario nunca tendrá ambos perfiles en una misma red de ReV.

3.2.1. Jugador

El jugador es el perfil básico de la red, su única funcionalidad es la de interactuar con los otros usuarios, además de gestionar sus recursos y acciones para resolver tramas. Veamos cuales son todas las funcionalidades que puede realizar un jugador:

- **Selección de red de ReV:** En esta pantalla el jugador podrá seleccionar en que ReV quiere entrar para interactuar, pues puede tener personajes distintos en diferentes ReV.
- **Visualización de la ficha:** Sección donde se verán los datos del personaje del ReV representado por el jugador.
- **Visualización de noticias:** Es una funcionalidad similar al muro de Facebook, en la que se mostrarán las cosas de interés y las publicaciones de la gente que seguimos. Permitirá, además, la creación de publicaciones.
- **Visualización del mapa de la partida:** En esta pantalla se podrán observar los datos geográficos de la partida.
- **Tramas:** Pantalla donde el jugador podrá invertir los recursos y movimientos de su personaje en acciones entre los eventos.
- **Historial de tramas:** visualización de la resolución de las tramas e información de las mismas.
- **Mensajes:** Al igual que en muchas redes sociales, los jugadores podrán enviar mensajes entre sí.
- **Dramatis Personae:** Lista de jugadores del ReV con los datos públicos de sus personajes.
- **Historia de la ciudad:** Breve resumen de lo pasado hasta ahora en la partida.

3.2.2. Organizador

El organizador es el administrador de la red social, por lo que él se encargará de gestionar y controlar los datos de los usuarios además de revisar y gestionar las tramas de estos. Veamos con detenimiento las funciones de los organizadores:

- **Gestión de jugadores:** visualización, modificación, aceptación y eliminación de jugadores.
- **Gestión de tramas:** creación, visualización, modificación y asignación de tramas.
- **Gestión de fichas:** visualización, modificación y eliminación de fichas de jugador.
- **Gestionar noticias:** creación de noticias, modificación y eliminación.
- **Gestión de mensajería:** visualización de los mensajes enviados por los usuarios.

3.3. Sistema de tramas

Una trama es según el Diccionario de la Lengua Española de Real Academia en su cuarta definición la “Disposición interna, contextura, ligazón entre las partes de un asunto u otra cosa, y en especial el enredo de una obra dramática o novelesca”.

La función del sistema de trama es que los organizadores puedan plantear a los jugadores distintas tramas, y los jugadores indicar la reacción de su personaje a dichas tramas e incluso proponer tramas que a sus personajes les podría interesar.

A continuación veremos distintos módulos para plantear y resolver una trama, todos estos módulos estarían disponibles en la plataforma, y sería decisión de los organizadores decidir que recursos utilizan o no para la resolución de la trama.

3.3.1. Módulo básico

Toda trama consta de un nombre y una descripción donde los organizadores les darán todos los detalles posibles a los jugadores sobre la trama en particular, y escribirán las respuestas que dará la trama tanto en caso de no superarla como en caso de completarla.

3.3.2. Módulo de pujas

Este módulo dará la opción de que se resuelvan las tramas mediante pujas de los jugadores, para ello este módulo añadiría al perfil del jugador distintos tipos de recursos, los cuales se utilizarían para pujar. El organizador, a la hora de crear la trama, tendría que indicar que recursos son válidos para pujar y el valor que tiene ese recurso para la puja, e indicar si quiere que sea una puja clásica, es decir el pujador que más invierte resuelve la trama o si se da un valor fijo, lo que quiere decir que se establece un valor, que una vez superado por el jugador resuelva la trama.

3.3.3. Módulo de puja conjunta

Es un módulo que complementa al anterior, el cual permitirá que varios jugadores trabajen en equipo para resolver una trama.

3.3.4. Módulo de temporizador

El propósito de este módulo es de poder establecer tiempo de apertura y cierre de una trama, por lo que un Master podría programar una trama para que se inicie mañana a las 3:30 y terminara pasado mañana a las 4:30, dando sólo ese tiempo para resolverla.

3.3.5. Módulo de tiradas

Con este módulo nos acercamos más a los juegos de rol clásicos, donde un jugador que desea hacer una acción ha de hacer una tirada basada en las características y atributos establecidos en la ficha. Este módulo, permitirá establecer que tiradas ha de realizar el sistema cogiendo como referencia los datos de la ficha del jugador y una dificultad.

Hay que tener en cuenta que este módulo tiene muchas funcionalidades internas, las cuales enumeramos:

- 1) Realizar tiradas de dados de dos a veinte caras.
- 2) Añadir bonificadores y penalizadores a la tirada.
- 3) Dificultad de la tirada, si hay que sacar al alza o a la baja.
- 4) Si queremos el resultado junto o separado.
- 5) En caso de separado, si la dificultad para lograr un éxito es para cada dado o en conjunto.
- 6) Si el resultado es separado, la posibilidad de que cierto rango de resultados resten éxitos.

3.3.6. Módulo de árbol condicional

La funcionalidad de este módulo es la de crear un árbol de tramas, es decir se establece una trama inicial como raíz y a partir de ahí se le indicará en caso de éxito que trama se tendrá que activar y en caso de fallo que otra trama tendrá que activar, así hasta que llegue a una trama hoja, o lo que es lo mismo, una trama que no activa a otra.

3.3.7. Módulo de múltiples resultados

El servicio que nos puede ofrecer este módulo es el de que una trama no se limite a resuelta o no, es decir, permitiéndonos tener franjas de resultados dependiendo de a dónde se haya llegado con el nivel de pujas o cómo de exitosa ha sido la tirada.

4. Ruby on Rails

Ruby on Rails, también conocido como RoR o Rails, es un framework de código abierto del lado de servidor que nos facilita la construcción de grandes aplicaciones web, escrito en el lenguaje de programación Ruby.

RoR maneja el paradigma de la arquitectura Modelo Vista Controlador (MVC) que nos permite configurar el controlador para darle lógica a nuestras vistas, y permitirnos manejar la información de una base de datos mediante consultas al ActiveRecord en base a los atributos o campos de nuestro modelo creado previamente.

Los principios fundamentales de RoR incluyen:

- **No te repitas:** del inglés Don't repeat yourself o DRY, significa que las definiciones deberían hacerse una sola vez. Dado que RoR es un full stack framework, los componentes están integrados de manera que no hace falta establecer puentes entre ellos.
- **Convención sobre configuración:** el programador sólo necesita definir aquella configuración que no es convencional. Así, cuando se diseña una aplicación partiendo de cero sin una base de datos preexistente, el seguir las convenciones de Rails significa usar menos código. Sin embargo, el comportamiento puede ser configurado si el sistema debe ser compatible con un sistema anterior.



4.1. Arquitectura Modelo Vista Controlador

El MVC es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son: el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. La MVC en RoR se compone de los siguientes elementos:

4.1.1. Modelo

En las aplicaciones web orientadas a objetos sobre bases de datos, el modelo consiste en las clases que representan a las tablas de la base de datos.

En RoR, las clases del modelo son gestionadas por ActiveRecord. Por lo general, lo único que tiene que hacer el programador es heredar una de las clases ActiveRecord::Base, y el programa averiguará automáticamente qué tabla usar y qué columnas tiene.

Las rutinas de validación de datos (p.ej. `validates_uniqueness_of:checksum`) y las rutinas relacionadas con la actualización (p.ej. `after_destroy:borrar_archivo`, `before_update:actualizar_detalle`) también se especifican e implementan en la clase del modelo.

4.1.2. Vista

La vista es la lógica de visualización, o cómo se muestran los datos de las clases del controlador. Con frecuencia en las aplicaciones web la vista consiste en una cantidad mínima de código incluido en HTML formando así plantillas.

Es necesario escribir un pequeño fragmento de código en HTML para cada método del controlador, que necesita mostrar información al usuario. El "maquetado" o distribución de los elementos de la página se describe separadamente de la acción del controlador y los fragmentos pueden invocarse unos a otros.

4.1.3. Controlador

Las clases del controlador responden a la interacción del usuario e invocan a la lógica de la aplicación, que a su vez manipula los datos de las clases del modelo y muestra los resultados usando las vistas. En las aplicaciones web basadas en MVC, los métodos del controlador son invocados por el usuario usando el navegador web.

La implementación del controlador es manejada por el ActionPack de Rails, que contiene la clase ApplicationController. Una aplicación Rails simplemente hereda de esta clase y define las acciones necesarias con métodos, que pueden ser invocados desde la web, por lo general en la forma:

`http://localhost:3000/controlador/acción`

Que invoca a 'controlador#acción', y presenta los datos usando el archivo de vista `/app/views/controlador/acción.html.erb`, a no ser que el método redirija a algún otro lugar.



4.2. Creación de Rutas

Como hemos visto antes, RoR trabaja las rutas por defecto de la misma manera, primero indica el controlador y después el método, el cual tiene el mismo nombre que la vista.

Pero hay un fichero que nos sirve para poder configurar nosotros las rutas, el cual se llama `routes.rb`, y lo podemos encontrar en `/config`. Este fichero es muy importante, pues aquí podremos configurar las rutas para las distintas vistas y definir cuál es la página principal introduciendo la siguiente línea:

```
root :to => 'controlador#acción'
```

Otras de las cosas que podemos hacer en este fichero es definir alias para las rutas de manera sencilla de la siguiente manera:

```
match /alias => 'controlador#acción'
```

De esta manera en vez de tener que llamar a la ruta completa podríamos llamar:

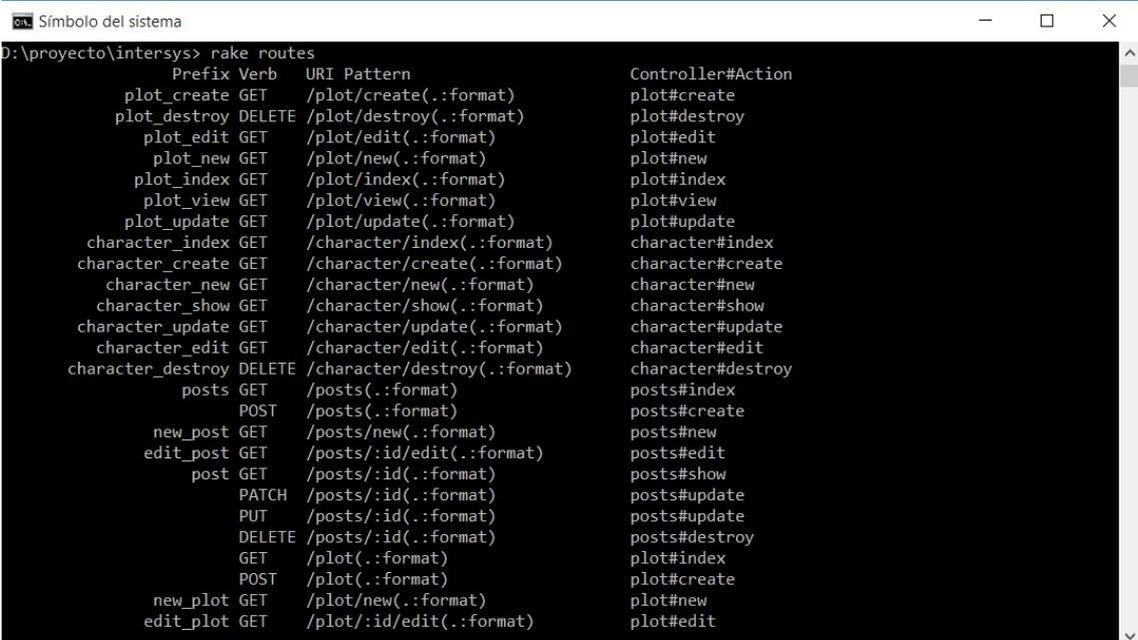
```
http://localhost:3000/alias
```

Si en el código queremos poner un enlace para ir ahí, tendremos que poner `Controlador_acción_path`, o en caso de tener alias, se le llamaría por este tal que así `alias_path`.

Por último y no menos importante, en caso de querer agregar una nueva acción al controlador, tendríamos que añadir la acción al fichero del controlador, crear una vista para ella y agregar la ruta en el fichero `route` con la siguiente línea:

```
get "controlador/acción"
```

Hay un comando que nos facilitará ver las rutas que están definidas, el cual es el siguiente: `rake route`, veamos con profundidad lo que nos muestra el comando.



```
ca Símbolo del sistema
D:\proyecto\intersys> rake routes
Prefix Verb  URI Pattern  Controller#Action
plot_create GET    /plot/create(.:format) plot#create
plot_destroy DELETE /plot/destroy(.:format) plot#destroy
plot_edit GET    /plot/edit(.:format) plot#edit
plot_new GET    /plot/new(.:format) plot#new
plot_index GET    /plot/index(.:format) plot#index
plot_view GET    /plot/view(.:format) plot#view
plot_update GET    /plot/update(.:format) plot#update
character_index GET    /character/index(.:format) character#index
character_create GET    /character/create(.:format) character#create
character_new GET    /character/new(.:format) character#new
character_show GET    /character/show(.:format) character#show
character_update GET    /character/update(.:format) character#update
character_edit GET    /character/edit(.:format) character#edit
character_destroy DELETE /character/destroy(.:format) character#destroy
posts GET    /posts(.:format) posts#index
POST    /posts(.:format) posts#create
new_post GET    /posts/new(.:format) posts#new
edit_post GET    /posts/:id/edit(.:format) posts#edit
post GET    /posts/:id(.:format) posts#show
PATCH /posts/:id(.:format) posts#update
PUT    /posts/:id(.:format) posts#update
DELETE /posts/:id(.:format) posts#destroy
GET    /plot(.:format) plot#index
POST   /plot(.:format) plot#create
new_plot GET    /plot/new(.:format) plot#new
edit_plot GET    /plot/:id/edit(.:format) plot#edit
```

Prefix: Es el identificador que tiene asignado cada URL, si le hemos puesto un alias a una URL esta saldrá aquí.

Verb: nos indica cuál será el verbo HTTP (p.ej. GET, POST, PUT, DELETE, etc.) con el cual se debe realizar la petición

URI Pattern: no muestra cuál será el patrón que seguirá el path de la URL. Ejemplo /posts nos mostrará un listado de todos los posts que hayamos creado.

Controller#Action: nos indica qué controlador y acción serán ejecutados cuando se reciba una petición, si hacemos la petición `http://localhost:3000/posts` utilizando GET, se creará una instancia de controlador posts y se ejecutará la acción index. Hemos de tener en cuenta que muchas RubyGems nos generan rutas que no están visibles en el fichero route.rb

La comprensión del enrutamiento es de vital importancia y fue uno de los principales problemas a la hora de intentar comprender desde cero una red social más completa como Diaspora*.



4.3. Gemas utilizadas

Al igual que muchos otros Frameworks, Rails nos permite instalar librerías y bibliotecas (llamadas RubyGems) desde la consola de Rails o desde el mismo símbolo del sistema (en Windows), a continuación explicaremos brevemente las gemas más importantes para el proyecto:

- **Devise:** nos genera un sistema de registro y login.
- **Carrierwave:** nos permitirá cargar ficheros.
- **Acts_as_votable:** nos permitirá crear un sistema de votaciones que usaremos para hacer el sistema de like.
- **Acts_as_commentable:** podremos construir con esto los post y los comentarios de la red social.
- **Acts_as_follower:** Construiremos con esta gema el sistema de seguimiento de gente en la red social.
- **Public_activity:** Nos permitirá ver la actividad pública de la red.

4.4. Estructura de directorios

Tras ver las cosas más destacables de un proyecto de Rails, ahora pasaremos a ver con más detenimiento la estructura de directorios, viendo así donde se ubican los distintos ficheros de la aplicación web

app: Contiene, los controladores, vistas, modelos, helpers y assets, la mayor parte del trabajo lo realizaremos aquí. En app, las subcarpetas más importantes son:

Models: aquí están todos los modelos de la red social.

Controllers: en este directorio están todos los controladores que se han creado para la web.

Views: dentro de este directorio encontraremos otros directorios que contienen todas las vistas.

config: Contiene archivos de configuración de nuestra aplicación, tales como bases de datos, el fichero de rutas del que hemos hablado en la sección 4.2 entre otros.

db: Contiene información de nuestro esquema actual de la base de datos o bien el archivo con extensión `sqlite` con la base de datos; las migraciones y los seeds (scripts que nos permiten poblar nuestra base de datos, con datos para prueba).

lib: Contiene librerías que nos permiten extender nuestra aplicación

log: Contiene registros de la aplicación como las peticiones del usuario, etc.

public: Contiene archivos estáticos y nuestros assets⁵ compilados.

test/spec: Contiene archivos con pruebas unitarias y funcionales de nuestra aplicación.

tmp: Contiene archivos temporales, tales como la caché, los archivos de sesión, etc.

vendor: Contiene los archivos de librerías externas.

Gemfile/Gemfile.lock: Estos archivos contienen todas las RubyGems necesarias para correr nuestra aplicación, tal y como se mencionó en la sección 4.3.

⁵ Ejemplos de assets son las hojas de estilo, el código en JavaScript de la página o las imágenes estáticas utilizadas en la misma.



5. Implementación de la demo

Parte de la idea de este proyecto era desarrollar una demo técnica mostrando un ejemplo de lo que se podría hacer con el sistema planteado, para ello, lo primero se impuso ciertas limitaciones, debido a la falta de tiempo y personal.

Tras tener claro hasta qué punto se iba a implementar, se intentó iniciar la demo con la red social Diaspora*, ya que disponía de las funcionalidades básicas, pero debido a problemas de comprensión de los conceptos básicos al final se tuvo que desarrollar las funcionalidades básicas de la red social desde cero en RoR para poder comprender los conceptos básicos, además de empezar a plantearnos las bases de datos básica.

Después de terminar con lo básico, se continuó con el desarrollo de las partes específicas de la red social, aquí es donde ya se ha tenido que desarrollar una base de datos más personalizada.

5.1. Limitaciones

Por la falta de tiempo y personal, se ha optado por una demo reducida con las funciones más esenciales de lo planteado en la sección 3 con las siguientes limitaciones:

- Nos hemos limitado a un único ReV en vez de realizar una red social que contenga redes de ReV, pues las funcionalidades importantes las podemos ver perfectamente con un único ReV.
- Escogimos sólo un juego de rol para realizar la demo sobre su sistema en particular, Vampiro la mascarada, de la editorial White Wolf. La razón para que escogiéramos dicho juego es que es muy frecuente realizar ReV basados en esta ambientación.
- Reducimos considerablemente el número de herramientas utilizadas en el sistema de tramas, quedándonos con el sistema de pujas como el principal sistema, pues consideramos que es uno de los más sencillos y lo suficientemente funcional para la demo.



5.2. Problemas de desarrollo con Diaspora*

Diaspora* es un software libre para servidores web que permite desarrollar redes sociales descentralizadas y privadas. El desarrollo del software se encuentra en manos de la comunidad de usuarios.

La red social de Diaspora* está construida sobre la base de una red de servidores, o pods, alojados por muchos individuos e instituciones. Cada servidor opera con una copia del software de Diaspora*, actuando como un servidor web personal. Los usuarios de la red pueden crear una cuenta en cualquier servidor que sea de su elección, pero aun así, pueden interactuar con otros usuarios en todos los demás servidores.

Las vistas de Diaspora* están escritas en HAML (HTML Abstraction Markup Language), el cual es un lenguaje de marcado ligero con el que se puede generar HTML a partir de un DSL, sin emplear el código embebido tradicional. Está diseñado para solucionar varios problemas de los motores de plantillas tradicionales y también para ser un lenguaje de marcado tan elegante como sea posible.

El problema es que para poder usar Diaspora* es necesario conocimientos básicos de RoR, los cuales no son fáciles de adquirir con un proyecto de tal envergadura, y a causa de que las vistas estén escritas en HAML dificultaba el entendimiento del mismo.

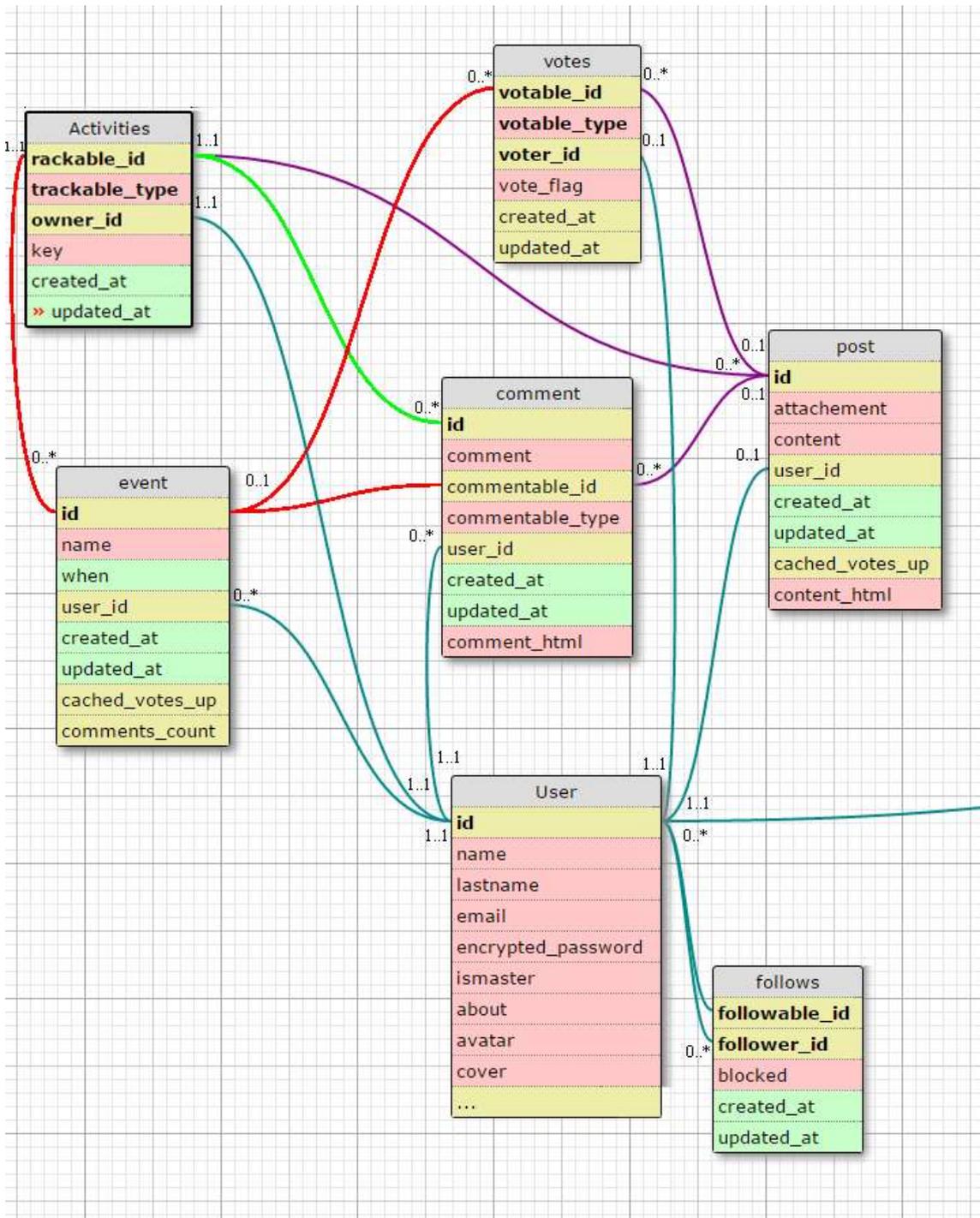
Por ello se optó al final por hacerlo con RoR desde el principio, pues era la manera más sencilla y óptima de comprender el funcionamiento de RoR.

5.3. Diseño de la Base de Datos

La Base de Datos (BD) es más sencilla de comprender si la dividimos en dos partes. La primera es la que hace referencia a la gestión de datos de la red social, y la segunda es donde se gestionan los datos de la partida y el sistema de puja de las tramas.

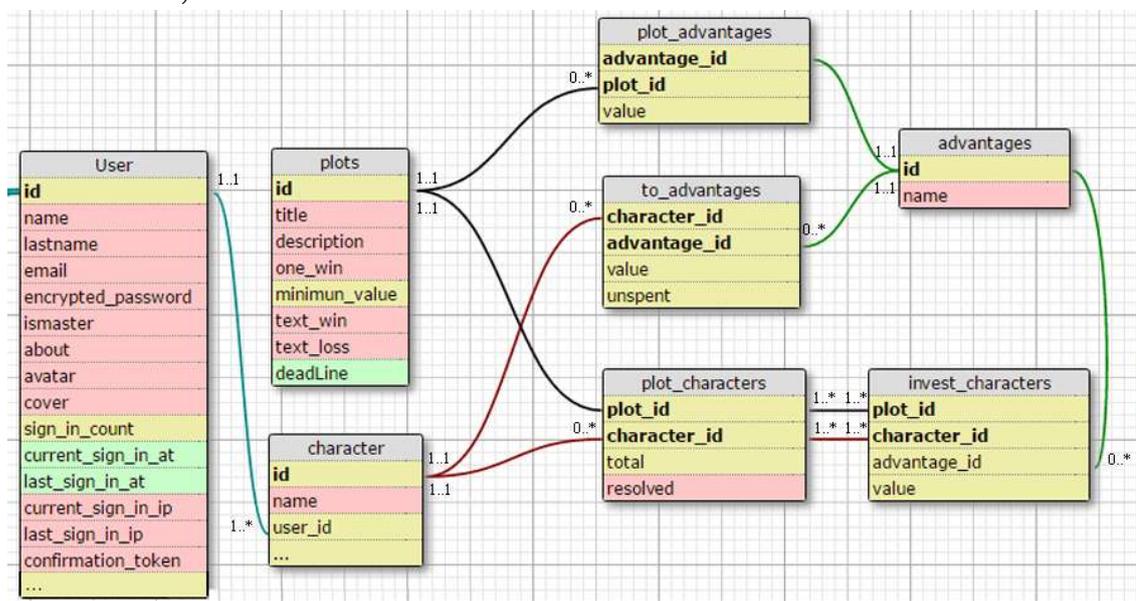
5.3.1. Base de Datos: Red social

Esta parte de la base de datos ha sido prácticamente diseñada por la inclusión de las gemas utilizadas para la construcción de la parte básica de la red social, la cual gestiona los usuarios, sus post, eventos y actividades de la gente que siguen, a continuación les mostramos la estructura de esa parte de la BD.



5.3.2. Base de Datos: Sistema de Tramas y Datos de partida

Esta parte de la base de datos corresponde a las funcionalidades específicas de la red social, es decir, se gestiona los datos del personaje jugador y el sistema de pujas de las tramas, veamos la estructura de la BD.



Como esta parte de la base de datos, sí que ha sido un diseño realizado por nosotros, la explicaremos un poco más.

Todo usuario puede tener más de un carácter (personaje), en la demo actual sólo puede tener una, pero como se plantea que en un futuro se pueda tener más de una, lo hemos planteado con vistas al futuro, una ficha puede contener varias ventajas (ventajas), para ello hemos realizado una tabla intermedia (to_advantages), que asocia que personaje tiene que ventajas y cuál es su valor en ella.

Por otro lado, tenemos los plot (tramas), a los cuales se les asocia unas ventajas y un valor asociado para la puja mediante la tabla intermedia plot_advantages, además de que se debe asignar una trama a uno o varios personajes usando la tabla intermedia plot_character, los cuales podrán invertir sus ventajas en resolver las tramas, esto se recoge en la tabla intermedia invest_character.

Aunque a primera vista pueda parecer complicado, es más sencillo de lo que parece, como podrán comprobar en el Anexo 1.

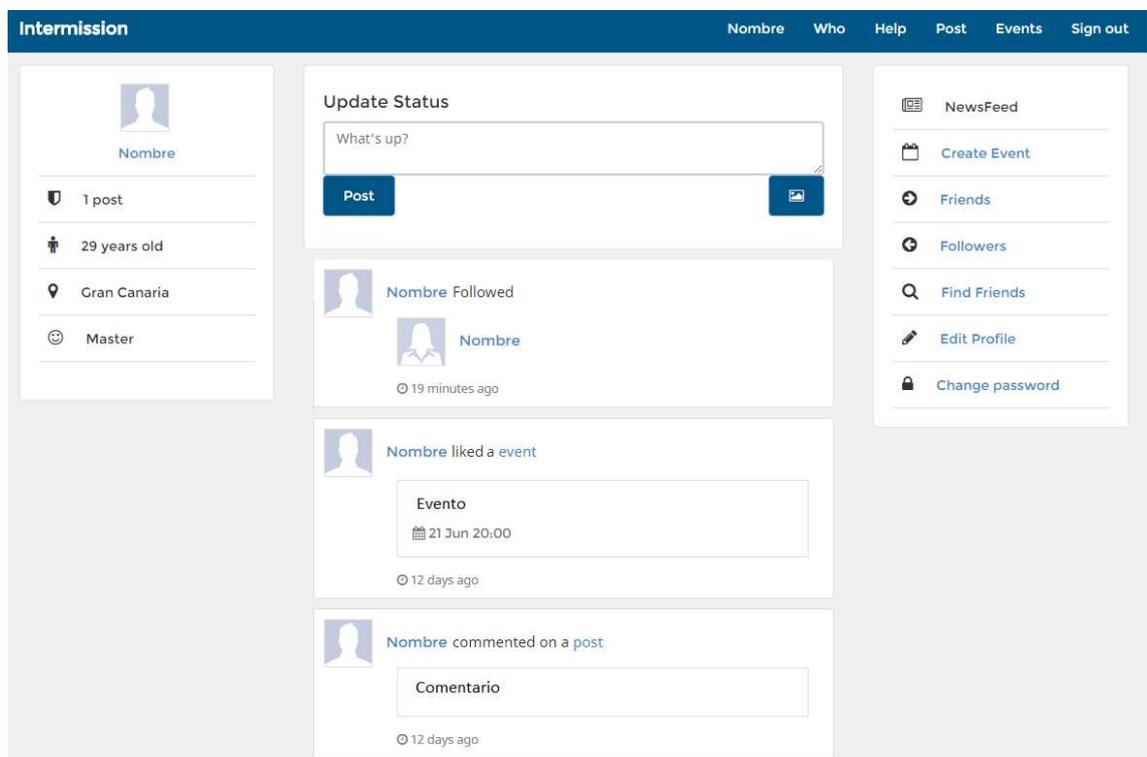
5.4. Funcionalidades básicas

En el fondo, la aplicación no deja de ser una red social, y ¿qué es una red social? es un medio de comunicación social por internet donde personas que comparten alguna relación debido a: intereses, actividades, amistad, etc. Donde ellos pueden interactuar y explorar los intereses de otros usuarios.

Por ello la demo debe constar con ciertas funcionalidades básicas de una red social:

- **Servicio de login:** Donde los usuarios puedan logrearse en el sistema, el servicio de login envía un correo para confirmar que el registro es real.
- **Perfil de usuario:** Los usuarios pueden rellenar campos con más información personal como la nacionalidad, el sexo, una descripción breve, una foto de perfil etc.
- **Sistema de posts:** Permite postear en el muro, es decir, podemos escribir un mensaje y los usuarios que nos siguen podrán ver dicho mensaje, comentar sobre él, darle a me gusta, etc...
- **Sistema de eventos:** similar al sistema de posts, pero para organizar eventos, por lo que adicionalmente tiene una fecha.
- **Sistema de seguidores:** Nos permite agregar a amigos que estén en la red social, para poder ver sus actividades en el muro.

Para ellos hemos usado como referencia el tutorial que pueden encontrar [aquí](#). Como referencia se provee una captura de pantalla de la aplicación desarrollada.



5.5. Funcionalidades específicas

Las funciones específicas para la gestión de ReV es lo que hace especial a esta red social, pues nos facilitarán el trabajo de gestión del ReV, reduciendo considerablemente el tiempo que tendríamos que invertir si los datos no estuvieran recolectados en una base de datos.

Lo primero explicaremos la ficha de jugador, en donde podremos ver los datos con los que trabajaremos y a continuación, explicaremos el sistema de tramas.

5.5.1. Ficha de jugador

En esta ficha podemos ver los datos numéricos del personaje del jugador. El usuario podrá consultarla y solicitar modificaciones que tendrán que ser aprobadas por un organizador. Estos datos son muy importantes, pues el sistema de tramas se basa en la utilización y consulta de dichos datos, como se ha mencionado en la introducción de esta sección, usamos la ficha del juego Vampiro: la Mascarada.

Como podemos ver en la imagen de la siguiente página, una ficha se compone de distintas secciones:

- La primera es una sección con datos del personaje, aunque sólo va a tener relevancia el clan y la generación, pues son datos importantes según el sistema de juego de vampiro.
- La segunda sección es la de atributos, esta sección es muy importante para el sistema, pues la mayoría de tiradas se basan en estos datos, los cuales son datos numéricos que van de uno a cinco.
- La tercera sección son las habilidades, las cuales son tan importantes como los atributos, pues también se usan para las tiradas y los valores van de cero a cinco.
- La cuarta sección es la de ventajas, la cual tiene dos tipos de datos, los que son fijos y van a tener todos los personajes, y los que pueden variar según el personaje. Aun así, estos datos numéricos tienen valores de uno a cinco.
- La última sección contiene tanto datos fijos que suelen ser numéricos como datos que varían, siendo en tal caso cadenas.

Hay que tener muy en cuenta todo esto a la hora de diseñar la base de datos, pues hay que diseñar tablas aparte para los datos que varían y una tabla intermedia que indica a qué personaje le corresponde esos datos y con qué valor, además de que a la ficha se le saca mayor partido cuando se usa el sistema de tiradas, pues con este sistema se usa un 90 % de los datos de la ficha, en cambio con el sistema de puja se usa sólo un 20 % de la ficha.

5.5.2. Sistema de tramas

El sistema de tramas es la herramienta principal para los organizadores de ReV, con el que podrán gestionar las acciones que desean realizar los jugadores con sus personajes entre el ReV anterior y el siguiente.

Como ya hemos comentado, para la demo lo hemos limitado a realizar la herramienta de pujas, la cual requiere que por un lado los jugadores gestionen sus recursos para invertir y que los organizadores creen pujas y designen a que jugadores va dirigida.

Veamos con más detalle el sistema de tramas, para que sea más sencillo de comprender, primero veremos las funcionalidades del organizador, seguido tras las del jugador.

5.5.2.1. Organizador

Los organizadores, son los encargados de la creación y asignación de tramas a los jugadores, adicionalmente, tendrán que realizar un control periódico de cómo han ido evolucionando las tramas y analizar los resultados para ver cómo afecta, tanto al jugador como a la ambientación del ReV en desarrollo. Veamos con más detalles estas herramientas.

5.5.2.1.1. Creación de tramas

Las tramas requieren de los siguientes campos: Título, Descripción, Tipo, Valor a alcanzar, Fecha límite, Texto de éxito, Texto de fracaso y el listado de recursos válidos para pujar con su valor de puja. Explicaremos estos campos a continuación:

- **Título:** Nombre de la trama.
- **Descripción:** Resumen conciso de la trama, el cual dará pistas de los recursos válidos para pujar.
- **Tipo:** Este campo es para indicar si es una puja inglesa, o más conocida como puja ascendente, es decir, sólo el jugador del ReV que pujará más resolverá la trama, o por el contrario, es una trama genérica y todos los jugadores que cumplan las condiciones resolverán la trama.
- **Valor a alcanzar:** requisito mínimo de valor de recursos a pujar para poder resolver la trama.
- **Fecha límite:** Fecha en la que finaliza la puja y se calcula quien es el ganador de la puja, si no hay ganador, el master podrá crear una prorroga donde los puntos gastados se acumulen con nuevos.
- **Texto de éxito:** Texto que saldrá a los jugadores en caso de ganar la puja.
- **Texto de fracaso:** Texto que saldrá a los jugadores en caso de perder la puja.
- **Listado de recursos:** Los recursos que la trama admite como válidos y el valor que tiene cada uno.

Tras toda esta información, puede parecer complicado su funcionamiento, pero se entenderá mejor con un ejemplo, pensemos que tenemos la siguiente trama creada:

Título: Explicar una trama.

Descripción: Intentare explicar como funciona internamente las tramas al lector

Tipo: inglesa

Valor a alcanzar: 10

Fecha límite: 01/01/2017

Texto de éxito: ¡Lo conseguiste!, el lector entendió a la perfección su funcionamiento.

Texto de fracaso: Lo siento mucho, el lector no entendió nada.

Listado de recursos:

| Nombre del recurso | Valor del recurso |
|--------------------|-------------------|
| Dialéctica | 3 |
| Gesticulación | 2 |
| Empatía | 5 |

Viendo este ejemplo, los jugadores que inviertan suficientes recursos para que la suma de valores de los mismos iguale o supere el valor a alcanzar para resolver la trama, por ejemplo , un jugador que invierta uno de dialéctica (3 puntos), uno de gesticulación (2 puntos), uno de belleza (0 puntos, al no estar en la tabla) y uno de empatía (5 puntos), resolvería la trama sin problemas, ya que la suma de los valores de los recursos invertidos sería diez, esto mismo pasaría con un jugador que invirtiera dos en dialéctica, y dos en gesticulación, pues los dos puntos de dialéctica valdrían seis y los dos puntos de gesticulación valdrían cuatro. Pero si por ejemplo hubiera un jugador que sólo invirtiera uno de dialéctica y uno de gesticulación, se quedaría a mitad, y el organizador decidiría si darle una prórroga o no. Si se la diera, los puntos invertidos se añadirían a los anteriores, por lo que en el último ejemplo, el jugador podría conseguirlo en 2 rondas.

The screenshot shows the 'New Plot' creation screen. On the left is a user profile for 'Nombre' with details like '1 post', '29 years old', 'Gran Canaria', and 'Master'. The main form has fields for 'Title', 'Description', 'Minimum value', 'English auction' (checked), 'Description winner', 'Description loser', a date field, a 'Contacto Politico' dropdown, a 'valor' input field, and a 'Player1' dropdown. There are 'Add', 'Assign', and 'Save' buttons. On the right is a sidebar with navigation options: NewsFeed, View users, View plots, Create plot, Create Event, Friends, Followers, Find Friends, Edit Profile, and Change password.

Como referencia en la siguiente página se provee una captura de pantalla de la pantalla de creación de una trama de la aplicación desarrollada.



Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

5.5.2.1.2. Asignación de tramas

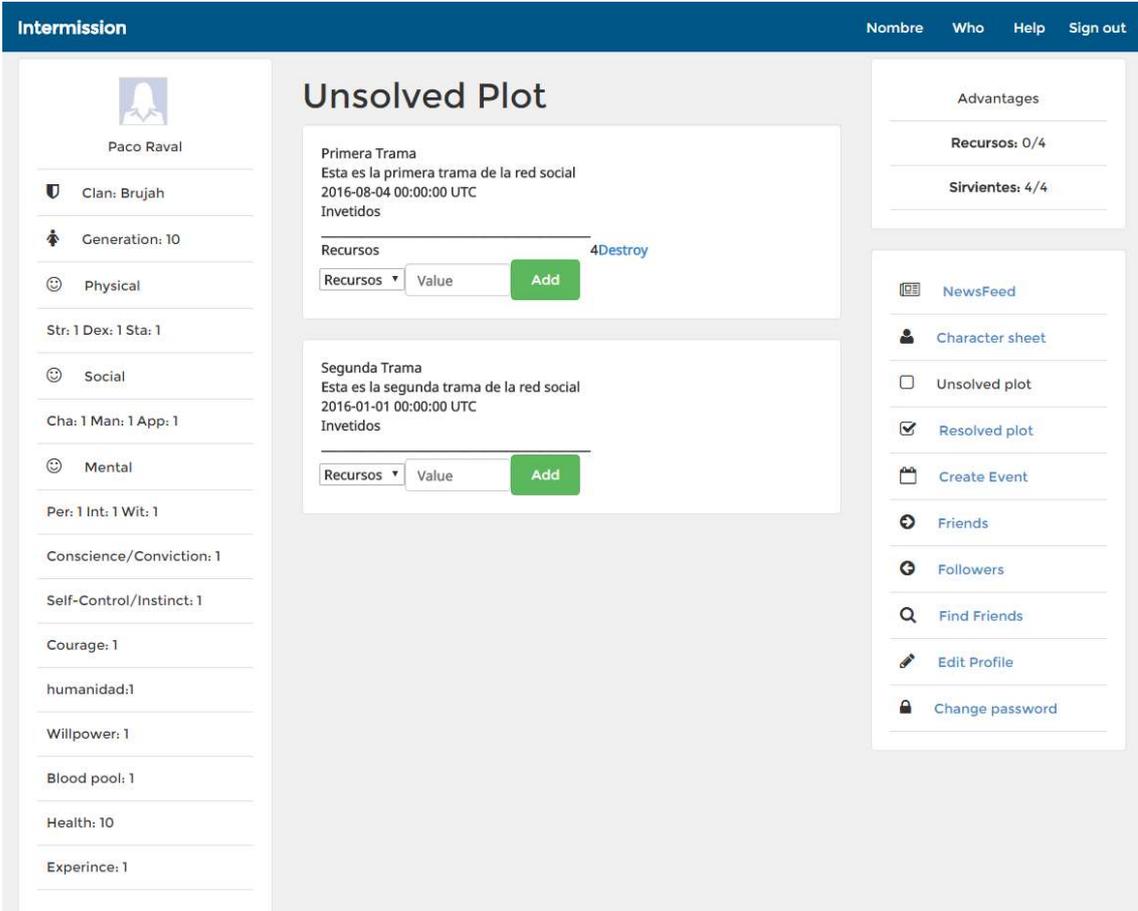
Las tramas se han de asignar a los jugadores, pues normalmente no todos los jugadores tendrán las mismas tramas, aunque se puede dar el caso de que varios jugadores tengan la misma trama. También pueden existir tramas genéricas para todos los jugadores, por lo que a la hora de asignar existe una función para asignársela a todos.

5.5.2.2. Jugador

Los jugadores, como principales usuarios de la red social, tienen como función interactuar entre ellos y el sistema, y dispondrán del sistema de pujas de tramas para realizar su función.

5.5.2.2.1. Sistema de pujas de tramas

Como hemos visto en la sección 5.5.2.1.1 Creación de tramas, donde se explicaba en detalle la creación y el funcionamiento de dicho proceso, a una trama se le asignan que recursos son válidos y que valor tienen de cara a resolver la trama.



The screenshot displays the 'Intermission' web application interface. At the top, there is a navigation bar with the name 'Intermission' and links for 'Nombre', 'Who', 'Help', and 'Sign out'. The main content area is divided into three columns. The left column shows a user profile for 'Paco Raval' with various attributes: Clan: Brujah, Generation: 10, Physical, Str: 1 Dex: 1 Sta: 1, Social, Cha: 1 Man: 1 App: 1, Mental, Per: 1 Int: 1 Wit: 1, Conscience/Conviction: 1, Self-Control/Instinct: 1, Courage: 1, humanidad: 1, Willpower: 1, Blood pool: 1, Health: 10, and Experiencia: 1. The middle column is titled 'Unsolved Plot' and contains two plot entries. The first entry, 'Primera Trama', is dated 2016-08-04 00:00:00 UTC and has 4 resources available. The second entry, 'Segunda Trama', is dated 2016-01-01 00:00:00 UTC and also has 4 resources available. Each entry includes a 'Recursos' dropdown menu, a 'Value' input field, and an 'Add' button. The right column contains a sidebar with various navigation options: Advantages, Recursos: 0/4, Sirvientes: 4/4, NewsFeed, Character sheet, Unsolved plot, Resolved plot, Create Event, Friends, Followers, Find Friends, Edit Profile, and Change password.

Los jugadores dispondrán de una pantalla donde podrán ver que tramas tienen asignadas y que recursos tienen disponibles para invertir, teniendo que distribuir sus recursos para resolver las tramas. Cuando llegue la fecha límite, estas se resolverán y pasarán al historial de tramas, donde los jugadores podrán ver los resultados de sus

pujas. Los jugadores también dispondrán de una sección donde proponer tramas a los organizadores.

Intermission Nombre Who Help Sign out



Paco Raval

 Clan: Brujah

 Generation: 10

 Physical

Str: 1 Dex: 1 Sta: 1

 Social

Cha: 1 Man: 1 App: 1

 Mental

Per: 1 Int: 1 Wit: 1

Conscience/Conviction: 1

Self-Control/Instinct: 1

Courage: 1

humanidad:1

Willpower: 1

Blood pool: 1

Health: 10

Resolved Plot

Primera Trama
Esta es la primera trama de la red social
2016-08-04 00:00:00 UTC
Result
Gane

0 1 1
Segunda Trama
Esta es la segunda trama de la red social
2016-01-01 00:00:00 UTC
Result
perdi

4 trama
trama gorda
2016-01-01 00:00:00 UTC
Result
Aqui esta tu princesa

25 3 1
3 trama
huy que complicada
2016-01-01 00:00:00 UTC
Result
Perdi

Advantages

Recursos: 0/4

Sirvientes: 4/4

 NewsFeed

 Character sheet

 Unsolved plot

Resolved plot

 Create Event

 Friends

 Followers

 Find Friends

 Edit Profile

 Change password



6. Conclusiones

Este trabajo ha requerido mucho tiempo debido a la investigación y aprendizaje, además de ser un proyecto extenso y con pretensiones de futuro, y con el cual hemos visto que es un proyecto factible y que si se plantea correctamente el modelo de negocio existe un nicho de mercado disponible.

Hemos conseguido implementar una Demo funcional durante el desarrollo del proyecto, la cual nos ha servido para introducirnos correctamente en el mundo de RoR y comprender sus principios básicos, además de ser un excelente campo de pruebas para el resto de funcionalidades pensadas en desarrollarse en un futuro.

Tras la realización de la demo, pensamos que intentar realizar un desarrollo en Diaspora* es algo más factible, ya que tras todo los conocimientos adquiridos sería viable realizar la siguiente versión con Diaspora*.

Antes de realizar una versión en Diaspora* consideramos importante implementar el resto de módulos aún no implementados en la demo, y tras ello, realizar la versión final en la plataforma Diaspora*.

7. Bibliografía

1. **Ponce Moreno, Santiago.** *Ruby on rails. Desarrollo Práctico de aplicaciones web.* Madrid: Editorial RC LIBROS, 2013. ISBN: 9788494127274
2. **Holzner, Steve.** *Beginning Ruby on Rails.* Editorial Wiley/Wrox, [Ebook] 2007. ISBN: 978-1-4571-0698-9
3. **Rappin, Noel.** *Professional Ruby on Rails.* Editorial Wiley/Wrox, [Ebook] 2008. ISBN: 978-1-4571-0439-8
4. **Diaspora* Project.** [En Línea] 2016. [Acceso el: 10 de Febrero de 2016] <https://diasporafoundation.org/>
5. **Wikipedia, la enciclopedia Libre: Diaspora.** [En Línea] 2016. [Acceso el: 12 de Febrero de 2016] https://es.wikipedia.org/wiki/Diaspora*
6. **The Official diaspora* wiki.** [En Línea] 2016. [Acceso el: 17 de Febrero de 2016] https://wiki.diasporafoundation.org/Main_Page
7. **Github: diaspora* social network.** [En Línea] 2016. [Acceso el: 17 de Febrero de 2016] <https://github.com/diaspora>
8. **Diaspora Community.** [En Línea] 2016. [Acceso el: 22 de Febrero de 2016] <https://www.loomio.org/g/EseV9p4X/diaspora-community>
9. **Wikipedia, la enciclopedia Libre: HAML.** [En Línea] 2016. [Acceso el: 4 de Marzo de 2016] <https://es.wikipedia.org/wiki/Haml>
10. **Ruby Gems HAML.** [En Línea] 2014 [Acceso el: 11 de Marzo de 2016] <https://rubygems.org/gems/haml/versions/4.0.7>
11. **HamL Tutorial.** [En Línea] 2015-2016. [Acceso el: 14 de Marzo de 2016] <http://haml.info/tutorial.html>
12. **HamL Documentacion.** [En línea] 2015. [Acceso el: 16 de Marzo de 2016] <http://haml.info/docs/yardoc/file.REFERENCE.html>
13. **Aprendiendo HAML.** [En Línea] 2015-2016. [Acceso el: 18 de Marzo de 2016] <http://picandocodigo.net/2011/aprendiendo-haml/>
14. **Convert HTML to HAML.** [En Línea] 2016 [Acceso el: 22 de Marzo de 2016] <http://htmltohaml.com/>
15. **Ruby on Rails: El desarrollo web que no molesta.** [En Línea] 2016. [Acceso el: 4 de Mayo de 2016] <http://www.rubyonrails.org.es/>
16. **Rails for zombies.** [En línea] 2016. [Acceso el: 11 de Mayo de 2016] <http://railsforzombies.org/>
17. **Getting Started with Rails.** [En línea] 2016. [Acceso el: 18 de Mayo de 2016] http://guides.rubyonrails.org/getting_started.html
18. **Youtube: Ruby on Rails - Red social.** [En línea] 2011. [Acceso el: 24 de Mayo de 2016] <https://www.youtube.com/watch?v=xJzDz3lUImE>



19. **How to develop a social network using Ruby on Rails.** [En línea] 2015. [Acceso el: 6 de Junio de 2016] <https://medium.com/rails-ember-beyond/how-to-build-a-social-network-using-rails-eb31da569233#.jv9jut1d4>
20. **Justin Achilli, Russell Bailey, Eddy Webb y Matthew McFarland.** *Edición 20 aniversario Vampiro la mascarada.* Madrid: Editorial Nosolorol, 2014. ISBN 978-84-942851-2-7
21. **Ruby on rails: active record migration.** [En línea] 2016. [Acceso el: 10 de Agosto de 2016] http://edgeguides.rubyonrails.org/active_record_migrations.html
22. **Making Select boxes with ease.** [En línea] 2016. [Acceso el: 17 de Agosto de 2016] http://guides.rubyonrails.org/form_helpers.html#making-select-boxes-with-ease

Anexo 1

1. Manual de usuario

Bienvenido al manual de usuario de Intermision System, la herramienta online para gestionar Roles en Vivo (ReV), en la que los organizadores podrán recopilar la información de sus jugadores entre los eventos

En este documento explicaremos las funcionalidades básicas de la red social, las cuales son las siguientes:

- Registro y login de usuario.
- Perfil de usuario.
- Muro de anuncios.
- Seguidores.

Posteriormente, explicaremos las funcionalidades más específicas, para usuarios organizadores o jugadores.

2. Registro y login de usuario

Para registrarnos tenemos que acudir a www.IntermisionSystem.com⁶ en la cual nos encontraremos con la siguiente pantalla.



⁶ Página web inexistente, nosotros, como trabajamos en local, usamos la dirección por defecto de los servidores de RoR : <http://localhost:3000/>



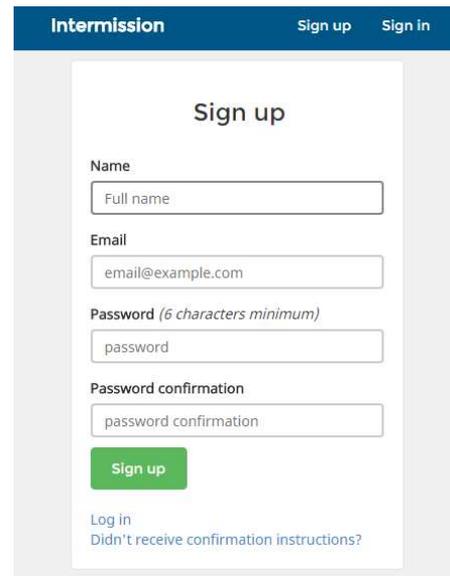
Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

En esta pantalla de bienvenida, nos tenemos que dirigir a Sign up, la cual nos dirigirá a la pantalla de registro.

En esta pantalla tenemos que rellenar los datos que nos piden, los cuales son los siguientes, nombre, Email, Contraseña, confirmación de contraseña. Tras rellenar estos datos, le daremos a Sign up, tras lo cual nos enviará un correo para confirmar la cuenta.

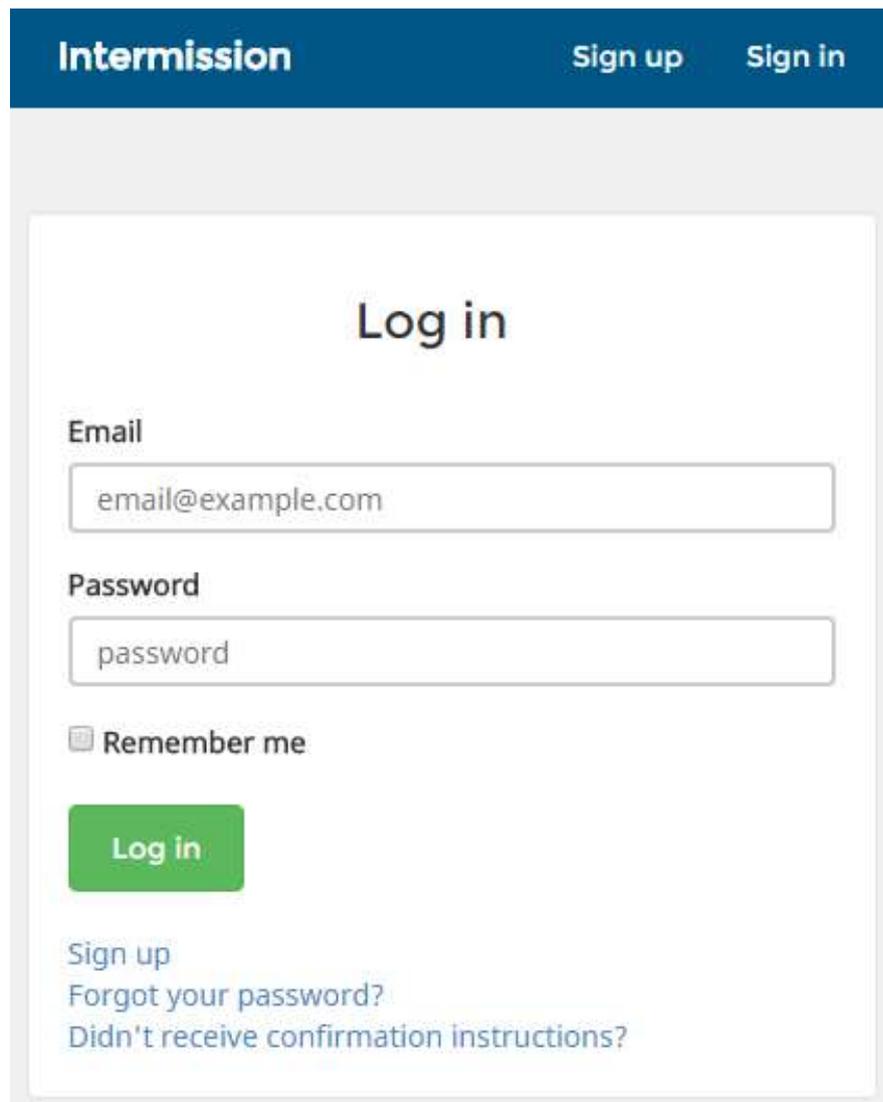
Los correos no se pueden repetir, por lo que si intenta registrarse con un correo existente le saldrá un mensaje como este.

Email has already been taken



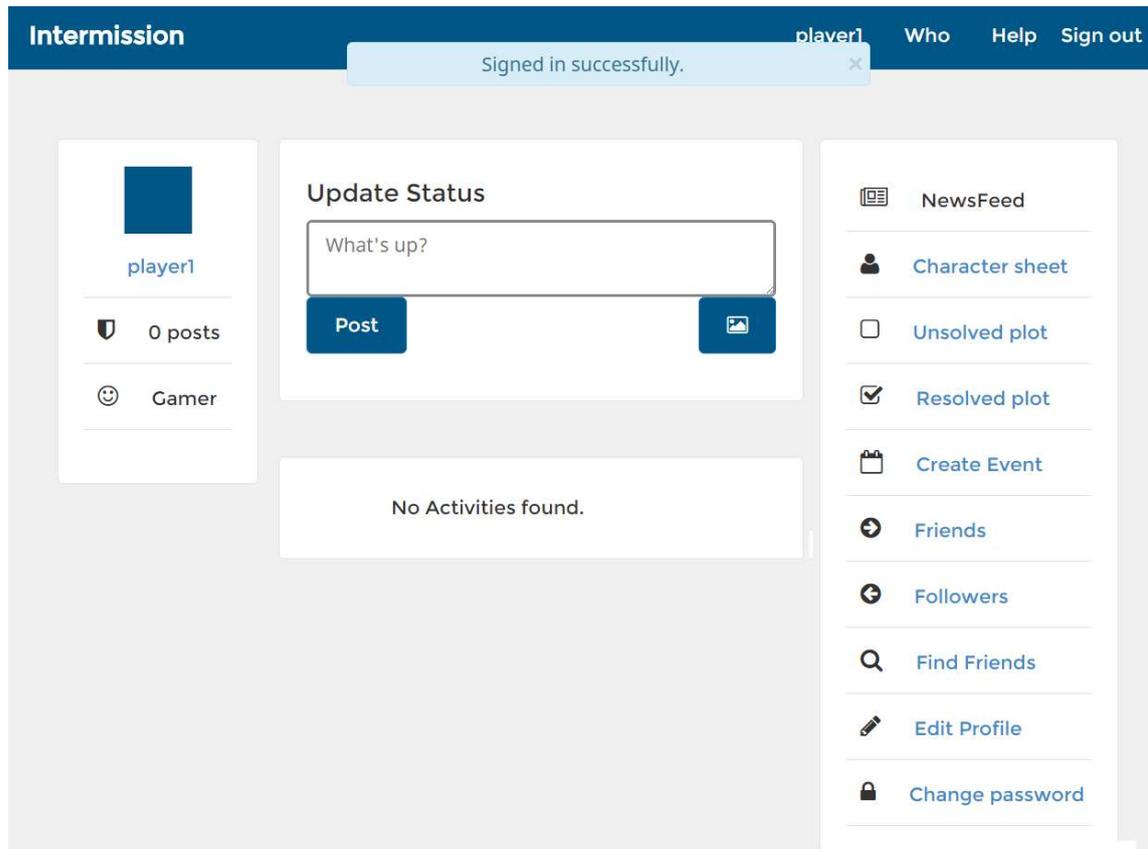
The screenshot shows the 'Sign up' form in the Intermission system. The form is titled 'Sign up' and is located within a blue header bar that also contains the 'Intermission' logo and 'Sign up' and 'Sign in' links. The form fields are: 'Name' (Full name), 'Email' (email@example.com), 'Password' (6 characters minimum, password), and 'Password confirmation' (password confirmation). A green 'Sign up' button is at the bottom. Below the button, there are links for 'Log in' and 'Didn't receive confirmation instructions?'.

Una vez hayamos confirmado el correo, ya nos podremos loguear dirigiéndose a Sign in, en la cual podremos ver la siguiente pantalla.



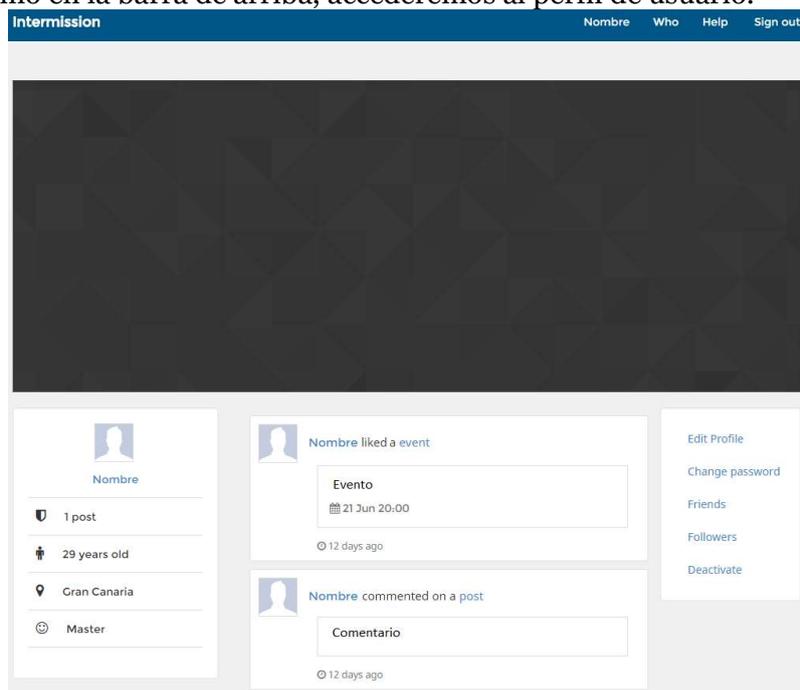
The screenshot shows the 'Log in' form in the Intermission system. The form is titled 'Log in' and is located within a blue header bar that also contains the 'Intermission' logo and 'Sign up' and 'Sign in' links. The form fields are: 'Email' (email@example.com) and 'Password' (password). There is a 'Remember me' checkbox. A green 'Log in' button is at the bottom. Below the button, there are links for 'Sign up', 'Forgot your password?', and 'Didn't receive confirmation instructions?'.

En esta pantalla nos bastará con introducir el correo y la contraseña para poder acceder al muro del usuario.



3. Perfil de usuario

Ahora que hemos visto como registrarse y acceder a la red, veremos cómo modificar el perfil de usuario. Para ello en la pantalla del muro podemos acceder de múltiples maneras, si apretamos en la foto, o en el nombre que está tanto debajo de la foto como en la barra de arriba, accederemos al perfil de usuario.



Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

El perfil por defecto tiene los datos mínimos, si deseamos agregar datos adicionales, nos dirigiremos a Edit Profile, sin embargo, si queremos cambiar la contraseña, lo haremos en Change password. En caso de que quisiéramos eliminar la cuenta, apretaremos a Desactive.

Intermission Nombre Who Help Sign out

Edit Profile

Avatar



Ningún archivo seleccionado

Cover



Ningún archivo seleccionado

Name

About

Location

Phone number

Dob

Sex

Como hemos podido ver en la imagen de la página anterior, los datos que podemos añadir adicionalmente a nuestro perfil en Edit Profile es una foto, un fondo para nuestro perfil, y una gran cantidad de datos personales como el nombre, la localidad, sexo, etc...

4. Muro de anuncios

El muro de anuncios es donde podemos ver la actividad, tanto nuestra como la de los otros usuarios que seguimos. Si nos fijamos, al principio del muro nos permite publicar un post.



Para ello sólomente tenemos que rellenar el campo de estado y apretar a post, si quisiéramos agregar una foto, apretaríamos el icono de foto, como vemos en la imagen.



Una vez publicado, el post saldrá en su muro y en el de todos los que le sigan. A un post publicado se le puede hacer múltiples cosas, entre ellas editarlo (botón de lápiz), eliminarlo (botón de basura), indicar que nos gusta (botón de like) o hacer un comentario (Botón de Comment), los cuales podremos ver en la página personal del post, a la cual podemos acceder si apretamos al link donde pone post.

Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

Intermission Nombre Who Help Sign out

 Nombre created a post

Comentario del master

2 months ago

0 likes 3 comments

unlike comment

Comments

Enter Comment

comment

 Nombre

juas

 Nombre

comentario del jugador



También podemos crear eventos, los cuales son casi idénticos a los post, pero con un campo de fecha, para poder crear un evento, nos dirigiremos a create event,

Event name

Event Name

When?

When is it?

create

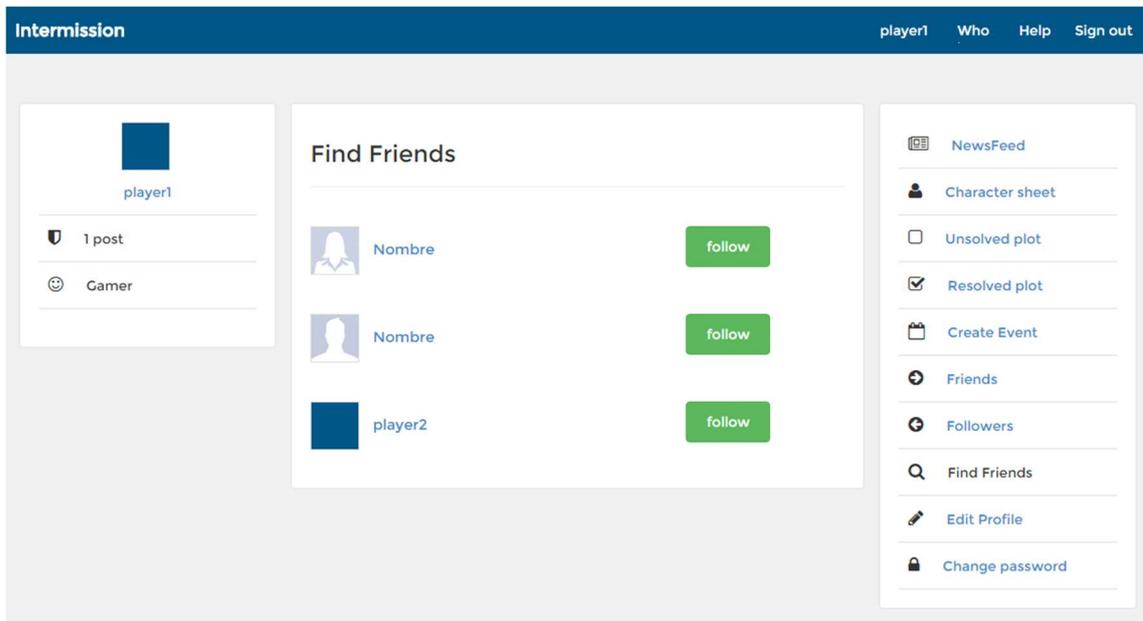
Todos los post, event, like como los comentarios se te notificarán en el muro, así que no te perderás ninguna actividad de ningún seguidor.

The screenshot shows the user profile for 'player1' on the 'Intermission' platform. The page is divided into three main sections: a profile sidebar, a central activity feed, and a right-hand navigation menu.

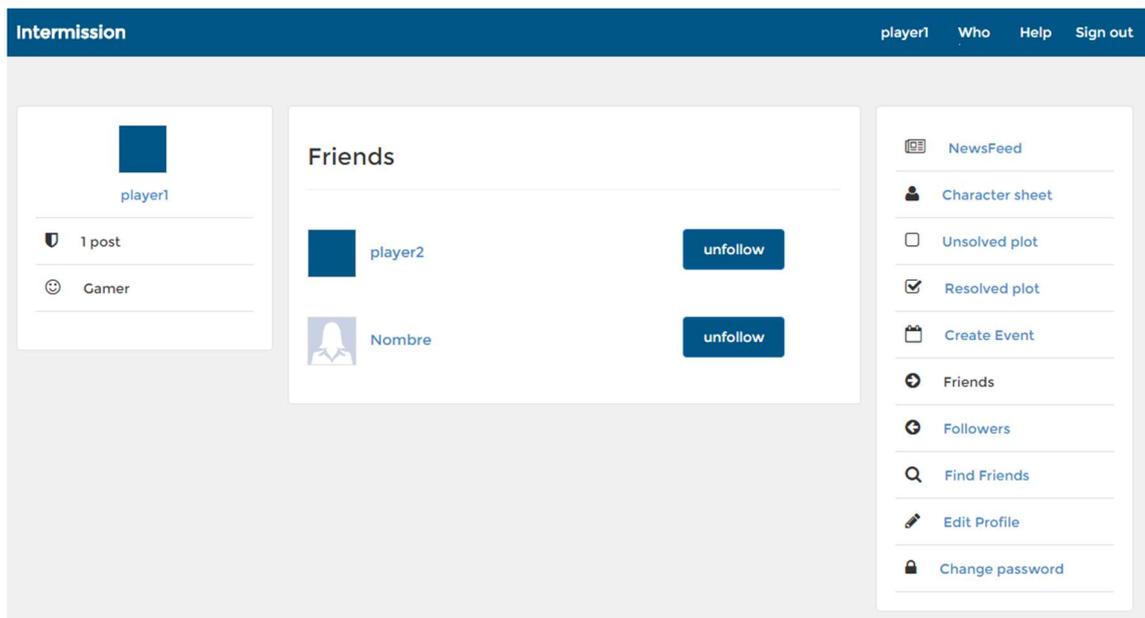
- Profile Sidebar (Left):** Displays the user's name 'player1', a shield icon indicating '1 post', and a 'Gamer' badge.
- Update Status (Top Center):** A form with the placeholder text 'What's up?' and a 'Post' button.
- Activity Feed (Center):**
 - Event:** 'player1 created an event' titled 'buscar piso' on '02 Sep 21:00', posted '7 minutes ago'. It has 0 likes and 0 comments.
 - Comment:** 'player1 commented on a post' with the text 'mua', posted '9 minutes ago'.
 - Post:** 'player1 created a post' with the text 'Esto es un Post', posted '16 minutes ago'. It has 0 likes and 1 comment.
- Navigation Menu (Right):** A list of links including 'NewsFeed', 'Character sheet', 'Unsolved plot', 'Resolved plot', 'Create Event', 'Friends', 'Followers', 'Find Friends', 'Edit Profile', and 'Change password'.

5. Seguidores

Para poder ver en el muro las actividades de otro usuarios, lo primero es agregarlos como friends , paras ello iremos a Find Friends, donde podemos agregar a otro usuarios.



Una vez agregados, nos saldrán sus actividades en el muro. Si queremos consultar quienes nos siguen, iremos a sección followers, pero si queremos saber a quién seguimos, iremos a la sección friends.



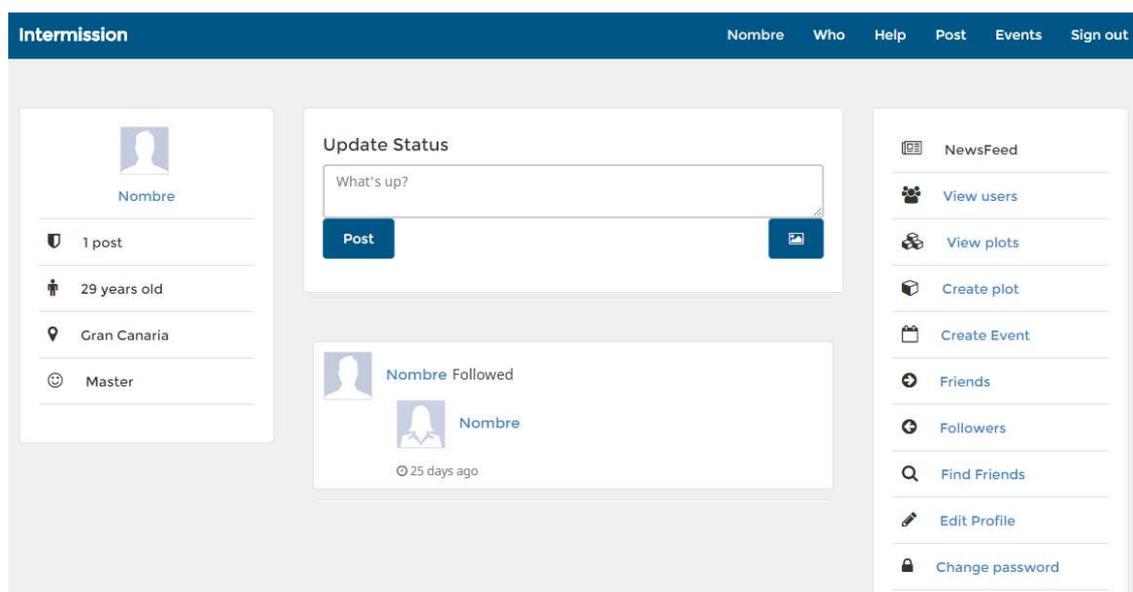
Tanto la sección de friends como followers son similares, sólo se diferencian por que el primero muestra a la gente que seguimos y el segundo la gente que nos siguen.

6. Manual de usuario: Organizador

En esta sección explicaremos las funciones especiales que tiene un usuario organizador, las cuales son las siguientes:

- Control de usuarios.
- Creación de tramas y asignación de tramas.

También tenemos que tener en cuenta que los organizadores tienen secciones que los jugadores no disponen.



Como podemos ver, arriba a la derecha disponen las siguientes secciones:

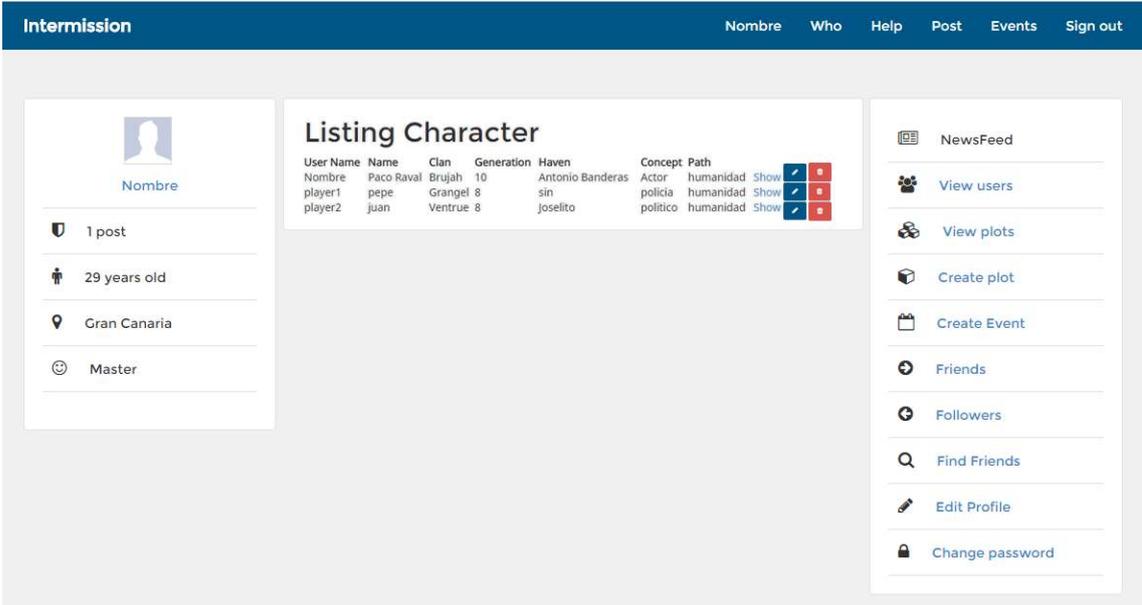
- Post: Listado de posts realizados en la red social, donde podemos editarlos, editarlos o borrarlos.
- Events: Listado de events realizados en la red social, donde podemos editarlos, editarlos o borrarlos

En la sección de la derecha, podemos ver las secciones:

- View user: Listado donde podrán gestionar las fichas de los jugadores, es decir, editarlas o borrarlas.
- View plot : Listado donde podemos administrar todas las tramas de la red, en ella podremos ver, editar o eliminar una trama.
- Create plot: Donde podremos crear tramas nuevas.

7. Control de usuarios

Como organizador, es necesario consultar los datos de los personajes (character) de vez en cuando, para ello disponemos de un listado de usuarios en donde consultar los datos.



The screenshot shows the Intermission system interface. At the top, there is a navigation bar with the name 'Intermission' on the left and links for 'Nombre', 'Who', 'Help', 'Post', 'Events', and 'Sign out' on the right. The main content area is divided into three sections:

- User Profile:** On the left, there is a profile card for a user named 'Nombre'. It includes a placeholder for a profile picture, the name 'Nombre', and several statistics: '1 post', '29 years old', 'Gran Canaria' (location), and 'Master' (role).
- Listing Character:** In the center, there is a table titled 'Listing Character' with the following data:

| User Name | Name | Clan | Generation | Haven | Concept | Path | Actor | humanidad | Show |
|-----------|------------|---------|------------|------------------|----------|-----------|-------|-----------|------|
| Nombre | Paco Raval | Brujah | 10 | Antonio Banderas | policia | humanidad | Show | ✓ | ✖ |
| player1 | pepe | Grangel | 8 | sin | politico | humanidad | Show | ✓ | ✖ |
| player2 | juan | Ventru | 8 | joselito | politico | humanidad | Show | ✓ | ✖ |
- Navigation Menu:** On the right, there is a vertical menu with the following options: 'NewsFeed', 'View users', 'View plots', 'Create plot', 'Create Event', 'Friends', 'Followers', 'Find Friends', 'Edit Profile', and 'Change password'.

Aquí podemos editar personajes de los usuarios e incluso eliminarlos, también podemos hacer esto con los posts y los events en unas pantallas similares, en la que sólo cambia los datos mostrados.

8. Creación de tramas y asignación de tramas.

La funcionalidad principal de un organizador es crear y asignar tramas, la cuestión es, ¿qué es una trama y para qué sirve? Pues lo mejor para entenderlo es mediante un ejemplo.

Imaginemos que somos organizadores de eventos de ReV de continuidad, cuya temática es el grado de ingeniería informática. Montamos un evento en el que los participantes son alumnos de cuarto que han de proponer un proyecto de final de grado, y conseguir que un profesor (uno de los organizadores) se lo acepte. Al terminar el evento, los participantes ya tiene a un organizador que les ha acogido su proyecto, pero el siguiente evento, el cual será la defensa de su TFG, será dentro de 3 meses, entonces ¿cómo simular que durante esos tres meses han estado trabajando en su TFG?, es aquí donde entran las tramas.

Los jugadores, durante estos tres meses, hablarán con los organizadores, para proponerles maneras de avanzar en sus TFG, con esa información, los organizadores crearán durante esos meses tramas específicas para los jugadores, imponiéndoles una dificultad y unos valores a aplicar a las ventajas que usen para pujar. Los jugadores verán por el sistema que se les ha asignado una trama, y pujará en consecuencia para intentar resolverla en el plazo que le ha asignado el organizador.

Cuando un organizador considere que se ha pasado el plazo de puja, al jugador le saldrá el resultado de la misma, y podrá invertir su recursos en resolver otras tramas.

Bien, veamos como creamos y asignamos una trama a un jugador, para crear una trama, nos dirigiremos a create plot, donde veremos la siguiente pantalla.

The screenshot shows the 'New Plot' interface. On the left, there's a user profile card with a placeholder image and the name 'Nombre'. Below it, there are statistics: '1 post', '29 years old', 'Gran Canaria', and 'Master'. The main form area has several input fields: 'Title', 'Description', 'Minimum value', a checkbox for 'English auction', 'Description winner', 'Description loser', and a date field 'dd/mm/yyyy --:--'. There are also dropdown menus for 'Contacto Politico' (with 'valor' selected) and 'Player1'. Green buttons for 'Add', 'Assign', and 'Save' are present. On the right, a sidebar menu includes: NewsFeed, View users, View plots, Create plot, Create Event, Friends, Followers, Find Friends, Edit Profile, and Change password.

Lo primero que debemos saber es que hay dos tipos de pujas, la puja por defecto y la puja inglesa. La puja por defecto es una puja individual, es decir, la trama puede estar asignada a varios jugadores, pero es una trama independiente para cada uno. En cambio, el otro tipo de puja disponible es la puja inglesa o clásica, en la cual, sólo se la lleva el jugador que haya pujado más y cumpla con la dificultad mínima.

En cualquiera de las dos tipos de tramas tendremos que introducir los siguientes datos:

- Título.
- Descripción.
- Dificultad.
- Tipo de puja.
- Descripciones en caso de que gane o pierda.
- Fecha de finalización.
- Asignar a uno o más jugadores.
- Indicar los valores de los recursos.

¿Qué significa indicar el valor de los recursos para esa trama?, lo que significa que invertir uno de ese recurso nos va a aportar X puntos para la puja, puede parecer complicado su funcionamiento, pero se entenderá mejor con un ejemplo, pensemos que tenemos la siguiente trama creada:

Título: Explicar una trama.

Descripción: Intentaré explicar como funciona internamente las tramas al lector

Tipo: inglesa

Valor a alcanzar: 10

Fecha límite: 01/01/2017

Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

Texto de éxito: ¡Lo conseguiste!, el lector entendió a la perfección su funcionamiento.

Texto de fracaso: Lo siento mucho, el lector no entendió nada.

Listado de recursos:

| Nombre del recurso | Valor del recurso |
|--------------------|-------------------|
| Dialéctica | 3 |
| Gesticulación | 2 |
| Empatía | 5 |

Viendo este ejemplo, los jugadores que inviertan suficientes recursos para que la suma de valores de los mismos iguale o supere el valor a alcanzar resolverán la trama, por ejemplo , un jugador que invierta uno de dialéctica (3 puntos), uno de gesticulación (2 puntos), uno de belleza (0 puntos, al no estar en la tabla) y uno de empatía (5 puntos), resolvería la trama sin problemas, ya que la suma de los valores de los recursos invertidos sería diez, esto mismo pasaría con un jugador que invirtiera dos en dialéctica, y dos en gesticulación, pues los dos puntos de dialéctica valdrían seis y los dos puntos de gesticulación valdrían cuatro. Pero si por ejemplo hubiera un jugador que sólo invirtiera uno de dialéctica y uno de gesticulación, se quedaría a mitad, y el organizador decidiría si darle una prórroga o no. Si se la diera, los puntos invertidos se añadirían a los anteriores, por lo que en el último ejemplo, el jugador podría conseguirlo en 2 rondas.

View plots nos permitirá ver un listado de las tramas dándonos la opción de verlas con más detalle, editarlas o eliminarlas.

9. Manual de usuario: jugador

Los jugadores disponen de distintas pantalla informativas, una para ver los datos de su personaje, otras donde ver las tramas, tanto resueltas como no resueltas.

The screenshot displays the Intermission user interface. At the top, there is a dark blue header with the text 'Intermission' on the left and 'Nombre Who Help Sign out' on the right. Below the header, the main content area is divided into three columns. The left column shows a user profile for 'Nombre' with a profile picture, a shield icon indicating '1 post', a person icon indicating '37 years old', a location pin icon for 'Valencia', and a smiley face icon for 'Gamer'. The middle column features an 'Update Status' section with a text input field containing 'What's up?', a 'Post' button, and a photo upload icon. Below this is a 'Nombre Followed' notification showing a profile picture and the name 'Nombre' with a timestamp of '25 days ago'. The right column contains a vertical navigation menu with the following items: 'NewsFeed', 'Character sheet', 'Unsolved plot', 'Resolved plot', 'Create Event', 'Friends', 'Followers', 'Find Friends', 'Edit Profile', and 'Change password'.

En la sección character sheet los jugadores podrán ver los datos de su personaje actualizados.

The screenshot shows a character sheet for a Vampire named Paco Raval. The interface includes a top navigation bar with 'Nombre', 'Who', 'Help', and 'Sign out'. On the left, there's a profile sidebar with a name field, a post count of 1, age of 37, location of Valencia, and a 'Gamer' status. The main content area is titled 'Vampire' and includes:

- The Masquerade**: Name: Paco Raval
- Nature**: Superviviente
- Haven**: Antonio Banderas
- Generation**: 10
- Clan**: Brujah
- Demeanor**: Confromista
- Concept**: Actor

 Below this are sections for **Attributes** (Strength, Dexterity, Stamina, Charisma, Manipulation, Appearance, Perception, Intelligence, Wits) and **Abilities** (Talents, Skills, Knowledges). The **Advantages** section lists Recursos (4), Sirvientes (4), Conscience/Conviction (1), Self-Control/Instinct (1), Courage (1), and various traits like humanidad (1), Willpower (1), Blood pool (1), Health (10), and Experience (1). On the right, a sidebar contains navigation links: NewsFeed, Character sheet, Unsolved plot, Resolved plot, Create Event, Friends, Followers, Find Friends, Edit Profile, and Change password.

En la pantalla de unsolved plot es donde podrás invertir tus recursos para resolver trama que te impondrán los organizadores.

The screenshot shows the 'Unsolved Plot' screen. The top navigation bar is the same as in the previous screenshot. The left sidebar now shows the character's name 'Paco Raval' and various attributes: Clan: Brujah, Generation: 10, Physical (Str: 1, Dex: 1, Sta: 1), Social (Cha: 1, Man: 1, App: 1), Mental (Per: 1, Int: 1, Wit: 1), Conscience/Conviction: 1, Self-Control/Instinct: 1, Courage: 1, humanidad: 1, Willpower: 1, Blood pool: 1, Health: 10, and Experience: 1. The main content area is titled 'Unsolved Plot' and contains two plot entries:

- Primera Trama**: Esta es la primera trama de la red social. Date: 2016-08-04 00:00:00 UTC. Invetidos. Resources: 4, Destroy button.
- Segunda Trama**: Esta es la segunda trama de la red social. Date: 2016-01-01 00:00:00 UTC. Invetidos. Resources: 4, Add button.

 The right sidebar is identical to the previous screenshot, showing navigation links for NewsFeed, Character sheet, Unsolved plot, Resolved plot, Create Event, Friends, Followers, Find Friends, Edit Profile, and Change password.



Intermission System: gestión de usuarios para la simulación de líneas argumentales en eventos de interpretación libre

Para invertir tus recursos en una trama, has de ir a la trama y seleccionar el recurso que quieres gastar y añadirlo, si no dispones de suficiente recurso no lo podrás añadir. Si quisieras retirar un recurso de una puja, basta con que le des a destroy para recuperarlos.

The screenshot displays the Intermission web interface. At the top, a dark blue header contains the site name 'Intermission' and navigation links: 'Nombre', 'Who', 'Help', and 'Sign out'. The main content area is divided into three columns. The left column shows the user profile for 'Paco Raval', including a profile picture, clan name 'Brujah', generation '10', and various attributes like 'Physical', 'Social', 'Mental', 'Conscience/Conviction: 1', 'Self-Control/Instinct: 1', 'Courage: 1', 'humanidad:1', 'Willpower: 1', 'Blood pool: 1', and 'Health: 10'. The middle column, titled 'Resolved Plot', lists three completed tramas: 1) 'Primera Trama' (2016-08-04) with result 'Gane'; 2) 'Segunda Trama' (2016-01-01) with result 'perdi'; 3) '4 trama trama gorda' (2016-01-01) with result 'Aqui esta tu princesa'. The right column contains a sidebar with 'Advantages' (Recursos: 0/4, Sirvientes: 4/4) and a 'NewsFeed' menu with options like 'Character sheet', 'Unsolved plot', 'Resolved plot', 'Create Event', 'Friends', 'Followers', 'Find Friends', 'Edit Profile', and 'Change password'.

Cuando se resuelva la trama, recuperarás tus recursos para poder invertirlos en otros recursos y podremos ver los resultados en Resolved plot como vemos en la imagen de arriba.