

PLATAFORMA DE PERFILES OSUC

2 de septiembre del 2022



Diego Costa
Agustin Covarrubias
Fernando Smith



Índice de contenido

1. Introducción	3
2. Identidad	3
2.1. Criterios de membresía y permanencia	4
2.2. Roles	5
3. Automatización	6
4. Logros	6
4.1. Logro de Participación	7
4.2. Logro por Solicitud	7
4.3. Logros Secretos	7
5. Lógica de Servidor	8
6. Modelo entidad relación	8
7. Navegación	8
8. Diseño e interacción	10
8.1. Landing page	10
8.2. Mock-ups	10
8.2.1. Página de inicio	10
8.2.2. Barra de navegación	10
8.2.3. Login	10
8.2.4. Perfil	10
8.2.5. Vista de administrador	11
8.2.6. Vista de recursos humanos	11
8.2.7. Vista de logros y contadores	11
8.2.8. Vista de configuración de administración	11
8.2.9. Paleta de colores tentativa	11

Índice de figuras

1. Flujo de miembros	4
2. Ejemplos de insignias	6
3. Tipos de insignias	7
4. Paleta de colores	11



1. Introducción

Open Source UC es una comunidad estudiantil de apasionados por la programación y el software Open Source. Como parte del relanzamiento de la comunidad, se decidió desarrollar una plataforma de gestión comunitaria que pueda:

- Gestionar a los miembros de la comunidad y sus perfiles
- Automatizar procesos de recursos humanos dentro de la organización
- Otorgar, gestionar y visualizar un sistema de logros

Esta plataforma debe constituir un Single Source of Truth (SSOT) sobre la identidad de los miembros en la organización, permitiendo que la coordinación de Open Source UC utilice la plataforma a modo de sistema de gestión de recursos humanos.

2. Identidad

La plataforma de perfiles de Open Source UC debe cumplir el rol fundamental de un Sistema de Manejo de Identidad, por lo que:

- Debe funcionar como la única fuente de verdad sobre quiénes integran la organización y su rol dentro de ella.
- Debe tener la capacidad de autenticar a los usuarios directa o indirectamente.
- Debe tener la capacidad de autorizar a los usuarios a acceder distintos recursos dentro de la organización.

De estos requerimientos se desprenden ciertas decisiones de diseño:

- La plataforma debe ser diseñada con extensibilidad en mente, permitiendo que eventualmente otras aplicaciones o servicios usen la plataforma a modo de Single Sign-on (SSO).
- La plataforma usará GitHub mediante OAuth para autenticar a sus usuarios, dado que esta plataforma ya centraliza a nuestros usuarios.
- La plataforma deberá implementar un sistema de acceso basado en roles (RBAC), que permita autorizar el acceso a distintas funcionalidades de la plataforma.



2.1. Criterios de membresía y permanencia

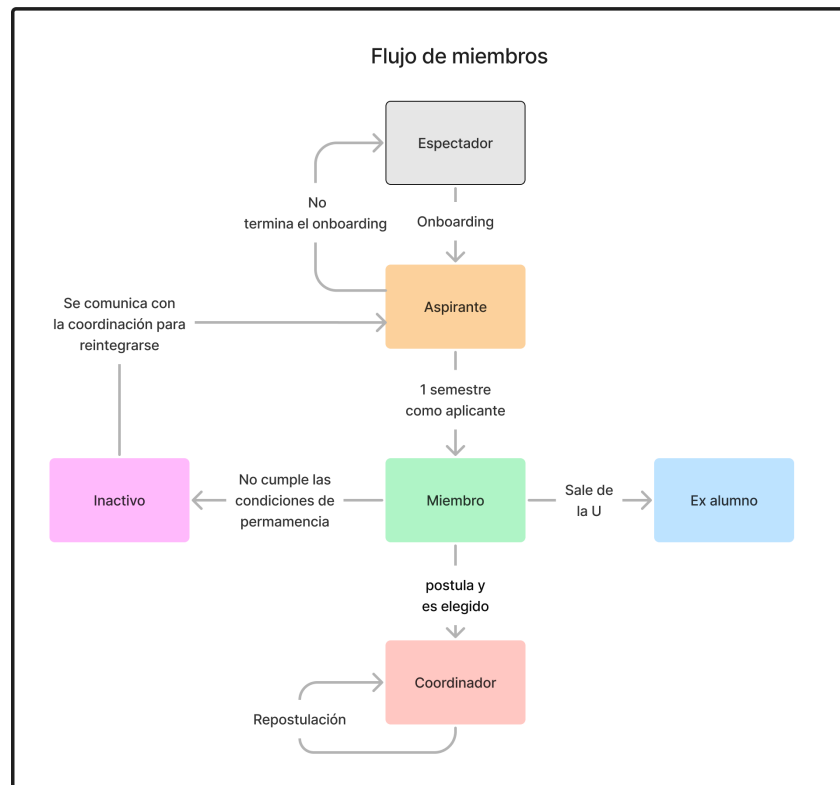


Imagen 1: Flujo de miembros

La idea de estos criterios es que no son fijos, sino adaptables para cada tipo de persona que ingrese al grupo¹. Por lo que pese a que se automaticen herramientas de medición, va a existir una supervisión manual para casos excepcionales².

Para ser considerado miembro o para cumplir los criterios de permanencia semestral, es necesario que se suban 3 niveles (aproximadamente 1500 puntos otorgados por las insignias).

Como cada insignia tiene puntos distintos, estos pueden ser algunos ejemplos de como acumular 1500 puntos:

¹La coordinación no es sujeta a los criterios de Permanencia

²Ejemplo: Se puede hacer una excepción a un usuario que participe muchísimo en un semestre y en el segundo no logro lo mínimo.



- Asistir a 1 o más talleres organizados por OSUC
- Haber realizado al menos una PR a un proyecto de OSUC.
- Haber participado en la mitad de las reuniones totales (Presenciales o Remotas) en las que se toma asistencia.
- Haber iniciado o convocado a uno o más proyectos/talleres que sea en nombre de OSUC.

Para poder hacer seguimiento a los criterios de membresía y permanencia, la plataforma deberá tener la capacidad de dinámicamente generar reportes identificando a los usuarios en incumplimiento de estos criterios.

2.2. Roles

Para simplificar el Sistema de Control de Acceso Basado en roles (RBAC), la estructura organizacional se mapea en tres distintos roles dentro del sistema:

- **Nivel 1 (Anónimo):**
 1. Puede navegar en los perfiles y descripciones de usuario.
 2. No puede registrarse ni iniciar sesión.
 3. No puede interactuar o modificar los perfiles.
 4. No tiene un perfil.
- **Nivel 2 (Usuario):**
 1. Puede navegar en los perfiles y descripciones de usuario.
 2. Puede iniciar sesión.
 3. Puede interactuar y modificar su perfil.
 4. Puede ganar insignias.
 5. No pueden registrarse.
- **Nivel 3 (Administrador):**
 1. Puede ganar, mirar, modificar y eliminar insignias propias y de todos los usuarios.
 2. Pueden mirar, ocultar y eliminar perfiles.
 3. Puede iniciar sesión.
 4. Puede modificar su propio perfil.
 5. Puede ver, crear, eliminar y editar insignias.
 6. Puede generar links de actividad (Se menciona en el apartado de [logros](#)).
 7. Pueden registrar a otros usuarios y definir cualquier rol.

De esta forma, los distintos roles orgánicos podrán ser enlazados directamente al sistema por correspondencia:

- **Espectador:** Nivel 1
- **Inactivo:** Nivel 2 - Se ocultarán sus perfiles
- **Aspirante:** Nivel 2
- **Miembro:** Nivel 2
- **Mentor:** Nivel 2



- **Exalumnos:** Nivel 2 - Aparecerán en un apartado especial.
- **Coordinación:** Nivel 3

3. Automatización

La comunidad sostiene una serie de procesos de entrada y de salida que han complicado significativamente el trabajo para la coordinación. Para facilitar este proceso, se propone diseñar integraciones con los siguientes servicios:

- **GitHub** Usando la API de **Organizations API** de GitHub para controlar y administrar la organización de **Open Source UC**, la página podrá gestionar los distintos permisos a repositorios y teams que se manejan en GitHub.
- **Telegram** Al tener una base de datos con los usuarios, se espera usar mediante la **Bots API**, para administrar automáticamente el ingreso, salida y asignación de cargos de miembros.
- **ForwardEmail** Se espera integrar un sistema de correos @osuc.dev, generando y eliminando alias utilizando la API de **ForwardEmail**.

4. Logros

La cultura de la comunidad se basa en tres pilares fundamentales. Los **proyectos**, la **interacción social** y la **educación**.

Para recompensar la proactividad dentro de nuestra comunidad, se utilizará un sistema de recompensas inspirado en las insignias de Boy Scouts.

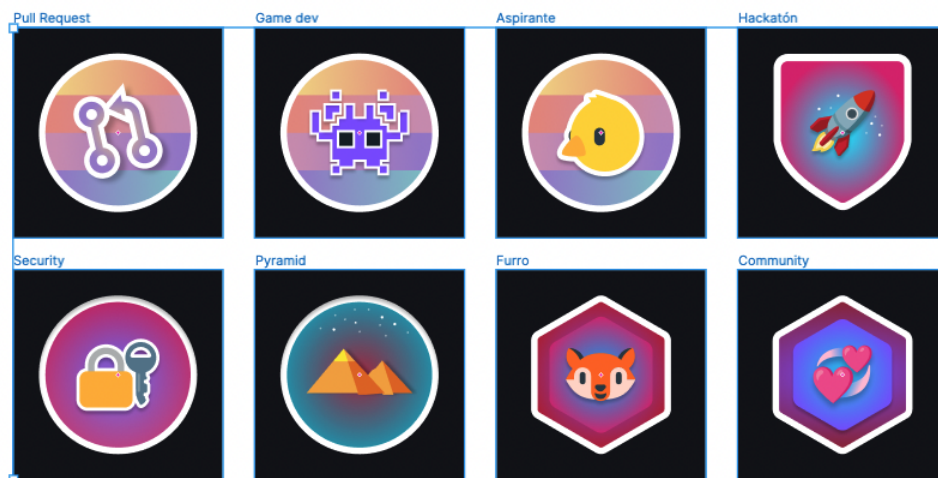


Imagen 2: Ejemplos de insignias

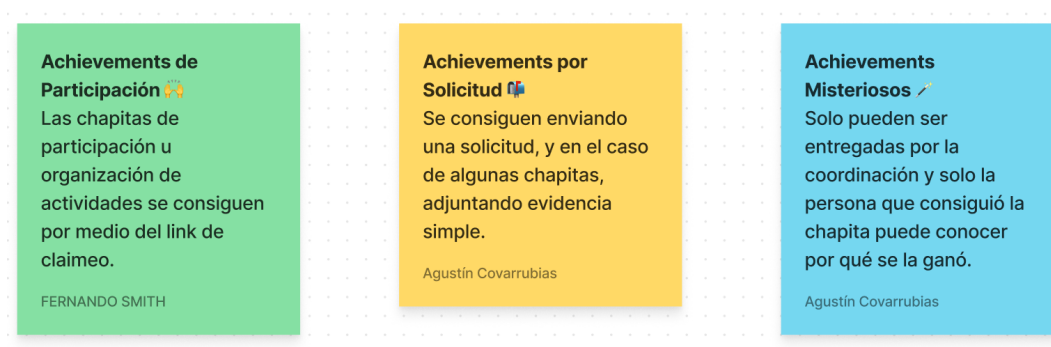


Imagen 3: Tipos de insignias

Como se puede apreciar en las imágenes, existirán 3 tipos distintos de logros, distinguibles visualmente por la figura geométrica de la insignia.

4.1. Logro de Participación

Un usuario podrá obtener un **Logro de Participación** accediendo a un link secreto autogenerado, cuando un Administrador genere la URI en uno de los [paneles administrativos](#). > **Ejemplo de uso** La Coordinadora Bárbara Irrarazaval organiza un taller de Julia para la comunidad de estudiantes. Para recompensar a los miembros que colaboraron, genera una URI y la envía al grupo de Telegram de la actividad, donde cada usuario durante 24 horas tiene la oportunidad de clickearlo y adjudicarse la insignia correspondiente.

4.2. Logro por Solicitud

Un usuario podrá pedir en cualquier momento un **Logro por Solicitud** mediante un botón en la página de insignias. Algunos de estos serán entregados instantáneamente, mientras que las insignias que la coordinación considere de mayor prestigio deberán ser aprobados por un Administrador en uno de los paneles administrativos. Al solicitar un logro protegido, el usuario será capaz de introducir la evidencia mediante texto y URL's en un cuadro de texto que aparecerá al pulsar el botón.

Ejemplo de uso Hernán Cabrera invitó a tres amigos a OSUC. Quiere obtener la insignia “Estafa Piramidal”, cuyo requisito acaba de cumplir. Para esto, debe ir a la página de insignias, clickear para obtenerla, y en el cuadro de prueba escribe los usuarios de GitHub de sus amigos.

4.3. Logros Secretos

Un administrador podrá darle a un usuario de su elección un **Logro Secreto**. Al hacer *hover* sobre estas insignias se debe mostrar su título solamente.

Ejemplo de uso La usuaria Valeria Riquelme utiliza una distribución de GNU/Linux como su sistema operativo principal. Como sabe que esta proeza no es una tarea fácil, decide presumirlo



en el grupo general de OSUC. En respuesta a esto, un coordinador le otorga la insignia secreta “Linus Torvalds”

5. Lógica de Servidor

Algunos de los procesos más importantes del lado del servidor son los siguientes:

- Proceso para incorporar nuevos usuarios en la lista de aspirantes
- Proceso para modificar los roles de usuarios
- Proceso para la asignación manual de logros secretos
- Proceso para la aprobación o rechazo de solicitudes de logros
- Generación de URLs para la obtención de logros de participación
- Proceso de autenticación de usuarios usando OAuth de GitHub
- Adición de integrantes a la organización de GitHub mediante la Organizations API
- Adición de integrantes a los grupos de Telegram mediante la Bots API
- Creación de alias de mails @osuc.dev vía la API de ForwardEmail
- Autorización de usuarios en base al sistema de controles de acceso

6. Modelo entidad relación

Se puede ver el modelo E/R [aquí](#) o adjunto como PDF de nombre “Diagrama E_R” en los anexos subidos en el repositorio.

7. Navegación

Debido a que los niveles son jerárquicos, se considera que cada nivel puede realizar todas las acciones de sus niveles inferiores.

1. El usuario no ha iniciado sesión o no tiene cuenta ([Nivel 1](#))
 - 1.1 El usuario puede navegar por la página de inicio
 - 1.2 El usuario puede entrar a la página de Logros
 - 1.3 El usuario puede buscar un perfil utilizando el *input* de texto en la página principal, al darle click a “Buscar”
 - 1.4 El usuario puede abrir un perfil
 - 1.4.1 El usuario puede ver detalles de las insignias al hacer *hover* con el mouse
 - 1.5 El usuario puede iniciar sesión con el OAuth de GitHub
2. El usuario tiene su sesión iniciada ([Nivel 2](#))
 - 2.1 El usuario no tiene su perfil creado
 - 2.1.1 El usuario ve un mensaje por defecto con las instrucciones para vincular el perfil de la página con un repositorio específico personal



2.2 El usuario tiene un perfil ya creado

2.2.1 Puede modificar su propio perfil (mediante GitHub)

2.2.2 El usuario puede ver su árbol de logros

2.2.2.1 El usuario puede pulsar un botón en un Logro por Solicitud para obtenerlo.

Si el logro está protegido, se abrirá un cuadro con un *input* de texto para justificar la obtención, reporte que será enviado al panel de Administración. Luego el botón queda como no seleccionable.

Si el logro no está protegido, se actualizará en la base de datos la obtención del logro y se verá reflejado en el perfil del usuario, y en la desaparición del botón de obtención en el árbol de logros

2.3 El usuario puede cerrar sesión

3. El usuario tiene su sesión iniciada (Nivel 3)

3.1 Podrá acceder a un apartado de “admin” en la barra de navegación

3.1.1 Podrá acceder a una vista de administración, donde verá un registro administrativo de actividad reciente de usuarios y administradores

3.1.2 Podrá acceder a una página de recursos humanos

3.1.2.1 Podrá buscar miembros mediante un *input* de texto y un botón de “Buscar”. Además, habrá un botón al final de cada usuario. Se abrirá un recuadro donde se puede cambiar el rol del usuario (nivel administrativo) y su estado de actividad.

3.1.2.2 Podrá generar reportes de actividad.

El reporte de aplicantes genera y descarga una lista en formato .csv con los **aplicantes** que no están cumpliendo con los criterios de membresía (Una cantidad de puntos arbitraria decidida en la configuración general).

El reporte de outboarding genera y descarga una lista en formato .csv con los **miembros** que no están cumpliendo con los requisitos de permanencia (Una cantidad de puntos arbitraria decidida en la configuración general)

3.1.2.3 Podrá importar aplicantes mediante un drag and drop o click para seleccionar un archivo. Debe subirse un .csv con un formato especificado que contenga el Telegram, Discord, GitHub y alias de mail deseada por el usuario

3.1.2.4 Podrá generar un cambio de coordinación. La idea de esta función es ser utilizada después de que las elecciones de la comunidad definan nuevas coordinaciones. Este botón permitirá escribir qué usuarios deben tener un **Nivel 3** de acceso.

3.1.2.5 Podrá reiniciar el onboarding. Esto significa promover automáticamente a los aplicantes que cumplen con los criterios, y dejar como Inactivos a los aplicantes que no lo cumplen.

3.1.3 Podrá acceder a una página de logros y contadores. Aquí se podrán crear logros nuevos, y generar links con UUID's para la obtención de Logros de Participación.

3.1.4 Podrá acceder a una página de configuración

3.1.4.1 Se podrá cambiar las variables de integración (API keys) y apretar guardar.

3.1.4.2 Se podrá cambiar los parámetros de los criterios de membresía y permanencia.

Como último detalle, el hecho de que un usuario esté “inactivo” en términos de la página solamnete



significa no aparecer en el listado general de miembros de la Landing Page.

8. Diseño e interacción

8.1. Landing page

Con el uso de Figma, se creó una página de inicio como prototipo en el siguiente [link](#)

8.2. Mock-ups

Con el objetivo de diseñar la plataforma, se creó un prototipo interactivo en base a los mock-ups diseñados en Figma. Este puede ser probado [aquí](#), aunque es importante ajustar el tamaño a la pantalla (fit to screen) apretando la “z” en el teclado.

8.2.1. Página de inicio

Esta vista poseerá una barra de navegación que contendrá el logo de la organización, una ventana para acceder a la ventana de insignias y otra para iniciar sesión. Luego existirá un apartado explicativo, para finalmente mostrar un menú en el cual aparecerán las miniaturas de los perfiles de coordinadores, miembros, aspirantes y un apartado especial para mostrar a los exalumnos que han contribuido significativamente.

8.2.2. Barra de navegación

La barra de navegación desplegada para usuarios que inicien sesión desde [Nivel 2](#), podrán observar un apartado para ir hacia su propio perfil, un panel de administración (Si se poseen los permisos) y uno para cerrar sesión.

8.2.3. Login

En esta página al usuario se le redireccionará a la autenticación de GitHub.

8.2.4. Perfil

Cada usuario sobre el [Nivel 2](#) poseerá un perfil en una página dinámica en el cual se mostrara su nombre, su fecha de ingreso, su nivel sobre la base del puntaje obtenido de las insignias, las insignias principales³ y una descripción.

En caso de que el usuario no tenga configurado su perfil en GitHub se mostrara un mensaje predefinido con instrucciones.

³Las insignias estarán en un menú desplegable si superan el máximo y tendrán una pequeña descripción al pasar por encima



8.2.5. Vista de administrador

Al seleccionar la vista de “admin” de la barra de navegación, el usuario podrá ver un historial de cambios realizados por los miembros registrados. Además, se despliega una barra lateral que les permite acceder a nuevas ventanas.

8.2.6. Vista de recursos humanos

En esta página se hará un resumen de miembros (visualizando una tabla de información), un apartado para generar reportes de actividad, un apartado para importar un archivo e importar usuarios y una zona de acciones que requieren una doble verificación que podrán ser definidas en un futuro.

8.2.7. Vista de logros y contadores

En esta página se mostrará un resumen de la base de datos, un apartado para administrar solicitudes, una para generar enlaces de logros (URI) y otra para definir nuevas insignias.

8.2.8. Vista de configuración de administración

En esta última ventana se podrán hacer modificaciones de parámetros que utilizara el servidor para distintas acciones.

8.2.9. Paleta de colores tentativa

Como una paleta tentativa se usó la página [colors](#) para definir los colores usados en la página.

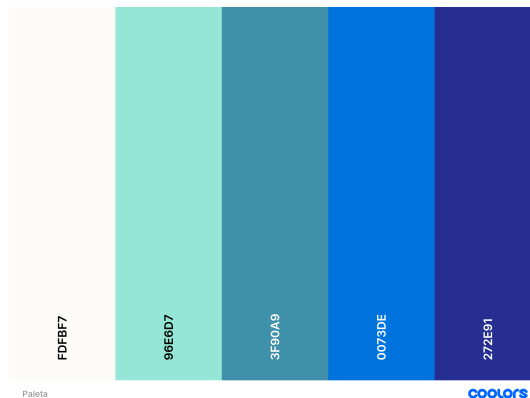


Imagen 4: Paleta de colores