

Sistemas de Información

Trabajo colaborativo con Slack



Componentes:

- Julián Lara Peso (i42lapej@uco.es)
- David Llagas Rodríguez (i42llrod@uco.es)
- Javier Ibañez Coca (i42ibcoj@uco.es)
- Víctor Jesús Castañeda Morales (i42camov@uco.es)

[ENTRAR SLACK](#)



UNIVERSIDAD
DE
CÓRDOBA

Índice:

EJERCICIO 1: Slack.....	3
Enunciado.....	3
Canales creados.....	3
EJERCICIO 2: Slack-Trello (VICTOR).....	6
Enunciado.....	6
Inserción de Trello.....	7
Comandos Trello.....	10
EJERCICIO 3: Slack-Taiga (JULIAN).....	14
Enunciado.....	14
Integración de Taiga.....	14
Ejemplos Prácticos.....	15
a) Crear una User Story (US) en Taiga.....	15
b) Agregar una tarea a la anterior US.....	16
c) Crea otra tarea en la misma US y borrala.....	17
d) Busque información sobre los Epics y sus diferencias con los US. Active los epics en los módulos de Taiga.....	17
EJERCICIO 4: Slack-Github (DAVID).....	19
Enunciado.....	19
Crear el repositorio de GitHub.....	19
EJERCICIO 5: Slack-... (JAVI).....	23
Enunciado.....	23
Conclusión.....	34
Bibliografía.....	34

EJERCICIO 1: Slack

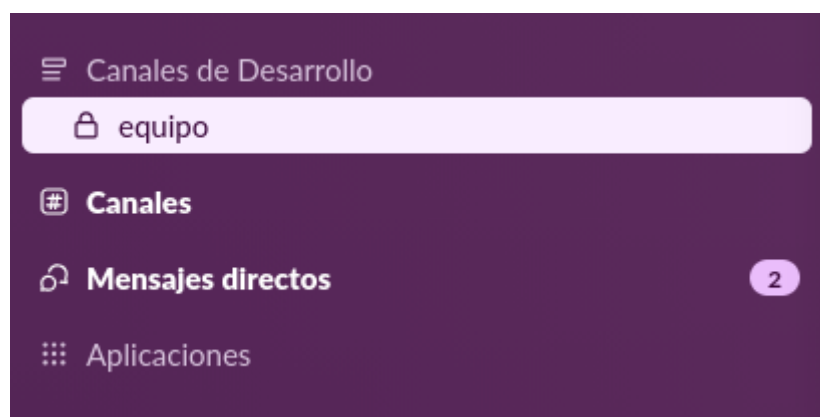
Enunciado

Imagina que has creado tu empresa y necesitas crear distintos canales de comunicación:

- a) Crea los canales públicos y privados que consideres necesarios para un proyecto de desarrollo web.
- b) Invita a los miembros de tu grupo según el perfil de desarrollador.

Canales creados

Para organizar el Slack de manera eficiente y organizada, empezamos dividiendo nuestro Slack en 2 secciones en las que se distribuyen todos los canales necesarios. El primero es el llamado “Canales de Desarrollo” en el que implementaremos todos los canales que estén relacionados con la implementación de la aplicación (frontend, backend o UI/UX) y el segundo es el de “Canales” que es general. En este encontramos canales que tratan temas importantes, pero no lo suficiente como para llegar a tener una sección propia (reuniones, marketing o soporte).

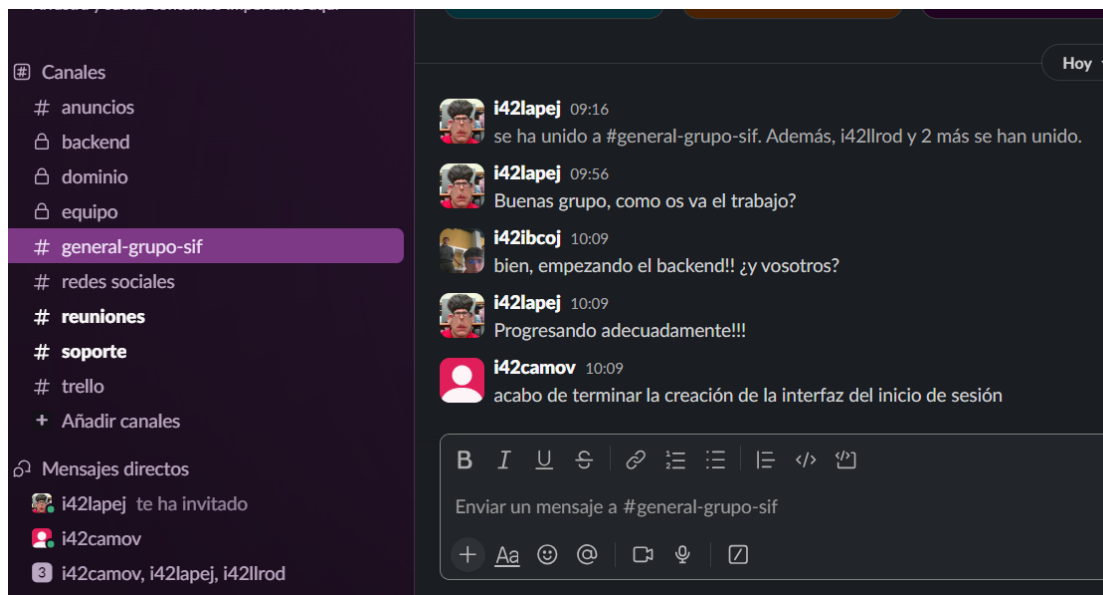


La primera sección de “Canales en Desarrollo” se centra en los canales necesarios para la implementación del proyecto. Hemos creado varios canales en esta sección:

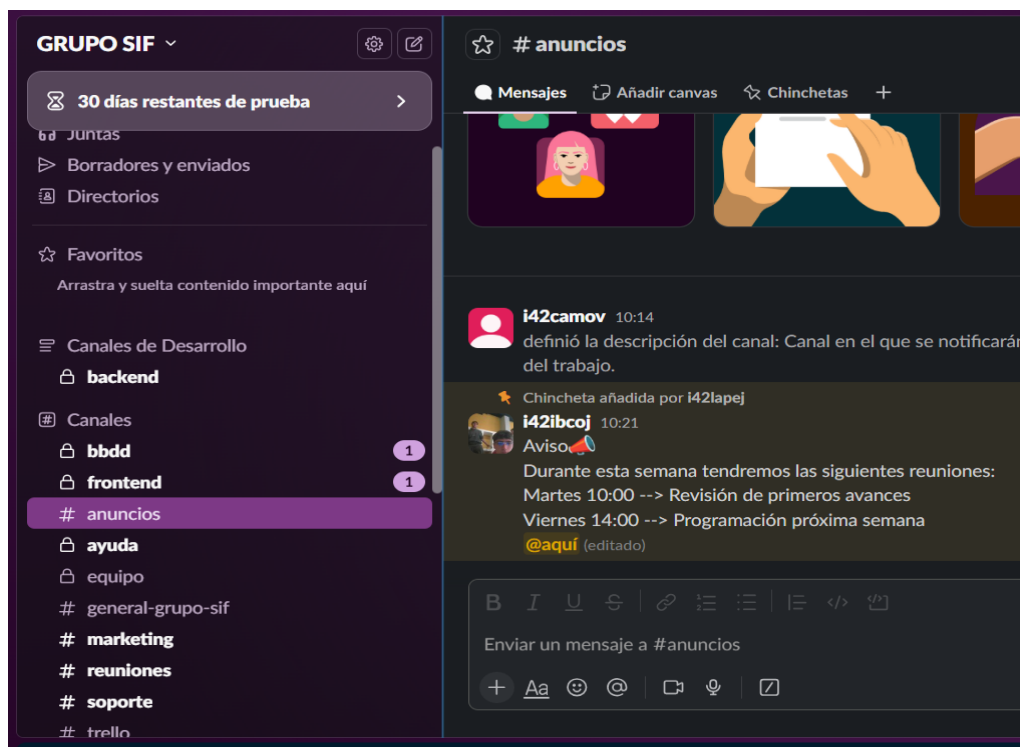
- ayuda: En este canal implementamos
- backend:
- frontend:
- design-ui/ux:
- bbdd:
- equipo:

c) Envía mensajes entre los miembros de tu equipo en canales públicos y privados. Añade archivos adjuntos en alguno de ellos.

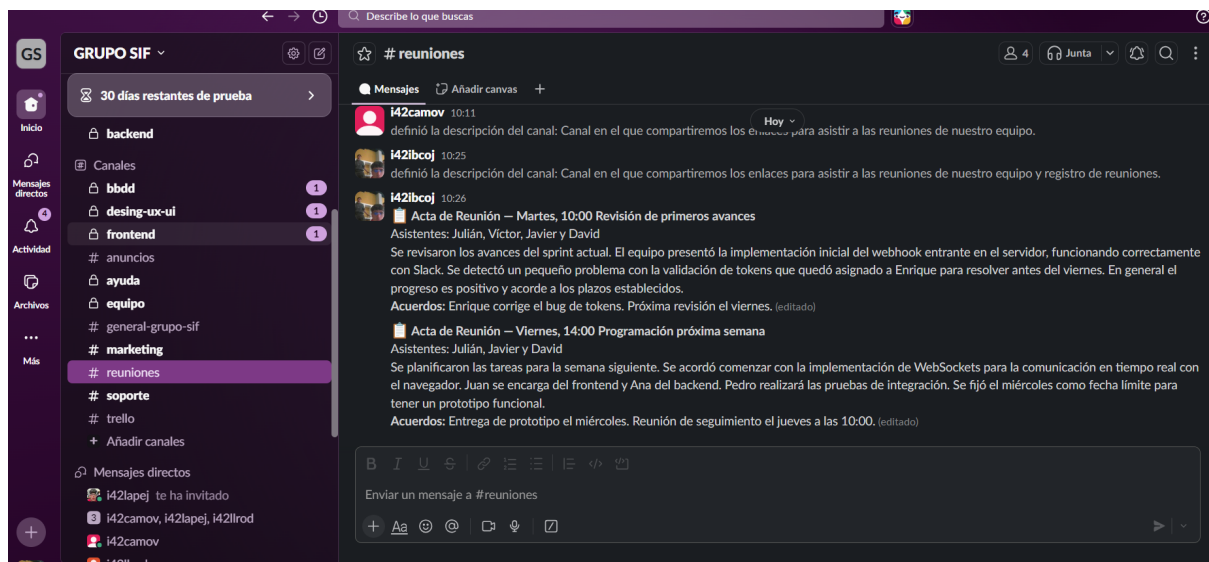
Hemos empezado a comunicarnos por los canales de comunicación correspondientes, en primer lugar por **#general-grupo-sif**



En segundo lugar por el canal de **#avisos**, se han comunicado las reuniones de esta semana

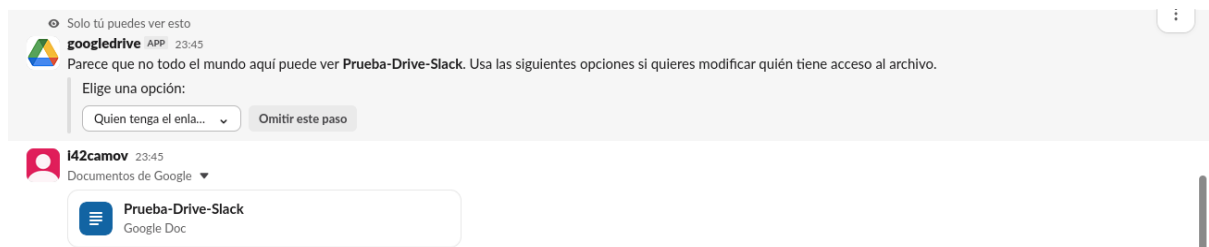
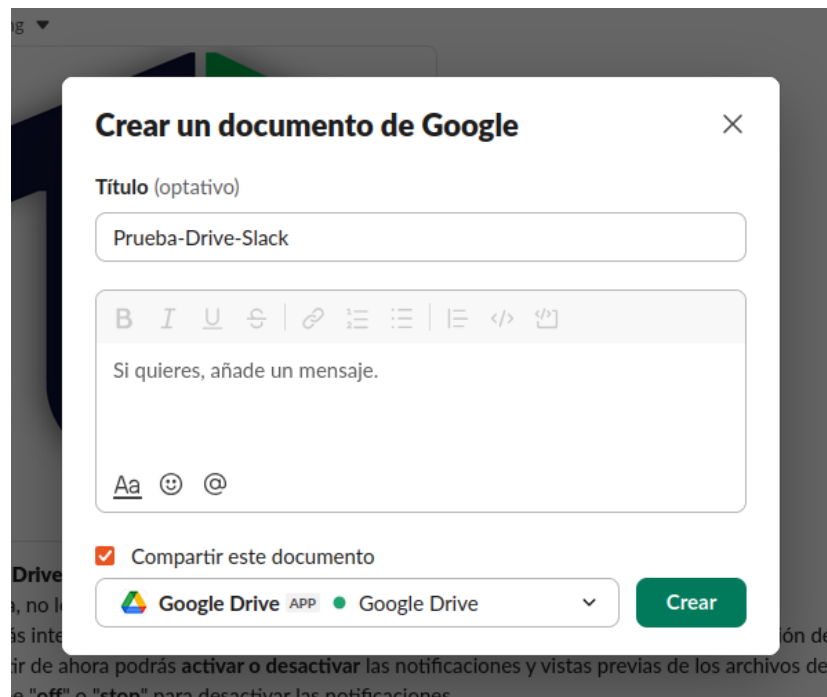


Tras las reuniones, en el canal **#reuniones**, se han documentado el contenido de estas para los que necesiten consultar algo, o los que no hayan podido asistir



d) Haz una integración en Slack de la herramienta Google Drive de la empresa y comprueba su funcionamiento subiendo un documento a la nube.

Para subir un archivo a Google Drive, lo creamos poniendo / y aparece un menú que pone “Crear archivo” con el logo de Google Drive al lado. Pulsamos y aparece esta ventana, hay que rellenarla y automáticamente se subirá el archivo a Drive como vemos en la segunda imagen (también pregunta quién puede acceder al archivo y con qué permisos).



EJERCICIO 2: Slack-Trello (VICTOR)

Enunciado

1. Crear un canal en Slack denominado “trello” donde se mostrarán las notificaciones de los cambios que se produzcan en el tablero de Trello de la práctica anterior.
2. Añade la aplicación Trello en Slack, para poder ejecutar comandos desde

Slack que interactúen con el tablero de Trello.

3. Integrar Slack desde Trello, indicando el canal de Slack (#trello) donde se mostrarán todas las notificaciones de cambio del estado del tablero. Haz cambios en el tablero para verificar que estos cambios se han notificado correctamente en el canal de slack.
4. Estudiar y recopilar en una tabla, que se agregará al informe de la práctica los comandos para interactuar con trello desde slack.
5. Utilizando los comandos anteriores realizar los siguientes ejercicios:
 - a) Ver en slack la información del tablero
 - b) Crear en la lista N° 1 la tarea “Integración Trello-Slack”.
 - c) Asigna dos miembros de tu equipo a la tarjeta anterior e indica como fecha de vencimiento de esta tarea, el día 6 de Marzo.
 - d) Crear en la lista N° 2 la tarea “Integración Taiga-Slack” y asignarle como fecha de vencimiento el día 7 de Marzo a las 23:00 horas.

Inserción de Trello

Para comenzar hemos creado un canal en nuestra sección de “Canales” que se llama #trello. Esta la usaremos para ejecutar desde ahí todos los comandos necesarios e interactuar con Trello. Posteriormente insertamos en nuestro Slack la aplicación de Trello, que se puede hacer por la ruta de: Mas > Herramientas > Aplicaciones > Trello. Al entrar se pide una cuenta de Trello con la que vincular, en este caso elegimos la mía, Víctor.



¿Te gustaría permitir que la siguiente
aplicación acceda a tu cuenta?

Trello Slack App



Has iniciado sesión como:

Nombre

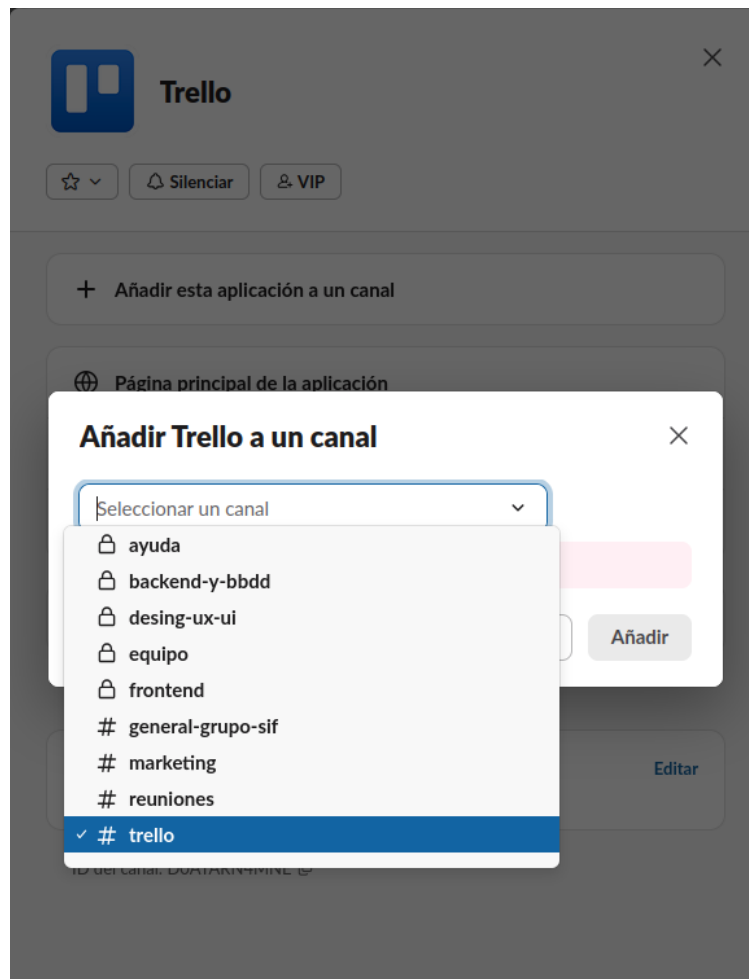
Víctor Jesús Castañeda Morales

Nombre de usuario

victorjesuscastanedamorales

[Cambiar de cuenta](#)

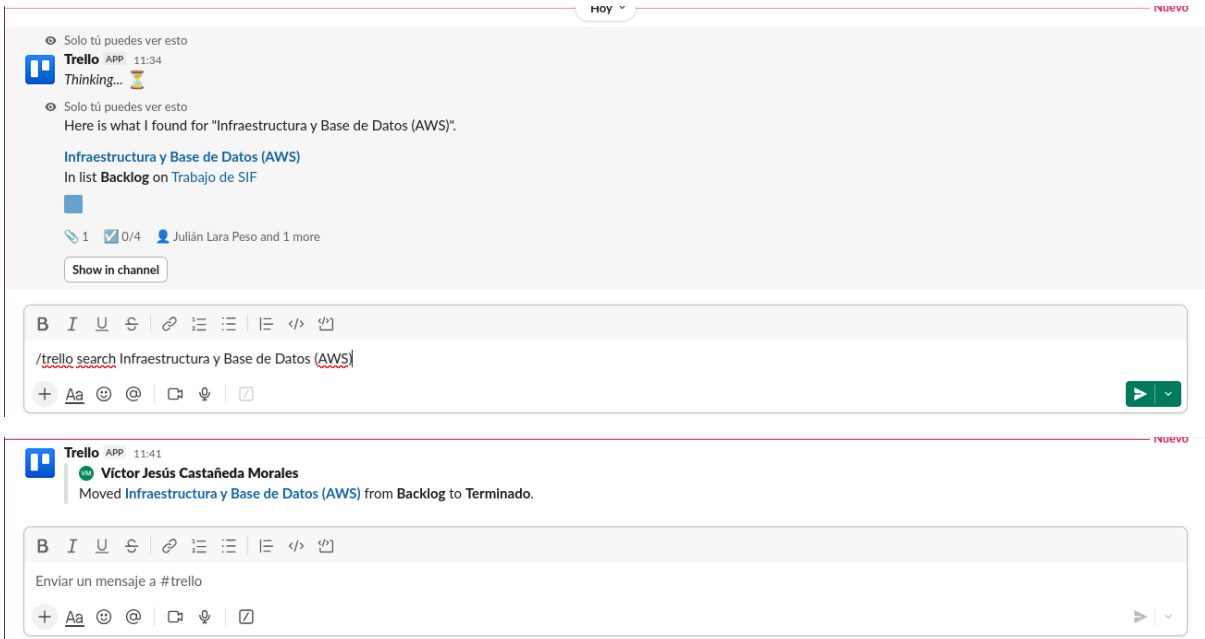
Una vez conectamos el Trello con nuestro Slack, tenemos que decirle al Trello de alguna forma que queremos que se ejecuten los comandos desde el canal de #trello ya creado. Eso se hace desde la aplicación de Trello, pulsamos en los 3 puntos, Abrir los detalles de la aplicación > Añadir esta aplicación a un canal y buscamos el canal de #trello.



Una vez vinculado, al ejecutar un comando te piden que conectes el canal a un tablero que tengas en tu Trello, si no no sabe a que tablero añadir tareas. Para ello te pone un enlace (<https://trello.com/platforms/slack>) desde el que puedes elegir que tablero quieres conectar.



Pulsamos en “Añadir a Slack” y y permitimos que Slack acceda a todo el tablero (en caso de que tengamos mas de un Slack eliges el que quieres conectar). Ahora ya tenemos configurado el Slack con Trello de forma que, ejecutando comandos desde el canal de #trello, podemos ver, añadir y modificar las tareas de nuestro panel de Trello y en caso de que hagamos algún cambio en el Trello, también se ven reflejadas.



Comandos Trello

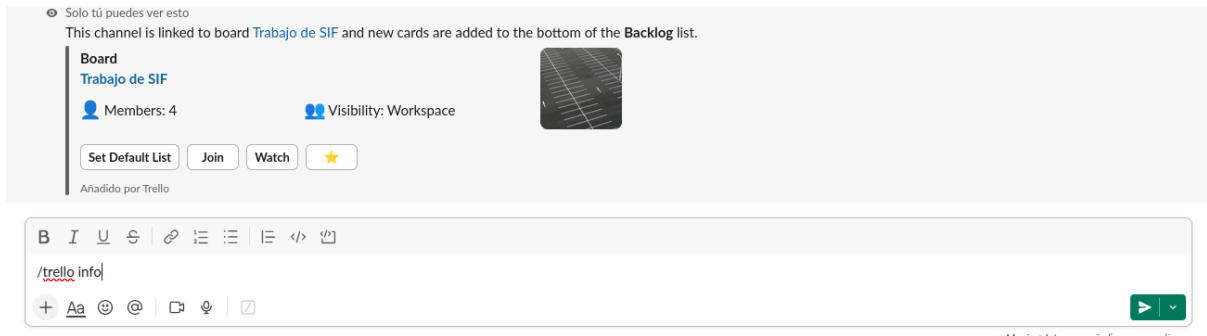
Ahora vamos a mostrar una tabla que muestra todos los comandos que se pueden utilizar para comunicarse con nuestro Trello desde Slack:

Comando	Función
TARJETAS	
/trello create	Crear una tarjeta
/trello search	Buscar tarjetas para mostrar en Slack
TABLEROS	

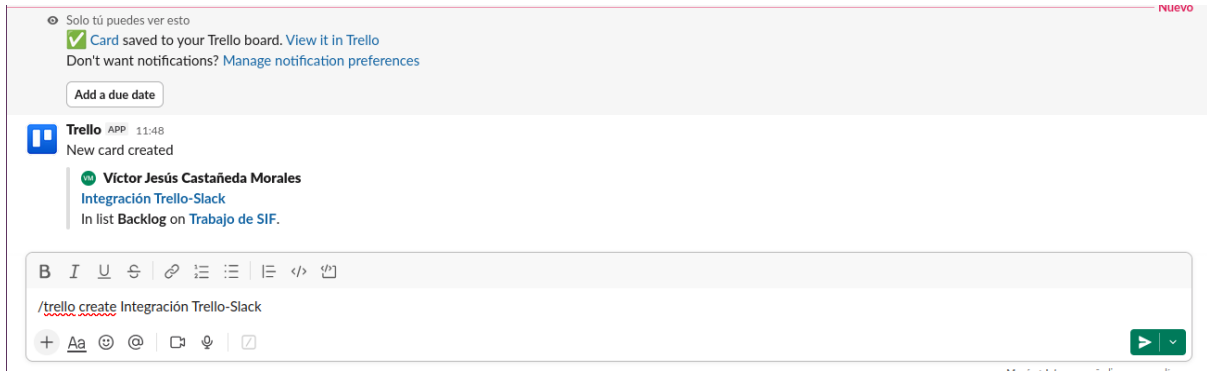
/trello link <board-url>	Vincular un tablero de Trello al canal actual de Slack
/trello info	Ver la información del tablero vinculado al canal
/trello set-list	Cambiar la lista donde se añaden las tarjetas
/trello unlink	Desvincula un tablero de Trello vinculado
CONFIGURACIÓN	
/trello login	Conectar la cuenta de Trello con la cuenta de Slack
/trello logout	Desconectar y desautorizar el miembro vinculado de Trello
/trello previews	Desactivar/activar las vistas previas de enlaces de Trello
/trello unlink-team	Desvincula un espacio de trabajo de Trello vinculado
/trello whoami	Verificar qué usuario de Trello tienes iniciado
/trello saveforlater	Alternar tu estado de participación en "Guardar para más tarde"
/trello ai	Alternar tu estado de participación en resumen de IA
/trello notifications	Alternar tu estado de participación en notificaciones de Bandeja de entrada
MISCELÁNEA	
/trello feedback	Enviar comentarios al equipo de Trello
/trello help	Listar todos los comandos disponibles

Por último, vamos a ejecutar el conjunto de acciones que se nos pide:

- Ver en Slack la información del tablero

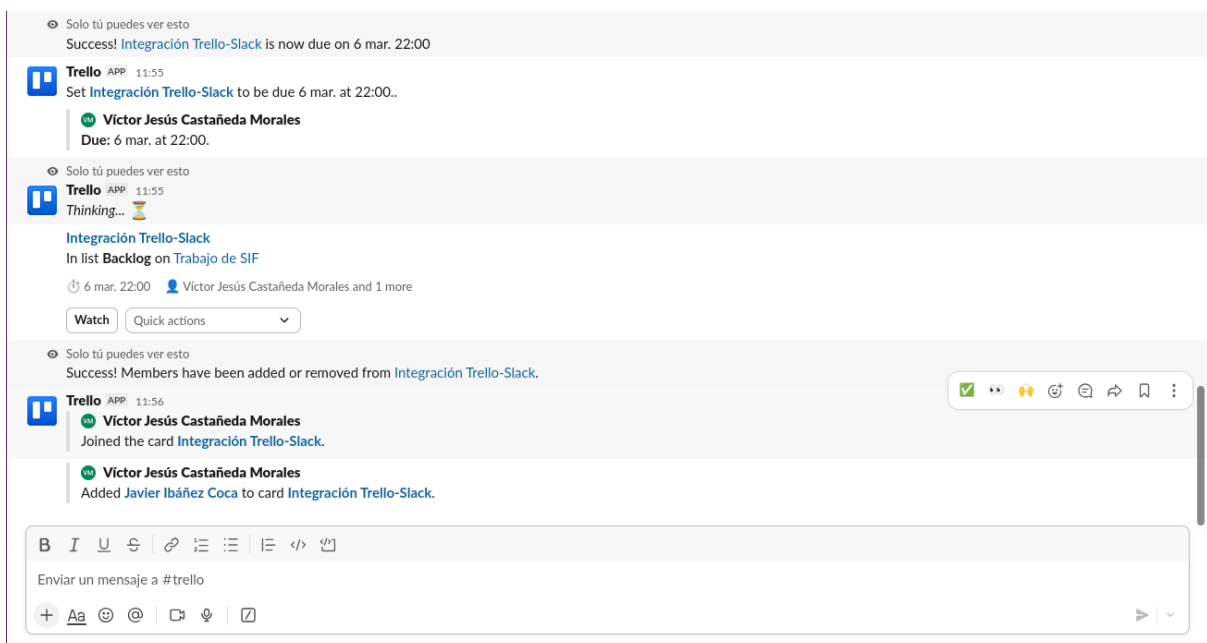


- Crear en la lista N°1 la tarea “Integración Trello-Slack”



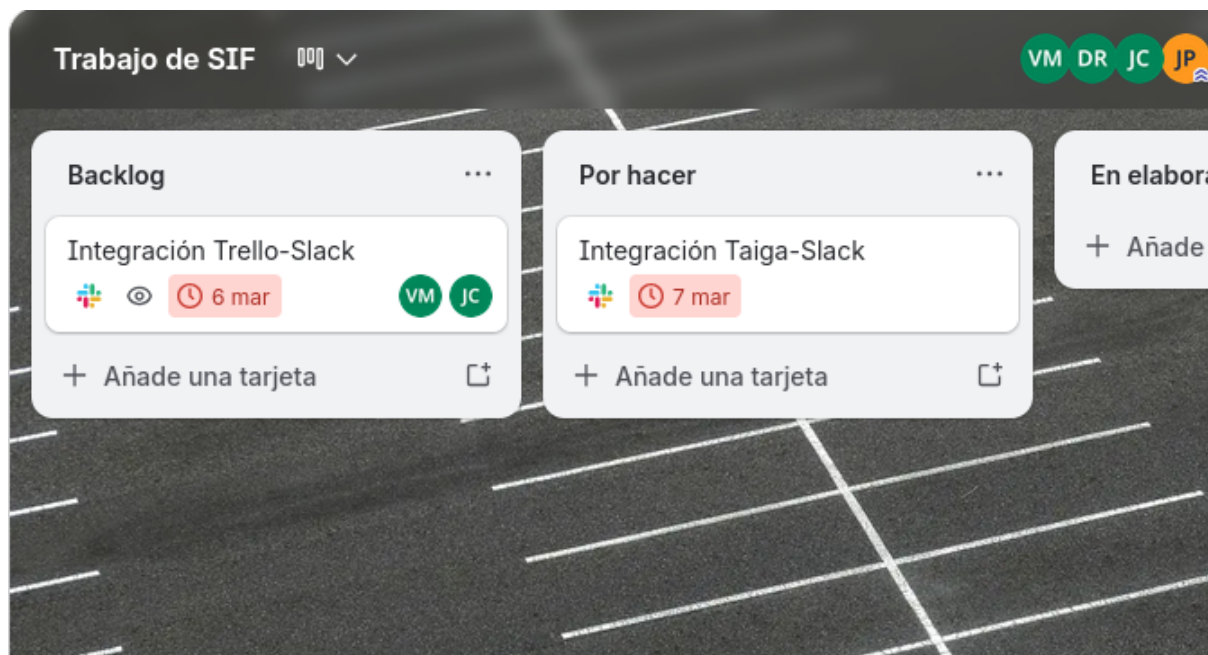
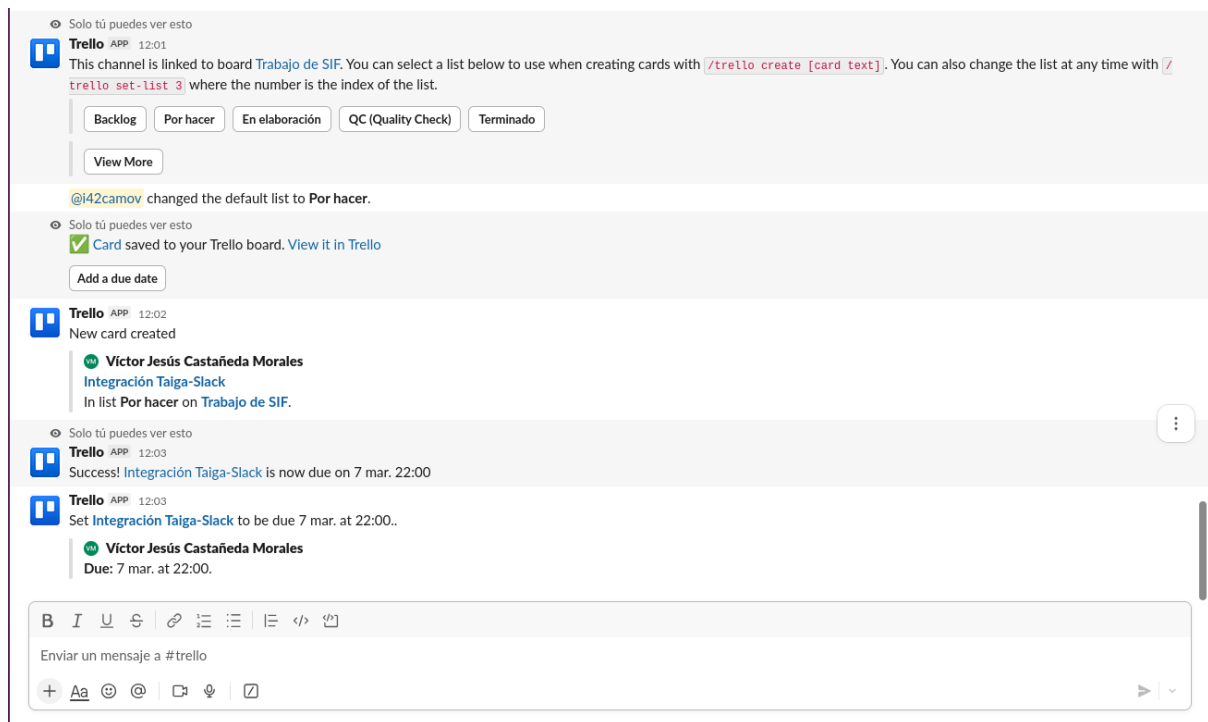
- Asigna dos miembros de tu equipo a la tarjeta anterior e indica como fecha de vencimiento de esta tarea, el día 6 de Marzo.

Esto se realiza mediante el comando /trello search. Cuando lo ejecutas, pulsas en “Show in channel”, aparece un desplegable de “Quick actions” y ahí puedes modificarlo



- Crear en la lista N° 2 la tarea “Integración Taiga-Slack” y asignarle como fecha de vencimiento el día 7 de Marzo a las 23:00 horas.

Para meterlo dentro de la lista N°2, tenemos que ejecutar el comando /trello set-list Por hacer. Ahí aparecen unos botones, uno por cada columna y hay que elegir cuál queremos (Por hacer) para que las tareas introducidas entren en esa lista. Al crear la tarea, aparece “Add due date” y ponemos la fecha indicada.



EJERCICIO 3: Slack-Taiga (JULIAN)

Enunciado

1. La integración de Slack con Taiga se realiza a través de los Webhooks. Investigue en qué consiste este término y agregue este estudio al informe de la práctica.
2. Crea un canal de Slack llamado “taiga” y configura la integración con el proyecto de Taiga creado en la práctica anterior utilizando webhooks. Realiza los siguientes ejercicios:
 - a) Crear una User Story (US) en Taiga y comprueba cómo se ha reflejado este cambio en el canal #taiga.
 - b) Dentro de la US anterior agregar una tarea que además tenga la siguiente descripción: Esta es una tarea de prueba y tiene un tiempo de @timetracking 2:00.
 - c) Crea otra tarea en la misma US y bórrala comprobando cómo se notifican en slack dichos cambios.
 - d) Busque información sobre los Epics y sus diferencias con los US. Active los epics en los módulos de Taiga y compruebe si los cambios en los epics se reflejan en Slack. Agregue todos estos resultados al informe de la práctica.

Integración de Taiga

Lo primero que debemos hacer, es buscar la aplicación “**Incoming Webhooks**”, en el apartado de aplicaciones de Slack. Al instalarla, podremos coger un enlace para pegar en otra web, siendo en este caso Taiga. Además, debemos configurar en que canal saldrán las actualizaciones de esta herramienta. En nuestro caso, en el canal #taiga.

En Taiga, vamos a los ajustes del proyecto y buscamos el apartado de plugins. Dentro de él, nos saldrá para poner un enlace en el apartado “Slack webhook url”. Pegamos y guardamos el proyecto.

Slack

Slack webhook url

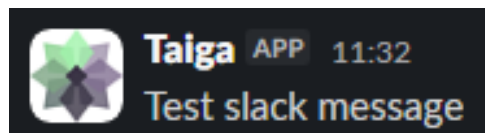
<https://hooks.slack.com/services/T0ARY2U5E7P/B0A5VKLSNTZ/U0DrDeU>

TEST

Slack webhook channel

#slackchannel

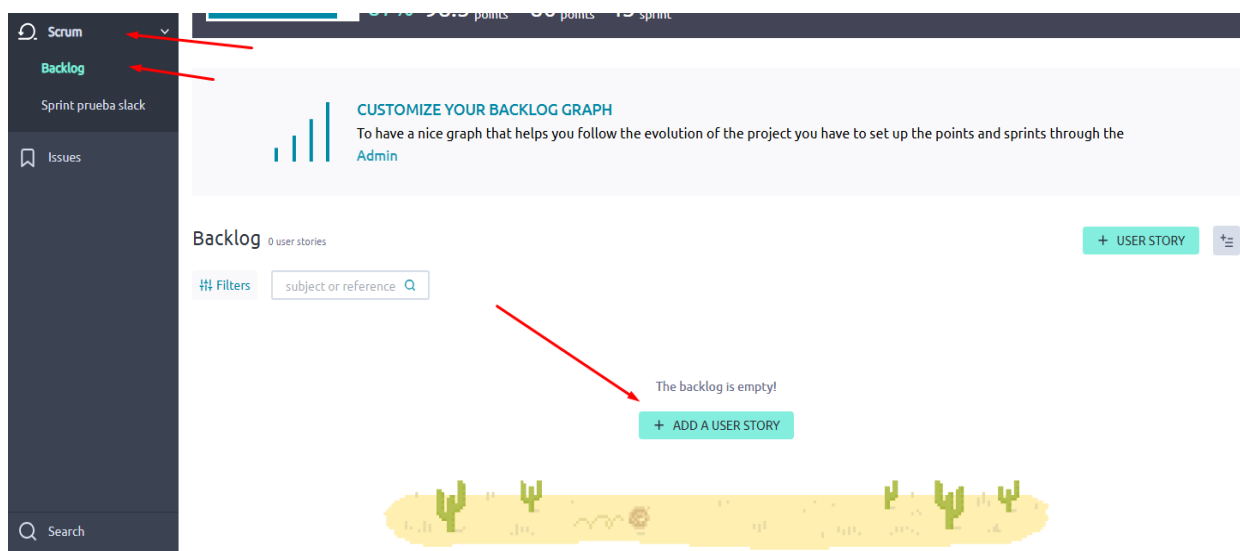
Tras esto, pulsamos el botón de “TEST” para verificar que todo esté bien conectado. Este el mensaje que nos saltará en el canal:



Ejemplos Prácticos

a) Crear una User Story (US) en Taiga

Crear un US es muy sencillo, ya lo hemos hecho en la práctica anterior, pero explicaré los pasos detalladamente. Primero, vamos al apartado de “Backlog” dentro de Scrum y le damos a “ADD A USER STORY”.



Creo una historia de usuario cualquiera y este cambio se verá reflejado en el canal seleccionado, indicando nombre, descripción y creador de esta.

New user story

OUR BACKLOG: Prueba para Slack

[Add tag +](#)

Esto es una prueba para comprobar el buen funcionamiento del Webhook de Slack

The backlog is empty!

0 Attachments +

Drop attachments here!

In progress ▼

LOCATION

☒ at the bottom ☐ on top

Julian Lara Peso

+ Add assigned + USE

POINTS

UX	1
Design	40
Front	3
Back	2
total points	46

🕒
👤
📁
🔒

CREATE

Taiga (Julian Lara Peso) APP 11:34

New User Story created: [#57: Prueba para Slack](#)

Creator

Julian Lara Peso

Description

Esto es una prueba para comprobar el buen funcionamiento del Webhook de Slack

b) Agregar una tarea a la anterior US

Para ello, creé un sprint de prueba y añadí nuestra US recién creada a este sprint. Ahora, dentro del apartado del propio sprint, pulsaremos el boton de + en US, y se nos abrirá este panel. Después de completar con la información requerida, veremos el cambio reflejado en nuestro canal.

Tarea prueba Slack

[Add tag +](#)

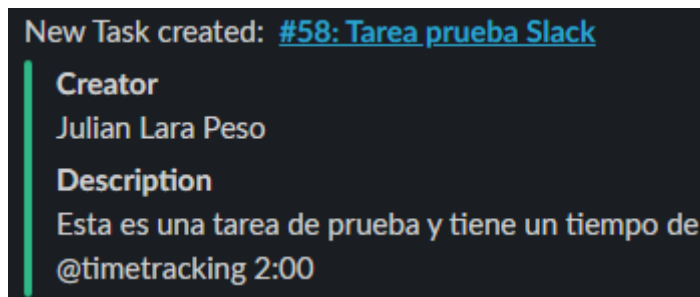
Esta es una tarea de prueba y tiene un tiempo de @timetracking 2:00

New ▼

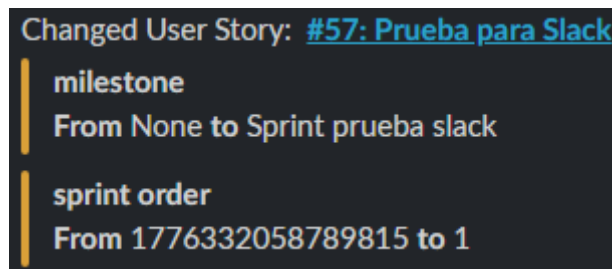
READY FOR TEST

Javier Ibañez Coca

🕒
📁
🔒

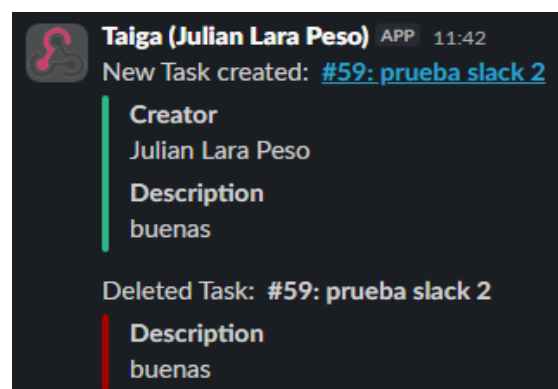
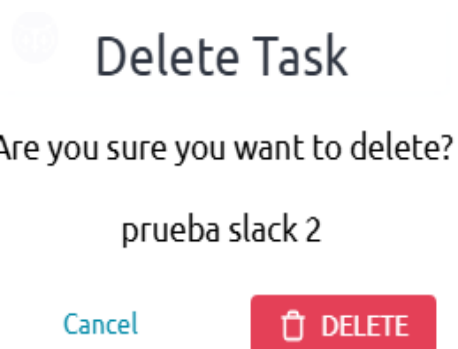


Además, me envió un aviso al mover nuestra US a un sprint, algo que no se nos pedía pero está bien saber.



c) Crea otra tarea en la misma US y borrarla

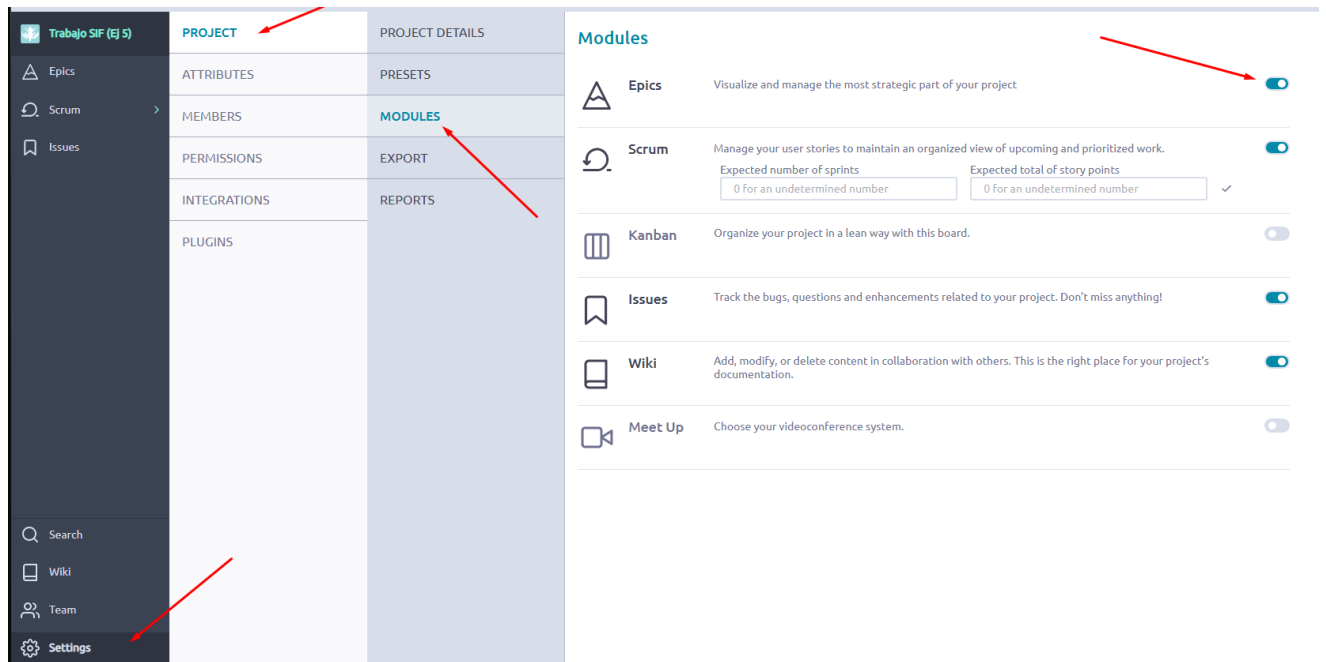
Para ello, haremos lo mismo que en el paso anterior. Cuando ya esté creada, deberemos borrar la tarea en los ajustes de esta misma. Después, nos saldrán estos avisos en Slacks.



d) Busque información sobre los Epics y sus diferencias con los US. Active los epics en los módulos de Taiga.

Los **Epics** son objetivos estratégicos de alto nivel que agrupan múltiples funcionalidades, mientras que las **User Stories (US)** son unidades de trabajo pequeñas y específicas que pueden completarse en un solo Sprint. En resumen, un Epic define el "qué" a gran escala y la US detalla el "cómo" para ser ejecutado de forma inmediata.

Para añadirlos, es tan sencillos como ir a ajustes → “Project” → “Modules” y activar la opción que dice “Epics”.



IMPORTANTE: No olvidar activar esta opción en el apartado de plugins.

Notify Epics

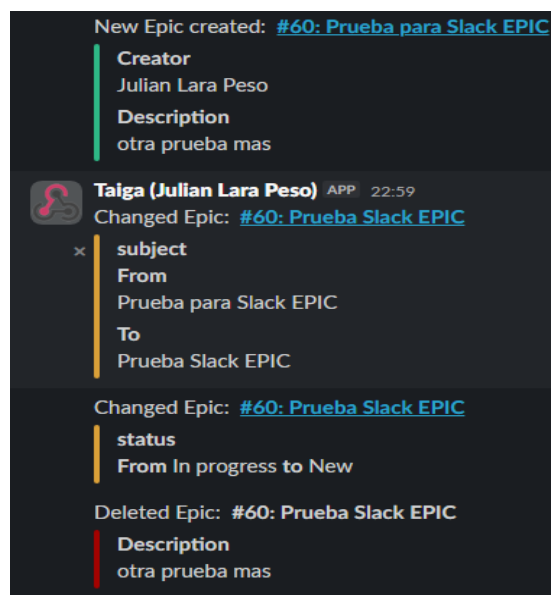
Create

Change

Delete



Ahora, pulsaremos en el apartado de “Epics” y crearemos uno. Para probar, modifique todo lo posible el “Epic” creado y lo borre, para así obtener todos los cambios confirmados:



EJERCICIO 4: Slack-Github (DAVID)

Enunciado

1. Crear un repositorio en GitHub llamado “sistemas_informacion” que contenga como mínimo el archivo Readme.
2. Agrega a Slack la aplicación de GitHub para recibir en Slack las notificaciones de modificación del repositorio.
3. Realice la integración de su repositorio GitHub en Slack
4. Modifique el archivo Readme haga un commit con el cambio. Compruebe si esto se refleja en el canal de Slack.

Crear el repositorio de GitHub

Siguiendo las instrucciones del ejercicio, el primer paso consistió en la creación de un repositorio en la plataforma GitHub denominado "**sistemas_informacion**". Durante el proceso de creación, se seleccionó la opción de inicializar el repositorio con un archivo **README.md**, cumpliendo así con el requisito de contenido mínimo exigido.

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1

General

Owner *

 holamellamodavidll-arch

Repository name *

sistemas_informacion

 sistemas_informacion already exists in this account

Great repository names are short and memorable. How about [musical-octo-goggles?](#)

Description

...

3 / 350 characters

2

Configuration

Choose visibility *

Choose who can see and commit to this repository

 Private

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

On ☒

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

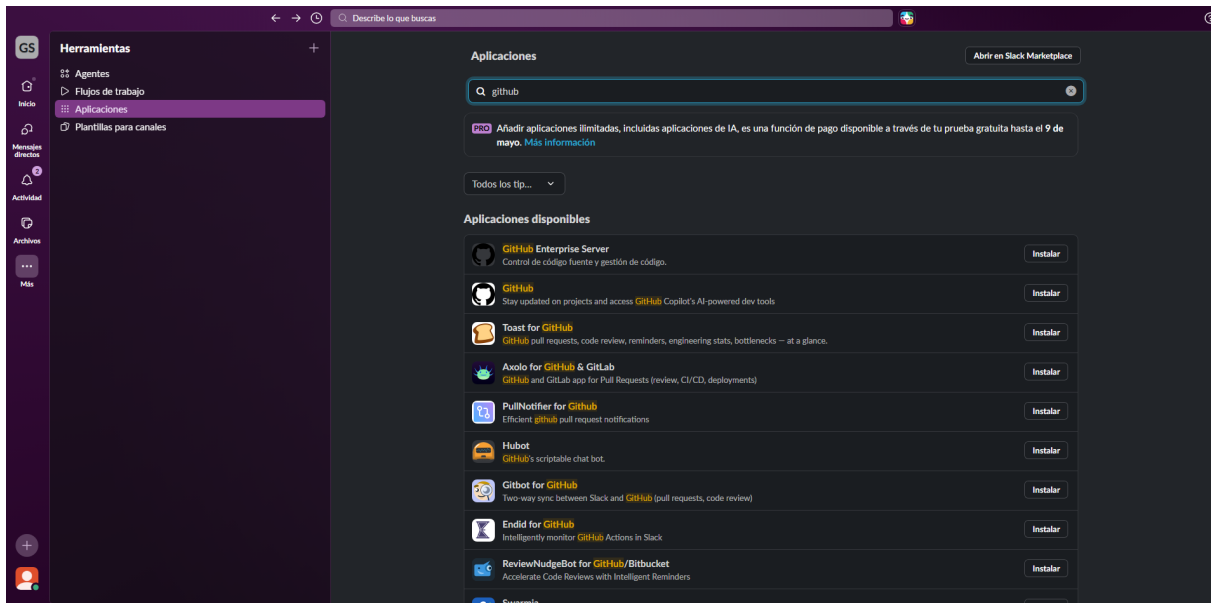
Add license

Licenses explain how others can use your code. [About licenses](#)

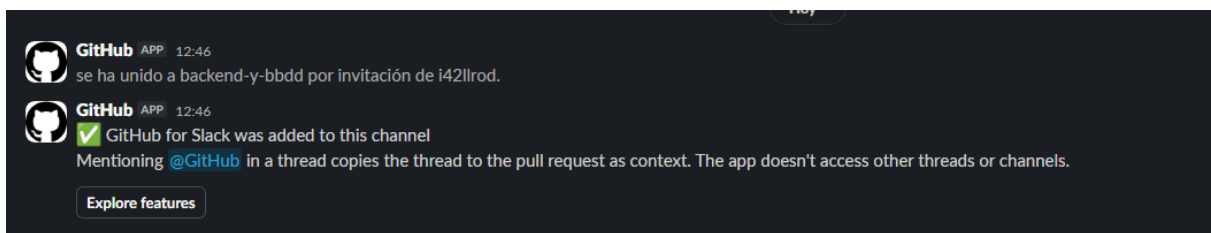
No license

Create repository

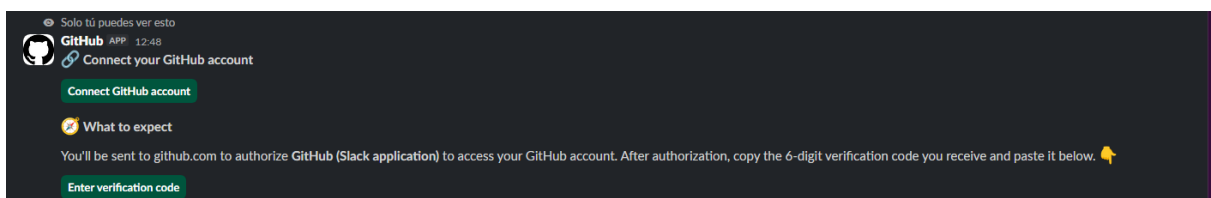
Para habilitar la comunicación entre ambas plataformas, se accedió al directorio de aplicaciones de Slack para buscar y agregar la aplicación oficial de **GitHub**. Esta aplicación es la encargada de recibir y procesar las notificaciones de modificación del repositorio para posteriormente enviarlas al canal correspondiente



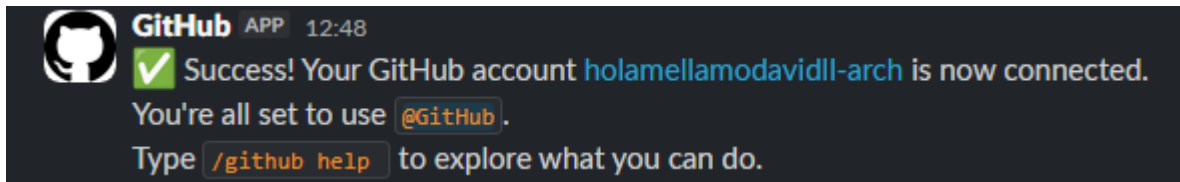
Una vez instalada la aplicación, se procedió a realizar la integración técnica de GitHub dentro del canal backend-y-bbdd con el comando `/invite @github`



A continuación nos pedirá conectar nuestra cuenta de GitHub, tenemos que iniciar sesión y verificar que somos nosotros con un código que se nos proporcionará desde Slack.



Una vez hemos iniciado sesión ya podemos hacer cualquier cosa, podemos ver todos los comandos que podemos usar escribiendo `/github help`.



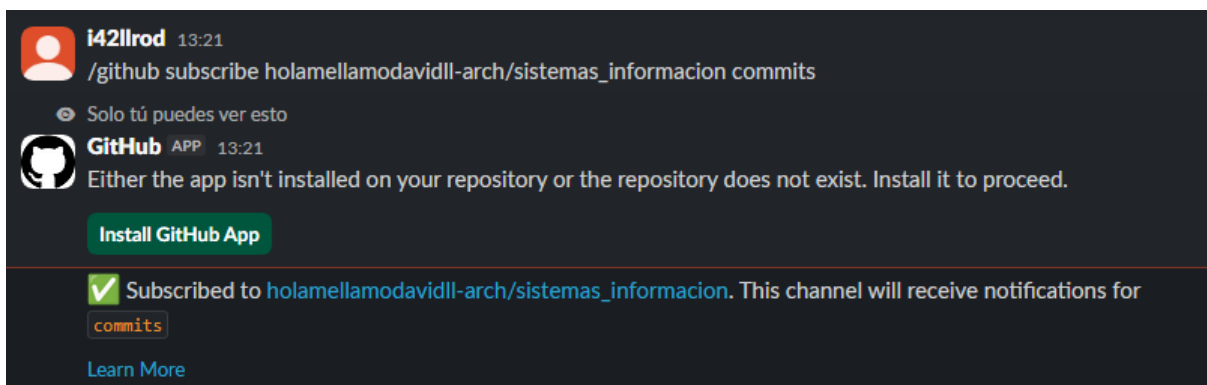
El comando principal ejecutado en el canal fue:

/github subscribe [usuario]/sistemas_informacion

Para asegurar que el canal notificará específicamente los cambios en el código y el archivo README, se activó la suscripción de commits mediante el comando:

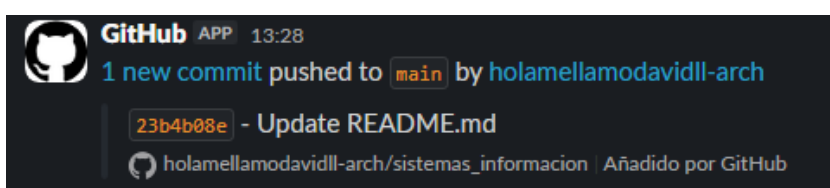
/github subscribe [usuario]/sistemas_informacion commits

Tras ejecutar estos comandos, el bot de GitHub solicitó una autenticación mediante un enlace web para verificar los permisos de acceso al repositorio, completando así el enlace entre ambas herramientas.



Para comprobar que la integración se realizó correctamente, se procedió a realizar una modificación directa en el archivo **README.md** desde la interfaz de GitHub. Al realizar el commit del cambio, se verificó de forma inmediata que el canal de Slack recibió una notificación automática detallando la siguiente información:

- El autor del cambio.
- El mensaje del commit realizado.
- El enlace directo al repositorio modificado.



EJERCICIO 5: Slack-... (JAVI)

Enunciado

1. Estudiar las posibles aplicaciones que se pueden integrar con Slack y seleccionar una que considere interesante para la asignatura y que no se haya tocado en esta práctica.

Indagando, hemos encontrado varias aplicaciones y hemos decidido estudiar más a fondo cuatro de ellas: Jira, Notion, Jenkins y PagerDuty.

Jira: es un gestor de tareas y problemas para equipos de desarrollo, que organiza el trabajo mediante metodologías ágiles, resolviendo el problema de quién hace qué y cuándo, además del control de los *sprints*.

Esta herramienta puede gestionar tareas, documentar en cierto modo, automatizar mediante reglas y es muy útil a nivel práctico.

Notion: Espacio de trabajo todo-en-uno: notas, *wikis*, bases de datos, documentos y gestión de proyectos. Resuelve el problema de la documentación y la colaboración en un documento, por lo que su principal uso es la documentación.

Jenkins: es un servidor de automatización de código abierto (*open source*) orientado a la integración y entrega continua (CI/CD). Resuelve el problema de garantizar que cada cambio en el código sea válido y desplegable de forma automática, sin intervención manual. Esta herramienta puede ejecutar *pipelines* de construcción y pruebas automáticamente tras cada *commit*, desplegar *software* en distintos entornos, integrarse con más de 1.800 *plugins* de terceros y notificar al equipo en Slack el resultado de cada *build*, siendo especialmente útil en entornos de desarrollo ágil y DevOps.

PagerDuty: es una plataforma de gestión de incidencias y operaciones digitales en tiempo real. Resuelve el problema de coordinar la respuesta cuando un sistema en producción falla, asegurando que la persona correcta sea avisada en el momento adecuado. Esta herramienta puede recibir alertas de múltiples fuentes de monitorización, enrutar automáticamente los avisos según turnos de guardia, escalar la incidencia si no hay respuesta, y gestionar todo el ciclo de vida del incidente directamente desde Slack, siendo especialmente útil para equipos de operaciones y SRE que necesitan minimizar el tiempo de caída de sus servicios.

Todas las opciones nos parecen interesantes, sin embargo, vamos a elegir Jenkins, puesto que, las otras integraciones, son reemplazables con las funciones u otras integraciones que hemos hecho en slack, y esta herramienta, cubre un aspecto importante que además hemos estudiado en asignaturas pasadas

Por lo que ahora pasamos a la integración

[< Explorar aplicaciones](#)



Jenkins CI

Añadir a Slack



-  desing-ux-ui
-  equipo
-  frontend
- # general-grupo-sif
- # jenkins
- # marketing
- # reuniones
- # taiga
- # trello

Elegir un canal...

[o crear un nuevo canal](#)

Añadir integración

Hemos tenido que instalar jenkins en un ordenador para usarlo:

 ¿Quieres que el gestor de contraseñas guarde la de esta página: "localhost:8080"? [Guardar](#) [Nunca](#)

Getting Started ×

Bienvenido a Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Jenkins 2.559

Getting Started



☒ Folders	☒ OWASP Markup Formatter	☐ Build Timeout	☐ Credentials Binding
☐ Timestamper	☐ Workspace Cleanup	☐ Ant	☐ Gradle
☐ Pipeline	☐ GitHub Branch Source	☐ Pipeline: GitHub Groovy Libraries	☐ Pipeline Graph View
☐ Git	☐ SSH Build Agents	☐ Matrix Authorization Strategy	☐ LDAP
☐ Email Extension	☐ Mailer	☐ Dark Theme	

```
** commons-lang3 v3.x Jenkins API
** Ionicons API
```

Folders

```
OWASP Markup Formatter
** ASM API
```

Configuramos una cosa y pasamos a instalarlo definitivamente en slack.

Plugins

📦 Updates

🛍️ Available plugins

⚙️ Installed plugins

⚙️ Advanced settings

🔍 SLACK

📦 Install

▼

↺

Install	Name ↓	Released	Health
☒	Slack Notification 795.v4b_9705b_e6d47 slack Plugins de notificación Integrates Jenkins with Slack, allows publishing build statuses, messages and files to Slack channels.	Hace 8 Mes 14 días	93
☐	Global Slack Notifier 1.5		

...	...
Pipeline Graph View	✓ Actualizado
Git	✓ Actualizado
EDDSA API	✓ Actualizado
Trilead API	✓ Actualizado
SSH Build Agents	✓ Actualizado
Matrix Authorization Strategy	✓ Actualizado
LDAP	✓ Actualizado
jsoup API	✓ Actualizado
Email Extension	✓ Actualizado
Mailer	✓ Actualizado
Theme Manager	✓ Actualizado
Dark Theme	✓ Actualizado
Loading plugin extensions	✓ Success
Apache HttpComponents Client 5.x API	✓ Actualizado
Slack Notification	... Pendiente
Loading plugin extensions	... Pending

→ [Volver al inicio de la página](#)

(puedes empezar a usar los plugins instalados inmediatamente)

→ ☐ Reiniciar Jenkins cuando termine la instalación y no queden trabajos en ejecución

Slack

Workspace ?

grupo-sif

Credential ?

slack-token-2



+ Add

Default channel / member id ?

#jenkins

☐ Custom slack app bot user ?

Avanzado ▾

Success

Test Connection

Clear cache

Completamos esta pestaña con los datos que piden.

Hoy ▾

 **i42ibcoj** 13:32
se ha unido a #jenkins. Además, i42lapej y 2 más se han unido.

 **i42ibcoj** 13:35
añadió una integración a este canal: [jenkins](#)

 **jenkins** APP 20:16
Slack/Jenkins plugin: you're all set on
<http://localhost:8080/>

Nuevo

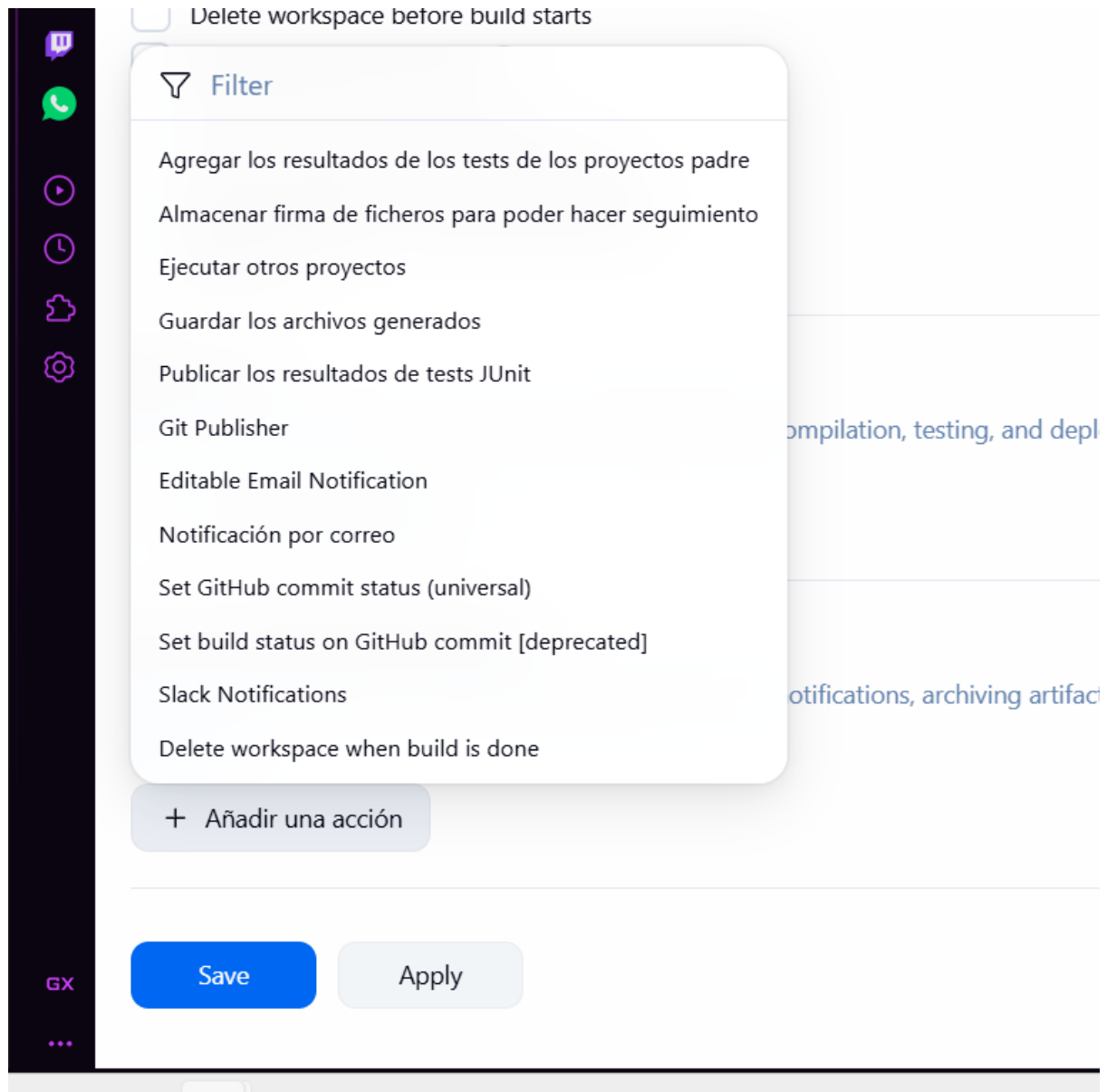
B *I* U ☒ | 🔗 🔧 🔧 🔧 | 🔧 🔧 🔧 🔧 | 🔧 🔧 🔧 🔧

Enviar un mensaje a #jenkins

ya está integrado

Probamos a crear un job

creamos un slack notificación



other jobs.

⋮ Slack Notifications



- ☒ Notify Build Start
- ☒ Notify Success
- ☒ Notify Aborted
- ☐ Notify Not Built
- ☐ Notify Unstable
- ☐ Notify Regression
- ☐ Notify Every Failure
- ☒ Notify Back To Normal

Avanzado ▾

probamos dándole a construir ahora

 / prueba-slack  / Tendencia del tiempo de ...

  [Sign in](#)

 Estado Actual

 Cambios

 Zona de Trabajo

 Construir ahora

 Configurar

 Borrar Proyecto

 Rename

Builds >

...

Today



#2 20:35



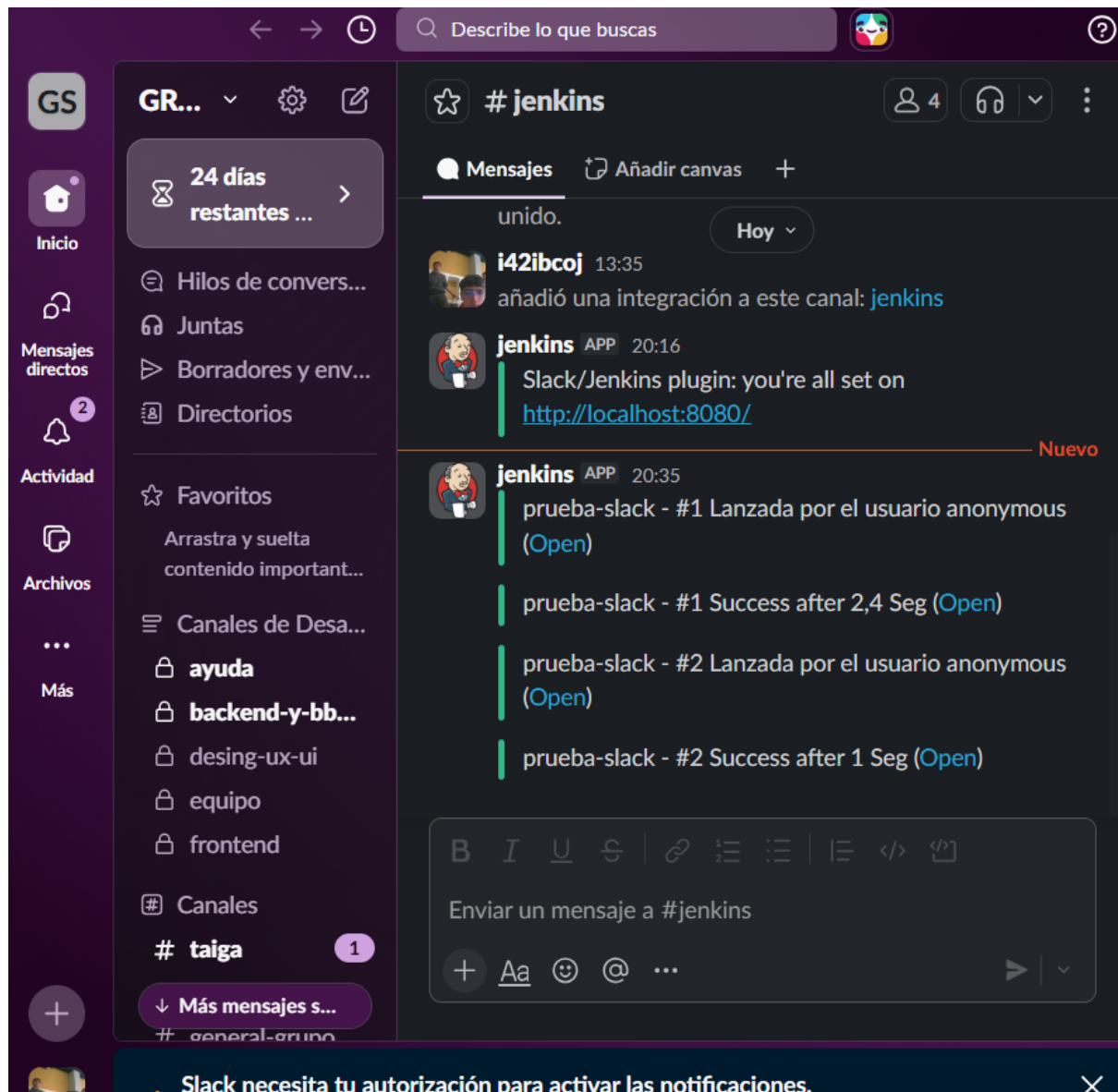


#1 20:35



Tendencia del tiempo de construcción

y sale en jenkins tras darle a construir



Conclusión

Bibliografía

Moodle Uco: <https://moodle.uco.es/moodlemap/>

