

Creación del Repositorio

Sprint deadline	20/10/2023
Tarjeta	SCRUM-12
Responsable	Martín Paez

Tabla 1 Detalle de la tarjeta correspondiente en Jira

1. Objetivos. Contenido de la tarjeta:

- Crear un mail del equipo.
- Crear una Organization en Github.
- Configurar permisos.
- Crear repositorio (.gitignore).
- Pushear Proyecto.
- Gestionar ramas.

2. Dependencias

En primer lugar, todas las tareas de desarrollo dependen de la creación del repositorio y estructura del proyecto. Por otro lado, todas aquellas actividades de diseño e investigación se pueden realizar en paralelo a esta.



3. Entregable

- Credenciales de acceso al nuevo mail.
- Invitación individual a la organización para cada miembro del equipo y a ["purple.crisalis@gmail.com"](mailto:purple.crisalis@gmail.com).
- Archivo ".gitignore" para trabajar con Java Spring y React.
- Plantilla de carpetas y configuraciones iniciales.
 - Una persona que clone el repositorio y tenga Docker con la base de datos acorde a la configuración realizada en el Workshop debería poder levantar el servidor y testear el acceso a la base de datos a través de algún método básico de la API.
 - Datos de conexión: *"sqlserver://localhost:1433, password=clave_2019, name=sa"*
- Principales ramas que determinan el flujo de trabajo.
- Permisos para creación de ramas por desarrollador.

1. Procedimientos

1.1. Google Drive

Se creó un nuevo correo en Gmail. Luego, en Google Drive se dispuso la carpeta *"/BootcampCrisalis"* con la finalidad de que cada miembro tenga compartido dicho directorio en su cuenta personal y pueda trabajar con el contenido sin necesidad de iniciar sesión directamente en la nueva cuenta.

Cabe mencionar que la documentación terminada se ira guardando en formato *"pdf"* dentro de la carpeta *"Docs"* del directorio raíz del repositorio.



1.2. Github Organizations

A diferencia de una cuenta individual en Github, una organización es una cuenta de nivel superior diseñada para colaboración y gestión de proyectos con múltiples miembros. Se creó una cuenta gratuita, y, se agregó individualmente, tanto a cada miembro del equipo, como al usuario purple.crisalis@gmail.com. Respecto a la asignación de permisos se profundizará en la sección “1.6 Gestión de ramas y permisos”.

1.3. Repositorios

Dentro de la organización se crearon dos repositorios: “Sistema-de-gestion” y “Para-Practicar-Git”. El primero, es donde se aloja nuestro proyecto; mientras que el segundo, se dispuso para que, aquel que solicite la membresía, pueda realizar pruebas con Github Organizations: “push”, “pull request”, “configuraciones”, etc.

1.4. Gitignore

Por un lado, originalmente se creó el repositorio con el archivo “.gitignore” para Java que provee Github durante el proceso. Por el otro, las reglas para trabajar con React las eligieron los miembros del equipo que iban a estar trabajando con el Frontend. Al ser pocas líneas, se unificaron los archivos en uno ubicado en la carpeta raíz del repositorio con las rutas necesarias actualizadas.

1.5. Template del Proyecto

Respecto al Backend, elegimos uno de los proyectos que se hicieron durante el Workshop de Spring. Se arreglaron algunos errores y se aseguró de que cada miembro pudiera clonar y efectivamente acceder a la base de datos desde la API.



En relación al Frontend, los miembros del equipo que iban a trabajar en él crearon un proyecto nuevo por línea de comandos.

Por último, se dispuso el código en las dos carpetas correspondientes, presentes en la raíz del repositorio.

1.6. Gestión de ramas y permisos

El rol gratuito de menor privilegio en Github es “Read”, le sigue “Triage”. Este último ya puede administrar pull request, mientras que el primero no puede crear ramas.

Como el rol de “Read” no permite la creación de ramas, y, para el alcance de nuestro proyecto el segundo es demasiado permisivo, se optó por simplificar el universo de roles a dos: “Read” y “Admin”. En función de ello, se le asignó el rol “Admin” a cada miembro del equipo y “Read” a purple.crisalis@gmail.com.

Originalmente se había realizado un “push” a la rama “master”, pero considerando que Github por defecto contempla de manera diferente a la rama “main”, decidimos emplear las ramas “main”, “dev” y “release”.

El flujo de trabajo será el descrito por los tutores en el Workshop, en el cual cada desarrollador hará un “push” a su rama en remoto y solicitará un “pull request” a la rama “dev”. Luego, el líder técnico decidirá si aceptarlo, para posteriormente pasar a una etapa de testing en la rama “main”. Finalmente, después de haber pasado los test, se podrá actualizar la rama “release” con las nuevas funcionalidades, allí se encontrará el entregable al final de cada “sprint”.

Respecto a la gestión individual de ramas, hay que mencionar que cada desarrollador creará las suyas, asignándoles un nombre que referencie a la tarjeta cuya funcionalidad desarrollará. Por consiguiente, se decidió que tendrá el siguiente formato: “SCRUM-X”.



4. Resultados

4.1. Credenciales de acceso al nuevo mail

- Email : bootcampcrisalis@gmail.com
- Contraseña : L4P4ssword

4.2. Github Organization

Organization URL : <https://github.com/PurpleTeamCrisalis>

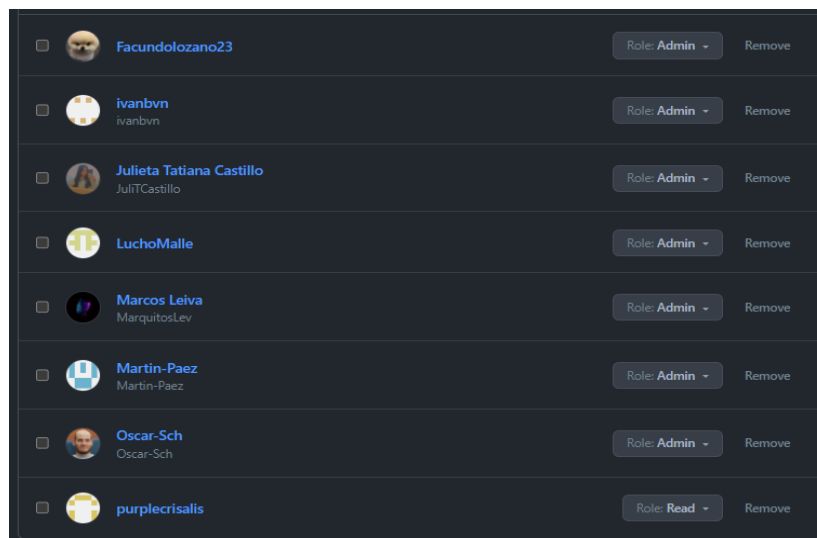


Ilustración 1 Captura de pantalla de la sección de "Administración de acceso" del repositorio "Sistema-de-gestion"

1.1. Repositorios



Ilustración 2 Captura de pantalla de la sección "Repositorios" de la Organización "PurpleTeamCrisalis". Se destaca el repositorio público principal del proyecto: "Sistema-de-gestión".



1.2. Plantilla de proyecto subido al repositorio

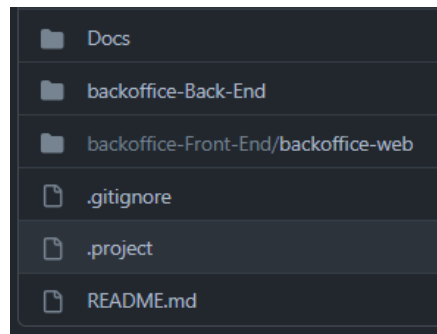


Ilustración 3 Captura de pantalla de la carpeta raíz del repositorio "Sistema-de-gestion". Se destaca la separación entre FrontEnd y BackEnd, el archivo ".gitignore" y la carpeta Docs (donde se podrá encontrar este documento en formato pdf).

1.3. Ramas y flujo de trabajo



Ilustración 4 Captura de pantalla de las ramas del repositorio "Sistema-de-gestion", obtenida a la hora de confeccionar el informe. Se destacan las ramas "main", "dev" y "release", así como las ramas individuales correspondientes a cada tarjeta.

2. Conclusión

Queda a disposición de cada miembro del equipo la carpeta “/BootcampCrisalis” para que todos puedan gestionar el contenido directamente desde su cuenta personal. El producto final se guardará en formato “pdf” en la carpeta “Docs” del directorio raíz del repositorio “Sistema-de-gestion”.

Además, todos los miembros del equipo tienen permiso de administrador en la Organización, por lo cual es fundamental la comunicación y respeto de las tareas individuales de cada uno.

Y, por último, cada desarrollador será responsable de crear y nombrar sus ramas con el formato “SCRUM-X” (en referencia a la tarjeta elegida) y solicitar el “pull request” a la rama dev, evitando modificar directamente la rama.