



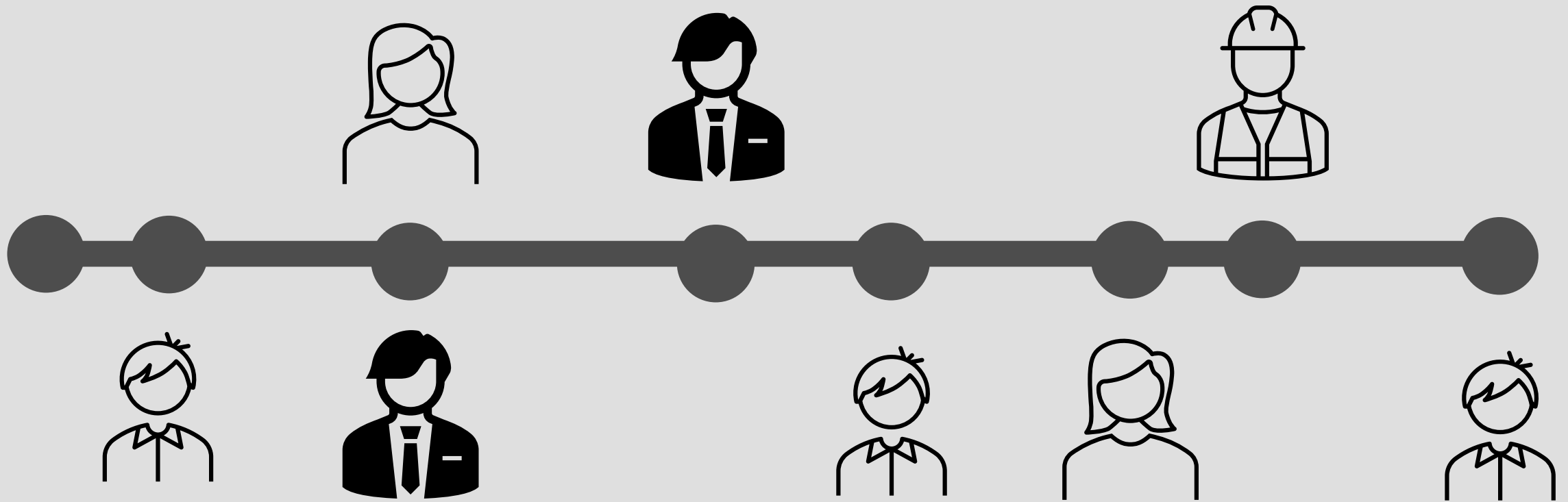
Guía para enseñanza e investigación colaborativa con **GitHub**

R.TeachingResearchGuide

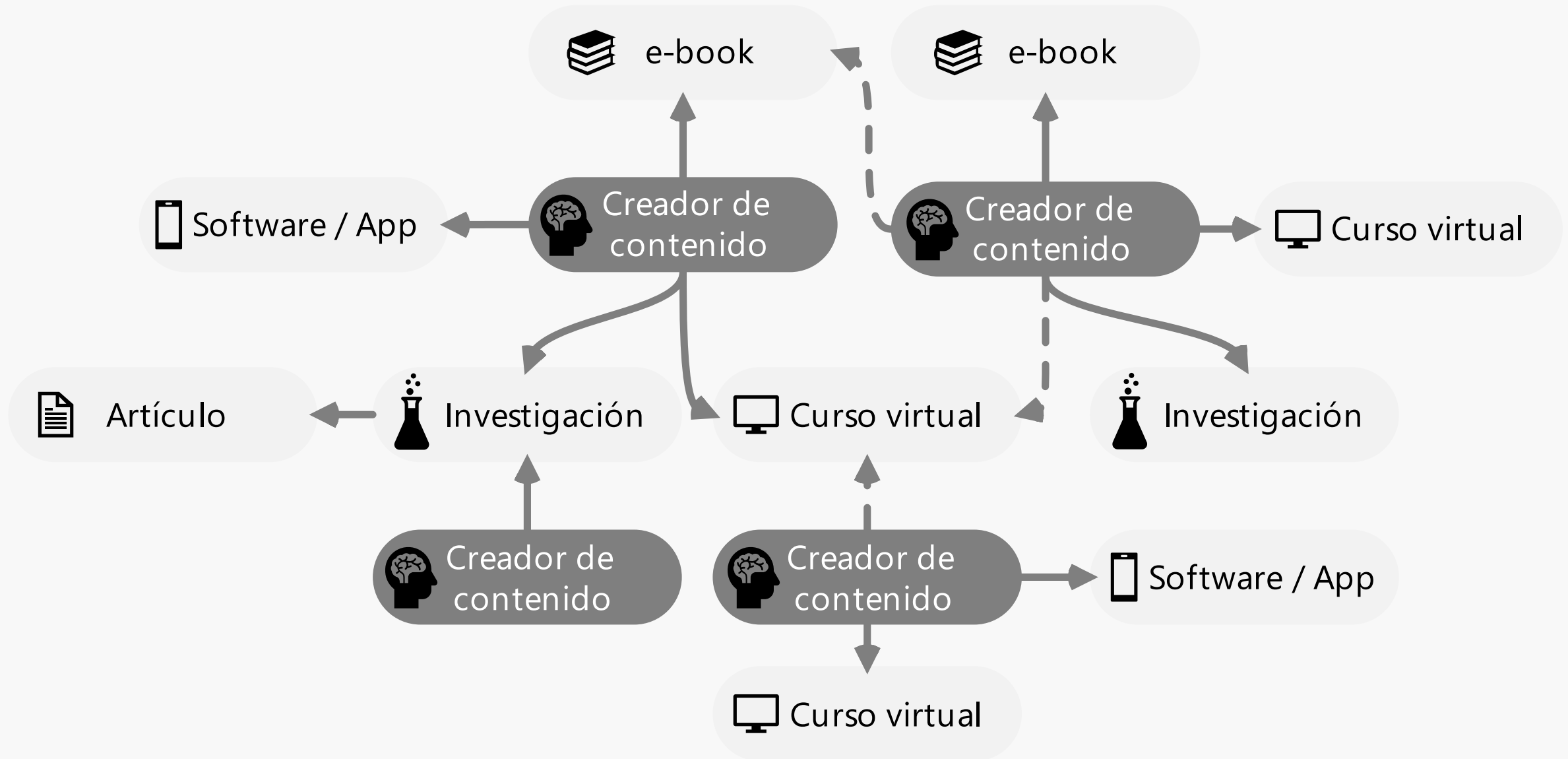
<https://github.com/rcfdtools>

Sección 1 - **Introducción, fundamentos e implementación** de GitHub

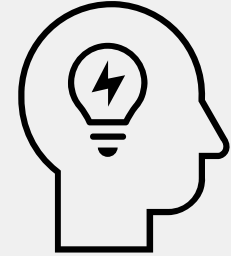
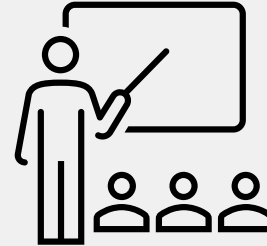
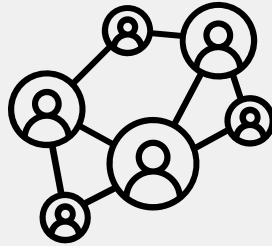
¿Qué es el desarrollo colaborativo?



Creación de contenidos de forma **individual o colectiva**



Beneficios de trabajar colaborativamente



- ✓ Combinación de experticia.
- ✓ Creación de redes colaborativas.
- ✓ Interacción directa entre grupos, profesores y estudiantes.
- ✓ Solución de casos de estudio desde diferentes perspectivas.
- ✓ Conocimiento compartido.
- ✓ Agrega valor a las organizaciones.



Cooperación vs. Colaboración

Alcance	Cooperación	Colaboración
Requiere	Respeto mutuo	Confianza mutua
Requiere	Transparencia	Vulnerabilidad
Incluye	Ideas compartidas	Valor compartido
Dependencia	Independencia	Interdependencia
Interacción	Corto plazo	Largo plazo
Involucra	Compartir ideas	Generar nuevas ideas



Agregar valor a mi organización

Combina tu **experticia** y la experticia de los miembros de tú equipo de trabajo **para crear contenidos** compartidos.



Agregar valor a mi organización

Crea oportunamente **contenidos** científicos **de alta calidad** con actualización permanente.

"Encuentra las necesidades de tus usuarios y dales lo que necesitan mucho más rápido que tus competidores"



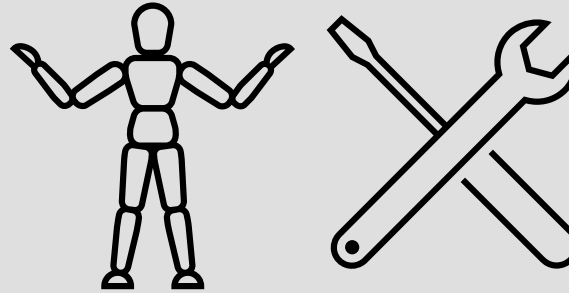
Agregar valor a mi organización

Busca siempre **la manera de dar valor** agregado a **todos los contenidos** en los que participes.



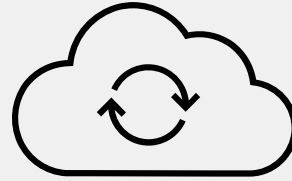
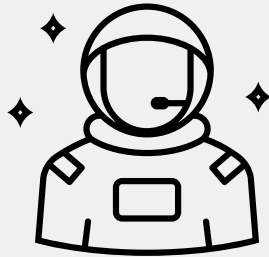
Agregar valor a mi organización

Interactúa con los miembros de tu comunidad.



Herramientas para desarrollo colaborativo

Características de las **plataformas** de desarrollo colaborativo



- ✓ Disponibilidad de los archivos fuente de uno o varios proyectos.
- ✓ Soporte y mantenimiento a investigaciones y sistemas informáticos ya concluidos y en explotación.
- ✓ Control de versiones y trazabilidad.
- ✓ Foros de discusión.
- ✓ Wikis para gestionar la documentación.

Plataformas de desarrollo colaborativo



GitHub



GitLab



Google Code



GNU Savannah



SOURCEFORGE



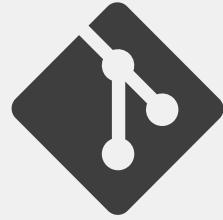
ATLASSIAN
Bitbucket

¿Cuál plataforma usar?

Requerimiento	GitHub	GitLab	Google Code
Creación de cuenta sin versión de prueba o que expira	✓		✓
Repositorios públicos sin login de usuario	✓		
Repositorios ilimitados por usuario sin versión de pago	✓		
Creación de organizaciones	✓	✓	
Clonación de repositorios entre usuarios y organizaciones	✓	✓	
Descarga directa de repositorio sin registro de usuario	✓		✓
Buscador de proyectos sin registro de usuario	✓		
Cargue masivo de archivos inferiores a 100 MB en cuentas libres	✓		
Gestión local de repositorio utilizando herramientas de desarrollo	✓	✓	✓
Entornos de discusión por repositorio público	✓	✓	✓
Llamado de recursos entre repositorios (archivos, paquetes de datos, librerías...)	✓	✓	✓
Aplicación móvil oficial con opciones de edición, discusiones, asuntos, notificaciones.	✓		



Fundamentos de **Git**



¿Qué es **Git**?

Git es un **software** que **permite rastrear** y **validar cambios** en archivos de un proyecto o un repositorio.

Arquitectura distribuida donde el código de cada desarrollador es también un repositorio que puede albergar el historial completo de todos los cambios.

Diseñado para ofrecer a usuarios:
rendimiento, seguridad y la flexibilidad

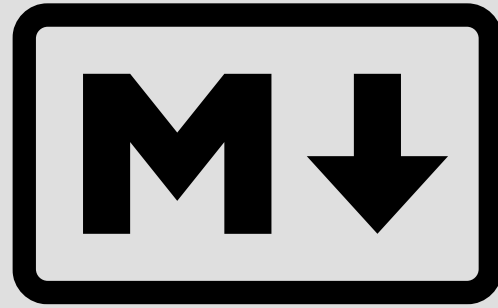
Fork: en GitHub, cualquier usuario puede clonar un repositorio público creado por otro usuario y este aparecerá dentro de los repositorios del usuario que realizó la clonación. Complementariamente, usuarios pueden descargar un comprimido completo de un repositorio específico.

Branch: las ramas en Git son utilizadas para de forma aislada, realizar modificaciones y depuraciones de prueba en el código o la documentación sin afectar la rama principal. En un repositorio pueden existir múltiples ramificaciones y una vez se verifica que los cambios son válidos, estos son incorporados a la rama principal.

Pull-request: una solicitud de incorporación de cambios es una forma de pedirle a otro desarrollador que fusione una de tus ramas en su repositorio. Esto no solo permite a los responsables del proyecto realizar un seguimiento de los cambios más fácilmente, sino que además permite a los desarrolladores iniciar debates sobre su trabajo antes de integrarlo con el resto de los archivos o el código base.

Push: la incorporación de las modificaciones realizadas a un documento Markdown, a un código o a cualquier elemento nuevo dentro del repositorio, son realizadas a través de una carga o Push.

Commit: al realizar modificaciones sobre el código o la documentación, es necesario incluir comentarios que ayuden a los demás usuarios a entender los cambios realizados.



Lenguaje **Markdown** y escritura **en GitHub**

Actividades prácticas en GitHub



Creación y gestión en línea de repositorios y documentos.



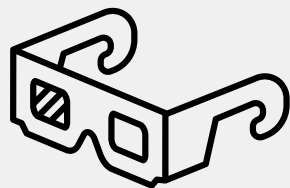
Fundamentos de organizaciones y equipos.



Centro de documentación Wiki.



Centro de discusiones.



Gracias por su atención