### **🔄 GitHub vs Bitbucket: ¿Cuál usar con ArgoCD?**

|  |  |  |
| --- | --- | --- |
| **Característica** | **GitHub** | **Bitbucket** |
| ✅ Popularidad | Muy alta. Comunidad masiva. | Menor, más usado por empresas que usan Jira. |
| 🔐 Repos privados gratuitos | Sí (ilimitados). | Sí (ilimitados). |
| 🔧 Integración con ArgoCD | Excelente. Muy bien documentado. | Muy buena también. |
| 🧩 Integración con Jira | No nativa, necesitas apps. | Nativa. Bitbucket es de Atlassian (dueño de Jira). |
| 🚀 CI/CD nativo | GitHub Actions (poderoso y flexible). | Bitbucket Pipelines (más simple). |
| 📁 Manejo de múltiples repos | Bueno, pero sin workspaces. | Usa “workspaces” para agrupar repos por equipo. |
| 🧑‍🤝‍🧑 Empresas grandes | Muy usado por proyectos open-source. | Preferido por empresas que usan el stack Atlassian. |
| 🗂️ Interfaz | Moderna, muy amigable. | Más técnica, más pensada para proyectos internos. |

### **🔍 Entonces, ¿cuándo usar cada uno?**

**Usa GitHub si:**

* Ya tienes tu cuenta y lo conoces bien.
* Te interesa la comunidad y el open source.
* Quieres usar GitHub Actions para CI/CD.
* No necesitas integración directa con Jira o herramientas Atlassian.

**Usa Bitbucket si:**

* Tu empresa usa Jira o Confluence.
* Te interesa una estructura por *workspace*.
* Quieres una interfaz más privada o empresarial.

Un **workspace** (espacio de trabajo) es una **forma de agrupar varios repositorios, usuarios y configuraciones** bajo una misma “organización” o equipo. Este concepto lo usa Bitbucket (y también GitLab de manera similar), pero **GitHub no lo llama así**, aunque tiene funcionalidades similares con las **organizaciones**.

### **🧱 ¿Para qué sirve un Workspace en Bitbucket?**

Aquí tienes sus funciones clave:

### **✅ 1. Organización por equipos o proyectos**

Un workspace permite:

* Agrupar repositorios por proyecto, empresa o cliente.
* Compartir acceso a todos los repos con tu equipo fácilmente.
* Separar tus proyectos personales de los de una empresa.

🔸 *Ejemplo*:  
 Tienes un workspace llamado microservices-infra, y dentro tienes:

* repo-api-gateway
* repo-auth-service
* repo-user-service

### **✅ 2. Control de acceso centralizado**

Desde un workspace puedes:

* Agregar miembros (usuarios) y darles permisos por defecto a todos los repos.
* Controlar quién puede hacer push, pull, administrar CI/CD, etc.

🔐 Muy útil en equipos de desarrollo.

### **✅ 3. Configuración compartida**

Puedes configurar:

* **Pipelines comunes** para CI/CD que todos los repos pueden heredar.
* Webhooks o integraciones con ArgoCD, Jira, Slack, etc.
* Políticas de ramas: proteger main, forzar PRs, etc.

### **✅ 4. Visibilidad empresarial**

Empresas usan workspaces para tener:

* Dashboard único con todos los proyectos.
* Auditoría centralizada.
* Gestión de secretos o configuraciones en un solo lugar.

### **¿Y en GitHub?**

GitHub tiene conceptos similares:

* **Organizaciones** → equivalente al workspace.
* **Teams** → roles y permisos.
* **Secrets compartidos**, políticas, acciones comunes, etc.

Un **workspace** (espacio de trabajo) es una **forma de agrupar varios repositorios, usuarios y configuraciones** bajo una misma “organización” o equipo. Este concepto lo usa Bitbucket (y también GitLab de manera similar), pero **GitHub no lo llama así**, aunque tiene funcionalidades similares con las **organizaciones**.

### **🧱 ¿Para qué sirve un Workspace en Bitbucket?**

Aquí tienes sus funciones clave:

### **✅ 1. Organización por equipos o proyectos**

Un workspace permite:

* Agrupar repositorios por proyecto, empresa o cliente.
* Compartir acceso a todos los repos con tu equipo fácilmente.
* Separar tus proyectos personales de los de una empresa.

🔸 *Ejemplo*:  
 Tienes un workspace llamado microservices-infra, y dentro tienes:

* repo-api-gateway
* repo-auth-service
* repo-user-service

### **✅ 2. Control de acceso centralizado**

Desde un workspace puedes:

* Agregar miembros (usuarios) y darles permisos por defecto a todos los repos.
* Controlar quién puede hacer push, pull, administrar CI/CD, etc.

🔐 Muy útil en equipos de desarrollo.

### **✅ 3. Configuración compartida**

Puedes configurar:

* **Pipelines comunes** para CI/CD que todos los repos pueden heredar.
* Webhooks o integraciones con ArgoCD, Jira, Slack, etc.
* Políticas de ramas: proteger main, forzar PRs, etc.

### **✅ 4. Visibilidad empresarial**

Empresas usan workspaces para tener:

* Dashboard único con todos los proyectos.
* Auditoría centralizada.
* Gestión de secretos o configuraciones en un solo lugar.

### **¿Y en GitHub?**

GitHub tiene conceptos similares:

* **Organizaciones** → equivalente al workspace.
* **Teams** → roles y permisos.
* **Secrets compartidos**, políticas, acciones comunes, etc.

### **🧠 Conclusión rápida:**

🔧 **Un workspace es como una "empresa virtual" dentro de Bitbucket** donde puedes agrupar y controlar todo lo relacionado a tu código fuente, equipos y automatizaciones.