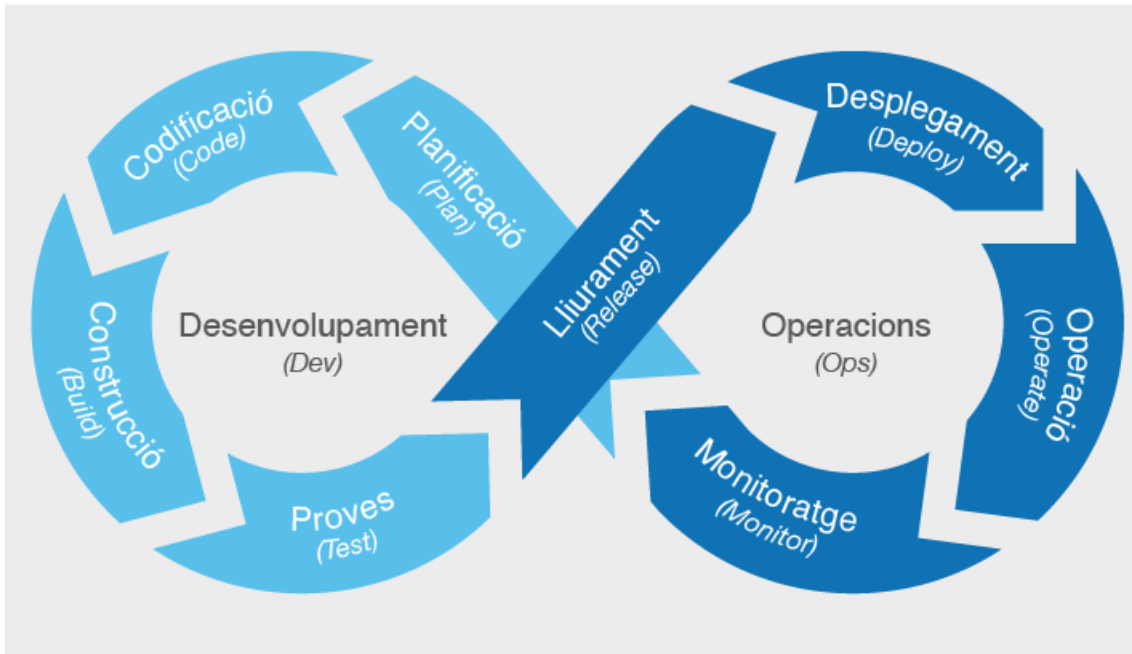


## El cicle DevOps



### 1. Context: El Cicle DevOps

- DevOps integra desenvolupament (Dev) i operacions (Ops) per accelerar i automatitzar el lliurament de programari de qualitat.
- Les fases del cicle són contínues, de manera que les aplicacions evolucionen constantment amb canvis iteratius i validats.

### 2. Fases rellevants amb exemples concrets

- Planificació (Plan): Els requisits del programari defineixen quines funcionalitats han de ser validades amb proves funcionals (exemple: comprovar que un formulari web funciona correctament).
- Codificació (Code): Els desenvolupadors implementen les funcionalitats requerides i escriuen proves unitàries per validar el codi a nivell intern.
- Proves (Test): Aquí és on entra Selenium, que simula l'acció dels usuaris en un navegador per comprovar si l'aplicació respon com s'espera.  
(Exemple: Simular que un usuari introdueix el seu nom en un formulari i comprovar si es mostra correctament.)

### **3. Lliurament (Release):**

- Aquesta fase inclou l'automatització de proves amb Github Actions.
- Cada cop que un desenvolupador fa un canvi al codi (push), Github Actions executa automàticament les proves unitàries i funcionals per assegurar que no s'han introduït errors.

### **4. Benefici d'integrar Selenium i Github Actions en DevOps**

- Eficiència: Evitem provar manualment cada funcionalitat després de cada canvi. Les proves són automàtiques i s'executen amb cada modificació al repositori.
- Qualitat contínua: Les proves detecten errors abans que el codi es desplegui, reduint el risc d'errors en producció.
- Velocitat: Automatitzant proves i validacions, els equips poden lliurar noves funcionalitats més ràpidament.