**Planificación del Pipeline de Integración y Entrega Continua (CI/CD)**

1. Commit y Push al repositorio: Los desarrolladores hacen commits y los pushean a la rama en la que estén trabajando, y más tarde fusionan los cambios de esta a la rama principal mediante un Pull Request.
2. Detección del cambio en el repositorio: GitHub Actions inicia el siguiente pipeline cada vez que se hace un push al repositorio:
   1. Compilación del código: Se ejecuta un script para compilar el código que se haya pusheado. De esta manera se detectan errores de compilación de manera temprana antes de que continue el pipeline.
   2. Ejecución de tests unitarios: Se ejecutan scripts diseñados por los desarrolladores cuya función es comprobar el comportamiento esperado de fragmentos pequeños del programa. En nuestro caso del paradigma orientado a objetos, los tests unitarios suelen comprobar el correcto funcionamiento de las clases individualmente.
   3. Construcción de la imagen Docker: Se ejecuta un comando para crear la imagen Docker de la aplicación. Así la aplicación se queda empaquetada en una imagen de memoria junto con todas sus dependencias, facilitando su portabilidad.
   4. Push de la imagen a un registro: Se sube a un registro de contenedores como Docker Hub o GitHub Container Registry, donde es versionado y almacenado.
   5. Despliegue en entorno de prueba: Se usa Docker para desplegar la imagen en un entorno que simule las condiciones de producción. También se pueden incluir scripts que realicen tests adicionales una vez en el entorno de prueba.
   6. Aprobación: Finalmente, los cambios son aprobados o bien de manera manual por uno de los programadores o bien de manera automática.

Con este pipeline automatizamos muchos procesos y hacemos más eficaz el desarrollo colaborativo.