# Integración Continua

Grupo Nº 6: Aresu, Bruno Legajo: 73446 // Boné, Nicolás Legajo: 72624 // Guanuco, Alejandro Legajo: 57790 // Caparroz, Ezequiel Legajo: 54346 // Jornet, Pablo. Legajo: 62901

# ¿Qué resuelve al Integración Continua?

Martin Fowler propuso con este modelo facilitar la detección de errores. "La integración continua no elimina los bugs, pero ayuda a encontrarlos y eliminarlos más fácil"

# ¿Cómo?

Desarrolladores deben *integrar el código en un repositorio* compartido con la mayor frecuencia posible. Los cambios del desarrollador se validan creando una compilación y ejecutando pruebas automatizadas.

Está práctica le brinda mayor importancia a la **automatización de pruebas**, para verificar que la aplicación no se rompa cada vez que se integren nuevos cambios en el branch principal.



#### ¿ Cómo aplicar la integración continua?

Mantener un único repositorio de origen.

Automatizar la creación del "build".

Cada commit debe basarse en una máquina de integración.

Realizar las pruebas en un entorno igual al de Producción.

Facilitar al equipo la última versión ejecutable.

Todos pueden ver lo que pasa

Automatizar la implementación.

# Ventajas

Menos errores en Producción

 Porque que con las regresiones son capturadas temprano por las pruebasautomatizadas.

Release más fáciles de crear.

Porque los problemas de integración ya se han resuelto.

Se reduce el "cambio de contexto"  Porque los desarrolladores reciben alertas pronto surgen los errores y trabajar para solucionarlos antes de pasar a otra tarea.

Costos de pruebas reducidos

 Porque un servidor dedicado puede ejecutar cientos de pruebas en cuestión de segundos.

Equipos de calidad menos atareados en pruebas.

 Porque tiene pruebas automáticas y puede dedicar más tiempo a otras mejoras

# Desventajas

Dedicación a crear pruebas automatizadas.

Contar con servidor de integración continua para monitorear y ejecutar las pruebas.

Realización de Merges de código con mayor frecuencia posible.

### Buenas prácticas para la integración continua

Integrar temprano y con frecuencia Mantener el "semáforo en verde" en todo momento

Escribir pruebas como parte de las User Stories

Escribir pruebas mientras se realiza el bug-fixing

La
integración
continua
permitirá al
equipo de
calidad
escalar la
calidad

## Tipos de pruebas

Pruebas de las pruebas de UI aseguran que la aplicación funcione correctamente desde la perspectiva del usuario

Pruebas de aceptación

se centran en los casos de negocio en lugar de los componentes en sí.

Pruebas de integración

se aseguran de que varios componentes se comportan correctamente juntos.

Prueba Unitaria

Se verifica el comportamiendo de métodos o funciones individuales

# Conclusión

- Luego de haber analizado la práctica de Integración Continua, consideramos que es una herramienta valiosa que permite detectar los errores rápidamente y mejorar la calidad del software durante el desarrollo del mismo. Permite una rápida retroalimentación de nuestro proyecto.
- La entrega continua abarca más que el proceso de pruebas e incluye al proceso de entrega. Las nuevas características y las modificaciones realizadas en el código llegan automáticamente al usuario final.

Fuentes Utilizadas https://www.campusmvp.es/recursos/post/integracion-continua-que-es-y-por-que-deberias-aprender-a-utilizarla-cuanto-antes.aspx

https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/integracion-continua/

http://blog.juliopari.com/integracion-continua-en-proyectos-agiles-de-software/

https://www.thoughtworks.com/continuous-integration

https://medium.com/@ajamaica/que-es-la-integraci%C3%B3n-continua-y-por-que-debr%C3%ADas-usarla-aea591f2a7d9