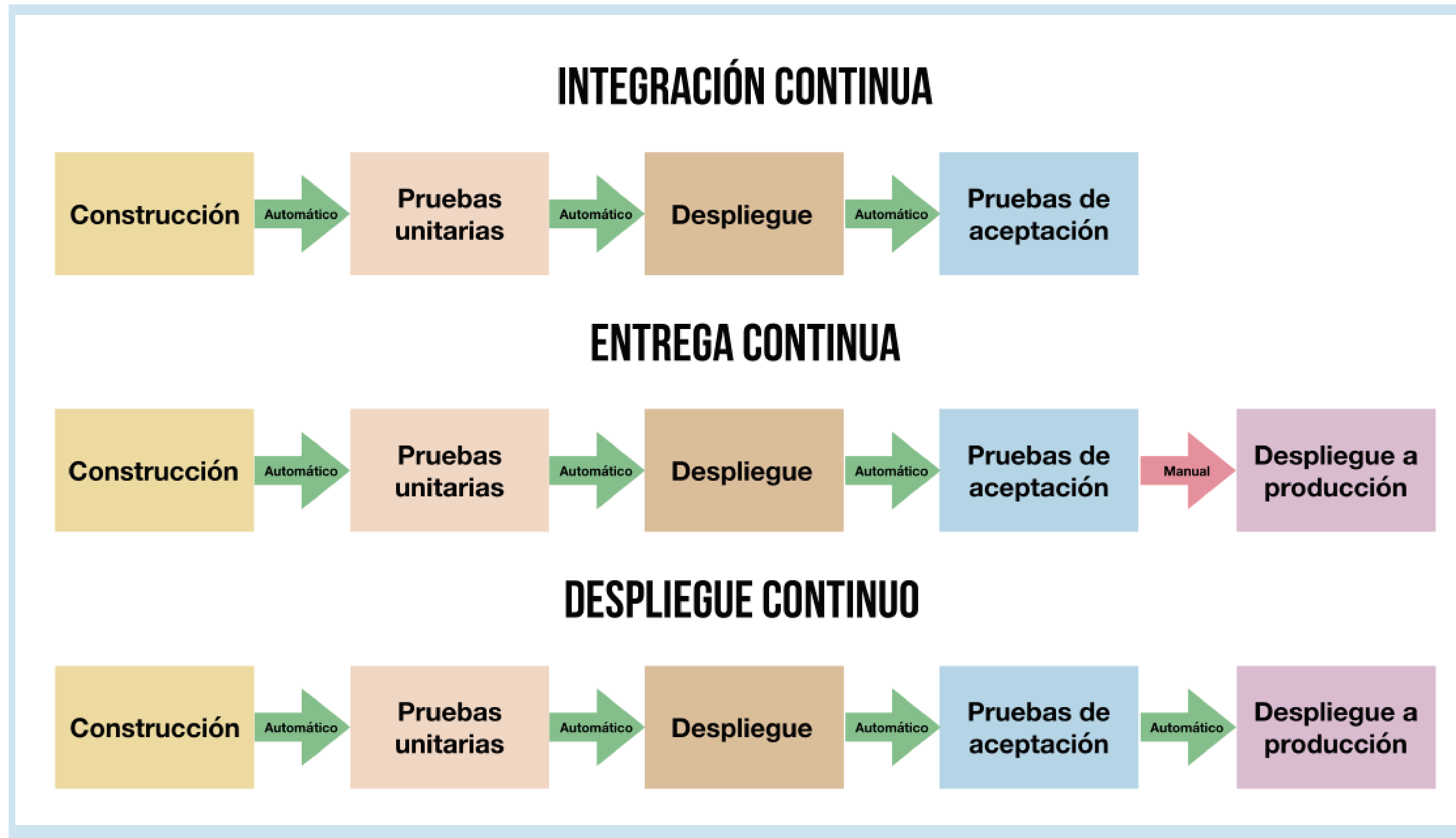


Universidad Tecnológica Nacional
Facultad Regional Córdoba
Cátedra de Ingeniería y Calidad de Software
Docentes: Judith Meles- Laura Covaro

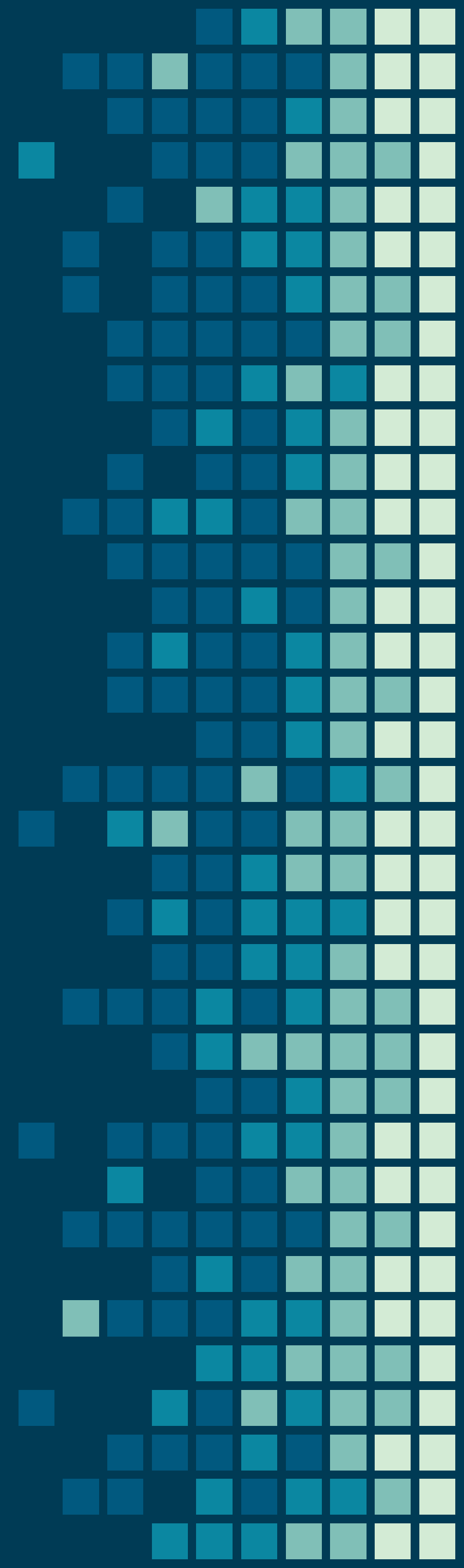
Prácticas Continuas en el Software (Continuous Practices)



Integración, Entrega y Despliegue Continuos (CI, CD, CD)

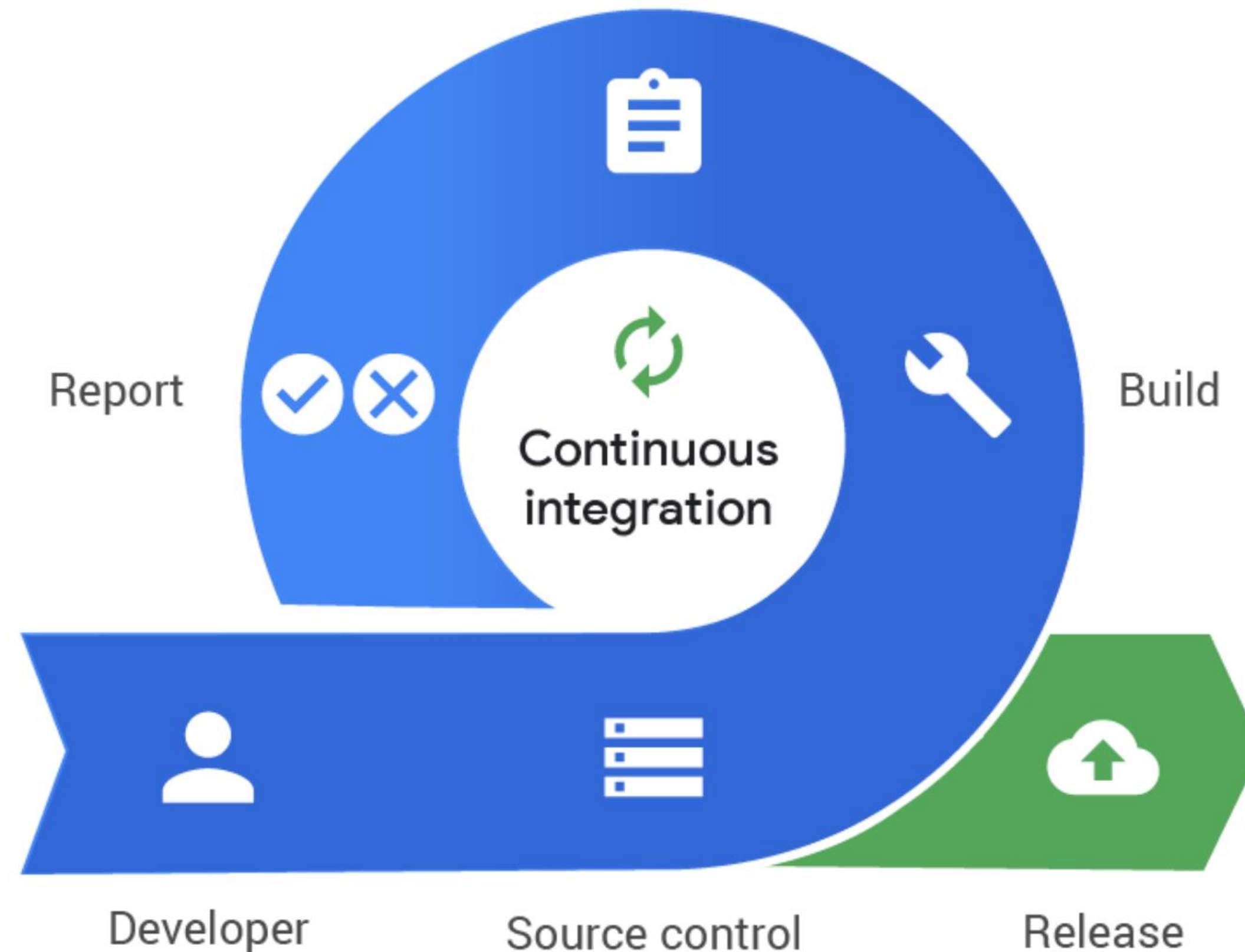


Continuous Integration / Integración Continua



Continuous Integration

- Continuous Integration es una práctica de desarrollo donde los miembros de un equipo integran su trabajo frecuentemente, generalmente una persona integra por lo menos diariamente.
- Cada integración es verificada por una herramienta de build automático (incluido un test) para detectar errores de integración tan pronto como sea posible



Esto anda si...(disciplina)

Mantenga un repositorio para el código donde todos los del equipo tengan acceso a la última versión y a las previas

Automatizar el proceso de build. Cualquiera puede lanzar un build

Automatizar el testing (smoke test). Cualquiera puede lanzar la suite de test cases

Todos tienen acceso al ejecutable que se cree que es el mejor ejecutable

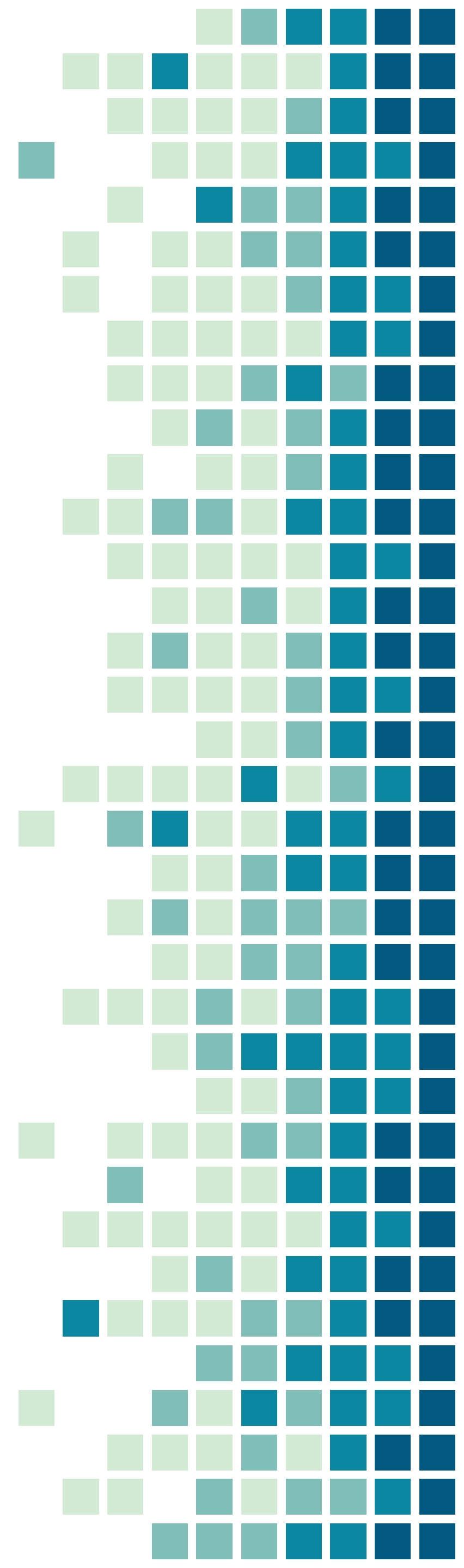
Se le dedica un tiempo de seteo y tiempo de mantenimiento

¿Qué es un build exitoso?

- Tener un build exitoso no es compilar todos los días!!!!
- La definición de “build exitoso” es un tanto agresiva.
 - Los fuentes deben estar en el repositorio (check operation)
 - Cada fuente se compila.
 - Los resultados de los object files son linkeados y se los deja listo para ejecución (jars)
 - Se arranca el sistema y se corre una suite de test cases.
 - Si todo esto se ejecuta sin errores y sin intervención humana, entonces hay un “build exitoso”
- Mientras más exhaustiva sea la suite de test cases más valor genera continuous integration.

Mas frecuente...

- Es mejor tener mayor frecuencia en las integraciones. Con equipos que nunca lo trabajaron es una contradicción para su experiencia previa
- Una clave para esto es la automatización



This figure displays a 20x20 grid of handwritten digits from the MNIST dataset. Each digit is represented by a small square, where the color intensity (ranging from light green to dark blue) corresponds to the pixel values of the digit. The digits are arranged in a regular grid, with each digit occupying a 10x10 pixel area. The background is white, and the digits are centered within their respective squares.

Coding Integration Testing Release

- [illegible]

- ❖ Coding y Unit Test
- ❖ Integraciones después de cada checking
- ❖ El código es constantemente pasado por un build y testado
- ❖ Hay posibles releases con funcionalidad incremental

Referencias

- Bersoff, E.H., "Elements of Software Configuration Management",
- IEEE Transactions on Software Engineering, vol 10, nro. 1, enero 1984, pp 79-87
- Little Book of Configuration Management – <http://www.spmn.com>
- SCM & the Agile Manifesto - <http://www.scmpatterns.com/agilescm/>

