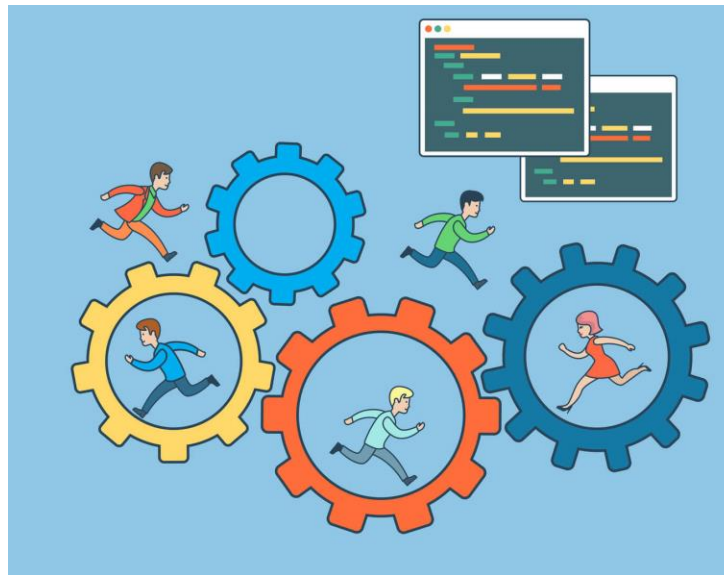


Bienvenidos a Orientación a Objetos 2



Orientación a Objetos 2

Introducción



En las teorías:

- Alejandra Garrido
- Federico Balaguer
- Gabriela Perez

En las explicaciones de práctica:

- Agustín Ortú
- Germán Ruiz
- Leonardo Loza Bonora
- Juan Cruz Gardey

En las consultas de práctica: Cristian, Diego, Felipe, Guadalupe, Ignacio, Luciana, Nero, Catalina, Martín, Lorena, Giuliana, Rafael, Mateo, Nathaniel, Camila

2 Bandas Horarias - todo presencial

- TEORIA
 - Lunes 8:30 a 10:30 hs - Aula 4
 - Lunes 16 a 18 hs. - Aula 5 (en mayo cambia el aula)
- HORARIOS DE PRACTICA
- Comienzan la semana del 17/3
- Explicación de práctica:
 - Lunes 18 a 19 hs: Aula 5
 - Martes 9 a 10 hs: Aula 10A
- Práctica:
 - 7 horarios de práctica disponibles (3 de mañana y 4 de tarde) donde deberán inscribirse esta semana

Objetos 2 - Aprobación de la materia

- Cursada:
 - se aprueba con un examen parcial, que tiene 2 recuperatorios. El parcial se aprueba por temas.
- Fechas de parcial:
 - 7 de junio
 - 28 de junio
 - 12 de julio
- Final de la materia:
 - Examen final
 - Promoción: se rinde el 12 de julio. Única fecha

Objetos 2 - Promoción

- Para promocionar la materia se requiere:
 - aprobar la cursada en 1ra o 2da fecha de parcial;
 - responder correctamente las preguntas extra que se agregan al parcial para acceder a la promoción;
 - aprobar un examen integrador de promoción que se rinde por única vez el 12/7/25 con calificación 6 o más.

- Todo el material estará disponible en la plataforma Cátedras
- Pero... no se queden solo con nuestras transparencias, es importante que estudien de los libros obligatorios de la materia:
 - Design Patterns. Elements of Reusable Objects Oriented Software. Gamma, Helm, Johnson, Vlissides, Addison-Wesley.
 - Refactoring: Improving the Design of Existing Code. Fowler, Martin. Addison-Wesley.
 - Refactoring to Patterns. Joshua Kerievsky. Addison Wesley.
 - Implementing Application Frameworks: Object-Oriented Frameworks at Work. Fayad, Schmidt, Johnson (*eds*).

Atención de estudiantes

ACCESIBILIDAD	A. Paola Amadeo	accesibilidad@info.unlp.edu.ar pamadeo@info.unlp.edu.ar
ORIENTACIÓN AL ESTUDIANTE	Ana Ungaro	orientación.estudiantil@info.unlp.edu.ar anaungaro@info.unlp.edu.ar
GÉNERO	Sofía Martin	ddhhygenero@info.unlp.edu.ar smartin@info.unlp.edu.ar
ASESORAMIENTO PEDAGÓGICO	Anahí Almán	direccion.pedagogica@info.unlp.edu.ar

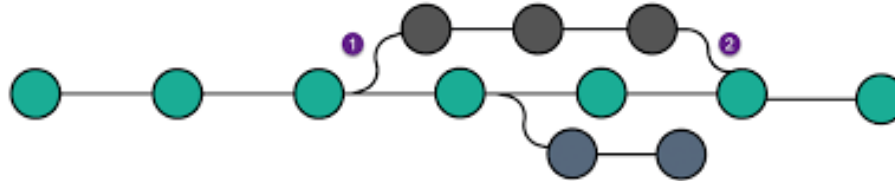
- **Métodos ágiles**

- se desarrolla en iteraciones
- cada iteración extiende código existente con nueva funcionalidad
- poca documentación
- testing automático e integración continua



- Arquitecturas complejas (arquitecturas modulares, capas con distintos concerns: interfaz de usuario, persistencia, seguridad, ...), arquitecturas orientadas a servicios (conexión con código ajeno, desarrollo de APIs, ...)
- El cambio es continuo, el crecimiento es continuo, la complejidad es cada vez mayor

Contexto: herramientas para el desarrollo



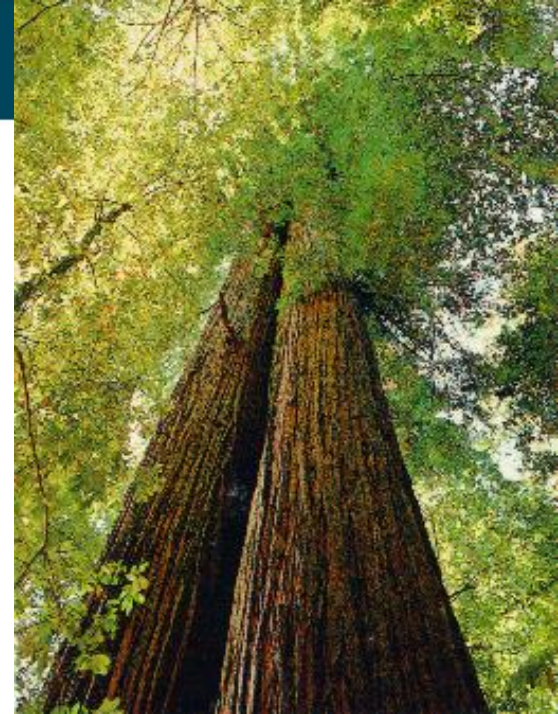
- Version Control Systems: Branching & merging, DevOps
 - qué implica?
 - entender código existente (50% del tiempo de mantenimiento)
 - extender código existente
 - code review antes de cada merge
 - entrega continua de alta calidad
- Herramientas de IA: Copilot, ChatGPT, Gemini, etc.
 - qué implica?
 - una gran ayuda para tareas rudimentarias
 - necesitamos desarrollar pensamiento crítico

- Estar mejor preparados para el cambio, porque:
 - El ritmo del cambio en la tecnología y en la demanda de inmediatez de los usuarios crece **exponencialmente**
- Desarrollar la capacidad de cambiar el software de manera **rápida y sustentable**
 - consciente, evitando el desperdicio de recursos, reconociendo oportunidades, responsable, con calidad, buscando la mejora continua



Diseñar es difícil!

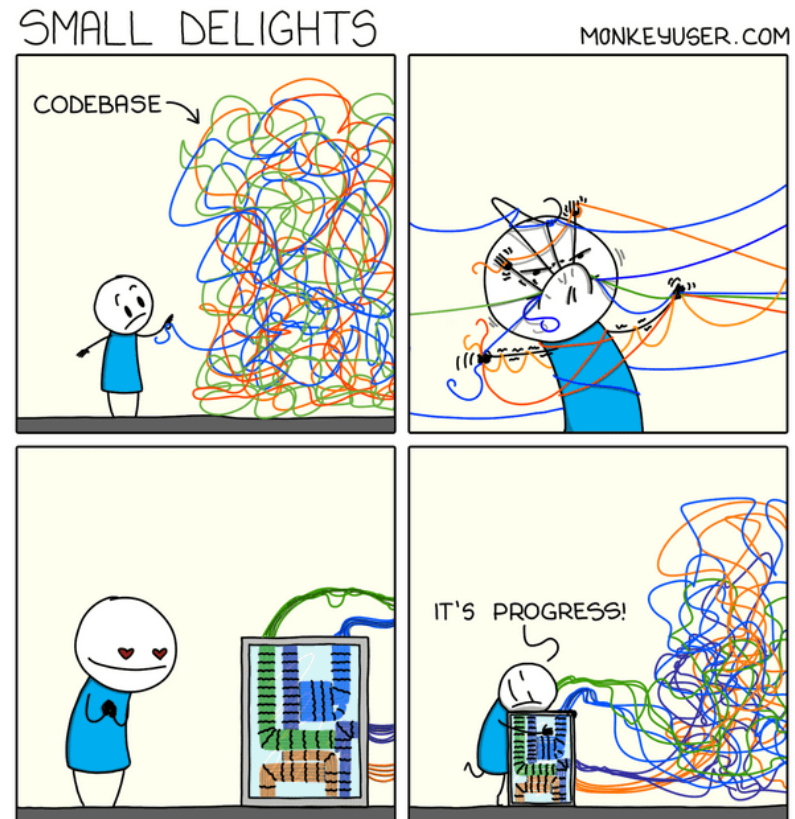
- Los elementos distintivos de la arquitectura de un sistema no surgen hasta *después* de tener código que funciona
- No se trata sólo de agregar, sino de *adaptar, transformar, mejorar*
- Recurriendo a la experiencia de diseñadores que han documentado y compartido sus diseños, que han sido revisados y catalogados



Qué vamos a aprender en Objetos 2

- Reconocer las fuerzas que llevan al deterioro de la arquitectura y aprender a reconocer oportunidades para mejorarla
- Con el objetivo de lograr código legible y extensible
- En cada iteración
- Paso a paso

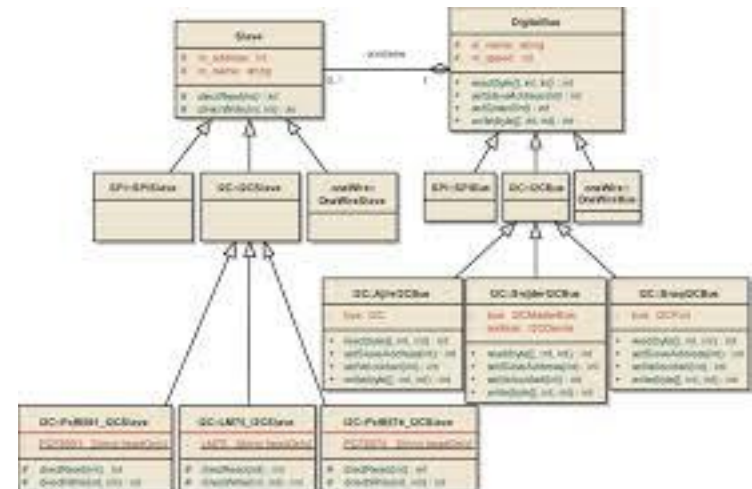
REFACTORING.



Qué más vamos a aprender en Objetos 2

- Construir diseños reusables, más sensibles a la evolución
- Reconocer las consecuencias positivas y negativas de un diseño (fundamentar y criticar diseños)
- Hacer evolucionar diseños, aplicando mejoras paso a paso

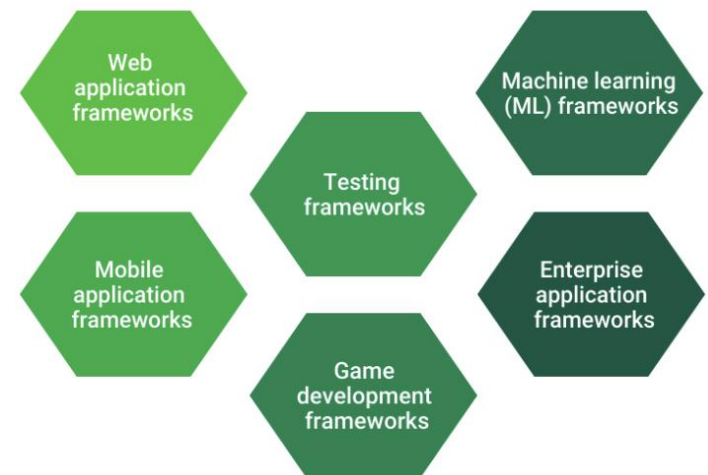
PATRONES DE DISEÑO



Qué otra cosa vamos a aprender en Objetos 2

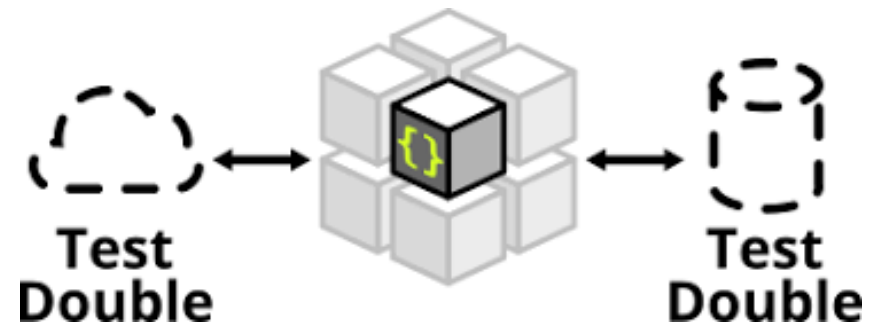
- Instanciar arquitecturas reusables para una familia de aplicaciones, reconociendo puntos de variabilidad y extensión de estas arquitecturas
- Entender las implicancias de utilizar estas arquitecturas
- Diseñar y programar estas arquitecturas

FRAMEWORKS.



Buscando calidad en lo que hacemos

- Importancia del testing
- Tipos de Tests
- Test Driven Development
- Estrategias de testing con test doubles



En resumen

- Es muy importante tener la materia al día
- Van a tener el material en Cátedras, para que les sirva de repaso. Las transparencias pueden estar incompletas y no tienen la función de reemplazar las teorías y explicaciones en el aula
- Es fundamental:
 - participar de las teorías y las explicaciones de práctica
 - leer la bibliografía que se les indica para cada tema
 - hacer los ejercicios de práctica y venir a las consultas a presentar y fundamentar sus diseños