



# Programación 4

InCo, FING, UdelaR



# Datos del curso

- Entorno Virtual de Aprendizaje (**EVA**)

<https://eva.fing.edu.uy/course/view.php?id=413>

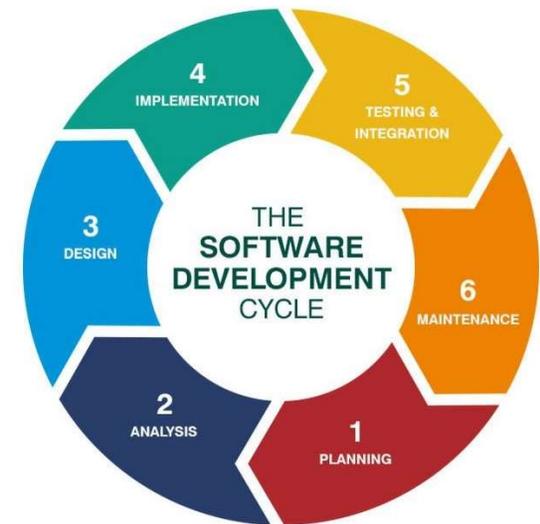
- Docentes

- Luis Costela
- Matías Esmoris
- Miguel Machado
- Raul Maglione
- Antonio Mauttone (Responsable de la Unidad Curricular)
- Leonel Peña
- Sebastián Pizard

# Motivación

La **Ingeniería de Software (IS)** es la disciplina relacionada con la construcción de forma efectiva y eficiente de sistemas de software confiables que satisfagan los requisitos de los clientes y usuarios.

Acompaña todas las fases del ciclo de vida del software, desde su concepción hasta su operación y evolución.



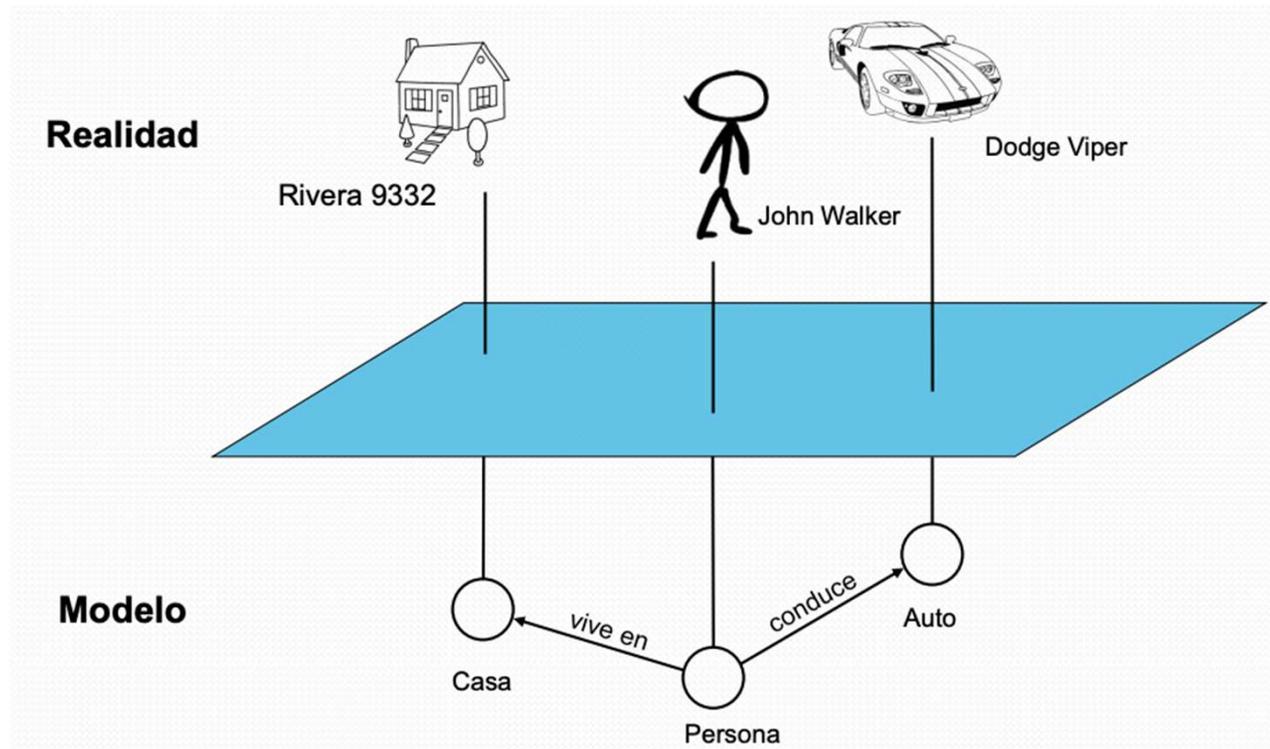
# Motivación

La **Orientación a Objetos (OO)** es un **paradigma de IS** que se enfoca en la resolución de un problema computacional como un conjunto de **objetos** (abstracciones del mundo real) que **interactúan** entre sí por medio de mensajes.



# Motivación

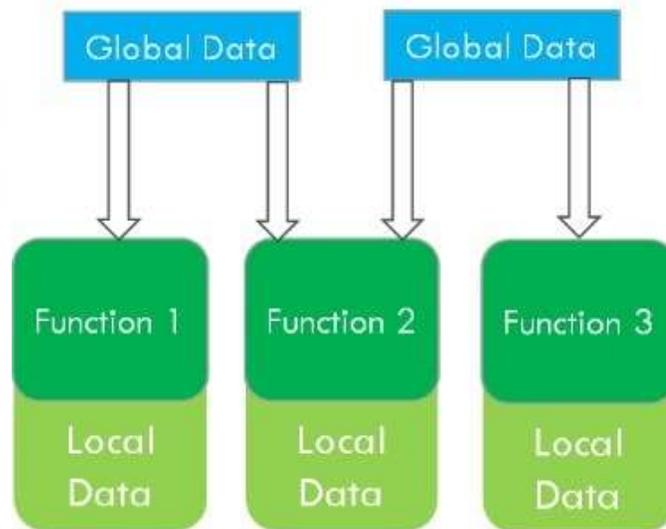
Plantea una forma de **concebir la realidad...**



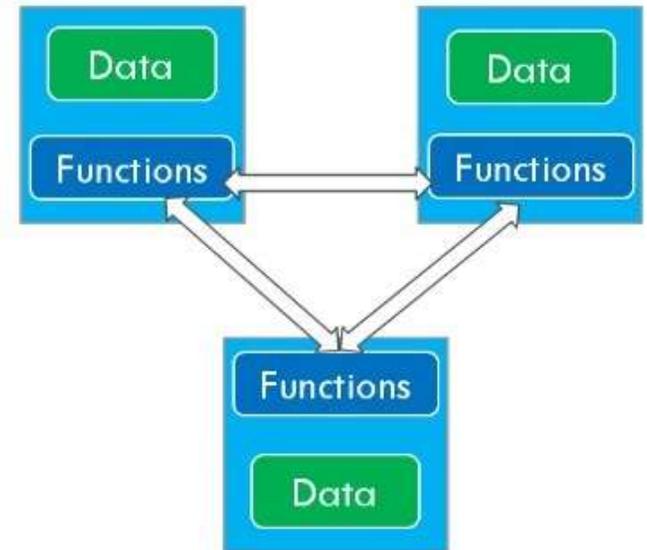
# Motivación

... y de plantear una **solución computacional**...

Procedural Oriented Programming

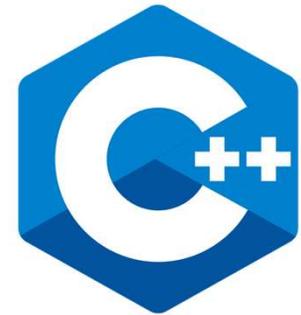
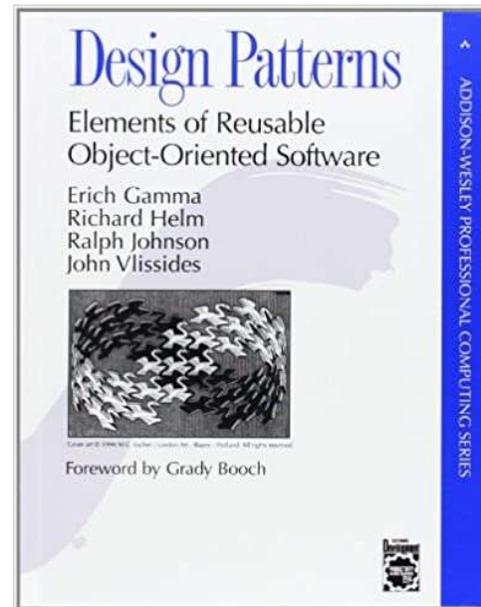


Object Oriented Programming



# Motivación

... utilizando **nuevas herramientas.**





# Objetivos

Introducir elementos necesarios para la **construcción de sistemas de software medianos y grandes.**

Presentar y aplicar para ello **conceptos de la orientación a objetos.**

# Objetivos específicos

- Dar a conocer **herramientas conceptuales** para el **análisis** y **diseño** de sistemas orientados a objetos.
- Presentar una **metodología básica** para el uso de dichas herramientas.
- Dar a conocer un **lenguaje de programación** que permita expresar los conceptos involucrados en la orientación a objetos.
- Adquirir experiencia en ese lenguaje para **poner en práctica** los conceptos anteriores.



# Temario

## 1. Conceptos básicos de OO

## 2. Análisis OO

Modelado del dominio

Comportamiento del sistema

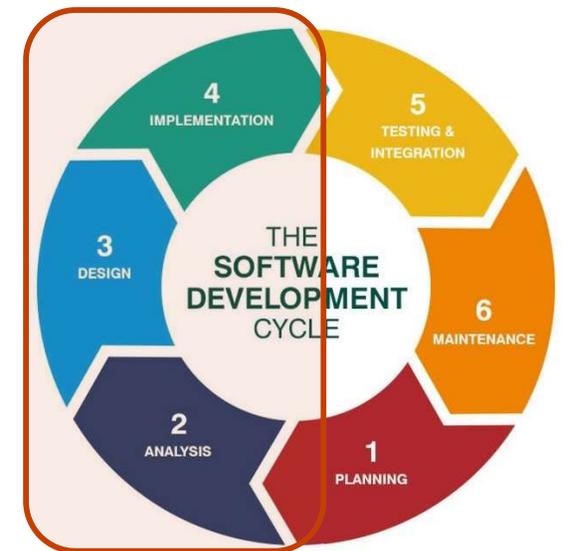
## 3. Diseño OO

Diseño de interacciones

Diseño de estructura

## 4. Implementación OO

Codificación





# ¿Y después?

Taller de Programación (Sem. 6)

Profundización de P4 + Testing + Java

Introducción a la IS (Sem. 7)

Visión general de IS

Proyecto de IS (Sem. 8)

Aplicación práctica de Intro. a la IS

Otros: Fundamentos de Bases de Datos

Taller de Sistemas Empresariales

# Cronograma

Semana	Teórico	Práctico	Laboratorio
1 (4/3)	01 - Introducción 02 - Desarrollo OO 03 - Conceptos Básicos de OO		
	03 - Conceptos Básicos de OO		
2 (11/3)		1	Lab 0
3 (18/3)		1	Lab 0
<b>Turismo (25/4, una semana)</b>			
4 (1º/4)	04 - Requerimientos e Introducción al Análisis 05 - Análisis: Modelado del Dominio	2	Lab 1
5 (8/4)		2	Lab 1
6 (15/4)	06 - Análisis: Comportamiento del Sistema	3	Lab 2
7 (22/4)		3	Lab 2
<b>Parciales (29/4, dos semanas)</b>			

# Cronograma

<b>8 (13/5)</b>	09 - Diseño: Introducción al Diseño 10 - Diseño: Diagramas de Comunicación	4	Lab 3
<b>9 (20/5)</b>	11 - Diseño: GRASP 12 - Diseño: Visibilidad	4	Lab 3
<b>10 (27/5)</b>	13 - Diseño: Guías 14 - Diseño: Estructura	5	Lab 3
<b>11 (3/6)</b>	15 - Diseño: Patrones de Diseño 16 - Implementación: Generación de Código 17 - Implementación: Manejo de Objetos 07 - Generación Parcial de Código	6	Lab 4
<b>12 (10/6)</b>		6	Lab 4
<b>13 (17/6)</b>			Lab 4
<b>14 (24/6)</b>	Defensa grupal e individual		
<b>Parciales (comienzan el 4/7, Programación 4 es el 6/7)</b>			



# Metodología de enseñanza

**Teórico/práctico** :: **Clases virtuales asincrónicas**. Los lunes se publicará una **guía de estudio**.

**Laboratorio** :: **proyecto grupal (5 integrantes por grupo)** y seguimiento a través de **monitoreos semanales obligatorios (modalidad virtual sincrónica)**, en horario a definir.

- Lab 0: Conceptos básicos de OO y C++
- Lab 1: Análisis, aspecto estático
- Lab 2: Análisis, aspecto dinámico
- Lab 3: Diseño
- Lab 4: Implementación en C++

Los **materiales y guías se encuentran publicados** en EVA y OpenFING (<https://open.fing.edu.uy/courses/p4>).



# Evaluación

La evaluación consta de dos partes: evaluación del laboratorio y evaluación del curso.

Las evaluaciones serán presenciales.

Ítem	Puntaje Máximo
Parcial individual único	85
Laboratorio grupal	10
Laboratorio individual	5

Se requiere un **mínimo en la evaluación grupal e individual del laboratorio** y un **mínimo de 60 en el total para exonerar** (25 para aprobar).

# Ambiente de trabajo

Es beneficioso crear un ambiente de trabajo propicio para el desarrollo del curso, tanto **tecnológico** como **humano**.

Un posible ambiente **tecnológico**:

- GCC (compilador) / valgrind / make
- Visual Studio Code (IDE)
- GitLab (control de versiones)
- StarUML (modelador)



# Las 5 "C" del trabajo en equipo

Se dice que un equipo de trabajo funciona como una maquinaria, con diversos engranes; todos deben funcionar a la perfección pues si uno falla, el equipo fallará. De acuerdo a las aportaciones de **Tom Peters**, estas son las 5C para mejorar el trabajo en equipo.



## COMPROMISO

Cada miembro del equipo se compromete a aportar lo mejor de sí mismo y a poner todo su empeño en realizar todas las tareas propuestas.



## CONFIANZA

Cada miembro del equipo confía en el buen hacer del resto de sus compañeros(as) y en la consecución del objetivo a lograr.



## COMPLEMENTARIEDAD

Cada miembro del equipo domina una determinada habilidad y/o conocimiento de tal manera que entre todos se complementan para sacar adelante el trabajo a realizar.



## COORDINACIÓN

El equipo "asume" una coordinador@ que ayuda a actuar de forma organizada en cuanto a la gestión de tiempos y tareas con la visión de sacar el proyecto adelante.



## COMUNICACIÓN

Los miembros del equipo se deben comunicar abiertamente, para comprender, analizar, realizar y tomar decisiones que les permita coordinar las actuaciones individuales y en grupo.

C+ TALENTO HUMANO

# Normas

## PARA TRABAJAR EN EQUIPO



### SER EDUCADO

Usa palabras como "por favor" y "gracias"



### COMUNICARSE

Siempre pide la palabra, la comunicación es respeto



### PROPONER

Entre más ideas, más inspiración



### COLABORAR

Usa las fortalezas de tus compañeros



### ACEPTAR

Respetar las ideas del resto



### CUMPLIR NORMAS

Trabajar en equipo es un reto grupal.



# Honestidad académica

Las interacciones benefician el dominio de los temas del curso, pero hay una línea clara entre obtener ayuda de otros y aprovecharte de su trabajo.

Es razonable comunicarte con compañeros de clase, solicitar ayuda a los docentes, discutir el material para comprenderlo mejor, buscar materiales en la web e incluso colaborar con compañeros para discutir soluciones a los efectos de identificar problemas y fortalecer los resultados.

Sin embargo, **la esencia de todo trabajo que entregues, debe ser tuya o de tu grupo, según el tipo de actividad**, por lo que no está permitido colaborar en la respuesta de las evaluaciones, utilizar una solución de otros, ni brindar una solución a otros.

**Al inscribirte al curso asumes el compromiso, responsabilidad y respeto que implica para contigo, tus compañeros y los docentes del curso.**



# Bibliografía

1. C. Larman. **Applying UML and patterns**. Prentice Hall (2ª Ed. 2001). ISBN 978-013-092-569-5. Nota: este libro utiliza una versión 1.x de UML
2. M. Fowler. **UML Distilled**. Addison Wesley (3ª Ed. 2003) - ISBN 978-032-119-368-1. Nota: este libro utiliza la versión 2.0 de UML
3. E. Gamma, R. Helm, R. Johnson, J. Vlissides. **Design Patterns**. Addison-Wesley (1995) - ISBN 0-201-63361-2
4. H.M. Deitel y P.J. Deitel. **Cómo programar en C/C++**. Prentice Hall (1995) - ISBN 968-880-471-1



*Bienvenidos*