



# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela

## **Diseño de Software**

1. Espacio de diseño
2. Diseño en el proceso de desarrollo de software
3. Captura de decisiones de diseño
4. Intercambio de decisiones de diseño

# Motivación

# Motivación

---

- El impacto “**negativo**” de los sistemas de Software.

# Diseño de Software

# Diseño de Software

---

- Es la construcción de abstracciones de datos y la organización de estas en una aplicación de software que funcione.
- Consideramos abstracción a variables, classes, objects, etc.
- No olvidar: usamos la abstracción para reducir la complejidad de problemas, en un dominio determinado, para alcanzar un propósito específico.

# Diseño de Software en la práctica

---

- En la práctica, el diseño de software es esencialmente un proceso de toma de decisiones.
- Por ejemplo:
  - ¿Debería usar una Lista o una Pila?
  - ¿Qué servicios debería usar determinada interface?
  - ¿Dónde se debe manejar determinado error?

# **Espacio de diseño**



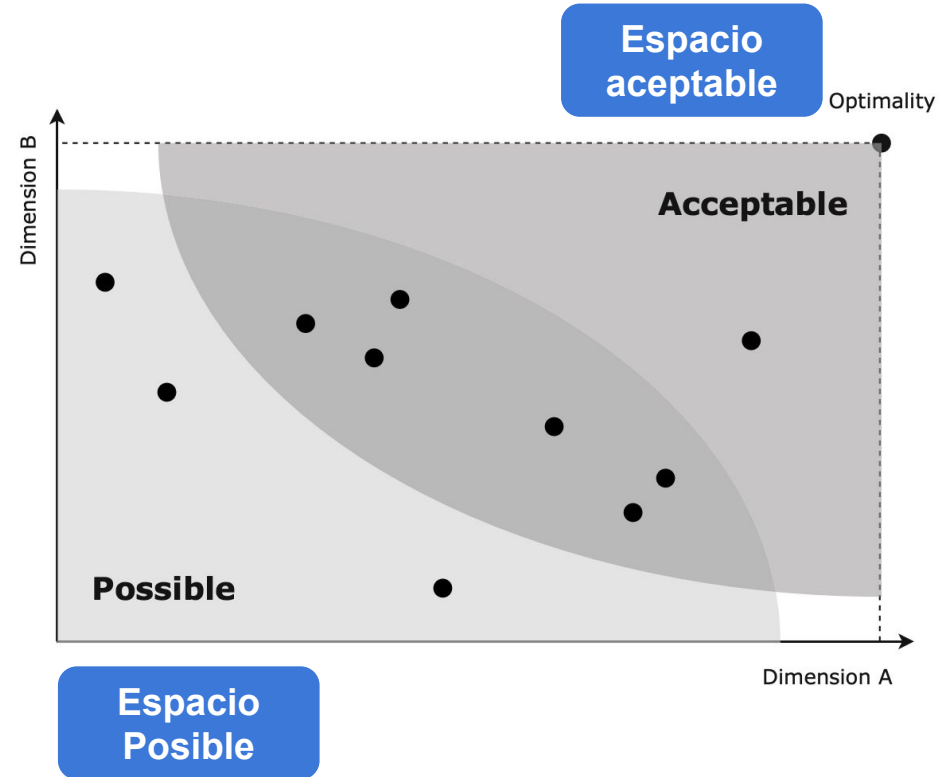
# Diseño de Software en la práctica

---

- El **espacio de diseño** se puede imaginar como un espacio geométrico  $n$ -dimensional donde cada dimensión corresponde a un atributo de calidad de diseño.
- Entre los atributos de calidad de diseño típicos para el software se incluyen: comprensibilidad, reutilización y facilidad de implementación.
- Dentro de dicho espacio de diseño, cada decisión de diseño específica (o conjunto coherente de decisiones) corresponde a una coordenada en el espacio que representa la consecuencia de la decisión.

# Diseño de Software en la práctica

- La figura ilustra la idea en dos dimensiones.
- En la práctica, cualquier decisión de diseño es buena en alguna dimensión, pero no tan buena en otra dimensión, lo que determina que en las decisiones de diseño siempre se debe transar algo.



# Diseño de Software en la práctica

---

- Todo esto, para decir que el diseño es un proceso de decisión para el cual rara vez hay una única "respuesta correcta".
- Lo que se tiene en realidad son soluciones que son mejores o peores en algunas dimensiones (incluidas algunas soluciones que son bastante malas en la mayoría de las dimensiones).
- El diseño es un conjunto de decisiones.

# Diseño de Software en la práctica

---

- El concepto de espacio de diseño puede sugerir, “erróneamente”, que una decisión de diseño es un proceso sistemático, casi como un proceso matemático.
- Es decir, el proceso en que se decide una decisión de diseño no es algo que se pueda formalizar matemáticamente.
- Entonces, ¿existe algún procedimiento?
  - El diseño es un proceso altamente heurístico: consiste en la solución iterativa de resolución de problemas guiado por la experiencia, principios generales y técnicas de diseño.

# **Diseño en el Proceso de Desarrollo de Software**

# Proceso de Desarrollo de Software

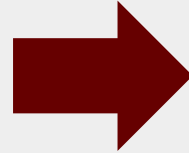
---

- Especificación de requerimientos - Desarrollo - Validación - Evolución
- La Ingeniería de software es la disciplina que se ocupa de que el proceso de desarrollo de software sea desarrollado de manera sistemática, disciplinada y medible

# Diseño en el Proceso de Desarrollo de Software

---

- Prácticas de desarrollo de software:
  - control de versiones
  - programación de pares
  - uso de convenciones de código
  - refactorización



**Diseño de Software**

# Proceso de Desarrollo de Software

---

- ¿Por qué refactorizar?
  - porque el desarrollador original no lo hizo bien
  - necesidad de agregar nuevas funcionalidades que no se ajustan al diseño original
  - Reducir debilidades de diseño acumuladas → Deuda técnica
- Por lo tanto, el diseño de software está en evolución continua.



# Captura de decisiones de diseño

# Proceso de Desarrollo de Software

---

- **Código fuente:**
  - seleccionando una estructura de datos.
  - comentarios en el código
- **Documentos**
  - Documentación del software
  - Diagramas UML
- **Control de versiones y plataformas de discusión**
  - commits, listas de correo, slack
- **Modelos especializados**
  - Desarrollo dirigido por modelos

# **Intercambio de conocimiento de diseño**

# Proceso de Desarrollo de Software

---

- **Patrones de diseño**

- Patrón MVC, Estrategia

- **Antipatrones de diseño**

- Código duplicado, Métodos largos, Code/Bad Smells

# Referencias

# Referencias

---

- Robert, C. (2012). Código limpio: Manual de estilo para el desarrollo ágil de software. Anaya Multimedia.



# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela