

# Análisis y diseño de aplicaciones I



- Profesores responsables
  - Fernando Machado
  - José Abadie
  - Ana Bergallo / Bettina Carrizo
  - Javier Yannone / Rodrigo Lujambio
- Profesores ayudantes
  - Sebastian Feirrés
  - Fabiana Costanian
  - Juan Serra

# “ANDIS 1” - Intenciones

- **Primarias:**
  - Desarrollar la capacidad de analizar, relevar y especificar requerimientos.
  - Diseñar modelos de software para cumplir con esos requerimientos.
  - Formalizar en el conocimiento de patrones arquitectónicos sencillos.
- **Continuas:** contribuir a desarrollar valores y habilidades para el *trabajo en equipo*, la *responsabilidad por el aprendizaje propio*, la *resolución de problemas*, el *pensamiento crítico*, la *comunicación efectiva* y el *aprendizaje permanente*.

# Resultados de aprendizaje



- Identificar y documentar requerimientos funcionales y no-funcionales de software aplicando una técnica de relevamiento de requerimientos, en sesiones de trabajo con partes interesadas y utilizando habilidades de facilitación, como parte de un equipo de análisis de requerimientos de software
- Analizar la consistencia, completitud y factibilidad de los requerimientos funcionales y no-funcionales de un sistema informático, recomendando mejoras a la especificación
- Especificar requerimientos funcionales y no-funcionales de software utilizando lenguajes de especificación estándar seleccionados para un proyecto, y describir los requerimientos de forma apropiada a una audiencia de partes interesadas
- Verificar y validar requerimientos funcionales y no-funcionales usando técnicas estándar, incluyendo inspección, modelado, prototipado, y desarrollo de casos de prueba
- Producir un diseño de alto nivel a partir de requerimientos funcionales que se pueda presentar a una audiencia no técnica considerando patrones arquitectónicos y de diseño
- Producir diseños detallados a partir de diseños de alto nivel utilizando principios de diseño y aspectos transversales para satisfacer requerimientos funcionales y no-funcionales
- Presentar a partes interesadas el diseño de un sistema de software simple usando una notación de modelado, incluyendo una explicación de cómo el diseño incorpora principios de diseño de sistemas y cómo soluciona los requerimientos arquitectónicamente significativos
- Evaluar y comparar los compromisos de posibles diseños alternativos para satisfacer requerimientos funcionales y no-funcionales y escribir una propuesta resumiendo las conclusiones para las partes interesadas
- Evaluar consideraciones de prueba de los atributos de calidad en el diseño de subsistemas o módulos para un desarrollador

# Estructura general del curso y metodología



- “TBL”
- Unidades Temáticas
- Proceso de aseguramiento de la preparación
- Bibliografía, guías de lectura, ejercicios
- Trabajo en equipos en clase
- Preparación individual *fuera de clase*
  - Consultas, foros, recursos de información

# Tipo de asignatura y dedicación requerida

- Asignatura del tipo **“B”**. **Del Plan de Estudios:**  
“Para el logro de los resultados de aprendizaje, el estudiante tiene que dedicar tiempo fuera del aula para realizar trabajos de campo, realizar proyectos o productos, tareas puntuales en laboratorios o estudios. Se basan en metodologías activas -ABP, ABE, u otras-, evaluación continua y trabajo colaborativo. **Las actividades fuera del aula requieren un esfuerzo importante del estudiante, son planificadas y supervisadas por el docente y esto queda claramente reflejado en la planificación de clases presentada en el programa del curso.”**
- **12 créditos, 85.2 horas clase, dedicación *mínima estimada* necesaria *fuera de clase*: 2 horas por cada hora de clase**

# Evaluación de alcance de los resultados del aprendizaje

La calificación del curso se compone de:

- El **componente individual**, integrado por:
  - Las pruebas de aseguramiento de la preparación individuales (de cada Unidad Temática).
  - Las evaluaciones parciales.
- El **componente de equipo**, compuesto por:
  - Las pruebas de aseguramiento de la preparación en equipo de cada Unidad Temática.
  - Los ejercicios de aplicación (calificados en forma unificada en cada Unidad Temática).
- **Concepto cualitativo**, compuesto por: Actitud en clase frente al docente y los compañeros, compromiso, cumplimiento de tareas domiciliarias, defensas, exposiciones, portafolios.

# Composición de calificaciones

COMPONENTE	ponderación	
<b>COMPONENTE INDIVIDUAL</b>		<b>60%</b>
Parciales	70	
iRATs	30	
<b>COMPONENTE DE EQUIPO</b>		<b>30%</b>
tRATs	40	
Trabajos de Aplicación	50	
iPeer	10	
<b>ACTITUD ACADÉMICA</b>		<b>10%</b>
<b>Total Calificaciones del Curso</b>		<b>100%</b>



# Evaluación de alcance de los resultados del aprendizaje



- Para lograr la **aprobación** del curso, es necesario que se cumplan **tres** condiciones **simultáneamente**:
  - Obtener una calificación de **60%** o superior en el componente individual (36 puntos del total/100).
  - Obtener una calificación de **75%** o superior en el promedio ponderado de los tres componentes
  - Verificar una asistencia no menor al **75%** de todas las clases efectivamente dictadas.

# Escala de notas UCU



- S: 96-100
- MB: 91-95
- BMB: 83-90
- B: 75-82
- R: 41-74
- D: 0-40

- Unidad 1 - Proceso y elicitación de requerimientos 14/03 al 02/04.
- Unidad 2 - Análisis y especificación 04/04 al 18/04.
- **PRIMER PARCIAL 23 DE ABRIL**
- Unidad 3 - Diseño de software 25/04 al 09/05.
- Unidad 4 – Interfaz de usuario 14/05 al 04/06.
- **SEGUNDO PARCIAL 28 DE MAYO**
- Unidad 5 – Implementando patrones de diseño 06/06 al 20/06.
- Unidad 6 – Implementando patrones de arquitectura 25/06 al 27/06.
- **TERCER PARCIAL 02 DE JULIO**

# Condiciones generales

- Horarios y asistencia.
  - Martes y jueves de 18:15 a 21:05
  - Asistencia por módulo
  - Pruebas individuales y en equipo
  - En clase, computadoras, asistencia
- Trabajos de aplicación en equipos y ejercicios domiciliarios
  - Trabajos en equipos, asistencia
  - Preparación fuera de clase
- Webasignatura canal oficial. Si prefieren otro canal **SECUNDARIO**:
  - *Discord* <https://discord.gg/jB2kycsx>

# Honestidad académica



## Utilización de Inteligencia Artificial

Se espera que utilicen herramientas de Inteligencia Artificial (AI, ChatGPT y herramientas de generación de imágenes por ejemplo), siempre como complemento y nunca como fuente primaria de información. Aprender a usar estas herramientas correctamente es una habilidad emergente, cada vez más necesaria. El alumno será responsable por los errores u omisiones en los contenidos generados por la herramienta.

Se deberán considerar las limitaciones de ChatGPT y herramientas similares. Se deberá verificar siempre la validez de las respuestas y siempre que alguna de ellas se utilice en un trabajo, entrega o presentación. Se deberá siempre incluir el prompt utilizado en cada caso y la fecha de consulta.

En las instancias de evaluación individual, la utilización de estas herramientas estará explícitamente prohibida. El Reglamento General de Estudiantes, numeral 7.7.2 h) considera como falta grave el fraude o intento de fraude en pruebas, trabajos académicos o cualquier instancia de evaluación del desempeño académico

# Uso adecuado IA



- <https://hbsp.harvard.edu/inspiring-minds/4-simple-ways-to-integrate-ai-into-your-class>

- Mastering the requirements process – Susan & James Robertson
- Service design patterns, Fundamental Design solutions for SOAP and RESTFul – Robert Daigneau
- Don't make me think – Steve Krug 3rd Edition.
- Design Patterns – Gamma
- <https://www.geeksforgeeks.org/>
- <https://refactoring.guru/>
- <https://www.nngroup.com/>
- Wikipaida

¿Preguntas?





# Encuesta equipos:



- <https://forms.gle/1EPFcuuujrDLSH197>