



**INSTITUTO/S:** Tecnología e Ingeniería

**CARRERA/S:** Licenciatura en Informática / Tecnicatura Universitaria en Informática

**MATERIA:** Desarrollo de Aplicaciones

**NOMBRE DEL RESPONSABLE DE LA ASIGNATURA:** Carlos Lombardi

**EQUIPO DOCENTE:** Christian Schiffino

**CUATRIMESTRE:** Licenciatura en Informática: 1<sup>ro</sup> / Tecnicatura Universitaria en Informática: 5<sup>to</sup>

**AÑO:** Licenciatura en Informática: 3<sup>ro</sup> / Tecnicatura Universitaria en Informática: 3<sup>ro</sup>

**PROGRAMA N°:** Licenciatura en Informática: 27 / Tecnicatura Universitaria en Informática: 20

**(Aprob. Por Cons.Directivo 04/06/2022)**

**Instituto/s:** Tecnología e Ingeniería

**Carrera/s:** Licenciatura en Informática/Tecnicatura Universitaria en Informática

**Nombre de la materia:** Desarrollo de Aplicaciones

**Responsable de la asignatura y equipo docente:** Carlos Lombardi / Christian Schiffino

**Cuatrimestre y año:** Licenciatura en Informática: 1<sup>ro</sup> del 3<sup>er</sup> año /Tecnicatura Universitaria en Informática: 5<sup>to</sup> del 3<sup>er</sup> año

**Carga horaria semanal:** 6 horas

**Programa N°:** Licenciatura en Informática: 27 /Tecnicatura Universitaria en Informática: 20

**Código de la materia en SIU:** 763

## **Desarrollo de Aplicaciones**

### **1. Fundamentación**

Esta materia permite a los/as alumnos/as profundizar e integrar los conocimientos adquiridos en materias anteriores como Elementos de Ingeniería de Software mediante la construcción de una aplicación pequeña partiendo desde una versión preliminar de los requerimientos y llegando a un prototipo operativo utilizando una metodología ágil muy actual en el ámbito profesional.

### **2. Propósitos y/u objetivos**

#### **Objetivos**

Que el /la estudiante:

- Lleve a cabo la construcción completa de una aplicación pequeña, partiendo desde una versión preliminar de los requerimientos y llegando a un prototipo operativo.
- Aplique en una experiencia concreta los conocimientos sobre ciclo de desarrollo y metodologías adquiridos en Elementos de Ingeniería de Software.
- Comprenda los beneficios de respetar los principios y prácticas adquiridos en materias anteriores para llevar a cabo un proyecto de desarrollo de software.
- Adquiera una noción inicial de qué es una arquitectura de software, y qué consecuencias trae trabajar con una determinada arquitectura en el desarrollo.
- Adquiera familiaridad con herramientas tecnológicas y metodológicas (ciclo de vida ágil) adecuados para el desarrollo grupal.

### **3. Programa sintético:**

- Herramientas metodológicas para trabajo en grupo. División de tareas planeando la reunión de los resultados. Definición de componentes con interfaces claras. Definición y gestión de componentes genéricos.
- Herramientas tecnológicas para trabajo en grupo. Repositorio de código, manejo de branches, integración. Repositorios de bibliotecas. Gestor de tareas e incidentes, estadíos en la evolución de una tarea. Prototipador gráfico.
- Nociones iniciales sobre arquitectura de software: qué parámetros definen una arquitectura, qué es una decisión de arquitectura, impacto en el desarrollo de trabajar bajo una cierta arquitectura.
- Aplicación concreta de las ideas de desarrollo iterativo, ceremonias de los procesos de desarrollo ágil, planificación y retrospectiva de un sprint.
- Relevamiento de requerimientos, interacción con expertos del dominio y/o usuarios.

#### **4. Programa analítico**

##### **4.1 Organización del contenido:**

UNIDAD 1: Metodología para el desarrollo grupal. Revisión de las ideas principales de las metodologías ágiles. Desarrollo de un sprint, planificación, selección de tareas. Seguimiento durante el desarrollo, reuniones periódicas. Presentación final del sprint, retrospectiva, reorganización de tareas. Rol del gestor de tareas e incidentes.

UNIDAD 2: Relevamiento. Relación con expertos del dominio y/o usuarios, su integración en el proceso de desarrollo. Herramientas de comunicación: prototipación gráfica, presentación de versiones incrementales. Especificación y priorización de relevamientos.

UNIDAD 3: Nociones básicas de arquitecturas de software. Marco que provee la arquitectura en un proyecto de desarrollo. Consecuencias de una decisión arquitectural. Arquitecturas Web, frontend, backend, separación, responsabilidades de cada uno, estrategias para comunicarlos.

UNIDAD 4: Desarrollo colaborativo. Revisión de conceptos sobre repositorios de código. Trabajo en branches, organización de branches, integración. Análisis del estado de un repositorio de código.

UNIDAD 5: Desarrollo front-end. Repaso de conceptos de front-end Web: generación dinámica de HTML, navegación entre páginas, gestión del estado de navegación, distintos mecanismos de interacción con el usuario. Frameworks/librerías básicas: Angular, React, VueJS.

UNIDAD 6: Desarrollo back-end. API REST, definición de endpoints, gestión de pedido y respuesta. Lenguajes de intercambio de información: XML / JSON / YAML. Integración con ORM

##### **4.2 Bibliografía y recursos obligatorios:**

**Hunt, A y Thomas, D.** (1999). *Pragmatic programmer: from journeyman to master*. USA: Addison Wesley.

**Martin, R. C.** (2008). *Clean code: A Handbook of Agile Software Craftsmanship*. 1<sup>ra</sup> ed. USA: Prentice hall.

**Alaimo, M.** (2021). *Scrum y algo más - un framework y muchos aprendizajes para creadores ágiles*. USA: MTN LABS LLC

#### **4.3 Bibliografía optativa:**

##### **Tutoriales:**

Elementos principales de JavaScript, Mozilla Development Network. Disponible en <https://developer.mozilla.org/en-US/docs/Web/JavaScript> .

Tutorial inicial de React. Disponible en <https://reactjs.org/tutorial/tutorial.html> .

Tutorial GIT. Disponible en <https://www.atlassian.com/es/git/tutorials> .

#### **5. Metodología de enseñanza:**

En paralelo al desarrollo de contenidos teóricos se irá implementando una aplicación definida al inicio de la cursada. El trabajo será en grupos organizados como equipos de desarrollo adoptando un ciclo de desarrollo ágil.

##### **Plan de trabajo en el campus:**

La materia posee un sitio (<https://unahur-desapp.github.io/unahur-desapp-site/>) que incluye el material y la información sobre actividades. Las entregas se hacen mediante repositorios de código fuente y documentos compartidos en un Google Drive, de acuerdo a las dinámicas características del campo profesional que son justamente uno de los objetos principales de la materia. Razón por la cual en el campus virtual se propondrá material educativo, apuntes de clase, bibliografía, así como también el programa y cronograma de la asignatura.

#### **6. Actividades de investigación y extensión (si hubiera)**

No aplica.

#### **7. Evaluación y régimen de aprobación**

Se organiza a partir de 4 entregas parciales del proyecto a realizar y una presentación final del mismo que permitan determinar el grado de aprendizaje del/la alumno/a.

De ser necesaria la recuperación de una entrega parcial, ello se llevará a cabo en forma conjunta con la entrega siguiente. La recuperación de la presentación final será en la oportunidad de la evaluación integradora, inmediatamente previo a la misma.

##### **7.1 Aprobación de la cursada**

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

El/La alumno/a deberá poseer una asistencia no inferior al 75% en las clases presenciales y encuentros sincrónicos.

La materia se cursa mayormente en forma de encuentros sincrónicos, a lo que se suman dos encuentros presenciales.

## **7.2 Aprobación de la materia**

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

**Promoción directa:** tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

**Evaluación integradora:** tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

**Examen final:** Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

## **7.3 Criterios de calificación**

Se establecen dos instancias de calificación, una correspondiente a la tercera entrega parcial, la otra respecto de la presentación final, en la escala de 0 a 10.

Para la calificación se tendrán en cuenta: la comprensión de los requerimientos obtenidos mediante las conversaciones con los stakeholders del proyecto que se lleva a cabo, la documentación de los mismos primariamente en forma de prototipos gráficos, la autoevaluación del avance logrado en cada entrega parcial en relación con los objetivos que el propio grupo establece al inicio de cada período, la comprensión de los conceptos principales de la arquitectura planteada tal como se refleja en el código producido, la gestión del proyecto llevada adelante por el grupo, la calidad del código, la contribucion concreta de cada integrante, y la presentación de una pequeña exposición a llevarse a cabo en cada instancia.

## 8. Cronograma

Semana		Teoría	Práctica	Modalidad
1	Presentación materia Objetivos Armado de equipos Selección de trabajos Uso de FIGMA		X	Virtual sincrónica
2	Selección de trabajos 1er Sprint: Planning	X	X	Virtual sincrónica
3	Metodología Agile			Virtual sincrónica
4	Relevamiento Análisis inicial Seguimiento	X		Virtual sincrónica
5	1er Sprint: Review 2do Sprint Planning		X	Virtual sincrónica
6	Uso de REACT parte 1	X		Virtual sincrónica
7	Seguimiento		X	Virtual sincrónica
8	2do Sprint: Review 3er Sprint Planning		X	Virtual sincrónica
9	Manejo de BRANCHES	X		Virtual sincrónica
10	Uso de REACT parte 2	X		Virtual sincrónica
11	3er Sprint: Review 4to Sprint Planning <b>PRESENTACION MEDIO TÉRMINO</b>		X	Presencial
12	Uso de BACKEND parte 1	X		Virtual sincrónica
13	Seguimiento		X	Virtual sincrónica
14	4to Sprint: Review 5to Sprint Planning		X	Virtual sincrónica
15	Uso de BACKEND parte 2	X		Virtual sincrónica
16	<b>PRESENTACION FINAL</b>		X	Presencial