

PROGRAMA de *Taller de Trabajo de Inserción Profesional*

Carrera/s: Tecnicatura Universitaria en Programación Informática

Asignatura: Taller de *Trabajo de Inserción Profesional*

Núcleo al que pertenece: *Complementario (TPI)*

Profesor: *Fernando Dodino, Susana Rosito y Micaela Alonso*

Asignaturas previas necesarias para favorecer el aprendizaje:

- Desarrollo de aplicaciones
- Elementos de Ingeniería de Software

Otros prerequisites: Tener aprobados el 100% del ciclo introductorio, el 100% del núcleo básico obligatorio, el 100% del núcleo avanzado obligatorio y haber obtenido al menos 8 créditos del núcleo complementario.

Objetivos:

Que el alumnado desarrolle un pequeño proyecto de software tal que:

- Integre los conocimientos adquiridos en las diferentes materias de la carrera.
- Promover el uso de la creatividad e iniciativa para afrontar con eficiencia el trabajo proyectado.
- Asegurar el empleo de una correcta metodología y criterio profesional para el logro del plan propuesto.
- De solución a un problema o necesidad real.
- La metodología, técnicas y tecnologías utilizadas sean de uso industrial y ya hayan sido aplicadas en materias anteriores.
- Gestione el alcance y el cronograma de demostraciones de las versiones intermedias del producto.
- El producto posea interfaz gráfica y persistencia de información.
- El producto cumpla con los requerimientos funcionales satisfaciendo atributos de calidad tales como mantenibilidad, flexibilidad y robustez.

Contenidos mínimos:

Metodologías ágiles: *backlog*, iteraciones, demostraciones. *Software Configuration Management (SCM)*. Integración continua. *Testing* automático. Arquitecturas de software. Persistencia de información. Interfaz gráfica de usuario.

Carga horaria semanal: 5 hs

Programa analítico:

Unidad 1: Plan de trabajo

- Metodología de trabajo: *Scrum*, generación de *backlog* inicial. Plan de entregas incrementales periódicas.
- Arquitectura de software inicial de un proyecto: Selección de tecnologías, identificación de los distintos *concerns* (vista, modelo, persistencia, interfaces externas, etc) de la aplicación y la interacción entre los mismos.
- Gestión de Riesgos
- Entorno de desarrollo: *IDE*, Integración continua, *SCM*, *Issue Tracker*, Herramientas de documentación.

Unidad 2: Desarrollo

- Solución de requerimientos funcionales.
- Diseño y Arquitecturas de software.
- *Testing* automático.
- Gestión del *backlog* a lo largo de un proyecto: cambio de alcance, detección de nuevos requerimientos y cambios de prioridades.

Unidad 3: Entrega final de un producto de software

- Documentación técnica.

- Instalación en un ambiente de pre-producción.

Bibliografía obligatoria:

- *Growing object-oriented software guided by tests*, Addison-Wesley Professional; 1 edition
- *Clean code: A Handbook of Agile Software Craftsmanship*, PRENTICE HALL; Edición: 1
- *The Art of Agile Development*, Shore & Warden

Patterns of Enterprise Application Architecture, Addison-Wesley Professional; 1 edition

Bibliografía de consulta:

- *"The Pragmatic Programmer: From Journeyman to Master"*, Andrew Hunt, David Thomas, 1999, Addison-Wesley, ISBN-13: 978-0201616224.
- *"BDD in Action: Behavior-Driven Development for the whole software life-cycle"*, John Ferguson Smart, 2014, Manning Publications, ISBN: 9781617291654.
- *"Extreme Programming Explained: Embrace Change"*, Kent Beck, 2004, Addison-Wesley, ISBN-13: 978-0321278654.
- *"Agile Testing: A Practical Guide for Testers and Agile Teams"*, Lisa Crispin, 2009, Addison-Wesley, ISBN-13: 978-0321534460.
- *"Release It!: Design and Deploy Production-Ready Software"*, Michael T. Nygard, 2007, Pragmatic Bookshelf, ISBN-13: 978-0978739218
- *"Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation"*, Jez Humble, David Farley, 2010, Addison-Wesley, ISBN-13: 978-0321601919
- *Continuous Integration: Improving Software Quality and Reducing Risk*, Addison-Wesley Professional; 1 edition

Object Thinking, Microsoft Press

Organización de las clases:

Esta materia es un taller cuyo objetivo es el desarrollo de un software de tamaño pequeño, siguiendo una metodología ágil de trabajo. Quienes cursen la materia pueden trabajar de manera solitaria o en grupos de 2.

Las primeras 3 semanas se dedican al desarrollo de un plan de trabajo, el cual incluye:

- Tema del TIP.
- Objetivos del trabajo.
- Plan de trabajo propuesto y su cronograma, que debe seguir un esquema iterativo con entregas parciales, presenciales e incrementales. La última entrega se considera la entrega final.
- Arquitectura propuesta, lo cual incluye las tecnologías a utilizar y la identificación de los distintos *concerns* del sistema y su relaciones.
- Currículum vitae del o de los estudiantes.

A partir de la aprobación del plan de trabajo, el/la estudiante se ajustará al mismo, debiendo cumplir todas las entregas parciales allí definidas. Cada entrega debe incluir un incremento del producto de software funcionando, y la documentación técnica respaldatoria. Los cambios en el plan de trabajo son permitidos, pero deben ser consensuados con el equipo docente.

Modalidad de evaluación:

Todas las entregas parciales y la entrega final serán presenciales y evaluadas por el equipo docente. Antes de la presentación presencial, debe enviarse la entrega en forma electrónica al profesor/ra del curso. El estudiantado deberá contestar a las preguntas que se le requiera. Cada entrega parcial puede resultar, a criterio del profesor/ra del curso, aprobada o rechazada. El/La docente puede decidir que el rechazo de una entrega parcial no implique la finalización del trabajo. La entrega final debe ser aprobada.

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18)

CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*			Evaluación
		Teórico	Práctico		
			Res Prob.	Lab.	
1	Definición de equipos y tema / Unidad 1			X	
2	Definición de backlog inicial y armado del ambiente de desarrollo/ Unidad 1			X	
3	Programación de un caso de uso testigo que pruebe la arquitectura propuesta / Unidad 1			X	
4	Presentación de plan de trabajo/ Unidad 1			X	
5	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	
6	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	X
7	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	
8	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	X
9	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	
10	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	X
11	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	
12	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X	X
13	Desarrollo de funcionalidades y su documentación técnica / Uni-			X	

	dad 2					
14	Desarrollo de funcionalidades y su documentación técnica / Unidad 2			X		X
15	Informe y documentación final. Instalación en ambiente de pre-producción / Unidad 3			X		X
16	Integrador					X

*INDIQUE CON UNA CRUZ LA MODALIDAD