

INSTITUTO DE DESARROLLO ECONÓMICO E INNOVACIÓN

Año: 2021



Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur.

PROGRAMA DE LA ASIGNATURA:

Laboratorio de Programación y Lenguajes
(IF009)

CÓDIGO: IF009

AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:

3 año

FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:

2019-04-07

CARRERA/S: Licenciatura en Sistemas 049/2017,
Analista Universitario de Sistemas 050/2017,

CARÁCTER: CUATRIMESTRAL (1ro)

TIPO: OBLIGATORIA

NIVEL: GRADO

MODALIDAD DEL DICTADO: PRESENCIAL (EN
LÍNEA)

MODALIDAD PROMOCION DIRECTA: SI

CARGA HORARIA SEMANAL: 6 HS

CARGA HORARIA TOTAL: 90 HS

EQUIPO DOCENTE

Nombre y Apellido	Cargo	e-mail
Matias Gel	Profesor Adjunto	mgel@untdf.edu.ar
Matias Moncho	Asistente de Primera	mmoncho@untdf.edu.ar

1. FUNDAMENTACION

EN EL MARCO DE LA PANDEMIA COVID-19 LA CURSADA 2021 SE DICTARA EN MODALIDAD "EN LINEA"

Esta asignatura se enfoca en el desarrollo de aplicaciones concretas por parte de los alumnos, lo cual constituye un ejercicio similar a su futura actividad profesional, requiriendo por lo tanto de creatividad, para el planteo de la solución de problemas e integración de conocimientos para la implementación de dicha solución.

La materia se ubica en el primer cuatrimestre del tercer año, razón por la cual los alumnos ya han incorporado conocimientos relativos a diseño de algoritmos y estructuras de datos, programación OO, arquitectura y diseño de software, bases de datos y otros, sin haber tenido aún la oportunidad de integrar todos estos conceptos en el desarrollo de aplicaciones concretas. Si bien la correlatividad anterior está asignada con Algorítmica y Programación II, los contenidos mínimos expresan que se debe trabajar sobre la base de los conocimientos adquiridos los dos primeros años, razón por la cual se parte de considerar los mismos. No obstante ello, se brindan a nivel teórico los conceptos básicos introductorios a cada tema que, siendo necesarios para desarrollar aplicaciones, no estén contemplados en la correlatividad anterior.

La falta de integración del tratamiento de los conceptos mencionados, brindados a los alumnos durante los dos primeros años, provoca en el tercer año la necesidad de trabajar sobre ciertos conocimientos específicos para el desarrollo de aplicaciones concretas que se pretende encarar en la presente asignatura, como son, modelos de persistencia y su vínculo con los lenguajes de programación, el mapeo OO-relacional, testing unitario, el manejo de herramientas integradas de desarrollo y frameworks como ambientes de trabajo y ejecución para la construcción ágil de aplicaciones.

De acuerdo a lo expuesto, se hace necesaria la enseñanza de conceptos fundamentales para el desarrollo de aplicaciones concretas en diferentes lenguajes con soporte de ambientes integrados

de programación que faciliten el desarrollo rápido de las mismas, los cuales incluyan una capa de persistencia y facilidades para el mapeo OO-relacional de manera ágil y sencilla, así como para la construcción de interfaces de usuario y testing de aplicaciones.

2. OBJETIVOS

a) OBJETIVOS GENERALES

Se pretende que el alumno sea capaz de desarrollar en forma ágil y eficiente aplicaciones concretas, utilizando y profundizando los conocimientos de diseño, programación y lenguajes adquiridos previamente, mediante la utilización de herramientas y ambientes de programación y ejecución apropiados para la integración de los diferentes aspectos que conforman una aplicación.

b) OBJETIVOS ESPECIFICOS

Se pretende que mediante los contenidos dictados y su aplicación en el desarrollo de aplicaciones concretas, el alumno:

- Profundice sus conocimientos sobre programación y lenguajes para desarrollar aplicaciones de software de calidad.
- . Aprenda a utilizar herramientas de testing unitario.
- Amplíe sus conocimientos sobre el paso del diseño a la implementación de programas.
- Aprenda a resolver el problema de mapeo OO-relacional en bases de datos.
- Aprenda a utilizar herramientas que le permitan agilizar el desarrollo de aplicaciones.
- Comprenda las ventajas de desarrollar aplicaciones en el marco de un framework de aplicación.
- Conozca lenguajes y técnicas apropiadas para el desarrollo ágil de aplicaciones.
- Esté en condiciones de analizar y realizar comparaciones entre frameworks y lenguajes de programación desde el punto de vista de la implementación de aplicaciones.

3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA

DEBIDO A LA PANDEMIA COVID-19 LA CURSADA 2021 SE EVALUARÁ "EN LINEA" DE LA SIGUIENTE FORMA:

Para la aprobación del cursado de la asignatura se requiere:

- Aprobar la totalidad de los Trabajos Prácticos con Nota igual o superior a seis (6).
- Defensa del Desarrollo de dos aplicaciones integradoras (a la mitad del cuatrimestre y al final del cuatrimestre) de los distintos conceptos analizados en la materia, en la cual debe obtener Nota igual o superior a seis 6.

La calificación final de la cursada es el promedio de las notas obtenidas en los trabajos prácticos y las aplicaciones integradoras. En ningún caso la nota obtenida podrá ser inferior a 6.

Para la promoción de la materia deberá tener un promedio de las entregas mayor a 7 y se le una defensa oral "en línea" del trabajo final

Para aprobar el final el alumno debe presentar y defender una aplicación desarrollada por el que cubra los contenidos de la materia con una nota no inferior a 6.

4. CONTENIDOS DE LA ASIGNATURA

CONTENIDOS MÍNIMOS

Desarrollo de aplicaciones concretas en las que se integren conocimientos adquiridos en los dos primeros años.

METODOLOGÍAS DE CLASES

Se dictarán clases teóricas relativas a los conceptos expuestos en el Programa analítico y en particular de su aplicación en el Lenguaje seleccionado para el desarrollo de la aplicación integradora, con un fuerte apoyo de ejemplos en ese y otros lenguajes conocidos por los alumnos, a los fines de que estén en condiciones de realizar comparaciones. Se realizarán presentaciones PowerPoint, así como la ejecución de programas ejemplos en los lenguajes seleccionados utilizando proyector y pantalla. Las herramientas IDE y los frameworks serán asimismo explicados mediante ejemplos desplegados en pantalla, desarrollados frente a los alumnos.

Durante las clases prácticas, se trabajará en base a una intensa práctica del funcionamiento y utilización de

herramientas que permita la agilización de aplicaciones, de los constructores provistos por los diferentes lenguajes a utilizar, mediante ejemplos y posterior ejercitación con el fin de comparar y evaluar las prestaciones de los mismos. Se realizará una intensa ejercitación y práctica de programación. Se utilizará un lenguaje específico como central para el desarrollo de aplicaciones en cada parte de la asignatura.

Se promoverá la lectura de artículos de interés y bibliografía específica, a los fines de generar hábitos de investigación y actualización de conocimientos, así como la búsqueda de información relativa a publicaciones sobre frameworks y herramientas en Internet, especialmente revistas científicas, proceedings de congresos y seminarios. Todas las instancias implican autocorrección y/o corrección por parte del docente. Además de la atención permanente y la revisión de los trabajos prácticos, se brindarán clases de consulta en forma previa a los exámenes parciales y ante requerimientos puntuales, se dictarán clases adicionales para aclarar dudas.

El uso de bibliografía por parte del alumno: Los apuntes de cátedra se reducirán a tópicos generales de cada tema, incluyendo los conceptos básicos de los mismos en forma de hilo conductor. Se estima conveniente inducir a los estudiantes a trabajar con bibliografía que complemente los apuntes de clase, señalando en cada caso la más apropiada al tema en cuestión.

Se entiende que entre la teoría y la práctica debe haber una relación integrada, dado el intensivo uso de ejemplos y ejercicios que implica la aplicación de conceptos teóricos en el desarrollo de aplicaciones. De esta forma, las clases teóricas están complementadas con ejemplos y problemas prácticos apropiados para la mejor comprensión de la implementación de soluciones en diferentes lenguajes. Algunas clases prácticas serán precedidas por una introducción teórica para abordar el tema, ejemplos y ejercitación para el trabajo individual o grupal (según el caso); en otros casos, durante las clases prácticas los alumnos avanzarán en el desarrollo de aplicaciones, realizando pruebas, consultas y correcciones. Además de los ejercicios correspondientes a cada TP, se propondrán trabajos integradores en ambas partes de la asignatura (aplicaciones de escritorio y aplicaciones WEB) para que los alumnos realicen la experiencia de desarrollar aplicaciones concretas reales.

PROGRAMA ANALITICO

MÓDULO I: Diseño y construcción de aplicaciones en entornos de desarrollo ágil

El Modelo Ágil para construcción de aplicaciones. Herramientas de desarrollo ágil. Entornos de desarrollo integrado (IDE). Código auto documentado y herramientas de extracción de código.

Construcción de aplicaciones en entornos de desarrollo integrado. Pautas y criterios para la transformación diseño-código. Lenguajes dinámicos para desarrollo de aplicaciones. Introducción al lenguaje Python

MÓDULO II: Pruebas unitarias

Importancia de la detección oportuna de errores. Clasificación Tipos de Pruebas. Prueba de Unidad. Procedimiento para pruebas de unidad. Soporte del lenguaje Python para pruebas de

unidad.

MÓDULO III: Lenguajes y Frameworks para desarrollo de aplicaciones WEB

Características de aplicaciones WEB. Capas de una aplicación WEB. Arquitectura de una aplicación WEB. Diseño de la aplicación. El patrón MVC. Concepto de Framework como ambiente de trabajo. La importancia del uso de frameworks para desarrollo rápido de aplicaciones. Frameworks para entornos de programación WEB. Frameworks MVC para el desarrollo WEB. Introducción al Framework Django.

MÓDULO IV: Manejo de persistencia

Concepto de persistencia. Persistencia y su relación con los lenguajes de programación. Soluciones al problema de la persistencia en Python. Mapeo OO – Relacional. Comparación de tecnologías ORM (Object Relational Mapping). Persistencia en Django. Mapeo de entidades y relaciones en Django. Operaciones CRUD con Django. Consultas dinámicas en Django. .

MÓDULO V: Desarrollo de interfaces de usuario utilizando el patrón MVC.

Las vistas y templates de Django como parte del patrón MVC. Vistas genéricas de Django El lenguaje de templates de Django. Modelado de interfaz del usuario con Django y html5. Formularios de Django. Vistas y validaciones de formularios.

MÓDULO VI: Autorización y Autenticación.

Sesiones de usuario en aplicaciones web. Manejo de autenticación en Django. Vistas genéricas para manejo de usuario y formularios. Desarrollo de templates para manejo de usuarios. Autorización en Django, permisos en modelos y vistas. Generación de interface de administrador con manejo de permisos.

Modulo VII: Testing funcional en aplicaciones web.

Introducción al framework selenium. Simulación de eventos de usuario. Integración con Django.

5. RECURSOS NECESARIOS

- Proyector
- Pc
- Acceso A Internet Por La Red WIFI De La Universidad

6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
1	I	Entornos de desarrollo ágil. Utilización de IDEs. Introducción al lenguaje python. Practica: Ejercicios en python	
2	I	Aspectos avanzados del lenguaje Python. Practica: Ejercicios en python	
3	II	Pruebas unitarias. Mecanismos del lenguaje Python para pruebas unitarias. TP 2: Ejemplos y ejercicios con pruebas unitarias. El alumno escribe el comportamiento de una serie de clases y sus pruebas unitarias.	
4	II	Introducción al desarrollo web, presentación del framework Django. Introducción a la persistencia en Django. Practica: Mapeo de un modelo de dominio.	

5	III,iv	Persistencia en Django. Ejecución de consultas complejas. Practica: Consultas sobre un modelo de dominio	
6	Iv	Persistencia en Django. Aspectos avanzados de persistencia. Practica: Consultas sobre un modelo de dominio. Modelado de dominio de la aplicación integradora.	
7	V	Introducción a las vistas en django. Practicas: Vistas básicas en django.	
8	V	Introducción a los templates de django, Practica: Diagramación de	
9	-	Vistas Avanzadas. Vistas genericas de django. Vistas del modelo. Practica: aplicación integradora primeras vistas y templates.	
10	VI	Revisión y entrega de la aplicación integradora	
11	VI	Formularios de django. Practica: formularios.	
12	VI	Formularios de django, validaciones. Practica: formularios aplicacion integradora	
13	VI	Seguridad de las aplicaciones en django. autenticación y autorización	
14	VI	Aplicación de administración en django. Practica: Aplicación admin de la aplicación integradora.	
15	-	Pruebas de integración en django. Practica: Aplicación integradora.	
16	-	Presentación de aplicación integradora.	
17	-	Reunión de cátedra para elaboración de Informe final.	

7. BIBLIOGRAFIA DE LA ASIGNATURA

Autor	Año	Título	Capítulo/s	Lugar de la Edición	Editor / Sitio Web
David Beazley	2009	Python Essential Reference	1-2		Addison-Wesley
Aidas Bendoraitis	2016	Web Development with Django Cookbook	1-6		Packt Publishing
Daniel M. Brandon	2008	Software Engineering For Modern Web Applications: Methodologies And Technologies	i, iv, V,VIII,XIII		Hershey - Edición Electrónica
Measey, Peter; Radtac	2015	Agile Foundations - Principles, Practices And Frameworks	I-II		BCS The Chartered Institute For IT - Edición Electrónica
Harry J.W. Percival	2017	Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium, and JavaScript	1-7		O'Reilly Media

Firma del docente-investigador responsable

VISADO		
COORDINADOR DE LA CARRERA	DIRECTOR DEL INSTITUTO	SECRETARIO ACADEMICO UNTDF
Fecha :	Fecha :	

Este programa de estudio tiene una validez de hasta tres años o hasta que otro programa lo reemplace en ese periodo