

**Seminario de Lenguajes
(Opción Android + Kotlin)**

Año 2026

Carrera/ Plan:

Licenciatura en Informática Plan 2021/Plan 2015

Licenciatura en Sistemas Plan 2021/Plan 2015

Analista Programador Universitario Plan 2021/Plan 2015

*Analista en Tecnologías de la Información y la Comunicación
Plan 2021/Plan 2017*

Año: 2º

Régimen de Cursada: Semestral

Carácter: Obligatoria

Correlativas: Taller de Programación

Profesor/es: **Esp. Fernández Sosa Juan**

Francisco Hs. semanales: 2 hs. de Teoría y 4 hs.
de Práctica.

FUNDAMENTACIÓN

Esta asignatura representa para el alumno una primera experiencia concreta con un lenguaje de programación. En el caso particular de esta opción, se trata de un taller para el desarrollo de aplicaciones móviles para dispositivos Android utilizando el lenguaje de programación Kotlin. El mismo se dicta en la Sala de PC, clases teóricas, y también en el aula 3 las clases prácticas. En ambos horarios se cuentan con los carritos móviles que posee la facultad, con computadores tipo notebooks para los estudiantes. Durante el transcurso del taller, el alumno resolverá problemas más complejos que los vistos en Taller de Programación, modularmente. Al finalizar el curso el alumno habrá estudiado el ecosistema de Android y los lineamientos básicos para el desarrollo de aplicaciones para este sistema operativo móvil.

OBJETIVOS GENERALES

Profundizar los conocimientos obtenidos por el alumno en los primeros cursos vinculados con Algoritmos y Programación, permitiéndole desarrollar un estudio teórico-práctico de algún lenguaje de programación (el lenguaje puede variar con los cambios tecnológicos), poniendo énfasis en el análisis formal de las características del lenguaje y su comparación con los que el alumno conociera a ese momento (típicamente Pascal).

RESULTADOS DE APRENDIZAJE

- 1.3. Describir los avances informáticos actuales e históricos y demostrar cierta visión sobre tendencias y avances futuros (Básico).
- 3.1. Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos (Básico).
- 3.3. Elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales así como aplicaciones emergentes (Básico).
- 3.4. Describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador (Básico).

3.5. Aplicar las correspondientes competencias prácticas y de programación en la creación de programas informáticos y/u otros dispositivos informáticos (Adecuado).

6.1. Organizar su propio trabajo de manera independiente demostrando iniciativa y ejerciendo responsabilidad personal (Básico).

6.3. Planificar su propio proceso de aprendizaje autodidacta y mejorar su rendimiento personal como base de una formación y un desarrollo personal continuos (Básico).

COMPETENCIAS

- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGS4- Aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje.
- CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS- CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

Estudio de un lenguaje de programación en el que se desarrollen aplicaciones concretas. En lo posible la oferta de lenguajes será variable y actualizada con el cambio tecnológico.

PROGRAMA ANALÍTICO

I- ASPECTOS ESENCIALES DE LA INGENIERÍA DE SOFTWARE PARA DISPOSITIVOS MÓVILES

Enfoques de desarrollo de aplicaciones móviles. Aplicaciones web móviles. Aplicaciones nativas. Aplicaciones Híbridas. Ventajas y desventajas. Requerimientos no funcionales.

II- INTRODUCCIÓN A ANDROID

¿Qué es ANDROID? Historia. Versiones. Características. Arquitectura. Dispositivos con Android. Fragmentación. Tienda de aplicaciones en Android (Google Play Store).

Entorno de desarrollo. Emuladores y SDK. Organización de un programa Android.

III- ESTRUCTURA DE APLICACIÓN ANDROID

Manifiesto de aplicación (Manifest). Permisos. Componentes de aplicación. Actividades. Servicios. Content Providers. Broadcast Receivers. Ciclo de vida una aplicación. Intents y Filters

IV- INTERFAZ DE USUARIO

Vistas, componentes y disposición. Tipos de componentes básicos. Layouts. Menú de opciones, menú contextual. Submenús. Eventos.

V- MODERNIZACIÓN DE LA UI: JETPACK COMPOSE

Introducción a la programación declarativa. Funciones Composable. Gestión de estados. Recomposición. Layouts modernos: Column, Row y Box. Manejo de listas con LazyColumn. Interoperabilidad entre Compose y el sistema de Vistas tradicional.

V. RECURSOS

Estilos y temas. Íconos e Imágenes. Internacionalización. Localización.

VI- NOTIFICACIONES

Toast. Dialogs, Status bar.

VII- SENSORES

Ubicación. WiFi. Cámara. Acelerómetro. Giroscopio.

VIII- ALMACENAMIENTO DE LA INFORMACIÓN

Shared Preferences. Internal Storage. External Storage. Base de datos SQLite. Almacenamiento en red

BIBLIOGRAFÍA

1. Android 6 for Programmers: An App-Driven Approach. Paul Deitel, Harvey Deitel, Alexander Wald. Prentice Hall; 3 edition. 2015.
2. Programming Android. Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura. O'Reilly Media; 3 edition. 2013.
3. Hello, Android: Introducing Google's Mobile Development Platform. Ed Burnette. Series: Pragmatic Programmers - Pragmatic Bookshelf; Third Edition edition. 2010.
4. Android Developers. <https://developer.android.com/training/index.html?hl=es>
5. Develop Android apps with Kotlin. <https://developer.android.com/kotlin>
6. Get started with Jetpack Compose. <https://developer.android.com/develop/ui/compose/documentation>

METODOLOGÍA DE ENSEÑANZA

La actividad curricular se organiza en Teoría, Práctica y Explicación de Práctica.

La teoría consiste en una presentación y explicación conceptual por parte del docente de los temas centrales donde los alumnos participan haciendo preguntas y planteando dudas.

Las explicaciones de la práctica permiten la articulación entre la teoría y la práctica. El docente desarrolla algún desafío modelo que resulta importante para la resolución de las ejercitaciones prácticas.

En la práctica se resuelven ejercicios en forma individual que son planteados en las guías de TP.

Se presentarán desafíos que serán resueltos por los alumnos, en los cuáles se evaluarán sus capacidades técnicas, los procesos de investigación empleados y sus aptitudes para defender los desarrollos de manera oral y escrita.

Los docentes formularán interrogantes acerca del estado actual de las plataformas móviles y su posible evolución.

Se detalla a continuación la validación de las competencias enunciadas previamente:

CGS2- Comunicarse con efectividad en forma oral y escrita.

Los alumnos deben presentar un trabajo escrito (individual/en comisión) que es evaluado teniendo en cuenta el contenido técnico, pero también la estructura, organización, sintaxis, claridad conceptual y la bibliografía consultada que debe ser citada rigurosamente. Posteriormente hay un coloquio (grupal/individual) donde los docentes interrogan sobre contenidos específicos del documento. Esto se refleja en planillas escritas que conforman documentación de evaluación del trabajo.

CGS4- Aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje.

En la cátedra se organizan actividades planificadas para los alumnos, en los que se proponen “desafíos” que deben convertirse en “ideas proyecto” y posteriormente en potenciales desarrollos del alumno. Se trata de que alumno logre abstraer una serie de pasos que respondan a una metodología clásica de investigación y lo ayuden a formarse en esta competencia:

- Búsqueda de bibliografía actualizada sobre el tema.
- Abstracción del desafío/problema como una “idea proyecto a resolver”.
- Expresión sintética de la especificación del proyecto, con recursos humanos requeridos y plan de tareas.
- Implementación y defensa oral/escrita de la solución al desafío.

La cátedra acompaña el proceso del alumno, para consolidar sus habilidades para esta competencia.

La evaluación de esta competencia se refleja en una planilla detallada, donde se indica la capacidad del alumno para desarrollar su aprendizaje y la formulación de la solución al desafío en forma autónoma.

CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras.

En la cátedra se plantean actividades planificadas para los alumnos, en los que se propone el estudio de la tecnología existente y prevista para un tipo de problema y se los “desafía” a presentar la posible evolución de la solución para ese tipo de problema y en qué podría mejorarse la solución/soluciones actuales.

Esto lleva al alumno a buscar bibliografía relacionada con el cambio tecnológico y formarse un criterio sobre las tendencias (por ejemplo, en los procesadores a utilizar o la migración de aplicaciones a móviles).

La cátedra acompaña el proceso del alumno, para contrastar las conclusiones del alumno y validar su habilidad para esta competencia. La evaluación de esta competencia se refleja en una planilla detallada, donde se indican los resultados del estudio bibliográfico del alumno y su capacidad para formular las ventajas potenciales del cambio tecnológico en el problema/área de conocimiento planteada.

CGT1- Identificar, formular y resolver problemas de Informática.

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real, especificación de los mismos como problemas resolubles desde la informática y en el desarrollo de soluciones verificables para los mismos.

La evaluación de esta competencia forma parte de las evaluaciones de trabajos prácticos y contenidos teóricos, reflejándose en la corrección de las pruebas escritas del alumno.

CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.

En la cátedra se pone énfasis en la capacidad del alumno para conocer técnicas y herramientas de aplicación en Informática (en lo posible siguiendo las tendencias marcadas por el cambio tecnológico) y en la aplicación efectiva de las mismas.

La cátedra acompaña el proceso con materiales para que el alumno estudie casos y valore la selección y empleo eficiente de herramientas y técnicas determinadas para cada problema.

La evaluación de esta competencia forma parte de las evaluaciones de trabajos prácticos y contenidos teóricos, reflejándose en la corrección de las pruebas escritas del alumno.

EVALUACIÓN

Para aprobar la cursada el alumno deberá:

- Presentar y exponer tres desafíos de programación obligatorios. En caso de desaprobado cualquiera de ellos, el alumno contará con una fecha más de re-entrega. Los desafíos se realizan en grupos conformados por 2 integrantes.
- Rendir un examen integrador hacia la finalización del curso y obtener una calificación mayor o igual a 6 (seis). En caso de desaprobado se tomarán hasta 2 recuperatorios.

Régimen de promoción:

- El alumno que apruebe la cursada también obtendrá la promoción. La calificación final se definirá en función del resultado de los trabajos prácticos y del examen.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	9/3	Introducción a Android. Android Studio. Primeros Pasos
2	16/3	Introducción a las Views. Presentación de la plataforma y del entorno de desarrollo. Estructura de una aplicación Android. Permisos
3	30/3	Intents
4	6/4	Ciclo de vida de aplicaciones y procesos
5	13/4	Layouts - Linear, Frame, ScrollView, Horizontal scroll, clase R, Relative 1/2
6	20/4	Layouts - Relative 2/2, TableLayout, GridLayout, ConstraintLayout. Dimensiones
7	27/4	Modernización UI. Intro a Jetpack Compose: Funciones Composable y el cambio de paradigma de UI.
8	4/5	UI Avanzada en Compose: Gestión de estado, Recomposición y Listas dinámicas.
9	11/5	Recursos. Internacionalización. Localización
10	18/5	Recursos. Menú de opciones, menú contextual. Submenús.
11	1/6	Dialogos y notificaciones
12	8/6	Sensores. Componentes de Aplicación
13	22/6	Consulta.
14	29/6	Consulta.
15	6/7	Exámen integrador
16	13/7	1er Recuperatorio Exámen Integrador
17	3/8	2do Recuperatorio Exámen Integrador

Evaluaciones previstas	Fecha
6/7/2026	Exámen Integrador
13/7/2026	1er Recuperatorio Exámen Integrador
3/8/2026	2do Recuperatorio Exámen Integrador

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

- jfernandez@lidi.info.unlp.edu.ar
- Plataforma IDEAS, curso “Seminario de Lenguajes Opción Android + Kotlin”
- Cartelera virtual de la Facultad

Fernández Sosa, Juan Francisco

Firma del/los profesor/es