Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Inteligencia Artificial
Sílabo 2027-I

1. CURSO

CS2B1. Desarrollo Basado en Plataformas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS2B1. Desarrollo Basado en Plataformas
2.2 Semestre	:	3er Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS112. Programación Orientada a Objetos I. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El desarrollo basado en plataformas se enfoca en el diseño y la implementación de aplicaciones dentro de entornos informáticos específicos. Este curso utiliza el marco de Proceso y Diseño de Software (SPD) para explorar las restricciones y capacidades únicas de las plataformas Web, Móviles e Interactivas. Los estudiantes aprenderán a aplicar preocupaciones arquitectónicas compartidas y APIs específicas de plataforma para construir soluciones de software robustas, escalables y centradas en el usuario.

5. OBJETIVOS

- Comprender los aspectos compartidos y las restricciones únicas de las plataformas modernas.
- Desarrollar aplicaciones profesionales para entornos Web y Móviles.
- Aplicar principios de computación interactiva para mejorar la experiencia del usuario.
- Diseñar arquitecturas de software que aprovechen los servicios específicos de la plataforma.

6. RESULTADOS DEL ESTUDIANTE

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usage)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usage)

7. TEMAS

Unidad 1: Aspectos Comunes: Plataformas, APIs y Restricciones (5 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general de las plataformas de desarrollo (es decir, web, móvil, juegos, robótica, embebida e interactiva). • Entrada/sensores/dispositivos de control/dispositivos hápticos • Limitaciones de recursos ◦ Computacional ◦ Almacenamiento de datos ◦ Memoria ◦ Comunicación • Requisitos - seguridad, disponibilidad de tiempo de actividad, tolerancia a fallos Conceptos de Confiabilidad, Ingeniería de Confiabilidad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad • Salida/actuadores/dispositivos hápticos ▪ Programación mediante Interfaces de Programación de Aplicaciones (API) específicas de la plataforma vs construcción de aplicaciones tradicional ▪ Programación bajo restricciones y requisitos de la plataforma (por ejemplo, herramientas de desarrollo disponibles, desarrollo, consideraciones de seguridad) Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad	▪ Enumerar las restricciones de la programación móvil [Listar/Enumerar]
Lecturas : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unidad 2: Aspectos Comunes: Lenguajes de Plataforma y Patrones Web (3 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general de los Lenguajes de plataforma (por ejemplo, Python, Swift, Lua, Kotlin) ▪ Técnicas para aprender y dominar un lenguaje de programación específico de la plataforma	▪ Enumerar las características de los lenguajes de scripting [Listar/Enumerar] ▪ Describir el modelo de tres capas (three-tier) de la programación web [Describir] ▪ Describir cómo se mantiene el estado en la programación web [Describir]
Lecturas : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unidad 3: Frontend, Frameworks y Estándares Web (12 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Lenguajes de programación web (por ejemplo, HTML5, JavaScript, Typescript, CSS) ▪ Plataformas, frameworks o meta-frameworks web • Servicios en la nube • API, Componentes Web (Web Components) ▪ Software como Servicio (SaaS) ▪ Estándares web como el modelo de objetos del documento (DOM), accesibilidad Accesibilidad y Diseño Inclusivo ▪ Consideraciones de Seguridad y Privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad	▪ Comparar y contrastar la programación web con la programación de propósito general [Comparar] ▪ Describir las diferencias entre Software-como-Servicio (SaaS) y productos de software tradicionales [Describir] ▪ Discutir cómo los estándares web impactan el desarrollo de software [Debatir] ▪ Evaluar una aplicación web existente contra los estándares web actuales [Evaluar]
Lecturas : [**Flanagan2020**], [**Duckett2014**], [**Fielding:2000:Rest**]	

Unidad 4: Plataformas Móviles: Fundamentos del Desarrollo (6 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Desarrollo con • Lenguajes de programación móvil • Entornos de programación móvil ▪ Restricciones de la plataforma móvil • Diseño de interfaz de usuario Comprender al Usuario: Objetivos individuales e interacciones con otros • Seguridad ▪ Acceso • Acceso a datos a través de APIs Construcción de Consultas • Diseño de puntos finales de API (endpoints) para aplicaciones móviles: trampas y consideraciones de diseño • Interfaces de red y web Modelado de Datos, Fundamentos de Redes y Comunicaciones	▪ Comparar la programación móvil con la programación de propósito general [Comparar] ▪ Evaluar los pros y los contras del desarrollo de aplicaciones móviles nativas y multiplataforma [Evaluar]
Lecturas : [**Deitel2017**], [**Hellman2013**]	

Unidad 5: Plataformas Móviles: Arquitectura y Restricciones (3 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Desarrollo • Desarrollo nativo (native) versus multiplataforma (cross-platform) • Patrones de diseño/arquitectura de software para aplicaciones móviles Diseño de API y Modelado, Principios y Arquitectura de Software, Calidad y Evaluación del Diseño ▪ Restricciones de la plataforma móvil • Diseño de interfaz de usuario responsivo (responsive) Accesibilidad y Diseño Inclusivo • Heterogeneidad y movilidad de los dispositivos • Diferencias en las experiencias de usuario (por ejemplo, entre aplicaciones móviles y basadas en web) • Compensación (tradeoff) entre energía y rendimiento ▪ Especificación y pruebas Prácticas de Desarrollo de Software, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas	▪ Desarrollar una aplicación móvil consciente de la ubicación con integración de API de datos [Crear] ▪ Construir una aplicación móvil impulsada por sensores capaz de registrar datos en un servidor remoto [Crear] ▪ Crear una aplicación de comunicación que incorpore telefonía y mensajería instantánea [Crear]
Lecturas : [**Deitel2017**], [**Hellman2013**]	

Unidad 6: Plataformas Móviles: Capacidades y Computación Asíncrona (3 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Capacidades (*affordances*) de la computación móvil • Aplicaciones conscientes de la ubicación (*location-aware*) • Computación impulsada por sensores (por ejemplo, giroscopio, acelerómetro, datos de salud de un reloj) • Telefonía y mensajería instantánea • Realidad aumentada Inmersión ▪ Computación asíncrona Algoritmos • Diferencia con la programación síncrona tradicional • Manejo del éxito mediante devoluciones de llamada (*callbacks*) • Manejo de errores de forma asíncrona • Pruebas de código asíncrono y problemas típicos en las pruebas	▪ Desarrollar una aplicación móvil consciente de la ubicación con integración de API de datos [Crear] ▪ Construir una aplicación móvil impulsada por sensores capaz de registrar datos en un servidor remoto [Crear] ▪ Crear una aplicación de comunicación que incorpore telefonía y mensajería instantánea [Crear]
Lecturas : [**Deitel2017**], [**Hellman2013**]	

Unidad 7: SPD-SEP/Interactiva (10 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pautas éticas al usar modelos de IA para ayudar en el periodismo y la creación de contenido ▪ Rendición de cuentas (*accountability*) por las salidas generadas por IA ▪ Comportamiento entre programadores de prompts (*prompt programmers*) y desarrolladores de IA ▪ Confianza con el público al usar modelos de IA	▪ Indicar un marco para la rendición de cuentas (*accountability*) en el despliegue de modelos de IA, incluyendo documentación clara y atribución [Evaluar] ▪ Discutir códigos de conducta éticos y estándares profesionales relevantes para la programación de prompts (*prompt programming*) y el desarrollo de IA [Debatir] ▪ Crear planes y materiales de comunicación para educar al público sobre las capacidades, limitaciones y salvaguardas éticas de la IA [Crear]
Lecturas : [**Duckett2014**], [**Deitel2017**]	

Unidad 8: Plataformas Interactivas de Datos y Análisis (6 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Plataformas de Cuadernos Computacionales (por ejemplo, Cuadernos Jupyter; Google Colab; R; SPSS; Observable) ▪ Plataformas de Análisis de Datos y SQL en la Nube (por ejemplo, BigQuery) Construcción de Consultas • Apache Spark • Visualizaciones de Datos Visualización ▪ Presentaciones Interactivas Respaldadas por Datos ▪ Herramientas de Diseño con Bucles de Retroalimentación de Baja Latencia (*Low-Latency Feedback Loops*) • Herramientas de renderizado • Herramientas de diseño gráfico	▪ Analizar grandes conjuntos de datos de forma interactiva [Analizar] ▪ Crear una pista de acompañamiento para una interpretación musical, como la codificación en vivo (*live coding*) [Crear] ▪ Crear cuadernos computacionales convincentes que construyan una narrativa para un objetivo/historia periodística dada [Crear] ▪ Implementar código interactivo que use un conjunto de datos y genere gráficos exploratorios [Implementar] ▪ Contextualizar los atributos de diferentes estilos de análisis de datos, como interactivo versus tubería (*pipeline*) de ingeniería [Contextualizar] ▪ Escribir un programa usando una plataforma de computación de cuadernos (por ejemplo, búsqueda, ordenación o manipulación de grafos) [Escribir]
Lecturas : [**Flanagan2020**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

********* EVALUATION MISSING ********

10. BIBLIOGRAFÍA BÁSICA