|  |  |
| --- | --- |
| Programa Académico | **Ingeniería de Sistemas** |
| Unidad Académica | **Ingeniería de Sistemas** |
| Asignatura | **Seminario Actualización II** |
| Código | **IS0903** |
| Semestre | **9** |
| Créditos Académicos | **3** |
| Tipo Asignatura | **Teórica - Practica** |
| Ciclo de Formación | **Profundización** |
| Componente de formación | **Desarrollo de Software** |
| Modalidad | **Presencial** |
| Descripción de la asignatura | El seminario de actualización II se integra en el programa académico como un curso diseñado para adaptarse dinámicamente a los avances tecnológicos, especialmente en el desarrollo de software. Es un componente esencial del currículo, su particularidad radica en la flexibilidad con que el comité curricular autoriza la actualiza los temas tratados, asegurando así que los estudiantes estén al tanto de las últimas tendencias, herramientas, y metodologías relevantes para la creación de soluciones en tecnologías de la información. Este enfoque innovador refleja la naturaleza "electiva" del curso en términos de contenido, aunque su asistencia es obligatoria para todos los estudiantes del programa. |

|  |  |
| --- | --- |
| Control de Versiones | **1.0** |
| Preparo | **Gonzalo Andrés Lucio López** |
| Fecha | **11/11/2023** |
| Acta Comité Curricular |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resultados de Aprendizaje de Asignatura | | | | |
| Código | Dimensión | Dominio Cognitivo | RAP | Descripción |
| **RAA-1** | **Saber** | Aplicar | D21 | Usar herramientas y tecnologías de desarrollo de software en el diseño e implementación de soluciones a problemas hipotéticos de baja escala, considerando criterios de eficiencia, sostenibilidad y viabilidad. |
| **RAA-2** | **Saber Hacer** | Analizar | E18 | Modelar la adecuación de diferentes herramientas y tecnologías de desarrollo para solucionar problemas específicos, evaluando ventajas, limitaciones y posibles sinergias entre tecnologías. |
| **RAA-3** | **Saber Hacer** | Crear | E23 | Desarrollar prototipos funcionales utilizando herramientas y tecnologías emergentes que demuestren soluciones innovadoras a problemas hipotéticos, enfocándose en la aplicabilidad, escalabilidad y el impacto tecnológico. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RAA | Temas | Resultado de Aprendizaje Especifico | | Rúbrica | | | | | |
| Peso % | Criterios de evaluación | Niveles de desempeño | | | |
| Id. | Descripción | Superior | Alto | Básico | Bajo |
| RAA-1 | Creación y configuración de proyectos con Ionic y Capacitor, enfocándose en la interfaz de usuario y la conexión con servicios externos. | RAE-1 | Diseñar y configurar aplicaciones móviles utilizando Ionic y Capacitor, optimizando la interfaz de usuario y asegurando la conexión efectiva con servicios externos. |  | El estudiante deberá demostrar la capacidad para conceptualizar, diseñar y desarrollar una aplicación móvil que no solo sea funcional sino también intuitiva para el usuario, empleando Ionic y Capacitor. Esto incluye la creación de una interfaz de usuario atractiva y la integración eficiente con APIs o servicios externos. | El estudiante crea una aplicación móvil con interfaces de usuario excepcionales, demostrando innovación y una integración perfecta con servicios externos. La aplicación ofrece una experiencia de usuario sobresaliente, con transiciones fluidas y una estructura intuitiva que mejora la interacción general. | El estudiante desarrolla una aplicación móvil con una interfaz de usuario bien diseñada y una buena integración con servicios externos. La aplicación es funcional, fácil de navegar y estéticamente agradable, aunque con margen de mejora en aspectos de innovación y experiencia de usuario. | El estudiante logra configurar una aplicación móvil con una interfaz de usuario funcional y una integración básica con servicios externos. La aplicación cumple con los requisitos mínimos de funcionalidad y usabilidad, pero carece de elementos que enriquezcan la experiencia de usuario y la eficiencia en la integración de servicios. | El estudiante presenta una aplicación móvil con deficiencias significativas en la interfaz de usuario y la integración con servicios externos. La navegación es complicada, la interfaz no es intuitiva, y la conexión con servicios externos es inestable o mal implementada, afectando negativamente la funcionalidad y la experiencia de usuario. |
| Desarrollo de APIs RESTful con Laravel, incluyendo rutas, controladores y autenticación de usuarios. | RAE-2 | Desarrollar una API RESTful utilizando Laravel que incorpore rutas, controladores y sistemas de autenticación de usuarios. |  | Los estudiantes serán evaluados en su capacidad para diseñar y construir una API RESTful con Laravel que maneje eficazmente las solicitudes HTTP, implemente patrones de diseño MVC a través de controladores y modelos, y asegure el acceso a través de sistemas de autenticación. | El estudiante diseña y desarrolla una API RESTful completa con Laravel que no solo cumple con todos los requisitos funcionales, sino que también implementa prácticas avanzadas de seguridad, optimización y documentación exhaustiva. La API gestiona las solicitudes de manera eficiente, utilizando controladores bien estructurados, y el sistema de autenticación es robusto y seguro. | El estudiante construye una API RESTful con Laravel que satisface los requisitos básicos, incluyendo rutas adecuadas, controladores funcionales y un sistema de autenticación efectivo. La API es segura y maneja las operaciones solicitadas correctamente, aunque puede mejorar en optimización y documentación. | El estudiante logra desarrollar una API RESTful que cumple con los requisitos mínimos de funcionalidad, con rutas y controladores que permiten realizar operaciones CRUD básicas y un sistema de autenticación simple. La API funciona correctamente, pero muestra áreas de mejora en términos de seguridad, estructura y eficiencia. | El estudiante presenta una API RESTful incompleta o con errores significativos en la implementación de rutas, controladores o autenticación. La API muestra deficiencias en funcionalidad, seguridad y manejo de errores, indicando una comprensión insuficiente de los conceptos y prácticas de desarrollo con Laravel. |
| Modelado de datos y configuración en MongoDB para integrarse con Laravel. | RAE-3 | Implementar modelos de datos en MongoDB y configurar su integración con Laravel para el desarrollo de aplicaciones web. |  | El estudiante demuestra capacidad para diseñar e implementar esquemas de datos eficientes en MongoDB, adecuados a las necesidades de la aplicación. Además, configura correctamente la integración entre MongoDB y Laravel, asegurando una comunicación efectiva entre la base de datos y la aplicación web. Esto incluye la creación de modelos en Laravel que representen y gestionen los datos de MongoDB, la implementación de operaciones CRUD a través de la interfaz de Laravel, y la aplicación de buenas prácticas de seguridad y optimización del rendimiento. | El estudiante diseña esquemas de datos complejos en MongoDB que se integran perfectamente con Laravel, demostrando una comprensión avanzada de las capacidades de ambos sistemas. Implementa operaciones CRUD con eficiencia, seguridad y optimización del rendimiento, aplicando las mejores prácticas en todo momento. | El estudiante implementa esquemas de datos en MongoDB y logra una integración funcional con Laravel. Realiza operaciones CRUD correctamente, aunque con espacio para mejorar en aspectos de optimización del rendimiento y seguridad. | El estudiante diseña e implementa esquemas de datos en MongoDB y logra una integración básica con Laravel. Puede realizar operaciones CRUD, pero con errores ocasionales y aspectos de seguridad y rendimiento que necesitan mejora. | El estudiante enfrenta dificultades significativas en el diseño de esquemas de datos en MongoDB y su integración con Laravel. Las operaciones CRUD son implementadas de manera ineficiente, con fallos en la seguridad y el rendimiento. |
| RAA-2 | Análisis de la integración entre Ionic, Laravel y MongoDB para desarrollar aplicaciones móviles con funcionalidades completas. | RAE-1 | Analizar la Integración de Ionic, Laravel y MongoDB para el Desarrollo de Aplicaciones Móviles |  | El estudiante deberá demostrar una comprensión profunda de cómo las tecnologías Ionic, Laravel, y MongoDB se integran para el desarrollo de aplicaciones móviles. Esto incluye la capacidad para identificar y justificar cómo cada tecnología contribuye al funcionamiento global de la aplicación, la evaluación de ventajas y desafíos de su integración, y la propuesta de soluciones innovadoras para problemas comunes en el desarrollo de aplicaciones móviles. | El estudiante presenta un análisis exhaustivo y detallado de la integración de Ionic, Laravel y MongoDB, mostrando una comprensión avanzada de cómo estas tecnologías se complementan para desarrollar aplicaciones móviles. Propone soluciones innovadoras y viables a problemas complejos, respaldadas por un razonamiento crítico y ejemplos claros de aplicaciones potenciales. | El estudiante realiza un análisis competente de la integración de las tecnologías, identificando correctamente las contribuciones de cada una al desarrollo de aplicaciones móviles. Muestra una buena comprensión de las ventajas y limitaciones de su integración, con propuestas de solución para problemas típicos, aunque con menor grado de innovación que en el nivel superior. | El estudiante demuestra una comprensión general de la integración de Ionic, Laravel y MongoDB, pero el análisis carece de profundidad en aspectos clave. Las soluciones propuestas a problemas de integración son básicas y no abordan completamente los desafíos más complejos. | El estudiante muestra una comprensión limitada de la integración de las tecnologías y su análisis es superficial o inexacto. Las soluciones propuestas son inadecuadas o irrelevantes para los problemas identificados, indicando una falta de comprensión crítica de cómo estas tecnologías pueden trabajar conjuntamente para el desarrollo de aplicaciones móviles. |
| Implementación de mecanismos de seguridad y gestión de sesiones en Laravel para proteger el API. | RAE-2 | Implementar mecanismos de seguridad y gestión de sesiones en Laravel para proteger un API. |  | El estudiante demuestra competencia en la implementación de mecanismos de seguridad y gestión de sesiones en Laravel al desarrollar un API que es resistente a vulnerabilidades comunes. Esto incluye la correcta configuración de autenticación, el manejo seguro de sesiones, y la aplicación de prácticas recomendadas de seguridad en el desarrollo de software. | El estudiante implementa mecanismos de seguridad y gestión de sesiones en Laravel de manera excepcional, demostrando una comprensión avanzada de las prácticas de seguridad en el desarrollo de APIs. Las soluciones son innovadoras, superan las prácticas estándar de seguridad y muestran un entendimiento profundo de la prevención de vulnerabilidades específicas. | El estudiante implementa mecanismos de seguridad y gestión de sesiones en Laravel de manera competente, siguiendo las mejores prácticas de seguridad. La configuración de seguridad es robusta, protegiendo eficazmente el API contra vulnerabilidades comunes, aunque con menor grado de innovación que en el nivel Superior. | El estudiante logra implementar mecanismos de seguridad y gestión de sesiones en Laravel con un entendimiento funcional de las necesidades de seguridad. La API está protegida contra vulnerabilidades básicas, pero la implementación podría mejorarse siguiendo más de cerca las mejores prácticas y recomendaciones de seguridad. | El estudiante muestra un entendimiento limitado de la seguridad y la gestión de sesiones en Laravel. La implementación de los mecanismos de seguridad es insuficiente, dejando el API vulnerable a ataques comunes y no cumple con las prácticas recomendadas de seguridad. |
| Mejora de la interacción entre la aplicación móvil Ionic y el API Laravel mediante el uso de Capacitor. | RAE-3 | Optimizar la Comunicación entre la Aplicación Móvil Ionic y el API Laravel Usando Capacitor |  | El estudiante demuestra competencia en la optimización de la comunicación entre una aplicación móvil desarrollada con Ionic y un API desarrollado con Laravel, utilizando Capacitor. Esto incluye la capacidad para implementar funcionalidades avanzadas que mejoren la interacción usuario-aplicación, asegurar una integración eficaz con el backend y facilitar un intercambio de datos fluido y eficiente | El estudiante demuestra una habilidad excepcional para optimizar la comunicación entre la aplicación Ionic y el API Laravel, aplicando técnicas avanzadas de Capacitor. Muestra innovación en la solución de problemas de integración, resultando en una experiencia de usuario altamente eficiente y una arquitectura de aplicación robusta. | El estudiante efectivamente mejora la interacción entre la aplicación Ionic y el API Laravel mediante el uso de Capacitor, implementando funcionalidades que reflejan una comprensión avanzada de las mejores prácticas de integración. La solución presentada es fiable, mejora significativamente la experiencia de usuario y la eficiencia de la comunicación entre el frontend y el backend. | El estudiante logra una mejora en la comunicación entre la aplicación Ionic y el API Laravel con el uso de Capacitor, aunque la implementación es estándar y podría mejorar en términos de innovación y eficiencia. La solución es funcional, pero hay oportunidades para optimizar la experiencia de usuario y la integración con el backend. | El estudiante muestra dificultades para mejorar la interacción entre la aplicación Ionic y el API Laravel usando Capacitor. La implementación es básica o incompleta, con problemas significativos en la integración, la experiencia de usuario y la eficiencia de la comunicación, indicando una comprensión limitada de las prácticas recomendadas y las capacidades de Capacitor. |
| RAA-3 | Desarrollo de funcionalidades CRUD en Ionic, utilizando servicios para la comunicación con el API de Laravel. | RAE-1 | Desarrollar y demostrar habilidades en la creación de funcionalidades CRUD en aplicaciones Ionic, empleando servicios Angular para interactuar con un API RESTful desarrollado en Laravel. |  | El estudiante será evaluado en su capacidad para diseñar, implementar y probar funcionalidades CRUD (Crear, Leer, Actualizar, Eliminar) en una aplicación desarrollada con Ionic, demostrando competencia técnica en la utilización de servicios Angular para la comunicación efectiva con un API RESTful creado en Laravel. | El estudiante crea una aplicación Ionic con funcionalidades CRUD completas, demostrando una integración avanzada y eficiente con el API de Laravel. La aplicación presenta una interfaz de usuario intuitiva y atractiva, seguridad robusta en las comunicaciones, y una excelente gestión de errores. El código es limpio, bien organizado y sigue las mejores prácticas de desarrollo. | El estudiante desarrolla una aplicación Ionic con funcionalidades CRUD bien implementadas, mostrando una buena integración con el API de Laravel. La interfaz de usuario es clara y funcional, y se aplican prácticas de seguridad adecuadas. Hay un manejo de errores efectivo y el código es coherente y mayormente sigue buenas prácticas. | El estudiante implementa las funcionalidades CRUD en Ionic con una integración básica al API de Laravel. La interfaz de usuario cumple con los requisitos mínimos y se observan intentos de implementar seguridad en la comunicación. El manejo de errores es limitado y el código necesita mejoras para alcanzar buenas prácticas de desarrollo. | El estudiante presenta dificultades significativas en la implementación de las funcionalidades CRUD en Ionic, con integración deficiente o incompleta con el API de Laravel. La interfaz de usuario es poco intuitiva o incompleta, faltan prácticas de seguridad en las comunicaciones, y el código es desorganizado o no sigue las prácticas recomendadas. |
| Diseño y creación de endpoints en Laravel para manejar operaciones CRUD y autenticación de usuarios, asegurando la comunicación efectiva con MongoDB. | RAE-2 | Desarrollar y Configurar Endpoints de API en Laravel para Operaciones CRUD y Autenticación de Usuarios con Integración en MongoDB |  | El estudiante será evaluado en su capacidad para conceptualizar, diseñar e implementar de manera efectiva una serie de endpoints en Laravel que no solo permitan realizar operaciones CRUD sobre una base de datos MongoDB, sino que también gestionen la autenticación de usuarios. Esto implica la creación de rutas claras, controladores eficientes y la integración segura con MongoDB, demostrando una comprensión profunda de la interacción entre el backend y la base de datos. | El estudiante diseña e implementa endpoints en Laravel con una estructura lógica y eficiente, demostrando innovación en la solución de problemas. Los endpoints manejan de manera segura y eficaz tanto las operaciones CRUD como la autenticación de usuarios, con una integración óptima y sin errores con MongoDB. El trabajo muestra un alto grado de optimización, seguridad y consideración por las mejores prácticas de desarrollo. | El estudiante desarrolla endpoints que cumplen con los requisitos funcionales para operaciones CRUD y autenticación de usuarios, integrándose adecuadamente con MongoDB. La implementación es segura y eficiente, aunque con margen de mejora en términos de optimización y uso de mejores prácticas. | El estudiante logra implementar endpoints para operaciones CRUD y autenticación de usuarios con una integración funcional en MongoDB. Sin embargo, la solución presenta deficiencias en términos de seguridad, optimización y estructuración del código, necesitando revisiones para alcanzar un estándar profesional. | El estudiante enfrenta dificultades significativas en el desarrollo de endpoints funcionales para CRUD y autenticación, con problemas en la integración con MongoDB. La implementación muestra fallos de seguridad, ineficiencias y una falta de comprensión de las prácticas recomendadas, requiriendo intervención significativa para corregir estos aspectos. |
| Implementación de operaciones CRUD en MongoDB a través de Laravel, incluyendo el modelado de esquemas de datos. | RAE-3 | Implementar operaciones CRUD en MongoDB utilizando Laravel, con énfasis en el modelado adecuado de esquemas de datos. |  | El estudiante demostrará su capacidad para integrar MongoDB con Laravel, diseñando y ejecutando operaciones CRUD (Crear, Leer, Actualizar, Eliminar) eficientemente. Esto incluye la habilidad para modelar esquemas de datos que reflejen las necesidades del sistema y permitan una manipulación de datos efectiva y organizada. | El estudiante no solo implementa todas las operaciones CRUD con precisión, sino que también demuestra una comprensión avanzada del modelado de esquemas de datos en MongoDB, optimizando las interacciones de la base de datos. Las soluciones propuestas son eficientes, seguras y muestran una integración sofisticada entre MongoDB y Laravel. El código es limpio, bien documentado, y sigue las mejores prácticas de desarrollo. | El estudiante implementa las operaciones CRUD correctamente y modela los esquemas de datos de manera efectiva, reflejando una buena comprensión de la integración entre MongoDB y Laravel. Hay un esfuerzo notable por optimizar el acceso a la base de datos y por asegurar las operaciones. El código es claro y mayormente sigue las buenas prácticas, con algunas áreas de mejora en eficiencia y seguridad. | El estudiante logra implementar las operaciones CRUD y el modelado de esquemas de datos con un nivel básico de comprensión. La integración entre MongoDB y Laravel cumple con los requisitos mínimos, pero hay deficiencias en la optimización y en la seguridad de las operaciones de datos. El código puede ser mejorable en términos de claridad y adhesión a las mejores prácticas. | El estudiante enfrenta dificultades significativas en la implementación de las operaciones CRUD y en el modelado de esquemas de datos. La integración entre MongoDB y Laravel es inadecuada, mostrando una comprensión limitada de cómo manipular y estructurar los datos eficazmente. Las operaciones de datos carecen de eficiencia y seguridad, y el código presenta errores fundamentales y no sigue las prácticas recomendadas. |