

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL

PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

1. **Unidad Académica:** Facultad de Ingeniería, Arquitectura y Diseño, Ensenada; Facultad Ciencias Químicas e Ingeniería, Tijuana; y Facultad de Ciencias de la Ingeniería y Tecnología, Valle de las Palmas
2. **Programa Educativo:** Ingeniero en Software y Tecnologías Emergentes
3. **Plan de Estudios:** 2022-1
4. **Nombre de la Unidad de Aprendizaje:** Validación y Verificación del Software
5. **Clave:** 40016
6. **HC:** 02 **HT:** 00 **HL:** 02 **HPC:** 00 **HCL:** 00 **HE:** 02 **CR:** 06
7. **Etapas de Formación a la que Pertenece:** Disciplinaria
8. **Carácter de la Unidad de Aprendizaje:** Obligatoria
9. **Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



Equipo de diseño de PUA

Irma Alejandra Amaya Patrón
J. Reyes Juárez Ramírez

Vo.Bo. de subdirector(es) de Unidad(es) Académica(s)

Humberto Cervantes De Ávila
Daniela Mercedes Martínez Platas
Noemí Hernández Hernández

Fecha: 20 de febrero de 2021

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

La verificación y la validación son mecanismos que se aplican a lo largo del proceso de desarrollo de software, permitiendo asegurar la calidad con base a las especificaciones/requerimientos, para satisfacer las expectativas del cliente/usuario.

La utilidad de esta unidad de aprendizaje radica en que le permite al estudiante adquirir habilidades de diseño de escenarios, aplicación de estrategias de verificación y validación utilizando pruebas, para asegurar la calidad del software.

Se imparte en la etapa disciplinaria con carácter obligatorio y pertenece al área de conocimiento Métodos y Tecnologías de Software.

III. COMPETENCIA GENERAL DE LA UNIDAD DE APRENDIZAJE

Diseñar estrategias de verificación y validación, utilizando pruebas y artefactos, para asegurar que el software cumple con los requerimientos, con una actitud crítica, metódica y respeto a los estándares.

IV. EVIDENCIA(S) DE APRENDIZAJE

Plan de verificación y validación para un proyecto de software real que contenga los siguientes elementos: planeación, ejecución y un reporte de seguimiento de problemas.

V. DESARROLLO POR UNIDADES
UNIDAD I. Fundamentos de la verificación y validación de software

Competencia:

Identificar los fundamentos de la verificación y validación de software, a través de la revisión de estándares y modelos, para valorar la importancia y los beneficios en el aseguramiento de la calidad acorde a los requisitos del usuario, con una actitud crítica y analítica.

Contenido:

Duración: 6 horas

- 1.1. Terminología y fundamentos de la V&V.
- 1.2. Objetivos y limitaciones de V&V.
- 1.3. Métricas y mediciones para la V&V.
- 1.4. Participación de V&V en diferentes puntos del ciclo de vida.
- 1.5. Ventajas de V&V con respecto al aseguramiento de la calidad de software.
- 1.6. Estándares y modelos de procesos para V&V
 - 1.6.1 Estándar IEEE 1012 V&V
 - 1.6.2 Niveles de integridad
 - 1.6.3 Recomendaciones V&V para requerimientos de software
- 1.7. El estándar ISO/IEC/IEEE 1220 -Verificación y Validación
 - 1.7.1 Procesos de verificación
 - 1.7.2 Proceso de validación
- 1.8. V&V en otros modelos de procesos
 - 1.8.1 Modelo CMMI Model: Verificación y Validación
 - 1.8.2 ISO/IEC 29110 and V&V: Verificación y Validación
 - 1.8.3 V&V en metodologías ágiles: Scrum

UNIDAD II. Técnicas de verificación y validación

Competencia:

Diferenciar las distintas técnicas de verificación y validación, mediante una revisión de los tipos de pruebas, para proponer las más adecuadas de acuerdo a la naturaleza del proyecto de software, con actitud analítica y propositiva.

Contenido:

Duración: 8 horas

2.1 Técnicas de verificación: pruebas estáticas

- 2.1.1 Inspecciones: de diseño, de código
- 2.1.2 Revisiones
- 2.1.3 "Walkthroughs"
- 2.1.4 "Desk-checking", "Checklists"
- 2.1.5 Auditorías

2.2 Técnicas de validación: pruebas dinámicas

- 2.2.1 Pruebas de caja negra
- 2.2.2 Pruebas de caja blanca
- 2.2.3 Pruebas unitarias
- 2.2.4 Pruebas de integración

2.3 Trazabilidad

- 2.3.1 Matriz de trazabilidad
- 2.3.2 Implementación de la trazabilidad
- 2.3.3 Validación a través de las fases de desarrollo

UNIDAD III. Implementación de pruebas de software

Competencia:

Aplicar técnicas de pruebas, mediante una planeación eficiente de escenarios y herramientas adecuadas, para verificar y validar la calidad de software, con actitud crítica y metódica.

Contenido:**Duración:** 10 horas

- 3.1. Desarrollo guiado por pruebas.
- 3.2. Manejo de excepciones.
- 3.3. Análisis de cobertura y pruebas basadas en estructura.
- 3.4. Desarrollo de casos de prueba basados en escenarios.
- 3.5. Pruebas basadas en perfiles operativos.
- 3.6. Pruebas de sistema y aceptación.
- 3.7. Pruebas a través de atributos de calidad.
- 3.8. Pruebas de regresión.
- 3.9. Herramientas de prueba y automatización.
- 3.10. Pruebas de interfaz de usuario: aspectos físicos.
- 3.11. Pruebas de usabilidad: percepción del usuario.
- 3.12. Pruebas de rendimiento.

UNIDAD IV. Planes de verificación y validación

Competencia:

Implementar un plan de verificación y validación, mediante la integración de técnicas y herramientas pertinentes, para asegurar la calidad de un producto de software, con actitud crítica, sistemática y con respeto a los estándares.

Contenido:

Duración: 8 horas

- 4.1 Planes de verificación y validación: estructura.
- 4.2 Alcances y limitaciones de los planes de V&V.
- 4.3 Convergencia del plan de V&V con el aseguramiento de la calidad.
- 4.4 Factores de éxito de la V&V.
- 4.5 Informes de fallos.
- 4.6 Técnicas de depuración y aislamiento de fallas.
- 4.7 Análisis de defectos y causas.
- 4.8 Seguimiento de problemas.

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD II				
1	Diseña una prueba estática para verificación	1. Atiende las orientaciones del profesor para elaborar el diseño de una prueba estática para verificación 2. Selecciona el artefacto de software para realizar la prueba estática (requerimientos, diseño, código, etc.). 3. Selecciona la prueba estática a aplicar. 4. Establece el procedimiento a realizar con base a la prueba seleccionada. 5. Ejecuta la prueba. 6. Analiza resultados de la prueba. 7. Elabora y entrega reporte de la prueba realizada.	• Computadora • Proyector • Internet • Artefacto de software • Entorno de ejecución	3 horas
2	Diseña una prueba dinámica para validación	1. Atiende las orientaciones del profesor para elaborar el diseño de una prueba dinámica para validación. 2. Selecciona la prueba dinámica a aplicar. 3. Establece el procedimiento a realizar con base a la prueba seleccionada. 4. Ejecuta la prueba. 5. Analiza resultados de la prueba. 6. Elabora y entrega reporte de la prueba realizada.	• Computadora • Proyector • Internet • Código fuente • Entorno de ejecución	3 horas

3	Diseña un mecanismo de trazabilidad	1. Atiende las orientaciones del profesor para elaborar el diseño de un mecanismo de trazabilidad. 2. Selecciona las fases del proceso de desarrollo para enlazar en la trazabilidad. 3. Selecciona los artefactos de cada fase para enlazar en la trazabilidad. 4. Diseña la matriz de trazabilidad con base a los artefactos. 5. Construye la matriz de trazabilidad. 6. Analiza las conexiones y correspondencia entre los artefactos hacia-atrás y hacia-adelante. 7. Valora el cumplimiento de requerimientos de calidad. 8. Elabora y entrega reporte de la trazabilidad.	• Computadora • Proyector • Internet • Artefactos de software • Entorno de ejecución	3 horas
UNIDAD III				
4	Diseña casos de prueba basados en escenario	1. Atiende las orientaciones del profesor para elaborar el diseño un caso de prueba. 2. Analiza los posibles escenarios de ejecución del software. 3. Selecciona los escenarios para diseñar casos de prueba. 4. Determina los pasos, variables y validaciones a realizar en cada escenario. 5. Ejecuta los casos de prueba para los escenarios considerados.	• Computadora • Proyector • Internet	3 horas

		6. Elabora un reporte de los casos de prueba.		
5	Diseña una prueba automatizada	1. Atiende las orientaciones del profesor para elaborar el diseño de una prueba automatizada. 2. Selecciona los módulos o componentes para aplicar la prueba automatizada. 3. Determina el entorno para realizar la prueba automatizada. 4. Establece la estructura de la prueba automatizada. 5. Ejecuta la prueba automatizada. 6. Elabora un reporte de la prueba automatizada.	• Computadora • Proyector • Internet • Artefactos de software • Entorno de ejecución	3 horas
6	Diseña una prueba de usabilidad	1. Atiende las orientaciones del profesor para elaborar el diseño de una prueba de usabilidad. 2. Selecciona los módulos o componentes para realizar la prueba de usabilidad. 3. Selecciona a los usuarios para realizar la prueba de usabilidad. 4. Determina el entorno para realizar la prueba de usabilidad. 5. Establece la estructura de la prueba de usabilidad. 6. Ejecuta la prueba de usabilidad. 7. Elabora un reporte de la prueba de usabilidad.	• Computadora • Proyector • Internet • Artefactos de software • Entorno de ejecución	3 horas
7	Diseña una prueba de aceptación	1. Atiende las orientaciones del profesor para elaborar el diseño de una prueba de aceptación.	• Computadora • Proyector • Internet • Artefactos de software • Entorno de ejecución	3 horas

		2. Selecciona los módulos o componentes para realizar la prueba de aceptación. 3. Selecciona a los usuarios para realizar la prueba de aceptación. 4. Determina el entorno para realizar la prueba de aceptación. 5. Establece la estructura de la prueba de aceptación. 6. Ejecuta la prueba de aceptación. 7. Elabora un reporte de la prueba de aceptación.		
UNIDAD IV				
8	Diseña un plan de V&V	1. Atiende las orientaciones del profesor para elaborar el diseño de un plan de V&V. 2. Selecciona el proyecto para realizar la V&V. 3. Determina la estructura del plan de V&V. 4. Selecciona los artefactos sobre los cuales hacer la V&V. 5. Establece las actividades y controles para realizar la V&V. 6. Determina las herramientas de medición para la V&V. 7. Presenta el plan de V&V.	• Computadora • Proyector • Internet • Artefactos de software	5 horas
9	Diseña e implementa un reporte de fallos	1. Atiende las orientaciones del profesor para elaborar el diseño de un reporte de fallos. 2. Selecciona los módulos/componentes sobre los cuales reportar los fallos.	• Computadora • Proyector • Internet	3 horas

		3. Ejecuta las pruebas para detección de fallos. 4. Registra los fallos encontrados. 5. Realiza y presenta el reporte de fallos.		
10	Implementa un análisis de defectos y causas.	1. Atiende las orientaciones del profesor para elaborar el diseño de una estrategia de análisis de defectos y causas. 2. Selecciona los tipos de defectos a considerar. 3. Determina las variables a considerar para determinar las causas de fallos. 4. Establece una relación entre fallos y causas. 5. Realiza el análisis de las causas. 6. Elabora y presenta un reporte de causas por defecto.	• Computadora • Proyector • Internet	3 horas

VII. MÉTODO DE TRABAJO

Encuadre: El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

Estrategia de enseñanza (docente):

- Técnica expositiva para la introducción a los conceptos formales.
- Aprendizaje basado en problemas.
- Ejercicios prácticos para el diseño y ejecución de pruebas.
- Aprendizaje basado en proyectos.

Estrategia de aprendizaje (alumno):

- Investigación.
- Trabajo en equipo.
- Realización de prácticas de laboratorio para la aplicación de pruebas.
- Elaboración de reportes de resultados y seguimiento de problemas.
- Elaboración de plan de verificación y validación.

VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

Criterios de acreditación

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir con los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

Criterios de evaluación

- Evaluaciones parciales.....	40%
- Prácticas de laboratorio	40%
- Plan de V&V	20%
Total.....	100%

.

IX. REFERENCIAS

Básicas

- Blokdyk, G. (2020). *Software Verification And Validation A Complete Guide - 2020 Edition (English Edition)*. USA; 5STARCook.
- Shah, N. (2019). *Software Quality Assurance and Testing for Beginners*. Exeter, United Kingdom; Independently published.
- R. S. Mann. (2019). *Software Testing and Quality Assurance*. Beau Bassin, Mauritius; LAP LAMBERT Academic Publishing
- Laporte, C. y., April, A. (2018). *Software Quality Assurance*. Hoboken, NJ, USA; Wiley IEEE Press.

Complementarias

- Debbabi M., Hassaïne F., Jarraya Y., Soeanu A., Alawneh L. (2010) *Verification and Validation in Systems Engineering*. Berlin, Heidelberg. Springer. Recuperado de https://libcon.rec.uabc.mx:4440/10.1007/978-3-642-15228-3_1 [clásica]
- IEEE. (2012). P1012 - Standard for System, Software, and Hardware Verification and Validation. Recuperado de <https://standards.ieee.org/project/1012.html> [clásica]
- Marcus S. Fisher. (2007). *Software Verification and Validation.*, Boston, MA.: Springer. Recuperado de https://libcon.rec.uabc.mx:4440/10.1007/978-0-387-47939-2_1 [clásica]

X. PERFIL DEL DOCENTE

El docente que imparta la unidad de aprendizaje de Verificación y Validación de Software debe contar con título de Ingeniero en Software, Licenciado en Ciencias de la Computación o afín, preferentemente con estudios de posgrado en Ciencias de la Computación, con conocimientos de Ingeniería de Requerimientos, Ingeniería de Procesos de Software, Ingeniería y Pruebas de Software, con experiencia en dirección de proyectos de software, y al menos dos años de experiencia docente. Debe ser organizado, metódico, analítico y crítico.