

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:** Herramientas de Desarrollo de Software
- 5. Clave:** 40017
- 6. HC:** 01 **HT:** 02 **HL:** 02 **HPC:** 00 **HCL:** 00 **HE:** 01 **CR:** 06
- 7. Etapa 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

Luz Evelia López Chico
Rodrigo Lara Melgoza

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 finalidad de la unidad de aprendizaje es eficientar el diseño e implementación de software con la utilización de los servicios integrales que ofrece el entorno integrado de desarrollo.

La utilidad de esta radica en que le facilita al estudiante las herramientas para el desarrollo de software.

Se imparte en la etapa disciplinaria con carácter obligatorio. Pertenece al área de conocimiento Desarrollo Tecnológico

III. COMPETENCIA GENERAL DE LA UNIDAD DE APRENDIZAJE

Desarrollar software, aplicando las herramientas de apoyo de los entornos integrados de desarrollo, para eficientar el diseño e implementación de software, de manera organizada y metódica, con una actitud de colaboración.

IV. EVIDENCIA(S) DE APRENDIZAJE

Producto de software basado en un marco de trabajo o bibliotecas de clases para algún dominio específico, utilizando herramientas de apoyo para asegurar la calidad del producto. Asimismo, deberá cumplir con las características de entrega que el docente determine.

V. DESARROLLO POR UNIDADES
UNIDAD I. Entornos integrados de desarrollo.

Competencia:

Analizar los entornos integrados de desarrollo, a partir de la identificación de sus características, tipos y usos, para emplearlos en el desarrollo de aplicaciones, con actitud reflexiva, sistemática y proactiva.

Contenido:

- 1.1 Definición y antecedentes
- 1.2 Tipos de entornos integrados de desarrollo
- 1.3 Ejemplos y uso de entornos integrados de desarrollo

Duración: 3 horas

UNIDAD II. Generadores de interfaz de usuario

Competencia:

Examinar los generadores de interfaz de usuario, a partir de la identificación de sus características, tipos y usos, para emplearlos en el desarrollo de aplicaciones, con actitud analítica, metódica y proactiva.

Contenido:

Duración: 3 horas

2.1 Definición y antecedentes

2.2 Tipos de generadores de interfaz de usuario

2.3 Ejemplos y uso de generadores de interfaz de usuario para diferentes tipos de aplicaciones

UNIDAD III. Herramientas para acelerar la construcción de software

Competencia:

Diferenciar las herramientas para acelerar la construcción de software, a partir del análisis de sus características, versiones y usos, para integrarlas de una manera eficiente en el proceso de desarrollo de software, con actitud creativa y sistemática.

Contenido:

Duración: 4 horas

- 3.1 Definiciones y antecedentes.
- 3.2 Bibliotecas de funciones y clases.
- 3.3 Marcos de trabajo.
- 3.4 Interfaces de programación de aplicaciones (API).
- 3.5 Ejemplos y utilización de las herramientas más importantes.

UNIDAD IV. Herramientas para pruebas unitarias

Competencia:

Diferenciar las herramientas para pruebas unitarias, a partir de la identificación de sus tipos y el análisis de sus características, métodos y usos, para validar el correcto funcionamiento del código, con responsabilidad, actitud creativa y sistemática.

Contenido:

Duración: 3 horas

- .1 Definición y antecedentes
- 4.2 Tipos de herramientas para pruebas unitarias
- 4.3 Ejemplos y uso de herramientas para pruebas unitarias

UNIDAD V. Herramientas de análisis de perfilado y rendimiento.

Competencia:

Distinguir las herramientas de análisis de perfilado y rendimiento, a partir de la identificación de sus tipos y el análisis de sus características, métodos y usos, para evaluar el rendimiento del software, con honestidad, actitud creativa y metódica.

Contenido:

- 5.1 Definición y antecedentes
- 5.2 Tipos de herramientas para el perfilado y rendimiento
- 5.3 Ejemplos y uso de herramientas de perfilado y rendimiento.

Duración: 3 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE TALLER

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD I				
1	Uso de entornos de desarrollo integrado (IDE)	<ol style="list-style-type: none"> 1. Atiende las orientaciones del profesor para el uso de los entornos de desarrollo integrado. 2. Identifica los componentes que se incluyen en el entorno de desarrollo integrado. 3. Investiga los diferentes tipos de entornos de desarrollo. 4. Analiza las diferencias entre los diversos entornos de desarrollo integrado. 5. Realiza y entrega el cuadro comparativo al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Lápiz o pluma ● Papel o cuaderno ● Computadora 	6 horas
UNIDAD II				
2	Uso de generadores de interfaz de usuario para diferentes tipos de aplicaciones	<ol style="list-style-type: none"> 1. Atiende las orientaciones del profesor para el uso de los generadores de interfaz de usuario 2. Identifica los componentes que se incluyen en el generador de interfaz de usuario 3. Investiga los diferentes tipos de generadores de interfaz de usuario 4. Analiza las características, tipos y usos de los diferentes generadores de interfaz de usuario 5. Caracteriza las diferentes aplicaciones para los generadores de interfaz de usuario 	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Lápiz o pluma ● Papel o cuaderno ● Computadora 	6 horas

		6. Realiza y entrega cuadro comparativo al docente para su retroalimentación.		
UNIDAD III				
3	Uso de herramientas para acelerar la construcción de software	<p>1. Atiende las orientaciones del profesor respecto al uso de los diferentes tipos de herramientas para acelerar la construcción de software como:</p> <ul style="list-style-type: none"> - Bibliotecas de funciones y clases - Marcos de trabajo - Interfaces de programación de aplicaciones (API) <p>2. Identifica las características de las herramientas para acelerar la construcción de software</p> <p>3. Elabora y entrega el cuadro comparativo realizado al docente para su retroalimentación.</p>	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Lápiz o pluma ● Papel o cuaderno ● Computadora 	8 horas
UNIDAD IV				
4	Uso de herramientas para pruebas unitarias	<p>1. Atiende las orientaciones del docente sobre el uso de herramientas para pruebas unitarias en bloques de códigos.</p> <p>2. Identifica las características de las herramientas para pruebas unitarias</p> <ul style="list-style-type: none"> - automática - ejecutarse en cualquier plataforma - replicarse cuanto se requiera <p>3. Investiga los diferentes tipos de herramientas unitarias</p>	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Lápiz o pluma ● Papel o cuaderno ● Computadora 	6 horas

		4. Analiza las diferencias entre los distintos tipos de pruebas unitarias 5. Realiza y entrega el cuadro comparativo al docente para su retroalimentación.		
UNIDAD V				
5	Uso de herramientas de análisis de perfilado y rendimiento.	1. Atiende las orientaciones del profesor para el uso de las herramientas de análisis de perfilado y rendimiento 2. Identifica las características de cada tipo de herramienta de análisis de perfilado y rendimiento. 3. Analiza las diferencias entre los distintos tipos de herramientas de análisis de perfilado y rendimiento. 4. Realiza y entrega un ejemplo de la aplicación de herramientas de perfilado y rendimiento al docente para su retroalimentación.	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Lápiz o pluma ● Papel o cuaderno ● Computadora 	6 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD I				
1	Uso de entornos de desarrollo integrado (IDE)	1. Utiliza el entorno de desarrollo integrado. 2. Experimenta con los componentes que se incluyen en el entorno de desarrollo integrado.	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Computadora ● Software de entorno de desarrollo integrado 	6 horas

		<p>4. Desarrolla software utilizando un entorno de desarrollo integrado.</p> <p>5. Entrega la actividad al docente para su retroalimentación.</p>		
UNIDAD II				
2	Uso de generadores de interfaz de usuario para diferentes tipos de aplicaciones	<p>1. Utiliza el generador de interfaz de usuario</p> <p>2. Experimenta con los componentes que se incluyen en el generador de interfaz de usuario</p> <p>3. Elabora una interfaz de usuario que contemple al menos los siguientes elementos:</p> <ul style="list-style-type: none"> - entrada de datos - visualización de información - manejo de menús - botones - cajas de búsqueda <p>4. Entrega la actividad al docente para su retroalimentación.</p>	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Computadora ● Software de entorno de desarrollo integrado ● Software de interfaz de usuario 	6 horas
UNIDAD III				
3	Uso de herramientas para acelerar la construcción de software	<p>1. Experimenta con los diferentes tipos de herramientas para acelerar la construcción de software como:</p> <ul style="list-style-type: none"> - Bibliotecas de funciones y clases - Marcos de trabajo - Interfaces de programación de aplicaciones (API) <p>2. Construye software empleando las herramientas que va a utilizar empleando al menos las siguientes herramientas: Bibliotecas de funciones y clases, Marcos de trabajo, e Interfaces de</p>	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Computadora ● Software de entorno de desarrollo integrado ● Herramienta de aceleración de construcción de software 	8 horas

		<p>programación de aplicaciones (API).</p> <p>3. Entrega la actividad realizada al docente para su retroalimentación.</p>		
UNIDAD IV				
4	Uso de herramientas para pruebas unitarias	<p>1. Experimenta el uso de herramientas para pruebas unitarias en bloques de códigos.</p> <p>2. Elabora reporte de los resultados de las pruebas unitarias</p> <p>3. Entrega reporte al docente para su retroalimentación.</p>	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Computadora ● Software de entorno de desarrollo integrado 	6 horas
UNIDAD V				
5	Uso de herramientas de análisis de perfilado y rendimiento.	<p>1. Experimenta con las herramientas de análisis de perfilado y rendimiento.</p> <p>2. Construye software</p> <p>3. Analiza el rendimiento del software creado utilizando las herramientas de análisis de perfilado y rendimiento.</p> <p>4. En función de los resultados del reporte realiza las modificaciones pertinentes.</p> <p>5. Entrega al docente la actividad completa para su retroalimentación.</p>	<ul style="list-style-type: none"> ● Recursos bibliográficos ● Computadora ● Software de entorno de desarrollo integrado 	6 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):

- Estudio de caso
- Método de proyectos
- Aprendizaje basado en problemas
- Técnica expositiva
- Ejercicios prácticos

Estrategia de aprendizaje (alumno):

- Investigación
- Estudio de caso
- Trabajo en equipo
- Exposiciones
- Organizadores gráficos
- Cuadros comparativos

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.....	30%
- Prácticas de taller y tareas.....	20%
- Prácticas de laboratorio.....	30%
- Producto de software	20%
Total.....	100%

IX. REFERENCIAS

Básicas

- Molyneaux, I. (2015). *The art of application performance testing*. Estados Unidos: O'Reilly.[clásica]
- Rossel, S. (2017). *Continuous Integration, Delivery and Deployment*. Reino Unido: Packt Publishing.
- Uzayr, S. B, Claud, N., Ambler, T. (2019). *JavaScript Frameworks for modern web development*. Estados Unidos: Apress.
- Zammeti, F. (2020). *Modern full-stack development*. Estados Unidos: Apress.

Complementarias

- Adamson, C. (2018). *Xcode Treasures*. Estados Unidos: Pragmatic Bookshelf.
- Banks, A., Porcello, E. (2020). *Learning React*. Estados Unidos: O'Reilly.
- Bennett, J. (2018). *Xamarin in action*. Estados Unidos: Manning Publications.
- Reid, J. (2020). *IOS Unit Testing by Example: Xctest Tips and Techniques Using Swift*. Estados Unidos: Pragmatic Bookshelf.
- Redmond-Pyle, D. (Marzo, 1996). Software development methods and tools: some trends and issues. *Software Engineering Journal*, 11 (2), 99-103. Recuperado de: <https://ieeexplore.ieee.org/document/487421> [clásica].
- Smyth, N. (2020). *Android Studio 4.0 Development Essentials*. Estados Unidos: Payload Media.
- Zaccagnino, C. (2020). *Programming Flutter*. Estados Unidos: Pragmatic Bookshelf.

X. PERFIL DEL DOCENTE

El docente que imparta la unidad de aprendizaje Herramientas de Desarrollo de Software debe contar con título de Ingeniero de software, Licenciado en ciencias de la computación o área afín, preferentemente con estudios de posgrado en ciencias de la computación. Debe contar con conocimientos de programación orientada a objetos, en el uso de bibliotecas y marcos de trabajo, patrones de software y herramientas de apoyo al desarrollo de software; al menos dos años de experiencia docente. Asimismo, debe ser creativo, organizado y analítico.