

Programas de Asignatura

Programación Creativa

A. Antecedentes Generales

1. Unidad Académica	Facultad de Diseño					
2. Carrera	Diseño					
3. Código de la asignatura	DICV311					
4. Ubicación en la malla	5º semestre, 3º año					
5. Créditos	6					
6. Tipo de asignatura	X	Obligatorio		Electivo		Optativo
7. Duración	17 Semanas					
8. Módulos semanales		Teóricos	2	Prácticos	1	Ayudantía
9. Horas académicas	68	Hrs. de Clase			34	Hrs. de Ayudantía
10. Pre-requisito	Representación de la Información					

Competencias de la Asignatura

	Competencias Genéricas		Competencias de Innovación		Competencias de Investigación		Competencias Tecnológicas
	Ética	X	Creatividad		Observación y Conceptualización	X	Representación y Visualización
	Emprendimiento y Liderazgo		Empatía		Herramientas Metodológicas	X	Dominio de Herramientas Tecnológicas y Procesos de Producción
	Responsabilidad Pública		Trabajo en Equipo		Jerarquización de la Información		Dominio y Uso de Materiales
X	Autonomía		Persuasión		Juicio Crítico		
	Eficiencia		Pensamiento Estratégico				
	Visión Global						
	Visión Analítica						
	Comunicación						

B. Aporte al Perfil de Egreso

La asignatura Programación Creativa contiene los elementos para el aprendizaje de lógicas computacionales en lenguajes de programación para la realización, por medio de un trabajo práctico de manera individual, de resultados gráficos de alta complejidad, en formatos digitales y análogos; y la resolución de problemas por medio de la identificación y caracterización de cada parte dentro de un total.

A través del trabajo personal y discusión grupal, el estudiante desarrollará procesos creativos y de documentación hacia el análisis crítico de un entorno contemporáneo cambiante en torno a la innovación tecnológica.

La asignatura se dicta en el ciclo de Licenciatura, pertenece a la línea Representación y medios, aportando en el desarrollo de la competencia genérica de **Creatividad** y **Autonomía**, así como las específicas de **Representación y Visualización** y **Dominio de Herramientas Tecnológicas y Procesos de Producción**.

C. Competencias y Resultados de Aprendizaje que desarrolla la asignatura

COMPETENCIAS GENÉRICAS	RESULTADOS DE APRENDIZAJE GENERALES
<ul style="list-style-type: none"> Autonomía 	<p>Aplica el lenguaje procedimental de la computación con lógica y metodología a fin de llegar a resultados gráficos de alta complejidad expresados en soportes análogos y digitales.</p> <ul style="list-style-type: none"> Analiza la construcción del proyecto, comprendiendo su elaboración, programación dentro de un sistema complejo. Valora el uso de procesos de documentación contemporáneos y el impacto que tiene en el trabajo creativo. Selecciona intencionalmente y de manera autónoma procedimientos adecuados frente a los desafíos del mundo académico y del trabajo, autorregulando y evaluando la programación del trabajo.
COMPETENCIAS ESPECÍFICAS	
<ul style="list-style-type: none"> Creatividad Representación y Visualización Dominio de Herramientas Tecnológicas y Procesos de Producción 	

D. Unidades de Contenidos y Resultados de Aprendizaje

UNIDADES DE CONTENIDOS	COMPETENCIA	RESULTADOS DE APRENDIZAJE
<p>UNIDAD I: INTRODUCCIÓN A LA PROGRAMACIÓN CREATIVA</p> <p>1.1 Inicios de la programación, programación creativa, escenario actual y posibilidades futuras.</p> <p>1.1.1 Niveles de programación humano/computadora</p> <p>1.1.2 Entornos y flujos contemporáneos de programación creativa</p> <p>1.1.3 Resultados actuales</p> <p>1.2 Lógica y Lenguaje</p> <p>1.2.1 Lenguajes de programación para la programación creativa</p>	<ul style="list-style-type: none"> Creatividad Herramientas tecnológicas y procesos de producción 	<ul style="list-style-type: none"> Identifica los inicios de la programación a través del estudio de casos Reconoce la programación creativa, escenario actual y posibilidades futuras, a través del estudio de casos Convierte el lenguaje común en lenguaje para la programación mediante el trabajo de pares. Construye figuras básicas incorporando condicionantes mediante el trabajo de pares. Utiliza lenguajes de programación para construir figuras básicas considerando las variables y condicionales la

<p>1.2.2 Pseudo código</p> <p>1.3 Programación de figuras básicas y condicionales</p> <p>1.3.1 Variables básicas y condicionales</p> <p>1.3.2 Documentación</p>		<p>programación</p>
<p>UNIDAD II: PROCESO DE PROGRAMACION</p> <p>2.1 Flujo de trabajo:</p> <p>2.1.1 Deconstrucción de proyecto</p> <p>2.1.2 Análisis de partes</p> <p>2.1.3 Pseudo Código</p> <p>2.1.4 Programación</p> <p>2.2 Programación de figuras básica, condicionales e interacciones</p> <p>2.2.1 Variables complejas</p> <p>2.2.2 Interacción</p> <p>2.2.3 Formatos de resultado</p>	<ul style="list-style-type: none"> • Creatividad • Representación y visualización • Herramientas tecnológicas y procesos de producción • Autonomía 	<p>.Utiliza referentes visuales para analizar y contrastar información desde el flujo de trabajo a programación de figuras básica.</p> <p>Utiliza diferentes recursos visuales y herramientas análogas - digitales para representar las etapas del flujo de trabajo tanto y programación de figuras básica.</p> <p>Construye figuras básicas utilizando condicionantes e interacciones mediante el trabajo practico</p> <p>Explora posibilidades formales, buscando de manera autónoma la resolución de problemas, enfrentando desafíos como una oportunidad de mejora.</p>
<p>UNIDAD III: VISUALIZACIÓN DE DATOS Y NUEVOS FORMATOS</p> <p>3.1 Programación de figuras básica, condicionales, interacciones y base de datos</p> <p>3.2 Visualización de datos</p> <p>3.2.1 Tipos de datos</p> <p>3.2.2 Estructura</p> <p>3.2.3 Obtención y limpieza de datos</p> <p>3.2.4 Incorporación al programa</p> <p>3.3 Nuevos formatos</p> <p>3.3.1 Programación en la web</p>	<ul style="list-style-type: none"> • Creatividad • Representación y visualización • Herramientas tecnológicas y procesos de producción 	<p>Utiliza la construcción de figuras básicas para crear las condicionantes necesarias e interacciones en bases de datos.</p> <p>Debería decir por ejemplo: Maneja las herramientas tecnológicas para migrar programas desde diferentes bases, cómo por ejemplo de un escritorio a la web.</p> <p>Identifica el potencial de diferentes formatos de programación para resolver situaciones y casos.</p>

E. Estrategias de Enseñanza

En la formación basada en competencias el proceso de enseñanza-aprendizaje se enfoca en el desarrollo de conocimientos, habilidades y destrezas, y en su aplicación a la resolución de problemas similares a los que un profesional debe enfrentar en el mundo del trabajo.

Requiere:

- Lograr profundidad en el conocimiento
- Promover pensamiento de orden superior, como análisis, síntesis, aplicación, evaluación, resolución de problemas.
- Diseñar experiencias de aprendizaje activo (práctico), contextualizado (enfrentar situaciones reales), social (en interacción con otros) y reflexivo (evaluar el propio aprendizaje y generar estrategias para mejorar).
- Implementar estrategias de enseñanza variadas y auténticas (similares a las que se encuentran en el mundo del trabajo).

Diseño UDD ha definido un conjunto de metodologías de enseñanza que ofrecen una amplia gama de posibilidades para promover aprendizajes efectivos y relevantes en los estudiantes. Para esta asignatura se sugiere dar prioridad a las siguientes estrategias:

- Clase expositiva
- Esquemas y organizadores gráficos
- Ejercicio práctico
- Bitácora
- Portafolio
- Presentación oral y/o de proyectos

F. Estrategias de Evaluación

La evaluación debe estar presente a lo largo de todo el semestre o bimestre, ya sea para identificar los conocimientos previos de los alumnos (evaluación diagnóstica), monitorear la efectividad del proceso de enseñanza-aprendizaje (evaluación formativa), verificar el nivel de logro de los resultados de aprendizaje y calificar el desempeño de los estudiantes (evaluación sumativa).

- Los **procedimientos de evaluación** permiten evidenciar el desempeño de los alumnos a través de la elaboración de distintos tipos de documentos o productos (textos escritos, presentaciones orales, pruebas, propuestas formales en soportes bi y tridimensionales, audiovisuales, desarrollo de proyectos, etc.). El profesor debe privilegiar aquellos que permitan integrar conocimientos y aplicarlos en función de resolver situaciones auténticas (similares a las que aborda un diseñador profesional).
Se deben utilizar al menos 2 procedimientos de evaluación diferentes a lo largo del curso, de manera de abordar diferentes complejidades y profundidades de conocimiento.
- Los **instrumentos de evaluación** permiten analizar la producción de los alumnos, mediante criterios claros, transparentes y objetivos; verificar en qué medida se cumplen los resultados de aprendizaje y cuantificar el nivel de logro a través de un puntaje y una nota. Dependiendo del tipo de contenido, se sugiere utilizar: listas de cotejo, escala de valoración o rúbrica. El instrumento de evaluación debe ser entregado al alumno junto con los criterios de evaluación, a lo menos un mes antes de su aplicación.

Instancias de evaluación:

Se deberán realizar al menos 4 evaluaciones calificadas durante el semestre, que en su totalidad podrán:

- Sumar el 100% de la Nota de Presentación a Examen, donde a su vez ésta equivaldrá al 70% de la nota final de la asignatura. Dejando 30% para el Examen Final.

- Sumar el 70% como promedio de la asignatura previo al Examen, dejando 30% para el Examen Final. Ninguna evaluación por sí sola podrá ponderar más del 25% de la nota total del curso.

Examen Final:

Se realizará un examen final, con una ponderación del 30% de la nota total del curso. La fecha de esta evaluación será fijada por el Calendario Académico de la Facultad de forma semestral.

A criterio de la Facultad de Diseño, se podrán establecer comisiones revisoras para calificar el examen final. En dicho caso, las calificaciones emitidas por estas comisiones equivaldrá al 70% de la nota del examen y el 30% restante será determinado por el o los profesores del curso.

G. Recursos de Aprendizaje

Bibliografía Obligatoria:

- Casey Reas and Ben Fry. (2014) Processing: A Programming Handbook for Visual Designers, (2nd Edition) The MIT Press.
- Daniel Shiffman. (2015) Learning Processing, Second Edition: A Beginner's Guide to Programming Images, Animation, and Interaction. Morgan Kaufmann.
- Daniel Shiffman. (2012). The Nature of Code: Simulating Natural Systems with Processing. PDF, Web.
- Data Flow. (2008). Visualising Information in Graphic Design. Gestalten.
- Data Flow 2. (2010). Visualising Information in Graphic Design. Shape.

Bibliografía Complementaria:

- The Coding Train (2016). Accompanying code and more for YouTube video tutorials
<https://github.com/CodingTrain/Rainbow-Code>