



Fundamentos de  
Informática  
**Grado en Diseño de  
interiores**



UNIVERSIDAD  
**NEBRIJA**

## GUÍA DOCENTE

**Asignatura:** Fundamentos de Informática

**Titulación:** Grado en Diseño de Interiores

**Carácter:** Básica

**Idioma:** Castellano

**Modalidad:** Presencial

**Créditos:** 6

**Curso:** 2º

**Semestre:** 1º

**Profesores/Equipo Docente:** D. Yoseph Daniel Sabbagh Pastor / D. Santiago Gil Begue

### 1. COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

#### 1.1. Competencias

##### Competencias generales

- CG6 Conocimientos de informática relativos al ámbito de estudio.
- CG7 Capacidad de gestión de la información
- CG10 Familiaridad con programas informáticos de relación general

##### Competencias específicas

- CET26 Que los estudiantes hayan demostrado poseer, comprender y aplicar los principales conceptos implicados en la programación de Internet así como aplicar los diferentes lenguajes de programación para el desarrollo integral de una página Web.

#### 1.2. Resultados de aprendizaje

Los efectos que cabe asociar a la realización por parte de los estudiantes de las actividades formativas anteriormente indicadas son los conocimientos de la materia, la aplicación con criterio los métodos de análisis y técnicas descritos en ella, redactar utilizando un lenguaje preciso y adecuado a la misma, y aprender por sí mismo otros conocimientos relacionados con la materia, que se demuestran:

- En la realización de los exámenes parcial, final y extraordinario en su caso.
- En la entrega del trabajo de asignatura obligatorio que el alumno debe entregar
- En la presentación e intervenciones orales en clase.

## 2. CONTENIDOS

### 2.1. Requisitos previos

Ninguno

### 2.2. Descripción de los contenidos

- Arquitectura de ordenadores
- Redes de ordenadores e Internet
- Sistemas Operativos
- HTML y páginas Web
- Introducción a Java. Manejo del entorno de desarrollo
- Entradas y salidas básicas: pantalla y teclado
- Tipos de datos y operadores
- Algorítmica y sentencias de control
- Clases y métodos
- Excepciones y paquetes
- Herencia y agregación.

### 2.3. Contenido detallado

Presentación de la asignatura.

Explicación de la Guía Docente.

#### 1. **Arquitectura básica de ordenadores.**

Sistemas operativos, componentes de un PC, periféricos.

Sistemas de almacenamiento, tarjetas gráficas.

Entornos gráficos de programación para arte generativo e interactivo

#### 2. **Programación con Processing.**

Entorno de desarrollo. Estructura de un programa. Setup() y Draw()

Variables, tipos de datos, operadores.

Manipulación de vectores y arrays.

Estructuras de control, bucles. If/then, for, do/while.

Funciones: variables, retorno.

Programación orientada a objetos. Clases: variables, constructores, métodos.

#### 3. **Creación de formas 2D y 3D**

Primitivas 2D: arcos, elipses, líneas, puntos, rectángulos, triángulos.

Curvas: Bezier, puntos de control, continuidad en tangencia y curvatura.

Formas libres basadas en vértices.

Atributos de primitivas: grosor, tipos de uniones, tipos de extremos.

Manejo de colores de entidades

Primitivas 3D: box y esfera.

Tipos de luces: ambiental, especular, direccional.

Tipos de perspectivas y definición de cámaras.

#### **4. Transformación de primitivas.**

Rotación (x,y,z), translación, escala.

Matrix stack. Pop, push y reset.

#### **5. Interacción teclado y ratón, animaciones**

Eventos de ratón: click, drag, moved, pressed, Wheel, posición x,y

Eventos de teclado: click, released, códigos de teclado.

#### **6. Manipulación de imágenes y píxeles**

Carga de imágenes y texturas

Operaciones de píxeles (carga, actualización)

Máscaras, filtros

#### **7. Manipulación de videos**

Carga de videos, reproducción, captura.

Controles: play, stop, pause, frame size, duración, loop, framerate, speed

Filtros básicos.

#### **8. Manipulación de audio**

Carga y reproducción de ficheros de audio.

Controles de audio: play, stop, pause, volumen, panning, rate, loop

Osciladores, reverb, noise, captura de micrófono, análisis de audio.

### **2.4. Actividades Dirigidas**

Durante el curso se podrán desarrollar algunas de las actividades, prácticas, memorias o proyectos siguientes, u otras de objetivos o naturaleza similares:

Actividad Dirigida 1 (AD1): Los alumnos, de manera individual, tendrán que realizar una búsqueda de 5 artistas que utilicen herramientas digitales para la realización de sus obras. El alumno elegirá una obra de uno de estos artistas y la analizará de acuerdo a unos criterios que se determinarán.

Actividad Dirigida 2 (AD2): Los alumnos, de manera individual, tendrán que realizar una búsqueda de festivales, eventos, concursos, exposiciones a nivel local o internacional.

### 2.5. Actividades formativas

| Actividad formativa                        | Horas | Porcentaje de presencialidad |
|--|-------|------------------------------|
| Clases de Teoría y Problemas               | 45    | 100%                         |
| Tutorías                                   | 15    | 100%                         |
| Estudio individual y trabajo de asignatura | 90    | 0%                           |
| TOTAL                                      | 150   |                              |

## 3. SISTEMA DE EVALUACIÓN

### 3.1. Sistema de calificaciones

El sistema de calificaciones finales se expresará numéricamente del siguiente modo:

- 0 - 4,9 Suspenso (SS)
- 5,0 - 6,9 Aprobado (AP)
- 7,0 - 8,9 Notable (NT)
- 9,0 - 10 Sobresaliente (SB)

La mención de "matrícula de honor" podrá ser otorgada a alumnos que hayan obtenido una calificación igual o superior a 9,0.

### 3.2. Criterios de evaluación

#### Convocatoria ordinaria

| Sistemas de evaluación                  | Porcentaje |
|---|------------|
| Participación y trabajo de programación | 25%        |
| Prueba parcial                          | 15%        |
| Examen final                            | 60%        |

#### Convocatoria extraordinaria

| Sistemas de evaluación                  | Porcentaje |
|---|------------|
| Actividades académicas dirigidas        | 10%        |
| Examen final o trabajo final presencial | 90%        |

Se evaluará no solo los conocimientos sino la adquisición de competencias en su conjunto, tales como la calidad de la expresión artística y aptitud del alumno para comunicar, expresada gráficamente en sus trabajos y verbalmente en sus intervenciones y participación en clase.

### **3.3. Restricciones**

#### Trabajo de programación

Una nota inferior a 4 en el trabajo de programación supone el suspenso automático de la asignatura en la convocatoria ordinaria.

#### Calificación mínima

La ponderación tanto del examen parcial como de los conceptos de participación y trabajos de programación, solo se aplicará si el alumno obtiene al menos un 4.5 en el examen final.

#### Asistencia

El alumno que, injustificadamente, deje de asistir a más de un 25% de las clases presenciales podrá verse privado del derecho a examinarse en la convocatoria ordinaria.

#### Normas de escritura

Se prestará especial atención en los trabajos, prácticas y proyectos escritos, así como en los exámenes tanto a la presentación como al contenido, cuidando los aspectos gramaticales y ortográficos. El no cumplimiento de los mínimos aceptables puede ocasionar que se resten puntos en dicho trabajo.

### **3.4. Advertencia sobre plagio**

La Universidad Antonio de Nebrija no tolerará en ningún caso el plagio o copia. Se considerará plagio la reproducción de párrafos a partir de textos de auditoría distinta a la del estudiante (Internet, libros, artículos, trabajos de compañeros...), cuando no se cite la fuente original de la que provienen. El uso de las citas no puede ser indiscriminado. El plagio es un delito.

En caso de detectarse este tipo de prácticas, se considerará Falta Grave y se podrá aplicar la sanción prevista en el Reglamento del Alumno.

## **4. BIBLIOGRAFÍA**

#### Bibliografía básica

Reas, C. & McWilliams, Ch. (2010). Form+Code in Design, Art, and Architecture. New York: Princeton Architectural Press.

Pearson, M. (2011). Generative Art: A Practical Guide using Processing. Connecticut: Manning Publications.

Bibliografía recomendada

Bohnacker, H., Gross, B. & Laub, J. (2010). Generative Design: Visualize, Program, and Create with Processing. New York: Princeton Architectural Press.

Reas, C. & Fry, B. (2014). Processing a Programming Handbook for Visual Designers. MIT Press.

Shiffman, D. (2012) The Nature of Code: Simulating Natural Systems with Processing. Creative Commons A-NC 3.0 (Descargable en <http://natureofcode.com/book/>)

Otros recursos

Processing. Video tutoriales gratuitos:

<http://learningprocessing.com/videos/>

<https://www.youtube.com/playlist?list=PL374E5107CB62B2BE>

Processing. Ejemplos:

<http://learningprocessing.com/examples/>

WebGL. Tutoriales:

[https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API)