



# Fundamentos de Informática

Grado en Diseño de  
interiores  
2018-19



UNIVERSIDAD  
NEBRIJA

## GUÍA DOCENTE

**Asignatura:** Fundamentos de Informática

**Titulación:** Grado en Diseño de Interiores

**Curso Académico:** 2018-19

**Carácter:** Básica

**Idioma:** Castellano.

**Modalidad:** Presencial

**Créditos:** 6

**Curso:** 2º

**Semestre:** 1º

**Profesores/Equipo Docente:** Dr. D. Javier Sánchez Sierra

### 1. COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

#### 1.1. Competencias

Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

Aprender de manera autónoma en el campo artístico.

Trabajar e interactuar en grupos multidisciplinares aportando los conocimientos propios del área de las Bellas Artes.

Gestionar el tiempo disponible para el estudio y el desarrollo de proyectos artísticos con iniciativa y perseverancia.

Aplicar metodologías heurísticas en la resolución de problemas relacionados con proyectos artísticos.

Desarrollar estrategias de autoaprendizaje de las tecnologías relevantes para cada proyecto artístico.

Desarrollar vías de relación e intercambio con otros campos de conocimiento.

#### 1.2. Resultados de aprendizaje

Que los estudiantes hayan demostrado poseer y comprender los conceptos y métodos de métodos algebraicos; espacios vectoriales; aplicaciones lineales; endomorfismos y su clasificación; espacios afines

y variedades lineales; y espacio euclídeo, indicados en el apartado anterior, aplicando esos conocimientos a la resolución de problemas de ejemplos reales de Ingeniería.

Que los estudiantes tengan la capacidad de modelizar en términos de matrices y sistemas lineales problemas geométricos en el espacio vectorial, afín o euclídeo, siendo capaces de conectar conceptos algebraicos y geométricos. Y trabajar en espacios métricos y clasificar cónicas y cuádricas

Que los estudiantes puedan expresarse y comunicarse con rigor matemático, utilizando con soltura y propiedad los conceptos e ideas adquiridos en esta materia, así como comprender y desarrollar razonamientos matemáticos.

## 2. CONTENIDOS

### 2.1. Requisitos previos

Ninguno.

### 2.2. Descripción de los contenidos

Arquitectura de ordenadores  
Redes de ordenadores e Internet  
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): Media artists. 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): Media art festivals. Los alumnos, de manera individual, tendrán que realizar una búsqueda de festivales, eventos, concursos, exposiciones a nivel local e internacional que tengan que ver con *Media art*.

En ambas actividades, el alumno ha de realizar una presentación de la actividad al resto de sus compañeros.

**2.5 actividades Formativas**

Clases de teoría y prácticas: (0 ECTS, 45h, 100% de presencialidad). Apoyándose en transparencias el profesor explica los conceptos y plantea los ejemplos y problemas. El material presentado en las transparencias de clase no constituirá la totalidad del contenido. El alumno deberá completar sus conocimientos con referencias bibliográficas, enlaces por Internet, tutoriales y ayuda del entorno integrado de desarrollo. El profesor propondrá a los alumnos el

desarrollo de un programa de ordenador, iniciado y desarrollado parcialmente en las propias sesiones presenciales en aula informática y completado por el alumno en su trabajo individual.

Tutorías: (0.6 ECTS, 15h, 100% de presencialidad) Consulta al profesor por parte de los alumnos sobre la materia en los horarios de tutorías o empleando mecanismos de tutoría telemática (correo electrónico y uso del campus virtual de la Universidad).

Estudio individual y trabajo de asignatura: (3,6 ECTS, 90h, 0% de presencialidad) Estudio individual del alumno utilizando los apuntes y programas explicados en clase, libros de la biblioteca, o apuntes del profesor o módulos de software disponibles en el campus virtual. Para facilitar el aprendizaje y la realización de las prácticas de programación, el alumno puede acceder, en un horario amplio, a las aulas informáticas donde dispondrá de los entornos de desarrollo y ordenadores con todos los programas informáticos de la asignatura. Los alumnos completarán el trabajo de prácticas de programación propuesto en clase, pudiendo utilizar o bien sus propios ordenadores personales o bien utilizando las aulas informáticas del campus donde dispondrá de los entornos de desarrollo y ordenadores con todos los programas informáticos de la asignatura. Deben entregar el programa final al profesor para ser evaluado.

### 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

Modalidad: Presencial

Sistemas de evaluación	Porcentaje
Actividades académicas dirigidas	25%
Prueba parcial (escrita/presentación trabajo)	15%
Examen final o trabajo final presencial	60%

##### Convocatoria extraordinaria

Modalidad: Presencial

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

### 3.3. Restricciones

#### Calificación mínima

Para poder hacer media con las ponderaciones anteriores es necesario obtener al menos una calificación de 5 en la prueba 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. Vídeo 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)

## 5. DATOS DEL PROFESOR

Nombre y Apellidos	Javier Sánchez Sierra
Departamento	Arte

Titulación académica	Doctor en Ingeniería Industrial
Correo electrónico	jsanchezzierra@gmail.com
Localización	Campus de Dehesa de la Villa. Sala de profesores
Tutoría	Contactar con el profesor previa petición de hora por e-mail
<p>Experiencia docente, investigadora y/o profesional, así como investigación del profesor aplicada a la asignatura, y/o proyectos profesionales de aplicación.</p>	<p>Doctor Ingeniero Industrial en Tecnun, Universidad de Navarra. Profesor asistente de asignaturas de Expresión Gráfica y Diseño asistido por Ordenador (CAD/CAM/CAE, modelado 3D sólidos y superficies, Pro Engineer/Rhino 3D ) y fundador del laboratorio de prototipado rápido de Tecnun.</p> <p>He sido profesor visitante de la Universidad de Washington, Seattle, a cargo de proyectos de diseño compartidos entre estudiantes de USA y España.</p> <p>He realizado mi tesis doctoral en el campo de la arquitectura textil, realizando software para el diseño de superficies tensadas, campo en el que he realizado numerosas publicaciones. Me especializo posteriormente en Dessau (Alemania) donde obtengo el título de Archineer en el campo de Estructuras Tensadas, impartido por los profesionales mas influyentes del sector.</p> <p>De 2008 a 2001 realizo una estancia post-doctoral en la Universidad de Stanfor, en el CCRMA (Center for Computer Research in Music and Acoustics). Participo en numerosos proyectos multidisciplinares con artistas, principalmente en temas de visualización musical, utilizando numerosas tecnologías: max/msp, pd, processing, programación de dispositivos móviles (iOS/Android), dispositivos hápticos, programación de arduino, etc.</p> <p>Imparto asignaturas de programación de dispositivos móviles (iOS/Android) como profesor asociado en varias universidades de Valencia: Universidad Politécnica de Valencia, ESAT (Escuela Superior de Arte y Tecnología) y Berklee College of Music.</p> <p>Cuanto con una amplia experiencia en el desarrollo de aplicaciones móviles para iOS. He trabajado como desarrollador freelance en más de 40 Apps.</p> <p>Desde 2011 he participado como cofundador en varias startups: Xculpture, DressApp, <a href="#">Lingualia</a>, <a href="#">Luraki</a>, <a href="#">Flits</a>. En la última de ellas, con sede en Hamburgo y enfocada en el sector musical, estoy a cargo de toda la parte tecnológica (CTO), que permite que los usuarios puedan realizar mezclas de streaming de video en tiempo real desde sus dispositivos móviles.</p> <p>He colaborado con artistas en desarrollos concretos (programación gráfica, processing, max/msp) de proyectos visuales:  + BIOMA de Solimán López, <a href="http://bioma.solimanlopez.com">http://bioma.solimanlopez.com</a>  + Jarek Kapuscinski, Univ. Stanford, USA <a href="http://jaroslawkapuscinski.com">http://jaroslawkapuscinski.com</a></p>

