



GUÍA DOCENTE DE SEMINARIO-TALLER DE SOFTWARE

La presente guía docente corresponde a la asignatura Seminario-Taller de Software (STI-S) del Grado en Ingeniería Informática, aprobada para el curso lectivo 2017-2018 en Junta de Centro y publicada en su versión definitiva en la página web de la Escuela Politécnica Superior. La guía docente de STI-S aprobada y publicada antes del periodo de matrícula tiene el carácter de contrato con el estudiante.

ASIGNATURA

SEMINARIO-TALLER DE SOFTWARE (STI-S)

1.1. Código

17818 de la titulación de Grado en Ingeniería Informática

1.2. Materia

Seminario-Taller de Informática

1.3. Tipo

Formación obligatoria

1.4. Nivel

Grado

1.5. Curso

1º

1.6. Semestre

1º



1.7. Número de créditos

3 créditos ECTS (1,25 presenciales + 1,75 no presenciales)

1.8. Requisitos previos

CONOCIMIENTOS Y HABILIDADES PREVIOS QUE PERMITEN ENCARAR EL APRENDIZAJE DE LA ASIGNATURA

Es requisito para la correcta adquisición de los contenidos, habilidades y destrezas de la asignatura tener buena capacidad de: lectura comprensiva de textos en español e inglés, búsqueda e identificación de bibliografía y redacción de memorias en la lengua española. Asimismo, se requiere una buena disposición tanto para el aprendizaje y desarrollo de la disciplina personal de estudio como para el trabajo en equipo.

RECOMENDACIONES

Para superar con éxito la asignatura Seminario-Taller de Software es muy recomendable estar cursando la asignatura Programación I, donde se adquiere un enfoque algorítmico para la resolución de problemas y conocimientos de C.

Además, se considera necesaria la asistencia continua a las sesiones de prácticas, de grupo, de resolución de problemas, de manejo de herramientas y de construcción de sistemas de software.

Por último, es recomendable disponer de un nivel de inglés que permita al estudiante leer bibliografía de consulta.

CORRELATIVAS

Las asignaturas Seminario-Taller de Software y Seminario-Taller de Hardware conforman la materia Seminario-Taller de Informática de la titulación de Grado en Ingeniería Informática.

La asignatura Seminario-Taller de Software está transversalmente relacionada con la asignatura Programación I, porque complementa los conocimientos, habilidades y actitudes adquiridos en dicha asignatura. STI-S sirve de base para los conocimientos, habilidades y actitudes que se adquirirán en las asignaturas de Programación II y Proyecto de Programación del segundo cuatrimestre. Este conjunto de asignaturas confluyen en resultados de aprendizaje, competencias y contenidos.

1.9. Requisitos mínimos de asistencia a las sesiones presenciales

Se plantean dos métodos de evaluación: evaluación continua (con asistencia obligatoria a clase) y evaluación no continua (sin asistencia obligatoria a clase). Los estudiantes deberán optar por uno u otro método a principio del curso (dos primeras semanas) y cumplir con los distintos requisitos de



evaluación que conlleva cada uno de los modelos, publicados en la presente guía docente (véase apartado 4).

EVALUACIÓN CONTINUA CON ASISTENCIA OBLIGATORIA A CLASE

La asistencia es obligatoria al menos en un 85%.

EVALUACIÓN NO CONTINUA SIN ASISTENCIA OBLIGATORIA A CLASE

La asistencia es obligatoria al menos en un 30%. Se debe estar presente en las sesiones de entrega de cada práctica para su evaluación correspondiente sobre la práctica realizada.

1.10. Datos del equipo docente

Dr. [David Domínguez Carreta](#) (Coordinador)
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Module: B-425
Teléfono: (+34) 91 497 2245
Email: david.dominguez@uam.es
Web Page: <http://arantxa.ii.uam.es/~domingue/>

Maryam Sabzevari
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-402
Teléfono: (+34) 91 497 2349
Email: maryam.sabzevari@uam.es

Dr. Ana Gonzalez Marcos
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Office - Module: B-332
Telefono: (+34) 91 497 2234
Email: ana.marcos@uam.es

Horario de atención al alumnado: Petición de cita previa por correo electrónico.



1.11. Objetivos del curso

Las **competencias comunes a la rama de Informática** que el estudiante adquiere con la asignatura Seminario-Taller de Software son:

C13. Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los sistemas de información, incluidos los basados en web.

Los **resultados del aprendizaje** que el estudiante adquiere con la asignatura Seminario-Taller de Software son:

1. Conocimiento a nivel de usuario (avanzado) y administración (intermedio) de distintos sistemas operativos.
2. Conocimiento de flujo de control en programación.
3. Capacidad para utilizar de manera eficaz y fluida entornos de programación, incluyendo herramientas de edición, compilación, ensamblaje y depurado de código.
4. Capacidad para el diseño y realización de pruebas para comprobar la corrección, eficiencia, usabilidad y fiabilidad de un programa.
5. Buenos usos de programación.
6. Manejo de algunas aplicaciones informáticas especializadas.

A continuación, se especifican los objetivos generales y específicos de la asignatura Seminario-Taller de Software que pretenden alcanzarse.

G1 - Caracterizar la disciplina informática.

G2 - Describir la estructura y funcionamiento de las computadoras considerando sus dos componentes principales: hardware y software.

G3- Identificar, describir y relacionar las funciones que realiza un sistema operativo.

G4- Manejar el entorno Linux.

G5- Diseñar, codificar, depurar, probar, ejecutar e interpretar programas en el lenguaje de programación procedimental C.



1.12. Contenidos del programa

PROGRAMA SINTÉTICO

UNIDAD 1. INTRODUCCIÓN A LA INFORMÁTICA.

UNIDAD 2. MANEJO BÁSICO DE HERRAMIENTAS DE PROGRAMACIÓN.

UNIDAD 3. PROGRAMACIÓN BÁSICA EN ENTORNOS IDE.

UNIDAD 4. MANEJO AVANZADO DE HERRAMIENTAS DE PROGRAMACIÓN.

PROGRAMA DETALLADO

1. INTRODUCCIÓN A LA INFORMÁTICA

- 1.1. Conceptos básicos
- 1.2. Estructura y funcionamiento de los ordenadores
- 1.3 Software de un ordenador

2. CONCEPTOS BÁSICOS DE SSOO

- 2.1. Definición de sistema operativo (SO)
- 2.2. Sistemas operativos de mayor difusión (Windows, UNIX y Linux)
- 2.3. Tareas de un SO: gestión de procesador, memoria y E/S de datos.
- 2.4 Entorno UNIX y Linux

3. PROGRAMACIÓN BÁSICA

- 3.1 Pseudocódigo y diseño de diagrama de flujo.
- 3.2. Introducción a los entornos de desarrollo integrados (IDE)
- 3.3. Programación de aplicaciones en un entorno Linux.

4. MANEJO AVANZADO DE HERRAMIENTAS DE PROGRAMACIÓN

- 4.1 Programar, ejecutar, interpretar y probar programas en el lenguaje de programación C en un entorno Linux.
- 4.2 Estructuración de programas y programación modular
- 4.3 Validación, depuración y ejecución de aplicaciones en Linux.



1.13. Referencias de consulta

Los recursos para el aprendizaje por unidad se detallan a continuación. Se distinguen entre lecturas básicas y lecturas recomendadas.

UNIDAD 1: INTRODUCCIÓN A LA INFORMÁTICA

Bibliografía Básica:

7. A. PRIETO, A. LLORIS, J. C. TORRES. *Introducción a la Informática*. 4ª ed. (McGraw Hill, 2006). Cap. 1: Introducción.

Bibliografía Recomendada:

8. J. G. BROOKSHEAR. *Computer Science: An Overview*. 9ª ed. (Pearson International Edition, 2007).

UNIDAD 2: CONCEPTOS BÁSICOS DE SSOO

Bibliografía Básica:

9. E. ANGUIANO, D. CAMACHO, C. NAVARRETE. *LINUX, Guía de Aprendizaje*. Prentice Hall, 2008

Bibliografía Recomendada:

10. J. GARCÍA de JALÓN, I. AGUINAGA, A. MORA. *Aprenda LINUX como Si Estuviera en Primero*. (Universidad de Navarra, 2000).
11. B. W. KERNIGHAN. *El Entorno de Programación UNIX*. (Prentice Hall Hispanoamericana, 1987).

UNIDAD 3: PROGRAMACIÓN BÁSICA

Bibliografía Básica:

12. B. W. KERNIGHAN, D. RITCHIE, *The C Programming Language*. 2ª ed. (Prentice Hall, 1988).

UNIDAD 4: MANEJO AVANZADO DE HERRAMIENTAS DE PROGRAMACIÓN

Bibliografía Básica:

13. B. W. KERNIGHAN. *La Práctica de la Programación*. (Pearson Educación, 2000).

Bibliografía Recomendadas:

14. H. SCHILDT. *C: Guía de Autoenseñanza*. (Osborne/McGraw Hill, 2001).
15. A. SIERRA URRECHO. *Programación en C/C++*. (Anaya Multimedia, 2005).

Nota: Esta asignatura no sigue un libro en concreto. No se recomienda a los estudiantes comprar ningún libro hasta no haber comparado su contenido con el programa y revisado previamente en la biblioteca.

Material electrónico de trabajo: los documentos electrónicos de trabajo se publican en la sección de STI-S en plataforma Moodle (<https://moodle.uam.es>)



2. Métodos docentes

En la asignatura Seminario-Taller de Software es eminentemente práctica. Todas las clases tendrán un fuerte contenido práctico, donde el profesor describirá de forma breve los principales contenidos teóricos que sean necesarios para el desarrollo de los módulos del curso. Para alcanzar los objetivos y capacidades previstos, la asignatura está organizada en sesiones prácticas en laboratorios con ordenadores, donde gran parte del trabajo se organiza en clases de tipo participativo en grupos pequeños.

SESIONES PRÁCTICAS DE LABORATORIO

Actividad del profesor:

Al comienzo de la sesión, el profesor asignará a cada equipo el trabajo que se debe realizar en el laboratorio. Supervisará el trabajo de los equipos durante la sesión. Se contemplan distintos tipos de actividades a realizar en las clases prácticas:

- Explicar y suministrar el guion de prácticas a completar fuera del laboratorio.
- Lectura y comprensión del material docente.
- Análisis y desarrollo de casos prácticos.
- Ayuda para el aprendizaje de herramientas, incluyendo tutoriales.
- Supervisar el trabajo de los grupos de trabajo en el laboratorio.

Los medios utilizados son el software del laboratorio y ordenadores del propio laboratorio para la ejecución y simulación de los programas realizados, así como el material docente publicado en línea.

Actividad del estudiante:

Actividad presencial: Dependiendo de la fase del desarrollo de la sesión en cuestión, los estudiantes deberán: i) comprender las explicaciones sobre las técnicas a emplear, ii) trabajar en equipo para aplicar dichas técnicas al desarrollo del proyecto, iii) redactar informes sobre los resultados obtenidos. En algunas sesiones, se requerirá la ejecución con el profesor presente, quien hará las preguntas oportunas a cada miembro del grupo para calificar de forma individual la práctica.

Actividad no presencial: Reuniones en equipo para finalizar el trabajo correspondiente. Redacción del informe de la práctica



TUTORÍAS EN EL AULA

Actividad del profesor:

Las tutorías se organizan bien para (a) realizar un seguimiento de los grupos más personalizado y resolver conflictos entre los integrantes; bien para (b) resolver dudas comunes planteadas por los alumnos a nivel individual o en grupo, surgidas a partir de las prácticas o el proyecto.

Actividad del estudiante:

Actividad presencial: Planteamiento de dudas individuales o en grupo y enfoque de posibles soluciones a las tareas planteadas.

REUNIONES DE EQUIPO

Actividad del estudiante:

Actividad no presencial: reuniones presenciales y en línea entre los distintos miembros del equipo para integrar los resultados parciales realizados por cada miembro.

LECTURAS OBLIGATORIAS Y ESTUDIO PERSONAL

Actividad del estudiante:

Actividad no presencial: Aprendizaje autónomo académicamente dirigido por el profesor a través de las tareas publicadas en la página de docencia en red.

3. Tiempo de trabajo del estudiante

| | | NºHoras | % |
|--------------------------------------------------|---------------------------------------------|-------------|-------------------|
| Sala de aula | Clases teóricas | 0 h (0%) | 45% = 34 horas |
| | Clases prácticas | 28 h (37%) | |
| | Tutorías durante el semestre | 3 h (4%) | |
| | Exámenes | 3 h (4%) | |
| Fuera de Clase | Actividade practicas | 18 h (24%) | 55% = 41 horas |
| | Estudio semanal (0,5 h x 14 weeks) | 7 h (10%) | |
| | Exam preparation | 16 h (21%) | |
| Total horas de trabajo: 25 horas x 3 ECTS | | 75 h | |



4. Métodos de evaluación y porcentaje en la calificación final

CRITERIOS DE EVALUACIÓN

El proyecto se evaluará en 4 prácticas, en los que el estudiante deberá entregar los entregables requeridos, o realizar las pruebas correspondientes. La Nota Final de **evaluación continua (NC)** de la asignatura se obtiene mediante la siguiente fórmula ponderada de las cuatro prácticas:

$$NC = (P1 + P2 + P3 + P4)/4$$

Donde **Pi** corresponde a la nota de la practica **i**. En caso de obtener una nota **NC ≥ 5** el alumno estará aprobado. En caso contrario, el alumno debe presentarse al Examen Final (**EF**) que será realizado en el mes de enero, e involucra todos los temas desarrollados en dichas prácticas, junto a los estudiantes que han optado por la **evaluación no continua**. En este caso la nota final será calculada mediante la fórmula:

$$NF = 50\% EF + 50\% NC$$

Dónde **NC** es el promedio de las prácticas definido arriba, y **EF** la nota del examen. Se considera que el estudiante se ha presentado a la convocatoria correspondiente si presenta al menos dos prácticas. Por debajo de este número el estudiante recibirá la calificación "No evaluado".

En relación con la **evaluación extraordinaria** para ambos métodos, los estudiantes deberán presentar una práctica de recuperación y presentarse al examen final de prácticas a desarrollarse en el mes de junio.

Nota: Seminario-Taller de Informática (**STI**) se aprueba con una nota mínima de **5**. La nota de Seminario-Taller de Informática se obtendrá mediante la media de las notas obtenidas en Seminario-Taller de Software (STI-S) y Seminario-Taller de Hardware (STI-H). Es necesario obtener una nota mínima de 5 en cada una de las asignaturas, STI-S y STI-H, para poder hacer media. En caso contrario, la nota final de STI será igual a: $50\% \times \text{Mínimo (5, STI-S)} + 50\% \times \text{Mínimo (5, STI-H)}$.



NORMAS

Los entregables se regirán por las siguientes normas:

1. Se propondrán cuatro prácticas a lo largo del curso, de entrega y/o evaluación obligatoria para superar la asignatura.
2. Las prácticas se evaluarán en una escala del 0 al 10.
3. Para aprobar las prácticas será necesario:
 - 3.1. Realizar la evaluación de rendimiento individual de la Práctica
 - 3.2. Entregar el código fuente correspondiente a las Prácticas 3 y 4.
La misma se evaluará también con respecto a:
 - Corrección funcional: respuesta correcta a datos de entrada, ya sean normales, excepcionales o incorrectos.
 - Organización del código y calidad del mismo.
 - Entregar una memoria, según se especifica después en este documento.
4. En el enunciado de las prácticas en la Web estarán publicadas las fechas de entrega de las mismas.
5. Toda práctica entregada con el requisito de desarrollo de software que no compile o con errores de ejecución se penalizará un 50% en la nota final de prácticas.
6. Una práctica no entregada supone un suspenso en las prácticas.
7. Por cada día de retraso en la entrega se penalizará un 10% de la nota final de prácticas.
8. La composición de los equipos se establecerá el primer día de prácticas y después no podrá modificarse, salvo circunstancias excepcionales.
9. Las notas asignadas a los integrantes de un mismo equipo por una misma práctica podrán ser diferentes cuando el profesor de prácticas encuentre justificación para ello.

ATENCIÓN: Cualquier copia descubierta que se haya realizado a lo largo del curso, tanto en cualquiera de las actividades de teoría desarrolladas, como en el proyecto, serán penalizadas con rigurosidad. La penalización por copia implica la aplicación de la normativa interna de la EPS, que supone suspender la convocatoria actual y no poder presentarse a la siguiente.

MEMORIA

La memoria cuando esta sea requerida deberá incluir:

1. Nombre de los estudiantes, número y nombre del equipo, turno, fecha y número de práctica.
2. Descripción del problema o práctica en cuestión.
3. Código, o análisis para la resolución del problema planteado.
4. Aspectos de interés en la realización de la práctica a tener en cuenta por el profesor.
5. Discusión y razonamiento de los resultados obtenidos.



ENTREGAS

La entrega consistirá en un fichero comprimido (.zip), en el que se incluirá, para cada práctica realizada, lo siguiente:

- Código fuente, debidamente comentado, de todos los programas realizados en la práctica (ficheros .c).
- Ficheros de cabecera (ficheros .h).
- Fichero Makefile.
- Ficheros de datos necesarios para la ejecución del programa, si los hay, bien como un manual de utilización del programa.
- Memoria de la práctica.

No se entregarán, por tanto, los ficheros ejecutables, con el código objeto (*.o), ni los ficheros de los proyectos y los entornos de trabajo. Dentro del fichero comprimido, cada fichero tendrá un nombre significativo que indique su función. Las extensiones dadas a los nombres de los ficheros serán las usuales: c para los ficheros fuentes, h para los ficheros de cabecera, etc.

En el caso de entrega electrónica de prácticas, el nombre del fichero de entrega debe ser *PxGyEz.zip* donde *x* es el número de práctica, y el grupo correspondiente y *z* el número de equipo dentro del grupo. Por ejemplo, si el equipo 6 del grupo 64 quiere entregar la Práctica 3 debería enviar un fichero con el nombre *P3G64E6.zip*.

Se entregarán las prácticas desde el Sistema de Entrega de Prácticas de la plataforma Moodle (<http://uam-virtual.es>) de la Escuela Politécnica Superior donde se estructura STI-S.



5. Cronograma

| Semana | Contenido | Horas presencial | Horas no presenciales |
|--------|------------------------------------------------------------------------------------------------------------------------|------------------|----------------------------------|
| 1 | - Presentación y motivación de la asignatura, su normativa y formación de grupos para la realización de las prácticas. | 2 | - |
| 2 | - Unidad 1 Introducción a la Informática | 2 | 2: Lectura de Unidad 1 |
| 3+4 | - Sesión de laboratorio para realizar la Práctica 1. | 2 | 2: Realización de Práctica 1 |
| 5 | - Unidad 2 Manejo básico de herramientas de programación | 2 | 2: Lectura de Unidad 2 |
| 6+7 | - Sesión de laboratorio para realizar la Práctica 2. | 4 | 4: Realización de Práctica 2 |
| 8 | - Unidad 3 Programación básica | 2 | 2; Lectura de Unidad 3 |
| 9 | - Introducción a la Práctica 3. | 2 | 2: Realización de Práctica 3 |
| 10+11 | - Sesión de laboratorio para realizar la Práctica 3. | 4 | 4: Finalización de la Práctica 3 |
| 12 | - Unidad 4 Manejo avanzado de herramientas de programación | 2 | 2: Lectura de Unidad 4 |
| 13 | - Introducción a la Práctica 4. | 2 | 2: Realización de Práctica 4 |
| 14+15 | - Sesión de laboratorio para realizar la Práctica 4. | 04:00:00 | 4: Finalización de Práctica 4 |