

DATOS DE LA ASIGNATURA*

* Asignatura en experiencia piloto de implantación del sistema de créditos ECTS

Nombre:			
Metodología de la Programación I			
Denominación en inglés¹:			
Programming Methodology			
Código:	Año del Plan de Estudios:	Tipo:	
460004042	Publicación BOE: 27-07-2004	<input checked="" type="checkbox"/> Troncal <input type="checkbox"/> Obligatoria <input type="checkbox"/> Optativa	
Créditos:			
	Totales:	Teóricos:	Prácticos:
Créditos L.R.U.	9,00	4,50	4,50
Créditos E.C.T.S.	7,2	3,6	3,6
Departamento:			
Tecnologías de la Información			
Área de Conocimiento:			
Lenguaje y Sistemas Informáticos			
Curso:	Cuatrimestre:	Ciclo:	
Primero	1º Cuatrimestre	Primero	
Web de la asignatura:			
Plataforma de docencia virtual de la UHU			

¹Para su inclusión en el Complemento Europeo al Título

DATOS DE LOS PROFESORES

Nombre:	e-mail:	Teléfono:	Despacho:
Jose Manuel Martín Ramos	jmmartin@dti.uhu.es	959217637	46
Lourdes Ortiz Sañudo	lourdes@dti.uhu.es	959217391	47
Francisco Roche Beltrán	roche@dti.uhu.es	959217654	22
Antonio Suarez Fábregas	asuarez@dti.uhu.es	959217677	39

DATOS ESPECÍFICOS DE LA ASIGNATURA

1.1. Descriptores de la asignatura:

Diseño de algoritmos. Análisis de algoritmos. Lenguajes de programación.
Diseño de programas: descomposición modular y documentación.

1.2. Descriptores de la asignatura (en inglés)²:

Texto a rellenar por el profesor

²Para su inclusión en el Complemento Europeo al Título

2. Situación de la asignatura.

2.1. Prerrequisitos:

No tiene

2.2. Contexto dentro de la titulación:

La asignatura de Metodología de la Programación I es la primera asignatura en la que el alumno aprecia el papel central que representa la abstracción en la tarea de programar, a la vez que conoce un lenguaje de programación de alto nivel para codificar los programas, en el paradigma de la programación orientada a objetos. Es por tanto, imprescindible tanto en asignaturas específicas de programación como en aquellas en las que se hace uso de un lenguaje de programación:

- Estructuras de Datos I
- Estructuras de Datos II
- Metodología de la Programación II
- Programación en Internet
- Ingeniería del Software
- Sistemas Operativos
- Ampliación de Sistemas Operativos
- Programación Concurrente.

2.3. Recomendaciones:

El alumno debe estudiar la asignatura consultando la bibliografía sugerida por los profesores y asistir con regularidad a las tutorías que estos ofertan. Debido al marcado carácter práctico de la asignatura, se recomienda que el alumnado realice la mayor parte de los ejercicios propuestos de las relaciones de problemas.

Los alumnos que cursen esta asignatura deberían :

- Saber leer, comprender y expresarse correctamente tanto de forma oral como escrita.
- Tener conocimientos matemáticos a nivel de Bachillerato.
- Conocer el idioma inglés tanto a nivel escrito como oral (nivel medio).

3. Competencias a adquirir por los estudiantes.

3.1. Competencias transversales o genéricas.

3.1.1. Competencias instrumentales:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de análisis y síntesis.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de organización y planificación.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Comunicación oral y escrita en lengua nativa.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimiento de una lengua extranjera.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de gestión de la información.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Resolución de problemas.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Toma de decisiones.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: conocimientos de Informática.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.2. Competencias personales:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en equipo.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en un equipo de carácter interdisciplinar.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en un contexto internacional.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidades en las relaciones interpersonales.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Capacidad para comunicarse con expertos de otras áreas.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Reconocimiento a la diversidad y la multiculturalidad.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Razonamiento crítico.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Compromiso ético.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Desarrollar actitudes y hábitos de confianza.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.3. Competencias sistémicas:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Aprendizaje autónomo.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Adaptación a nuevas situaciones.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de aplicar los conocimientos en la práctica.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidad para trabajar de forma autónoma.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Creatividad.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Liderazgo.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimiento de otras culturas y costumbres.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Iniciativa y espíritu emprendedor.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Motivación por la calidad.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Sensibilidad hacia temas medioambientales.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.2. Competencias específicas.

3.2.1. Competencias cognitivas (saber):

Conocimientos de Informática.
Dominio de nuevas tecnologías TICs.
Proveer al alumno de conocimientos generales sobre programación.
Enseñar a utilizar lenguajes de programación concretos de amplio uso, a transcribir a este lenguaje y ejecutar en una máquina real sus propios algoritmos.

3.2.2. Competencias procedimentales e instrumentales (saber hacer):

Diseño y redacción de documentación técnica.
Planificación y organización.
Enseñar al alumno las técnicas básicas para la resolución de problemas mediante programas de ordenador.
Promover la idea de que la construcción de programas es una metodología en la cual se deben seguir procesos sistemáticos para alcanzar el objetivo con una calidad adecuada.

3.2.2. Competencias actitudinales (ser):

Aprendizaje autónomo.
Creatividad.
Acostumbrar al alumno a programar respetando unas normas y directrices genéricas.
Dotar de capacidad analítica para enfrentarse a problemas reales y para saber elegir y aplicar las técnicas adecuadas para construir algoritmos que los resuelvan de forma eficaz y eficiente.

4. Objetivos:	
<ul style="list-style-type: none"> • Desarrollar una aproximación disciplinada a la especificación, implementación, verificación y documentación de programas. • Apreciar el papel central que representa la abstracción en la tarea de programar • Desarrollar en el alumno la capacidad de resolución de problemas mediante las técnicas de diseño de algoritmos y aplicarlo a la codificación de programas • Conocer y utilizar adecuadamente estructuras de datos básicas, algoritmos y esquemas de uso general • Aprender un lenguaje de programación de alto nivel estructurado, general y extendido que use el paradigma de la programación orientado a objetos. • Proporcionar los fundamentos teóricos y prácticos básicos para cursar posteriores estudios en programación. 	

5. Metodología (en horas de trabajo del estudiante):		
	Primer Cuatrimestre	Segundo Cuatrimestre
	Presenciales	
Clases de teoría	31,5	0,0
Clases de problemas	13,5	0,0
Clases prácticas	31,5	0,0
Actividades académicas dirigidas	6,5	0,0
	Exámenes	
	8,0	0,0
	No presenciales	
Estudio de clases teóricas (factor de trabajo: 1,20)	37,8	0,0
Estudio de clases de problemas y prácticas (factor de trabajo: 1,30)	58,5	0,0
Preparación de actividades académicamente dirigidas y otras actividades	4,9	0,0
Total:	192,2	0,0
Trabajo total del estudiante: 192,2 horas.		
Horas presenciales:	83,0	Horas no presenciales:
		101,2
	Exámenes:	8,0

6. Técnicas docentes.	
6.1. Técnicas docentes utilizadas:	
<input checked="" type="checkbox"/> Sesiones académicas de teoría <input checked="" type="checkbox"/> Sesiones académicas de problemas <input checked="" type="checkbox"/> Sesiones prácticas en laboratorio <input checked="" type="checkbox"/> Seminarios, exposiciones y debates <input type="checkbox"/> Trabajo en grupos reducidos <input checked="" type="checkbox"/> Resolución y entrega de problemas/prácticas <input checked="" type="checkbox"/> Realización de pruebas parciales evaluables <input checked="" type="checkbox"/> Otras: Tutorías especializadas <input type="checkbox"/> Otras: Especificar	
6.2. Desarrollo y justificación:	
Sesiones académicas de teoría	
<ul style="list-style-type: none"> • Consisten en clases magistrales donde se impartirá la base teórica de la asignatura y se expondrán ejemplos aclaratorios de la misma al grupo. Las sesiones teóricas tendrán una duración de 1,5h cada una, sumando un total de 31,5h y se irán intercalando con las sesiones de problemas a lo largo del curso, de manera que una vez finalizado un tema teórico con sus correspondientes sesiones 	

académicas de teoría, se impartirán sesiones de problemas.

- La metodología usada para impartir la teoría y los ejemplos aclaratorios será la exposición mediante cañón, retroproyector y/o uso de pizarra. El profesor podrá solicitar la participación activa del alumno mediante preguntas rápidas, teniendo en cuenta los alumnos que más participen a la hora de evaluar.
- En la página Web de la asignatura localizada en la plataforma de docencia virtual de la UHU, se encontrarán las transparencias y otros materiales de referencia necesarios para el seguimiento de las sesiones.

Sesiones académicas de problemas / Resolución y entrega de problemas

- Consisten en la realización de problemas relacionados con los conceptos y métodos operativos de la asignatura. Las sesiones de problemas tendrán una duración de 1,5h cada una, sumando un total de 13,5h. Los grupos serán reducidos y homogéneos, constando cada sesión de problemas de dos fases. Una primera en la que el profesor explicará uno o varios problemas tipo y una segunda en la que los alumnos podrán ser evaluados por el profesor con la resolución de un problema similar a los expuestos en la fase primera.
- En la página Web de la asignatura localizada en la plataforma de docencia virtual de la UHU, se encontrará un compendio de problemas para resolver, y se irán indicando aquellos que van siendo resueltos en las sesiones. El resto de problemas deben de ser resueltos por los alumnos de forma voluntaria y las soluciones propuestas por ellos podrán ser comprobadas haciendo uso de las horas de tutorías y de atención al alumno del profesor. La asistencia a las sesiones de problemas es obligatoria.

Sesiones prácticas en laboratorio / Resolución y entrega de prácticas

- Consisten en el diseño e implementación de programas escritos en C++ utilizando orientación a objetos desde el primer día. Los alumnos dispondrán con antelación del problema a resolver y la metodología de trabajo. El trabajo se realizará de forma individual. Las sesiones prácticas tendrán una duración de 1,5h cada una, sumando un total de 31,5h. La asistencia a las sesiones de laboratorio es obligatoria.

Seminarios, exposiciones y debates

- Se impartirán con el objetivo de acercar al alumnado a la evaluación de sus competencias en el contexto de esta asignatura.

Realización de pruebas parciales evaluables

- A lo largo del curso se realizarán 2 pruebas parciales evaluables sobre las prácticas en laboratorio. La primera prueba parcial (práctica) se realizará una vez desarrolladas las dos primeras prácticas y consistirá en realizar una o varias modificaciones a los programas realizados durante esas dos primeras prácticas. De igual forma, una vez desarrollada la práctica tercera, se realizará una segunda prueba práctica, de modificación de los programas desarrollados en dicha práctica.

Tutorías especializadas

- En la página Web de la asignatura localizada en la plataforma de docencia virtual de la UHU, los profesores irán colocando a disposición de los alumnos un compendio de problemas resueltos, de los propuestos en las sesiones de resolución de problemas, y que no haya dado tiempo a resolver o que considere el profesor que son interesantes. Estos problemas, junto con otros problemas tipos, comunes en los exámenes serán expuestos y explicados por los profesores de la asignatura en una tutoría colectiva que se impartirá antes de finalizar el curso.

7. Bloques temáticos:

Bloque I: Introducción a los Lenguajes de Programación.

Tema 1. Lenguajes de Programación.

Bloque II: Algoritmos y Tipos de Datos.

Tema 2. Algoritmos. Tipos de Datos, Operadores y Expresiones.

Tema 3. Estructuras de Control.

Tema 4. Tipos de Datos Estructurados.

Bloque III: Técnicas de Diseño de Programas.

Tema 5. Diseño Descendente.

Tema 6. Diseño Modular.

8. Temario desarrollado:

TEORIA:

Unidad didáctica nº 1.- Introducción a los Lenguajes de Programación.

TEMA 1. LENGUAJES DE PROGRAMACIÓN

- 1.1. Concepto e historia de los lenguajes de programación.
- 1.2. Paradigmas de Programación.
- 1.3. Clasificación. Lenguaje de bajo, medio y alto nivel.
- 1.4. Traductores: Compiladores e Intérpretes. Proceso de generación de código ejecutable. Compilación y Enlazado.
- 1.5. Programación Orientada a Objetos.

Unidad didáctica nº 2.- Algoritmos y Tipos de Datos.

TEMA 2. ALGORITMOS. Tipos de Datos, Operadores y Expresiones

- 2.1. Concepto de algoritmo. Estructura general.
- 2.2. Palabras clave, Identificadores, constantes y comentarios. Variables y objetos.
- 2.3. Tipos de datos y Clases. Clasificación.
- 2.4. Operadores de asignación, aritméticos, relacionales y lógicos.
- 2.5. Expresiones y orden de precedencia.
 - 2.5.1. Reglas de precedencia y asociatividad.
 - 2.5.2. Conversiones de tipos en las expresiones.
 - 2.5.3. Coherencia de tipos. Moldeado de tipos.
- 2.6. Operaciones básicas de entrada-salida.

TEMA 3. ESTRUCTURAS DE CONTROL

- 3.1. Sentencia Secuenciales.
- 3.2. Sentencias Condicionales.
- 3.3. Sentencias Iterativas.
- 3.4. Macros de sentencias.

TEMA 4. TIPOS DE DATOS ESTRUCTURADOS

- 4.1. Registros y Estructuras de registros. Registros jerárquicos.
- 4.2. Vectores y Matrices. Vectores de Registros
- 4.3. Clases y Objetos.
- 4.4. Declaraciones privadas y públicas. Ámbitos de visibilidad.

Unidad didáctica nº 3.- Técnicas de diseño de programas.

TEMA 5. DISEÑO DESCENDENTE

- 5.1. Programación Estructurada.
- 5.2. Declaraciones globales y declaraciones locales. Ámbitos de visibilidad.
- 5.3. Funciones y Métodos: Interfaces.
- 5.4. Parametrización. Parámetros por valor y por referencia.
- 5.5. Constructores y destructores.
- 5.6. Paso de estructuras de datos complejos a funciones y métodos.
- 5.7. Sobrecarga de métodos y operadores.

TEMA 6. DISEÑO MODULAR

- 6.1. Programación Modular.
- 6.2. Concepto de módulo.
- 6.3. Creación y utilización de módulos
- 6.4. Importación y exportación.
- 6.5. Definición y uso de Módulos.

LABORATORIO:

Unidad didáctica nº 2.- Algoritmos y Tipos de Datos.

- Práctica 1. Introducción a C++, uso de clases.
Práctica 2. Uso de cadenas y tablas.

Unidad didáctica nº 3.- Técnicas de diseño de programas.

- Práctica 3. Diseño descendente.

9. Bibliografía.

9.1. Bibliografía general:

- METODOLOGÍA DE LA PROGRAMACIÓN I: INTRODUCCIÓN AL DISEÑO ORIENTADO A OBJETOS. A. Márquez, Lourdes Ortiz, M^a Pilar Polo, Fco. Roche y Ana M^a Roldán. Servicio de Publicaciones de la Universidad de Huelva.
- COMO PROGRAMAR EN C/C++. H.M. Deitel. Edt. PEARSON Prentice Hall.
- C++ ESTÁNDAR. E. Hernández Orallo. Edt. Paraninfo, Thomson Learning.
- PROGRAMACIÓN EN C++ PARA INGENIEROS. F. Xhafa, P. Vázquez, J. Marco, X. Molinero y A. Martín. Edt. Thomson.
- EL LENGUAJE DE PROGRAMACIÓN C++, B. Stroustrup. Ed. PEARSON Addison Wesley.
- PROGRAMACIÓN Y DISEÑO EN C++, J.P. Cohoon, J.W. Davidson. Edt. Mcgraw-Hill

9.2. Bibliografía específica:

- RESOLUCIÓN DE PROBLEMAS CON C++. W. Savitch. PEARSON Addison Wesley.
- Libros electrónicos:
<http://mindview.net/Books/TICPP/ThinkingInCPP2e.html>
http://www.dsic.upv.es/~jorallo/libro_c++/fuentes/ejercicios.html

10. Técnicas de evaluación.

10.1. Técnicas de evaluación utilizadas:

- Examen teórico-práctico
- Trabajos desarrollados durante el curso
- Participación activa en las sesiones académicas
- Controles periódicos de adquisición de conocimientos
- Examen práctico en aula de informática
- Otras: Sesiones de problemas
- Otras: Seminarios

10.2. Criterios de evaluación y calificación:

- **Examen Teórico-práctico.** Consistente en una prueba donde el alumno deberá resolver distintos problemas y/o cuestiones teóricas referentes a los temas teóricos desarrollados durante el cuatrimestre.
- **Examen práctico en aula de informática.** Una vez desarrolladas las dos primeras prácticas se realizará una evaluación sobre el contenido de las mismas. Así mismo en la última sesión de clase se realizará otra prueba consistente en la modificación de la práctica tercera que haya realizado el alumno.
- **Sesiones de problemas.** En todas las actividades formativas el profesor valorará la participación activa del alumno. Se valorará de forma directa tanto en las sesiones de problemas como en las del laboratorio.

Se resumen en esta tabla:

Aspecto	Criterio	Instrumento	Peso
Conocimientos teóricos.	Exposición escrita de conocimientos y su aplicación a distintas situaciones.	Examen Teórico Escrito	50%
Conocimientos prácticos.	Capacidad de implementar programas en C++	Examen de modificación de las prácticas 1 y 2.	10%
		Examen de modificación de práctica 3.	15%
Asistencia y Participación	Participación activa en las sesiones, aportando soluciones válidas a los problemas y asimilando los conocimientos	Sesiones de problemas	25%

La nota del alumno será:

a) Si (**Examen Teórico Escrito + Sesiones de problemas**) ≥ 3.5

(Examen Teórico Escrito + Sesiones de problemas) + Examen de modificación de las prácticas 1 y 2 + Examen de modificación de la práctica 3

b) En caso contrario, si (**Examen Teórico Escrito + Sesiones de problemas**) < 3.5

(Examen Teórico Escrito + Sesiones de problemas)

Criterio para guardar las notas para convocatorias posteriores:

Examen Teórico y Sesiones de Problemas:

- La nota del examen teórico y de las sesiones de problemas se guarda hasta la convocatoria de septiembre siempre que sumando las dos se obtenga una nota igual o superior a 3,5.
- Para la convocatoria de diciembre no se guarda ni la teoría ni los problemas.

Examen Práctico en Laboratorio:

- La nota de prácticas en laboratorio (hasta 2,5 puntos) se guarda hasta la convocatoria de septiembre, independientemente de la nota sacada por el alumno. No obstante el alumno puede presentarse al examen de prácticas de laboratorio de septiembre, pero en ese caso la nota sacada en febrero no se conserva, de forma que, si se presenta un alumno a prácticas en septiembre la nota obtenida en ese examen será la definitiva.
- Para la convocatoria de diciembre se guardarán las prácticas siempre que la nota de prácticas de laboratorio haya sido superior o igual a 1,25 puntos.

11. Organización docente semanal (en horas presenciales del alumno)**11.1. Primer cuatrimestre:**

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	3,0	0,0	0,0		0,0	0,0	Presentac. y T1
2ª	3,0	0,0	0,0		0,0	0,0	T1 y T2
3ª	3,0	0,0	1,5		0,0	0,0	T2
4ª	3,0	0,0	3,0		0,0	0,0	T3
5ª	0,0	3,0	3,0		0,0	0,0	T3
6ª	3,0	0,0	3,0		0,0	0,0	T4
7ª	0,0	3,0	3,0		0,0	0,0	T4
8ª	3,0	0,0	3,0		0,0	0,0	T5
9ª	1,5	1,5	3,0		0,0	0,0	T5
10ª	3,0	0,0	0,0		0,0	2,5	T5
11ª	3,0	0,0	3,0		0,0	0,0	T5
12ª	3,0	0,0	3,0		0,0	0,0	T5
13ª	1,5	1,5	3,0		0,0	0,0	T5
14ª	0,0	3,0	3,0	Seminario preparación evaluación de competencias	3,5	0,0	T5
15ª	1,5	1,5	0,0	Seminario preparación evaluación de competencias	3,0	2,5	T6
Periodo de exámenes						3,0	
Totales	31,5	13,5	31,5		6,5	8,0	

11.2. Segundo cuatrimestre:

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	0,0	0,0	0,0		0,0	0,0	
2ª	0,0	0,0	0,0		0,0	0,0	
3ª	0,0	0,0	0,0		0,0	0,0	
4ª	0,0	0,0	0,0		0,0	0,0	
5ª	0,0	0,0	0,0		0,0	0,0	
6ª	0,0	0,0	0,0		0,0	0,0	
7ª	0,0	0,0	0,0		0,0	0,0	
8ª	0,0	0,0	0,0		0,0	0,0	
9ª	0,0	0,0	0,0		0,0	0,0	
10ª	0,0	0,0	0,0		0,0	0,0	
11ª	0,0	0,0	0,0		0,0	0,0	
12ª	0,0	0,0	0,0		0,0	0,0	
13ª	0,0	0,0	0,0		0,0	0,0	
14ª	0,0	0,0	0,0		0,0	0,0	
15ª	0,0	0,0	0,0		0,0	0,0	
Periodo de exámenes						0,0	
Totales	0,0	0,0	0,0		0,0	0,0	

12. Mecanismos de control y seguimiento:

12. Mecanismos de control y seguimiento:

- Se propone la realización de encuestas de satisfacción entre los alumnos.
- Índices de rendimiento (tasa de éxito/fracaso, % de presentados a examen, etc.)
- Control de Asistencia