

Programación do Módulo Profesional Fundamentos de Programación

Ciclo Formativo Superior de Administración de Sistemas Informáticos Régime de Adultos

Departamento de Informática

I.E.S. de Rodeira - Cangas

Curso 2006-2007

PROFESORA

Eva Pereira Loira

4 de setembro de 2006

ÍNDICE:

1. INTRODUCCIÓN.....	3
2. ORGANIZACIÓN DOS CONTIDOS	4
2.1. Capacidades terminais elementais e criterios de avaliación asociados.....	4
2.2. Relación secuenciada das unidades de traballo.	6
2.3. Elementos curriculares de cada unidade.....	8
2.4. Descrición das Unidades de Traballo.	9
Unidade de Traballo 1. Introducción.....	9
Unidade de Traballo 2. Metodoloxía da Programación.....	10
Unidade de Traballo 3. Programación Estructurada.....	11
Unidade de Traballo 4. Desenvolvemento de Software Orientado a Obxectos.....	13
Unidade de Traballo 5. Creación de Programas.....	15
Unidade de Traballo 6. Ferramentas de Programación Visual.....	17
Unidade de Traballo 7. Desenvolvemento Web.....	19
3. CRITERIOS DE AVALIACIÓN	20
3.1. Avaliación Inicial.....	20
3.2. Avaliación Sumativa.....	20
3.3. Avaliación dos alumnos que perden o dereito á avaliación continua.....	21
3.4. Avaliación dos alumnos co módulo pendente doutros cursos (avaliación extraordinaria).....	21
4. MATERIAIS E RECURSOS DIDÁCTICOS	22
4.1. Bibliografía.....	22
4.2. "Software".....	22

1. INTRODUCCIÓN.

Este módulo encádrase dentro do Ciclo formativo correspondente ó Técnico Superior en Administración de Sistemas Informáticos e ten asignada unha duración en Horario Vespertino de Adultos de 265 horas ademais de 27 horas das 80 horas a disposición do centro, repartidas en 350 sesións lectivas de 50´.

Tratouse de enfocar este desenvolvemento curricular poñendo en situación de profesional ó suxeito que vai ser obxecto directo do proceso de ensino-aprendizaxe, planificando este proceso conforme ás tarefas que este futuro profesional terá encomendadas e estimando o nivel de coñecementos cos que principiará a aprendizaxe, así como o desenvolvemento e a ampliación de conceptos e habilidades que adquirirá no mesmo período de tempo noutros módulos deste Ciclo formativo.

Este módulo está asociado a unidade de competencia 4 que leva por título:

"Propor e coordinar cambios para mellora-la explotación de sistema e as aplicacións".

Para conseguir levar a cabo as realizacións desta unidade de competencia deseñáronse dous módulos, Fundamentos de Programación, de primeiro curso, e Desenvolvemento de Funcións no Sistema Informático, de segundo curso. De tal maneira que o primeiro suporá a adquisición dos coñecementos básicos necesarios de programación que serán completados e complementados no segundo módulo para alcanzar as metas propostas.

Debido a que o obxectivo final deste módulo non constitúe máis ca un paso intermedio na consecución das realizacións da correspondente unidade de competencia, o enunciado do contido organizador non coincide co da unidade de competencia da que deriva. Así é que como enunciado do contido organizador deste módulo formularemos o seguinte:

Elaborar, adaptar e probar programas para mellorar a explotación do sistema e as aplicacións.

A este contido de tipo procedimental atópanse ligados unha serie de contidos conceptuais e actitudinais que orixinan os contidos soporte das destrezas e habilidades que o alumno debe adquirir.

No **Proxecto Curricular do Ciclo Formativo**, concretamente no **Anexo II**, detállase a adaptación do DCB ás condicións específicas do IES de Rodeira.

Descríbense no citado documento as **capacidades terminais elementais**, así como os **contidos conceptuais, procedimentais e actitudinais**.

Tamén se describe no Proxecto Curricular do Ciclo Formativo a **metodoloxía** que se aplicará.

2. ORGANIZACIÓN DOS CONTIDOS

2.1. Capacidades terminais elementais e criterios de avaliación asociados.

Os criterios que se seguirán para avaliar cada unha das capacidades terminais elementais son os seguintes:

CAPACIDADES TERMINAIS ELEMENTAIS	CRITERIOS DE AVALIACIÓN
1. Precisa-los datos de entrada, de proceso e os resultados para obter, recoñecendo o fluxo de datos do proceso. 2. Analiza-las dispoñibilidades de uso do sistema en canto a capacidades de memoria e dispositivos. 3. Deseña-las estruturas de datos que interveñen no proceso, a nivel lóxico ou de abstracción. 4. Documenta-las estruturas de datos deseñadas. 5. Argumenta-la elección das estruturas de datos en termos de eficiencia e operatividade.	<ul style="list-style-type: none"> - Definir as estruturas de datos que manexa unha linguaxe de programación estruturada, a súa utilidade e ámbito de aplicación. - Definir as estruturas de datos que manexa unha linguaxe de programación orientada a obxectos, a súa utilidade e ámbito de aplicación. - Analiza-las dispoñibilidades de uso do sistema en canto a capacidades de memoria e dispositivos. - Sobre problemas de programación propostos: <ul style="list-style-type: none"> o Elixir as estruturas máis adecuadas para representar e manexar os datos do problema. o Documenta-las estruturas de datos deseñadas. o Argumenta-la elección das estruturas de datos en termos de eficiencia e operatividade. o Deseñar e interpretar os algoritmos de tratamento das estruturas para a resolución do problema. o Argumenta-la elección das estruturas de datos en termos de eficiencia e operatividade
6. Identifica-los requerimentos e especificacións do proceso.	<ul style="list-style-type: none"> - Realizar unha análise dos requirimentos e especificacións do proceso, xerando a documentación correspondente.
7. Diferencia-los distintos procesos implícitos no problema. 8. Aplica-la metodoloxía de desenvolvemento TOP-DOWN, de refinamentos sucesivos en cada proceso.	<ul style="list-style-type: none"> - Partindo dos requirimentos identificados e documentados na fase de análise, realizar un deseño dos distintos procesos implícitos no problema: <ul style="list-style-type: none"> o Usando as técnicas de programación estruturada e modular. Xustificar do seu uso. o Usando as técnicas de programación orientada a obxectos. Xustificar o seu uso.
9. Seleccionar funcións específicas, existentes en librerías do contorno da linguaxe, de utilidade no proceso.	<ul style="list-style-type: none"> - Seleccionar funcións ou clases específicas, existentes en librerías do contorno da linguaxe, de utilidade no proceso.
10. Deseñar e representa-los algoritmos de resolución dos distintos procesos, aplicando técnicas de programación. 11. Determina-los conxuntos de entradas de proba que demostrarán a corrección do algoritmo. 12. Valida-los algoritmos deseñados mediante conxuntos de datos de probas predefinidos seguindo criterios de eficiencia e eficacia. 13. Manexar correctamente os editores de programas, compiladores e manuais da linguaxe de programación. 15. Codifica-los programas fonte correspondentes ós algoritmos de solución deseñados. 17. Avalia-lo funcionamento da aplicación mediante conxuntos de datos de proba predefinidos.	<ul style="list-style-type: none"> - Sobre problemas de programación propostos : <ul style="list-style-type: none"> o Identificar e definir as estruturas de datos propias dos problemas. o Realizar o deseño de obxectos correspondente. o Definir os métodos necesarios para a comunicación entre os obxectos. o Describir os algoritmos correspondentes a cada método. o Codificar programas fonte con comentarios significativos e concisos, que definan axeitadamente as estruturas de datos e utilicen correctamente as clases, atributos, métodos, instrucións e librerías da linguaxe. o Verificación dos formatos de entrada e saída do programa o Valida-los algoritmos e obxectos deseñados mediante conxuntos de datos de probas predefinidos seguindo criterios de eficiencia e eficacia. - Avaliar a importancia da claridade e lexibilidade dos programas para facilitar o mantemento e o traballo en equipo - Documentar o código con comentarios significativos, concisos e lexibles. - Manexar correctamente os editores de programas, compiladores e manuais da linguaxe de programación.

Módulo Profesional Fundamentos de Programación

CICLO FORMATIVO: ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS, Réxime de Adultos
I.E.S. de RODEIRA - CANGAS DO MORRAZO CURSO 2006/2007

14. Empregar xeradores de código de pantallas e informes.	<ul style="list-style-type: none">- Utilidade das interfaces gráficas de usuario. Comparación de GUIs utilizadas por diversas aplicacións informáticas comerciais.- Describir a utilidade e a funcionalidade dos diferentes elementos que forman parte dunha GUI. Selección dos elementos adecuados segundo o problema.- Realizar o deseño dunha GUI dados uns requirimentos e características.- Utilizar o modelo de eventos. Introducir o modelo de eventos no deseño.- Construir interfaces gráficas de usuario, mediante:<ul style="list-style-type: none">o Linguaxes de programación.o Ferramentas visuais xeradoras de código.o Creación de páxinas web.
16. Integra-los procesos probados e depurados.	<ul style="list-style-type: none">- Sobre problemas de programación propostos, integra-los procesos probados e depurados.
18. Redacta-la documentación completa que facilite o mantemento da aplicación.	<ul style="list-style-type: none">- Sobre problemas de programación propostos, redacta-la documentación completa que facilite o mantemento da aplicación.

2.2. Relación secuenciada das unidades de traballo.

A proposta de programación está constituída por unha relación secuenciada de unidades de traballo onde se integran e desenvolven os contidos descritos no Proxecto Curricular de Ciclo Formativo.

Este módulo impartirase en 350 sesións (292 horas) a razón de 11 sesións semanais de 50 minutos.

U.T. 1: Introducción.	7 sesións
------------------------------	------------------

U.T. 2: Metodoloxía da Programación.	6 sesións
---	------------------

U.T. 3: Programación Estruturada.	70 sesións
Deseño de algoritmos.	20
Introducción á linguaxe de programación C.	25
Deseño modular, funcións e estruturas de datos.	25

U.T. 4: Desenvolvemento de Software Orientado a Obxectos.	10 sesións
--	-------------------

U.T. 5: Creación de Programas.	102 sesións
Introducción á linguaxe de programación Java.	2
Comezando a programar en Java.	50
Programación orientada a obxectos usando clases.	40
Acceso a ficheiros.	10

U.T. 6: Ferramentas de Programación Visual.	95 sesións
Netbeans.	10
Microsoft C# .NET.	30
Deseño de interfaces gráficas de usuario.	5
Interface gráfica de usuario Windows con .NET.	50

U.T. 7: Desenvolvemento Web.	60 sesións
HTML.	10
Páxinas de estilo CSS.	10
JavaScript	10
Utilización dunha ferramenta visual de creación de páxinas web.	10
Desenvolvemento dun sitio web.	20

TOTAL	350 sesións
--------------	--------------------

2.3. Elementos curriculares de cada unidade.

Unidade de Trabajo 1. Introducción.

Conceptos básicos de programación de tal xeito que o alumno comece a familiarizarse cos termos, contornos, materiais e finalidades do módulo.

Unidade de Trabajo 2. Metodoloxía da programación.

Presenta ó alumno os métodos e técnicas que permitirán desenvolver bos programas. Asimesmo exporanse as ferramentas de deseño de algoritmos, así como as pautas a seguir no seu deseño e as técnicas de programación empregadas na actualidade. Os contidos son de tipo conceptual principalmente, tamén procedimental en menor medida, e complementaranse cos contidos das seguintes unidades.

Unidade de Trabajo 3. Programación estruturada.

Preténdese que o alumno adquira as destrezas e coñecementos asociados á metodoloxía de programación estruturada, que saiba interpretar os problemas e deseñar e construír algoritmos que os resolvan.

Pretende iniciar ó alumno para que descubra os distintos tipos de datos que se empregan nas linguaxes de programación, para dedicarse logo á descrición dos tipos de datos que manexa o C e o seu emprego nos programas así como as estruturas de programación características da linguaxe de maneira que o alumno poida comezar a resolver problemas sinxelos seguindo sempre os mesmos pasos: Interpretación dos problemas, deseño do algoritmo, codificación en C, probas, depuración e documentación.

O alumno aprenderá tamén a estruturar o código empregando un deseño modular. Aplicará os coñecementos teóricos á práctica utilizando a linguaxe C, estruturando o código en funcións.

É unha unidade de traballo de contido basicamente procedimental, aínda que tamén se introducen conceptos.

Unidade de Trabajo 4. Desenvolvemento de Software Orientado a Obxectos.

Presenta ó alumno os conceptos asociados á metodoloxía de desenvolvemento de aplicacións orientada a obxectos. Tras describir os elementos básicos, estúdanse as principais técnicas de representación da información nun deseño orientado a obxectos utilizando UML.

Unidade de Trabajo 5. Creación de programas.

Aplicando os coñecementos adquiridos nas anteriores unidades de traballo, os alumnos comezarán a deseñar e crear programas utilizando a linguaxe de programación Java. Como a sintaxe de Java é igual a C en moitos aspectos, non será difícil para os alumnos adaptarse á nova linguaxe, na que ademais aplicarán os conceptos estudados na unidade de traballo 4.

Unidade de Trabajo 6. Ferramentas de Programación Visual.

Os alumnos aprenderán a utilizar a ferramenta de desenvolvemento para Java Netbeans. Aprenderán a linguaxe C# de Microsoft e utilizarán o entorno gráfico de desenvolvemento. Crearán aplicacións con interfaces gráficas de usuario para Windows.

Unidade de Trabajo 7. Desenvolvemento Web.

Os alumnos deseñarán e crearán sitios web utilizando HTML, CSS e JavaScript. Tamén aprenderán a crear os sitios web utilizando unha ferramenta visual.

2.4. Descripción das Unidades de Traballo.

Unidade de Traballo 1. Introducción.

Tempo estimado: 7 sesións de 50 minutos.

Conceptos

- Sistemas de procesamento da información.
 - A información e a súa representación.
 - Os sistemas de numeración.

- Linguaxes de programación.
 - Linguaxe máquina.
 - Linguaxe ensambladora.
 - Linguaxes de alto nivel. Compiladores e intérpretes.

- Entornos de desenvolvemento.
 - O editor de textos para a codificación de programas.
 - Compilación, interpretación e execución de programas. Exemplos.

Procedementos

- Recoñecemento dos distintos compoñentes dos sistemas de información.
- Transformacións entre sistemas de numeración.
- Aplicación de táboas de códigos, p.e. código ASCII
- Transformación de ficheiros con código fonte en C e Java en código compilado.
- Dominio do editor de textos vi. Coñecemento doutros editores de texto.

Contidos mínimos

- Recoñecemento dos distintos compoñentes dos sistemas de información.
- Transformacións entre sistemas de numeración.
- Diferencias e características de linguaxe máquina, ensambladora e de alto nivel.
- Diferencias e características dos compiladores e intérpretes.
- Dominio do editor de textos vi.

Unidade de Traballo 2. Metodoloxía da Programación.

Tempo estimado: 6 sesións de 50 minutos.

Conceptos

- Metodoloxía da programación.
 - Importancia do deseño.
 - Análise do problema.
 - Deseño do algoritmo. Deseño descendente.
 - Resolución do algoritmo na computadora.
 - Codificación.
 - Compilación e execución.
 - Verificación e depuración.
 - Ferramentas CASE.

- Técnicas de programación.
 - Programación estruturada. Linguaxes de terceira xeración.
 - Programación orientada a obxectos. OOPL.
 - Ferramentas de programación Visual.

- Documentación das aplicacións informáticas.

Procedementos

- Realización das fases de análise e deseño dun programa.
- Selección da técnica de programación máis adecuada para distintos casos.
- Identificación e descrición da documentación que se debe realizar para diversas aplicacións informáticas.

Contidos mínimos

- Interpretación dos problemas. Fases en que se divide a resolución dun problema.
- Metodoloxía da programación. Fases. Diferencias entre as fases. Utilidade. Importancia da utilización dunha metodoloxía.
- Que son e para que serven as ferramentas CASE.
- Características, diferencias, vantaxes e desvantaxes das diferentes técnicas de programación.
- Selección da técnica de programación máis adecuada para distintos casos.
- Identificación e descrición da documentación que se debe realizar para diversas aplicacións informáticas. Importancia da documentación.

Unidade de Trabajo 3. Programación Estructurada.

Tempo estimado: 70 sesións de 50 minutos.

Conceptos

- Deseño de algoritmos (20 sesións).
 - Ferramentas e notacións para o deseño de algoritmos.
 - Diagramas de fluxo.
 - Seudocódigo.
 - Variables e tipos de datos.
 - Estructuras básicas dos programas.
 - Lineal.
 - Alternativa.
 - Repetitiva.

- Introducción á linguaxe de programación C (25 sesións).
 - Comentarios.
 - Variables, constantes e tipos de datos, punteiros. Conversión de tipos (casting).
 - Operadores: aritméticos, de asignación, relacionais e lóxicos.
 - Estructuras de datos: lineais, repetitivas, iterativas.
 - Punteiros.

- Deseño modular, funcións e estruturas de datos (25 sesións).
 - O deseño modular en programación estruturada.
 - Funcións:
 - Declaración de funcións.
 - Paso de argumentos ás funcións.
 - Chamadas ás funcións.
 - Ámbito das variables: locais e globais.
 - Estructuras de datos.
 - Táboas.
 - Estructuras.

Procedementos

- Creación e deseño de algoritmos utilizando diagramas de fluxo eseudocódigo.
- Uso das técnicas de programación estruturada e modular. Xustificación do seu uso.
- Elección das estruturas de programación necesarias para a resolución dos problemas.
- Elección e utilización das estruturas de datos adecuadas segundo o problema.
- Creación de programas en C.
- Uso das variables e das funcións de librería estándar C.
- Codificación, compilación e montaxe dos programas.
- Describir as estruturas de datos estáticas que manexa unha linguaxe estruturada, a súa utilidade e ámbito de aplicación.
- Citar as operacións máis comúns sobre as estruturas estáticas de datos e explicar os seus algoritmos.
- Xustificar a importancia da elección axeitada das estruturas de datos para a resolución de problemas de programación.
- Describir as características propias da programación estruturada e modular e xustificar as vantaxes que comporta.
- Clasificar as instrucións típicas das linguaxes estruturadas segundo a súa función.
- Interpretar a sintaxe da linguaxe e as súas instrucións.
- Definir as instrucións, funcións e librerías da linguaxe máis básicas e a súa utilidade.
- Avaliar a importancia da claridade e lexibilidade dos programas para facilitar o mantemento e o traballo en equipo.

Módulo Profesional Fundamentos de Programación

CICLO FORMATIVO: ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS, Réxime de Adultos
I.E.S. de RODEIRA - CANGAS DO MORRAZO CURSO 2006/2007

- Elaborar funcións utilizando as técnicas da programación estruturada e modular.
- Documentar o código con comentarios significativos, concisos e lexibles
- Verificación dos formatos de entrada e saída da aplicación.
- Provocar erros e verificar os seus tratamentos.
- Resolución de exercicios en grupo. División do traballo en áreas funcionais.
- Sobre un problema de programación proposto:
 - Identificar e definir as estruturas de datos propias do problema.
 - Describir os algoritmos de tratamento das estruturas para a solución do problema.
 - Codificar un programa fonte con comentarios significativos e concisos, que defina axeitadamente as estruturas de datos e utilice correctamente as instrucións, funcións e librerías da linguaxe.
 - Elaborar un conxunto de datos de proba do programa deseñado.

Contidos mínimos

- Creación e deseño de algoritmos utilizando diagramas de fluxo e pseudocódigo.
- Uso das técnicas de programación estruturada e modular. Xustificación do seu uso.
- Elección e utilización das estruturas de datos adecuadas segundo o problema.
- Creación de programas en C.
- Uso das variables de librería estándar C.
- Codificación, compilación e montaxe dos programas.
- Describir as estruturas de datos estáticas que manexa unha linguaxe estruturada, a súa utilidade e ámbito de aplicación.
- Utilizar punteiros correctamente.
- Citar as operacións máis comúns sobre as estruturas estáticas de datos e explicar os seus algoritmos.
- Describir as características propias da programación estruturada e modular e xustificar as vantaxes que comporta.
- Clasificar as instrucións típicas das linguaxes estruturadas segundo a súa función.
- Interpretar a sintaxe da linguaxe e as súas instrucións.
- Avaliar a importancia da claridade e lexibilidade dos programas para facilitar o mantemento e o traballo en equipo.
- Elaborar funcións utilizando as técnicas da programación estruturada e modular.
- Documentar o código con comentarios significativos, concisos e lexibles
- Verificación dos formatos de entrada e saída da aplicación.
- Sobre un problema de programación proposto:
 - Identificar e definir as estruturas de datos propias do problema.
 - Describir os algoritmos de tratamento das estruturas para a solución do problema.
 - Codificar un programa fonte con comentarios significativos e concisos, que defina axeitadamente as estruturas de datos e utilice correctamente as instrucións, funcións e librerías da linguaxe.
 - Elaborar un conxunto de datos de proba do programa deseñado.

Unidade de Traballo 4. Desenvolvemento de Software Orientado a Obxectos.

Tempo estimado: 10 sesións de 50 minutos.

Conceptos

- Fundamentos de modelado orientado a obxectos.
 - Definición de obxecto.
 - Clasificación.
 - Persistencia.
 - Comunicación entre obxectos: as mensaxes.
- Requisitos do software. Casos de uso.
- Interacción entre obxectos.
 - Diagramas de interacción.
 - Diagramas de secuencia.
 - Diagramas de colaboración.
 - Mensaxes.
- Clases e relacións entre clases.
 - Notación gráfica.
 - Abstracción.
 - Encapsulación.
 - Relacións entre clases: asociación, agregación, xeralización, especialización.
 - Polimorfismo.
 - Diagrama de clases.
- Comportamento de obxectos.
 - Diagrama de estados.
 - Diagrama de actividade.
- Componentes.
- Distribución e despregue de componentes.
- UML. Diagramas de UML.

Procedementos

- Programación orientada a obxectos. Principais diferencias coa programación estruturada.
- Xustificar a elección dunha metodoloxía orientada a obxectos. Vantaxes.
- Describir e explicar as diferencias entre: obxecto, clase, herdanza, polimorfismo, abstracción e encapsulación.
- Análise e representación de problemas utilizando técnicas de orientación a obxectos.
- Interpretación dos diversos diagramas de UML. Finalidade de cada un. Diferencias.
- Construcción de diagramas de clases.

Contidos mínimos

- Describir o desenvolvemento de software orientado a obxectos e as principais diferencias co desenvolvemento de software "clásico".
- Definición de obxecto, clasificación, persistencia, mensaxe, caso de uso, abstracción, encapsulación, polimorfismo.
- Descrición e correcta representación das relacións entre clases: asociación, agregación, xeralización, especialización, herdanza.

Módulo Profesional Fundamentos de Programación

CICLO FORMATIVO: ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS, Réxime de Adultos
I.E.S. de RODEIRA - CANGAS DO MORRAZO CURSO 2006/2007

- Describir e explicar as diferencias entre: obxecto, clase, herdanza, polimorfismo, abstracción e encapsulación.
- Análise e representación de problemas utilizando técnicas de orientación a obxectos.
- UML: concepto, para que se utiliza, diagramas de UML.
- Interpretación e descrición dos diversos diagramas de UML. Elementos de que consta cada tipo de diagrama. Finalidade de cada un. Diferencias.
- Construcción de diagramas de clases.
- Xustificar a elección dunha metodoloxía orientada a obxectos. Vantaxes.

Unidade de Trabajo 5. Creación de Programas.

Tempo estimado: 102 sesións de 50 minutos.

Conceptos

- Introducción á linguaxe de programación Java (2 sesións).
 - Nomenclatura habitual na programación en Java.
 - Variables e tipos de datos. Conversión de tipos primitivos (casting).
 - Operadores: aritméticos, de asignación, relacionais e lóxicos.
 - Estructuras de programación: comentarios, estruturas lineais, alternativas e repetitivas.

- Comezando a programar en Java (50 sesións).
 - Táboas.
 - Clases.
 - Variables membro e locais. Visibilidade.
 - Creación de obxectos.
 - Métodos. Visibilidade. Paso de argumentos a métodos.
 - A clase String.
 - Directorio de clases.
 - Paquetes.

- Programación orientada a obxectos usando clases (40 sesións).
 - Sobrecarga de métodos.
 - Constructores.
 - Métodos e variables de clase.
 - Herdanza.
 - Clases, métodos e variables finais.
 - Clases e métodos abstractos.
 - Interfaces.
 - Clases internas.
 - Excepcións.

- Acceso a ficheiros (10 sesións).

Procedementos

- Codificación, compilación e execución de programas utilizando Java.
- Uso de variables, tipos de datos e operadores. Conversión de tipos primitivos cando é preciso.
- Análise e representación de problemas utilizando técnicas de orientación a obxectos.
- Deseño de problemas aplicando técnicas de orientación a obxectos.
- Utilización correcta das estruturas de datos simples e compostas.
- Codificación das clases e propiedades correspondentes ós deseños realizados.
- Selección da visibilidade de clases, métodos e propiedades que corresponda en cada caso.
- Creación e utilización de obxectos e métodos.
- Agrupamento das clases en paquetes.
- Utilización dos paquetes de Java.
- Aplicación correcta no deseño e codificación de programas dos conceptos de sobrecarga de métodos, constructores, métodos e variables de clase e de obxecto, herdanza, métodos e variables finais, clases finais, clases e métodos abstractos, interfaces e clases internas.
- Aplicación correcta do control de erros mediante excepcións nos programas deseñados e codificados.
- Avaliar a importancia da corrección do deseño para facilitar a reutilización de código, o mantemento e o traballo en equipo.
- Avaliar a importancia da claridade e lexibilidade dos programas para facilitar o mantemento e o traballo en equipo.
- Documentar o código con comentarios significativos, concisos e lexibles
- Verificación dos formatos de entrada e saída da aplicación.

- Provocar erros e verificar os seus tratamentos.
- Acceder a ficheiros externos desde programas Java utilizando diferentes técnicas.
- Sobre un problema de programación proposto:
 - Identificar e definir as estruturas de datos propias do problema.
 - Realizar o deseño de obxectos correspondente.
 - Definir os métodos necesarios para a comunicación entre os obxectos.
 - Describir os algoritmos correspondentes a cada método.
 - Codificar un programa fonte con comentarios significativos e concisos, que defina axeitadamente as estruturas de datos e utilice correctamente as clases, propiedades, métodos e instrucións da linguaxe.
 - Elaborar un conxunto de datos de proba do programa deseñado.

Contidos mínimos

- Codificación, compilación e execución de programas utilizando Java.
- Uso de variables, tipos de datos e operadores. Conversión de tipos primitivos cando é preciso.
- Análise e representación de problemas utilizando técnicas de orientación a obxectos.
- Deseño de problemas aplicando técnicas de orientación a obxectos.
- Utilización correcta das estruturas de datos simples e compostas.
- Codificación das clases e propiedades correspondentes ós deseños realizados.
- Selección da visibilidade de clases, métodos e propiedades que corresponda en cada caso.
- Creación e utilización de obxectos e métodos.
- Utilización dos paquetes de Java.
- Aplicación correcta no deseño e codificación de programas dos conceptos de sobrecarga de métodos, constructores, métodos e variables de clase e de obxecto, herdanza, métodos e variables finais, clases finais, clases e métodos abstractos, interfaces e clases internas.
- Aplicación correcta do control de erros mediante excepcións nos programas deseñados e codificados.
- Creación de programas con acceso a ficheiros externos. Distintas formas de acceso.
- Documentar o código con comentarios significativos, concisos e lexibles
- Sobre un problema de programación proposto:
 - Identificar e definir as estruturas de datos propias do problema.
 - Realizar o deseño de obxectos correspondente.
 - Definir os métodos necesarios para a comunicación entre os obxectos.
 - Describir os algoritmos correspondentes a cada método.
 - Codificar un programa fonte con comentarios significativos e concisos, que defina axeitadamente as estruturas de datos e utilice correctamente as clases, propiedades, métodos e instrucións da linguaxe.
 - Elaborar un conxunto de datos de proba do programa deseñado.

Unidade de Traballo 6. Ferramentas de Programación Visual.

Tempo estimado: 95 sesións de 50 minutos.

Conceptos

- Ferramentas de programación visual. Vantaxes e Desvantaxes.
- O entorno de desenvolvemento Netbeans (10 sesións).
 - Descrición dalgunhas ferramentas de programación visual.
 - Instalación en utilización de Netbeans.
- Microsoft C# .NET (30 sesións).
 - Introducción a Microsoft .NET.
 - .NET Framework.
 - Uso de variables.
 - Instrucións e excepcións.
 - Uso de parámetros.
 - Modificadores de visibilidade ou de acceso.
 - Métodos e atributos de clase.
 - Táboas.
 - Conversións de datos.
 - Encapsulación.
 - Herdanza.
 - Clases seladas.
 - Clases abstractas.
 - Interfaces.
 - Os tipos parciais de C#
 - Documentación de código fonte utilizando XML
- Deseño de interfaces gráficas de usuario (5 sesións).
 - Deseño da interface home-máquina.
 - Directrices para o deseño de interfaces, un caso concreto, C.U.A. de IBM.
 - Documentación.
- Interface gráfica de usuario Windows con .NET (50 sesións).
 - Construción dunha aplicación Windows básica.
 - Creación de Interfaces de usuario con Visual Studio .NET
 - Os controis principais.
 - Menús.
 - MDI.
 - Cadros de diálogo.

Procedementos

- Instalación e configuración de Netbeans.
- Desenvolvemento de programas en Java utilizando Netbeans.
- Utilidade das interfaces gráficas de usuario. Comparación de GUIs utilizadas por diversas aplicacións informáticas comerciais.
- Describir a utilidade e a funcionalidade dos diferentes elementos que forman parte dunha GUI. Selección dos elementos adecuados segundo o problema.
- Utilización do modelo de eventos. Introducción do modelo de eventos no deseño.
- Realización do deseño dunha GUI dados uns requirimentos e características.
- Instalación e configuración de .NET.
- Sobre un problema de programación proposto, utilizando C# .NET:
 - Identificar e definir as estruturas de datos propias do problema.
 - Realizar o deseño da interface gráfica de usuario.
 - Realizar o deseño do modelo de eventos.
 - Realizar o deseño de obxectos correspondente.
 - Definir os métodos necesarios para a comunicación entre os obxectos.
 - Describir os algoritmos correspondentes a cada método.
 - Codificar un programa fonte con comentarios significativos e concisos, que defina axeitadamente as estruturas de datos e utilice correctamente as clases, propiedades,

Módulo Profesional Fundamentos de Programación

CICLO FORMATIVO: ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS, Réxime de Adultos
I.E.S. de RODEIRA - CANGAS DO MORRAZO CURSO 2006/2007

métodos e instrucións da linguaxe, así como os elementos correspondentes á interface gráfica de usuario.

- Instalar o programa creado nun entorno de explotación.

Contidos mínimos

- Finalidade da utilización de ferramentas de programación visual. Vantaxes e desvantaxes.
- Descrición dalgunhas das ferramentas de programación visual máis utilizadas na actualidade.
- Instalación e configuración de Netbeans e .NET.
- Creación de programas en Java utilizando Netbeans.
- Sobre un problema de programación proposto, utilizando C# .NET:
 - Identificar e definir as estruturas de datos propias do problema.
 - Realizar o deseño da interface gráfica de usuario.
 - Realizar o deseño do modelo de eventos.
 - Realizar o deseño de obxectos correspondente.
 - Definir os métodos necesarios para a comunicación entre os obxectos.
 - Describir os algoritmos correspondentes a cada método.
 - Codificar un programa fonte con comentarios significativos e concisos, que defina axeitadamente as estruturas de datos e utilice correctamente as clases, propiedades, métodos e instrucións da linguaxe, así como os elementos correspondentes á interface gráfica de usuario.

Unidade de Trabajo 7. Desenvolvemento Web.

Tempo estimado: 60 sesións de 50 minutos.

Conceptos

- HTML (10 sesións).
- Páxinas de estilo CSS (10 sesións).
- JavaScript (10 sesións).
- Utilización dunha ferramenta visual de creación de páxinas web (10 sesións).
- Desenvolvemento dun sitio web (20 sesións)

Procedementos

- Deseño, creación e publicación de sitios web empregando HTML, CSS e JavaScript:
 - Sen utilizar ferramentas visuais.
 - Utilizando ferramentas visuais.

Contidos mínimos

- Deseño, creación e publicación de sitios web empregando HTML, CSS e JavaScript:
 - Sen utilizar ferramentas visuais.
 - Utilizando ferramentas visuais.

3. CRITERIOS DE AVALIACIÓN

No Proxecto Curricular do Ciclo Formativo detállase como se realizará a avaliación e indícanse os criterios xerais de avaliación dos alumnos, que se concretan a continuación.

3.1. *Avaliación Inicial.*

Realízase unha avaliación inicial co obxectivo de establecer o nivel de coñecementos, capacidades e habilidades de inicio.

3.2. *Avaliación Sumativa.*

O criterio seguido para avaliar a consecución das distintas capacidades terminais requiridas nas distintas unidades de traballo ten en conta 4 apartados:

- a. Proba escrita e/ou práctica. Ó finalizar cada unidade de traballo realízase unha proba escrita e/ou práctica na que se comprobará o grao de comprensión da unidade por parte do alumno. Na U.T. 7, na que os alumnos deberán crear un sitio web, a nota do apartado a) estará formada pola proba escrita e/ou práctica (60%) e a realización do sitio web (40%), sendo necesario aprobar por separado ambas partes. Si se detecta que o alumno copiou parcialmente ou totalmente o sitio web a nota será de 0 puntos.
- b. Traballo na aula e realización dos exercicios propostos.
- c. A falta de puntualidade reiterada na asistencia á clase poderase penalizar na nota da avaliación.
- d. A falta do debido respecto á profesora que imparte o módulo profesional tamén se poderá penalizar na nota da avaliación. Entenderanse como faltas de respecto á profesora, aparte das que xa están recollidas na lexislación vixente: empregar computadores ou o acceso a Internet sen que a profesora o teña autorizado ou indicado explicitamente. Penalízase de forma considerable copiar parcialmente ou totalmente nos exercicios que puntúan para nota.

Avaliarase cada un dos apartados anteriores cunha nota entre 0 e 10.

O alumno só poderá aproba-lo módulo se obtén unha nota maior que 5 en todas e cada unha das probas escritas e en todas e cada unha das probas prácticas.

No cálculo da nota final de cada unidade de traballo, o apartado a) terá un peso do 80%, e o apartado b) un 20%.

Os apartados c e d só penalizarán.

Os alumnos que non superen as probas deberán realizar tarefas de reforzo.

Avaliación trimestral.

Ó finalizar o primeiro e segundo trimestres farase unha proba avaliativa á que acudirán os alumnos que non superaron as probas continuas sempre que fixeran as tarefas de reforzo que previamente lles propuxo a profesora.

Na proba trimestral avaliarase de maneira independente cada unidade de traballo non superada, sendo necesario obter unha nota superior a 5 en cada unha das U.T. para superar o trimestre.

No cálculo da nota final de cada unidade de traballo, tal e como se comenta máis arriba, o apartado a) terá un peso do 80%, e o apartado b) un 20%.

A **calificación trimestral** obtida no módulo será a suma de cualificacións obtidas en cada unha das unidades de traballo ponderadas, cada unha delas, polo número de sesións na aula dedicadas á unidade de traballo.

Propóranse actividades e probas de recuperación para os alumnos que suspendesen algunha das avaliacións.

Avaliación final do módulo.

Realizarase unha proba avaliativa final á que acudirán os alumnos que non superaron algunha das unidades de traballo.

Nesta proba avaliarase de maneira independente cada unha das unidades de traballo non superadas, sendo necesario obter unha nota superior a 5 en cada unha delas para superar o módulo.

Calcularase a nota correspondente a cada unha das unidades de traballo da maneira descrita máis arriba, tendo o apartado a) un peso do 80%, e o apartado b) un 20%.

Dependendo das unidades de traballo a recuperar polo alumno poderase dividir esta proba avaliativa final en varias sesións lectivas, e tamén se poderá dividir nunha proba teórica e unha ou varias probas prácticas.

O **módulo considérase superado se** o alumno consegue unha puntuación de 5 ou máis puntos en todas e cada unha das probas. En caso de superación de tódalas probas, a nota final obtida no módulo será a suma de cualificacións obtidas en cada unha das unidades de traballo ponderadas, cada unha delas, polo número de sesións na aula dedicadas á unidade de traballo.

3.3. Avaliación dos alumnos que perden o dereito á avaliación continua.

Avaliarase ós alumnos que perden o dereito á avaliación continua mediante varias probas a realizar nunha ou varias xornadas lectivas. Nas probas esixírase o mesmo nivel de coñecementos que se lles esixe ós alumnos que asisten a clase normalmente.

Nestas probas avaliarase de maneira independente cada unha das unidades de traballo, sendo necesario obter unha nota superior a 5 en cada unha delas para superar o módulo.

O **módulo considérase superado se** o alumno consegue unha puntuación de 5 ou máis puntos en todas e cada unha das probas. En caso de superación de tódalas probas, a nota final obtida no módulo será a suma de cualificacións obtidas en cada unha das unidades de traballo ponderadas, cada unha delas, polo número de sesións na aula dedicadas á unidade de traballo.

3.4. Avaliación dos alumnos co módulo pendente doutros cursos (avaliación extraordinaria).

A estes alumnos resolveránselle tódalas dúbidas relacionadas co módulo que sexa preciso.

Propoñeráselles a realización de exercicios diversos, que abranguen tódolos coñecementos impartidos no módulo.

Avaliaranse mediante as seguintes probas:

- a. Proba teórica, escrita, con cuestións teóricas e exercicios acerca dos contidos do módulo.
- b. Realización do deseño e codificación de programas que resolvan problemas propostos utilizando a linguaxe Java. O deseño realizarase por escrito. A codificación realizarase no ordenador utilizando o editor de textos vi e o java jdk de Sun.
- c. Realización do deseño e codificación de programas que resolvan problemas propostos utilizando a linguaxe C# e o entorno de programación de Microsoft .NET. O deseño realizarase por escrito. A codificación realizarase no ordenador.
- d. Deseño e creación dun sitio web, utilizando correctamente HTML, CSS e JavaScript. O deseño realizarase por escrito, despois creárase o sitio web no ordenador.

Avaliarase de maneira independente cada unha das probas, sendo necesario obter unha nota superior a 5 en cada unha delas para superar o módulo.

4. MATERIAIS E RECURSOS DIDÁCTICOS

4.1. *Bibliografía.*

- Material dos sitios de Internet:
 - <http://asin.iesrodeira.com>
 - Java:
 - <http://www.programacion.com>
 - <http://www.javahispano.org>
 - <http://java.sun.com>
 - <http://www.netbeans.org>
 - UML
 - <http://www.usecases.org>
 - <http://www.dsic.upv.es/~uml>
 - HTML
 - <http://www.programacion.com/html/tutorial/curso/>
 - <http://sestud.uv.es/manual.esp/>
 - <http://www.w3.org/TR/REC-html40/>
 - <http://www.chami.com/html-kit/>
 - .NET
 - <http://www.microsoft.com>
 - <http://www.mono-project.com>
 - <http://ndoc.sourceforge.net>
 - Outros sitios de Internet que se vaian indicando durante o curso.
- BRUCE ECKEL. Piensa en Java. Prentice Hall, 2003.
- JESÚS BOBADILLA. Java a través de Ejemplos. Ra-Ma, 2003.
- M. RODRÍGUEZ GÓMEZ-STERN, M.A. BESTEIRO GOROSTIZAGA, Desarrollo de Aplicaciones .NET con Visual C#. McGraw-Hill 2002
- ANDREW TROELSEN, Pro C# 2005 and the .NET 2.0 Platform, Third Edition, Apress 2005
- GREGORIO CABRERA, GUILLERMO MONTOYA, Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. McGraw-Hill, 1999.
- DIEGO R. LLANOS FERRARIS. Curso de C bajo Unix. Paraninfo.

4.2. *"Software".*

- Sistemas operativos:
 - Linux.
 - Microsoft Windows XP Professional.
- Entornos de desenvolvemento:
 - GNU/Linux GCC. <http://www.gnu.org>
 - DJGPP. <http://www.delorie.com/djgpp>

Módulo Profesional Fundamentos de Programación

CICLO FORMATIVO: ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS, Réxime de Adultos
I.E.S. de RODEIRA - CANGAS DO MORRAZO CURSO 2006/2007

- Editor de textos vim. <http://www.vim.org>
- Java™ 2 Platform Standard Edition 5.0 (JDK 5.0), ou versións posteriores.
<http://java.sun.com>
- Netbeans 5.0, ou versións posteriores. <http://www.netbeans.org>
- Microsoft Visual Studio .NET 2005