

Anexo III. Modelo de programación de proba libre de módulos profesionais

1. Identificación da programación

Centro educativo

Código	Centro	Concello	Ano académico
27015773	Muralla Romana	Lugo	2013/2014

Ciclo formativo

Código da familia profesional	Familia profesional	Código do ciclo formativo	Ciclo formativo	Grao	Réxime
IFC	Informática e comunicacións	CSIFC02	Desenvolvemento de aplicacións multiplataforma	Ciclos formativos de grao superior	Probas libres
IFC	Informática e comunicacións	CSIFC03	Desenvolvemento de aplicacións web	Ciclos formativos de grao superior	Probas libres

Módulo profesional

Código MP	Nome	Horas
MP0487	Contornos de desenvolvemento	107

Profesorado responsable

M^a del Carmen Fernández Lameiro
Silvia Quintana Domao

Índice

Rexenerar co cursor no índice e premendo F9 (actualizar campos)

1. Identificación da programación.....	1
Centro educativo.....	1
Ciclo formativo.....	1
Módulo profesional.....	1
Profesorado responsable.....	1
2. Resultados de aprendizaxe e criterios de avaliación.....	3
2.1. Primeira parte da proba.....	3
2.1.1. Resultados de aprendizaxe do currículo que se tratan.....	3
2.1.2. Criterios de avaliación que se aplicarán para a verificación da consecución dos resultados de aprendizaxe por parte do alumnado.....	3
2.2. Segunda parte da proba.....	3
2.2.1. Resultados de aprendizaxe do currículo que se tratan.....	4
2.2.2. Criterios de avaliación que se aplicarán para a verificación da consecución dos resultados de aprendizaxe por parte do alumnado.....	4
3. Mínimos exixibles para alcanzar a avaliación positiva e os criterios de cualificación.....	4
4. Características da proba e instrumentos necesarios para o seu desenvolvemento.....	5
4.1. Primeira parte da proba.....	5
4.2. Segunda parte da proba.....	5

2. Resultados de aprendizaxe e criterios de avaliación

2.1. Primeira parte da proba

2.1.1. Resultados de aprendizaxe do currículo que se tratan

Resultados de aprendizaxe do currículo
RA1. Recoñece os elementos e as ferramentas que interveñen no desenvolvemento dun programa informático, e analiza as súas características e as fases en que actúan ata chegar á súa posta en funcionamento.
RA2. Avalía contornos integrados de desenvolvemento, e analiza as súas características para editar código fonte e xerar executables.
RA3. Verifica o funcionamento de programas, para o que deseña e realiza probas.
RA4. Optimiza código empregando as ferramentas dispoñibles no contorno de desenvolvemento.
RA5. Xera diagramas de clases e valora a súa importancia no desenvolvemento de aplicacións, empregando as ferramentas dispoñibles no contorno.
RA6. Xera diagramas de comportamento e valora a súa importancia no desenvolvemento de aplicacións, empregando as ferramentas dispoñibles no contorno.

2.1.2. Criterios de avaliación que se aplicarán para a verificación da consecución dos resultados de aprendizaxe por parte do alumnado

Criterios de avaliación do currículo
CA1.1. Recoñeceuse a relación dos programas cos compoñentes do sistema informático: memoria, procesador, periféricos, etc.
CA1.2. Identificáronse as fases de desenvolvemento dunha aplicación informática.
CA1.3. Diferenciáronse os conceptos de código fonte, obxecto e executable.
CA1.4. Recoñecéronse as características da xeración de código intermedio para a súa execución en máquinas virtuais.
CA1.5. Clasificáronse as linguaxes de programación.
CA1.6. Avaliouse a funcionalidade das ferramentas utilizadas en programación.
CA2.7. Identificáronse as características comúns e específicas de diversos contornos de desenvolvemento.
CA3.1. Identificáronse os tipos de probas.
CA3.2. Definíronse casos de proba.
CA3.3. Identificáronse as ferramentas de depuración e proba de aplicacións ofrecidas polo contorno de desenvolvemento.
CA4.1. Identificáronse os patróns de refactorización máis usuais.
CA4.4. Identificáronse as posibilidades de configuración dun analizador de código.
CA5.1. Identificáronse os conceptos básicos da programación orientada a obxectos.
CA5.2. Recoñecéronse os métodos de análise e modelaxe máis empregados no ámbito do desenvolvemento de aplicacións informáticas.
CA5.3. Identificáronse as ferramentas para a elaboración de diagramas de clases.
CA5.4. Interpretouse o significado de diagramas de clases.
CA6.1. Identificáronse os tipos de diagramas de comportamento.
CA6.2. Recoñeceuse o significado dos diagramas de casos de uso.
CA6.3. Interpretáronse diagramas de interacción.
CA6.5. Interpretouse o significado de diagramas de actividades.
CA6.7. Interpretáronse diagramas de estados.

2.2. Segunda parte da proba

2.2.1. Resultados de aprendizaxe do currículo que se tratan

Resultados de aprendizaxe do currículo
RA1. Recoñece os elementos e as ferramentas que interveñen no desenvolvemento dun programa informático, e analiza as súas características e as fases en que actúan ata chegar á súa posta en funcionamento.
RA2. Avalía contornos integrados de desenvolvemento, e analiza as súas características para editar código fonte e xerar executables.
RA3. Verifica o funcionamento de programas, para o que diseña e realiza probas.
RA4. Optimiza código empregando as ferramentas dispoñibles no contorno de desenvolvemento.
RA5. Xera diagramas de clases e valora a súa importancia no desenvolvemento de aplicacións, empregando as ferramentas dispoñibles no contorno.
RA6. Xera diagramas de comportamento e valora a súa importancia no desenvolvemento de aplicacións, empregando as ferramentas dispoñibles no contorno.

2.2.2. Criterios de avaliación que se aplicarán para a verificación da consecución dos resultados de aprendizaxe por parte do alumnado

Criterios de avaliación do currículo
CA2.1. Instaláronse contornos de desenvolvemento, propietarios e libres.
CA2.2. Engadíronse e elimináronse módulos no contorno de desenvolvemento.
CA2.3. Personalizouse e automatizouse o contorno de desenvolvemento.
CA2.4. Configurouse o sistema de actualización do contorno de desenvolvemento.
CA2.5. Xeráronse executables a partir de código fonte de diferentes linguaxes nun mesmo contorno de desenvolvemento.
CA2.6. Xeráronse executables a partir dun mesmo código fonte con varios contornos de desenvolvemento.
CA2.9. Xeráronse e probáronse instaladores de aplicacións en contornos propietarios e libres.
CA3.4. Utilizáronse ferramentas de depuración para definir puntos de ruptura e seguimento.
CA3.5. Utilizáronse as ferramentas de depuración para examinar e modificar o comportamento dun programa en tempo de execución.
CA3.6. Efectuáronse probas unitarias de clases e funcións.
CA3.7. Executáronse probas automáticas.
CA3.8. Documentáronse as incidencias detectadas.
CA4.2. Elaboráronse as probas asociadas á refactorización.
CA4.3. Revisouse o código fonte usando un analizador de código.
CA4.5. Aplicáronse patróns de refactorización coas ferramentas que proporciona o contorno de desenvolvemento.
CA4.6. Realizouse o control de versións integrado no contorno de desenvolvemento.
CA4.7. Utilizáronse ferramentas do contorno de desenvolvemento para documentar as clases.
CA5.5. Instalouse o módulo do contorno integrado de desenvolvemento que permite o uso de diagramas de clases.
CA5.6. Trazáronse diagramas de clases a partir das especificacións destas.
CA5.7. Xerouse código a partir dun diagrama de clases.
CA5.8. Xerouse un diagrama de clases mediante enxeñaría inversa.
CA6.4. Elaboráronse diagramas de interacción sinxelos.
CA6.6. Elaboráronse diagramas de actividades sinxelos.
CA6.8. Formuláronse diagramas de estados sinxelos.

3. Mínimos exixibles para alcanzar a avaliación positiva e os criterios de cualificación

Tódolos criterios de avaliación son mínimos esixibles.

A primeira proba terá carácter eliminatorio. O profesor ou a profesora do módulo profesional cualificará esta primeira parte da proba de cero a dez puntos. Para a súa superación as persoas candidatas deberán obter unha puntuación igual ou superior a cinco puntos.

O profesor ou a profesora do módulo profesional cualificará a segunda parte da proba de cero a dez puntos. As persoas que non superen a primeira parte da proba serán cualificadas cun cero nesta segunda parte. Para a superación da segunda parte da proba as persoas candidatas deberán obter unha puntuación igual ou superior a cinco puntos e deberá de aprobar cada un dos apartados ou exercicios nos que se divida.

A cualificación final será a media aritmética das cualificacións obtidas en cada unha das partes, expresada con números enteiros, redondeada á unidade máis próxima. No caso das persoas aspirantes que suspendan a segunda parte da proba, a puntuación máxima que poderá asignarse será de catro puntos.

4. Características da proba e instrumentos necesarios para o seu desenvolvemento

4.1. Primeira parte da proba

Proba escrita con preguntas de contestación breve que se desenvolverán en dúas sesións de 50 minutos como máximo e versará sobre unha mostra suficientemente significativa dos criterios de avaliación establecidos na programación para esta parte.

4.2. Segunda parte da proba

A segunda parte da proba consistirá no desenvolvemento de un ou de varios supostos prácticos que versarán sobre unha mostra suficientemente significativa dos criterios de avaliación establecidos na programación para esta parte. Durará como máximo 4 sesións de 50 minutos e desenvolverase nun ordenador do instituto con:

- Sistema operativo Windows 7 Professional.
- Navegadores IE, Firefox, e Google Chrome
- Contorno de desenvolvemento NetBeans 7.0.1.
- Máquina virtual PC 6.0.156.0. con Windows 7 para facer instalacións.
- Constará dalguns dos seguintes apartados ou exercicios:
- Instalar NetBeans 6.5. o Visual Studio 2008 nunha máquina virtual. Engadir módulos, personalizar e configurar calquera deses contornos. O software será subministrado polo profesorado.
- Depurar código Java en NetBeans. O código a depurar será subministrado polo profesorado.
- Deseñar, codificar e executar probas unitarias Junit en NetBeans. O código a probar será subministrado polo profesorado.
- Realizar refactorizacións sobre código Java en NetBeans. O código a refactorizar será subministrado polo profesorado.
- Analizar código Java de forma dinámica e de forma estática con NetBeans. O código a analizar será subministrado polo profesorado.
- Instalar Subversion e realizar control de versións dende NetBeans.
- Utilizar Javadoc para documentar clases Java en NetBeans. O código a documentar será subministrado polo profesorado.
- Trazar diagramas UML de clases utilizando NetBeans 6.5.
- Trazar diagramas UML sinxelos de interacción, actividades e estados utilizando NetBeans 6.5.