

## 1. Identificación da programación

### Centro educativo

Código	Centro	Concello	Ano académico
36015159	Chan do Monte	Marín	2022/2023

### Ciclo formativo

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

### Módulo profesional e unidades formativas de menor duración (\*)

Código MP/UF	Nome	Curso	Sesións semanais	Horas anuais	Sesións anuais
MP0487	Contornos de desenvolvemento	2022/2023	0	107	0

(\*) No caso de que o módulo profesional estea organizado en unidades formativas de menor duración

### Profesorado responsable

Profesorado asignado ao módulo	SAMUEL LOUREIRO CARDOSO
Outro profesorado	

Estado: Pendente de supervisión inspector

## 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 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.

#### 2.1.2. 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.
CA1.3 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.
CA2.1 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.
CA2.7 Identifícanse as características comúns e específicas de diversos contornos de desenvolvemento.
CA3.1 Identifícanse os tipos de probas.
CA3.2 Definíronse casos de proba.
CA3.3 Identifícanse as ferramentas de depuración e proba de aplicacións ofrecidas polo contorno de desenvolvemento.
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.

**Critérios de avaliación do currículo**

CA3.6 Efectuáronse probas unitarias de clases e funcións.

CA3.7 Executáronse probas automáticas.

CA3.8 Documentáronse as incidencias detectadas.

CA4.1 Identifícanse os patróns de refactorización máis usuais.

CA4.2 Elaboráronse as probas asociadas á refactorización.

CA4.3 Revisouse o código fonte usando un analizador de código.

CA4.4 Identifícanse as posibilidades de configuración dun 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.

CA5.1 Identifícanse os conceptos básicos da programación orientada a obxectos.

CA5.2 Recoñécéronse os métodos de análise e modelaxe máis empregados no ámbito do desenvolvemento de aplicacións informáticas.

CA5.3 Identifícanse as ferramentas para a elaboración de diagramas de clases.

CA5.4 Interpreouse o significado de diagramas de 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.

CA6.1 Identifícanse os tipos de diagramas de comportamento.

CA6.2 Recoñeceuse o significado dos diagramas de casos de uso.

CA6.3 Interpretáronse diagramas de interacción.

CA6.4 Elaboráronse diagramas de interacción sinxelos.

CA6.5 Interpreouse o significado de diagramas de actividades.

CA6.6 Elaboráronse diagramas de actividades sinxelos.

CA6.7 Interpretáronse diagramas de estados.

CA6.8 Formuláronse diagramas de estados sinxelos.

## 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 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.

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

Criterios de avaliación do currículo
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.
CA2.1 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.
CA2.7 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.
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.

**Crterios de avaliación do currículo**

CA3.6 Efectuáronse probas unitarias de clases e funcións.

CA3.7 Executáronse probas automáticas.

CA3.8 Documentáronse as incidencias detectadas.

CA4.1 Identificáronse os patróns de refactorización máis usuais.

CA4.2 Elaboráronse as probas asociadas á refactorización.

CA4.3 Revisouse o código fonte usando un analizador de código.

CA4.4 Identificáronse as posibilidades de configuración dun 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.

CA5.1 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 Interpretoise o significado de diagramas de 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.

CA6.1 Identificáronse os tipos de diagramas de comportamento.

CA6.2 Recoñeceuse o significado dos diagramas de casos de uso.

CA6.3 Interpretáronse diagramas de interacción.

CA6.4 Elaboráronse diagramas de interacción sinxelos.

CA6.5 Interpretoise o significado de diagramas de actividades.

CA6.6 Elaboráronse diagramas de actividades sinxelos.

CA6.8 Formuláronse diagramas de estados sinxelos.

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

A proba teórica divídese en varios bloques. É imprescindible a superación de cada un deles ( calificación  $\geq 5$  ) para superar a proba.

A cualificación de esta proba será a media ponderada de todos os bloques no caso que se superen todos. No caso contrario, a cualificación será a media ponderada dos bloques non superados.

A proba teórica é eliminatoria: é imprescindible superar esta proba para facer a proba práctica.

A proba práctica divídese en varios supostos prácticos. É imprescindible a superación de cada un deles ( calificación  $\geq 5$  ) para superar a proba.

A cualificación de esta proba será a media ponderada de todos os supostos no caso que se superen todos. No caso contrario, a cualificación será a media ponderada dos bloques non superados.

A nota obtida no módulo será media das notas obtidas na proba teórica e práctica.

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

#### 4.a) Primeira parte da proba

Proba teórica tipo test e/o preguntas curtas de relacionar, razoar ou completar relativas aos conceptos necesarios para garantir a consecución dos criterios de avaliación indicados no punto 2.

Nas probas de tipo test no caso de existir, non penalizan as respostas non contestadas. Cada erro penalizará a metade dunha resposta correcta.

Elementos necesarios:

- Documento de identificación (DNI ou pasaporte, etc....)
- Bolígrafo azul ou negro.

#### 4.b) Segunda parte da proba

Proba práctica que consistirá na execución dun ou varios supostos relacionados co currículo do módulo.

Ferramentas utilizadas:

Contorno Eclipse, VSCode, VSCodium ou Netbeans.