

PROGRAMACIÓN DIDÁCTICA

DESENVOLVIMIENTO DE APLICACIONES MULTIPLATAFORMA

# **Contornos de Desarrollo**

Curso 2026/2027

Rubén Montero Vázquez

Podes acceder á versión dixital desta programación didáctica co seguinte QR<sup>1</sup>:



## Índice

Índice.....	1
1. Introducción.....	2
2. Concreción do currículo en relación coa súa adecuación ás características do ámbito produtivo.....	3
3. Tipos de actividades e metodoloxía.....	5
4. Relación e secuencia de unidades didácticas.....	11
5. Deseño das unidades didácticas.....	13
6. Procedementos e criterios de cualificación.....	42
7. Procedemento e criterios de recuperación das partes non superadas.....	48
8. Avaliación inicial e medidas de reforzo educativo.....	52
9. Educación en valores e actividades complementarias.....	56
10. Seguimento da programación e avaliación da práctica docente.....	58
11. Bibliografía.....	60

---

1 Os códigos QR poden conter URLs maliciosas. Sospeita sempre da URL antes de acceder

## 1. Introducción

Esta programación didáctica está implementada no contexto do actual *Decreto 20/2026, do 9 de marzo, polo que se establece a ordenación xeral da formación profesional do sistema educativo de Galicia*, que ven a asentar o cambio que se ven producindo no panorama da Formación Profesional dende a entrada en vigor do *Real Decreto 659/2023, do 18 de xullo, polo que se desenvolve a ordenación do Sistema de Formación Profesional*.

Nome do módulo		<b>Contornos de Desenvolvemento</b>		
Nome do ciclo		<b>Desenvolvemento de Aplicacións Multiplataforma</b>		
Curso	Duración da sesión	Sesións semanais	Sesións anuais	Horas anuais
<b>1º</b>	<b>50 minutos</b>	<b>3</b>	<b>94</b>	<b>78</b>

A formación contribúe a alcanzar os obxectivos xerais d), e), h), i), j), e r) e as competencias d), f), h), i), j), r), t), v) e w) recollidas no *Decreto 105/2011*.

O centro educativo de aplicación é o Centro de Formación Profesional Afundación A Coruña, que consta de 1º e 2º de ciclo de Desenvolvemento de Aplicacións Web e 1º e 2º de Desenvolvemento de Aplicacións Multiplataforma. Non hai oferta modular polo que a programación cínguese a unha oferta de tipo D. A modalidade de acceso é a FP Dual Xeral e, excepcionalmente (véxase Apartado 5.15) algún ou varios estudantes poderían facer 80h de Formación en Empresa neste 1º curso, aínda que esta posibilidade só se contempla para favorecer a organización de prácticas en empresa de 1º e non se levou a cabo realmente ningún ano, polo que habería que esperar ao peche da colaboración centro-empresa para o desenvolvemento da programación didáctica de empresa.

O alumnado ten na meirande parte unhas expectativas realistas e formación inicial axeitada para cursar o ciclo, aínda que tamén esperamos unha porcentaxe de estudantes que requiran reforzo e orientacións adicionais, os cales serán valorados inicialmente (véxase Apartado 8.1) ou casos de necesidades específicas de apoio educativo (véxase Apartado 8.2). O obxectivo principal do alumnado é a inserción laboral, e en menor parte estudar a dobre titulación DAM/DAW ou matricularse posteriormente en estudos universitarios.

## 2. Concreción do currículo en relación coa súa adecuación ás características do ámbito produtivo

O sector produtivo no eido da informática está a experimentar fortes transformacións dende o 2020 en virtude das novas ferramentas de xeración de código baseadas en tecnoloxía LLM que xa todos coñecemos. As diferentes estratexias de adopción destas IAs están a producir resultados moi variados en distintas empresas, pero o obxectivo é claro: Incrementar a automatización nos procesos para axilizalos o máximo posible. Como repercute isto nas novas xeracións de programadores e programadoras?

Esta programación didáctica é un intento de dar resposta a esa complicada pregunta, mediante a planificación do módulo de Contornos de Desenvolvemento, que dentro do ciclo formativo ten un papel misceláneo e complementador ao resto de módulos. A súa supervivencia é dudosa, xa que non está recollido no *Catálogo Modular*, e só segue pertencendo ao currículo oficial a través do *Decreto 105/2011*, dada a *Disposición Transitoria Única* do *Decreto 20/2026*. Pero, precisamente, a nosa intención é explotar a relación directa e indirecta do currículo de Contornos de Desenvolvemento coas novas ferramentas de IA que fannos dudar de cales son agora as habilidades esenciais no eido da informática. A aposta desta programación didáctica baséase nos seguintes puntos:

- Todo coñecemento (fundamentos de programación estruturada, patróns de deseño, protocolos de comunicación, novas tendencias de *containerización* e escalado de aplicacións...) é valioso.
- Aínda que os LLM poidan xerar resultados (*snippets* de código, aplicacións enteiras...) máis rápido ca unha persoa, o traballo humano segue a ser imprescindible. Os LLM cometen erros que precisan ser revisados por humanos, e a perspectiva e intuición dalguén con coñecemento multidisciplinar aporta valor non reemplazable. Ademais, moitas veces, a sensación de que a coa IA faise todo “máis rápido” é falsa<sup>2</sup>.

---

2 <https://metr.org/blog/2026-02-24-uplift-update>

- A pesar disto, a realidade do novo programador ou programadora é que debe ser enormemente autoesixente e afrontar un mercado laboral máis competitivo.

O ámbito produtivo da provincia da Coruña, onde se enmarca esta programación didáctica, está formado por un tecido empresarial diverso e abundante conformado por moitas PEMES e varias grandes compañías. Da experiencia propia traballando para e con numerosas empresas obteño que o máis demandado nos perfís profesionais é (1) capacidade de aprender autónomamente e actitude de querer facelo, (2) base de coñecementos sólida e (3) boa alfabetización en IA.

Con respecto ao terceiro punto, os nosos e nosas estudantes xa coñecen e empregan con frecuencia moitas ferramentas de IA xerativa no seu día a día. Esta programación didáctica pon o acento en orientar e guiar sobre o uso axeitado destas ferramentas para a obtención de información, a comparación de solucións e a propia práctica para a aprendizaxe. Os LLM deben ser usados coma un “profesor de apoio”, non coma un un “reemplazo do estudante”.

O segundo punto que demanda o sector produtivo está (e seguirá estando) aliñado co principal obxectivo na educación: Ensinar. Non podemos incurrir na contradición de sinalar que os LLM só deben ser unha ferramenta e logo proceder a adicar a maioría da nosa enerxía e tempo a eles. O deseño e execución de probas, os patróns de refactorización, as linguaxes de modelado,... Son contidos teóricos necesarios deste módulo que se traballan en profundidade e que deben pasar a formar parte da base competencial do alumnado.

Sobre o primeiro punto e máis importante: Esta programación non só está focalizada en que os estudantes aprendan e melloren as súas capacidades para aprender de xeito autónomo. Tamén pon especial atención á motivación do alumnado e ao obxectivo de promover a curiosidade e o inconformismo coma características imprescindibles non só dun bó traballador ou traballadora, senón tamén dunha persoa feliz e completa. A comunicación humana e empatía serán pezas clave na actuación docente, quen sempre debe adaptar a profundidade dos contidos e os ritmos de ensinanza-aprendizaxe ao clima na aula, así coma atender a diversidade e fomentar o respecto, a solidariedade, a autoesixencia e a bondade.

### 3. Tipos de actividades e metodoloxía

Do mesmo xeito que copiar (CTRL+C) e pegar (CTRL+V) código é un forte indicativo de antipatrón nun proxecto *software*, esta programación didáctica ten varias tipoloxías de actividades que se repiten ao longo do curso, como é habitual na práctica docente. Non tería moito sentido replicar as súas descrições no Apartado 5: Deseño das Unidades Didácticas.

Polo tanto, expóñense agora os “arquetipos” de actividade, que serán referenciados máis adiante.

#### 3.1. Tipo “A”: Presentación interactiva. Investigación e elaboración independente dun esquema/resumo de contidos teóricos

Este tipo de actividade comeza cunha breve discusión introdutoria de 5-10 minutos na



que o docente lanza preguntas á clase relacionadas cos contidos teóricos da actividade sen chegar a aportar as respostas, tan só para fomentar o clima de participación. A continuación, empregando a ferramenta *Wayground*<sup>3</sup> proxéctase unha presentación interactiva de elaboración propia durante aproximadamente 20-30 minutos. Cada estudante

conéctase co seu dispositivo móbil á presentación, que non só consta de material visual para apoiar as explicacións senón tamén de preguntas que deben ser contestadas polo alumnado entre certas diapositivas para así manter a súa atención.



As preguntas en sí non son avaliábeis xa que chegan antes de que se produza ningunha explicación teórica para abordalas. De feito, a meirande parte dos contidos nos que se sustenta a actividade non están reflectidos na presentación *Wayground*. Dita presentación serve para contextualizar o traballo a realizar, e, ao final ten unha última diapositiva que enuncia en forma de preguntas cales son os contidos e coñecementos que se deben adquirir.

---

3 <https://wayground.com>

Despois da presentación, esa última diapositiva queda proxectada e o alumnado recibe instrucións para buscar información en liña, de xeito autónomo e crítico, sobre as preguntas plantexadas. Debe adicar o resto de tempo da actividade (unha ou dúas sesións) a elaborar un esquema/resumo dos contidos plantexados. As tarefas do docente neste tempo serán acompañar e orientar ao alumnado de xeito individual ou en grupos



pequenos plantexando como a busca de información pode ser de máis calidade, cales serían as mellores estratexias para sintetizar a información e como se podería traballar de xeito autónomo para afianzar o coñecemento (por exemplo, pedindo a un LLM un exame tipo *test* sobre os contidos teóricos).

Como se avalía este tipo de actividade? Os esquemas/resumo son entregados ao docente e revisados fóra da aula mediante unha rubrica na que se valora a completude e claridade dos mesmos. Unha pequena parte da nota de avaliación está ligada a esta revisión.

Como se controla que os estudantes non copian e pegan o esquema obtido do LLM ou dun compañeiro ou compañeira? Sinxelamente, non se pode garantir isto completamente. Pero, por iso, a maior parte da nota de avaliación da actividade (70-80% da nota) obténse do resultado dun exame individual tipo *test* e/ou resposta curta. Este exame faise interrompendo outra actividade varias sesións despois, cando cada estudante xa recibiu as correccións sobre o seu esquema/resumo e tivo oportunidade de revisalas e comprendelas.



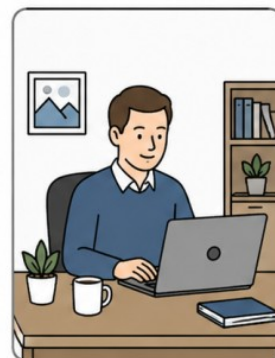
Neste tipo de actividade foméntase o traballo autónomo de xeito efectivo, practicando e mellorando as competencias para aprender a aprender. Tamén, aínda que os alumnos e alumnas colaboren no proceso de elaborar o esquema/resumo, a mensaxe por parte do docente debe ser clara: “Debemos aprender significativamente para que o desempeño individual no exame, e no futuro máis aló do exame, sexa satisfactorio.”

Como desvantaxes nesta tipoloxía de actividade debemos sinalar que pon moita “esperanza” na autonomía dos estudantes. Polo tanto, só se emprega coma actividade breve e

introdutoria a un novo bloque de materia, e se debe ter en conta que a súa efectividade está limitada polo contexto e capacidades iniciais do alumnado. Quizais noutro centro ou etapa educativa non funcionaría igual de ben.

### 3.2. Tipo “B1”: Traballo autónomo nun repositorio Git con tarefas aleatorizadas individualmente e tests automáticos

Este tipo de actividade comeza co traballo previo do docente, que crea un usuario e proxecto para cada estudante nunha instalación autoxestionada de GitLab e aleatoriza un número (aproximadamente entre 25 e 50) de tarefas que son publicadas mediante *scripts* de xestión de elaboración propia. Cada estudante pode entón acceder as súas tarefas (*issues*) no portal web de GitLab e traballar nelas no seu ordenador. Cada tarefa ten un *test* automático asociado personalizado para cada alumno ou alumna, quen pode executalo en calquera momento para verificar o seu avance. Todo o material para o módulo de Contornos de Desenvolvemento e para outros módulos está dispoñible de xeito aberto no proxecto PATACA<sup>4</sup>.



Nesta actividade, obviamente, o apoio docente é esencial. A primeira sesión comezará cunha explicación introdutoria teórica por parte do docente apoiándose unicamente no encerado, e/ou unha exposición práctica da ferramenta ou contido a traballar. A continuación, o alumnado pasará a traballar autónomamente nas tarefas publicadas no seu GitLab. Se o contido teórico da actividade é grande, o docente empregará tempo de sesións adicionais para fragmentar a parte expositiva teórico-práctica. A idea é que por cada tres ou catro sesións, haxa unha ou dúas exposicións teórico-prácticas de entre 20 e 25 minutos. A claridade, concisión e oratoria do docente deben estar traballadas

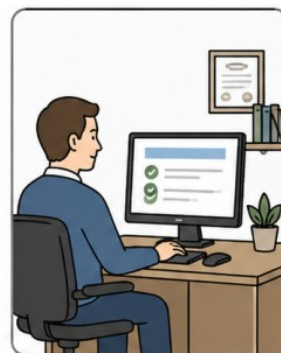


<sup>4</sup> <https://gitlab.com/ruben.montero/developer-course-automation>

previamente e focalizarse en motivar o interese do alumnado na actividade e fomentar a súa participación en todo momento empregando recursos adicionais cando sexa posible.

O resto do tempo que o docente non estea expoñendo de xeito maxistral, fará un seguimento individual ou en grupos pequenos do avance na práctica do alumnado, resolvendo dúbidas, dando explicacións complementarias, fortalecendo a súa relación cordial co alumnado e facendo un seguimento específico aos estudantes con necesidades educativas especiais que foron valorados ao inicio e durante o curso. É imprescindible que o docente non adique nin un só minuto ás súas tarefas na mesa do profesor. Isto distánciao do alumnado, desmotiva aos estudantes e supón unha contradición aos principios de respecto, compromiso e solidariedade que queremos trasladar en todo momento.

Esta actividade avalíase mediante a execución dos *tests* automáticos personalizados para cada alumno ao remate da data de entrega. Como se garante que cada alumno ou alumna traballa de xeito autónomo, motivado e sen copiarse ou aplicar técnicas fraudulentas? O traballo docente para que acompañar e motivar ao alumnado é moi



importante neste aspecto. Tamén, as tarefas a realizar son pequenas e están aleatorizadas, polo que copiar dun compañeiro supón despois ter que adaptar o teu código aos teus enunciados. Desde un punto de vista pragmático, non paga a pena. Outro valor engadido desta metodoloxía é que o traballo do alumnado queda rexistrado nun repositorio Git, polo que pódese facilmente revisar *a posteriori* e ver a que hora se fixo cada *commit*, e os cambios que contiña. É relativamente sinxelo impugnar un traballo fraudulento.

Coma desvantaxe, ata que non se traballan os contidos de control de versións e o alumnado non ten entendido “que é Git”, non se pode poñer en práctica este tipo de actividade.

### **3.3. Tipo “B2”: Traballo autónomo nun repositorio Git con tarefas aleatorizadas individualmente**

Este subtipo é idéntico ao “B1”, pero as tarefas no repositorio GitLab do estudante non van acompañadas de *tests* automáticos que permiten tanto ao estudante coma ó docente

avaliar o traballo. En lugar diso, o propio obxectivo das tarefas é o desenvolvemento de *tests* automáticos de código e/ou de interface por parte do estudante.

### 3.4. Tipo “C1”: Traballo autónomo nunha práctica PDF con campos de resposta editables homoxénea para todo o alumnado

Este tipo de actividade é similar ao tipo “B” en que existen exposicións teórico-prácticas por parte do docente que acompañan un traballo autónomo. Pero en lugar de traballar nun repositorio Git personalizado, cada alumno descargará da aula virtual un enunciado práctico en PDF que contén campos editables para encher as respostas



dos resultados obtidos durante a execución de comandos, instalación de ferramentas, etc. É unha metodoloxía de traballo habitual e non excesivamente innovadora, pero na que se fomenta o uso crítico e intelixente da IA nos enunciados PDF, que teñen pequenas xanelas amarelas con texto complementario suxerindo cal sería o “uso correcto da IA” para cada apartado. Ademais, algúns enunciados teñen texto branco sobre fondo branco con preguntas adicionais. Isto serve para identificar a aquelas persoas que suban o PDF da práctica a un LLM e copien e peguen as respostas. Está probado que segue sendo eficaz en xuño de 2026, e



aínda que non garante ao 100% a identificación de entregas fraudulentas, cada actividade deste tipo vai acompañada dun exame de avaliación tipo *test* e/ou resposta curta, polo que asúmese que xunto co traballo de acompañamento do docente e tendo en conta a etapa educativa, o traballo do alumnado vai ser en global o axeitado. Nunha actividade específica (a Actividade 5) o exame é substituído por unha avaliación individualizada mediante rúbrica de observación.

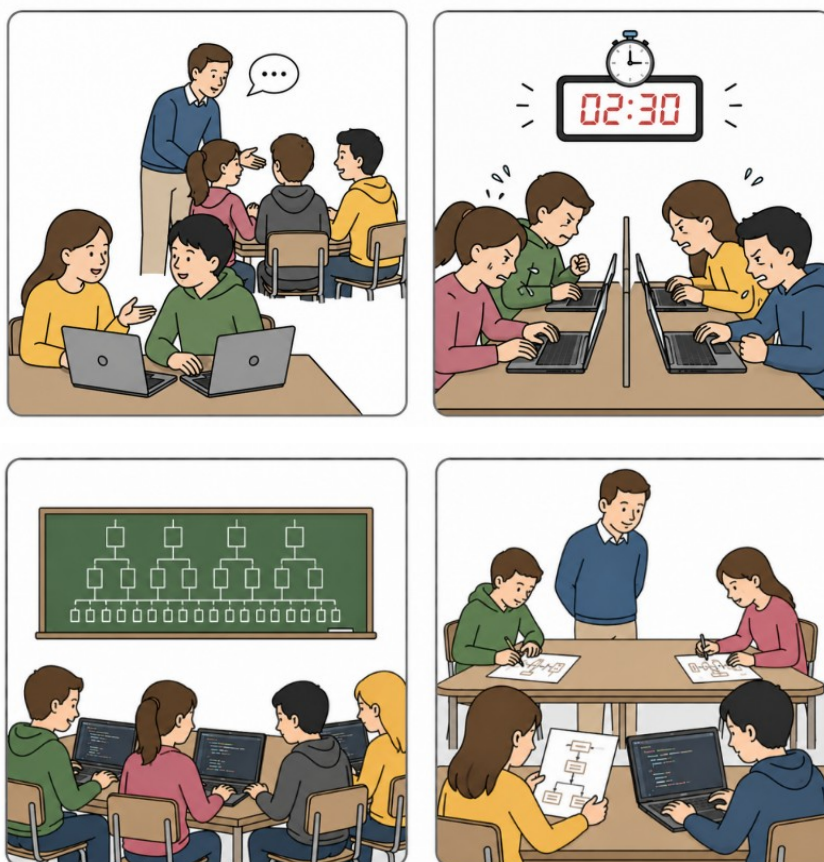
Como valor engadido a este tipo de actividade, destacamos que o uso de campos editables nos enunciados en PDF facilita a extracción sistemática cun *script* Python da información entregada por todos os estudantes, e facilita a súa avaliación mediante rúbrica por parte do docente.

### 3.5. Tipo “C2”: Traballo autónomo nunha práctica PDF homoxénea para todo o alumnado

Este subtipo é idéntico ao “C1”, pero o enunciado PDF da práctica non contén campos editables. As instrucións indican aos estudantes que se rexistren en GitHub e leven a cabo tarefas relacionadas ao uso de ferramentas de control de versións. A avaliación desta actividade faise cunha rúbrica sobre o traballo de cada estudante no seu repositorio, sen necesidade de exame tipo *test*.

### 3.6. Outros tipos

Hai varios tipos distintos de actividades que se levan a cabo durante o curso e implican máis colaboración entre o alumnado e innovación educativa: Traballo colaborativo en Git, competicións “contra reloxo”, campionatos de carácter eliminatorio, *role-play* colaborativo no eido do levantamento de requisitos *software*... Sen embargo, estas actividades son únicas e non se repiten, polo que serán explicadas na súa respectiva descripción no Apartado 5.



#### 4. Relación e secuencia de unidades didácticas

A secuenciación das unidades didácticas fíxose de forma lóxica atendendo a como van evolucionando as necesidades do alumnado ao longo do curso. Nas primeiras semanas encaixan mellor actividades nas que se da un *feedback* máis áxil ao alumnado e se transmite pronto (non de palabra, senón na práctica) que se debe axustar o esforzo ao nivel educativo. Tamén se pon o acento na alfabetización en IA para promover o seu uso correcto e ético.

A UD4 rompe coa orde establecida de resultados de aprendizaxe no currículo oficial coa intención de introducir tempranamente as ferramentas de control de versións necesarias para as actividades seguintes. O RA3, relacionado ás probas *software*, considérase de especial relevancia no eido produtivo na actualidade e ocupa todo o segundo trimestre. Outras xustificacións son explicadas no Apartado 5: Deseño das Unidades Didácticas.

UD	Actividade	Tipo	Sesións	Res. de aprendizaxe
1	1. Introducción e avaliación inicial. Linguaxes	A	2	RA1
	2. Java - Programas, bytecode, JVM	C1	4	RA1
2	3. Fases de desenvolvemento	A	2	RA1
	4. Java - Procesador, memoria, pila de execución	C1	4	RA1
3	5. Instalación e uso de IDEs	C2	6	RA2
	6. Uso responsable da IA. Avaliación individual.	Outro	6	RA2
4	7. Instalación e uso básico de Git	C2	5	RA4
5	8. Traballo colaborativo en Git. Ramas	Outro	6	RA4
	9. Forensethón: Que <i>commit</i> rompeu a <i>app</i> ?	Outro	1	RA4
6	10. Verificación de programas. Conceptos básicos	A	2	RA3
	11. Probas unitarias. Campionato de probas	Outro	8	RA3
7	12. Depuración	C1	6	RA3
8	13. XadrezApp. Requisitos e casos de proba	Outro	4	RA3
9	14. XadrezApp. Probas Android con Espresso	B2	6	RA3
10	15. Probas web con Cypress	B2	5	RA3
11	16. Introducción a diagramas de clase UML	A	2	RA5
	17. Introducción a diagramas de estado e secuencia	A	3	RA6
12	18. UML – Diagramas de clase	B1	6	RA5
13	19. UML – Diagramas de comportamento	B1	6	RA6
14	20. CLI Java a partir de diagrama de estados	Outro	4	RA6
15	21. Refactorización e optimización	B1	6	RA4

No seguinte calendario expónse a temporalización prevista para as unidades didácticas e actividades para o curso 2026/27, con datas e períodos vacacionais recollidos na *Orde do 4 de xuño de 2026 pola que se aproba o calendario escolar para o curso 2026/27*, tendo en conta tamén os festivos de distinto ámbito e unha previsión das semanas de avaliacións.

As 3 sesións semanais do módulo de Contornos de Desenvolvemento teñen lugar os mércores (2 sesións) e os venres (1 sesión).

As recuperacións do primeiro trimestre teñen lugar a semana do 14 de decembro e as do segundo trimestre, a semana do 5 de abril. En previsión disto, as Actividades 11 e 17 que coinciden nesas datas adecúanse á realización de exames de recuperación por parte dos suspensos paralelamente á realización das actividades polo resto da clase.

Setembro 2026							Outubro 2026							Novembro 2026							
	1	2	3	4	5	6				1	2	3	4							1	
7	8	9	10	11	12	13	5	6	7	8	9	10	11	2	3	4	5	6	7	8	
14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15	
21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22	
28	29	30					26	27	28	29	30	31		23	24	25	26	27	28	29	
														30							
Decembro 2026							Xaneiro 2027							Febreiro 2027							
	1	2	3	4	5	6				1	2	3	1	2	3	4	5	6	7		
7	8	9	10	11	12	13	4	5	6	7	8	9	10	8	9	10	11	12	13	14	
14	15	16	17	18	19	20	11	12	13	14	15	16	17	15	16	17	18	19	20	21	
21	22	23	24	25	26	27	18	19	20	21	22	23	24	22	23	24	25	26	27	28	
28	29	30	31				25	26	27	28	29	30	31								
Marzo 2027							Abril 2027							Maio 2027							
1	2	3	4	5	6	7				1	2	3	4							1	2
8	9	10	11	12	13	14	5	6	7	8	9	10	11	3	4	5	6	7	8	9	
15	16	17	18	19	20	21	12	13	14	15	16	17	18	10	11	12	13	14	15	16	
22	23	24	25	26	27	28	19	20	21	22	23	24	25	17	18	19	20	21	22	23	
29	30	31					26	27	28	29	30			24	25	26	27	28	29	30	
														31							
Xuño 2027																					
	1	2	3	4	5	6															
7	8	9	10	11	12	13															
14	15	16	17	18	19	20															
<b>21</b>	22	23	24	25	26	27															
28	29	30																			

	Períodos de vacacións ou festivos
	Período de exames de avaliación trimestral

## 5. Deseño das unidades didácticas

### 5.1. Unidade Didáctica 1: Linguaxes de programación. Ferramentas Java. JVM

UD1	Actividade 1: Introducción e avaliación inicial. Linguaxes	2 sesións	T1   Setembro
		<b>Tipo:</b> A (ver Apartado 3)	
<p>A discusión introdutoria desta actividade constará de preguntas coma: “Que esperades de estudar este ciclo? Que é un programa informático? Que é unha linguaxe de programación? Que tipos hai?” Non se darán respostas directas por parte do docente, pois preténdese fomentar o interese e a inquietude, así coma o clima de participación.</p> <p>A presentación expositiva <i>Wayground</i> terá preguntas non avaliadas. Pero o informe con datos de preguntas usarase para a avaliación inicial. Os estudantes con só 0 ou 1 preguntas acertadas serán obxecto de especial atención e obterán apoio específico nas seguintes actividades prácticas, estudando cada caso particularmente.</p> <p>A última diapositiva do <i>Wayground</i> contén as orientacións para levar a cabo, de xeito autónomo, a elaboración dun esquema/resumo sobre características das linguaxes de programación. Dito resumo será avaliado polo docente despois das sesións. Eses contidos tamén serán avaliados mediante un exame tipo <i>test</i> e/ou resposta curta tras a Actividade 2.</p>			
Criterios avaliación		Instrumento	UD
CA1.5. Clasifícaronse as linguaxes de programación.		Rúbrica do resumo	20%
Contidos	BC1.1. Concepto de programa informático		
	BC1.4. Tipos de linguaxes de programación		
	BC1.5. Características das linguaxes máis difundidas		
Tarefas dos estudantes			
Participación en discusión introdutoria			
Participación activa en presentación <i>Wayground</i>			
Creación de esquema/resumo sobre tipos de linguaxes			
Tarefas do docente			
Discusión introdutoria e presentación con <i>Wayground</i>			
Elaboración de avaliación inicial a partir dos resultados das preguntas da presentación			
Acompañamento no traballo autónomo			
Corrección de esquemas/resumos			

<b>Obxectivos específicos</b>	
Fomentar o clima de participación e o interese pola materia, así coma introducir ao alumnado no uso responsable da IA como ferramenta para adquirir coñecementos (e non para substituílo no proceso de aprendizaxe). Aprender, de forma introdutoria, sobre características das linguaxes de programación. Estes obxectivos están plantexados así para que sirvan de transición útil entre o período de vacacións do que ven o alumnado e o ritmo de traballo e filosofía de curiosidade e esforzo constante desta programación.	
<b>Resultados ou produtos</b>	Informe <i>Wayground</i> con desglose de respostas para a avaliación inicial
	Esquema ou resumo de cada estudante sobre características das linguaxes
	Nota de avaliación do esquema de cada estudante e correccións no mesmo
<b>Recursos específicos</b>	
Presentación <a href="https://wayground.com/admin/presentation/69d769bca434c82a728a9c25">https://wayground.com/admin/presentation/69d769bca434c82a728a9c25</a>	
Dispositivos móbiles (estudantes)	

UD1	Actividade 2: Java - Programas, bytecode, JVM	4 sesións	T1   Setembro
		<b>Tipo:</b> C1 (ver Apartado 3)	
<p>Tras entregar aos alumnos e alumnas os esquemas da Actividade 1 con correccións e valoración, indícase que ao final desta Actividade 2 haberá unha proba de avaliación escrita.</p> <p>A introdución teórico-práctica desta actividade C1 tratará sobre que é Java, que é a máquina virtual de Java e que é o <i>bytecode</i>, poñéndoo en contraposición a código compilado de C.</p> <p>Durante as seguintes tres sesións prácticas o docente actuará como asistente e facilitador, resolvendo dúbidas e facendo seguimento activo do traballo do alumnado, especialmente daqueles con peor resultado na avaliación inicial.</p> <p>A práctica ten unha sección 6 complementaria non avaliable, que invita aos estudantes que rematen antes do tempo previsto a afrontar desafíos máis complexos.</p>			

<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA1.3. Diferenciáronse [...] código fonte, obxecto e executable.		Rúbrica da práctica	10%
CA1.4. [...] características da xeración de código intermedio [...]		Rúbrica da práctica	10%
CA1.5. Clasificáronse as linguaxes de programación.		Exame escrito	25%
CA1.6. Avaliouse a funcionalidade das ferramentas [...]		Rúbrica da práctica	10%
		Exame escrito	25%
<b>Contidos</b>	BC1.3. Código fonte, código obxecto e código executable. [...] linguaxe intermedia.		
	BC1.6. Proceso de obtención de código executable a partir do código fonte [...]		
<b>Tarefas dos estudantes</b>			
Traballo autónomo para cubrir o documento da práctica, apoiándose de forma dirixida na IA			
Realización do exame de avaliación da Actividade 1 e 2			
<b>Tarefas do docente</b>			
Seguimento activo do traballo autónomo e apoio específico			
Exposición teórico-práctica inicial e exposicións complementarias			
Valoración das prácticas mediante rúbrica			
Corrección dos exames da Actividade 1 e 2			
<b>Obxectivos específicos</b>			
Aprendizaxe dos comandos principais para compilación e execución de código Java desde terminal, mentres se promove de forma dirixida e orientada o uso da IA de xeito autónomo e responsable. Estes obxectivos están aliñados co obxectivo global do curso e tamén serven de base para o traballo sobre IDEs en seguintes unidades didácticas.			
<b>Resultados ou produtos</b>	Documento da práctica cuberto por cada estudante		
	Avaliación da práctica mediante rúbrica de avaliación		
	Nota do exame de avaliación da Actividade 1 e 2		
<b>Recursos específicos</b>			
Práctica: <a href="https://pd-contornos-ruben.org/assets/UD1_Actividade2_BytecodeJVM.pdf">https://pd-contornos-ruben.org/assets/UD1_Actividade2_BytecodeJVM.pdf</a>			
Exame tipo <i>test</i> e/ou resposta curta da Actividade 1 e 2			

## 5.2. Unidade Didáctica 2: Fases de desenvolvemento. Uso do procesador, memoria, etc.

UD2	Actividade 3: Fases de desenvolvemento	2 sesións	T1   Setembro
		<b>Tipo:</b> A (ver Apartado 3)	
<p>Ao inicio da actividade discutirase sobre: “Cantas persoas desenvolveron determinado xogo?” “Como se sabe se un programa funciona ben ou mal?” ou “Como se colabora con outras persoas cando se fai <i>software</i>?” Despois ten lugar a presentación <i>Wayground</i> na que o informe das respostas usarase para complementar a avaliación inicial da Actividade 1.</p> <p>A última diapositiva contén indicacións para a elaboración autónoma dun esquema/resumo sobre características enunciadas acerca de desenvolvemento <i>software</i>, metodoloxías e fases de desenvolvemento. Dito resumo será avaliado polo docente despois das sesións.</p>			
Criterios avaliación		Instrumento	UD
CA1.2. Identificáronse as fases de desenvolvemento [...]		Rúbrica do resumo	20%
Contidos	BC1.2. Fases do desenvolvemento [...] análise, deseño, codificación, probas, [...]		
Tarefas dos estudantes			
Participación en discusión introdutoria			
Participación activa en presentación <i>Wayground</i>			
Creación de esquema/resumo sobre tipos de linguaxes			
Tarefas do docente			
Discusión introdutoria e presentación con <i>Wayground</i>			
Actualización da avaliación inicial a partir dos resultados das preguntas da presentación			
Acompañamento no traballo autónomo			
Corrección de esquemas/resumos			
Obxectivos específicos			
Repetir a metodoloxía da Actividade 1 (tipo A) para que o alumnado se sinta familiarizado e reforce a súa relación responsable e saudable coa IA. Aprender, de forma eminentemente teórica, sobre fases de desenvolvemento e distintas metodoloxías.			
Resultados ou produtos	Informe <i>Wayground</i> con desglose de respostas para actualizar a avaliación inicial		
	Esquema ou resumo de cada estudante sobre fases de desenvolvemento		
	Nota de avaliación do esquema de cada estudante e correccións no mesmo		
Recursos específicos			
Presentación <a href="https://wayground.com/admin/presentation/69f05cd64e6fb4d56e95c03e">https://wayground.com/admin/presentation/69f05cd64e6fb4d56e95c03e</a>			
Dispositivos móbiles (estudantes)			

UD2	Actividade 4: Java - Procesador, memoria, pila de execución	4 sesións	T1   Set. – Out.
		<b>Tipo:</b> C1 (ver Apartado 3)	

Ao inicio da primeira sesión entrégase aos alumnos e alumnas os esquemas da Actividade 3 con correccións e valoración, que deberán ser tidas en conta xunto ao aprendido nesta práctica da Actividade 4 para un exame tipo *test* e/ou resposta curta ao final da última sesión.

A exposición teórico-práctica desta Actividade 4 tipo C1 trata sobre a relación entre a lóxica dos programas e o consumo de recursos *hardware* real que se produce durante a execución, centrándose en CPU e memoria, e facendo unha introdución sobre tamaño dos tipos de variables ou programas multifío. Só usárase o encerado coma apoio.

O traballo autónomo dos estudantes será acompañado doutra explicación teórico-práctica ao inicio da segunda ou terceira sesión, empregando IntelliJ IDEA e un programa Java sinxelo similar aos que plantexa a práctica.

Ao remate da última sesión, despois da entrega da práctica, realizarase o exame individual.

<b>Criterios avaliación</b>	<b>Instrumento</b>	<b>UD</b>
CA1.1. [...] relación dos programas con [...] memoria, procesador	Rúbrica da práctica	15%
	Exame escrito	25%
CA1.2. Identificáronse as fases de desenvolvemento [...]	Exame escrito	25%
CA1.6. Avaliouse a funcionalidade das ferramentas [...]	Rúbrica da práctica	15%

<b>Contidos</b>	BC1.1. Concepto de programa informático
	BC1.2. Fases do desenvolvemento dunha aplicación: análise, deseño, [...]

<b>Tarefas dos estudantes</b>	<b>Tarefas do docente</b>
Traballo autónomo sobre un documento con orientacións prácticas no ordenador, apoiándose de forma dirixida na IA de ser preciso	Acompañamento no traballo autónomo e apoio específico
Exame de avaliación da Actividade 3	Valoración das prácticas mediante rúbrica Corrección dos exames da Actividade 3

<b>Tarefas dos estudantes</b>
Traballo autónomo para cubrir o documento da práctica, apoiándose de forma dirixida na IA
Realización do exame de avaliación da Actividade 3 e 4
<b>Tarefas do docente</b>
Seguimento activo do traballo autónomo e apoio específico
Exposición teórico-práctica inicial e exposicións complementarias
Valoración das prácticas mediante rúbrica
Corrección dos exames da Actividade 3 e 4

<b>Objetivos específicos</b>	
<p>Aprendizaxe sobre a relación entre a CPU e a memoria empregada polos programas, distinguindo os conceptos de clase e instancia, coma base necesaria para que o alumnado poida entender e discutir distintas solucións <i>software</i> non só no módulo de Contornos de Desenvolvemento, senón tamén en Programación. Adicionalmente, promóvese de xeito dirixido e orientado o uso responsable e autónomo da IA.</p>	
<b>Resultados ou produtos</b>	Documento da práctica cuberto por cada estudante
	Avaliación da práctica mediante rúbrica de avaliación
	Nota do exame de avaliación da Actividade 3 e 4
<b>Recursos específicos</b>	
Práctica: <a href="https://pd-contornos-ruben.org/assets/UD2_Actividade4_ProcesadorMemoriaJava.pdf">https://pd-contornos-ruben.org/assets/UD2_Actividade4_ProcesadorMemoriaJava.pdf</a>	
Exame tipo <i>test</i> e/ou resposta curta da Actividade 3 e 4	

### 5.3. Unidade Didáctica 3: IDEs. Uso responsable da IA

UD3	Actividade 5: Instalación e uso de IDEs	6 sesións	T1   Outubro
		<b>Tipo:</b> C2 (ver Apartado 3)	
<p>Nesta actividade introdúcense os IDEs e as súas funcionalidades comúns e específicas.</p> <p>Na introdución teórico-práctica da primeira sesión discútese sobre que é un IDE e que problemas resolve, ironizando sobre como sería o desenvolvemento <i>software</i> empregando java e javac desde liña de comandos. Fálase de IDEs axéuticos (IA), e as oportunidades que ofrecen en contraposición aos desafíos coma a dificultade de acceso a traballos júnior.</p> <p>Como mínimo, prevese que haxa outra sesión expositiva sobre o uso de Android Studio explicando a integración do código XML e Kotlin, e o uso do emulador.</p> <p>O traballo autónomo do alumnado nesta actividade práctica tipo C2 será avaliado ao remate da seguinte UD3, ao final de outubro. Polo tanto o traballo de seguimento activo por parte do docente será especialmente importante, prestando especial atención a que cada estudante reciba axuda e atención individualizada. Despois dun mes de setembro moi interactivo e con moito <i>feedback</i>, este mes de outubro non ten tantas avaliacións.</p>			

Criterios avaliación		Instrumento	UD
CA2.1. Instaláronse contornos de desenvolvemento [...]		Lista de cotexo	10%
CA2.2. Engadíronse e elimináronse módulos no contorno [...]		Lista de cotexo	10%
CA2.4. Configurou o sistema de actualización do contorno [...]		Lista de cotexo	10%
CA2.5. Xeráronse executables a partir de código fonte [...]		Lista de cotexo	15%
CA2.6. Xeráronse executables a partir dun mesmo código [...]		Lista de cotexo	15%
<b>Contidos</b>	BC2.1. Funcións dun contorno de desenvolvemento.		
	BC2.2. Instalación dun contorno de desenvolvemento.		
	BC2.4. Instalación e desinstalación de módulos adicionais.		
	BC2.5. Personalización do contorno.		
	BC2.6. Mecanismos de actualización.		
	BC2.7. Contornos de desenvolvemento máis empregados (libres e comerciais).		
	BC2.8. Uso [...] dun contorno [...]: edición de programas e xeración de executables.		
	<b>Tarefas dos estudantes</b>		
Traballo autónomo sobre unha práctica de IDEs que servirá de base para familiarizarse no traballo de programación non só nunha contorna, senón comprendendo a variada oferta.			
<b>Tarefas do docente</b>			
Exposicións teórico-prácticas inicial e na terceira sesión.			
Acompañamento no traballo autónomo.			
<b>Obxectivos específicos</b>			
Familiarización cos IDEs principais empregados para este e outros módulos.			
<b>Resultados ou produtos</b>	Tutorial de IntelliJ IDEA completado		
	Proxecto en VSCode e complementos instalados		
	Proxecto en Android Studio con certas configuracións no IDE		
	Proxecto en Cursor empregando críticamente algúns <i>prompts</i>		
<b>Recursos específicos</b>			
Práctica: <a href="https://pd-contornos-ruben.org/assets/UD3_Actividade5_IDEs.pdf">https://pd-contornos-ruben.org/assets/UD3_Actividade5_IDEs.pdf</a>			

UD3	Actividade 6: Uso responsable da IA. Avaliación individual.	6 sesións	T1   Outubro
		<b>Tipo:</b> Outro	
<p>Esta Actividade 6 ten lugar de forma lóxica despois do último apartado de práctica da Actividade 5, que pide instalar Cursor (IDE axéntico) e empregar certos <i>prompts</i> de forma crítica. Ponse o acento en como usar a IA de forma intelixente centrándose na aproximación <i>human-in-the-loop</i><sup>5</sup>, fortalecendo a competencia do alumnado para xulgar a súa propia</p>			

<sup>5</sup> <https://www.ibm.com/think/topics/human-in-the-loop>

aprendizaxe e entender o “*trade-off*” entre completar un proceso máis rápido grazas á IA e a peor interiorización dos aspectos do propio proceso.

O docente, ao inicio dos dous primeiros días desta actividade (21 e 23 de outubro) exporá dous problemas a resolver empregando IA e buscará a discusión e o debate na clase. Daranse orientacións para facelo e acentuarase o espírito crítico, a observación das limitacións da IA e as implicacións de seguridade.

Despois, adicarase o resto de cada sesión a que os alumnos e alumnas deseñen individualmente unha estratexia e elaboren e refinen *prompts* para abordar a tarefa.

Os dous últimos días (tres sesións repartidas entre o 28 e 30 de outubro) lévase a cabo unha avaliación de xeito individualizado. O docente empregará unha lista de cotexo para revisar o traballo da Actividade 5 e unha rúbrica de avaliación dos *prompts* da Actividade 6.

<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA2.3. [...] automatizouse o contorno de desenvolvemento.		Rúbrica de avaliación	40%
<b>Contidos</b>	BC2.1. Funcións dun contorno de desenvolvemento.		
	BC2.3. Ferramentas e asistentes do contorno.		
	BC2.9. Características dos contornos de desenvolvemento.		
<b>Tarefas dos estudantes</b>			
Participación nas discusións introdutorias das dúas primeiras sesións			
Traballo autónomo elaborando, refinando e rexistrando os <i>prompts</i> das tarefas			
Facilitamento da avaliación individualizada de cada estudante no seu posto			
<b>Tarefas do docente</b>			
Discusión introdutoria nas dúas primeiras sesións			
Plantexamento de tarefas e orientacións sobre como abordalas, nas dúas primeiras sesións			
Acompañamento no traballo individual			
Avaliación individualizada das Actividades 5 e 6			
<b>Obxectivos específicos</b>			
Incidir no uso responsable da IA como ferramenta para adquirir coñecementos (e non para substituír ao estudante no proceso de aprendizaxe), comprendendo que o uso para a produción conleva unha perda na calidade da interiorización.			
<b>Resultados ou produtos</b>	Reflexións críticas escritas polo alumno ou alumna en relación á IA		
	Avaliación mediante lista de cotexo da Actividade 5		
	Avaliación mediante rúbrica de avaliación da Actividade 6		
<b>Recursos específicos</b>			
N/A			

## 5.4. Unidade Didáctica 4: Introducción a control de versións

UD4	Actividade 7: Instalación e uso básico de Git	5 sesións	T1   Novembro
		<b>Tipo:</b> C2 (ver Apartado 3)	
<p>A primeira sesión arranca coa introdución teórica a que é o sistema de control de versións Git e como funciona: SHA dun “commit”, “staging area”, “push” e “pull”, etc. Despois da introdución teórica, no proxector da aula o docente mostra o uso práctico dos conceptos vistos.</p>			
<p>As catro seguintes sesións consisten nunha práctica de traballo autónomo e guiado no que usuario creará un repositorio con “git init” e realizará diversas operacións sobre o mesmo, desde liña de comandos e desde VSCode. Deberá, máis tarde, subilo a GitHub (creando unha conta senón ten unha) e o docente fará de facilitador e orientador durante a práctica.</p>			
<p>A avaliación desta actividade lévase a cabo despois do remate, revisando o enlace ao repositorio de GitHub que cada estudante entregou e valorando mediante unha rúbrica de observación o traballo feito polo estudante.</p>			
<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA4.6. Realizouse o control de versións [...]		Rúbrica do traballo no repositorio	100%
<b>Contidos</b>	BC4.4. Control de versións. Estrutura [...]. Repositorio. Ferramentas [...]		
	BC4.5. Clientes para control de versións. Integración no contorno [...]		
<b>Tarefas dos estudantes</b>			
Participación na primeira sesión teórica			
Traballo autónomo guiado na práctica			
<b>Tarefas do docente</b>			
Exposición teórica apoiada no encerado e co ordenador no proxector na primeira sesión			
Acompañamento activo do alumnado no traballo autónomo			
Avaliación do traballo nos repositorios mediante rúbrica			
Verificación, durante as sesións de traballo, da autoría de cada estudante			
<b>Obxectivos específicos</b>			
<p>Introdución á base teórica de Git, entendido tanto dende a súa versión máis simple desde liña de comandos coma integrado nun IDE. O entendemento de Git desde o uso no CLI é moi importante para manexarse con soltura ante conflitos e problemas habituais. Ademais, a pesar de que existen moitos cambios no ámbito produtivo dende fai 5 anos ou menos, o control de versións non perdeu importancia xa que garante o control do código, facilita a</p>			

colaboración e revisión e serve para auditalo	
<b>Resultados ou produtos</b>	Repositorio en GitHub entregado por cada estudante
	Avaliación, mediante rúbrica, de cada repositorio
<b>Recursos específicos</b>	
Práctica: <a href="https://pd-contornos-ruben.org/assets/UD4_Actividade7_Git.pdf">https://pd-contornos-ruben.org/assets/UD4_Actividade7_Git.pdf</a>	

## 5.5. Unidade Didáctica 5: Colaboración en Git

UD5	Actividade 8: Traballo colaborativo en Git. Ramas	6 sesións	T1   Novembro
		<b>Tipo:</b> Outro	
<p>No principio da primeira sesión fórmanse grupos de traballo de 3 ou 4 persoas, escollidos polo profesor ou profesora, priorizando a proximidade dos postos de estudantes e procurando non agrupar persoas que, en base á avaliación inicial do módulo e da reunión do 15 de outubro, teñan necesidades especiais.</p> <p>Ao principio de cada día faise unha introdución de 15-20 minutos sobre os obxectivos prácticos de cada grupo para dita sesión.</p> <p>O primeiro día, venres 13, só nunha sesión, despois de facer os grupos e da introdución, cada equipo só deberá crear un repositorio con ramas “main” e “develop”, que será publicado en GitHub polo <i>product owner</i> de cada equipo, quen tamén configurará os permisos de acceso do resto de membros.</p> <p>Nas dúas sesións do seguinte día, mércores 18, cada <i>developer</i> clonará o repositorio e creará dúas ramas con tarefas sinxelas, xerando tamén “pull requests” asociadas. Serán revisadas e fusionadas. Durante a execución da segunda “pull request”, o docente plantexará un cambio que obrigará a facer “commit” na rama antes da fusión. Non se traballará con “código”</p>			

realmente. Só con ficheiros de texto e imaxes.

A sesión do venres 20 repetiranse obxectivos similares, pero, plantexaráanse tarefas a implementar que terán conflitos e deberán ser resoltos. Diferenciarase a estratexia de “merge commit” e “rebase”.

No último día plantexaráanse ata 5 obxectivos máis avanzados: “reset”, “amend”, “stash”, “cherry-pick” e “rebase” interactivo con “squash”. Cada “developer” deberá enfocarse nun ou dous deles e cumprir obxectivos no repositorio, trasladando despois a súa aprendizaxe ao resto do equipo.

A avaliación desta actividade realízase mediante un exame final, tipo *test*, que se leva a cabo na última media hora desta última sesión.

<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA4.6. Realizouse o control de versións integrado no contorno [...]		Exame escrito	85%
<b>Contidos</b>	BC4.4. Control de versións. Estrutura [...]. Repositorio. Ferramentas [...]		
	BC4.5. Clientes para control de versións. Integración no contorno [...]		
<b>Tarefas dos estudantes</b>			
Colaboración en grupos de traballo na realización de ramas de código			
Colaboración en grupos de traballo na revisión e fusión doutras ramas			
Colaboración en grupos, explicando o funcionamento de comandos específicos			
Realización do exame escrito			
<b>Tarefas do docente</b>			
Exposición do marco teórico e tarefas a realizar para cada día da actividade			
Supervisión e orientacións no traballo levado a cabo			
Corrección do exame escrito			
<b>Obxectivos específicos</b>			
Aprendizaxe sobre o traballo colaborativo en Git e o proceso de revisión de código. Fomentar o espírito crítico e a importancia da revisión concienzuda do código, que agora trascende á revisión do feito por outra persoa. Tamén debemos revisar o que propón a IA			
<b>Resultados ou produtos</b>	Repositorio en GitHub co proxecto de cada equipo		
	Exame individual tipo <i>test</i> e/ou resposta curta		
<b>Recursos específicos</b>			
Exame tipo <i>test</i> e/ou resposta curta			

UD5	Actividade 9: Forensethón: Que <i>commit</i> rompeu a <i>app</i> ?	1 sesión	T1   Novembro
		<b>Tipo:</b> Outro	
<p>Esta actividade con carácter <i>gamificado</i> lévase a cabo antes dos exames do primeiro trimestre. O docente entregará o enlace a un repositorio Git cun proxecto Java con varios problemas. A continuación, dará unha breve introducción á ferramenta “git bisect” e pedirá a cada grupo da Actividade 8 que traballe de xeito colaborativo e autoorganizado en averiguar dende cando unha funcionalidade está rota na aplicación. Cada grupo recibe un problema diferente.</p> <p>O grupo debe, nunha sesión, facer “git bisect” ata atopar o <i>commit</i> responsable do problema, logo analizar o <i>commit</i> e identificar o problema e o autor. Sorprendentemente, o autor do <i>commit</i> erróneo será “ti@ti-o-fixeches.example.com”. Isto servirá para explicar a importancia de aceptar os erros propios, responsabilizarse deles, e valorar a comunicación asertiva.</p> <p>Tamén, como peche desta actividade, animarase aos estudantes a subir a GitHub os seus traballos e comprender a importancia disto na búsqueda de emprego no futuro. Desexaráselle sorte ao alumnado para a seguinte semana.</p>			
<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA4.6. [...] control de versións [...]		Lista de cotexo dos obxectivos grupais	15%
<b>Contidos</b>	BC4.4. Control de versións. Estrutura [...]. Repositorio. Ferramentas [...]		
	BC4.5. Clientes para control de versións. Integración no contorno [...]		
<b>Tarefas dos estudantes</b>			
Traballo non guiado, colaborativo e autoorganizado, analizando un repositorio Git			
<b>Tarefas do docente</b>			
Explicación inicial sobre “git bisect” e plantexamento do desafío			
Acompañamento e supervisión da participación de todos os membros dos equipos			
<b>Obxectivos específicos</b>			
Fomentar o espírito colaborativo e a motivación asociada a unha actividade de carácter extraordinario. Poñer en valor a ferramenta Git, comprendendo o uso real que se lle pode dar no eido profesional			
<b>Resultados ou produtos</b>	Control dos participantes na actividade		
	Control dos grupos que acharon o problema antes do final da sesión		
<b>Recursos específicos</b>			
Repositorio: <a href="https://github.com/very-serious-dev/shopping_list_manager">https://github.com/very-serious-dev/shopping_list_manager</a>			

## 5.6. Unidade Didáctica 6: Introducción á calidade *software*. Probas unitarias

UD6	Actividade 10: Verificación de programas. Conceptos básicos	2 sesións	T2   Decembro
		<b>Tipo:</b> A (ver Apartado 3)	
<p>Ao inicio do segundo trimestre, despois dunha breve conversa sobre o reencontro tras as vacacións de Nadal, presentarase a nova unidade didáctica e o docente lanzará preguntas coma: “Cando alguén paga por <i>software</i>, como se mide se está satisfeito co resultado? A xente paga por <i>software</i>? Como se proba o <i>software</i>? Que tipos de erros poden suceder?” Igual que ao inicio do curso, non se darán respostas directas por parte do docente para fomentar a inquietude e a participación.</p> <p>Despois, terá lugar a presentación <i>Wayground</i> e o alumnado traballará de xeito autónomo e supervisado activamente na elaboración dun esquema/resumo sobre tipos de requisitos, tipos de probas, e outros conceptos e características relacionados co tema. Dito resumo será entregado ao final da actividade e avaliado polo docente para a seguinte clase.</p>			
<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA3.1. Identificáronse os tipos de probas.		Rúbrica do resumo	20%
<b>Contidos</b>	BC3.2. Tipos de probas: funcionais, estruturais, regresión, etc.		
	BC3.7. Normas de calidade.		
<b>Tarefas dos estudantes</b>			
Participación en discusión introdutoria			
Participación activa en presentación <i>Wayground</i>			
Creación de esquema/resumo sobre tipos de probas			
<b>Tarefas do docente</b>			
Discusión introdutoria e presentación con <i>Wayground</i>			
Acompañamento no traballo autónomo			
Corrección de esquemas/resumos			
<b>Obxectivos específicos</b>			
Manter activas as orientacións sobre o uso responsable da IA e aprender, de xeito introdutorio, sobre o proceso de proba e aseguramento da calidade dos produtos <i>software</i> .			
<b>Resultados ou produtos</b>	Esquema ou resumo de cada estudante sobre tipos de probas e características		
	Nota de avaliación do esquema de cada estudante e correccións no mesmo		
<b>Recursos específicos</b>			
Presentación <a href="https://wayground.com/admin/presentation/69fc4b8ce8d9d33350dd009f">https://wayground.com/admin/presentation/69fc4b8ce8d9d33350dd009f</a>			
Dispositivos móbiles (estudantes)			

UD6	Actividade 11: Probas unitarias. Campionato de probas	8 sesións	T2   Dec. - Xaneiro
		<b>Tipo:</b> Outro	
<p>Tras entregar aos estudantes os esquemas da Actividade 10 con correccións, a primeira sesión (venres 11) desta actividade consiste nunha introdución teórica por parte do docente sobre os contidos a traballar: Probas unitarias, caixa branca e caixa negra, clases de equivalencia, análise de valores límite e cubrimento de código. Na seguinte sesión (mércores 16) o docente procederá a demostrar de xeito práctico montando un proxecto Maven que empregue junit [1] e JaCoCo que é unha proba @Test e como interpretar o informe de cobertura de código xerado por JaCoCo. A segunda sesión dese día, os estudantes clonarán un proxecto “Churrería” en GitHub con esqueletos de clases e métodos sen implementar.</p> <p>Ao final desta sesión, formanse parellas de “contrincantes” na primeira fase do “campionato de probas”. Os estudantes reciben especificacións das funcionalidades de código que deben ser probadas, e crean unha rama propia na que suben os seus @Test. Isto pode ser feito mércores, xoves, e a primeira metade da sesión do venres 18. Na segunda metade dese día, o docente fai “git push” da implementación do código e cada estudante fusiona iso coa súa rama (de tal xeito que agora sí se poden probar os @Test), e executa as súas probas. Despois, documentan os resultados nun breve documento dunha páxina que será avaliado, e comparan os seus resultados de cobertura e número de erros detectados cos do seu “contrincante”. Calqueira modificación nos casos de proba feitos que sexa posterior á hora de subida da implementación da “Churrería” na segunda sesión, descalificará ao participante.</p> <p>De cada parella, o gañador participará nunha sesión similar á volta de vacacións o venres 8 de xaneiro, sobre un repositorio “Taller”. Os 6-7 clasificados finais, recibindo o ánimo e expectación do resto de compañeiros e compañeiras, participarán na ronda final o mércores 13 de xaneiro co repositorio “Hotel turístico”. Os non clasificados poden participar de forma non oficial e opcional en cada ronda.</p> <p>O último día da actividade, venres 15, estará destinado á realización dun exame escrito sobre as Actividades 10 e 11.</p>			

En cada ronda do campionato, e especialmente na primeira, hai moitas peculiaridades do código a verificar que non están explícitamente relacionadas coas especificacións dadas e requiren pensamento lateral. Isto pretende fomentar o espírito crítico fronte a uso de ferramentas de IA, que non estarán prohibidas durante a actividade, pero si poden limitar a capacidade para revisar e cuestionar o código dun xeito máis profundo e non cinguido a especificacións. Por exemplo, na “Churrería” hai especificacións sobre os métodos `abrirChurrería()` e `pecharChurrería()` que deben devolver “false” se non funcionaron correctamente. As especificacións só indican que non se pode `pecharChurrería()` se a churrería xa está pechada, pero non falan doutros métodos `quentarAceite()`; e `deixarAceiteArrefriar()`; polo que debe verificarse tamén que o aceite non está quente antes de `pecharChurrería()`.

Aínda que non hai cuestións fora das especificacións dadas que estean ligadas á nota de avaliación recibida polo traballo na primeira ronda, isto recalca que o traballo de QA de análise de requisitos é moi importante.

<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA3.1. Identificáronse os tipos de probas.		Exame escrito	25%
CA3.6. Efectuáronse probas unitarias [...]		Exame escrito	25%
CA3.7. Executáronse probas automáticas.		Rúbrica do traballo da ronda “Churrería”	10%
		Rúbrica do traballo da ronda “Churrería”	10%
CA3.8. Documentáronse as incidencias [...]		Rúbrica do traballo da ronda “Churrería”	10%
<b>Contidos</b>	BC3.3. Procedementos e casos de proba.		
	BC3.5. Validacións		
	BC3.6. Probas de código: cubrimento, valores límite, clases de equivalencia, etc		
	BC3.8. Probas unitarias: ferramentas		
<b>Tarefas dos estudantes</b>			
Participación nas sesións teórico-prácticas iniciais			
Implementación e execución de probas <code>jUnit</code> , documentando resultados na primeira ronda			
Participación coma clasificado ou coma espectador, opinando e axudando nas outras rondas			
Realización do exame escrito			
<b>Tarefas do docente</b>			
Exposición teórica introdutoria co apoio do encerado			
Exposición práctica de <code>jUnit</code> e <code>JaCoCo</code> en <code>Maven</code>			
Axuda aos estudantes con problemas de configuración de <code>Maven</code>			

Plantexamento do campionato, subindo o código da implementación á metade de cada ronda	
Valoración mediante rúbrica do documento de resultados de probas de cada estudante	
Corrección dos exames da Actividade 10 e 11	
<b>Obxectivos específicos</b>	
<p>Adquisición de coñecementos específico de junit, JaCoCo e Maven, entendendo que hai outros contornos e ferramentas que cubren necesidades similares. Fomento do espírito crítico para probar código, que é moi aplicable ao uso crítico de ferramentas de IA, así coma promoción da participación en eventos de carácter competitivo ou divulgativo.</p> <p>Adicionalmente, xustificando polo momento do curso académico no que se desenvolve a actividade, preténdese manter ao alumnado interesado no módulo e que despois dos exames da primeira avaliación o interese dos estudantes se manteña activo, así coma facilitar que retomar as clases despois de Nadal sexa interesante e rápido.</p>	
<b>Resultados ou produtos</b>	Rama de código de cada estudante cos seus “tests” sobre o proxecto “Churrería”
	Documento descritivo de cada estudante dos erros detectados na “Churrería”
	Avaliación mediante rúbrica da calidade e completitude de dito documento
	Ramas de código con “tests” para a ronda “Taller”
	Ramas de código con “tests” para a ronda “Hotel turístico”
Exame individual tipo <i>test</i> e/ou resposta curta	
<b>Recursos específicos</b>	
Repositorio “Churrería”: <a href="https://github.com/very-serious-dev/contornos_churreria">https://github.com/very-serious-dev/contornos_churreria</a>	
Repositorio “Taller”: <a href="https://github.com/very-serious-dev/contornos_taller">https://github.com/very-serious-dev/contornos_taller</a>	
Repositorio “Hotel turístico”: <a href="https://github.com/very-serious-dev/contornos_hotel">https://github.com/very-serious-dev/contornos_hotel</a>	
Exame tipo <i>test</i> e/ou resposta curta	

## 5.7. Unidade Didáctica 7: Depuración

UD7	Actividade 12: Depuración	6 sesións	T2   Xaneiro
		<b>Tipo:</b> C1 (ver Apartado 3)	
<p>Esta actividade tipo C1 contempla unha primeira sesión de introdución teórico-práctica á ferramenta do depurador en IntelliJ IDEA para proxectos Java, empregando o proxector e abrindo un proxecto novo no que se establecen puntos de ruptura e se mostra o funcionamento de distintas ferramentas (“Step-Over”, “Step-Into”, “Evaluate Expression”, “Resume”,...). O resto de sesións o alumnado traballa autónomamente nunha práctica e o docente revisa un a un o traballo que van desempeñando, achegándose aos que precisen máis apoio, e resolvendo activamente dúbidas.</p> <p>A práctica ten unha sección 5 complementaria non avaliable, que invita aos estudantes que rematen antes do tempo previsto a afrontar desafíos máis complexos.</p> <p>Esta actividade, ao contrario do que se indica no Apartado 3, non contempla un exame escrito de avaliación.</p>			
<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA3.3. Identificáronse as ferramentas de depuración [...]		Rúbrica da práctica	30%
CA3.4. Utilizáronse [...] puntos de ruptura e seguimento.		Rúbrica da práctica	30%
CA3.5. [...] examinar e modificar o comportamento [...] na execución		Rúbrica da práctica	40%
<b>Contidos</b>	BC3.4. Mecanismos e ferramentas de depuración: puntos de ruptura, [...]		
	BC3.5. Validacións		
<b>Tarefas dos estudantes</b>			
Traballo autónomo sobre un documento con orientacións prácticas no ordenador			
<b>Tarefas do docente</b>			
Acompañamento no traballo autónomo e apoio específico			
Valoración das prácticas mediante rúbrica			
<b>Obxectivos específicos</b>			
Aprendizaxe das características comúns das ferramentas de depuración e como se empregan para avaliar e inspeccionar o código, entendendo que é indispensable como desenvolvedor/a			
<b>Resultados ou produtos</b>	Documento da práctica cuberto por cada estudante		
	Avaliación da práctica mediante rúbrica de avaliación		
<b>Recursos específicos</b>			
Práctica: <a href="https://pd-contornos-ruben.org/assets/UD7_Actividade12_Depuracion.pdf">https://pd-contornos-ruben.org/assets/UD7_Actividade12_Depuracion.pdf</a>			

## 5.8. Unidade Didáctica 8: Requisitos e casos de proba

UD8	Actividade 13: XadrezApp. Requisitos e casos de proba	4 sesións	T2   Febreiro
		<b>Tipo:</b> Outro	
<p>Ao inicio da actividade confórmanse grupos de catro (ou tres, de ser preciso) persoas que reciben estas orientacións: Dous actuarán coma “clientes” e pedirán un “produto” que idearán no momento e que xirará en torno a una <i>app</i> móbil de xadrez. Os outros dous actuarán coma “consultores” e tentarán definir os requisitos que lles piden. Despois da primeira sesión de traballo, a actividade páusase e o docente da unha introdución teórica expositiva a levantamento de requisitos, definición de casos de proba, historias de usuario, Gherkin, etc.</p> <p>O traballo reanúdase o seguinte venres 5. Xa cunhas directrices máis específicas, cada par de “consultores” de cada grupo continúa traballando en xerar un documento con casos de uso e requisitos funcionais. Á volta de carnaval, o seguinte venres 12, o <i>role-play</i> invírtese e os grupos cámbianse. As dúas persoas que foron “clientes” nun grupo, agora serán “consultores” e estarán emparelladas con dúas persoas diferentes.</p> <p>O resultado producido por cada persoa no seu rol de “consultor” será entregado e avaliado mediante unha rúbrica. Permítese que cada parella entregue o mesmo documento ou non. O traballo será devolto con comentarios da revisión do docente.</p>			
Criterios avaliación		Instrumento	UD
CA3.2. Definíronse casos de proba.		Rúbrica do documento producido	100%
Contidos	BC3.1. Planificación de probas.		
	BC3.3. Procedementos e casos de proba.		
Tarefas dos estudantes			
Participar nunha actividade grupal coma “cliente” que explica como quere unha <i>app</i> de xadrez			
Participar nunha actividade grupal coma “consultor” que define casos de uso e requisitos			
Tarefas do docente			
Dar orientacións na actividade inicial			
Explicación expositiva sobre contidos na segunda sesión do primeiro día			
Guiar e orientar na parte final do traballo grupal			
Avaliar as entregas empregando a rúbrica			
Obxectivos específicos			
Introducir ferramentas e marcos de traballo que amplíen o entendemento de que son as probas. Comprender que non só validan o código, senón que forman parte da relación co			

cliente. Fomentar o clima participativo e dinámico, especialmente despois dunha Actividade 12 eminentemente práctica e outro bloque final de actividades do segundo trimestre que tamén ten, principalmente, carga de traballo individual.

<b>Resultados ou produtos</b>	Documento de definición de casos de proba e requisitos de cada parella
	Avaliación do mesmo mediante rúbrica
<b>Recursos específicos</b>	
N/A	

### 5.9. Unidade Didáctica 9: Probas en dispositivos móbiles

UD9	Actividade 14: XadrezApp. Probas Android con Espresso	6 sesións	T2   Febreiro
		<b>Tipo:</b> B2 (ver Apartado 3)	
<p>Tras o traballo previo do docente creando as contas no GitLab autoxestionado e aleatorizando as tarefas (coma se describe no Apartado 3), a primeira sesión lévase a cabo unha introdución práctica co ordenador e o proxector sobre cómo se instala a dependencia Espresso en Android Studio e prográmanse un par de probas sinxelas. Despois, cada alumno ou alumna recibe instrucións para acceder a GitLab e programar casos de proba con Espresso sobre un repositorio persoal que contén o código de XadrezApp. O código é igual para cada estudante, pero os requisitos que se piden para os casos de proba non son os mesmos, polo que algúns obterán distintos resultados nas súas probas ca outros. Durante as sesións de traballo nestes enunciados prácticos (<i>issues</i>) en GitLab, o docente complementará o traballo con breves sesións expositivas prácticas, resolverá dúbidas ou bloqueos e prestará especial atención a aqueles estudantes que teñan máis dificultades.</p>			
<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA3.2. Definíronse casos de proba.		Resultados da execución dos <i>tests</i>	50%
CA3.7. Executáronse probas automáticas.		Resultados da execución dos <i>tests</i>	50%

<b>Contidos</b>	BC3.1. Planificación de probas.
	BC3.3. Procedementos e casos de proba.
	BC3.5. Validacións
	BC3.9. Automatización e documentación das probas.
<b>Tarefas dos estudantes</b>	
Programación de casos de proba Espresso nun repositorio propio de GitLab con “XadrezApp”	
<b>Tarefas do docente</b>	
Preparación dos usuarios e repositorios para os estudantes nun GitLab autoxestionado	
Sesións expositivas prácticas de Espresso	
Resolución de dúbidas e acompañamento activo durante a parte práctica	
Corrección do traballo no repositorio mediante execución de probas automáticas	
<b>Obxectivos específicos</b>	
Achegarse ás ferramentas típicas dun entorno produtivo, empregando un proxecto en GitLab coma marco de traballo para a creación de probas con Espresso en Android. Esta metodoloxía de actividade tipo B2 ten unha forte relación co que podería ser un entorno produtivo real, e será reutilizada durante o segundo curso do ciclo.	
<b>Resultados ou produtos</b>	Histórico de <i>commits</i> no repositorio de cada estudante
	Resultado da execución dos <i>tests</i> programados por cada estudante
<b>Recursos específicos</b>	
PATACA: <a href="https://gitlab.com/ruben.montero/developer-course-automation">https://gitlab.com/ruben.montero/developer-course-automation</a>	

## 5.10. Unidade Didáctica 10: Probas en páxinas web

UD10	Actividade 15: Probas web con Cypress	5 sesións	T2   Marzo
		<b>Tipo:</b> B2 (ver Apartado 3)	
<p>Continúase a actividade anterior cunha nova práctica sobre probas en páxinas web e o mesmo fluxo de traballo retómase esta vez empregando a librería Cypress para realización de probas <i>end-to-end</i>. O documento da práctica propón a realización de probas sobre páxinas reais e anima ao uso da IA para o desenvolvemento da meirande parte do código necesario para requisitos de probas, pero invita á reflexión crítica sobre que probas son necesarias e por que.</p>			

A primeira sesión, antes de arrancar coa práctica, o docente empregará o seu ordenador e o proxector da aula para introducir a ferramenta Cypress nunha sesión expositiva práctica, e complementará con outras breves sesións similares máis adiante. O resto do tempo, actuará coma orientador do traballo dos estudantes, mantendo a atención neles e fomentando a reflexión sobre o que fan, inquirindo activamente cales son os problemas que atopan e resolvendo dúbidas.

A avaliación desta actividade tipo B2 realízase igual que na Actividade 14 anterior, sobre os *tests* subidos por cada estudante ao seu repositorio.

<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA3.2. Definíronse casos de proba.		Resultados da execución dos <i>tests</i>	50%
CA3.7. Executáronse probas automáticas.		Resultados da execución dos <i>tests</i>	50%
<b>Contidos</b>	BC3.1. Planificación de probas.		
	BC3.3. Procedementos e casos de proba.		
	BC3.5. Validacións		
	BC3.9. Automatización e documentación das probas.		
<b>Tarefas dos estudantes</b>			
Programación de casos de proba Cypress nun repositorio propio de GitLab			
<b>Tarefas do docente</b>			
Preparación dos repositorios para os estudantes nun GitLab autoxestionado			
Sesións expositivas prácticas de Cypress			
Resolución de dúbidas e acompañamento activo durante a parte práctica			
Corrección do traballo no repositorio mediante execución de probas automáticas			
<b>Obxectivos específicos</b>			
Continuar familiarizándose con ferramentas típicas dun entorno produtivo, empregando un proxecto en GitLab coma marco de traballo das probas Cypress. Este actividade terá un especial valor para aqueles estudantes que fagan 2º de Desenvolvemento de Aplicacións ben porque cambien de ciclo ou porque fagan a dobre titulación.			
<b>Resultados ou produtos</b>	Histórico de <i>commits</i> no repositorio de cada estudante		
	Resultado da execución dos <i>tests</i> programados por cada estudante		
<b>Recursos específicos</b>			
PATACA: <a href="https://gitlab.com/ruben.montero/developer-course-automation">https://gitlab.com/ruben.montero/developer-course-automation</a>			

## 5.11. Unidade Didáctica 11: UML. Conceptos básicos

UD11	Actividade 16: Introducción a diagramas de clase UML	2 sesións	T3   Marzo
		<b>Tipo:</b> A (ver Apartado 3)	
<p>Ao inicio da actividade tense unha breve conversa para “aterriar” despois das vacacións e procédese coa introdutoria sobre UML. Despois ten lugar a presentación <i>Wayground</i>, na que a última diapositiva contén indicacións para a elaboración dun esquema/resumo sobre conceptos de diagramas de clase UML, que será avaliado polo docente e devolto a cada estudante con correccións. O alumnado recibirá o aviso de que para esta actividade e para a seguinte haberá un exame sorpresa posterior.</p>			
<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA5.1. Identificáronse os conceptos básicos da POO		Rúbrica do resumo	10%
CA5.2. Recoñecéronse métodos de análise e modelaxe [...]		Rúbrica do resumo	5%
CA5.4. Interpretouse o significado de diagramas de clases.		Rúbrica do resumo	10%
<b>Contidos</b>	BC5.1. Clases: atributos, métodos e visibilidade.		
	BC5.2. Métodos de análise e modelaxe.		
	BC5.3. Obxectos: instanciación		
	BC5.4. Relacións: herdanza, composición e agregación.		
	BC5.5. UML. Diagramas estruturais.		
	BC5.6. Notación dos diagramas de clases: interpretación.		
<b>Tarefas dos estudantes</b>			
Participación en discusión introdutoria			
Participación activa en presentación <i>Wayground</i>			
Creación de esquema/resumo sobre diagramas de clase UML			
<b>Tarefas do docente</b>			
Discusión introdutoria e presentación con <i>Wayground</i>			
Acompañamento no traballo autónomo			
Corrección de esquemas/resumos			
<b>Obxectivos específicos</b>			
Entender a versatilidade e universalidade de UML, e valorar a súa utilidade en distintos contextos e a súa prevalencia ao longo dos anos.			
<b>Resultados ou produtos</b>	Esquema ou resumo de cada estudante sobre diagramas de clase UML		
	Nota de avaliación do esquema de cada estudante e correccións no mesmo		
<b>Recursos específicos</b>			
Presentación <a href="https://wayground.com/admin/presentation/6a1553c2139b944a72066583">https://wayground.com/admin/presentation/6a1553c2139b944a72066583</a>			
Dispositivos móbiles (estudantes)			

UD11	Actividade 17: Introducción a diagramas de estado e secuencia	3 sesións	T3   Abril
		<b>Tipo:</b> A (ver Apartado 3)	
<p>Actividade tipo A análoga á anterior, esta vez sobre diagramas de estado e secuencia UML. Despois da presentación <i>Wayground</i> o traballo autónomo será seguido e apoiado activamente polo docente. O último día da actividade (7 de abril) cada estudante entregará o seu esquema/resumo e recibírao o venres 9 con revisións e correccións. Poderá empregalo para preparar, xunto ao resultado da Actividade 16, o exame sorpresa que o docente anunciará que terá lugar “algún día do mes de abril”.</p>			
Criterios avaliación		Instrumento	UD
CA6.1. Identificáronse os tipos de diagramas de comportamento.		Rúbrica do resumo	5%
CA6.4. Elaboráronse diagramas de interacción sinxelos.		Rúbrica do resumo	10%
CA6.8. Formuláronse diagramas de estados sinxelos.		Rúbrica do resumo	10%
		Exame “sorpresa”	20%
CA5.1. Identificáronse os conceptos básicos da POO		Exame “sorpresa”	15%
CA5.4. Interpretouse o significado de diagramas de clases.		Exame “sorpresa”	15%
Tarefas dos estudantes			
Participación en discusión introdutoria			
Participación activa en presentación <i>Wayground</i>			
Creación de esquema/resumo sobre os diagramas de estado e secuencia			
Tarefas do docente			
Discusión introdutoria e presentación con <i>Wayground</i>			
Acompañamento no traballo autónomo			
Corrección de esquemas/resumos			
Obxectivos específicos			
<p>Avanzar na aprendizaxe de UML e comprender a súa versatilidade para representar cuestións do mundo real máis aló do código: Procesos, actividades, etc. Ademais, o “exame sorpresa” pretende reafirmar no alumnado a realidade de que o traballo para aprender é constante e non debe cingirse a estudar e reproducir contidos un día particular.</p>			
Resultados ou produtos	Esquema ou resumo de cada estudante sobre diagramas de clase UML		
	Nota de avaliación do esquema de cada estudante e correccións no mesmo		
	Exame escrito “sorpresa” levado a cabo durante o mes de abril		
Recursos específicos			
Presentación <a href="https://wayground.com/admin/presentation/6a15634cf1dbf62a483e462f">https://wayground.com/admin/presentation/6a15634cf1dbf62a483e462f</a>			
Dispositivos móbiles (estudantes)			
Exame tipo <i>test</i> e/ou resposta curta da Actividade 16 e 17			

## 5.12. Unidade Didáctica 12: UML. Diagramas de clase

UD12	Actividade 18: UML – Diagramas de clase	6 sesións	T3   Abril
		<b>Tipo:</b> B1 (ver Apartado 3)	
<p>Esta actividade tipo B1 será similar ás Actividades 14 e 15 que xa foron levadas a cabo polo alumnado o trimestre anterior, pero nesta ocasión os <i>tests</i> de código non serán o obxectivo, senón o medio de validar o seu traballo. Concretamente, cada estudante recibe tarefas aleatorizadas na súa conta do GitLab e debe implementar clases daccordo a diagramas UML (diferentes para cada un) e diagramas, habendo <i>tests</i> jUnit para validar dito código.</p> <p>Contémplanse dúas variacións en función do desempeño global dos estudantes. No primeiro caso, se os estudantes obtiveron un desempeño xeral bo ou moi bo, e exhibiron na súa maioría un nivel alto de responsabilidade e non “facer trampas” durante o curso, entón cada estudante disporá dos <i>tests</i> jUnit de avaliación dende o inicio da actividade. No segundo caso, os <i>tests</i> non estarán dispoñibles dende o inicio, para evitar a “trampa” de copiar o código dun compañeiro e modificalo só naqueles puntos onde os <i>tests</i> fallen. Os <i>tests</i> individualizados só se subirán ao repositorio despois da verificación de que dito estudante completou todas as tarefas. A actividade avalíase coa execución dos <i>tests</i> automáticos por parte do docente.</p>			
Criterios avaliación		Instrumento	UD
CA5.3. [...] ferramentas para [...] diagramas de clases.		Resultados dos <i>tests</i>	10%
CA5.4. Interpretouse o significado de diagramas de clases.		Resultados dos <i>tests</i>	10%
CA5.5. Instalouse o módulo [...] de diagramas de clases.		Resultados dos <i>tests</i>	10%
CA5.6. [...] diagramas de clases a partir das especificacións.		Resultados dos <i>tests</i>	10%
CA5.7. Xerouse código a partir dun diagrama de clases.		Resultados dos <i>tests</i>	50%
CA5.8. [...] diagrama de clases mediante enxeñaría inversa.		Resultados dos <i>tests</i>	10%
Contidos	BC5.1. Clases: atributos, métodos e visibilidade.		
	BC5.4. Relacións: herdanza, composición e agregación.		
	BC5.5. UML. Diagramas estruturais.		
	BC5.6. Ferramentas de deseño de diagramas. Módulos [...]		
	BC5.7. Creación de diagramas de clases.		
	BC5.8. Xeración de código desde o diagrama de clases.		
	BC5.9. Obtención do diagrama de clases a partir do código.		
Tarefas dos estudantes			
Programación de clases Java e diagramas nun repositorio propio de GitLab			

<b>Tarefas do docente</b>	
Preparación dos repositorios para os estudantes nun GitLab autoxestionado	
Sesiões expositivas teórico-prácticas complementarias	
Resolución de dúbidas e acompañamento activo durante a parte práctica	
Corrección do traballo no repositorio mediante execución de probas automáticas	
<b>Obxectivos específicos</b>	
<p>Afondar no coñecemento e soltura con UML que iniciouse na Actividade 16 así coma aproximarse a unha metodoloxía de traballo <i>test-driven development</i>. Tamén, a esta actividade denominaráselle intencionadamente “<i>sprint</i>”, para que se vincule coa metodoloxía Scrum estudada na Actividade 3 e así (1) reforzar a conexión do aprendido e a idea de que o coñecemento non é un ente “illado”, (2) achegar ao alumnado a metodoloxías actuais do eido produtivo e (3) incrementar a motivación e o interese polo que se fai</p>	
<b>Resultados ou produtos</b>	Histórico de <i>commits</i> no repositorio de cada estudante
	Resultado da execución dos <i>tests</i> para cada estudante
<b>Recursos específicos</b>	
PATACA: <a href="https://gitlab.com/ruben.montero/developer-course-automation">https://gitlab.com/ruben.montero/developer-course-automation</a>	

### 5.13. Unidade Didáctica 13: UML. Diagramas de comportamento

UD13	Actividade 19: UML – Diagramas de comportamento	6 sesións	T3   Abril - Maio
		<b>Tipo: B1 (ver Apartado 3)</b>	
<p>Esta actividade será similar a levada a cabo a mediados de abril, recordando que o exame “sorpresa” das Actividades 16 e 17 terá lugar nestas semanas.</p> <p>Nesta actividade o traballo é autónomo e supervisado activamente polo docente, quen se enfoca en contribuir á motivación de cada individuo e escoitar as súas inquedanzas, especialmente neste período en que se van a ir achegando os exames finais. Cada estudante traballará nas súas tarefas aleatorizadas que piden a implementación de código a partires da</p>			

análise e comprensión de diversos tipos de diagramas, e viceversa. Contémplanse tamén as dúas opcións descritas na Actividade 18 con respecto á entrega dos *tests* automáticos ao alumnado. A actividade avalíase coa execución dos *tests* automáticos por parte do docente.

<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA6.2. [...] significado dos diagramas de casos de uso.		Resultados dos <i>tests</i>	5%
CA6.3. Interpretáronse diagramas de interacción.		Resultados dos <i>tests</i>	20%
CA6.4. Elaboráronse diagramas de interacción sinxelos.		Resultados dos <i>tests</i>	10%
CA6.5. Interpretouse o significado de diagramas de actividades.		Resultados dos <i>tests</i>	20%
CA6.6. Elaboráronse diagramas de actividades sinxelos.		Resultados dos <i>tests</i>	10%
CA6.7. Interpretáronse diagramas de estados.		Resultados dos <i>tests</i>	20%
CA6.8. Formuláronse diagramas de estados sinxelos.		Resultados dos <i>tests</i>	10%
<b>Contidos</b>	BC6.2. Diagramas de casos de uso: actores, escenario e relación de comunicación.		
	BC6.3. Diagramas de interacción: tipos (diagramas de secuencia e comunicación)		
	BC6.4. Diagramas de secuencia.		
	BC6.5. Diagramas de comunicación.		
	BC6.6. Diagramas de actividades.		
	BC6.7. Diagramas de estados.		
<b>Tarefas dos estudantes</b>			
Programación Java e realización de diagramas nun repositorio propio de GitLab			
<b>Tarefas do docente</b>			
Preparación dos repositorios para os estudantes nun GitLab autoxestionado			
Sesiões expositivas teórico-prácticas complementarias			
Resolución de dúbidas e acompañamento activo durante a parte práctica			
Corrección do traballo no repositorio mediante execución de probas automáticas			
<b>Obxectivos específicos</b>			
Profundización nos coñecementos adquiridos na Actividade 17 e continuación da metodoloxía da Actividade 18 seguindo as mesmas xustificacións			
<b>Resultados ou produtos</b>	Histórico de <i>commits</i> no repositorio de cada estudante		
	Resultado da execución dos <i>tests</i> para cada estudante		
<b>Recursos específicos</b>			
PATACA: <a href="https://gitlab.com/ruben.montero/developer-course-automation">https://gitlab.com/ruben.montero/developer-course-automation</a>			

## 5.14. Unidade Didáctica 14: UML e código

UD14	Actividade 20: CLI Java a partir de diagrama de estados	4 sesións	T3   Maio
		<b>Tipo:</b> Outro	
<p>Esta penúltima actividade do curso pretende agregar distintas aprendizaxes feitas ao longo do ano, non só no módulo de Contornos de Desenvolvemento senón tamén en Programación e, deste xeito, contribuir á autopercepción positiva do que se aprendeu e mentalizar aos estudantes de que o traballo continuará o seguinte curso.</p>			
<p>A primeira sesión do venres 7 cada estudante ideará un diagrama de estados para unha situación cotiá (preparar unha comida, viaxar en transporte público,...) con entre 6 e 10 estados. Nas sesións da seguinte semana, cada estudante recibirá aleatoriamente un diagrama dun compañeiro ou compañeira. O docente poderá reflexionar durante a fin de semana sobre os mesmos para tentar distribuilos do xeito máis xusto posible, de tal maneira que estudantes con necesidades educativas especiais ou peores resultados durante o curso non reciban os diagramas máis complexos. Cada estudante implementará unha CLI Java a partires do diagrama recibido respectando unhas especificacións dadas. A avaliación será levada a cabo polo docente despois da entrega do 14 de maio, que cada estudante debe facer nun repositorio propio de GitHub. Clonará e probará cada CLI, verificándoo e avaliándoo mediante unha rúbrica da consecución do obxectivo e da calidade do código.</p>			
<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA6.7. Interpretáronse diagramas de estados.		Rúbrica da CLI	100%
<b>Contidos</b>	BC6.7. Diagramas de estados.		
<b>Tarefas dos estudantes</b>			
Elaboración dun diagrama de estados sobre un escenario do mundo real			
Implementación dun CLI Java a partires do diagrama dun compañeiro			
<b>Tarefas do docente</b>			
Sesión introdutoria na que se darán exemplos de diagrama para a tarefa plantexada			
Acompañamento activo e orientacións durante as sesións de traballo			
Avaliación da CLI de cada estudante mediante rúbrica			
<b>Obxectivos específicos</b>			
Do mesmo xeito que na Actividade 8 e na Actividade 13 os estudantes traballaban de xeito colaborativo, nesta actividade tamén preténdese fomentar o espírito de compañerismo e a			

mellora da capacidade de autocrítica: Cando o traballo é avaliado polo docente hai unha certa “desconexión” da aceptación dos erros cometidos en moitos casos, mentres que cando se xulga o traballo dun compañeiro/a (neste caso, o diagrama de estados) é máis fácil aplicar un sentido crítico que de forma indirecta tamén serve para un mesmo. Quérese tamén reforzar os coñecementos de UML e as habilidades de programación nesta recta final do curso

<b>Resultados ou produtos</b>	Diagrama de estados (non avaliado) feito por cada estudante
	Histórico de <i>commits</i> e código resultante do CLI no GitHub de cada alumno
<b>Recursos específicos</b>	
N/A	

### 5.15. Unidade Didáctica 15: Patróns de deseño e refactorización

UD15	Actividade 21: Refactorización e optimización	6 sesións	T3   Maio
		<b>Tipo:</b> B1 (ver Apartado 3)	
<p>Esta actividade tipo B1 pon o broche final ao curso, e está especialmente pensada para unha duración de 2 semanas nas que potencialmente, algún alumno ou alumna, pode ir a facer 80h de formación en empresa de 1º curso tal e como se contempla na normativa. Polo tanto, o traballo da actividade está especialmente aliñado co que pode observar e practicar o alumnado que non estea no centro senón que pase o seu tempo na empresa e sexa avaliado de xeito separado dacordo co seu programa formativo de empresa.</p> <p>Para os que permanezan no centro educativo, que previsiblemente serán a maioría ou todos os estudantes, o traballo lévase a cabo de xeito autónomo nun proxecto con tarefas aleatorizadas de xeito individual no que se piden facer diversas refactorizacións a código xa existente e practícanse algúns patróns de deseño (<i>singleton</i> e <i>factoría</i>).</p> <p>O docente resolverá as dúbidas e plantexará tarefas adicionais e de repaso a aqueles</p>			

estudiantes que rematen as tarefas antes da data de entrega. Posteriormente, valorará o traballo mediante a execución de *tests* automáticos.

<b>Criterios avaliación</b>		<b>Instrumento</b>	<b>UD</b>
CA4.1. Identificáronse os patróns de refactorización máis usuais.		Resultados dos <i>tests</i>	10%
CA4.2. Elaboráronse as probas asociadas á refactorización.		Resultados dos <i>tests</i>	15%
CA4.3. Revisouse o código fonte usando un analizador de código		Resultados dos <i>tests</i>	10%
CA4.4. [...] configuración dun analizador de código.		Resultados dos <i>tests</i>	15%
CA4.5. Aplicáronse patróns de refactorización [...]		Resultados dos <i>tests</i>	30%
CA4.7. [...] ferramentas [...] para documentar as clases.		Resultados dos <i>tests</i>	20%
<b>Contidos</b>	BC4.1. Refactorización: concepto, limitacións e patróns máis usuais.		
	BC4.2. Refactorización e probas. Ferramentas de axuda á refactorización.		
	BC4.3. Uso e configuración de analizadores de código.		
	BC4.6. Documentación. Uso de comentarios. Alternativas.		
<b>Tarefas dos estudantes</b>			
Programación Java e realización de diagramas nun repositorio propio de GitLab			
<b>Tarefas do docente</b>			
Preparación dos repositorios para os estudantes nun GitLab autoxestionado			
Sesións expositivas teórico-prácticas complementarias			
Resolución de dúbidas e acompañamento activo durante a parte práctica			
Corrección do traballo no repositorio mediante execución de probas automáticas			
<b>Obxectivos específicos</b>			
Coñecer máis en profundidade as ferramentas propias dos IDEs para a refactorización do código así coma patróns habituais, tanto de refactorización coma de deseño. Distinción entre cambios efectuados por IDEs axénticos e cambios sen asistencia intelixente, valorando a importancia de Git para xulgar críticamente as modificacións levadas a cabo en ambos casos.			
<b>Resultados ou produtos</b>	Histórico de <i>commits</i> no repositorio de cada estudante		
	Resultado da execución dos <i>tests</i> para cada estudante		
<b>Recursos específicos</b>			
PATACA: <a href="https://gitlab.com/ruben.montero/developer-course-automation">https://gitlab.com/ruben.montero/developer-course-automation</a>			

## 6. Procedementos e criterios de cualificación

### 6.1. Xustificación e explicación do proceso de avaliación

A avaliación e a cualificación soen confundirse na práctica. Esta confusión débese principalmente á tradición cualificadora da avaliación. A avaliación abrangue moitos aspectos relacionados con reunir información, valorala e tomar decisións para mellorar a aprendizaxe, mentres que a cualificación límitase a expresar numericamente o resultado da avaliación [2].

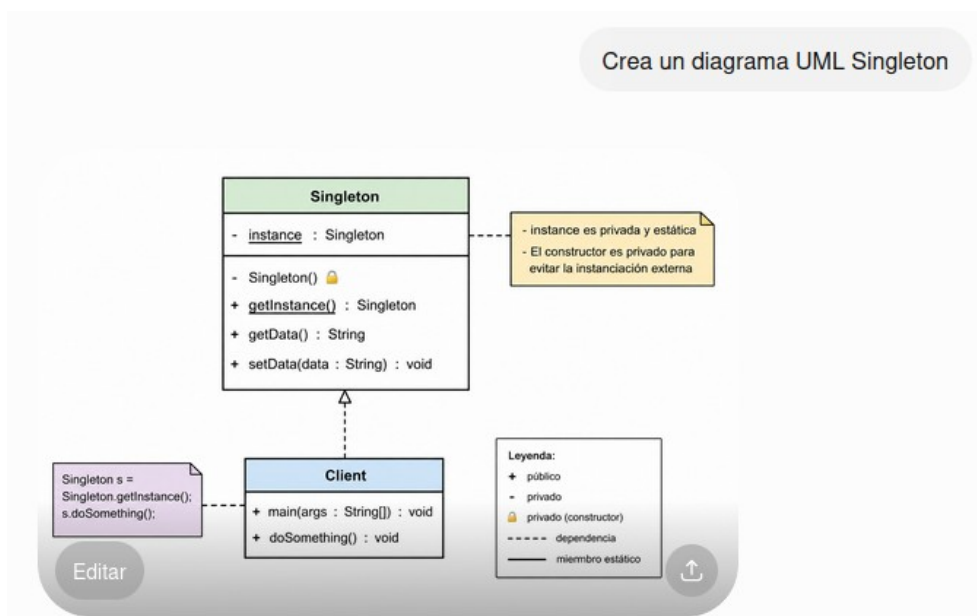
Aínda que o obxectivo deste apartado é detallar os procesos utilizados para cualificar a cada estudante, explicaremos e xustificaremos todos os procesos de avaliación. A avaliación é esencial para a aprendizaxe, xa que é a expresión observable da consecución dos obxectivos propostos e ten unha función autorreguladora e metacognitiva [3]. Un exemplo sinxelo é a avaliación da UD3. O docente sentarase a carón de cada estudante (durante as 3 sesións desa semana) para revisar, cotexar, comentar e discutir o traballo individual. Esta conversa non só persigue o obxectivo de obter unha nota numérica, senón tamén escoitar activamente cales foron as dificultades atopadas durante as actividades e entregar valoracións e consellos que axuden á mellora de cada estudante no seu proceso de aprendizaxe. Moitas veces non será nin sequera de forma enunciativa, senón con preguntas: “Por que che custou avanzar na última parte da práctica?” “Que cousas cres que che fan perder a concentración e como repercute isto en ti?” “Como te sintes cando estás bloqueado?” “Como manexas esa frustración?” Esta clase de avaliación continua, empática e positiva está presente en todo o curso no traballo do docente de seguimento e acompañamento activo.

Por outra banda, os novos LLM introducen un desafío á hora de avaliar. A aposta nesta programación consiste en:

- Ensinar de xeito temprano sobre o uso correcto e ético da IA e dar *feedback*, tamén o antes posible, das posibles consecuencias negativas do seu uso (e abuso).

- Facer exames escritos, sen acceso a ferramentas de IA, naquelas unidades onde é relativamente sinxelo empregar IA de forma desleal para saltarse o proceso de aprendizaxe.

Con respecto a este último punto, atopo un conflito persoal en canto á miña compartición co exposto por Francisco Bellas, poñente no Congreso de Intelixencia Artificial na Educación de 2024 en Santiago de Compostela, quen indicou que “a nova era da IA obriga a cada docente a analizar as súas asignaturas e reinventalas na medida do posible asegurando que estas novas ferramentas só se empregan de xeito que melloren a aprendizaxe.” Sen embargo, en moitos módulos, e en Contornos de Desenvolvemento particularmente, isto supón un gran desafío. Por que? Vexamos a continuación:



...e agora preguntémonos: Como se pode facer unha avaliación orientada a resultados de aprendizaxe, se un resultado de aprendizaxe é formular patróns de deseño en UML pero isto pode facerse a golpe de *prompt*?

A base de coñecemento segue a ser algo importante, digamos fundamental, para o estudante (véxase Apartado 2). Polo tanto, para usar a IA no noso favor, na meirande parte das actividades este uso está activamente dirixido e aconsellado (véxase Apartado 3). Pero na avaliación, a pesar do conflito persoal, haberá unha gran parte de unidades didácticas onde se

empreguen probas escritas onde o acceso á IA non está permitido. Isto non é sinónimo de que se vaian a facer probas escritas “desaxeitadas”. Ditas probas non van estar enfocadas en que o estudante lembre e escriba os contidos teóricos que aprendeu. As probas escritas deben medir a capacidade do estudante de integrar o aprendido con nova información plantexada nos enunciados. Por exemplo: Na Actividade 1 os estudantes investigan de xeito autónomo sobre tipos e características de linguaxes de programación. No exame escrito, o enunciado explica como funcionan as palabras “var” e “dynamic” na linguaxe C#, e pide unha valoración sobre se algunha delas introduce un comportamento dinámico nunha linguaxe de tipado estático coma C#. O alumno ou alumna non só aprende sobre esa característica de C# (o seu nivel de atención durante o exame será moi alto) senón que traballará mentalmente en integralo co que sabe.

O deseño das rúbricas de avaliación tamén é un punto clave: Son concisas, xustas e inequívocas, distribuindo a puntuaxe dun xeito racional e proporcional ao esforzo e ao grao de consecución dos resultados de aprendizaxe medidos a través dos criterios.

## 6.2. Cómputo das notas

As probas escritas son de tipo *test* e/ou resposta curta nos exames finais de unidade didáctica por cuestións de obxectividade e tempos de corrección. Tanto eses exames escritos coma os exames finais en cada trimestre terán un encabezado no que constarán os criterios de cualificación específicos de dito exame.

A nota resultante da cualificación de cada trimestre (véxanse Apartados 6.3, 6.4 e 6.5) redondéase<sup>6</sup> ao número natural máis próximo.

A nota final do curso é a media aritmética de cada unha das notas (con decimais, sen redondear) de cualificación de cada trimestre, redondeada<sup>5</sup>.

No caso de que algunha avaliación non sexa aprobada na primeira oportunidade, aplicaranse os criterios e procedementos de recuperación descritos no Apartado 7.

---

<sup>6</sup> En caso de equidistancia, redondéase ao número natural superior

### 6.3. Cualificación do primeiro trimestre (T1)

Avaliación continua						Exame final (T1)
	RAs*	Obxecto de avaliación	Instrumento	Peso na UD	Peso na av. cont.	
UD1. Linguaxes de programación. Ferramentas Java. JVM	RA1	Esquema/resumo	Rúbrica de avaliación	20%	4%	
		Traballo práctico	Rúbrica de avaliación	30%	6%	
			Exame escrito	50%	10%	
UD2. Fases de desenvolvemento. Uso do procesador, memoria, etc.	RA1	Esquema/resumo	Rúbrica de avaliación	20%	4%	
		Traballo práctico	Rúbrica de avaliación	30%	6%	
			Exame escrito	50%	10%	
UD3. IDEs. Uso responsable da IA	RA2	Traballo práctico (IDEs)	Lista de cotexo	60%	12%	
		Traballo práctico ( <i>prompts</i> )	Rúbrica de avaliación	40%	8%	
UD4. Introducción a control de versións	RA4	Traballo práctico	Rúbrica de avaliación	100%	20%	
UD5. Colaboración en Git	RA4	Traballo colaborativo	Exame escrito	85%	17%	
		Desafío en grupo	Lista de cotexo	15%	3%	
					100%	
* Para ver desglose por CAs, véxase Apartado 5					Multiplicador:	*0.6
						*0.4

O traballo continuo durante o trimestre (37% exames, 63% comprobación de traballo práctico) ten un valor do 60% da nota final de primeira oportunidade. O outro 40% obténse da nota do exame de avaliación, que ten unha nota mínima de 3 sobre 10. Se non se chega a ese mínimo, a nota da avaliación é suspensa (máximo: 4). Se do cómputo total a nota é 4.5 ou maior, a avaliación é aprobada.

#### 6.4. Cualificación do segundo trimestre (T2)

Avaliación continua						Exame final (T2)
	RA*s*	Obxecto de avaliación	Instrumento	Peso na UD	Peso na av. cont.	
UD6. Introducción á calidade software.	RA3	Esquema/resumo	Rúbrica de avaliación	20%	4%	
Probas unitarias		Traballo práctico	Rúbrica de avaliación	30%	6%	
			Exame escrito	50%	10%	
UD7. Depuración	RA3	Traballo práctico	Rúbrica de avaliación	100%	20%	
UD8. Requisitos e casos de proba	RA3	Traballo colaborativo	Rúbrica de avaliación (documento producido)	100%	20%	
UD9. Probas en dispositivos móbiles	RA3	Traballo práctico	Rúbrica de avaliación	100%	20%	
UD10. Probas en páxinas web	RA3	Traballo práctico	Exame escrito	100%	20%	
					100%	
* Para ver desglose por CAs, véxase Apartado 5					Multiplicador:	*0.6
						*0.4

As características do cómputo da cualificación son similares ao primeiro trimestre. 60% de peso da nota para o traballo continuo (30% exames, 70% rúbricas) e 40% de peso no exame final. Este ten un mínimo, especificado na cabeceira do exame, de 3 sobre 10. Se non se chega a ese mínimo, a nota da avaliación é suspensa (máximo: 4).

## 6.5. Cualificación do terceiro trimestre (T3)

Avaliación continua						Exame final (T3)
	RA*s	Obxecto de avaliación	Instrumento	Peso na UD	Peso na av. cont.	
UD11. UML. Conceptos básicos	RA5	Esquema/resumo (1)	Rúbrica de avaliación	25%	5%	
	RA6	Esquema/resumo (2)	Rúbrica de avaliación	25%	5%	
			Exame escrito		50%	
UD12. UML. Diagramas de clase	RA5	Traballo práctico	Resultados de <i>tests</i> automáticos	100%	20%	
UD13. UML. Diagramas de comportamento	RA6	Traballo práctico	Resultados de <i>tests</i> automáticos	100%	20%	
UD14. UML e código	RA6	Traballo colaborativo	Rúbrica de avaliación	100%	20%	
UD15. Padróns de deseño e refactorización	RA4	Traballo práctico	Resultados de <i>tests</i> automáticos	100%	20%	
					100%	
* Para ver desglose por CAs, véxase Apartado 5 %					Multiplicador: *0.6	*0.4

Aquí repítense os elementos dos anteriores trimestres: 60% avaliación continua e 40% de exame, con nota mínima de 3 sobre 10. Neste trimestre, o peso da avaliación directa do traballo práctico (90%) é maior, aliñándose así cos criterios no segundo curso.

## 7. Procedemento e criterios de recuperación das partes non superadas

### 7.1. Recuperación das avaliacións non superadas

Os estudantes que non superen unha avaliación na primeira oportunidade poderán optar a unha segunda oportunidade.

Para a recuperación do exame do trimestre realizarase un novo exame, de características e dificultade similar, durante a semana de recuperacións correspondente establecida no calendario: 14-18 de decembro (T1), 5-9 de abril (T2) e 14-18 de xuño (T3). A data exacta concrétase nas xuntas de avaliación. O exame faise xunto aos compañeiros e compañeiras en horario normal de clase no caso do primeiro e segundo trimestre. Entre o exame de primeira oportunidade e o exame de segunda oportunidade sempre hai unha semana na que o docente aproveitará polo menos unha clase para comentar os erros cometidos e dar orientacións para a preparación da recuperación. A nota de exame (40% do total) despois do exame de recuperación será a media aritmética entre a nota do exame na primeira oportunidade e a nota de recuperación, manténdose a limitación de mínimo de 3 sobre 10. Isto faise así para evitar que alumnos e alumnas non traballen de xeito constante e deixen o esforzo para uns poucos días antes da recuperación.

En caso de que a cualificación do traballo continuo nunha ou varias unidades didácticas sexa inferior a 5 sobre 10, pódese optar a recuperalo.

Para recuperar o traballo continuo, este debe repetirse durante o trimestre seguinte ou na semana posterior á de exames de avaliación no caso do terceiro trimestre. Os traballos poden ser entregados e avaliados en calqueira momento, incluíndo horario normal de clase. Tamén se poden repetir os exames parciais de tipo *test* e/ou resposta curta, pero estes non se realizarán individualmente, senón nunha data acordada cos interesados e interesadas, e só se repetirán unha vez. No caso de que un traballo práctico non fose aprobado porque se fixo de forma fraudulenta a primeira vez, o docente requereirá unha memoria PDF con capturas de pantalla suficientes que acrediten que a segunda entrega faise de forma honesta e/ou unha

defensa oral do mesmo. A recuperación dunha unidade didáctica debe facerse de xeito completo, repetindo todas as actividades, non só algunhas delas. Cando se recupere unha unidade didáctica, os instrumentos e criterios de cualificación serán os mesmos que na primeira oportunidade, pero a nota máxima acadable será de 5 sobre 10.

No caso de que, cos procedementos de recuperación establecidos neste apartado, algún estudante manteña algún, ou varios, trimestres suspensos, poderá optar ao exame final que está separado por trimestres e descrito no seguinte Apartado 7.2.

## **7.2. Cualificación do alumnado con perda ao dereito á avaliación continua**

Cando un alumno ou alumna teña máis de 6 faltas sen xustificar durante o curso, dacordo co establecido na *Orde do 12 de xullo de 2011* e ratificado na *Orde do 1 de xullo de 2025 pola que se dictan instrucións para formación profesional no curso 2025/26*, será apercibido. De incurriren en máis de 10 faltas sen xustificar, perderá o dereito á avaliación continua.

A perda do dereito á avaliación continua será, en consideración da excepcionalidade da medida, discutido e comentado nos claustros e xuntas de avaliación pertinentes co resto do profesorado. Se se considera favorable para o estudante, os traballos continuos serán corrixidos e avaliados polo docente en caso de que sexan entregados, pero a nota non se tomará en conta para a cualificación do estudante. Tamén, se así o desexa e se valora como favorable para o mesmo, o estudante poderá presentarse aos exames do trimestre e recibir as revisións nos mesmos, sen que a cualificación sexa tida en conta.

Estas medidas tómanse para favorecer que o alumnado con perda ó dereito á avaliación continua que manteña interese na materia poida adquirir as competencias e coñecementos necesarios cunha pauta de aprendizaxe constante e axeitada, de tal xeito que estea preparado para o exame extraordinario ao final do curso. O docente, para isto, irá comentando ao longo do curso os resultados de aprendizaxe esperable.

En todo caso, a cualificación dos alumnos con perda ó dereito á avaliación continua realízase cun exame extraordinario na semana de avaliacións de recuperación final (14-18 de

xuño). Este exame consta de tres partes diferenciadas: Primeiro trimestre (RA1, RA2, RA4), segundo trimestre (RA3) e terceiro trimestre (RA5, RA6, RA4). Para cada unha das partes, o exame en papel conterá varios exercicios de distinto tipo (a destacar: explicación do resultado dun comando ou operación no *debugger*, sinalización en capturas de pantalla de determinados botóns dacordo co solicitado, resposta curta completando un fragmento de código, interpretación ou produción de UML, resposta teórica elaborada,...) e tamén terá unha parte na que se instrúe ao estudante a levar a cabo tarefas no ordenador. O uso do ordenador estará supervisado en todo momento polo docente, que se sentará na fila de asentos detrás dos estudantes, e non estará permitido o uso de ferramentas de IA nin outra ferramenta non explicitamente indicada no exame. Non cumprir este requisito suporá unha cualificación de suspenso (1) en todo o exame.

Os criterios de cualificación concretos estarán especificados de xeito claro na cabeceira do exame. A nota de cualificación do módulo será a media aritmética das tres partes do exame (T1, T2 e T3), cada unha sobre 10. Para aprobar o módulo, todas as partes deben ter unha cualificación igual ou superior a 5 sobre 10 (4,9 non é 5).

En caso de aprobar algunha ou varias partes do exame (T1, T2 e T3), pero non todas, esa nota será preservada para o estudante durante o máximo dun ano académico para facilitar a súa recuperación do módulo, de acordo ao previsto no seguinte Apartado 7.3.

### **7.3. Procedemento de seguimento, recuperación e avaliación das materias pendentes**

Se un estudante está a cursar Contornos de Desenvolvemento en segunda convocatoria, no caso de repetir o 1º curso, a avaliación terá lugar cos mesmos procedementos e instrumentos que na avaliación ordinaria, cunha excepcionalidade: Se este alumno ou alumna aprobou algún trimestre no curso anterior, entón non estará obrigado a repetir esa avaliación e conservarase a nota do ano anterior. Non obstante, o docente remarcará vigorosamente que, igualmente, debe asistir á clase (ou perderá o dereito á avaliación continua) e é aconsellable que aproveite esas sesións para repasar os contidos, facer novamente as prácticas e aproveitar a oportunidade de mellorar a súa cualificación nese

trimestre. Adicionalmente, promoverase que o estudante adquira un rol diferente neste curso e “asista” e “axude” aos compañeiros que están a cursar o módulo por primeira vez, buscando nisto unha transformación no modelo mental e na autoconcepción do estudante de si mesmo.

Se o estudante está a cursar 2º curso entón non terá que recuperar as partes do módulo (T1, T2 e/ou T3) que xa aprobouse na convocatoria anterior, de forma ordinaria ou extraordinaria. O esforzo de recuperación non se trasladará a un único exame, se non a unha recuperación baseada na avaliación continua e retroalimentación [4]. Nunha xuntanza inicial na primeira semana do curso explicarase que se deben volver a facer todos os traballos das UD1, UD2, UD3, UD4, UD6, UD7, UD9, UD10, UD11, UD12, UD13 e UD15 que correspondan. Os enunciados publicaranse na aula virtual paulatinamente, con distintas datas de entrega acordadas cos estudantes na xuntanza inicial, e serán revisados e cualificados polo docente. Con respecto á Actividade 5 da UD3, será preciso elaborar unha memoria en PDF. Con respecto a outras actividades (6, 8, 11, 13) que teñen unha metodoloxía colaborativa, o docente clarificará na xuntanza inicial que haberá algunhas preguntas adicionais no exame de avaliación de cada trimestre. As Actividades 9 e 14 teñen un carácter complementario e non será esixibles. Os exames de avaliación de trimestre (40% da nota) terán lugar en datas acordadas cos estudantes, sempre a partires das avaliacións ordinarias do primeiro trimestre (4 de decembro) e antes das avaliacións previas á formación en empresa. Priorizaranse datas nas que a carga de avaliación de 2º sexa menor e que non perxudiquen aos estudantes con respecto á consecución dos obxectivos de 2º curso, polo que estas accións coordinaranse co profesorado dos módulos de 2º.

En caso de alumnado que veña a matricularse dende outro centro, o procedemento de recuperación non será diferente, coa única excepcionalidade de que se fará unha breve entrevista co estudante para indagar sobre o modelo pedagóxico e os procedementos de avaliación levados a cabo no outro centro coa intención de explicar as diferencias neste centro, facer unha pequena transición e abrirse a posibles suxerencias de mellora ou inquiredanzas por parte do estudante.

## **8. Avaliación inicial e medidas de reforzo educativo**

### **8.1. Avaliación inicial**

A avaliación inicial, especialmente no eido da formación profesional, plantexa o desafío de medir as competencias, non definidas de xeito observable, que se establecen no marco normativo [5]. Neste centro e nivel educativo, previsiblemente, a meirande parte dos estudantes acceden a 1º de Desenvolvemento de Aplicacións Multiplataforma cunhas expectativas claras e realistas do que van aprender, e cunha base mínima suficiente para arrancar cos obxectivos do curso, tanto de estudantes que acceden dende ciclo medio coma aqueles que acceden desde bacharelato. Pero isto non é así no 100% dos casos. Por iso, a avaliación inicial está orientada a identificar o antes posible unha desviación grande con respecto ao nivel de competencia esperable ou ao coñecemento inicial en relación ao desenvolvemento de aplicacións.

A Actividade 1 contempla unha presentación interactiva na que, entre outros contidos, plantéxanse as preguntas: “Que é un programa informático?” “Que é un proceso informático?” “Que é unha linguaxe de programación?” ou “Cales das seguintes son linguaxes de programación?” Aqueles estudantes con 0 ou 1 respostas correctas serán tidos en conta e terase unha breve conversa con eles durante os seguintes días para coñecer as súas expectativas e coñecementos actuais. No caso de que exista unha evidente e pronunciada desconexión cos obxectivos do ciclo, e/ou esta sexa subliñada no cuestionario da Actividade 3, estes casos serán comentados co resto de profesores do módulo na xuntanza de avaliación inicial do 15 de outubro. Se estas evidencias están remarcadas por outras observacións doutros docentes, o titor do grupo terá unha breve entrevista cos estudantes para profundizar na razón destas diferencias, recalcar os obxectivos do ciclo e contextualizar axeitadamente aos alumnos nos estudos que están a realizar, levando a cabo accións titoriais adicionais de ser preciso. Para o caso de Contornos de Desenvolvemento, o seguimento activo do docente durante as actividades de traballo autónomo centrarase nestes estudantes que poden precisar máis atención e apoio, perseguindo deste xeito unha igualdade real de oportunidades.

## **8.2. Alumnado que se incorpora ao longo do curso**

No caso de estudantes que se incorporan por razóns extraordinarias, ao longo do curso, distingúirase a aqueles que se incorporen antes de novembro (por exemplo, porque despois de esperar ao prazo de matrícula extraordinario na universidade non entraron na carreira que querían e optaron polo ciclo) e aqueles que por circunstancias excepcionais, se incorporen despois de novembro.

No caso de incorporarse antes de novembro, levarase a cabo unha entrevista individual de orientación para explicar os obxectivos do módulo e os procedementos de avaliación, e proporcionaráselle ao estudante os enunciados das tarefas da UD1, UD2 e UD3 sen que teña opción de participar no cuestionario *Wayground* das Actividades 1 e 3. As datas de entrega estarán adaptadas e se buscará un oco específico para a avaliación da UD3 de ser preciso, así coma a realización das probas escritas da UD1 e UD2.

No caso de que o estudante se incorpore despois de novembro, valorarase co claustro de profesores medidas excepcionais de adaptación curricular, sendo, en ausencia de consenso ou debida xustificación, necesaria a realización do exame de recuperación final para aqueles trimestres nos que non puido participar.

## **8.3. Atención á diversidade e alumnado con necesidades específicas de apoio educativo**

A Orde do 8 de setembro de 2021 desenvolve o Decreto 229/2011, do 7 de decembro, polo que se regula a atención á diversidade do alumnado dos centros docentes en Galicia, e en conxunto establecen a resposta educativa en Galicia para atender á diversidade do alumnado. En liña co recollido nos Artigos 8 e 9, nesta programación didáctica plantéxase:

- Para alumnado con TDAH ou dificultade de concentración recoñecida médicamente: Incremento do tempo para a realización das probas escritas. Ata 1 hora extra nas probas trimestrais, e o tempo de duración do recreo nas probas de unidade didáctica.
- Para alumnado con trastornos do espectro autista e/ou problemas de adaptabilidade social: Realización da avaliación da UD3 (entrevista individual) de xeito illado ao contexto da clase (por exemplo: nun recreo) ou mediante conversa por escrito na aula

virtual ou por email. Na Actividade 9 permitírase a súa participación individual, sempre tentando favorecer a participación e inclusión no grupo mediante acompañamento activo do docente. Nas Actividades 11 e 13 que se desenvolven de xeito colaborativo permitírase a participación de xeito separado do resto da clase (por exemplo, nunha localización na aula que non lle cause pánico social) sempre, tamén, tentando favorecer a participación e inclusión mediante acompañamento activo do docente.

- Para alumnado cunha discapacidade psíquica: Ampliación de ata dúas semanas dos prazos de entrega das prácticas, e acompañamento e orientacións individuais se o alumno está disposto a empregar o tempo de recreos e así o solicita.
- Para alumnado con altas capacidades intelectuais: Para fomentar o interese e promover o desenvolvemento axeitado deste tipo de alumnado que adoita “aburrirse” na clase e, en moitas ocasións, “atopar inxusto” que se lles propoñan tarefas adicionais, vaise empregar a ferramenta LeetCode<sup>7</sup>. Esta páxina web ten moitos problemas de programación e algoritmia. O docente pedirá ao estudante ou estudantes de altas capacidades que escollan arbitrariamente un problema a resolver, de dificultade alta, e, de xeito semanal ou coa frecuencia que o docente estime oportuna (e posible para el) resolverá ou tentará resolver dito problema. De xeito voluntario, o estudante ou estudantes tamén tentarán resolvelo. Se o conseguen, contrastarán e discutirán a súa solución coa do docente. Deste xeito, a explotación e expansión desas altas capacidades non se limita a “máis traballo para o estudante”, senón a un interese recíproco polo docente que pretende fomentar, sobre todo, a motivación do estudante e conseguir que sinta o éxito de explotar as súas capacidades a través de verse “a outro nivel” durante a discusión das solucións.
- Para alumnado con discapacidade visual: Reubicación dentro da aula para estar máis preto da primeira fila, consensuada previamente co titor do grupo.

---

7 <https://leetcode.com/>

As adaptacións pedagóxicas levadas a cabo durante o curso non se cinguirán só a estes casos e se valorarán tamén outras circunstancias que poidan ter lugar.

Adicionalmente, non se terá en consideración só aos estudantes que poidan presentar un certificado médico dalgún tipo de discapacidade ou que teñan outro xeito de demostrar que precisan necesidades específicas de apoio educativo. O alumnado sen necesidades específicas tamén, sempre, debe ser tratado do xeito máis individualizado que se poida e na medida do posible deben adaptarse as actuacións as súas circunstancias.

Unha posible actuación que se considera é a adaptación das UD8 e UD9. Estas Actividades 13 e 14 están relacionadas a unha aplicación Android de xadrez (XadrezApp). No caso de que un ou varios estudantes demostren un alto nivel de interese por outro xogo ou afición diferente (por exemplo: Fútbol, baloncesto,...) barállase a realización destas unidades didácticas en torno a este interese distinto do alumnado. Co paso dos anos, espérase que esta adaptación curricular amplíe as opcións posibles en torno ás que xiren estas unidades didácticas e fagan máis personalizable o proceso de ensinanza-aprendizaxe. Así mesmo, tamén se poden barallan adaptacións, en función dos intereses do alumnado, nos problemas de IA da Actividade 6 e o CLI plantexado na Actividade 14.

## 9. Educación en valores e actividades complementarias

### 9.1. Educación en valores

A integración de valores na aula é unha peza clave na motivación do alumnado [6] e está explicitamente recollido no *Real Decreto 659/2023, do 18 de xullo, polo que se desenvolve a ordenación do sistema de Formación Profesional* que implementa a *Lei Orgánica 3/2022*, que, entre outros, a educación en igualdade, inclusión, innovación, ética profesional, e sustentabilidade, deben ser parte dos principios curriculares.

A principal actuación a levar a cabo en liña coa educación en valores é a acción constante e sostida do docente durante todos os días do curso, escollendo tono, palabras e liñas de discurso que demostren respecto, empatía e cidadanía responsable, e perseguindo ser referente para o alumnado nese ámbito, e sendo en todo momento consciente da súa responsabilidade na aula e na sociedade.

Dúas actividades específicas que se contemplan, alén doutras que poidan adoitarse en función de circunstancias adicionais, son a análise de resultados do mesmo *prompt* en lingua galega e en lingua castelá e/ou inglesa, na semana do Día das Letras Galegas de 2027, coma breve apuntamento nos últimos minutos da clase que demostre o sesgo lingüístico dos LLM e abra unha breve discusión sobre as implicacións disto, poñendo en valor a nosa lingua e a nosa cultura. Tamén, a exposición das citas electorais que terán lugar nos vindeiros anos, surxindo isto a partires das eleccións municipais de maio de 2027. Preténdese, nunha breve conversa de 15-20 minutos, dar a coñecer as distintas citas electorais e os organismos para os cales se dirixen, comprendendo as diferencias competenciais en materia legislativa de Concellos, Parlamento de Galicia, Congreso e Senado de España e Parlamento europeo. O profesorado dos ciclos ofertados da rama de administración e finanzas do centro pode contribuír a esta actividade.

## 9.2. Actividades complementarias

As actividades complementarias tamén son unha oportunidade para que a aprendizaxe dos estudantes non se quede só na aula e os conduzan á adquisición de valores e/ou ao achegamento a realidades que sería imposible ensinar desde o centro.

Para complementar a aprendizaxe no módulo de Contornos de Desenvolvemento (e moitos outros) repetiranse este ano tres actividades complementarias xa feitas outros anos:

Primeiro, a asistencia á Feira de Emprego GaliciaTIC 2027, que ten lugar en novembro en Santiago de Compostela e acolle a estudantes de FP de informática de varios centros. Nesta actividade os estudantes poden establecer contactos con distintas empresas que, en *stands*, explican en que consiste o traballo que desenvolven e cal é o perfil profesional que buscan. É unha oportunidade excelente para que os estudantes adquiran contactos de cara á formación en empresa en 2º curso e, potencialmente, en 1º curso, e tamén, que pregunten que IDEs empregan os traballadores e como é a relación da empresa co uso da IA. Adicionalmente, hai charlas que teñen lugar de forma paralela e nas que, en anos anteriores, destacamos a exposición do *influencer* Mouredev ou a mesa de debate de profesores de FP e responsables de recursos humanos de empresas na que participou Rubén Montero, e foron de grande interese para o alumnado.

En segundo lugar, a asistencia ao OpenDay Deloitte que anualmente invita ao noso centro e realízase no primeiro trimestre. Esta experiencia permite a estudantes de 1º curso participar en diversas charlas e coñecer as liñas de traballo desta consultora.

Por último, programárase unha visita ao Centro de Investigación en Tecnoloxías da Información e Comunicación (CITIC) de A Coruña. Pode solicitarse facilmente e concertarse con antelación. Os estudantes sempre devolven un *feedback* moi positivo da experiencia e amplían o seu horizonte sobre a realidade tecnolóxica actual, aprendendo sobre liñas de investigación de vangarda e incorporando ás súas expectativas de futuro novas realidades que non contemplaban con anterioridade, coma a investigación científica.

As actividades complementarias serán recollidas na Programación Xeral Anual (PXA).

## **10. Seguimento da programación e avaliación da práctica docente**

Non existe unha “programación didáctica perfecta”. A planificación pedagóxica para un curso académico sempre depende do contexto do alumnado e da actualidade do eido profesional, así que adaptarse é fundamental para manter a calidade na ensinanza.

### **10.1. Avaliación da práctica docente**

Esta programación didáctica pon un gran peso á actuación docente, tanto nas sesións expositivas teórico-prácticas (que, a pesar do nome, deben estar orientadas máis á “interactividade” que á mera “exposición” de contidos) coma ao seguimento e acompañamento activo do alumnado durante o traballo autónomo.

Polo tanto, ao final de cada trimestre, e despois da recibir a nota de avaliación (a fin de non “branquear” os resultados) os estudantes poderán responder a un formulario anónimo a través da aula virtual que emprega os seguintes indicadores (1-10) en canto ás actuacións do docente: Claridade expositiva, adecuación dos contidos, adecuación do estilo de transmisión, grao de interese promovido no alumnado e grao de innovación docente. Ademais, un campo de texto no que se poderán escribir suxerencias.

Esta retroalimentación será empregada ao final de cada trimestre co fin de mellorar o estilo docente no seguinte período, e tamén valorarase conxuntamente ao final de curso de cara ao próximo ano académico. Non obstante, a retroalimentación verbal obtida ao longo das sesións será o primeiro indicador para modificar parámetros da actuación (a destacar: ritmo de exposición, reforzo dos contidos, adaptación dos exemplos), e será solicitada retroalimentación ao alumnado polo menos unha vez cada 3 ou 4 semanas para coñecer as súas necesidades.

### **10.2. Avaliación da programación**

A adecuación dos procedementos de avaliación, os tempos e os tipos de actividades recollidos na programación didáctica será tamén obxecto de estudo e refinamento.

Sendo esta unha programación na que o traballo autónomo de cada alumno e alumna ten moita importancia, é crucial adaptar correctamente os tempos de cada actividade ao

esfuerzo requerido para completalas. Se hai poucas sesións para prácticas moi longas, o alumnado sentirá fatiga e a presión empeorará o seu desempeño. Se hai moitas sesións para prácticas curtas, o alumnado vaise desconectar rápidamente dos obxectivos e sentirá unha forte desmotivación. Para as Actividades 2, 4, 5, 7, 12, 14, 15, 18, 19, 21 medirase, ao final de cada trimestre onde corresponda, o prazo medio de entrega. Dado que pode coñecerse a data de entrega (na aula virtual, ou o último *commit*) das prácticas, pode observarse a antelación da data de entrega do alumnado. Este indicador valorarase do seguinte xeito:

Antelación media da entrega (en minutos, contabilizados só nos tempos de clase)					
+150	100-150	75-100	30-75	10-30	0-10
Moi, moi mal	Moi mal	Regular	Ben	Regular	Mal

Un resultado laranxa, vermello ou negro requerirá a revisión e toma de decisións para correxir futuros problemas similares de xeito inmediato. Tamén se a porcentaxe de entregas é menor ao 90%.

Ademais da adaptación dos tempos das entregas ou do contido das prácticas, o formulario trimestral de avaliación docente tamén permitirá aos alumnos expresar (1-10) a súa percepción da idoneidade do procedemento de avaliación, e da metodoloxía de cada actividade. Realizarase con eses datos un seguimento de cada unidade didáctica recopilando tamén a porcentaxe de alumnos con resultados positivos e negativos, e, do mesmo xeito, un seguimento final que será recollido no apartado de seguimento da aplicación PROENS<sup>8</sup>. Describiranse outros posibles desvíos no cumprimento da programación, o resumo das actividades de recuperación realizadas e propostas de mellora para o vindeiro curso.

### 10.3. Valoración global co claustro ao final do curso

Ao final do curso, valorarase co resto de docentes o grao global de consecución dos obxectivos didácticos e reflectirase o grao de cumprimento das programacións (total, na meirande parte, parcialmente) no informe de final de curso.

<sup>8</sup> <https://www.edu.xunta.gal/proens>

## 11. Bibliografía

[1] G. V. Leytón-Yela, V. A. Bucheli-Guerrero y H. A. Ordoñez-Erazo, "Herramientas usadas para la evaluación formativa automatizada en cursos de programación asistidos por computadora," *Revista Científica*, no. 44, 2022, doi: 10.14483/23448350.19662.

[2] L. M. Romero-González, «Análisis de la evaluación y la calificación. ¿Es posible una coordinación pedagógica entre ambos enfoques?», *epsir*, vol. 9, pp. 1–21, jul. 2024.

[3] J. Nieto Ortiz, M. Ángeles Millán Gutiérrez, D. Lizcano Casas, y J. Roa González, «Características de la evaluación en la Formación Profesional desde un enfoque basado en las competencias», *ASE*, n.º 39, jun. 2023, doi: 10.23824/ase.v0i39.787.

[4] V. M. López-Pastor, T. Fuentes Nieto, C. Fernández Garcimartín, M. Molina Soria y C. Pascual Arias, "Importancia de la evaluación formativa en formación inicial del profesorado: revisión de lecciones aprendidas," *Revista Contemporânea de Educação*, vol. 18, no. 41, pp. 54–74, 2023, doi: 10.20500/rce.v18i41.54774

[5] J. Nieto Ortiz y M. L. Cacheiro González, «La evaluación de las competencias en la formación profesional desde un enfoque basado en los resultados de aprendizaje», *RIO*, n.º 27, pp. 173–196, nov. 2021.

[6] L. Collados Torres, M. Á. Hernández Prados y A. P. Moya Sáez, "Desmotivación en formación profesional: una aproximación desde la educación en valores," *Revista Científica UISRAEL*, vol. 11, no. 2, pp. 95–116, 2024. doi: 10.35290/RCUI.V11N2.2024.1128.