

OTRAS DISPOSICIONES

DEPARTAMENTO DE EDUCACIÓN

1344

ORDEN de 14 de febrero de 2023, del Consejero de Educación, por la que se establecen tres programas de especialización profesional.

El Estatuto de Autonomía del País Vasco, en su artículo 16, atribuye la competencia propia sobre la enseñanza en toda su extensión, niveles y grados, modalidades y especialidades a la Comunidad Autónoma del País Vasco, sin perjuicio del artículo 27 de la Constitución y Leyes Orgánicas que lo desarrollen, de las facultades que atribuye al Estado el artículo 149.1.30.^a de la misma y de la alta inspección necesaria para su cumplimiento y garantía.

La Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la formación profesional, deroga la Ley Orgánica 5/2002, de 19 de junio, de las cualificaciones y de la formación profesional, que tenía por finalidad la ordenación de un sistema integral de formación profesional, cualificaciones y acreditación, que respondía con eficacia y transparencia a las demandas sociales y económicas a través de las distintas modalidades formativas. A día de hoy, con la Ley Orgánica 3/2022 se responde a las necesidades y a los modelos que la nueva economía requiere.

En el ámbito laboral, de acuerdo con lo dispuesto en el artículo 12.2 del Estatuto de Autonomía, corresponde a la Administración General de la Comunidad Autónoma del País Vasco la competencia de ejecución de la legislación del Estado, especialmente, en lo que aquí es más relevante, promoviendo la cualificación de los trabajadores y las trabajadoras y su formación integral.

Para mejorar la empleabilidad de las personas, tanto en el corto como en el largo plazo, se va a requerir de nuevas estrategias y mecanismos. Por un lado, incrementando las horas dedicadas a los procesos de adquisición de competencias como única forma de lograr el mayor grado de especialización que demandan ámbitos cada vez más complejos. Por otro lado, la demanda de trabajadoras y trabajadores con una formación y competencias que se ajusten al entorno competitivo actual exige romper con esquemas anteriores y evolucionar desde un modelo formativo orientado al «puesto de trabajo» hacia otro centrado en el «campo profesional». Un cambio de paradigma que coloca a la persona en el centro promoviendo la adquisición o consolidación de competencias técnicas, personales y sociales, que garanticen la polivalencia y funcionalidad necesarias.

El establecimiento de cualificaciones más adecuadas a las necesidades reales del tejido productivo debe permitir, por una parte, adecuar la formación de las personas que estudian formación profesional a las necesidades cada vez más especializadas de las empresas y, por otra, mejorar la cualificación de los trabajadores y las trabajadoras dotándoles de las competencias que demandan los sectores productivos generadores de empleo.

La mejora de la formación profesional, en términos de eficacia, exige una especialización de la oferta y una planificación de la misma más ajustada a las necesidades del mercado laboral, especialmente en aquellos sectores y puestos de trabajo emergentes, que generen más empleo y que sean estratégicos para el futuro de la economía del País Vasco.

La formación profesional se revela, en este contexto, como un elemento clave para facilitar las herramientas que deben dar respuesta a las cualificaciones demandadas por los puestos de trabajo presentes y futuros.

jueves 16 de marzo de 2023

El hecho de que existan numerosas demandas provenientes de los sectores productivos relevantes para la economía origina la necesidad de impulsar la elaboración de unos programas de formación que den respuesta rápida tanto a la adecuación y mejora de la empleabilidad de las personas como a las demandas de mayor especialización del tejido productivo y que puedan ser certificados por la Administración de la Comunidad Autónoma del País Vasco. Estos programas, certificados de esta forma, no darán lugar a un título o certificación académica, certificación profesional o certificación parcial acumulable en tanto que las competencias no estén incluidas en el Catálogo Nacional de Cualificaciones Profesionales.

En el Decreto 32/2008, de 26 de febrero, por el que se establece la ordenación general de la Formación Profesional del Sistema Educativo, modificado por el Decreto 14/2016, de 2 febrero, se establecen los programas de especialización profesional del País Vasco en el ámbito de la formación profesional, así como su reconocimiento y certificación, que acredite su valor dentro del marco normativo vigente.

Es por todo ello que en la Ley 4/2018, de 28 de junio, de Formación profesional del País Vasco, en el capítulo V, se establece el Marco Vasco de Cualificaciones y Especializaciones Profesionales, con objeto de dar respuesta a nuestro mercado de trabajo a través del sistema general de formación profesional.

En él se incluirán las certificaciones y acreditaciones propias de los programas de especialización profesional del País Vasco. La Ley de Aprendizaje a lo Largo de la Vida ya establece el sistema de acreditación de las actividades de aprendizaje a través de diferentes vías; en esta ley se trata de complementar aquella regulación con referencia a una de las actividades que se desea promover de forma singular: los programas de especialización en el ámbito profesional, actividades que requieren de un reconocimiento y certificación que reconozca su valor dentro del marco normativo vigente.

Con este referente para su elaboración, se han analizado las demandas de sectores productivos estratégicos en nuestra economía y de esta forma se han definido los programas de especialización profesional que se incluyen en la presente Orden.

Por todo lo expuesto,

RESUELVO:

Artículo 1.– Objeto.

1.– La presente Orden tiene por objeto establecer la estructura de tres programas de especialización profesional que se incorporan en los anexos, de acuerdo con lo establecido en el artículo 12 ter del Decreto 32/2008, de 26 de febrero, por el que se establece la ordenación general de la Formación Profesional del Sistema Educativo en el País Vasco.

2.– Los programas de especialización que se establecen son los siguientes:

Anexo I: Tecnología Java.

Anexo II: Fabricación de Piezas y Elementos de Madera a partir de Madera en Rollo.

Anexo III: Carpintería de Ribera.

Artículo 2.– Finalidad.

1.– Estos programas están dirigidos a satisfacer las necesidades de especial cualificación demandadas por diversos sectores productivos estratégicos del País Vasco, especialmente en el ámbito industrial, de modo que se permita mejorar su competitividad.

2.– Así mismo, estos programas permitirán mejorar la empleabilidad de las personas que estudian formación profesional, así como de los titulados y las tituladas y de los y las profesionales cualificados y cualificadas, posibilitando profundizar en los conocimientos y ampliar las competencias profesionales requeridas por determinados sectores productivos.

Artículo 3.– Desarrollo.

1.– Estos programas se desarrollarán, prioritariamente, alternando la actividad entre el centro de formación profesional y las empresas. En la planificación para la puesta en marcha de cada programa de especialización profesional se especificará el desarrollo del mismo tanto en los centros de formación profesional como en la o las empresas, respetando en todo caso la definición de la estructura y las condiciones e impartición de cada programa.

2.– Entre el profesorado que actúe en la impartición del programa, el centro nombrará un coordinador o una coordinadora responsable de la coordinación del proceso de evaluación en el centro y en las empresas.

Artículo 4.– Oferta y autorización.

1.– La Viceconsejería de Formación Profesional, podrá planificar en los centros de formación profesional dependientes del departamento competente en materia de educación, o autorizar en centros privados o centros dependientes de otras administraciones que lo soliciten, la impartición de los programas de especialización profesional, siempre que dicho centros tengan ya autorizado y estén impartiendo alguno o algunos de los ciclos formativos asociados al programa, según se indica en el apartado d) del currículo correspondiente.

2.– En el caso de programas de especialización profesional incluidos como formación complementaria en programas de formación profesional dual en régimen de alternancia de más de dos años de duración, la autorización por parte de la Viceconsejería de Formación Profesional de dichos programas llevará implícita la autorización del programa de especialización profesional incluido en los mismos.

3.– Tal como se indica en el párrafo 3 del artículo 12 ter del Decreto 32/2008, de 26 de febrero, por el que se establece la ordenación general de la Formación Profesional del Sistema Educativo en el País Vasco, modificado mediante el Decreto 14/2016, de 2 de febrero, de modificación del Decreto por el que se establece la ordenación general de la Formación Profesional del Sistema Educativo, de manera excepcional y previa autorización de la Viceconsejería de Formación Profesional, podrán ofertarse estos programas de especialización profesional a los titulados y tituladas de formación profesional, así como a profesionales que sean propuestos para ello por las empresas colaboradoras en la impartición del programa y que cumplan los requisitos de experiencia y formación que se establezcan. Se cursarán en alternancia con la estancia formativa en las empresas. La solicitud para esta modalidad de oferta deberá estar debidamente motivada, justificando las razones que justifican esta excepcionalidad.

4.– En cualquiera de los casos, además de los aspectos propios de la organización del programa, en la solicitud deberá señalarse expresamente el profesorado del centro de formación profesional y los instructores de empresa que participen en el mismo, a los efectos de verificar el

jueves 16 de marzo de 2023

cumplimiento de los requisitos de especialidad, formación y experiencia. Esta información podrá ser sustituida por una declaración suscrita por el Director o Directora del centro de formación y la representación de la empresa de aportar personal con la cualificación necesaria antes del inicio de la actividad, lo cual deberá ser verificado antes del inicio de la actividad a instancia de la persona titular de la Dirección de Formación y Aprendizaje.

Artículo 5.– Formalización de los acuerdos entre los centros de formación y las empresas.

1.– En relación con los programas que se desarrollen en el marco de la formación profesional dual en régimen de alternancia para personas que cursan un ciclo formativo de formación profesional, su desarrollo se realizará de acuerdo con los términos establecidos en el Decreto 83/2015, de 2 de junio, por el que se establece la Formación Profesional Dual en Régimen de Alternancia en la Comunidad Autónoma del País Vasco.

2.– En relación con la modalidad prevista para titulados y tituladas de formación profesional o profesionales propuestos por empresas que se indica en el párrafo 3 del artículo 4 de la presente Orden, en el caso de que se desarrolle en alternancia con la actividad en la empresa, el acuerdo con cada empresa colaboradora para el desarrollo de los programas se plasmará en un convenio entre el centro de formación profesional y la empresa participante, con las características y competencias que se indican en el artículo 7 del citado Decreto 83/2015, de 2 de junio. En este supuesto, con carácter general, dada la especial naturaleza de estos programas propuestos desde el tejido productivo, la suma de horas dedicadas a la impartición en el centro de formación no podrá suponer más del 40 % de la duración total establecida para el programa.

3.– Los aspectos relativos a las obligaciones asumidas con respecto a la financiación y contratación de seguros u otros deberán reflejarse expresamente en el convenio suscrito de acuerdo con lo establecido en este artículo.

4.– En dicho documento se indicará expresamente la identidad del coordinador o coordinadora indicado en el artículo 3.2 de la presente Orden.

Artículo 6.– Requisitos y obligaciones de las empresas participantes.

1.– Las empresas participantes en cualquiera de las modalidades estarán sujetas a los requisitos y obligaciones recogidas en el Decreto 83/2015, de 2 de junio, por el que se establece la Formación Profesional Dual en Régimen de Alternancia en la Comunidad Autónoma del País Vasco. En particular, para las modalidades indicadas en el párrafo 3 del artículo 4 de la presente Orden, deberán contar con centros de trabajo ubicados en la Comunidad Autónoma del País Vasco.

2.– Las empresas participantes deberán facilitar a cada persona participante en el programa una formación teórica y práctica, suficiente y adecuada, en materia preventiva en el momento de su incorporación, en los términos señalados en la Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales, y en sus normas de desarrollo, en cuanto les sean de aplicación.

Artículo 7.– Requisitos del profesorado impartidor.

Los requisitos exigibles al profesorado e instructores e instructoras serán los señalados en cada uno de los programas de especialización profesional.

Excepcionalmente, en aquellos programas de especialización para cuya impartición sea necesario algún tipo de habilitación o formación acreditada por parte de administraciones distintas de la educativa, la posesión de dicha acreditación será requisito imprescindible para el profesorado impartidor.

Artículo 8.– Condición del alumnado durante su estancia en la empresa y financiación de los costes.

En los programas de especialización, desarrollados en alternancia entre el centro de formación profesional y la empresa, podrá participar el siguiente alumnado:

a) El alumnado que desarrolla el programa de especialización como formación complementaria mientras está cursando un ciclo de formación profesional dual del sistema educativo. Tendrá la condición de becaria o becario, y la empresa abonará los costes correspondientes a la beca y los costes de Seguridad Social que le correspondan como entidad becante, así como el coste de formación en la empresa.

b) El alumnado titulado. Tendrá la condición de becaria o becario y la empresa abonará los costes correspondientes a su beca y a la Seguridad Social que le correspondan como entidad becante, así como el coste de formación en la empresa.

c) Profesionales de la empresa colaboradora. Tendrán una relación laboral con la empresa y el coste de su formación correrá a cargo de la misma.

Artículo 9.– Seguros de los programas.

1.– En el caso de los programas financiados mediante becas, deberán cumplirse las obligaciones señaladas en el Real Decreto 1493/2011, de 24 de octubre, por el que se regulan los términos y las condiciones de inclusión en el Régimen General de la Seguridad Social de las personas que participen en programas de formación, en desarrollo de lo previsto en la disposición adicional tercera de la Ley 27/2011, de 1 de agosto, sobre actualización, adecuación y modernización del sistema de la Seguridad Social.

2.– Asimismo, en el supuesto de la oferta excepcional contemplada en el párrafo 3 del artículo 4 de la presente Orden, el Centro de Formación profesional deberá disponer de una póliza de accidentes que garantice las coberturas de fallecimiento, incapacidades, así como la asistencia sanitaria por dicho riesgo y una póliza de responsabilidad civil que otorgue cobertura a la actividad desarrollada por el alumnado en el marco del programa.

Artículo 10.– Programaciones.

1.– El centro autorizado para la impartición de cada programa de especialización profesional deberá elaborar una programación para el desarrollo del mismo, respetando en todo caso la definición de la estructura y las condiciones de impartición de cada programa. En dicha programación deberán establecerse, en relación con las competencias a adquirir, el desarrollo de los contenidos de los ámbitos de formación. Asimismo, en el desarrollo de los contenidos deberán manifestarse las actividades de aprendizaje claves a realizar, especificando aquellas que se desarrollarán en el centro de formación profesional y aquellas que se desarrollarán en el contexto de la empresa.

2.– En la programación deberá indicarse, además del profesorado que interviene en la formación por parte del centro, el personal que asume responsabilidades de formación, como instructor o instructora, por parte de la empresa. En dicha programación se establecerán también los aspectos de coordinación entre la persona que desarrolle la función de coordinador por parte del centro y el instructor o instructora que designe la empresa.

Artículo 11.– Proceso de Evaluación y certificación.

1.– En la programación del programa deberá establecerse el proceso de evaluación de los resultados de aprendizaje que logre el alumnado.

jueves 16 de marzo de 2023

2.– Deberá de informarse al alumnado al inicio del programa de las características del proceso de evaluación, así como de los criterios para la calificación.

3.– Como mínimo, en dos momentos intercalados proporcionalmente en el calendario del programa, deberán de realizarse sesiones de evaluación de seguimiento del progreso de cada alumno o alumna en el programa. El alumnado será informado documentalmente de posibles resultados parciales que haya logrado, la evolución de sus aprendizajes y, en su caso, las actividades de refuerzo necesarias, que serán planificadas en el marco de la programación.

4.– La valoración de cada uno de los ámbitos en particular y del programa en su conjunto, corresponde al profesorado que intervenga en su impartición. En aquellos ámbitos que se desarrollan parcialmente o en su totalidad en el contexto de la empresa, tendrán asignado un profesor o profesora del centro que compartirá con el instructor o la instructora o, en su caso, recogerá en contacto con el mismo o la misma la valoración de la evolución alcanzada por cada persona participante en el programa.

5.– La valoración realizada del aprendizaje de las personas participantes en el programa deberá recogerse en una sesión de evaluación específica al final del mismo y documentarse en un acta, con la firma de profesorado interviniente en el proceso. Una copia de esta acta deberá ser remitida por el Director o Directora del centro a la Viceconsejería de Formación Profesional, para que proceda a expedir las certificaciones correspondientes.

6.– La formación relativa a los programas de especialización profesional del País Vasco no dará lugar a un título o certificación académica, certificación profesional o certificación parcial acumulable en tanto que las competencias no estén incluidas en el Catálogo Nacional de Cualificaciones Profesionales. La Viceconsejería de Formación Profesional expedirá una certificación del programa a aquellas personas que sean evaluadas positivamente en el mismo, de acuerdo con lo señalado en el artículo 12 ter del Decreto 32/2008, de 26 de febrero, por el que se establece la ordenación general de la Formación Profesional del Sistema Educativo en el País Vasco, modificado mediante el Decreto 14/2016, de 2 de febrero, de modificación del Decreto por el que se establece la ordenación general de la Formación Profesional del Sistema Educativo.

DISPOSICIÓN ADICIONAL.– Las lenguas en la oferta de los programas.

La Viceconsejería de Formación Profesional impulsará que los programas de especialización profesional se puedan cursar tanto en las dos lenguas oficiales de la Comunidad Autónoma del País Vasco como en otras lenguas extranjeras, o en modelo mixto entre ellas, adaptando su oferta de manera progresiva.

DISPOSICIÓN FINAL.– Entrada en vigor.

La presente Orden entrará en vigor el día siguiente al de su publicación en el Boletín Oficial del País Vasco.

En Vitoria-Gasteiz, a 14 de febrero de 2023.

El Consejero de Educación,
JOKIN BILDARRATZ SORRON.

ANEXO I A LA ORDEN DE 14 DE FEBRERO DE 2023
PROGRAMA DE ESPECIALIZACIÓN EN TECNOLOGÍA JAVA

a) DATOS DE IDENTIFICACIÓN.

Denominación: TECNOLOGÍA JAVA.

Código: EP033.

Duración: 700 horas.

b) PERFIL PROFESIONAL.

Competencia general:

Desarrollar, implantar, documentar y mantener aplicaciones web y multiplataforma, utilizando tecnología Java avanzada, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.

Campo profesional:

Esta figura profesional ejerce su actividad en empresas de servicios de tecnología de la información.

Las ocupaciones y puestos de trabajo más relevantes:

- Desarrollador o desarrolladora Java.
- Desarrollador o desarrolladora Web.

Competencias técnicas, personales y sociales para su intervención profesional:

a) Desarrollar aplicaciones Java, libres de vulnerabilidades, cumpliendo con las necesidades de uso y siguiendo estándares de desarrollo de software.

b) Realizar planes de pruebas, verificando el funcionamiento de los componentes desarrollados.

c) Gestionar entornos de desarrollo, adaptando su configuración para permitir el desarrollo y despliegue de aplicaciones en Java.

d) Desarrollar aplicaciones multiplataforma en lenguaje Java, utilizando librerías y herramientas adecuadas a las especificaciones.

e) Desarrollar interfaces gráficos de usuario, empleando componentes visuales estándar o específicos.

f) Desarrollar aplicaciones web en entorno cliente, utilizando librerías del lenguaje de programación y librerías de estilos.

g) Desarrollar prototipos interactivos para presentar aplicaciones gráficas.

h) Desarrollar aplicaciones web en entorno servidor, utilizando la plataforma Jakarta EE.

i) Desarrollar JSPs, utilizar Java Standard Tag Library (JSTL).

j) Desarrollar aplicaciones web entorno servidor seguras, utilizando el protocolo HTTPS.

jueves 16 de marzo de 2023

k) Desarrollar aplicaciones en Java, manteniendo la integridad, consistencia y persistencia de los datos.

l) Desarrollar aplicaciones avanzadas en Java, utilizando mapeos entre objetos y tablas.

m) Desarrollar aplicaciones en Java, utilizando las funcionalidades de los Frameworks, manteniendo la integridad, consistencia y persistencia de los datos.

n) Desarrollar servicios web RESTFul, utilizando las funcionalidades de los Frameworks.

o) Desarrollar y asegurar aplicaciones en Java, aplicando mecanismos de gestión de usuarios y control de acceso de los Frameworks y realizando pruebas unitarias y de integración.

p) Consumir servicios web RESTFul, utilizando las funcionalidades de los Frameworks.

q) Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y gestión de versiones.

r) Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación, verificando su comportamiento y realizando modificaciones.

s) Resolver situaciones, problemas o contingencias con iniciativa y autonomía en el ámbito de su competencia, con creatividad, innovación y espíritu de mejora en el ámbito personal y en el de las y los miembros del equipo.

t) Organizar y coordinar equipos de trabajo con responsabilidad, supervisando el desarrollo del mismo, manteniendo relaciones fluidas y asumiendo el liderazgo, así como aportando soluciones a los conflictos grupales que se presenten.

u) Comunicarse con sus iguales, superiores y personas bajo su responsabilidad, utilizando vías eficaces de comunicación, transmitiendo la información o conocimientos adecuados y respetando la autonomía y competencia de las personas que intervienen en el ámbito de su trabajo.

v) Supervisar y aplicar procedimientos de gestión de calidad, de accesibilidad universal y de «diseño para todas las personas», en las actividades profesionales incluidas en los procesos de producción o prestación de servicios.

c) FORMACIÓN

Ámbitos de aprendizaje	Asignación horaria
1. Programación básica en Java	40 horas
2. Programación gráfica en Java. Multiplataforma	60 horas
3. Programación en entorno Cliente	100 horas
4. Programación en entorno Servidor	150 horas
5. Persistencia de información en Java	80 horas
6. Frameworks de desarrollo	120 horas
7. Desarrollos avanzados	100 horas
8. Gestión del desarrollo en equipo de una aplicación	50 horas
TOTAL HORAS	700 horas

RESULTADOS DEL APRENDIZAJE DEL PROGRAMA:

RESPONSABILIDAD Y AUTONOMÍA EN LA ACTIVIDAD PROFESIONAL (Transversales al programa).

Esta persona asume la responsabilidad de desarrollar, implantar, documentar y mantener aplicaciones web y multiplataforma, utilizando tecnología Java avanzada.

Asociados al ámbito 1: PROGRAMACIÓN BÁSICA EN JAVA.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Desarrollar aplicaciones, siguiendo estándares de desarrollo de software.

Criterios de valoración:

- a) Se han identificado las normas y estándares para el desarrollo de software.
- b) Se ha reconocido la necesidad de utilizar un estándar de diseño de software.
- c) Se ha utilizado un modelo de desarrollo estándar para el diseño de la aplicación.
- d) Se han cumplido todas las fases del desarrollo de aplicaciones.
- e) Se ha instalado y ejecutado un entorno de desarrollo integrado.
- f) Se han creado modelos libres de vulnerabilidades.

Conocimientos:

- Normas y estándares para el desarrollo de software: SDLC, Standard PASTA... Beneficios de su utilización.
- Entornos integrados de desarrollo: editor de código de programación, entorno de ejecución, depurador e IDE de interfaz gráfica.
- Fases para el desarrollo de aplicaciones: conceptualización, definición, diseño, desarrollo, pruebas y despliegue.
- Entornos integrados libres y propietarios.
- Métricas y modelos libres de vulnerabilidades. Standard OWASP.

2.– Desarrollar aplicaciones, utilizando las estructuras básicas de control del lenguaje.

Criterios de valoración:

- a) Se han utilizado diferentes estructuras de selección.
- b) Se han utilizado diferentes estructuras de repetición.
- c) Se han utilizado diferentes estructuras de salto.
- d) Se ha creado una secuencia lógica de programación.

e) Se han utilizado datos, expresiones y operadores.

f) Se ha comentado y depurado el programa.

g) Se ha creado un programa ejecutable estable.

Conocimientos:

- Operativa de las estructuras de control.
- Categorías de las estructuras de control: secuencial, condicional o selectiva, iterativa o repetitiva y de salto.
- Tipos de datos, expresiones y operadores.
- Estructuras de control básicas: selección, repetición y salto.
- Depuración y documentación de programas: perspectiva Debug en Eclipse, vista variables, vista breakpoints, vista Debug, botones para controlar la depuración...
- Realización de programas ejecutables. Generación de ejecutables: compilación, depuración y ejecución.

3.– Desarrollar aplicaciones, utilizando las características avanzadas de los lenguajes orientados a objetos.

Criterios de valoración:

a) Se ha creado una clase y se han creado objetos que instancian elementos de esa clase.

b) Se ha creado una estructura de clases.

c) Se han creado atributos con diferentes accesos.

d) Se han creado métodos, con y sin argumentos.

e) Se han creado clases, atributos y métodos.

f) Se han creado clases y subclases.

g) Se han creado objetos de clases y subclases y se ha accedido a atributos de clase y superclase.

h) Se han diseñado métodos homónimos en clases y subclases implementando polimorfismo.

i) Se han creado métodos de subclase que redefinen los métodos de superclase.

Conocimientos:

- Concepto de clase. Concepto, forma general y definición de clase.
- Instanciación de clases. Creación de objetos. Constructores y destructores.
- Estructura y miembros de una clase: atributos y métodos. Componentes static.
- Definición de atributos y control de acceso. Especificadores de acceso: public, protected y private.

- Definición de métodos con y sin parámetros.
- Encapsulación y visibilidad. Ocultación de atributos y ocultación de métodos.
- Tratamiento de la herencia.
- Jerarquía de clases: subclase y superclase.
- Concepto de polimorfismo. Diseño, uso y aplicación.

4.– Reutilizar componentes existentes, seleccionando las opciones técnicas para su desarrollo.

Criterios de valoración:

- a) Se han identificado las ventajas y desventajas de la reutilización de código.
- b) Se han creado rutinas con y sin parámetros.
- c) Se han creado rutinas y subrutinas.
- d) Se han instalado librerías externas.
- e) Se han utilizado métodos de librerías externas.
- f) Se ha accedido a código reutilizable de repositorios para incorporarlo en la aplicación.

Conocimientos:

- Reutilización de código. Ventajas y desventajas.
- Creación de rutinas y subrutinas. Concepto de subrutina y elementos de la declaración de una subrutina.
- Utilización de librerías Externas. Importación de librerías en Java.
- Identificación y características de repositorios de códigos. Acceso a código.

5.– Escribir programas que almacenan y gestionan información, utilizando procedimientos específicos del lenguaje y librerías de clases.

Criterios de valoración:

- a) Se han utilizado ficheros para almacenar y recuperar información.
- b) Se han creado programas utilizando métodos de acceso al contenido de los ficheros.
- c) Se han instalado sistemas gestores de bases de datos orientados a objetos.
- d) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- e) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.
- f) Se han programado conexiones con bases de datos.
- g) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

Conocimientos:

- Ficheros de datos. Tipos, operaciones y modos de acceso. Ficheros secuenciales. Filereader y Filewriter. Buffering. Ficheros binarios. Concepto de registro.
- Manipulación de ficheros de datos utilizando serialización.
- Conexión y acceso a bases de datos orientadas a objetos.
- Herramientas de establecimiento de conexiones a BBDD relacionales: JDBC-ODBC.
- Herramientas de manipulación de datos de una BBDD relacional. SQL.

6.– Crear aplicaciones libres de vulnerabilidades, utilizando herramientas de seguridad.

Criterios de valoración:

- a) Se han identificado los principios básicos de la seguridad en programación.
- b) Se ha efectuado el control de introducción de datos.
- c) Se ha validado la entrada de datos en formularios.
- d) Se han controlado las excepciones producidas por la entrada errónea de datos.
- e) Se han controlado los mensajes emitidos en situaciones erróneas.
- f) Se ha controlado la inyección de código SQL en los campos de texto.

Conocimientos:

- Principios básicos de la seguridad en programación. Análisis de vulnerabilidades. TOP 10 vulnerabilidades.
- Input validation. Control de entradas y datos en formularios.
- Vulnerabilidades: entrada de datos, excepciones e inyección de SQL y código.

7.– Probar las aplicaciones de manera eficiente, utilizando técnicas estándar.

Criterios de valoración:

- a) Se han identificado los tipos de pruebas.
- b) Se ha depurado el código y se ha producido una aplicación estable.
- c) Se ha refactorizado el código.
- d) Se han realizado pruebas manuales de resultado.
- e) Se han utilizado métodos automatizados de prueba.

Conocimientos:

- Tipos de pruebas. Unitarias, de integración, de sistema.
- Técnicas de depuración de código. Refactoring.

- Diseño y ejecución de pruebas manuales. Control de entradas y salidas.
- Herramientas software para la realización de pruebas automatizadas, de gestión de pruebas y de pruebas funcionales.

8.– Generar documentación de aplicación, utilizando productos software adecuados.

Criterios de valoración:

- a) Se ha comentado el código.
- b) Se han realizado los diagramas de clases.
- c) Se han realizado los diagramas de comportamiento.
- d) Se han documentado los constructores y métodos de clase.
- e) Se han utilizado productos software para la automatización de la documentación.

Conocimientos:

- Objetivos de la documentación. Características de los comentarios. Estrategia.
- Diagramas de clases. Estructuración de una aplicación.
- Diagramas de comportamiento. Diseño de la navegación de una aplicación.
- Estrategia de documentación de constructores y métodos.
- Herramientas automatizadas de documentación. Javadoc.

Asociados al ámbito 2: PROGRAMACIÓN GRÁFICA EN JAVA. MULTIPLATAFORMA.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Desarrollar aplicaciones gráficas en Java, utilizando entornos de desarrollo.

Criterios de valoración:

- a) Se han analizado distintos entornos de desarrollo.
- b) Se ha instalado un entorno de desarrollo integrado.
- c) Se han instalado las herramientas necesarias (plugins).
- d) Se ha personalizado un entorno de desarrollo.
- e) Se ha configurado el sistema de actualización del entorno de desarrollo.
- f) Se han generado ejecutables a partir de código fuente.

Conocimientos:

- Entornos de Desarrollo Integrado (IDE) para la creación de aplicaciones gráficas. Concepto y tipos. Eclipse, NetBeans, IntelliJ IDEA...

- Instalación de un entorno de desarrollo para la creación de aplicaciones gráficas: Eclipse.
- Instalación de herramientas (plugins) para la creación de aplicaciones gráficas. WindowBuilder.
- Personalización y configuración de Eclipse.
- Generación de programas ejecutables. Creación de un ejecutable.JAR o.EXE.

2.– Reconocer las interfaces gráficas de usuario, valorando su utilidad en el desarrollo de aplicaciones Java.

Criterios de valoración:

- a) Se han identificado componentes gráficos.
- b) Se han identificado tipos de contenedores gráficos.
- c) Se han identificado componentes atómicos.
- d) Se han identificado componentes de texto.
- e) Se han identificado componentes de generación de menús.
- f) Se han identificado tipos de Layouts.

Conocimientos:

- Estructura y componentes de la biblioteca Swing.
- Iniciación y conocimiento de la plataforma JavaFx.
- Contenedores gráficos: usos y clases. JFrame, JDialog, Panel...
- Componentes atómicos: descripción, creación e instanciación. JLabel, JButton...
- Componentes de texto: usos y finalidades. JTextField, JPasswordField...
- Componentes de generación de menús gráficos. Utilidades. JMenu, JMenuItem, JMenuBar...
- Tipos de Layouts. Especificaciones y usos. AbsoluteLayout, BorderLayout,...

3.– Reconocer los tipos de eventos que se producen en el desarrollo de aplicaciones Java, valorando su utilidad.

Criterios de valoración:

- a) Se han identificado situaciones que provocan eventos.
- b) Se han identificado diferentes tipos de eventos.
- c) Se han identificado los tipos de eventos más usados en Java según los componentes.
- d) Se han identificado eventos comunes a varios componentes.

Conocimientos:

- Eventos. Definición e identificación.

- Tipos de eventos Java según los componentes. ActionListener, FocusListener, ...
- Eventos comunes a varios componentes. ActionListener común a varios botones.

4.– Desarrollar aplicaciones gráficas en Java, usando componentes gráficos y controlando los eventos.

Criterios de valoración:

- a) Se han identificado las funciones del modo Diseño y del modo Código Fuente.
- b) Se han usado Layouts.
- c) Se han usado componentes de texto.
- d) Se han usado componentes de generación de menús.
- e) Se han usado tipos de eventos en aplicaciones gráficas.
- f) Se han usado eventos comunes a varios componentes.
- g) Se han corregido fallos del asistente del modo Diseño revisando el código en el modo Código Fuente.

Conocimientos:

- Modo Diseño. Añadir componentes en modo Diseño. Modo Código Fuente.
- Aplicaciones gráficas en Java. Biblioteca Java Swing.
- Layouts. BorderLayout, AbsoluteLayout...
- Componentes gráficos en aplicaciones gráficas. JFrame, JLabel, JButton...
- Componentes de texto, JTextbox...
- Generación de menús en aplicaciones gráficas. JMenu, JMenuBar, JMenuItem...
- Eventos en aplicaciones gráficas. ActionListener, FocusListener...
- Eventos comunes a varios componentes. ActionListener...
- Corregir los errores del modo Diseño manipulando el Código Fuente.

5.– Generar interfaces gráficas de usuario mediante editores visuales, utilizando las funcionalidades del entorno de desarrollo.

Criterios de valoración:

- a) Se han utilizado los asistentes de un editor visual.
- b) Se han identificado los componentes de un interfaz gráfico.
- c) Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- d) Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.

- e) Se ha analizado y modificado el código generado por el editor visual.
- f) Se han asociado a los eventos las acciones correspondientes.
- g) Se han corregido los errores de código producidos por el editor visual.

Conocimientos:

- Diseño de interfaces gráficas. WindowBuilder.
- Componentes de un interfaz gráfico. JFrame, JLabel, JButton, JTextbox...
- Propiedades de los componentes de un interfaz gráfico. Variable, Text, Title...
- Modo Código Fuente y Modo Diseño.
- Eventos asociados a componentes. ActionListener, FocusListener, ...
- Ubicación de componentes usando el modo Diseño del editor.

6.– Desarrollar aplicaciones que incluyen interfaces gráficas de usuario basados en XML, utilizando herramientas específicas.

Criterios de valoración:

- a) Se han reconocido las ventajas de generar interfaces de usuario a partir de su descripción en XML.
- b) Se ha generado la descripción del interfaz en XML usando un editor gráfico.
- c) Se ha analizado el documento XML generado.
- d) Se ha modificado el documento XML.
- e) Se han asignado acciones a los eventos.
- f) Se ha generado el código correspondiente al interfaz a partir del documento XML.

Conocimientos:

- Herramientas de generación de interfaces de usuario XML. Qt Designer. Qt Jambi.
- Ventajas de generar interfaces de usuario a partir de su descripción en XML.
- Estructura del documento XML que contiene el interfaz de usuario.
- Asignación de acciones a eventos en Qt Jambi. Signals y Slots.

7.– Diseñar interfaces gráficas, aplicando criterios de usabilidad.

Criterios de valoración:

- a) Se han reconocido los beneficios de crear un diseño web con criterios de usabilidad.
- b) Se han creado menús contextuales cuya estructura y contenido siguen los estándares establecidos.

jueves 16 de marzo de 2023

- c) Se han distribuido los controles en la interfaz de usuario.
- d) Se ha utilizado el componente más apropiado en cada caso.
- e) Se han utilizado herramientas de legibilidad.
- f) Se han realizado pruebas para evaluar el interfaz gráfico generado.

Conocimientos:

- Criterios de usabilidad. Simpleza, claridad...
- Menús Contextuales. JPopupMenu.
- Barras de menús. JMenu.
- Barras de Herramientas. JMenuBar.
- Selección y ubicación de componentes. WindowBuilder
- Herramientas de legibilidad: flesh, inflesz, readability index calculator
- Generación de mensajes: extensión y usabilidad.
- Pruebas de usabilidad. Card Sorting, Tree test...

8.– Desarrollar aplicaciones para dispositivos móviles, empleando tecnologías y librerías específicas.

Criterios de valoración:

- a) Se ha generado la estructura de clases necesaria para la aplicación.
- b) Se han analizado y utilizado las clases que modelan ventanas, menús, alertas y controles.
- c) Se han utilizado las clases necesarias para la conexión y comunicación con dispositivos inalámbricos.
- d) Se han utilizado las clases necesarias para el intercambio de mensajes de texto y multimedia.
- e) Se han utilizado las clases necesarias para establecer conexiones y comunicaciones HTTP y HTTPS.
- f) Se han utilizado las clases necesarias para establecer conexiones con almacenes de datos, garantizando la persistencia.
- g) Se han realizado pruebas de interacción usuario-aplicación.
- h) Se han empaquetado y desplegado las aplicaciones desarrolladas en dispositivos móviles reales.

Conocimientos:

- Estructura de clases de una aplicación. Android Manifest. Directorios SRC, RES, BIN...
- Generación de ventanas, menús, alertas y controles. Android Studio.

- Conexión y comunicación con dispositivos inalámbricos. API inalámbricas de Android.
- Intercambio de mensajes de texto y multimedia. Android Sharesheet.
- Conexiones y comunicaciones HTTP y HTTPS. HttpURLConnection.
- Conexiones con almacenes de datos. API de SharedPreferences.
- Pruebas de interacción usuario-aplicación. Optimización de aplicaciones a partir de emuladores. BlueStacks.
- Empaquetado y despliegue de una aplicación. Generación de una APK Android.

Asociados al ámbito 3: PROGRAMACIÓN JAVA ENTORNO CLIENTE.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Seleccionar la arquitectura y el lenguaje de programación web entorno cliente, considerando las necesidades del proyecto.

Criterios de valoración:

- a) Se han descrito las arquitecturas utilizadas en la programación web en entorno cliente.
- b) Se han clasificado los lenguajes de programación web entorno servidor.
- c) Se han reconocido las necesidades de un proyecto.

Conocimientos:

- Arquitecturas de programación web entorno cliente: Monolítica, N-capas, mono-servidor, multi-servidor. Características diferenciadoras, ventajas y desventajas.
- Lenguajes de programación web entorno cliente: propósito, paradigma, popularidad, última revisión y características particulares.
- Necesidades de proyectos. Selección de arquitectura y lenguaje de programación web entorno cliente.

2.– Generar código y funcionalidades básicas, utilizando un lenguaje de programación web entorno cliente.

Criterios de valoración:

- a) Se han desarrollado e invocado funciones básicas.
- b) Se han manejado eventos de una aplicación.
- c) Se ha buscado y manipulado en un árbol Document Object Model (DOM).
- d) Se han realizado llamadas Hypertext Transfer Protocol (HTTP) asíncronas.
- e) Se ha recogido y tratado distintos tipos de respuestas de peticiones HTTP asíncronas.
- f) Se han depurado códigos escritos en un lenguaje de programación web entorno cliente.

Conocimientos:

- Desarrollo de funciones básicas: uso de operadores, declaraciones y control de flujo; e invocación de estas en el código principal.
- Manejo de eventos de una aplicación: onclick, onfocus, onkeydown, etc.
- Búsqueda y manipulación de un árbol DOM: JQuery, movimiento entre nodos, edición de nodos, innerHTML y eliminación de nodos.
- Desarrollo de peticiones HTTP asíncronas: AJAX.
- Obtención y trato de respuestas de peticiones HTTP asíncronas: desglose de mensaje HTTP y Json.
- Depuración de códigos escritos con herramientas del desarrollador de un navegador web.

3.– Desarrollar objetos, utilizando un lenguaje de programación web entorno cliente.

Criterios de valoración:

- a) Se han diseñado y programado objetos.
- b) Se han construido objetos de múltiples maneras.
- c) Se han desarrollado funciones para interactuar con objetos.
- d) Se han desarrollado herencias de objetos.
- e) Se ha interactuado entre instancias de objetos y herencias.
- f) Se han seleccionado los accesos a las funciones y propiedades.
- g) Se han creado y manipulado conjuntos de objetos.

Conocimientos:

- Desarrollo de objetos con sus constructores, atributos y funciones.
- Construcción y declaración de objetos: constructor y Json.
- Desarrollo de funciones en los objetos, funciones como atributos de objetos.
- Desarrollo de herencias de objetos: prototype y polimorfismo.
- Interacción entre objetos y objetos y sus herencias.
- Accesos de funciones y propiedades: privateField, privateMethod, PRIVATE_STATIC_FIELD, privateStaticMethod, protected (#).
- Creación y manipulación de conjuntos de objetos: arrays, listas, sets... Aplicación de los conceptos de programación orientada a objetos.

4.– Desarrollar proyectos en un lenguaje de programación web entorno cliente, utilizando librerías de este lenguaje.

Criterios de valoración:

- a) Se han identificado las librerías más populares de un lenguaje.
- b) Se han seleccionado las librerías necesarias para cada proyecto.
- c) Se han instalado y configurado las librerías en un proyecto.
- d) Se han utilizado las librerías necesarias para cada proyecto.

Conocimientos:

- Búsqueda e identificación de las librerías más populares de javascript: jquery, D3.js, wForms... Descripción por propósito, soporte y características particulares.
- Selección de librerías para cada proyecto según requisitos.
- Entorno de desarrollo. Instalación y configuración de librerías en un proyecto.
- Uso de librerías: búsqueda de documentación, identificación de la información precisa, lectura y aplicación.

5.– Desarrollar una presentación de capa gráfica mediante un prototipo interactivo.

Criterios de valoración:

- a) Se han descrito los patrones básicos de diseño.
- b) Se han realizado bocetos desde una descripción dada.
- c) Se han identificado los errores de diseño.
- d) Se han realizado cambios de diseño.
- e) Se ha utilizado una herramienta para el desarrollo de prototipos de aplicaciones gráficas.

Conocimientos:

- Descripción de patrones básicos de diseño de aplicaciones gráficas:
 - Colocación de información según relevancia.
 - Colocación de botones según funcionalidad.
 - Creación de ventanas según tipo de actividad, conjunto de colores, etc. Ventanas emergentes, pestañas...
- Desarrollo de bocetos de una aplicación gráfica desde una descripción dada. Diseño de: pantalla, ventana, botón, cuadro de texto, enlace, etc.
- Identificación de errores de diseño: colocación de elementos, intuitividad de la aplicación, contraste de colores, coherencia con el propósito de la aplicación, etc.

jueves 16 de marzo de 2023

- Realización de cambios a un diseño de aplicaciones gráficas.

- Herramientas: Figma, Penpot, draw.io, Lucidchart, etc.

6.– Desarrollar hojas de estilos para las aplicaciones web de entorno cliente, utilizando librerías orientadas a estas.

Criterios de valoración:

- a) Se han desarrollado e importado hojas de estilo.
- b) Se han identificado las librerías más populares para hojas de estilo.
- c) Se han seleccionado las librerías y se han aplicado estilos según requisitos del proyecto.
- d) Se han instalado y configurado librerías para hojas de estilos.
- e) Se han utilizado librerías para hojas de estilos.

Conocimientos:

- Desarrollo e inclusión de ficheros de estilos: CSS...
- Identificación de las librerías más populares de hojas de estilo y describirlas por: soporte, propósito y características distintivas.
- Selección de librerías. Aplicación de estilos según requisitos del proyecto.
- Entorno de desarrollo. Instalación y configuración de librerías para hojas de estilo en un proyecto.
- Uso de librerías para hojas de estilo: búsqueda de documentación, identificación de la información precisa, lectura y aplicación.

7.– Desarrollar aplicaciones web entorno cliente, utilizando Frameworks como apoyo para estos.

Criterios de valoración:

- a) Se han identificado los Frameworks más populares.
- b) Se han seleccionado los Frameworks necesarios.
- c) Se ha instalado y configurado un Framework en un proyecto.
- d) Se han utilizado Frameworks en un proyecto.

Conocimientos:

- Búsqueda e identificación de Frameworks para desarrollo de aplicaciones web entorno cliente: Angular, vue.js, react, NodeJS...
- Necesidades del proyecto. Características de Framework: requisitos, propósito, características particulares, tipo de software, mantenimiento/soporte... Selección de Frameworks.
- Instalación y configuración de un Framework en un proyecto.
- Uso de Framework: búsqueda de documentación, identificación de la información precisa, lectura y aplicación.

Asociados al ámbito 4: PROGRAMACIÓN JAVA EN ENTORNO SERVIDOR.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Seleccionar arquitecturas de programación web entorno servidor, considerando las necesidades del proyecto.

Criterios de valoración:

- a) Se ha caracterizado la arquitectura Modelo Vista Controlador (MVC).
- b) Se ha analizado el propósito de cada entidad de la arquitectura MVC.
- c) Se han descrito distintos patrones de diseño web.
- d) Se han descrito distintas arquitecturas de un programa de servidor.

Conocimientos:

- Caracterización del MVC. Descripción y relación entre cada entidad.
- Análisis del propósito. Modelo, vista y controlador.
- Patrones de diseño: MVVC, MVP.
- Arquitecturas web de un servidor: SPA, microservicios, arquitectura serverless, P2P.

2.– Interactuar con los protocolos HyperText Transfer Protocol (HTTP) y HyperText Transfer Protocol Secure (HTTPS), reconociendo los métodos, códigos de respuesta y estructura del mensaje.

Criterios de valoración:

- a) Se han descrito las características básicas de los protocolos HTTP y HTTPS.
- b) Se ha reconocido la estructura del mensaje.
- c) Se ha extraído información de un mensaje.
- d) Se ha desglosado y analizado una url.

Conocimientos:

- Características básicas de los protocolos HTTP y HTTPS:
 - Puerto.
 - Métodos GET, POST, PUT y DELETE.
 - Conjuntos de códigos de respuesta: 1xx, 2xx, 3xx, 4xx y 5xx.
 - Códigos de respuesta más habituales.
- Estructuras de los mensajes: Cabecera de petición (Método, url, versión y host), cabecera de respuesta y cuerpo.
- Extracción de información de mensajes de petición y respuesta.
- Desglose y análisis de una url: host, dirección de petición, parámetros enviados por url.

3.– Desarrollar aplicaciones web básicas, utilizando Servlets de J2EE.

Criterios de valoración:

- a) Se ha configurado el entorno de desarrollo.
- b) Se ha descrito el ciclo de vida de un Servlet.
- c) Se han reconocido las funciones de Servlet.
- d) Se ha configurado la aplicación y su contexto.
- e) Se ha declarado un Servlet.
- f) Se ha desarrollado un Servlet que escuche peticiones HTTP.
- g) Se han utilizado las funciones de objetos request y response.
- h) Se han manejado los eventos del ciclo de vida de los Servlet.
- i) Se han manejado errores y excepciones producidas por Servlet.

Conocimientos:

- Configuración del entorno de desarrollo Integrated Development Environment (IDE) con el servicio HTTP.
- Fases y actores del ciclo de vida de un Servlet.
 - Fases: inicio, destrucción, peticiones.
 - Actores: cliente, contenedor; y variables del contenedor.
- Funciones más utilizadas de Servlet: init, destroy, do<Método>.
- Configuración de la aplicación y su contexto: web.xml y context.xml.
- Declaración de un servlet en el web.xml.
- Desarrollo de una clase que herede HttpServlet: métodos init, destroy y do<Método>.
- Funciones de objetos request y response de Servlet: HttpServletRequest, HttpServletResponse, redirect, dispatch, etc.
- Manejo de eventos del ciclo de vida de los Servlet: Listeners.
- Manejo de excepciones: ServletException, IOException; y errores: variables de entrada o salida inválidas, campos vacíos, etc.

4.– Gestionar la sesión y las cookies de una aplicación, utilizando Cookie y HttpSession de J2EE.

Criterios de valoración:

- a) Se han descrito los tipos de sesión.
- b) Se han construido y manipulado objetos de sesión y se ha configurado su ciclo de vida.

jueves 16 de marzo de 2023

- c) Se han manejado eventos de una sesión.
- d) Se ha interactuado con el objeto de una sesión desde un Servlet.
- e) Se ha descrito la estructura de una cookie.
- f) Se han clasificado las cookies en función de los valores de sus atributos.
- g) Se han construido y modificado cookies.
- h) Se han enviado y recibido cookies desde un servlet.

Conocimientos:

- Descripción de tipos de sesión: cookies, url rewriting, hidden fields.
- Construcción y manipulación de objetos de sesión: objeto HttpSession. Edición del ciclo de vida de una sesión en el web.xml.
- Manejo de eventos de una sesión: Listeners.
- Interacción con el objeto sesión: construcción, modificación, obtención de atributos.
- Estructura de una cookie: nombre, valor y atributos (domain, path, expires, max-age, httponly, secure).
- Atributos y clasificación de cookies: de sesión, persistentes, seguras, Httponly, super-cookies, zombie-cookie, first-party/third-party.
- Construcción y modificación de cookies: javax.servlet.http.Cookie.
- Envío y recepción de cookies: construcción de cookie, recepción de cookies en el Servlet, almacenamiento de cookies en el cliente.

5.– Desarrollar páginas web dinámicas, utilizando Java Server Pages (JSP).

Criterios de valoración:

- a) Se ha descrito el ciclo de vida de los JSP.
- b) Se ha desarrollado un JSP con sus elementos básicos.
- c) Se ha incluido un JSP en un Servlet.
- d) Se han desarrollado formularios en JSP.
- e) Se ha obtenido información desde un JSP.
- f) Se han utilizado objetos en el JSP.
- g) Se han manejado errores y excepciones en un JSP.

Conocimientos:

- Ciclo de vida de JSP: compilación, análisis, conversión a Servlet, compilación de Servlet, inicialización, implementación y destrucción.

- Elementos básicos de un JSP: elementos HyperText Markup Language 5 (HTML5), comentarios JSP, scriptlets JSP, directivas y acciones.
- Inclusión de un JSP en un Servlet: ServletDispatcher y ServletResponse.
- Desarrollo de formularios en JSP. Recogida de información: cajas de texto, seleccionador, calendario, etc.
- Obtención de información desde un JSP: URI/URL, cuerpo de HTTP, parámetros del Servlet, propiedades del web.xml, etc.
- Uso de objetos en el JSP: JavaBeans.
- Manejo de errores: variables incorrectas o inexistentes y excepciones producidas por objetos de Java.

6.– Desarrollar JSPs con objetos, utilizando Lenguaje de Expresiones (EL).

Criterios de valoración:

- a) Se ha interactuado con objetos.
- b) Se han obtenido cabeceras, parámetros de la petición y parámetros del contexto.
- c) Se han manipulado los valores de los tipos.
- d) Se ha accedido a la información de errores y excepciones.

Conocimientos:

- Interacción de objetos. Obtención de atributos, elementos de listas, valores de mapas. Acceso a métodos de objetos.
- Cabecera, parámetros de la petición y contexto: header, headeValues, cookie, initParam, pageContext.
- Manipulación de valores de tipos: descripción de los tipos, mezcla de valores de distintos tipos y conversión de tipos.
- Acceso a información de errores y excepciones: status_code, exception_type, message, request_uri, exception.

7.– Desarrollar algoritmos en un JSP, utilizando Java Standard Tag Library (JSTL).

Criterios de valoración:

- a) Se ha instalado la librería JSTL.
- b) Se han analizado los distintos conjuntos de etiquetas de JSTL.
- c) Se ha controlado el flujo del algoritmo.
- d) Se han gestionado las variables.
- e) Se han importado otros ficheros.

- f) Se han manejado excepciones.
- g) Se han añadido paquetes de idiomas.
- h) Se ha localizado la ejecución de la aplicación.
- i) Se han formateado fechas y valores.

Conocimientos:

- Instalación de la librería JSTL: inserción de los ficheros.jar en el proyecto.
- Análisis de las sub librerías de JSTL: Core, I18n & Formatting, Sql.
- Control de flujo del algoritmo: condicionales e iteraciones.
- Gestión de variables: definición de estructura HTML y asignación de valores a propiedades de objetos.
- Importación de ficheros: c:import.
- Manejo de excepciones: c:catch.
- Añadido de paquetes de idiomas: JSTL: I18n.
- Localización de la ejecución de una aplicación: java.util.Locale.
- Formateo de fechas y valores: fmt:formatDate, TimeZone, parseDate, etc. Valores: fmt:formatNumber.

8.– Desarrollar filtros para una aplicación web, utilizando objetos de J2EE.

Criterios de valoración:

- a) Se ha descrito un filtro.
- b) Se ha construido un filtro.
- c) Se ha implementado una cadena de filtros.
- d) Se ha declarado un filtro.
- e) Se ha desarrollado un control de acceso.
- f) Se ha desarrollado un filtro de compresión de respuestas.

Conocimientos:

- Descripción de un filtro:
 - Propósito: bloqueo de acceso, transformación de la respuesta.
 - Caracterización de la función doFilter de javax.servlet.Filter.
 - Cadenas de filtros.

jueves 16 de marzo de 2023

- Construcción de un filtro. Implementación `javax.servlet.Filter`. Desarrollo de las funciones `doFilter`, `init` y `destroy`.
- Implementación de una cadena de filtros.
- Declaración de filtros: `web.xml`.
- Desarrollo de un control de acceso.
- Desarrollo de un filtro de compresión de respuestas: `GZIPOutputStream`.

9.– Aplicar seguridad declarativa a una aplicación web, utilizando HTTPS.

Criterios de valoración:

- a) Se han obtenido claves RSA.
- b) Se ha configurado el servicio.
- c) Se ha configurado la aplicación.

Conocimientos:

- Obtención de claves RSA: `OpenSSL` o `JSSE`.
- Configuración de un servicio para aceptar peticiones HTTPS.
- Configuración de aplicación para que recursos específicos se pidan por HTTPS: `<transport-guarantee>`

Asociados al ámbito 5: PERSISTENCIA DE INFORMACIÓN EN JAVA.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Realizar operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

Criterios de valoración:

- a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- b) Se han utilizado formatos en la visualización de la información.
- c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.
- d) Se han utilizado ficheros de texto para almacenar y recuperar información.
- e) Se han utilizado ficheros binarios para almacenar y recuperar información.
- f) Se ha usado serialización de objetos para almacenar y recuperar información.
- g) Se han desarrollado aplicaciones para almacenar y recuperar información de ficheros.
- h) Se han desarrollado aplicaciones que controlan las excepciones durante los procesos de almacenamiento y recuperación de información de ficheros.

Conocimientos:

- Comando java en línea de comandos.
- Entrada y Salida de información por los dispositivos estándar. System.out.
- Entrada y Salida de información con formato. System.out.printf
- Entrada y Salida de información en ficheros de texto. FileWriter, PrintWriter, BufferedWriter, FileReader, BufferedReader, readLine
- Entrada y Salida de información en ficheros binarios. FileOutputStream, BufferedOutputStream, write, FileInputStream, BufferedInputStream, read.
- Acceso aleatorio en ficheros. RandomAccessFile, seek.
- Entrada y Salida de información de objetos serializados en ficheros. Serializable, serialVersionUID. FileOutputStream, ObjectOutputStream, writeObject, FileInputStream, ObjectInputStream, readObject, ByteArrayOutputStream.
- Control de Excepciones de Entrada / Salida. IOException, FileNotFoundException, ClassNotFoundException.

2.– Desarrollar aplicaciones Java, utilizando bases de datos relacionales y manteniendo la integridad y consistencia de los datos.

Criterios de valoración:

- a) Se han reconocido las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
- b) Se han instalado sistemas gestores de bases de datos relacionales.
- c) Se han desarrollado aplicaciones Java utilizando JDBC.
- d) Se han desarrollado aplicaciones Java que manipulan datos obtenidos desde bases de datos relacionales.
- e) Se han desarrollado aplicaciones Java que gestionan la información estructural de bases de datos relacionales.
- f) Se han desarrollado aplicaciones Java que controlan las excepciones generadas por el trabajo con bases de datos relacionales.
- g) Se han reconocido las características de los procedimientos almacenados en bases de datos relacionales (PL/SQL).
- h) Se han desarrollado aplicaciones Java que ejecutan procedimientos almacenados en bases de datos relacionales (PL/SQL).

Conocimientos:

- Sistemas Gestores de Bases de Datos Relacionales. Definición y tipos. MySQL, Oracle...
- Instalación de Sistemas Gestores de Bases de Datos Relacionales. Instalación de MySQL.

- Instalación de un driver JDBC (MySQL Connector).
- JDBC:
 - Comunicación con bases de datos relacionales.
 - Almacenamiento de la información en bases de datos relacionales.
 - Ejecución de consultas en bases de datos relacionales y muestra de la información.
- Desarrollo de aplicaciones Java con acceso a Bases de Datos Relacionales.
- Control de Excepciones generadas al trabajar con Bases de Datos Relacionales. SQLException.
- Desarrollo de aplicaciones Java que ejecutan consultas de inserción, borrado, modificación... en Bases de Datos Relacionales.
- Procedimientos almacenados en bases de datos relacionales (PL/SQL).
- Aplicaciones Java que ejecutan procedimientos almacenados en bases de datos relacionales (PL/SQL).

3.– Desarrollar aplicaciones Java utilizando Bases de Datos Orientadas a Objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

Criterios de valoración:

- a) Se han reconocido las características de las Bases de Datos Orientadas a Objetos.
- b) Se han reconocido las características de Java Persistence API (JPA).
- c) Se han instalado sistemas gestores de Bases de Datos Orientados a Objetos.
- d) Se han creado bases de datos y estructuras necesarias para el almacenamiento de objetos.
- e) Se han reconocido las características del lenguaje de consulta Java Persistence Query Language (JPQL).
- f) Se han utilizado herramientas que permiten almacenar objetos en Bases de Datos Orientadas a Objetos.
- g) Se han utilizado herramientas que permiten recuperar, actualizar y eliminar objetos de Bases de Datos Orientadas a Objetos.
- h) Se han utilizado herramientas que permiten almacenar tipos de datos estructurados, compuestos y relacionados.
- i) Se han utilizado herramientas que permiten controlar las excepciones.

Conocimientos:

- Bases de Datos Orientadas a Objetos. Definición y tipos. ObjectDB, Db4o...
- JPA. Paquete javax.persistence.

jueves 16 de marzo de 2023

- ObjectDB. @Entity, EntityManager, EntityTransaction, EntityManagerFactory...
- Aplicaciones Java con acceso a Bases de Datos Orientadas a Objetos mediante ObjectDB.
- JPQL. Consultas.
- Control de Excepciones. IllegalArgumentException, TransactionRequiredException, EntityExistsException...

4.– Desarrollar aplicaciones avanzadas Java, utilizando un Object Relational Mapping (ORM) como capa de persistencia de una aplicación.

Criterios de valoración:

- a) Se han reconocido y etiquetado las entidades de una aplicación Java.
- b) Se han reconocido las características de un ORM.
- c) Se ha configurado un ORM.
- d) Se ha creado una base de datos a través de un código etiquetado.
- e) Se ha obtenido el código de las entidades implicadas en una base de datos.
- f) Se han desarrollado aplicaciones Java que usan un ORM para crear, acceder y consultar objetos de las bases de datos creadas.

Conocimientos:

- Entidades de una aplicación Java. Estados de las entidades: desasociado (detached), efímero (transient) y persistido (persistent).
- Características de un ORM. Mapeo entre objetos y tablas de bases de datos relacionales.
- Ejemplo de un ORM. Hibernate.
- Configuración de Hibernate.
- Creación de una base de datos en Hibernate.
- Estructura de una base de datos en Hibernate.
- Hibernate. Acceso a bases de datos desde aplicaciones Java.

Asociados al ámbito 6: FRAMEWORKS DE DESARROLLO.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Seleccionar Frameworks Java en entorno servidor, analizando su potencial.

Criterios de valoración:

- a) Se han identificado las características de los Frameworks de desarrollo de Java.
- b) Se han reconocido las ventajas de la programación mediante objetos sencillos (Plain Old Java

Objects).

c) Se han reconocido las funcionalidades que aportan los Frameworks de desarrollo.

d) Se han descrito los distintos tipos de contenedores y sus características.

e) Se han identificado los patrones de diseño fundamentales del Framework.

f) Se han diferenciado las distintas formas de Inyección de Dependencias.

Conocimientos:

- Frameworks de desarrollo y sus características: Spring, Struts, JSF, GWT, Grails...
- Ventajas sobre EJBs, independencia sobre APIs, simplificación de las pruebas de código.
- Manejo del ciclo de vida, resolución de dependencias, búsqueda de componentes, configuración de la aplicación, manejo de conexiones a bases de datos...
- Contenedores ligeros (Spring Container) / pesados (Servidores de aplicaciones Java).
- Inversión de Control e Inyección de Dependencias.
- Inyección de constructor y de método (setter).

2.– Desarrollar aplicaciones Java en entorno servidor, considerando las especificaciones del Framework.

Criterios de valoración:

a) Se ha creado un proyecto del Framework utilizando las herramientas apropiadas.

b) Se han utilizado distintos métodos para establecer los metadatos de configuración.

c) Se ha analizado el sistema de resolución de dependencias.

d) Se han creado programas de consola que usan los distintos tipos de inyección de dependencias.

e) Se han creado programas mediante la Inyección de Dependencias automáticas.

f) Se han creado Beans gestionados por el Framework.

g) Se han creado Beans de acuerdo con su ciclo de vida (visibilidad).

Conocimientos:

- Creación de proyectos con Maven, Gradle o similar.
- Gestión de archivos de configuración: XML, anotaciones y basados en Java.
- Dependencias Maven, archivo pom.xml.
- Setter injection, Constructor injection, atributo depends-on.
- Autowiring y distintos tipos: byType, byName, y constructor.

- Búsquedas mediante el contexto de aplicación.
- Ciclos de vida: singleton, prototype, request y session.

3.– Construir aplicaciones web, siguiendo el patrón Modelo-Vista-Controlador (MVC).

Criterios de valoración:

- a) Se han analizado las características y beneficios del modelo MVC.
- b) Se han utilizado objetos y controles en el servidor para generar el aspecto visual de la aplicación web en el cliente.
- c) Se han utilizado controladores para recoger las peticiones recibidas en el servidor.
- d) Se han recogido los datos a través de formularios para responder a los eventos de la aplicación web.
- e) Se han validado los datos recibidos a través de formularios y se han mostrado los errores de validación.
- f) Se han identificado y aplicado los parámetros relativos a la configuración de la aplicación web.
- g) Se han escrito aplicaciones web con mantenimiento de estado y separación de la lógica de negocio.
- h) Se ha probado y documentado el código.

Conocimientos:

- Teoría MVC.
- Configuración del Servlet que funciona como front-controller: declaración y definición de mappings mediante XML o anotaciones.
- Creación de vistas, modelos y controladores.
- Creación de controles en los formularios para recoger datos desde la vista, validación de los datos recibidos a través de formularios.
- Control de sesión y cookies.
- Definición Servlets y acceso al `WebApplicationContext`.
- Elaboración de pruebas unitarias, utilizando objetos de prueba (mocks).

4.– Configurar la aplicación web, integrándola con el sistema operativo y su entorno.

Criterios de valoración:

- a) Se han reconocido los elementos de configuración de una aplicación.
- b) Se ha analizado el entorno de una aplicación web.
- c) Se ha configurado un origen de datos.
- d) Se ha establecido la configuración del servidor.

jueves 16 de marzo de 2023

e) Se han configurado los logs.

f) Se han creado clases para establecer la configuración de la aplicación.

Conocimientos:

- Bean wiring/Property wiring.
- Abstracción del entorno, propiedades del sistema de la Java Virtual Machine (JVM).
- Variables de entorno del sistema operativo.
- Argumentos de línea de comando.
- Configuración de logs (Logback, log4j...).
- Ficheros de configuración de propiedades de la aplicación.

5.– Desarrollar aplicaciones de acceso a almacenes de datos, aplicando medidas para mantener la integridad de la información en base a las características del Framework.

Criterios de valoración:

a) Se han analizado las tecnologías que permiten el acceso mediante programación a la información disponible en almacenes de datos.

b) Se han creado aplicaciones que establecen conexiones con bases de datos.

c) Se ha insertado, recuperado, actualizado y eliminado información almacenada en bases de datos.

d) Se ha realizado una gestión adecuada de las excepciones y errores de las operaciones sobre bases de datos.

e) Se han utilizado conjuntos de datos para almacenar la información.

f) Se han ejecutado procesos de base de datos desde la aplicación Java.

g) Se ha integrado la aplicación con una implementación del estándar JPA.

h) Se ha insertado, recuperado, actualizado y eliminado información almacenada en bases de datos utilizando un Framework JPA.

Conocimientos:

- Estudio de distintas tecnologías en función del Framework: JDBC, Spring JDBC...
- Acceso a datos utilizando las herramientas JDBC del Framework.
- Creación de conexiones a base de datos utilizando DataSource o DriverManager.
- Operaciones de lectura, escritura, modificación y borrado de datos mediante JDBC.
- Abstracción de la jerarquía de clases de excepciones y errores (DataAccessException...).
- Encapsulación de consultas SQL en objetos Java.

jueves 16 de marzo de 2023

- Llamamiento a procedimientos almacenados desde Java (SimpleJdbcCall...).
- Integración y configuración de una implementación JPA con el Framework.
- Creación de las clases necesarias para las operaciones con la base de datos, usando la implementación JPA.

6.– Desarrollar servicios web, implantando la estructura de sus componentes y utilizando un Framework de desarrollo de Java.

Criterios de valoración:

- a) Se ha programado un servicio web REST.
- b) Se han utilizado varios códigos de estado en la respuesta del servicio.
- c) Se han configurado servicios web mediante anotaciones.
- d) Se han programado servicios que envían datos en formatos distintos.
- e) Se han gestionado las excepciones del servicio web.
- f) Se ha verificado el funcionamiento del servicio web.
- g) Se ha programado el código de cliente para consumir el servicio web.

Conocimientos:

- Creación de un servicio REST usando los distintos métodos de HTTP y utilizando la configuración XML.
- Códigos de estado para la respuesta HTTP.
- Anotaciones en sustitución de XML.
- Respuestas en formato JSON y XML.
- Lógica para la gestión de excepciones en el código del servicio y envío del mensaje de error.
- Ejecución de Tests Unitarios de servicios mediante Junit.
- Código Javascript/librería Javascript.

7.– Desarrollar aplicaciones web, aplicando el paradigma de la Programación Orientada a Aspectos (POA).

Criterios de valoración:

- a) Se han analizado las diferencias entre la Programación Orientada a Aspectos (POA) y la Programación Orientada a Objetos (POO) y el modo en que pueden complementarse.
- b) Se han descrito los conceptos básicos de la POA.
- c) Se han programado consejos.
- d) Se han utilizado distintos métodos para designar puntos de corte.

e) Se han identificado diversas formas de declarar aspectos.

f) Se han programado los distintos objetos que intervienen en la POA.

g) Se ha configurado POA mediante anotaciones.

Conocimientos:

- Descripción de la filosofía de la POA y sus diferencias, ventajas y desventajas respecto a la POO. Complementación.
- Conceptos básicos: aspecto, punto de corte, consejo, introducción, modelo, consejero.
- Tipos de consejos (Advice): anterior, posterior, anterior y posterior...
- Puntos de corte según tipo, método...
- Uso de XML y de anotaciones para la implementación de POA.
- Uso del paradigma de POA.
- Uso de anotaciones como método de configuración de POA.

Asociados al ámbito 7: DESARROLLOS AVANZADOS.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Mejorar la seguridad de una aplicación web, protegiendo las peticiones web.

Criterios de valoración:

a) Se han descrito los principales vectores de ataque a una aplicación web.

b) Se ha analizado la gestión de sesiones en servidor y en cliente.

c) Se ha añadido una librería que permite gestionar la seguridad.

d) Se ha creado un modelo de usuario basado en las necesidades de seguridad del Framework.

e) Se han elaborado controles que permitan a los usuarios registrarse, acceder y abandonar la aplicación web.

f) Se han establecido mecanismos de preservación de la privacidad del usuario mediante el cifrado de contraseñas.

g) Se han implementado mecanismos de protección contra ataques de falsificación de petición en sitios cruzados (CSRF).

Conocimientos:

- Descripción de la seguridad informática. Vulnerabilidades de una aplicación web: autenticación y autorización, fijación de sesión, XSS (Cross Site Scripting)...
- Sesiones en el servidor en aplicaciones web monolíticas y sesiones en el cliente. Frameworks: Angular, Vue.js o React.js... Beneficios de Oauth frente a HTTP Basic.

jueves 16 de marzo de 2023

- Gestión de dependencias. Archivo pom.xml de Maven. Otras herramientas en Spring Security: spring-boot-starter-web y spring-boot-starter-security.
- Definición de usuario y gestión de los atributos de la clase. Framework de trabajo en Spring: contrato UserDetails, en Laravel, clase por defecto User.
- Interacción con el usuario: formularios y recuperación de la información en los formularios.
- Funciones Hash. Información en la base de datos. Cifrado de contraseñas.
- Cross-Site Request Forgery. Protecciones contra CSRF del Framework (campo «_csrf» en Spring Security, directiva @CSRF en Laravel...).

2.– Establecer mecanismos de autenticación y control de acceso de los usuarios, utilizando las funcionalidades de un Framework de desarrollo.

Criterios de valoración:

- a) Se han analizado tecnologías y mecanismos del Framework que permiten gestionar los accesos.
- b) Se han creado autorizaciones y roles para dividir los permisos de acceso a los distintos apartados de la aplicación.
- c) Se han asegurado las rutas a las que acceden las peticiones mediante uso de autorizaciones.
- d) Se han asegurado las rutas a las que acceden las peticiones mediante uso de roles.
- e) Se ha denegado, a todas las peticiones, el acceso a algunas rutas.
- f) Se han agrupado las rutas de acceso en función de los permisos.

Conocimientos:

- Proceso de autenticación y autorización y sus interacciones. Clases y ficheros implicados: mappings, controladores, servicios y vistas. Autorización por permisos (puertas o autoridades) o por roles (políticas).
- Creación de autorizaciones (o puertas) y roles de usuario para separar niveles de acceso a la aplicación.
- Mecanismos de control de acceso a las rutas de la aplicación mediante autorizaciones (en Spring Security contrato GrantedAuthority métodos Authority, Gates de Laravel...).
- Mecanismos de control de acceso a las rutas de la aplicación (en Spring Security contrato GrantedAuthority métodos Role, Gates & Policies de Laravel...).
- Denegación de acceso a rutas (en Spring Security método denyAll).
- Control del acceso a grupos de rutas (en Spring Security: MVC, Ant y Regex matchers, en Laravel grupos de rutas y middleware).

3.– Consumir servicios web, usando las herramientas de un Framework.

Criterios de valoración:

- a) Se han descrito los tipos de clientes de servicios web.
- b) Se han creado objetos capaces de consumir servicios web y se han analizado sus métodos.
- c) Se han realizado peticiones GET y se ha recogido la respuesta de un servicio web.
- d) Se han realizado peticiones POST y se ha recogido la respuesta de un servicio web.
- e) Se han realizado peticiones PUT y se ha recogido la respuesta de un servicio web.
- f) Se han realizado peticiones DELETE y se ha recogido la respuesta de un servicio web.
- g) Se ha consumido un servicio que cumple con el principio HATEOAS.

Conocimientos:

- Tipos de clientes: síncronos y asíncronos; sencillos o que siguen la filosofía HATEOAS (Hypermedia As The Engine Of Application State: hipermedia como motor del estado de la aplicación).
- Análisis de la clase RestTemplate. Creación de objetos y uso de sus métodos principales.
- Peticiones para consumir un recurso:
 - GET. Método getObject de RestTemplate en Spring.
 - POST. Método postForObject de RestTemplate en Spring.
 - PUT. Método put de RestTemplate en Spring.
 - DELETE. Método delete de RestTemplate en Spring.
 - Otros similares en otro Framework.
- HATEOAS: Hypermedia as the Engine of Application State, en español, hipermedia como motor del estado de la aplicación. Consumo de servicios web mediante herramientas capaces de entender la estructura HATEOAS (Traverson en Spring).

4.– Desarrollar filtros que manejan las peticiones HTTP, aplicando los mecanismos establecidos por el Framework.

Criterios de valoración:

- a) Se han identificado las ventajas de los filtros para la gestión de peticiones.
- b) Se han analizado y caracterizado tecnologías y mecanismos que permiten crear filtros.
- c) Se han clasificado los tipos de filtros existentes atendiendo a su modo de funcionamiento.
- d) Se han añadido filtros al principio de una cadena de filtros.
- e) Se han añadido filtros en medio de una cadena de filtros existente.
- f) Se han creado filtros que reemplazan otros filtros en la misma posición de la cadena.

Conocimientos:

- Lógica de negocio antes de pasar la petición a un controlador (comprobaciones de seguridad).
- Tecnologías y mecanismos para la creación de filtros. Interfaces HandlerInterceptor o Filter en Spring. Middleware en Laravel.
- Cadenas de filtros y su funcionamiento:
 - Filtros que se ejecutan antes de enviar la petición a un controlador o después de ejecutar el controlador.
 - Tipos de filtro en función de su localización en la cadena de filtros.
- Creación de filtros que se ejecutan:
 - Al principio de una cadena de filtros (en Spring Security implementando la interfaz Filter y utilizando WebSecurityConfigurerAdapter).
 - En medio de una cadena de filtros (en Spring Security implementando la interfaz Filter y utilizando WebSecurityConfigurerAdapter, método addFilterAfter).
 - En lugar de otro filtro (en Spring Security implementando la interfaz Filter y utilizando WebSecurityConfigurerAdapter, método addFilterAt).

5.– Comunicar aplicaciones, enviando mensajes asíncronos.

Criterios de valoración:

- a) Se han analizado las ventajas de la comunicación asíncrona entre aplicaciones.
- b) Se han definido los componentes del estándar JMS (Java Message Service).
- c) Se ha añadido una implementación de JMS a una aplicación web.
- d) Se ha configurado una implementación de JMS en una aplicación web Java.
- e) Se ha programado un proveedor y enviado mensajes a los receptores.
- f) Se ha programado un receptor para recibir los mensajes de un proveedor.
- g) Se han gestionado las excepciones y errores que pueden ocurrir en el proceso.

Conocimientos:

- Ventajas de la comunicación asíncrona: separación de intereses, desacoplamiento de aplicaciones, conexión de las partes...
- Principales componentes de JMS: proveedor, clientes, mensajes, objetos administrados.
- Dependencias necesarias: Apache ActiveMQ, Artemis...
- Configuración de la implementación de JMS. Integración en el Framework.
- Creación de un proveedor y envío de mensajes (interfaz OrderMessagingService en Spring).

jueves 16 de marzo de 2023

- Creación de un receptor y recepción de mensajes (interfaz OrderReceiver).
- Control de excepciones del estándar JMS (JMSEException).

6.– Desarrollar pruebas unitarias y de integración, usando las herramientas de un Framework de desarrollo.

Criterios de valoración:

- a) Se ha configurado una librería.
- b) Se han creado clases de prueba.
- c) Se han utilizado aserciones para realizar las pruebas unitarias.
- d) Se han realizado las pruebas unitarias, utilizando distintos métodos.
- e) Se han realizado pruebas de integración, utilizando objetos simulados.

Conocimientos:

- Configuración de librerías. Uso de un gestor de dependencias (Maven o Gradle o similar).
- Integración de JUnit en las clases del Framework.
- Métodos de prueba. Uso de anotaciones de JUnit.
- Pruebas de integración. Framework de pruebas.
- Herramientas de simulación de objetos: Mockito, integración de Mockito en Spring, ...

Asociados al ámbito 8: GESTIÓN DE DESARROLLO EN EQUIPO DE UNA APLICACIÓN.

DESTREZAS Y HABILIDADES A ADQUIRIR:

1.– Instalar entornos de desarrollo, configurando sus características.

Criterios de valoración:

- a) Se han instalado diferentes entornos de desarrollo.
- b) Se han configurado los entornos de desarrollo.
- c) Se han instalado las librerías adecuadas a cada proyecto.
- d) Se han instalado los complementos o plugins necesarios para cada proyecto.

Conocimientos:

- Entornos de desarrollo. Características, áreas de trabajo e integración. Instalación y configuración.
- Entornos de desarrollo apropiados para desarrollo JAVA: Eclipse, Netbeans...
- Complementos de un entorno de programación. Instalación y configuración.
- Librerías. Definición, usabilidad e instalación.

2.– Generar la documentación de una aplicación, utilizando los estándares actuales.

Criterios de valoración:

- a) Se ha creado la estructura de la documentación según el estándar elegido.
- b) Se ha documentado la arquitectura del sistema.
- c) Se ha documentado el modelo de datos aceptado.
- d) Se ha desarrollado la guía técnica.
- e) Se ha desarrollado la guía de uso.
- f) Se ha desarrollado la guía de instalación.
- g) Se han construido los prototipos de uso.

Conocimientos:

- Documentación de aplicaciones. Estándar ISO del ciclo de vida del software. Características y aplicabilidad.
- Estructura de la documentación oficial del lenguaje.
- Arquitectura del sistema. Diseño y documentación.
- Modelo de datos. Diseño, implementación y documentación.
- Guía técnica, guía de uso y guía de instalación. Estructura y desarrollo.
- Prototipos. Características, diseño y documentación.

3.– Gestionar versiones de una aplicación, utilizando las herramientas software adecuadas.

Criterios de valoración:

- a) Se han creado versiones alternativas de un proyecto en un repositorio.
- b) Se han desarrollado distintas metodologías de trabajo en el control de versiones.
- c) Se ha sincronizado el entorno de desarrollo con las plataformas de gestión de versiones.
- d) Se ha gestionado el desarrollo de una aplicación en grupo a través de plataformas de gestión de versiones.
- e) Se ha configurado una herramienta para desplegar una aplicación desde un repositorio.
- f) Se ha configurado un repositorio para emplearlo como herramienta de integración continua.

Conocimientos:

- Herramientas software para la generación y control de versiones. VCS, Git, GitFlow, CVS...
- Metodologías de trabajo al utilizar control de versiones: Desarrollo lineal, integración continua, desarrollo paralelo, ramas.

- Sincronización, confirmaciones y envíos.
- Desarrollo lineal, propagación de errores.
- Desarrollo paralelo. Ramas y unión de ramas.
- Concepto de repositorio. Características, tipos. Desarrollo de procesos de integración continua en repositorios.

4.– Garantizar la fiabilidad de una aplicación, utilizando plataformas integradas de pruebas.

Criterios de valoración:

- a) Se han desarrollado test unitarios con una herramienta informática a partir de entradas y salidas dadas.
- b) Se han utilizado plataformas integradas orientadas a la prueba.
- c) Se han realizado pruebas de integración, regresión y carga.
- d) Se han realizado pruebas de seguridad mediante las aplicaciones software.
- e) Se han realizado pruebas automáticamente mediante sistemas de integración continua.

Conocimientos:

- Pruebas unitarias. Características. Aplicación. Github actions.
- Plataformas integradas orientadas a la prueba. Metodologías ágiles. Scrum.
- Pruebas de integración, de regresión y de carga. Diferencias y aplicación.
- Herramientas software para realizar pruebas de seguridad. Estándar ASVS.
- Integración continua. Automatización de pruebas. Jenkins, Github...

5.– Desplegar e implantar aplicaciones, definiendo el diagrama de implementación adecuado.

Criterios de valoración:

- a) Se ha definido el diagrama de implementación.
- b) Se han definido los diagramas de despliegue.
- c) Se han instalado y configurado servidores Web y de bases de datos.
- d) Se han añadido y configurado los componentes y módulos necesarios para el procesamiento de código en el servidor.
- e) Se ha utilizado una herramienta de despliegue de aplicaciones desde un proyecto en un repositorio.
- f) Se han reconocido las características de la integración continua y despliegue continuo.

Conocimientos:

- Diagramas de implementación. UML.

- Diagramas de componentes y de despliegue.
- Servidores web. Plan de Implantación de aplicaciones.
- Servidores de Bases de datos. Creación y mantenimiento de Bases de datos.
- Bases de datos en servidor. Creación, mantenimiento y diseño de accesos.
- Repositorios de aplicación. Despliegue de aplicaciones a partir de un repositorio.
- Herramientas de despliegue de aplicaciones:
Integración continua, Distribución/Implementación continua. CI/CD.Jenkins, Azure DevOps, AWS...

d) TÍTULOS ASOCIADOS AL PROGRAMA.

- Técnico Superior en desarrollo de aplicaciones multiplataforma.
- Técnico Superior en desarrollo de aplicaciones web.

Asimismo, de manera excepcional y previa autorización de la Viceconsejería de Formación Profesional, también podrán participar en estos programas de especialización, profesionales con más de 3 años de experiencia que sean propuestos para ello por las empresas colaboradoras en la impartición del programa.

e) SECTOR ECONÓMICO Y DEMANDANTES.

Demanda de formación que proviene de empresas tecnológicas, pertenecientes al sector de la informática y comunicación.

f) REQUISITOS DEL PROFESORADO E INSTRUCTORES.

Apartado 1.– Especialidades del profesorado y atribución docente en los ámbitos de aprendizaje del programa de especialización profesional.

El profesorado del centro de formación deberá poseer los requisitos regulados para alguna de las especialidades que a continuación se indican:

Ámbitos de aprendizaje	Especialidades del profesorado
1. Programación básica en Java	Profesor o Profesora de Enseñanza Secundaria: <ul style="list-style-type: none"> • Informática
2. Programación gráfica en Java. Multiplataforma	
3. Programación Java en entorno cliente.	
4. Programación Java en entorno servidor	
5. Persistencia de información en Java	
6. Frameworks de desarrollo	
7. Desarrollos avanzados	
8. Gestión del desarrollo en equipo de una aplicación	

jueves 16 de marzo de 2023

Apartado 2.– Titulaciones requeridas para la impartición de los ámbitos de aprendizaje que conforman el programa para los centros de titularidad privada o titularidad pública de otras Administraciones distintas a la educativa.

El profesorado del centro de formación deberá poseer los requisitos de titulación, formación y experiencia laboral regulados para la impartición de los módulos profesionales de los ciclos formativos de referencia del programa cuya docencia se atribuye a alguna de las especialidades de profesorado que se indican para cada ámbito de aprendizaje en el apartado anterior.

Apartado 3.– Requisitos de experiencia y formación del personal instructor aportado por la empresa.

En relación con el personal instructor aportado por la empresa o empresas participantes en la formación, deberá tener una experiencia laboral en actividades relacionadas con el perfil del programa de al menos 3 años, o acreditar una formación relacionada con los resultados de aprendizaje del programa de al menos 5 años.