|  |  |
| --- | --- |
| **DESCRIPCIÓN:** | |
| En esta actividad los alumnos de forma individual deberán trabajar con repositorios de información cumpliendo estándares:   * Creación de repositorios cumpliendo estándares. * Diseñar repositorios de información   AL finalizar la prueba, el alumno entregará en formato digital un archivo con las explicaciones de los ejercicios realizados. | |
| **MEDIOS PARA SU REALIZACIÓN:** | |
| AULA DE GESTIÓN  Equipos audiovisuales  PCs instalados en red  Cañón con proyección e internet  Software específico de la especialidad  Pizarra para escribir con rotulador  Material de aula  Mesa y silla para el formador  Mesa y sillas para alumnos  Para el desarrollo de la actividad, el aula posee las dotaciones e infraestructuras necesarias para ejecutar la misma de forma correcta, puesto que previamente ha sido homologada por el Servicio Canario de Empleo | |
| **PAUTAS DE ACTUACIÓN DEL FORMADOR** | |
| Antes del inicio de la actividad, el docente explicará en qué consiste la tarea, el tiempo para su realización y repartir el material necesario para su realización.  Al finalizar la actividad los alumnos deberán entregar lo que han hecho en formato digital al docente, para poder archivarlo en Onedrive, donde se almacenarán todos los ejercicios prácticos realizados por los alumnos, siendo este documento la evidencia documental para el SCE | |
| **ESPECIFICACIONES PARA LA EVALUACIÓN DE LA PRÁCTICA** | |
| **Resultados a comprobar** | **Indicadores de Logro** |
| 1. En un supuesto práctico, debidamente caracterizado, realizar las funciones de creación de repositorios para cubrir las necesidades de gestión de información, de acuerdo a un diseño especificado en lo que al repositorio respecta. **(CE1.2)** | 1. Identificación de diseños específicos para cubrir las necesidades de gestión de información. |
| 1. Cumplimentación de la gestión de la información ajustada al diseño de repositorios. |
| 1. Medición de calidad del código verificando legibilidad, eficiencia y ausencia de errores. |
| 1. Creación de repositorios cumpliendo los estándares. |

| **SISTEMA DE VALORACIÓN** | | | |
| --- | --- | --- | --- |
| **Resultado a comprobar** | **Indicadores de logro** | **Escalas de medida** | **Valoración** |
| 1. En un supuesto práctico, debidamente caracterizado, realizar las funciones de creación de repositorios para cubrir las necesidades de gestión de información, de acuerdo a un diseño especificado en lo que al repositorio respecta.   **(CE1.2)** | 1. Identificación de diseños específicos para cubrir las necesidades de gestión de información. | * Identifica más del 75% según pautas dadas. * Identifica entre el 50% y el 75% según pautas dadas.   Identifica menos del 50%. | 2.5  1.25  0 |
| 1. Cumplimentación de la gestión de la información ajustada al diseño de repositorios. | * Cumplimenta más d l 75% según pautas dadas. * Cumplimenta entre el 50% y el 75% según pautas dadas.   Cumplimenta menos del 50%. | 2.5  1.25  0 |
| 1. Medición de calidad del código verificando legibilidad, eficiencia y ausencia de errores. | * Mide más del 75% según pautas dadas. * Mide entre el 50% y el 75% según pautas dadas.   Mide menos del 50%. | 2.5  1.25  0 |
| 1. Creación de repositorios cumpliendo los estándares. | * Crea más del 75% según pautas dadas. * Crea entre el 50% y el 75% según pautas dadas.   Crea menos del 50%. | 2.5  1.25  0 |
| **Nota:** | | |  |

**Instrucciones de la prueba.**

**Lee detenidamente la INFORMACION:**

* La presente prueba evaluable tendrá una duración de 3 horas y se valora en una escala de 0 al 10.

**Si tienes alguna duda o pregunta comunícala a el/la docente. ¡Mucha suerte!**

**SUPUESTO PRÁCTICO:**

Crear un proyecto llamado MotoShop como sigue:

* Crear una superclase llamada Moto.  La clase Moto tiene los siguientes campos y métodos.
  + String modelo
  + int speed;
  + double Price;
  + String color;
  + double getSalePrice();
  + void print(); //Muestra el modelo, la velocidad, el color
* Crear una subclase de Moto y llamarla Custom.  La clase Custom tiene los siguientes campos y métodos.
  + boolean cromada ;
  + double getSalePrice();  // Si cromada getSalePrice() + 20% de sobreprecio
  + void print(); //Muestra el modelo, la velocidad, el color y el precio de venta.
* Crear una subclase de Moto y llamarla Sport.  La clase Sport tiene los siguientes campos y métodos
  + int year;
  + int cilindrada;
  + double getSalePrice();  // al precio de venta calculado de la clase Moto, sumar si cilindrada > 2000 un 5% al precio, si es menor o igual de 2000 restar un 5% de descuento al precio.
  + void print(); //Muestra el modelo, la velocidad, el color y el precio de venta.
* Crear una subclase de Moto y llamarla Vespino. La clase Vespino tiene los siguientes campos y métodos.
  + int plazas;
  + double getSalePrice();  // Si plazas = 2, 5% descuento, si plaza =1, 10% descuento.
  + void print(); //Muestra el modelo, la velocidad, el color y el precio de venta.
* Crear la clase Ejecutor que contiene el método main().  Realizar lo siguiente dentro del método main().
  + Crear un objeto de la clase Custom e inicializar todos los campos con valores apropiados.
  + Crear dos objetos de la clase Sport e inicializar todos los campos con valores apropiados.
  + Crear un objeto de la clase Vespino e inicializar todos los campos con valores apropiados.
* Mostrar los datos de todos los objetos mediante el método print().