

**PRUEBA SYMFONY V1**

**Gómez Junco Isaac**

**11 Febrero de 2023**

# PRUEBA TEÓRICA

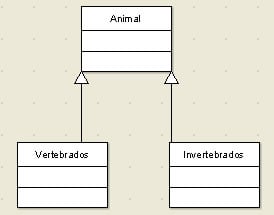
## Indicaciones Generales

* Descargar este archivo
* Resolverlo en sus ambientes locales
* Crear dos directorios en el repositorio git correspondiente:
  + Teoría
  + Algoritmos
* Subir el examen resuelto al directorio Teoría
* Para los algoritmos o Subir la evidencia en video
* Usar comentarios en las funciones, clases, parámetros con la finalidad de seguir más fácilmente la lógica aplicada.
* Todos los archivos deben ser subidos al repositorio git entregado
* Ubicar en la rama master / main la resolución definitiva
* Tiempo máximo de entrega: 1H

## Preguntas

### POO

1. Si en una subclase se declara un método que tiene el mismo tipo de argumentos que un método declarado por una de sus superclases esto se conoce como:
   1. Method overloading
   2. Operator overloading
   3. Operator overriding
   4. Method overriding
2. Son conceptos asociados a la POO
   1. Herencia
   2. Extensibilidad
   3. Polimorfismo
   4. Todos los anteriores
3. Los constructores se usan para
   1. To build a user interface.
   2. Initialize a newly created object.
   3. To create a sub-class.
   4. Free memory.
4. En POO la Delegación es
   1. Cuando un método quiere atribuir un comportamiento a una clase
   2. Cuando un objeto pasa una relación de una clase a un método
   3. Cuando se determina un error
   4. Cuando un objeto pasa un mensaje a otro objeto con el propósito de satisfacer una petición.
5. Cuál es el principio SOLID que mejor se adapta a: “Es mejor tener muchas clases pequeñas y especializadas que tener una sola clase gigante”
   1. Single Responsibility
   2. Open Closed
   3. Liskov substitution
   4. Interface Segregation
6. Que concepto de la POO se representa en la siguiente imagen



* 1. Realización
  2. Polimorfismo
  3. Herencia
  4. Asociación

### GIT

1. Para agregar comentarios a un commit se usa
   1. Git commit "I'm coding!"
   2. Git add "I'm coding!"
   3. Git commit -m "I'm coding!"
   4. Git message "I'm coding!"
2. Git pull es una combinación de
   1. branch and checkout
   2. add and commit
   3. fetch and merge
3. You want to upload the code to your remote git url, which git command you will run?
   1. git fetch
   2. git checkout
   3. git push
   4. git buid
4. Where are the files stored before they are committed to the local repository?
   1. Stashed Area
   2. Cache Area
   3. Staging Area
   4. Squashed Area
5. ¿En qué circunstancias debería usar un solo guión dentro de un comando de Git, en lugar de un guion doble?
   1. Cuando se usa una opción de varios caracteres
   2. Al emitir comandos de Git con múltiples opciones
   3. Cuando se usa una opción de un solo carácter
   4. Al hacer referencia a un alias de Git
6. ¿Cuál es el comando para añadir el repositorio remoto “https://abc.xyz/d/e.git” como “origin”?
   1. git add origin https://abc.xyz/d/e.git
   2. git remote https://abc.xyz/d/e.git
   3. git origin=https://abc.xyz/d/e.git
   4. git remote add origin https://abc.xyz/d/e.git

1. Git commit -- amend sirve para:
   1. Alertar en caso de que exista un conflicto en el area de trabajo
   2. Revisar conflicto antes de subir el trabajo al repositorio remoto
   3. Editar el mensaje del commit sin modificar su instantánea
   4. Agregar un nuevo mensaje al commit modificando su instantánea
2. ¿Cuál es la manera correcta de usar el comando push?
   1. None of them
   2. Git push <remote> <branch>
   3. Both correct
   4. Git push <branch> <remote>
3. Which command(s) you would need to run mandatorily to push remotely
   1. git add, git commit
   2. git commit, git pull
   3. git add, git commit, git pull
   4. git add, git log, git pull

### VARIOS

1. ¿Un endpoint para eliminar registros debe consumirse con el método PATCH?
   1. Verdadero
   2. Falso
2. El código html:

&lt;video width=“320″ height=“240″ controls=“controls”&gt; <br />&lt;source src=“test.mp4″ type=“video/mp4″ /&gt; <br />&lt;/video&gt;

* 1. Desplegar audio
  2. Desplegar video
  3. Desplegar texto
  4. Ninguna de las anteriores

1. Un microservicio es:
   1. Un servicio con máximo 5 líneas de codigo
   2. Un componente de sottware enfocado en se ligero e implementar funcionalidad independiente
   3. Se usa con contenedores
   4. Todas las anteriores
2. Que protocolos son los más comunes para exponer un servicio
   1. http y https
   2. soap y rest
   3. rpc y jms
   4. esb y microservicios
3. ¿Cuantos métodos de petición HTTP existen actualmente?
   1. 4
   2. 6
   3. 9
   4. 8
4. Cuando nos referimos al formato de intercambio de mensajería nos referimos a
   1. rest y soap
   2. xml y json
   3. http y https
   4. modelo canónico
5. ¿Qué hace el siguiente comando en Linux, mkdir –p /home/prueba/directorio?

Crea un nuevo directorio llamado "directorio" dentro de otro directorio llamado "/home/prueba". La opción "-p" crea directorios padres en el caso de que no exista, para el ejemplo sería "/home/prueba".

1. ¿Cuál de los siguientes directorios contiene archivos de configuración en Linux?
   1. /dev/
   2. /bin/
   3. /root/
   4. /etc/
2. ¿Cuál de las siguientes acciones se realiza con el comando <find / -name ‘\*’>
   1. Print a file with name \* in /
   2. List all files in / directory
   3. List all files and directories in / directory
   4. List all files and directories recursively starting from /
3. Explique la diferencia entre **cat file1 > file 2** y **cat file1 >> file2**

El comando **cat file1 > file 2** hace un redireccionamiento a la salida del archivo file1 y lo sobre escribe en el archivo file 2, si file2 ya tiene contenido, dicho contenido se reemplaza por el contenido de file1, a diferencia de **cat file1 >> file2** que lo que hace es agregar el contenido del archivo file1 al archivo file2 solo que conservando el contenido del archivo file2, los datos de file1 se añaden al final de los datos de file2, en este casi si file2 no se encuentra creado, se procederá a crear con la información existente en file1.

1. ¿Principales características de Docker?
   1. Fast, Scalable, Reliable
   2. Lightweight, Sociable, Secure
   3. Lightweight, Standard, Secure
   4. Lightweight, Scalable, Fast
2. Which command is used to create a bridge network bridge1 in docker?
   1. docker create network -d bridge bridge1
   2. docker network create -d bridge bridge1
   3. docker network create bridge1 bridge
   4. docker network create -d bridge1 bridge
3. \_\_\_\_\_\_\_\_is a tool for defining and running multi-container Docker applications.
   1. Docker Swarn
   2. Docker Hub
   3. Docker Cloud
   4. Docker Compose
4. docker exec -it container\_id bash.
   1. Activate default VM machine
   2. Access a running container
   3. Build an image
   4. Commit changes done in a Docker image
5. El comando docker build -t my\_user/repo\_name:1.0 es usado para
   1. Activate default VM machine
   2. Access a running container
   3. Build an image
   4. Commit changes done in a Docker image

### ALGORITMOS

1. Asumiendo que se tiene la siguiente estructura

|  |  |  |
| --- | --- | --- |
| Registro | Nombre | Padre |
| 1 | Carlos | 0 |
| 2 | Juan | 0 |
| 3 | Maria | 1 |
| 4 | Carlos 1 | 3 |
| 5 | Sofia | 4 |
| 6 | Hernán | 1 |
| 7 | Wilson | 2 |
| 8 | Diana | 2 |
| 9 | Mateo | 7 |
| 10 | Efrain | 6 |
| ……… | …. | …. |
| ……. | ……. | …. |

Escriba un algoritmo que permita representar en orden descendente el árbol genealógico. El resultado debe representarse así:

Carlos Juan

Maria Wilson

Carlos 1 Mateo

Sofia Diana

Hernan

Efrain

1. Escribir un programa para invertir los bits de un número entero (32 bits sin signo)

<?php

$n = 123456789;

$invertido = invertir($n);

function invertir($num) {

$result = 0;

for ($i = 0; $i < 32; $i++) {

$result <<= 1;

$result |= $num & 1;

$num >>= 1;

}

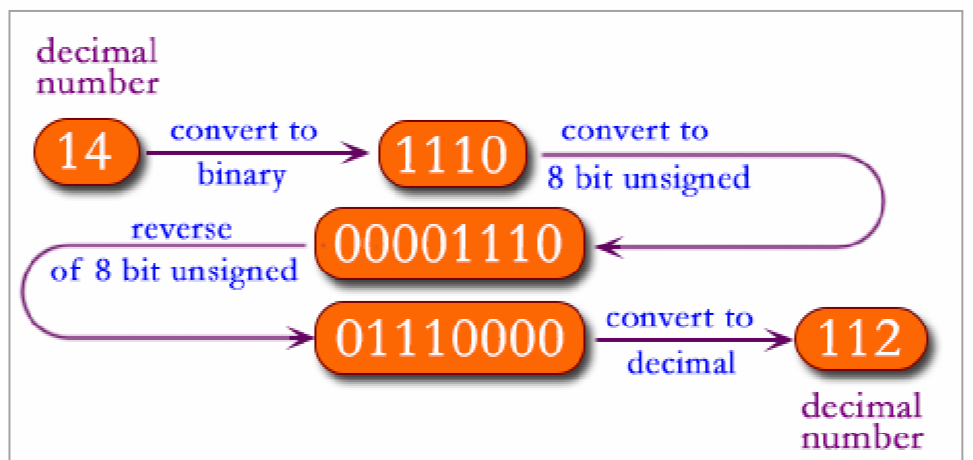
return $result;

}

echo "Número anterior: $n\n";

echo "Número invertido: $invertido\n";

?>



1. Escriba un algoritmo para determinar si un número ingresado por teclado es Fibonacci o no

Inicializamos dos variables a y b, las cuales corresponden a los dos primeros números de la secuencia de Fibonacci.

Pedir al usuario que ingrese el número n que se desea comprobar.

Mientras b sea menor o igual a n, hacer lo siguiente:

Si b es igual a n, entonces n es un número de Fibonacci, por lo que se puede imprimir un mensaje indicando que el número es de Fibonacci y luego salir del programa.

Si b es mayor que n, entonces n no es un número de Fibonacci, por lo que se puede imprimir un mensaje indicando que el número no es de Fibonacci y luego salir del programa.

Si b es menor que n, actualizar a y b como b y a + b respectivamente y volver a la segunda etapa del ciclo.

**En forma de pseudocódigo quedaría de la siguiente forma:**

a ← 0

b ← 1

es\_fibonacci ← falso

Escribir "Ingrese un número entero:"

Leer n

Mientras b ≤ n

Hacer

Si b = n Entonces

es\_fibonacci ← verdadero

Salir del ciclo

Fin Si

c ← a + b

a ← b

b ← c

Fin Mientras

Si es fibonacci Entonces

Escribir "El número pertenece a la secuencia Fibonacci."

Sino

Escribir "El número no es de Fibonacci."

Fin Si

# PRUEBA PRÁCTICA

## Indicaciones Generales

* En el repositorio GIT asignado se debe crear una carpeta (practica) bajo la cual se deben subir los archivos relacionados con este trabajo
* Tiempo máximo de entrega: 4H

## Enunciado

El departamento de consultorio médico requiere de un sistema para llevar el registro de las citas médicas diarias. Los pacientes son los únicos que pueden agendar citas médicas, la misma que pasará hacia los médicos para que ingresen el tipo de cita y la cantidad de minutos que duró cuando esta sea atendida. Al final, todas las citas atendidas llegan a caja para que se genere la factura por cada una de ellas, su valor dependerá del tipo de cita con la que se registró.

El administrador podrá tener acceso a todas las facturas generadas en el sistema.

Todos los usuarios se deben registrar de forma autónoma con el rol que le corresponda.

*Se necesitará crear varias entity, como "Paciente", "Medico", "Cita", "Factura" y "Usuario", y establecer relaciones entre ellas para poder realizar las operaciones necesarias. Además, se requerirá implementar un sistema de autenticación para la gestión de usuarios y roles.*

## Entregables

* Aplicación con comentarios en código
* Scripts para la creación de las estructuras en base de datos
* Documentación (instructivo de despliegue, manual de usuario, dercas)
* Uso de los archivos README en el repositorio para instrucciones generales

## Puntuación

* Instructivos, manuales, documentación,DER: 10 puntos
* Aplicación: 70 puntos
* Uso de recursos tales como bootstrap, frameworks, etc, 10 puntos
* Documentación del código: 5 puntos
* Video que evidencia el funcionamiento de la aplicación: 5 puntos