Aplicación a la programación orientada a objetos.

Nombre: Isaac Meza

Licenciado: Ing.Byron Herrera.

Realizar 30 preguntas de selección múltiple de lo que hemos visto.

1.-¿Qué significa Angular?

a.- Angular es un marco de desarrollo para dirección aplicaciones de varias páginas.

b.- Angular es un marco de desarrollo para construir aplicaciones web de una sola página

c.-Angular es un arco teórico para disminuir aplicaciones web de una sola página.

d.- Angular es un desarrollo productivo para aplicaciones web de una sola app.

e.-Angular no es un marco de desarrollo para construir aplicaciones web de una sola página.

2.- ¿Qué sitio web mantiene Angular?

a.-Google

b.-Wikipedia

c.-Gmail

d.-Youtube

e.-Twich

3.-¿Cuál es la diferencia entre AngularJS y Angular?

a.-AngularJS es la versión más antigua de Angular1x, mientras que Angular sin el sufijo "JS" se refiere a las versiones más recientes 2.x y posteriores.

b.-Angular está reescrito desde cero y presenta mejor rendimiento y modularidad en comparación con AngularJS.

c.-Angular está reescrito desde cero y presenta mejor rendimiento y modularidad en comparación con AngularJS.

d.-Angular está reescrito desde cero y presenta mejor rendimiento y modularidad en comparación con Angularjoredc.

e.-AngularJS está reescrito desde cero y presenta mejor rendimiento y modularidad en comparación con Angular.

4.-¿Cuál es el ciclo de vida de un componente en Angular.?

a.-ngOnInit, ngOnChanges, ngDoCheck, ngAfterContentInit, ngAfterViewInit

b.-ngOnInit, ngDoCheck, ngAfterContentInit, ngAfterViewInit

c.-ngOnInit, ngDoCheck, ngAfterContentInit

d.-ngOnInit, ngDoCheck, ngAfterViewInit

e.- ngDoCheck, ngAfterContentInit, ngAfterViewInit

5.- ¿Cómo se realiza la comunicación entre componentes en Angular?

a.- @Input y @Output

b.- @Inpus y @Autput

c.-@Inpot y @utput

d.-@Mnput y @Iutput

e.-@Inpf y @Outpf

6.- Explique el propósito de @ViewChild en Angular.

a.-No se utiliza para acceder a un componente hijo desde el componente padre

b.-Se utiliza para acceder a un componente padre desde el componente hijo

c.-Se utiliza para acceder a un componente hijo desde el componente padre

d.-Se utiliza para predecir a los componentes hijos desde el componente padre

e.-No se utiliza para acceder a un componente padre desde el componente hijo

7.- ¿Qué es una directiva estructural en Angular?

a.-compartir datos y realizar operaciones

b.-Los servicios son clases singleton que no proporcionan funcionalidades compartidas entre componentes.

c.-La inyección de dependencias es un patrón en el que un componente recibe sus dependencias en lugar de crearlas internamente.

d.-Los servicios son clases singleton que proporcionan funcionalidades compartidas entre componentes.

e.-Los servicios son clases singleton que proporcionan funcionalidades compartidas entre componentes regulares e irremediables.

8.- ¿Cómo se realiza la prueba de un servicio en Angular?

a.-Se puede probar un servicio creando una instancia de la clase del servicio y luego utilizando métodos de TestBed para proporcionar dependencias y realizar afirmaciones.

b.-Se puede probar un servicio creando una instancia del objeto del servicio y luego utilizando métodos de TestBed para proporcionar dependencias y realizar afirmaciones erróneas.

c.-Se puede probar un servicio creando una instancia de la clase del servicio y luego utilizando métodos de TestBed para proporcionar dependencias y realizar afirmaciones nulas.

d.-Se puede probar un servicio creando una instancia de la clase del servicio y luego utilizando métodos de TestBed para proporcionar dependencias y realizar alternativas.

e.-Se puede probar un servicio creando una instancia de la clase del servicio y luego utilizando métodos de TestBed para proporcionar dependencias y realizar afirmaciones concretas.

9.-¿Qué es un interceptor HTTP en Angular ?

a.-Un interceptor HTTP permite la rehabilitación de las solicitudes y respuestas HTTP.

b.-Un interceptor HTTP permite la manipulación de las solicitudes y respuestas HTTP.

c.-Un interceptor HTTP permite la prohibición de las solicitudes y respuestas HTTP.

d.-Un interceptor HTTP permite la manipulación de las anécdotas y respuestas HTTP.

e.-Un interceptor HTTP no permite la manipulación de las solicitudes y respuestas HTTP.

10.-¿Qué es C + +?

a.-C + + es un lenguaje de programación de propósito general que extiende el lenguaje C.

b.-C + + es un lenguaje de programación de propósito limitado que extiende el lenguaje D.

c.-C + + es un lenguaje de programación de propósito general que extiende el lenguaje A.

d.-C + + es un lenguaje de programación de propósito general que no extiende el lenguaje C.

e.-C + + es un objeto de programación de propósito reducido que extiende el lenguaje C.

11.-¿Qué significa POO?

a.-Programación objetiva de origen.

b.-Prolongación orientada a objetos.

c.-Programación orientada a objetos.

d.-Prolongación objetiva de origen .

e.-Preparación ornamental objetiva.

12.- ¿Cuáles son las siglas de programación orientada a objetos?

a.-POAO

b.-POO

c.-OPO

d.-POS

e.-DSO

13.-¿Quién fue el desarrollador del C + +?

a.-Bjarne Stroustrup

b.-Jaime Stain

c.-Tom Holan

d.-Robert Lewandowski

e.-Isaac Newton

14.- ¿Cuáles son los operadores de la gestión dinámica de la memoria C + +?

a.-Do y if

b.-Swif y Do

c.-New y Do

d.-Delete y If

e.-New y Delete

15.- ¿Cuál es la clave que se declara en la estructura C + +?

a.-Remember

b.-New

c.-Delete

d.-Struct

e.-Stop

16.-¿Qué funciones abren archivos C + +?

a.-new y swif

b.-ofstream y ifstream

c.-downstream y deletestrem

d.-gostem y gorosei

e.-saturn y it

17.-¿Para qué sirve la función “Ofstream”?

a.-para escribir en archivos

b.-para reescribir en archivos

c.-para borrar archivos

d.-para tachar archivos

e.-para cerrar archivos

18.-¿Para qué sirve la función “Ifstream”?

a.--para leer y borrar archivos

b.-para leer desde archivos

c.-para borrar archivos

d.-para tachar archivos

e.-para cerrar archivos

19.- ¿Qué función cierra archivos C + +?

a.-close()

b.-closs()

c.-driving()

d.-tach()

e.-open()

20.-¿Qué son las excepciones en C + +?

a.-Las excepciones son eventos imprevistos que no ocurren durante la ejecución.

b.-Las excepciones son eventos imprevistos que ocurren durante la percepción.

c.-Las excepciones son eventos imprevistos que ocurren durante la codificación.

d.-Las excepciones son eventos imprevistos que ocurren durante la ejecución.

e.-Las excepciones son eventos previstos que ocurren durante la ejecución.

21.- ¿Que bloques manejas las excepciones en C + +?

a.-do y while

b.-try, catch y throw.

c.-rading y try

d.- carrot, snuffy y throw.

e.-shilfi, catch

22.- ¿Qué es la sobrecarga de operadores en C + +?

a.-La sobrecarga de operadores no permite definir comportamientos específicos para operadores en clases definidas por el usuario.

b.-La sobrecarga de operadores permite definir comportamientos generales para operadores en clases competitivas por el usuario.

c.-La sobrecarga de operadores permite definir comportamientos específicos para operadores en clases definidas por el usuario.

d.-La sobrecarga de operadores permite definir comportamientos específicos para operadores en clases definidas por el monitor.

e.-La sobrecarga de operadores permite definir comportamientos específicos para operadores en objetos definidos por los receptores.

23.- ¿Para qué sirve “endl” en C + +?

a.-se utiliza para imprimir una nueva línea y vaciar

b.-para escribir en archivos

c.-para borrar archivos

d.-para tachar archivos

e.-para cerrar archivos

24.- ¿Qué es Node.js?

a.-Node.js no es un entorno de ejecución de JavaScript del lado del servidor

b.-Node.js es un entorno de ejecución de JavaScript del lado del servidor

c.-Node.js es un entorno de ejecución de Java del lado del servidor

d.-Node.js es un entorno de reducción de JavaScript del lado del servidor

e.-Node.js es un entorno de ejecución de JavaScript del lado del diseño

25.-¿Qué es npm?

a.-Node Package Manager

b.-Nope Pack Messing

c.-Not Pass Mach

d.-Node Pass Manager

e.-Nope Package Mach

26.- ¿Para qué sirve un “module.exports”?

a.-Crea un módulo en Node.js

b.-Recrea un módulo en Node.js

c.-Deshace un módulo en Node.js

d.-Actualiza un módulo en Node.js

e.-Crece un módulo en Node.js

27.- ¿Que se utiliza para manejar solicitudes HTTP GET?

a.-app.go()

b.-app.ret()

c.-app.get()

d.-app.sold()

e.-app.vild()

28.- ¿Que se utiliza para manejar solicitudes HTTP POST?

a.-app.go()

b.-app.ret()

c.-app.post()

d.-app.past()

e.-app.posd()

29.-¿Cómo funciona el enrutamiento de parámetros en Express.js.?

a.-con un “req.params”

b.-con un “red.params”

c.-con un “ram.params”

d.-con un “req.paramas”

e.-con un “req.pirams”

30.-¿Qué es Passport.js?

a.-Passport.js es un autenticador para Nude.js.

b.-Passport.js es un middleware de autenticación para Node.jd.

c.-Passport.js es un mind de autenticación para Node.js.

d.-Passport.js no es un middleware de autenticación para Node.js.

e.-Passport.js es un middleware de autenticación para Node.js.