HERRAMIENTAS AVANZADAS PARA EL DESARROLLO DE APLICACIONES.

1. Una señal:
    a. Sólo puede tener conectado un manejador.
    b. Sólo puede estar conectada con métodos de una clase
    c. Puede tener conectados varios manejadores
    d. Ninguna de las anteriores.

2. La signatura de un manejador conectado a una señal:
    a. Está delimitada por la signatura de la señal
    b. No está delimitada por la signatura de la señal
    c. Depende de si la clase a la cual pertenece está dentro de un espacio de nombres.
    d. Ninguna de las anteriores

3. A una señal con visibilidad *pública*:
    a. Solo le podemos conectar manejadores públicos.
    b. Sólo le podemos conectar funciones independientes de cualquier clase
    c. Le podemos conectar cualquier manejador independiente de su visibilidad.
    d. Ninguna de las anteriores.

4. En Vala los espacios de nombres sólo se pueden crear así:
    a. ¿Espacio de nombres…? ¿Qué es eso?
    b. using namespace name;
    c. namespace name {…}
    d. Ninguna de las anteriores

5. En Vala la clase *ArrayList<T>* forma parte…
    a. Del lenguaje, es un tipo de dato básico
    b. De nada. Esa clase no existe
    c. De una biblioteca externa
    d. Ninguna de las anteriores.

6. En Vala la cláusula *requires* representa:
    a. Una excepción
    b. Una postcondición
    c. Una precondición
    d. Ninguna de las anteriores

7. Git es un sistema de control de versiones:
    a. Centralizado
    b. Distribuido
    c. Centralizado en unos casos, distribuidos en otros
    d. Ninguna de las anteriores.

8. La operación *commit* de Git:
    a. Traslada los datos del repositorio local a la copia maestra del mismo.
    b. Traslada los cambios hechos en la copia de trabajo al repositorio local.
    c. Traslada los cambios hechos en la copia de trabajo a un repositorio remoto.
    d. Clona repositorios.

9. La operación *branch* de Git:
    a. Es la única que podemos usar para crear ramas.
    b. Es la única que podemos usar para cambiar de ramas.
    c. Es la única que podemos usar para renombrar una rama.
    d. Ninguna de las anteriores.

10. En Vala para que funcione el mecanismo de *señales/manejadores*:
    a. No es necesario hacer nada, el lenguaje lo proporciona.
    b. Debemos derivar de la clase GLib.Object.
    c. Se debe compilar el código con una opción especial.
    d. Deberemos derivar de la clase Gtk.Object.

11. En Vala una función *lambda* puede hacer de manejador.
    a. Nunca
    b. Siempre
    c. Cuando coincida su lista de parámetros con los de la señal
    d. Cuando no tenga parámetros.

12. En Vala una señal…
    a. Nunca puede tener implementación
    b. Siempre puede tener implementación
    c. Sólo podrá tener implementación cuando se declare virtual.
    d. Ninguna de las anteriores

13. La biblioteca *Gtk+*…
    a. Implementa su tecnología de señales/manejadores
    b. Aprovecha la tecnología de señales/manejadores de GLib/GObject
    c. No usa el concepto de señales/manejadores
    d. Ninguna de las anteriores.

14. Para poder usar la biblioteca Gtk+ desde Vala
    a. Debemos llamar al compilador con la opción "- -ptk gtk+-2.0"
    b. No se necesita ninguna opción especial de compilación
    c. Debemos llamar al compilador con la opción "- -ptk gmodule -2.0"
    d. Es necesario construir previamente el interfaz gráfico de la aplicación con glade.
15. Los interfaces de usuario generados por *glade*…
    a. Se guarda en archivos binarios
    b. Se guardan como código fuente Vala
    c. Constituyen la única manera de dotar de interfaz gráfico a una aplicación que use Gtk+
    d. Son archivos de texto en formato XML

16. Para modificar el mensaje de un *commit*:
    a. Un mensaje de un commit no se puede modificar
    b. Utilizamos la opción - - amend para modificar el último commit
    c. Debemos hacer un checkout del archivo y luego usamos - - amend
    d. Ninguna de las anteriores.

17. En la programación dirigida por eventos, la cola de eventos
    a. Es necesaria
    b. No es necesaria
    c. Depende del lenguaje de programación usado.
    d. Ninguna de las anteriores.

18. La autoconexión de señales en el marco Glade +Gtk se hace
    a. Sobre un objeto de la clase Window
    b. Es un método estático de la clase Window
    c. Sobre un objeto de la clase Builder
    d. Es un método estático de la clase Builder.

19. Para guardar una copia temporal de tu directorio de trabajo
    a. Utilizamos la operación Git bisect
    b. Utilizamos la operación Git clone
    c. Utilizamos la operación Git stash
    d. No se puede realizar una copia , el directorio de trabajo es único

20. Git guarda toda la meta-información de un proyecto en
    a. Una base de datos relacional
    b. En un directorio único para cada usuario
    c. En un directorio único por proyecto
    d. En un archivo de configuración

21. En Vala el operador *as*
    a. Es un operador matemático
    b. Es un operador de conversión de tipos en tiempo de compilación
    c. Es un operador de conversión de tipos en tiempo de ejecución
    d. Ninguna de las anteriores.

22. En el lenguaje Vala colocar el símbolo *"(?)"* tras el nombre de un tipo
    a. Es un error
    b. No significa nada
    c. Indica que la variable que se declare de ese tipo puede contener el valor (null)
    d. Indica que la variable que se declare de ese tipo sólo podrá contener el valor (null)

23. Sqlite
    a. Requiere de un proceso servidor
    b. Requiere de un servidor y de un cliente
    c. En la versión 3 (sqlite3) sí requiere de un proceso servidor
    d. Ninguna de las anteriores.

24. En el patrón de arquitectura MVC aplicado en Vala usando Gtk
    a. **El modelo es representado por una clase, y el controlador sería implementado por la librería Gtk+**
    b. El modelo es representado por una clase, y el controlador por una clase interfaz
    c. La vista se implementa siempre con Glade, en formato XML
    d. B y C son ciertas.

25. Trabajando con Git si dos programadores modifican el mismo archivo
    a. **No tiene porque producirse un conflicto**
    b. Siempre se producirá un conflicto
    c. Git no permite modificar simultáneamente un mismo archivo
    d. Ninguna de las anteriores

26. En Git la clave *Sha-1* está asociada a
    a. Cada copia del repositorio
    b. Cada usuario que puede hacer commits en el proyecto
    c. Un conjunto de commits
    d. **Cada commit.**

27. En un sistema de control de versiones centralizado la operación *push*
    a. No existe
    b. **Hace el mismo papel que en un distribuido**
    c. Es equivalente a pull
    d. Es equivalente a rebase

28. Si en Git quisiéramos deshacer un commit sin perder la parte de la historia del proyecto donde aparece este commit
    a. No podemos
    b. **Usaremos la orden "git revert"**
    c. Usaremos la orden "git reset"
    d. Usaremos la orden "git rewind"

29. En git la operación *pull* equivale a
    a. **Fetch + merge**
    b. Merge
    c. Push
    d. Rebase

30. La operación *checkout* de git
    a. Sirve para cambiarse de una rama a otra con la opción -b
    b. **La podemos usar para cambiar de rama**
    c. Sirve únicamente para actualizar la copia del trabajo con la versión del repositorio local
    d. Ninguna de las anteriores

31. En el patrón de arquitectura MVC
    a. **Un modelo puede tener varias vistas sobre él**
    b. Una vista puede referirse a varios modelos si estos lo permiten
    c. Una vista puede referirse a más de un modelo siempre
    d. Ninguna de las anteriores

32. Cuando compilamos en Vala una aplicación que hace uso de Gtk y Glade
    a. Debemos añadir la opción del compilador: - - with -gtk
    b. Debemos añadir la opción del compilador: - - with -glade
    c. **Debemos añadir la opción del compilador: - - pkg gtk -2.0**
    d. Debemos añadir la opción del compilador: - - pkg gmodule-2.0

33. Para consultar los metadatos de una bbdd en Sqlite
    a. Utilizamos la operación .tables
    b. Utilizamos la operación .datbases
    c. **Consultamos la tabla de sqlite_master**
    d. Ninguna de las anteriores

34. En Gtk el modo de leer un elemento de interfaz de usuario creado en glade es
    a. Con el método get.ui.element
    b. Con el método get.widget o get.button etc…
    c. Con el método read_object
    d. **Con el método get_object**

35. Los SVC , según la forma de almacenar los contenidos, se clasifican en
    a. Centralizados y colaborativos
    b. Distribuidos y exclusivos
    c. Centralizados y exclusivos
    d. **Centralizados y Distribuidos**

36. El método *connect_after*
    a. Sólo admite funciones lambda manejadoras de la señal
    b. Siempre tiene que ser declarado virtual
    c. A y B son verdaderas
    d. **A y B son falsas**

37. En Vala la conexión de las señales con los manejadores
    a. Debe realizarse siempre en el main
    b. Debe realizarse siempre en el constructor
    c. No es obligatoria su conexión
    d. **Ninguna de las anteriores**

38. Las librerías dinámicas
    a. No hay que compilarlas
    b. Deben llevar el prefijo lib y la extensión .a
    c. Hacen crecer el ejecutable final
    d. **Ninguna de las anteriores**

39. En el interfaz de comandos Sqlite para ejecutar un fichero con comandos sql
    a. Utilizamos la operación .file
    b. Utilizamos la operación .exec
    c. Utilizamos la operación .read
    d. Ninguna de las anteriores

40. La clase *Window*
    a. Pertenece al espacio de nombre GLib
    b. Pertenece al espacio de nombre Gtk
    c. Pertenece al espacio de nombre Glade
    d. Pertenece al espacio de nombre Widget

41. Para que una aplicación escrita en Vala que hace uso de Gtk pueda funcionar una vez compilada:
    a. Hemos tenido que añadir una línea así: "using Gtk;"
    b. Hemos tenido que iniciar Gtk: "Gtk.init(ref args)"
    c. Basta con añadir la opción del compilador: "- - pkg gtk -3.0"
    d. No hay que hacer nada especial

42. En Vala una señal
    a. Nunca puede tener implementación
    b. Siempre puede tener implementación
    c. Sólo podrá tener implementación cuando se declare virtual
    d. Ninguna de las anteriores

43. ¿Cuál de estas afirmaciones es falsa?
    a. A una señal podemos conectarle una función lambda
    b. A una señal podemos conectar una función anónima
    c. A una señal no le podemos conectar una función lambda
    d. Una señal puede ser desconectada de su callback