

Cristian Lage Fernández

Exercicios

1. Define, según tu punto de vista, las atribuciones de todos los integrantes de la empresa MyfpSoft.

MyfpSoft es una empresa dedicada al desarrollo de software. Cuenta con dos personas de amplio recorrido en la informática: Andrino, que es analista programador, y Emma, que es arquitecta de software. El equipo se completa con dos nuevas incorporaciones: Manuela y Dimas. Ambos han terminado el ciclo de DAW y entran en esta nueva empresa para trabajar como programadores. ¿Sabrán ellos cuáles son sus atribuciones al igual que las de Adriano y Emma?.

Las atribuciones de todos los integrantes son las siguientes

1. Andrino (Analista Programador):

- Desarrollar y mantener el código de software.
- Participar en el diseño de soluciones técnicas.
- Analizar los requisitos del cliente y traducirlos en especificaciones técnicas.
- Colaborar con Emma en la definición de la arquitectura del software.
- Depurar y solucionar problemas técnicos en el código.
- Colaborar con Manuela y Dimas en el desarrollo de componentes.

2. Emma (Arquitecta de Software):

- Diseñar la arquitectura de software, incluyendo la estructura de la aplicación y las tecnologías a utilizar.
- Colaborar con Andrino en la implementación técnica de la arquitectura.
- Supervisar el progreso del proyecto y asegurar que se sigan las mejores prácticas de desarrollo.
- Tomar decisiones de diseño importantes y resolver desafíos técnicos complejos.
- Colaborar con Manuela y Dimas para asegurarse de que comprendan la arquitectura general.

3. Manuela y Dimas (Programadores):

- Desarrollar el código de software según las especificaciones proporcionadas por Andrino y Emma.
- Colaborar estrechamente con Andrino en la implementación técnica de componentes específicos.
- Realizar pruebas de unidad y correcciones de errores en el código.
- Participar en reuniones de equipo para discutir el progreso del proyecto y los desafíos técnicos.
- Aprender y mejorar sus habilidades de programación bajo la guía de Andrino y Emma.

En cuanto a si Manuela y Dimas sabrán cuales son sus atribuciones, es responsabilidad de la empresa proporcionar una orientación clara y comunicar los roles y responsabilidades de manera efectiva. Andrino y Emma, con su experiencia, pueden desempeñar un papel importante en la guía de Manuela y Dimas para que así pueden comprender mejor sus roles y puedan contribuir al equipo de manera efectiva. Además, las reuniones regulares de equipo y la comunicación abierta pueden ayudarles a aclarar cualquier pregunta o duda que tengan respecto a sus atribuciones en la empresa.

2. Genera un informe con los pros y contras de utilizar software a medida o estándar para el problema siguiente. Piensa cómo debería comunicarse la tienda física con la online. Seguramente, tendrás que investigar algo sobre el tema.

MyfpSoft tiene un cliente que le ha pedido que le construyan una tienda online para venta de sus productos farmacéuticos.

La farmacia cliente ya tiene un sistema de gestión. Un sistema en Windows con base de datos en SQL Server realizado y mantenido por la empresa Trompisoft.

Dimas se pregunta si es mejor utilizar software a medida o estándar. No sabe muy bien cuáles son las ventajas e inconvenientes de uno o otro. Tampoco sabe cómo va a comunicarse la tienda online con la tienda física.

Software a Medida:

Pros:

1. **Adaptación Específica:** Se puede diseñar el software para satisfacer las necesidades específicas de la farmacia cliente, incluyendo características únicas de la industria farmacéutica.
2. **Escalabilidad:** Puede diseñarse para adaptarse al crecimiento futuro de la empresa y permitir la incorporación de nuevas funcionalidades de acuerdo con las necesidades cambiantes.
3. **Integración más Profunda:** Permite una integración más estrecha con el sistema de gestión existente de la farmacia, lo que facilita la sincronización de datos en tiempo real.

Contras:

1. **Costos y Tiempo:** El desarrollo a medida suele ser más costoso y requiere más tiempo en comparación con la implementación de software estándar.
2. **Mantenimiento Continuo:** Requiere un compromiso constante para el mantenimiento, actualizaciones y corrección de errores.
3. **Dependencia de Personal Especializado:** Se necesita personal con habilidades específicas para desarrollar y mantener el software a medida.

Software Estándar:

Pros:

1. **Rapidez de Implementación:** Los software estándar suelen estar listos para su implementación inmediata, lo que acelera el proceso.
2. **Costo Inicial Menor:** Por lo general, tiene un costo inicial más bajo en comparación con el desarrollo a medida.
3. **Soporte de la Comunidad:** Puede contar con una comunidad de usuarios y soporte técnico disponibles para resolver problemas comunes.

Contras:

1. **Limitaciones de Personalización:** Puede no satisfacer todas las necesidades específicas de la farmacia cliente y requerir adaptaciones.
2. **Integración más Limitada:** Puede ser más complicado integrar un software estándar con el sistema de gestión existente de la farmacia.
3. **Menos Flexibilidad a Largo Plazo:** Puede no ser tan adaptable a los cambios futuros en el negocio o la industria.

Comunicación entre Tienda Online y Sistema de Gestión:

La comunicación entre la tienda online y el sistema de gestión de Trompisoft es fundamental. Esto se puede lograr de varias maneras:

1. **API Personalizada:** Si se elige el software a medida, MyfpSoft puede desarrollar una API personalizada que permita la comunicación en tiempo real entre la tienda online y el sistema de gestión.
2. **Middleware:** Se puede utilizar un middleware o un software de integración para facilitar la comunicación entre sistemas, independientemente de si se elige software a medida o estándar.
3. **Importación/Exportación de Datos:** Se pueden programar tareas automatizadas para importar/exportar datos periódicamente entre la tienda online y el sistema de gestión.

3. Clasifica los lenguajes siguientes según lo que se expone a continuación.

Dimas y Manuela están tomando café. Entre ellos, se han picado para saber si pueden clasificar los siguientes lenguajes. Los dos dicen saber mucho sobre lenguajes de

programación. A los dos les encantaba la asignatura Entornos de Desarrollo y, por lo tanto, creen poder clasificar los lenguajes que, a continuación, se presentan:

1. AngularJS

- Es un Framework
- De alto nivel
- Interpretado
- No utiliza una máquina virtual
- Se utiliza principalmente para el desarrollo web (frontend)
- Se basa en JavaScript

2. PHP

- Es un lenguaje de programación
- De alto nivel
- Interpretado
- No utiliza una máquina virtual
- Se utiliza principalmente para el desarrollo web (backend)
- No se basa en ninguna tecnología específica

3. Java

- Es un lenguaje de programación
- De alto nivel
- Compilado e interpretado (bytecode)
- Utiliza una máquina virtual (JVM)
- Se utiliza en una amplia variedad de aplicaciones, incluyendo aplicaciones de propósito general, aplicaciones web, aplicaciones móviles (Android)
- Se basa en la plataforma Java

4. JavaScript

- Es un lenguaje de programación
- De alto nivel
- Interpretado
- No utiliza una máquina virtual
- Se utiliza principalmente para el desarrollo web (frontend)
- Se basa en ECMAScript

5. Ruby

- Es un lenguaje de programación
- De alto nivel
- Interpretado
- No utiliza una máquina virtual
- Se utiliza principalmente en aplicaciones de propósito general y desarrollo web
- Se basa en Ruby on Rails para el desarrollo web

6. Python

- Es un lenguaje de programación
- De alto nivel

- Interpretado
- No utiliza una máquina virtual
- Se utiliza en una amplia variedad de aplicaciones, incluyendo aplicaciones de propósito general, ciencia de datos, desarrollo web y más
- Se basa en la sintaxis de Python

7. Cobol

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Se utiliza principalmente en aplicaciones empresariales y sistemas heredados

8. C

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Se utiliza en una amplia variedad de aplicaciones, incluyendo sistemas operativos, desarrollo de sistemas y aplicaciones de propósito general

9. Pascal

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Fue utilizado principalmente en la enseñanza y el desarrollo de aplicaciones científicas y educativas

10. Turbo Pascal

- Es una versión específica de Pascal
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Utilizado principalmente en la enseñanza y el desarrollo de aplicaciones

11. C++

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Se utiliza en una amplia variedad de aplicaciones, incluyendo desarrollo de sistemas, videojuegos y aplicaciones de propósito general
- No se basa en ningún framework específico

12. Objective-C

- Es un lenguaje de programación

- De alto nivel
- Interpretado (a través de la máquina virtual Objective-C)
- Utiliza una máquina virtual (Objective-C Runtime)
- Se utiliza principalmente en el desarrollo de aplicaciones iOS y macOS
- No se basa en ningún framework específico

13. Visual Basic

- Es un lenguaje de programación
- De alto nivel
- Interpretado o compilado, dependiendo de la implementación
- No utiliza una máquina virtual
- Se utiliza principalmente en aplicaciones de Windows y desarrollo rápido de aplicaciones (RAD)
- No se basa en ningún framework específico

14. Swift

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Se utiliza principalmente en el desarrollo de aplicaciones iOS, macOS, watchOS y tvOS
- No se basa en ningún framework específico

15. Ensamblador

- Es un lenguaje de bajo nivel
- De bajo nivel
- Compilado (generalmente a código máquina)
- Utiliza la arquitectura de la máquina directamente
- Se utiliza para la programación de bajo nivel y para optimizaciones específicas
- No se basa en ningún framework específico

16. Fortran

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Se utiliza principalmente en aplicaciones científicas y de ingeniería
- No se basa en ningún framework específico

17. Ada

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Se utiliza en aplicaciones críticas de alto rendimiento, como sistemas de defensa y aeroespaciales
- No se basa en ningún framework específico

18.Prolog

- Es un lenguaje de programación lógico
- De alto nivel
- Interpretado
- No utiliza una máquina virtual
- Se utiliza principalmente en aplicaciones de inteligencia artificial y procesamiento de lenguaje natural
- No se basa en ningún framework específico

19.Tcl

- Es un lenguaje de programación
- De alto nivel
- Interpretado
- No utiliza una máquina virtual
- Se utiliza en aplicaciones de automatización, scripts y desarrollo de interfaces de usuario
- No se basa en ningún framework específico

20.Simula

- Es un lenguaje de programación
- De alto nivel
- Interpretado o compilado
- No utiliza una máquina virtual
- Se utiliza principalmente en simulaciones y modelado de sistemas
- No se basa en ningún framework específico

21. Smalltalk

- Es un lenguaje de programación
- De alto nivel
- Interpretado
- Utiliza una máquina virtual (Smalltalk Virtual Machine)
- Se utiliza principalmente en aplicaciones de propósito general y en entornos de desarrollo orientados a objetos
- No se basa en ningún framework específico

22.Pperl

- Es un lenguaje de programación
- De alto nivel
- Interpretado
- No utiliza una máquina virtual
- Se utiliza en una amplia variedad de aplicaciones, incluyendo scripting, desarrollo web y administración de sistemas
- No se basa en ningún framework específico

23.Verilog

- Es un lenguaje de descripción de hardware (HDL, por sus siglas en inglés)

- De alto nivel en el contexto del diseño de hardware
- Compilado (sintetizado en circuitos electrónicos)
- Se utiliza para describir circuitos digitales y sistemas embebidos
- No se usa en aplicaciones de propósito general, sino en diseño de hardware
- No se basa en ningún framework específico

24.Scala

- Es un lenguaje de programación
- De alto nivel
- Compilado
- No utiliza una máquina virtual
- Se utiliza en aplicaciones de propósito general, desarrollo web y aplicaciones concurrentes
- Se basa en la JVM (Java Virtual Machine)

25.Lisp

- Es un lenguaje de programación
- De alto nivel
- Interpretado o compilado, dependiendo de la implementación
- Utiliza una máquina virtual (en el caso de implementaciones como Common Lisp)
- Se utiliza en aplicaciones de inteligencia artificial, procesamiento de lenguaje natural y desarrollo de sistemas avanzados
- No se basa en ningún framework específico

26.Node.js

- Es un entorno de ejecución de JavaScript en el servidor
- De alto nivel
- Interpretado
- Utiliza el motor de JavaScript V8 de Google
- Se utiliza principalmente en el desarrollo de aplicaciones web del lado del servidor
- Se basa en el ecosistema de JavaScript y npm (Node Package Manager)

4. Busca información sobre VB.NET y completa tus conocimientos con los vistos en el libro.

Dimas es un entusiasta de los lenguajes de programación. El otro día, en una revista, leyó algo relativo a VB.NET y, posteriormente, buscando en internet, accedió a la página de myfpschool.com aprendiendo a VB.NET.

VB.NET (Visual Basic .NET) es un lenguaje de programación que forma parte de la plataforma .NET de Microsoft. Es una evolución de Visual Basic, un lenguaje de programación popular que se ha utilizado durante décadas para el desarrollo de aplicaciones de Windows.

Algunas características clave de VB.NET son:

1. **Orientado a objetos:** VB.NET es un lenguaje de programación orientado a objetos, lo que significa que se basa en conceptos como clases, objetos, herencia, encapsulamiento y polimorfismo.

2. **Facilidad de uso:** Aunque ha evolucionado significativamente desde las primeras versiones de Visual Basic, VB.NET todavía se considera un lenguaje amigable para los principiantes. Su sintaxis es más legible y natural en comparación con algunos otros lenguajes.
3. **Integración con el entorno de desarrollo:** VB.NET se integra de manera excelente con Visual Studio, la suite de desarrollo de Microsoft. Visual Studio proporciona un entorno de desarrollo completo con herramientas de depuración, edición de código y diseño de formularios.
4. **Amplia biblioteca de clases:** VB.NET aprovecha la biblioteca de clases .NET Framework, que es una colección extensa de clases y bibliotecas que simplifican el desarrollo de aplicaciones. Esto incluye clases para trabajar con bases de datos, interfaces de usuario, redes y más.
5. **Seguridad y administración de memoria:** VB.NET se ejecuta en el entorno administrado de .NET, lo que significa que ofrece características de seguridad integradas y administración automática de memoria, lo que ayuda a prevenir errores comunes como fugas de memoria y desbordamientos de búfer.
6. **Soporte para múltiples plataformas:** Con .NET Core y .NET 5 (y versiones posteriores), VB.NET también se puede usar para desarrollar aplicaciones multiplataforma, incluyendo aplicaciones de escritorio, web y móviles.
7. **Uso en aplicaciones Windows:** Aunque .NET Core y .NET 5+ tienen un enfoque más amplio, VB.NET todavía se utiliza ampliamente en el desarrollo de aplicaciones de Windows Forms y Windows Presentation Foundation (WPF).

En cuanto a la referencia a myfpschool.com, es posible que sea un recurso en línea que ofrece tutoriales y cursos relacionados con VB.NET y programación en general. Puedes utilizar recursos en línea como este para aprender más sobre VB.NET y profundizar en tus habilidades de programación en el lenguaje.

5. ¿Qué lenguaje de programación crees que elegirán y cuáles son sus ventajas e inconvenientes?

Manuela pregunta a Dimas qué lenguaje de programación cree que va a elegir Emma para la tienda online. Tampoco saben por qué Emma elige el lenguaje de programación y no Andrino.

1. JavaScript:

- *Ventajas:* Ampliamente utilizado en el desarrollo web, soportado por todos los navegadores, gran cantidad de bibliotecas y frameworks (como React, Angular y Vue.js), lo que facilita la creación de interfaces de usuario interactivas y dinámicas.
- *Inconvenientes:* Puede ser propenso a errores si no se gestionan adecuadamente, y el ecosistema JavaScript es grande y en constante evolución, lo que puede hacer que sea difícil mantenerse actualizado.

2. Python:

- *Ventajas:* Fácil de aprender y leer, una amplia comunidad de desarrollo, utilizado tanto en desarrollo web (Django y Flask) como en aplicaciones de análisis de datos y aprendizaje automático, lo que lo hace versátil.
- *Inconvenientes:* No es tan rápido como algunos otros lenguajes, lo que podría ser un problema si se necesita un alto rendimiento.

3. **Ruby:**

- *Ventajas:* Ruby on Rails es un framework web popular conocido por su simplicidad y rapidez en la creación de aplicaciones web, promueve el principio de "convención sobre configuración", lo que acelera el desarrollo.
- *Inconvenientes:* Puede tener una curva de aprendizaje para quienes no están familiarizados con Ruby, y la flexibilidad de Rails puede ser limitante en algunos casos.

4. **PHP:**

- *Ventajas:* Ampliamente utilizado en el desarrollo web, especialmente en la creación de sitios web dinámicos, tiene una amplia base de usuarios y una gran cantidad de recursos en línea.
- *Inconvenientes:* Ha sido criticado en el pasado por problemas de seguridad, pero ha mejorado en ese aspecto con el tiempo.

5. **Java:**

- *Ventajas:* Fuerte en el desarrollo empresarial, escalabilidad, gran cantidad de bibliotecas y frameworks (como Spring), alto rendimiento y seguridad.
- *Inconvenientes:* Requiere más código en comparación con algunos otros lenguajes, lo que puede hacer que el desarrollo sea más lento.

La elección del lenguaje de programación dependerá de factores como la experiencia previa de Emma y Andrino, los requisitos específicos de su tienda en línea, el rendimiento necesario y las preferencias personales. Además, es importante recordar que, en algunos casos, se pueden utilizar múltiples lenguajes juntos, como JavaScript para la parte del cliente y Python o Ruby para el servidor, según las necesidades del proyecto.

6.¿ Qué crees que Dimas explicará a Manuela?

Dimas está emocionado con el nuevo lenguaje de programación que ha elegido Emma para e desarrollo.

Por su parte, Dimas está estudiando Angular.JS. Manuela quiere que Dimas le explique qué es y como funciona ese lenguaje.

Dimas explicará a Manuela que Angular.JS es un framework de desarrollo web que utiliza JavaScript y ofrece muchas características poderosas para la creación de aplicaciones web modernas. Le proporcionará una visión general de cómo funciona y por qué lo encuentra emocionante para el desarrollo de aplicaciones.

7. Dale una serie de consejos a Manuela.

Manuela y Dimas están emocionados con el nuevo proyecto. Han escuchado hablar a Andrino y Emma sobre las fases de desarrollo. No saben muy bien qué son las fases de desarrollo de su nuevo proyecto y qué tendrán que hacer en cada una de ellas.

A Manuela le han encargado la documentación. No sabe qué tiene que incluir en los documentos y si tiene que preparar uno o varios documentos. Está algo perdida.

Una serie de consejos para Manuela pueden ser:

Comprende las fases de desarrollo: Antes de comenzar a documentar el proyecto, es esencial que comprendas las fases de desarrollo. Puedes preguntar a Andrino y Emma o buscar recursos en línea para entender en qué consisten estas fases y cuáles son sus objetivos.

1. Identifica las necesidades de documentación: Cada fase del proyecto puede requerir diferentes tipos de documentación. Por ejemplo, en la fase de planificación, podrías necesitar un documento de alcance del proyecto. En la fase de diseño, podrías requerir diagramas de flujo o especificaciones técnicas. Identifica qué documentos son necesarios en cada etapa.
2. Colabora con Dimas y otros miembros del equipo: Trabaja de cerca con Dimas y otros miembros del equipo para recopilar la información necesaria. Pregunta a Dimas sobre los detalles técnicos y los requisitos del proyecto para asegurarte de que la documentación sea precisa.
3. Define una estructura clara: Crea una estructura para tus documentos que sea fácil de seguir. Por lo general, una documentación de proyecto incluye una introducción, objetivos, alcance, requerimientos, diseño, implementación, pruebas, y conclusiones. Asegúrate de que cada sección sea clara y esté bien organizada.
4. Mantén la documentación actualizada: A medida que el proyecto avance, es importante mantener la documentación actualizada. Asegúrate de registrar cualquier cambio o actualización en los documentos correspondientes.
5. Sé claro y conciso: Evita utilizar jerga técnica innecesaria en tus documentos. Debes asegurarte de que cualquier persona que lea la documentación pueda entender fácilmente lo que está escrito.
6. Revisa y verifica la documentación: Antes de finalizar cualquier documento, revísalo y verifica que esté completo y preciso. Puedes pedir a otros miembros del equipo que lo revisen también para obtener retroalimentación.
7. Mantén un registro de versiones: Si es necesario, lleva un registro de versiones de tus documentos para poder rastrear los cambios realizados a lo largo del proyecto.
8. Pregunta cuando tengas dudas: Si te sientes perdida o tienes dudas sobre qué incluir en la documentación, no dudes en preguntar a tus compañeros de equipo o a los líderes del proyecto. La comunicación es clave.
9. Aprende de la experiencia: A medida que avances en el proyecto y ganes experiencia en la documentación, trata de aprender de cada fase. Esto te ayudará a mejorar en futuros proyectos.

8. Realiza un listado de tareas y asígnaselas a los distintos roles de un desarrollo software.

Algunas tareas podrían ser:

- Cargar la base de datos con datos de prueba.
- Entrevista con el cliente para establecer requisitos.
- Elección de las herramientas de desarrollo.
- Elección del lenguaje de desarrollo.
- Pruebas unitarias.
- Pruebas finales.
- Crear la estructura de la base de datos.
- Establecer los requisitos del proyecto.
- Entrega de la primera versión del proyecto.
- Documentación técnica.
- Documentación de usuario.

Establece en qué paso del desarrollo software va a realizarse dicha tarea, con lo cual lo más eficiente es que elabores un cuadro con las siguientes columnas: tarea, paso de desarrollo y rol que la ejecuta.

Tarea	Paso de Desarrollo	Rol Responsable
Entrevista con el cliente	Inicio del Proyecto	Analista de Negocios
Establecer los requisitos del proyecto	Inicio del Proyecto	Analista de Negocios
Elección de las herramientas de desarrollo	Inicio del Proyecto	Arquitecto de Software
Elección del lenguaje de desarrollo	Inicio del Proyecto	Arquitecto de Software
Crear la estructura de la base de datos	Diseño de la Base de Datos	Ingeniero de Datos
Cargar la base de datos con datos de prueba	Desarrollo	Ingeniero de Datos
Desarrollo de módulos y funcionalidades	Desarrollo	Desarrolladores
Pruebas unitarias	Desarrollo	Desarrolladores
Integración de módulos	Desarrollo	Desarrolladores
Pruebas de integración	Desarrollo	Equipo de Pruebas
Pruebas de rendimiento	Desarrollo	Equipo de Pruebas
Pruebas funcionales	Desarrollo	Equipo de Pruebas
Pruebas finales	Desarrollo	Equipo de Pruebas
Documentación técnica	Finalización del Proyecto	Equipo de Documentación
Documentación de usuario	Finalización del Proyecto	Equipo de Documentación
Entrega de la primera versión del proyecto	Finalización del Proyecto	Gestor de Proyecto