

Glosario-Arquitectura.

Ivan David Palmar Martinez.

Instructor: Samuel Padilla.

SENA.

2023.

Glosario.

Clase 15/02/2023.

-Arquitectura: La arquitectura de software se ha ido descubriendo a medida que el tiempo y los proyectos han ido avanzando, esto quiere decir que hay ciertos estándares y tipos de arquitectura que se diferencian por tamaño de proyecto, la arquitectura de software conserva y tiene diferentes planificaciones basadas en los diferentes modelos de arquitectura (Cliente-Servidor, PipeLine, MVC, etc).

Relación: La relación que tiene la arquitectura de software con mi carrera (ADSO) es básicamente todo ya que se necesita una clara planificación para llevar a cabo los proyectos, esto quiere decir que se necesita una estructura clara desde el primer momento para realizar el proceso de creación de la solución de software de la mejor manera, implementar un arquitectura bien establecida es lo primordial.

-Modelado: El modelado sirve para simplificar todo lo que podría ser el sistema de información sin tanta complejidad, solo con los procesos más importantes y específicos que el mismo podría contener, esto a su vez permite una mejor comunicación con el cliente.

Relación: El modelado en mi carrera me permitirá conectar más con el sistema, modelar los procesos de una manera más sencilla para tener los objetivos más claros, por otro lado también tendría mejor comunicación con mi cliente ya que podría modelar sus necesidades para que queden plasmadas desde el primer día.

Clase 24/02/2023.

-Problemas esenciales : Los problemas esenciales se basan en aquellos que motivan a los equipos de proyecto, tal vez la complejidad del mismo es un reto muy grande y hay que buscar otras alternativas y realizar o repartir más tareas entre los mismo del equipo de proyecto, la tolerancia al cambio es un problema esencial porque siempre hay que tener en cuenta que el proyecto final tendrá que ir adaptándose a las situaciones de mercado, no es adecuado ni rentable hacer un proyecto plano que solo esté vigente por un corto periodo de tiempo, los problemas esenciales son simplemente aquellos que los equipos se irán encontrando a medida que el desarrollo del mismo se vaya ejecutando.

Relación: La relación principal con los problemas esenciales y mi carrera es que seguramente me enfrentare a muchos de ellos a lo largo de mi vida profesional, ya los estoy experimentado en mi primer proyecto con mis compañeros ya que a veces puede parecer bastante complejo el enfrentarse a retos con un gran tamaño o tener las ideas claras para que el aplicativo no se quede estancado, es mi primer proyecto esto me pasara a mayor escala a lo largo que avance mi carrera profesional.

-Problemas Accidentales: Los problemas accidentales son aquellos que nos podemos encontrar por ejemplo con los lenguajes de programación ya que encontrar el lenguaje de programación adecuado puede ser un reto, para que el equipo de desarrollo se sienta cómodo será muy clave un lenguaje que entiendan todas las partes del equipo, también los mismos entornos de programación donde se desarrollara el proyecto será indicado encontrar un área que facilite dicho problema, los problemas accidentales son aquellos que se irán dando a través de los recursos que se quieran disponer para la elaboración del proyecto.

-Prototipado rápido: El prototipado rápido son todas aquellas versiones del sistema que irán saliendo a lo largo del tiempo para ser presentadas al usuario y que este pueda evaluarlas, esto se hace con el fin de aclarar opciones sobre los requerimientos del usuario y aplicar dichas aclaraciones a las versiones posteriores del sistema.

Relación: En relación con mi carrera el prototipado rápido será un punto clave, por que no se puede ir de manera engegueda sin seguir unos parámetros dados por el usuario, entonces seguramente se tendrán que realizar muchas versiones o interfaces de usuario anteriores a la versión final, para que el mismo pueda dar aclaraciones de lo que le gusto o de lo que se tendrá que cambiar para la versión posterior del sistema, esto claramente trabajando metodologías ágiles, porque en metodologías tradicionales no es posible presentar versiones del proyecto al usuario.

Clase 27/02/2923

-Experto del dominio(Stakeholders): Los stakeholders en software son aquellas personalidades que se verán afectadas por el desarrollo del proyecto, pueden existir dentro o fuera de la compañía , estas personas tiene un grana interés en el producto final siendo considerados como los usuarios finales o las personas a las cuales va dirigido el proyecto. Existen cuatro tipos de stakeholders que son demasiado importantes para el desarrollo de un proyecto, estas personalidades son : Los inversores, los empleados, los clientes y los proveedores.

Relación: Estas personalidades a lo largo de mi carrera profesional me las encontrare en todos los proyectos y serán personas que me ayuden a crecer como

profesional ya que sabré cómo guiar mis proyectos, como cumplir sus necesidades a través de todas las indicaciones que me pueden proporcionar sin contar el levantamiento de requerimientos, estas personas siempre estarán en contacto conmigo ya que mi proyecto irá guiado a ellos.

-Analista: El analista en el mundo del software es el encargado de empezar por así decirlo con la ejecución del proyecto ya que el analista hará el proceso de levantamiento de requerimientos, reuniones con el cliente, entrevistas al cliente y toda la información que le pueda proporcionar el cliente para tener una guía clara de por donde se puede desarrollar el proyecto o cual metodología será la más indicada, es una persona muy importante en el desarrollo de sistemas de información.

Relación: Estoy casi seguro que existirán demasiados proyectos en los cuales yo tenga que cumplir con este rol, estar haciendo el levantamiento de requerimientos, estar en contacto constante con el cliente o usuario final para saber su punto de vista en el desarrollo del proyecto, en si sera un rol que cumplire y que tendre que hacer en mis proximos puestos de trabajo.

-Cliente(Dueño del producto): El dueño del producto o product owner es el encargado de decir las necesidades del cliente, este mantendrá un contacto especial con el mismo ya que actuará como la voz del cliente para que el equipo de proyecto tenga una visión más clara del producto final. Una de sus tareas más importantes es ordenar la bitácora de procesos haciéndola mucho más transparente, visible y entendible para el resto del equipo, en la bitácora es donde se llevará a cabo todo el seguimiento de las tareas propuestas para cada uno de los integrantes del equipo de desarrollo, aparte de llevar la bitácora y ser el encargado de explicar y decir las necesidades del cliente este también será el encargado de la toma de decisiones.

Relación: En algún punto de mi vida profesional tendrá nuevamente que cumplir este rol, está claro que cumplire con el mismo cuando ya tenga una experiencia bastante amplia en el mundo laboral de desarrollo de software, y después de haber pasado por todos los demás roles que están antes de este, tendré que cumplir con las tareas de un Product Owner que es un líder de proyecto básicamente es el guía y la persona que está encima del equipo de proyecto para que el mismo funcione de la mejor manera y su desarrollo se pueda llevar a cabo en el menor tiempo posible.

-Administrador de sistemas: El administrador de sistemas es la persona que está atenta y al mando de que el sistema funcione correctamente, este es el encargado del mantenimiento del sistema donde se ocupa de todas las quejas de los usuarios y va resolviendo todo los problemas que puedan ir surgiendo ya cuando el proyecto se haya desplegado.

Relación: Mi relación principal con este rol es que tendre o estare en contacto con el mismo ya cuando el proyecto pueda ser desplegado para notificar fallas previstas por los usuarios que ocupan el sistema de información, tal vez en algún punto también llegue a cumplir con este rol, ya sea realizando el mantenimiento del sistema o actualizando las prestaciones del mismo si es necesario.

-Devops: Los DEVOPS es una filosofía que promueve el desarrollo de aplicaciones o sistemas de información en mucho menos tiempo, es un marco de trabajo que demanda un despliegue rápido del sistema de información. Este marco de trabajo se utiliza para tener mayor agilidad al momento de realizar proyectos, siempre ocupando el menor tiempo para hacer las cosas más rápidas y eficientes.

Relación: El marco de trabajo DEVOPS me permitirá en mis futuros proyectos realizarlos de manera mucho más rápida y eficiente esto significa poder realizar el despliegue de proyectos completamente terminados en un menor tiempo, me

permitirá ganar experiencia de una manera más rápida y eficaz porque podré ganar conocimientos mucho más rápido realizando proyectos de este forma.

-Tester: Los testers o probadores de software son aquellos que prueban el sistema para identificar los distintos errores que él mismo puede sufrir, básicamente prueban el sistema hasta un punto donde pueda llegar a colapsar, ellos comunican los errores o fallos a los superiores para que puedan realizar el debido protocolo de mantenimiento al sistema.

Relación: Seguramente en algún punto de mi vida tendré que ser Tester o probador de software para comunicar los fallos o errores que encuentro dentro de el sistema de información, para realizar este trabajo tendré que ser demasiado bueno en cada uno de los procesos necesitados para desarrollar un sistema, ya que para testearlo necesito primero saber como crearlo.

-Desarrollador: Un desarrollador de software es la persona que se encarga de diseñar y crear un sistema de información o software a través de las distintas herramientas creadas para esto, los lenguajes de programación (Python, Java, C++), la interfaz que podrá observar el usuario final (Html, JS, CSS) y las diferentes bases de datos (MySQL, Sqlite3, Postgre, etc).

Relación: El papel de desarrollador creo que sera el mas importante en mi vida profesional aunque obviamente los demás roles son demasiado indispensables, simplemente en este rol iento que es donde se crea y se desarrolla principalmente el sistema de información, gracias obviamente a las indicaciones recibidas en los requerimientos pero dandole énfasis a los lenguajes de programación ,dándole importancia a la documentación del lenguaje donde se va a realizar y teniendo en claro cuál es la meta y tiempo para cumplir dicha meta, y siendo sincero esto me encanta ya que vivo por el aprendizaje de nuevas cosas que me ayuden en el futuro

en este caso aprender un lenguaje de programación nuevo es un mundo muy increíble y satisfactorio para mi.

-Arquitecto: Los arquitectos de software son las personas que, diseñan, desarrollan el sistema de software pero también son las personas que toman las decisiones de más alto nivel en el proceso de un sistema de desarrollo ya sea un software a la medida para un cliente específico o un sistema orientado a consumidores ya sean juegos o aplicación.

Relación: Cuando tenga la suficiente experiencia como para ser arquitecto de software seguramente tendré que estar a cargo de todo el proceso de toma de decisiones, inclusive creo que en este punto ya he hecho un papel de arquitecto con mi grupo de proyecto y creo que ellos también ya lo han hecho, por que hemos tomado decisiones importantes para el proyecto y hemos tanto diseñado como programado en visión al proyecto y de lo que el cliente principal (Tecnico en hoteleria) desea.

-Gestor del proyecto: El gestor de proyecto supervisa todas las acciones tomadas y procesos dentro del proyecto, de principio a fin, es como un supervisor que se encarga de supervisar debidamente cada proceso del proyecto a realizar.

Relación: Desde mi punto de vista el supervisor en este caso serían los instructores ya que ellos están al pendiente de las acciones que realizamos o vamos a tomar dándonos una guía de lo que sería más conveniente para nuestro sistema de información, ellos ya conocen cual es la meta final de nuestro proyecto y nos ayudan en el paso a paso supervisando nuestras decisiones.

-Facilitador: El facilitador o en otras palabras el scrum master es la persona que se encarga de llevar los procesos en el backlog, básicamente la gestión de plazo o tiempo de entrega de las diferentes tareas, también del presupuesto que se tiene y

se dispondrá para cada tarea y la supervisión de los miembros del equipo de proyecto, claramente hablando de metodologías ágiles, también será los encargados de hacer la planeación de los sprints.

Relacion: La relacion que el scrum master tiene con mi carrera es que seguramente en el futuro será mi jefe y la persona que esté a cargo de los procesos que yo realizare para el proyecto, también en estos momentos no siento que haya un scrum master para el proyecto pero veo total compromiso por parte de los instructores para cumplir este rol.

Clase 1/03/2023

-API: Una API es una interfaz de programación para el usuario básicamente conecta dos componentes que tienen alguna relación pero que están planeados totalmente diferentes y pueden estar creados en diferentes lenguajes de programación, pero se pueden conectar gracias a los distintos protocolos.

Relación: Las API serán un recurso muy importante en mi carrera ya que tendre que elaborar y entender las API's, este tipo de interfaz se ve más que todo en estilos distribuidos porque estos estilos tienden a conectar varias API's aun no he podido experimentar lo que es conectar un API con otra API, pero ya tengo algunas referencias de lo que son y de cómo se pueden crear, ya que son muy útiles para facilitar algunos procesos dentro del desarrollo.

-Http: El protocolo HTTP es aquel encargado de conectar los intercambios de información entre los servidores y los diferentes clientes, su función es principalmente conectar los datos realizados en la web con los clientes.

Relación: El protocolo HTTP está directamente relacionado con mi profesión ya que tendre que crear este tipo de urls a través de los protocolos esto quiere decir, que

cada interfaz de usuario que se cree para mi proyecto actual o para mis futuros proyectos serán con el protocolo HTTP o HTTPS.

-Url: La URL es la dirección o enlace único y totalmente específico que existe para cada página que existe en la web, cada una regularmente con el protocolo HTTP o HTTPS, el dominio y una dirección que las distingue de las demás o un nombre específico, las urls sirven para distinguir las diferentes páginas unas de otras.

Relación: Las URLS en mi vida profesional serán de muchísima importancia ya que las tendré que crear, investigar sobre ellas, para principalmente distinguir mi página web o las diferentes interfaces de usuario unas de otras, para que no haya repetición con ninguna ya existente.

-Timeline Service: El timeline Service representa el tiempo de vida del sistema, todo lo que conlleva crearlo, y lo que pasara despues con el sistema, su duración etapas de mantenimiento mientras está vigente y como se dará paso a su cierre o finalización, siempre teniendo en cuenta todos los momentos destacados para el desarrollo del mismo.

Relación: Siempre tendré que tener en cuenta el TimeLine Service ya que representa muchísimo cuando uno hace parte del proyecto esto me refiero a que siempre que yo pueda estar en un equipo de desarrollo es algo que tendré que discutir ya que es muy importante por el hecho de que se dispone el tiempo de vida útil del proyecto, sus etapas de mantenimientos. como la disponibilidad de donde se despliegue dicho sistema o por cuánto tiempo el sistema se encontrará en dicho servidor.

-Sistema de despliegue: El sistema de despliegue consiste en desplegar o sea montar la aplicación ya completamente desarrollada en un servidor para que los usuarios puedan consultar el aplicativo desde su dispositivo, el sistema de

despliegue puede ser en un servidor de pago o en un servidor gratuito esta etapa por lo general siempre es de las finales ya que el proyecto está terminado y solo queda realizar el proceso de despliegue en el servidor que el equipo de trabajo en conjunto haya elegido.

Relación: La etapa de despliegue es algo que aún está bien alejado en el punto en que se encuentra mi proyecto, pero ya he podido discutir con los integrantes de mi equipo como queremos que sea el despliegue, hemos decidido montar la aplicación en un servidor apache, en si la relación clara de esta etapa con mi carrera es que será el punto de cierre para cualquier proyecto que realice a lo largo de mi vida, será una de las ultimas y mas importantes etapas de cualquier proyecto porque de esto depende como funcionara el sistema en la web.

-Equipo de trabajo: Los equipos de trabajo en el mundo del software se distinguen por tener personas plenamente establecidas para cada tarea, empezando por el analista que es el encargado de realizar todo el proceso de levantamiento de requerimientos y de reuniones con los clientes, el diseñador que es el encargado de realizar todos los diagramas UML según los requerimientos que hizo el analista, seguidamente están los desarrolladores que según los diagramas UML que diseñaron los diseñadores harán toda la parte de programación del aplicativo, estos son unos de los roles más importantes de un equipo de trabajo, sus tareas se irán anotando en el Backlog del proyecto.

Relación: Creo que los equipos de trabajo en esta carrera son indispensables es imposible o casi imposible pensar en un proyecto de software solamente desarrollado por una persona, claramente es posible pero con los equipos de proyecto se hace muchísimo mas facil y sencillo realizar todo este tipo de tareas que conllevan el desarrollo del proyecto, en si lo que he visto hasta ahora me ha gustado

mucho porque creo que en un equipo de proyecto se puede repartir mucho mas fácil la carga de trabajo para cada integrante.

Clase 3/03/2023

-Requerimientos funcionales: Los requisitos funcionales son todas las funciones internas que tendrá el sistema, básicamente como el sistema debe comportarse frente a diferentes acciones que los usuarios vayan a realizar dentro del aplicativo.

Relación: Siempre que me toque cumplir el rol de analista tendré que separar los requerimientos que levante en el proceso de levantamiento de requerimientos tanto en requerimientos funcionales como no funcionales, principalmente los funcionales serán muchísimos más y son un punto demasiado fuerte para mi carrera ya que de ellos depende la funcionalidad del sistema de información cuando ya esté operando en la web.

-Requerimientos no funcionales: Los requerimientos no funcionales son todos aquellos que no tienen que ver directamente con el funcionamiento final de un sistema de información, si no los que son más de apariencia, como los colores de las interfaces o los diseños de las mismas que el usuario final tendrá para observar cuando ya el sistema de información sea desplegado.

Relación: Serán una parte importante de mi carrera ya que en ellos cuando me toque el papel de analista podrá especificar según las necesidades del cliente, pues todos los colores o diseños que él prefiera para el aplicativo final.

-Etapa de diseño: La etapa de diseño es en la cual se dejan listos y completamente terminados los requerimientos de bases de datos y de las redes, también esta etapa es muy clave para el desarrollo porque se crean todas las interfaces de usuarios

gracias a los programadores y a las indicaciones de los diseñadores y entre ambos aclaran todas las funcionalidades de cada interfaz.

Relación: La etapa de diseño va a ser en mi vida profesional un punto demasiado clave ya que según el rol que posea en la compañía o proyecto podría como diseñador dejar los requerimientos de bases de datos y redes ya terminados para que empiece la programación o si hago el rol de programador en el proyecto en esta etapa me divertire mucho ya que podré con los requerimientos dados por el diseñador realizar todas las diferentes interfaces de usuario y definir sus funcionalidades.

-Documentación: La documentación en software es toda la información enfocada para describir el sistema en funcionalidades y características, tanto para las personas que desarrollan el sistema, como para los usuarios finales la documentación del sistema les permitirá tener una referencia clave de lo que es capaz el mismo.

Relación: La documentación en mi carrera no solo será importante, será importantísima tanto para mi desempeño en el equipo de proyecto como para mi propio aprendizaje dentro de las características de los sistemas, ya que la documentación me permite estudiar más acerca de los proyectos ya creados y de los que se van a crear, también para estudiar y comprender las herramientas que se van a usar para el desarrollo del proyecto.

-Prioridades del backlog: Las prioridades del Backlog se evalúan según los diferentes criterios de prioridad, osea el tamaño o la dificultad de la tarea, si la tarea tienen mayor dificultad debe de estar dentro de la prioridades principales de la bitácora, esto quiere decir que son tareas que se cumplan con anterioridad en relación con los demás.

Relación: Las prioridades de la bitácora serán muy importantes cuando yo pueda cumplir el rol de product owner ya que tendré que tener muy en cuenta la prioridad de la tarea que se va a realizar y así mismo planear mi proyecto para que no se crucen las tareas o que estén mal ubicadas cuando controle la bitácora del proyecto.

-FeedBack: El feedback es muy importante en un sistema de información ya que es el encargado de recopilar todas las opiniones de los usuarios cuando el aplicativo ya se encuentra en la web, este feedback nos permite controlar opiniones y tener un retroalimentación del sistema gracias a los usuarios. para saber en que se puede realizar un proceso de mantenimiento.

Relación: El feedback me podrá guiar a través del mantenimiento que se requiere hacer o los diferentes cambios que se podrían aplicar al sistema, también es una parte que totalmente tendré en cuenta ya que gracias a los usuarios y sus opiniones y criterios se puede enfocar de una mejor manera los siguientes pasos del sistema.

clase 4/03/2023

-Requerimientos de producto: Los requerimientos de producto son aquellos que describen la prestación o funcionalidad que tendrá el producto cuando esté desarrollado y también sirven para describir para qué tipo de personas irá enfocado el sistema cuando se realice su despliegue.

Relación: Serán una clave o apartado muy importante ya que gracias a ellos a lo largo de mi carrera profesional podré distinguir que tipo de proyecto o sistema es el que se quiere realizar, también podrá identificar para qué tipo de población o clientes va dirigido, como lo estoy haciendo en este momento, en el proyecto ya realizamos la parte de funcionalidad y de para quién va dirigido en este caso va dirigido a el tecnico en hoteleria y turismo de la sede de Sibaté.

-Requerimientos de negocio: Los requerimientos de negocio son los que primero se plasman en la realización de un proyecto, porque son los que se refieren principalmente a los objetivos que se quieren lograr esto contando tanto para objetivos como mejoras que se podrían aplicar al negocio para hacerlo mucho más eficiente.

Relación: Los requerimientos de negocio son los que se planean y plantean al principio de un proyecto entonces seguramente serán la base para construir el gran sistema o aplicativo, me tendré que relacionar demasiado con este tipo de requerimientos para llevar mi nivel de comprensión de requisitos a un nivel muy alto, para la elaboración de un proyecto enfocado al negocio (mercado) .

-Requerimientos de usuario: Los requerimientos de usuario son aquellos que plantean y especifican las necesidades que el cliente tiene y que se tienen que incluir al sistema o aplicativo, son todas aquellas instrucciones que el cliente específico cuando se hizo el levantamiento de requerimientos, como los módulos (Reserva, Acceso y facturación) o los colores y diseños.

Relación: Ya en mi proyecto con mi grupo de proyecto hicimos la fase de levantamiento de requisitos de usuario, en total nos salieron 12 requerimientos que serán aplicados al proyecto final donde el cliente en este caso el área de turismo nos especificó sus necesidades y lo que tenía que tener obligatoriamente el sistema, con esto quiero decir que es una etapa por la cual pasaré bastante ya que tendré que hacer y evaluar requerimientos de usuario en algún punto de mi vida.

-Requerimientos de sistema: Los requerimiento de sistema son todos esos requerimientos que describen los servicios que el sistema tiene que ofrecer al usuario final según lo que él mismo pidió en el levantamiento de requerimientos, y

también sus respectivas restricciones asociadas al funcionamiento del producto final.

Relación: Hasta ahora he tenido que llenar un montón de documentos con requerimientos del sistema y me he dado cuenta que son aquellos que especifican las diferentes acciones que el sistema debe ofrecer pero por otro lado también especifican los que se tiene que restringir para que el sistema sea lo más eficiente y eficaz posible.

-Requerimientos de proyecto: Son aquellos en los cuales se identifican los recursos, se especifican especialmente en las licencias que tendrá el aplicativo, los acuerdos de servicio, la infraestructura o sea se definirá una estructura clara para el proyecto o una arquitectura, son demasiado importantes ya que recalcan que tendrá que tener el proyecto.

Relación: Estos requerimientos los llegue a manipular cuando yo y mi equipo realizamos la propuesta técnica del proyecto donde especificamos en toda la documentación como íbamos a realizar los diferentes requerimientos del proyecto o sea toda la parte de infraestructura y de acuerdos de servicio que tendremos para el desarrollo eficaz de nuestro proyecto.

Clase 6/03/2023

-Riesgos: Los riesgos son partes muy importantes y que seguramente se irán dando a medida que el proyecto vaya creciendo son simplemente problemas que se van a ir encontrando por el camino tales como; el desconocimiento de la tecnología base del proyecto esto quiere decir que siempre se tiene que estudiar la documentación de las herramientas que se utilizaran para el desarrollo del sistemas de información, otro problema muy común es el alto nivel de complejidad técnica

osea cuando la solución o problema planteado requiere una tecnología o niveles demasiado complejos para llevar a cabo el desarrollo y otro problema o riesgo muy común es la necesidad de usar tecnología inmadura, osea tecnología que ha sido creada o desarrollada muy recientemente esto es un riesgo ya que no pasa como en Java C++ que hay mucho código escrito, claro no hay nada de malo en escribir código el problema llega cuando se necesitan ciertas referencias y no hay ni siquiera información clara de la herramienta en su documentación.

Relación: Seguramente en mi carrera y cuando desarrolle sistemas de información de alto nivel, tendré que tener muy en cuenta los riesgos que pueden ir apareciendo con el tiempo, creo que hasta el momento no he sufrido con mi grupo de proyecto ninguno de los riesgos anteriores, tal vez el desconocimiento en alguna de las tecnologías pero eso se ha ido arreglando con el tiempo, ya que aún nos queda bastante por aprender.

-Atributos de calidad: Los atributos de calidad son aquellos que se tienen que seguir en el desarrollo de un sistema para considerarlo de calidad, los 5 principales y que cumplen con el lineamiento de un software de calidad son; Funcionalidad, confiabilidad, usabilidad, rendimiento y mantenimiento.

Relación: En mi opinión para el desarrollo de un buen software a la medida es cumplir con los lineamientos de calidad, ya que si se cumple con cada uno de ellos se puede estar seguro primero que el proyecto será buenamente aprobado y segundo que no será algo del momento sino que de verdad se usará y tendrá un alcance mayor al esperado con el cliente, digo esto ya que tratare de aplicar los 5 atributos principales de calidad a cada uno de los proyectos en los que participe.

-Conocimientos del dominio: El conocimiento del dominio se trata en conocer o estar especializado y bien preparado en relación con el entorno de la aplicación,

preparase para su propio desarrollo, tener conocimientos claros en que se desea y cómo se hará.

Relación: Mi meta como profesional o como tecnólogo será ser un experto en todos los diferentes conocimientos de los dominios que existen para el desarrollo de software, simplemente comprenderlos y aprenderlos de punto a punto ya que es una ventaja impresionante saber tanto y especializarse sobre un tema tan importante como el conocimiento de lo que está realizando cuando se crea un nuevo aplicativo.

-Restricciones: Las restricciones casi como los riesgos son aquellas directivas que uno plasma desde el inicio del proceso de desarrollo pero estas son las que no se pueden realizar desde un primer momento, siempre van a haber restricciones tanto de tecnología como a nivel de presupuesto pero las restricciones más conocidas son; restricción de efectivo, restricción de la oferta, restricción de la demanda y la restricción del mercado.

Relación: Las restricciones son cosas que trataré siempre de tener muy en cuenta simplemente por el hecho de que son las directivas o protocolos que no se pueden realizar obligatoriamente, ya que quedan plasmadas en la conversación con el cliente o usuario final, siento que de estas restricciones tal vez a veces con mi equipo de desarrollo tratamos de sobre pasar pero pensamos más claramente y sabemos lo que hay en la propuesta técnica que es básicamente lo que se tiene que cumplir finalmente.

Clase 10/03/2023

-Interfaz pública: Las interfaces públicas son todas aquellas que pueden ser visualizadas y manipuladas por el usuario final en la pantalla de su dispositivo, esto se puede conocer como front-end que consiste en el desarrollo de todas las

interfaces gráficas que el usuario visualizará según las especificaciones en el levantamiento de requerimientos.

Relación: En este punto del desarrollo del proyecto yo y mi equipo de proyecto nos encontramos realizando todas las diferentes interfaces gráficas, según los módulos que especificamos en grupo de proyecto y gracias a los que el cliente suministro o pudimos sacar de los requerimientos, menciono esto para dar a entender que ya estoy en el proceso de aprender todo lo relacionado con las interfaces gráficas o públicas donde se encuentran todos los diseños y casi todos los requerimientos no funcionales.

-Arquitectura multinivel-Cliente servidor: La arquitectura cliente-servidor consiste en que los demandantes en este caso los usuarios o clientes, hacen peticiones al sistema de información el cual reparte sus tareas en los diferentes conocidos proveedores de recursos, ellos básicamente son los servidores, un gran ejemplo para definir este modelo de arquitectura es; Google ya que nosotros a través de un equipo (computadora o teléfono móvil) hacemos peticiones a a el sistema y este las ejecuta según la medida en que el pueda, en la parte de la base de datos, los datos se encuentran almacenados de forma remota y persistente, en la parte del usuario o que el usuario visualiza en la página casi siempre se utiliza JavaScript que es para recibir y asociar las interacciones del usuario con la pagina, y por último por el lado del servidor y de la parte lógica casi siempre se manejan lenguajes como: Python, Java, Ruby, Php, etc.

Relación.

La relación que encuentro con este modelo para mi carrera es que seguramente lo tendré que usar si sigo en ella, ya que como sus metodologías indican es una arquitectura demasiado utilizada en la actualidad desde hace unos años atrás,

tambien a dia de hoy la estamos implementado con nuestro proyecto ya que se puso en decisión que usamos un modelo de arquitectura.

-Rest API: La REST API es aquella interfaz donde dos sistemas de información se juntan (pueden no tener nada en común) para realizar un tarea específica, estos se juntan a través de internet de una manera totalmente segura, la mayoría de empresas o clientes grandes donde el sistema requiere de una alta capacidad y mucho desarrollo, para agilizar tareas a veces se compran o rentan diferentes aplicaciones de terceros, por ejemplo una muy conocida para procesar los pagos virtuales es mercado pago que renta sus servicios por un módico precio, esto se usa cuando el estilo de desarrollo será un estilo distribuido basado en microservicios.

Relación: Hasta ahora en el siguiente trimestre vamos a entrar con tema de las REST API pero no de manera muy profunda ya que nuestra aplicación tendrá un estilo monolítico osea no será de microservicios, pero entiendo que el concepto de REST API lo usaré de forma adecuada cuando desarrolle sistemas con estilo distribuido con los cuales podré conectar las diferentes API's.

-Back-End: El back-end es toda la parte lógica que el usuario final no podrá ver ni observar, básicamente donde se encontrarán todos los procesos de consultas y todos los procesos que se desarrollan en la interfaz tendrán que tener programación el que cliente no verá, el front-end por lo general se desarrolla con diferentes frameworks para el lenguaje de programación que se esté usando para el desarrollo (Python, C++, Java, etc).

Relación: El back-end es en lo que me quiero especializar, simplemente me fascina lo que tiene que ver con aprender lenguajes de programación, me gusta la complejidad que los mismos dan, por las cosas tan increíbles que se pueden realizar con ellos, los algoritmos que uno puede hacer son muy increíbles cuando se

realizan de buena forma y el resultado es el esperado, el back-end tiene demasiado que ver con mi carrera ya que para realizar el sistema siempre se necesitará la parte de la lógica.

-Postman: Postman es un sitio web totalmente gratuito que se usa para hacer la creación y utilización de APIs mucho más sencilla, el sitio web permite hacer pruebas para comprobar que las diferentes API 's que se van a usar funcionan correctamente.

Relación: Será un sitio web que seguramente visitare bastante para hacer pruebas con las API 's que utilizare para mi sistema que estoy realizando ahora o para los sistemas que realizare o desarrollare en un futuro, ya sean las propias mías que yo cree o las que se puedan reutilizar y no tengan derechos de autoría.

Clase del 13/03/2023

-Flujo de datos: El flujo de datos en software es como se describen y almacenan los diferentes datos en una instancia del proceso, obviamente el aplicativo tiene que estar ya en la web para que se tenga el control de datos en el flujo del programa.

Relación: El flujo de datos como todas las partes del sistema será indispensable porque de este flujo de datos depende de que el sistema no se caiga ni colapse en algún punto, el flujo de datos tiene que estar bien establecido desde la creación del proyecto, cosa que se relaciona bastante con lo que yo y mi grupo de proyecto hemos estado haciendo hasta ahora ya que hemos planteado todo para que no hayan estas cargas o el sistema pueda llegar a quebrarse en algún punto.

-Componentes: Los componentes del software son los que ofrecen servicios ya plenamente predefinidos y son capaces de conectarse con otros componentes para cumplir con un objetivo o tarea específica.

Relación: Estos componentes son todas las partes interconectables que ofrece el sistema que se va a desarrollar, cada parte del sistema es un componente del mismo, esto quiere decir que en mi carrera profesional siempre tendré que tener en cuenta que componente del sistema de información estoy realizando para saber que conexión hará con los demás componentes.

-Esquema de datos: Un esquema de datos comprende toda la configuración lógica de una base de datos relacional, esto quiere decir los diferentes campos y datos que tendrán la diferentes tablas, también se refiere a las conexiones que habrá entre tablas, todo este esquema recae para que no haya redundancia en los datos que se ingresaran para ser procesados, muy parecido a la normalización.

Relación: El esquema de datos es un proceso indispensable para cualquier sistema de información, porque todos los sistemas deben de tener una base de datos orientada por un esquema de datos para las bases de datos relacionales si la base de datos es no relacional no importa mucho el esquema de datos, con esto quiero decir que en mi proyecto con mi grupo de proyecto y gracias los distintos instructores ya hemos completado nuestro esquema de datos, a través de la normalización la base de datos ha quedado sin ninguna redundancia en los datos, puedo decir que ya soy competente para configurar de un forma lógica una base de datos.

Clase del 15/03/2023

-Invocación explícita: La invocación explícita en software es la que permite como poder hacer que dos o más componentes se conozcan entre sí aunque se hayan desarrollado de forma independiente y que esto no cause ningún fallo en el sistema.

Relación: Seguramente en algún punto de mi vida profesional necesitare usar este tipo estructura para el sistema que esté desarrollando, ya que seguramente cuando el sistema a desarrollar sea de alto nivel se tendrá que crear los componentes por separados para posteriormente unirlos y que el sistema fluya como normalmente lo haría con la invocación implícita.

-Invocación implícita: La invocación implícita consiste en cómo hacer que las aplicaciones puedan mandar mensajes o interacciones entre sí sin la mera necesidad de saber quien es el que está recibiendo y contestando dichos mensaje lanzados por el sistema, la invocación implícita tiene dos partes super importantes que son; Publicar y suscribir y el Enterprise Service Bus.

Relación: Este tipo de invocación seguramente la aprenderé cuando avance aún más el curso pero se me hace muy interesante este tipo de estructura ya que permite realizar acciones medio anónimas solo con la manipulación del sistema.

-Orientación a servicios: Esta arquitectura que es la de Microservicios se hace cargo de dividir la aplicación en diferentes servicios que son totalmente independiente, pero se comunican a través de una API o GateWay que es la entrada o puerta de acceso de cada una para que el usuario pueda interactuar con cada una de ellas en el mismo sitio web o aplicativo, esto se hace para no programar o escribir código innecesario o sea para contar con mayor rapidez y agilidad en el proceso de creación del aplicativo y también para no pasar por procesos legales que llevaría implementar servicios directamente programados por los programadores.

Relación: Este tipo de arquitectura, me refiero al tipo de arquitectura orientada a servicios, seguramente tendré que experimentar este tipo de arquitectura en mis próximos proyectos, creo que tendrá un gran impacto en mi tanto como profesional

como para mejorar mi nivel de aprendizaje en el tema de arquitecturas ya que se dice que es la arquitectura más usada hoy en día, simplemente por que si se cae una parte del sistema no colapsa todo el sistema, solo colapsa la parte en la que hubo la caída, esto puede ayudar muchísimo al proceso de mantenimiento ya que no se tiene que encontrar el problema dentro de un sistema gigantesco solo buscar la parte que está colapsada y arreglarla.

-Estilos monolíticos: El estilo monolítico permite darle prioridad a la eficiencia entre las comunicaciones de los distintos componentes, tiene muchísima más eficiencia ya que solo se hace uso de un módulo no es una aplicación modular o sea prioriza el rendimiento, la curva de aprendizaje del sistema es más fácil ya que todas las piezas del mismo se encuentran en un solo lugar, es mucho más fácil modificar este estilo ya que permite la modificación de una manera más rápida y la capacidad de prueba es muchísimo mayor.

Relación: Este estilo ya está teniendo impactos en mi carrera porque el estilo monolítico es el que vamos a usar para el proyecto y sus diferentes fases de desarrollo, principalmente elegimos este modelo porque es más sencillo y rápido de desarrollara y no contamos ni con los recurso o tiempo para desarrollar una aplicación modular, pienso que en mi vida profesional tendré que aprender mucho sobre este estilo ya que lo aplicare bastante en aplicativos que no sean necesariamente de un alto nivel de complejidad.

-Estilos distribuidos: El estilo distribuido consiste en permitir la modularidad del aplicativo teniendo como ventajas, la disponibilidad del sistema cuando se requiera alterar cualquier módulo ya que los componentes son desplegados independientemente, el uso de diferentes recursos dando prioridad a todos por igual.

Relación: Este fue el estilo que como grupo pensamos en integrar primero a nuestro aplicativo, pero por cuestiones de tiempo no pudimos incorporarlo, pero ya en mi vida profesional siento sinceramente que será un estilo demasiado clave para tener en cuenta al momento de realizar un nuevo sistema ya que se nota que es mucho más sencillo y específico desarrollar los aplicativo con el estilo distribuido.

Links.

- <https://www.scrum.org/resources/what-is-a-product-owner#:~:text=The%20Product%20Owner%20is%20accountable,is%20transparent%2C%20visible%20and%20understood>
- <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- <https://www.northware.mx/blog/7-errores-comunes-en-proyectos-de-desarrollo-de-software/>
- <http://www.sidar.org/recur/desdi/traduc/es/visitable/nuevos/Rapido.htm#:~:text=Prototipado%20R%C3%A1pido&text=%C2%BFEn%20qu%C3%A9%20consistie%3F,hardware%2C%20para%20su%20posterior%20evaluaci%C3%B3n.>
- <https://www.conceptatech.com/blog/how-to-define-stakeholders-for-your-software-development-project#:~:text=The%20term%20%E2%80%9Cstakeholder%20refers%20to,interest%20in%20the%20final%20product.>
- <https://treball.barcelonactiva.cat/porta22/es/fitxes/A/fitxa5905/administradora-de-sistemas.do#:~:text=Se%20encarga%20de%20que%20el%20conjunto%20del%20sistema%20funcione%20correctamente,los%20problemas%20que%20les%20surgen.>

- <https://www.netapp.com/es/devops-solutions/what-is-devops/#:~:text=DevOps%20es%20un%20marco%20de,o%20productos%20para%20los%20clientes.>
- [https://www.educaweb.com/profesion/probadores-software-testers-238/#:~:text=Los%20probadores%20de%20software%20\(tambi%C3%A9n,detectan%20errores%20y%20los%20comunican.](https://www.educaweb.com/profesion/probadores-software-testers-238/#:~:text=Los%20probadores%20de%20software%20(tambi%C3%A9n,detectan%20errores%20y%20los%20comunican.)
- <https://www.edix.com/es/instituto/desarrollador-software/>
- <https://www.freelancermap.com/blog/es/que-hace-arquitecto-software/#:~:text=Los%20arquitectos%20de%20software%20dise%C3%B1an,en%20el%20proceso%20de%20desarrollo.>
- <https://www.wrike.com/es/project-management-guide/faq/cuales-son-las-funciones-y-responsabilidades-de-un-gestor-de-proyectos-para-tener-exito-en-el-trabajo/>
- <https://resources.workable.com/es/facilitador-de-proyectos-descripcion-del-puesto>
- <https://aws.amazon.com/es/what-is/api/#:~:text=Las%20API%20son%20mecanismos%20que,meteorolog%C3%ADa%20contiene%20datos%20meteorol%C3%B3gicos%20diarios.>
- <https://keepcoding.io/blog/que-es-el-protocolo-http/#:~:text=El%20HTTP%20o%20HyperText%20Transfer,servidores%20y%20orientada%20a%20transacciones.>
- https://developer.mozilla.org/es/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL
- https://es.wikipedia.org/wiki/Despliegue_de_software

- <https://www.startechup.com/es/blog/software-development-team/#:~:text=Una%20equipo%20de%20desarrollo%20de%20software%20es%20un%20grupo%20de,rendir%20cuentas%20de%20sus%20esfuerzos>.
- <https://visuresolutions.com/es/blog/functional-requirements/#:~:text=Un%20requisito%20funcional%20es%20una,caracter%C3%ADsticas%20que%20el%20usuario%20detecta>.
- <https://rootstack.com/es/blog/cuales-son-las-fases-en-el-desarrollo-de-software#:~:text=los%20usuarios%20finales-,Etapa%20de%20dise%C3%B1o,base%20de%20datos%20y%20redes>.
- <https://keepcoding.io/blog/que-es-la-documentacion-de-software/#:~:text=La%20documentaci%C3%B3n%20de%20software%20se,como%20una%20documentacion%20de%20procesos>
- <https://muyagile.com/que-es-el-backlog/#:~:text=los%20%C3%ADtems%20del%20Backlog%20deben%20estar%20priorizados%2C%20esto%20es%2C%20ordenados,posici%C3%B3n%20dentro%20de%20la%20pila>.
- <https://www.freshworks.com/latam/freshdesk/customer-feedback/#:~:text=Un%20software%20de%20feedback%20de,de%20varios%20canales%20de%20comunicaci%C3%B3n>.
- <https://www.ibm.com/docs/es/spm/7.0.0?topic=start-defining-product-requirements>
- <http://www.pmoinformatica.com/2020/12/tipos-de-requerimientos-analisis-negocio.html#:~:text=Los%20requerimientos%20de%20negocio%20son,que%20quiere%20lograr%20la%20organizaci%C3%B3n>.
- <https://qbdgroup.com/es-es/blog/validacion-de-sistemas-informatizados-requisitos-de-usuario-urs/>

- <https://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>
- <https://www.linkedin.com/pulse/qu%C3%A9-es-un-requerimiento-en-el-proyecto-jos%C3%A9-daniel-esterkin/?originalSubdomain=es>
- <https://www.piranirisk.com/es/blog/gestion-de-riesgos-proyectos-de-software>
- <https://osvaldogalvan.com/blog/5-atributos-de-calidad-de-software/>
- <https://gravitar.biz/tecnologia-negocios/teoria-de-restricciones/>
- <https://rockcontent.com/es/blog/interfaz-de-usuario/#:~:text=Interfaz%20de%20software,la%20pantalla%20de%20su%20dispositivo.>
- <https://aws.amazon.com/es/what-is/restful-api/#:~:text=La%20API%20RESTful%20es%20una,llevar%20a%20cabo%20varias%20tarefas.>
- <https://assemblerinstitute.com/blog/que-es-postman/#:~:text=Postman%20es%20una%20plataforma%20que,una%20extensi%C3%B3n%20en%20Google%20Chrome!>
- <https://www.ibm.com/docs/es/spm/8.0.2?topic=document-data-flow>
- <https://www.lucidchart.com/pages/es/que-es-un-esquema-de-base-de-datos#:~:text=Un%20esquema%20de%20base%20de%20datos%20representa%20la%20configuraci%C3%B3n%20de%20la%20l%C3%B3gica,controlan%20una%20base%20de%20datos.>
- <https://github.com/MineiToshio/CursosPlatzi/blob/master/Fundamentos%20de%20Arquitectura%20de%20Software/README.md>

