

Sistema de información 2 es también conocido como software de gestión o software de gestión empresarial.

Este tipo de software tiene 3 características

- Trabajan con cantidades enormes de datos, con todo tipo de bases de datos, en la materia se desarrollara un software que trabajará una gran cantidad de datos. Para poblar la base de datos se usará dataset.
- El software también esta diseñado para trabajar con múltiples usuarios simultáneamente con distintos roles ávidos y por haber, con permisos diferentes para estos roles desde diferentes tipos de dispositivos en diferentes idiomas. Prever el uso simultáneo de datos, también la trazabilidad o bitácora.
- Se tiene que implementar políticas de negocio de las empresas, las forma de trabajar de la empresa, conocido como ingeniería de requisitos

Se realizará en el semestre un proyecto principal, y proyectos de exámenes

Estoy en el Grupo 8 tema: recursos humanos.

Se desarrollara con la finalidad de ser un software de suscripción.

Traer un informe sobre el tema a desarrollar con 3 ejemplos, ver algunos síntomas como ejemplo, preguntas que serán aclaradas por el ingeniero

Clase 13 de agosto

Conseguir 3 aplicaciones echas en cualquier parte del mundo que se hayan echo en cualquier parte del mundo, almenos una debe ser open source y documentar, suscripciones, requerimientos que se necesitan, aprender a utilizar y documentar con capturas de pantalla, ni no se consigue captura adjuntar aspectos técnicos, traer siempre 2 caratulas, una para el docente y otra para el registro, sacar fotos a éstas caratulas con notas.

L caratula debe contener logo de la universidad, grupo, integrantes, nombre del documento, título del proyecto o tema a tratar, lista de integrantes: apellido paterno, materno y nombre ordenado de forma alfabética a través del apellido paterno, fecha dela presentación y el lugar: santa cruz de la sierra, atento a la entrega por la plataforma previo registro en grupos de la materia.

Clase 15 de agosto

Fecha de entrega final del proyecto será 2 semanas antes del último día de clases según calendario actual sin tomar los subsecuentes.

Habrán revisiones intermedias, serán válidos si se tienen las presentaciones previas.

El proyecto debe ser extraordinario.

Se debe conocer el proyecto de los compañeros entendiendo el contexto, 2 exámenes parciales 20 cada uno, un proyecto final 30 proyecto 30.

Habrán prácticos, el examen final será paralelo con el otro horario, el examen será el mismo, sin segunda instancia ni repechaje.

La presentaciones inciden en la nota final

Para la siguiente clase de manera formal, especializarse en el tema que se tratará en I proyecto final, entendiendo el contexto del sistema.

1. Fundamentación teoría e histórico necesario para entender el tema, leyendo 3 libros.
Documentar en un resumen para exponer.
Adjuntar citas biográficas.

Todos deben saber el tema, hará preguntas a los integrantes sobre el tema a desarrollar.
Colocar el la caratula los nombres de los que trabajaron

2. Buscar 3 aplicaciones, sistemas hechas en cualquier parte del mundo, tiene que ser del tipo comercial, desarrollado por una empresa que comercializa software, se tiene que aprender a usar la aplicación, demostrar como se usa en clases como un taller, de las 3 por lo menos una o 2 debe ser open source, ponerla a correr de manera local o en la nube.

3. Conocer el mercado, la demanda para conocer el mercado, ver 3 casos de estudio como potenciales clientes representativos.

4. En el futuro se pedirá innovación.

5. Traer diapositivas, cuando toque presentar el programa, tener planes de contención, grabar un video demostrativo sin audio editado centrado en las partes fundamentales de la aplicación, nosotros debemos explicar los paso. Tener todo público en OneDrive para compartir con el resto de la clase.

6. La presentación tendrá un tiempo de 40 minutos, ni más ni menos, sino será catalogado como mal, prohibido leer, hacer simulación previa de la explicación, así diapositivas llenos de texto.

7. Asegurarse de que un 70% será demostrando los 3 software, el 30% será la parte teórica,

8. El ingeniero puede definir quien expone para día miércoles 20 de agosto, hasta el inicio de la clase se subirá la documentación.

Clase 18 de agosto

3 referencias en la fundamentales, basado en 3 libros, normativa internacional y normativa boliviana, Consultar con conocidos del área

Clase 22 de agosto

Para la próxima traer propuestas de desarrollo,

- entender el problema.
- Crear, diseñar la solución óptima como solución del problema.
- Escribir código.
- Asegurarse que el software esté libre de falla.

Análisis, diseño, implementación, pruebas

Estas actividades se repiten en cada etapa de desarrollo.

Para el lunes actuar como analista, arquitecto, product owner.

Traer una especie de enunciado completo, extenso y detallado, remarcar qué se puede hacer en este semestre.

- Traer descripción general del problema
- Plantear una propuesta como por ejemplo 10 o 15 módulos.

No mostrar como un aporte de CRUD de manera explícita sino las transacciones qué se han especificado.

Agregar IA en el proyecto, esto sería un buen aspecto de innovación.

En la propuesta debe incluir el uso de IA, no usar chatboot. No es necesario utilizar propios modelos sino modelos públicos como machine-learning, la IA debe ser la apropiada para nuestro proyecto.

Cuando se usa IA se necesitará un gran volumen de datos, poblar base de datos con datos consistentes mediante herramientas como dataset.

Los resultados se deben reflejar en:

- Determina estado.
- Proyecciones de lo que puede pasar.
- Mostrar datos en formularios o mejor en Dashboard.

Clase 27 de agosto

Trabajar sobre el perfil de proyecto par esto elaborar un documento con los siguientes puntos:

1. Carátula.- con el título "perfil de proyecto"

2. Tabla de contenido o índice.- no puede ser manual, debe ser navegable en el PDF, con al menos 3 niveles, título, subtítulo

3. Introducción.- captar la atención del cliente que quiera adquirir el producto, tomar en cuenta que de una introducción general del documento, motivar o tentado a leer al menos los detalles mas importantes del documento, como mínimo una página y máximo 2 páginas.

4. Antecedentes.- todo lo hecho con respeto al proyecto en el ámbito histórico, como un resumen de la primera presentación de los antecedentes, historia del proyecto desde 3 perspectivas: revisión de lo escrito como en los libros estudiados para la primera documentación con referencias los libros.

Descripción de las 3 aplicaciones relacionadas al tema que se hayan visto en los antecedentes.

Al menos 3 casos de estudio como potenciales clientes para el software que se va a desarrollar

5. Descripción del problema.-posiblemente 50 páginas, se debe describir el problema o la problemática del tema, se deben realizar entrevista, se puede tomar como referencia las políticas de negocio, con UML diagrama de actividades organizado por capaz y el modelo de negocio debe en representar esta descripción del problema.

6. Objetivos.- 2 tipos de objetivos general y específico, utilizar textos guía relacionada con metodología de investigación.

No sirven objetos ambiguos, la primera palabra del objeto debe ser un verbo, se tiene que tener la certeza de que se va a desarrollar, el objetivo debe reflejar lo que se va a desarrollar; implementar no es lo mismo que desarrollar, dentro de ámbito de la carrera, implementar es meter código.

7. Alcance.- colocar todo lo que se considere que el software tendrá, se debe describir en esta parte. Colocar solo las funcionalidades principales, en su ficha funcionalidades un desglose de lo que se pretende

8. Elementos de un sistema de información basado en computadoras.-

*se avanzara en un clase

9. Tegnologia.- se tendrá 3 capas

• Estrategia de desarrollo de software: podría tener elementos como procesos de desarrollo SCRUM, descrito en una clase, a nivel de modelos se usará UML 2.5 o superior.

• Fundamentación teoría: una descripción según los objetivos de los libros

• Herramientas: herramientas de gestión de proyecto como trello, se debe tener captura en la documentación; github para gestionar el código; herramientas CASE como architec, base de datos, despliegue, herramientas para el desarrollo de aplicaciones móvil, para realizar pruebas, para generar dato sintético los cuales son generado por un robot para poblar la bases de datos.

10. Costos.- no colocar costos de desarrollo, sino costos para poner en producción, desde un enfoque SAAS (software como servicio de suscripción), On Premise en el cual los datos solo son visibles para el usuario, lo que se ve de aquí son licencias.

Se hará este en una tablas comparativa.

Sin costos inventados sino cotizaciones.

11. Beneficios.- este punto debe animar al cliente a invertir.

Todas aquellas características que el cliente va a mejorar, cosa que el cliente vea justificada la inversión, no colocar nada líricamente sino demostrables, como ser el cliente sin el sistema y el cliente con el sistema, detallar por ejemplo el ahorro en tiempo, reducir costos, reducir esfuerzos

12. Bibliográfica.- colocar la lista de todas las fuentes de información consultadas para la realización de este trabajo, fuentes relacionadas con el tema, nombres y cargos de las personas entrevistadas

13. Anexos.- obligatorio las entrevistas escrituradas, como encuestas online, estudios de mercado.

Toda la información recopilada de las entrevistas, como formatos de documentos. Curricular de cada uno

Esta sería la documentación relacionada con el proyecto de la materia.

Normas APA para la elaboración de la documentación.

Para el 3 de septiembre la entrega

Clase 1 de septiembre

Elementos de un sistema de gestión basado en computadoras. -

Un sistema es conjunto de elementos que interactúan entre sí para un fin en específico.

Sus elementos serían:

1. Hardware. Brinda soporte físico al sistema. Sin el software no podría ejecutarse ni los datos podrían almacenarse.

2. Software. Es el conjunto de programas, aplicaciones e instrucciones que le dicen al hardware qué hacer, como el cerebro del sistema

3. Datos. Son la materia prima del sistema. Es la información que se captura, almacena, procesa y distribuye. Siendo el activo más valioso.

Vienen a la información básica a procesar.

4. Procesos. Son las políticas, reglas, instrucciones y métodos establecidos que definen cómo se debe usar el sistema. Son el manual de operaciones que garantiza su correcto funcionamiento.

Ejemplo: cuando se desarrolla el sistema se tiene que entender como trabaja la empresa, agregando mejoras como el hardware y el software para su posterior implantación mediante ingeniería de procesos, como entrevistas, encuestas, modelos de negocios, modelo de dominio.

5. Gente. Son los usuarios y el personal especializado que interactúan con el sistema. Son el elemento más crítico, ya que un sistema sin personas que lo usen correctamente es inútil.

Un actor es un actor externo que interactúa con el sistema. No referirse al actor como usuario, sino referirse con sus roles. Da el estímulo necesario para que el sistema funcione.

6. Documentación. Es un respaldo de su funcionamiento operativo, que refleja el proceso de desarrollo, el proceso de puesta en marcha y el uso del sistema.

Estos elementos se deben aplicar en el perfil de proyecto, no desde el punto de desarrollo, tomarlo como si estuviera en producción, sino de los requerimientos que el usuario debe cumplir para poner en marcha el sistema, como base de datos, servidor, etc., de manera local qué se requiere, qué sistema operativo.

A nivel de dato lo que el usuario tiene, este sistema trabaja con información estructurada, ingresados a través de formulario, en cada transacción el usuario debe ingresar los datos requeridos de manera detallada.

En procesos las distintas políticas de negocio que se debe de implementar en el sistema.

Gente, lo que se requiera, como actor: administrativo y demás roles como jefe de compra, etc.

Documentación que respalde el desarrollo de software, documentación de puesta en marcha y documentación de como el usuario debe utilizar, por ejemplo tener un chat boot que brinda información al usuario que debe usar.

Clase 3 de septiembre

Primer examen parcial

Desarrollar aplicaciones de software,

En un condominio existe copropietarios que son los que compran las cajas y los vivientes que no siempre son los propietarios.

Las casas están clasificadas por categoría, se debe saber cuantos viven allí y si tienen mascota, tienen número de casa, hay áreas comunes como piscinas, parques,.

Hay eventos en los cuales los residentes pagan por el uso de áreas comunes como por ejemplo la churrasquería, las solicitudes con tiempo para no chocar con otros inquilinos, se dispone de un parqueo público para los visitantes, la gente por lo general debe ser registrada si ingresa a pie o en vehículo, se tiene que saber si el visitante es un viviente o no.

Se dan situaciones como por ejemplo uno indebido de áreas comunes como perros asiendo sus necesidades en áreas comunes, vehículos mal parqueados, gente merodeando.

Existen pagos por las expensas que lo esperado es que se debe definir cuánto en lo que se paga, el pago puede ser fijo o por la categoría de la casa, además del precio de la reserva de un área común, uso de equipos como gimnasio, no debe entrar gente a beber a la piscina o gente extraña, la gente puede pagar al empezar cada mes por efecto, pagos por tarjeta, hay personas que pagan de forma automática, hay multas por retraso en los pagos, no recoger basura, no podar el su jardín, se puede restringir el uso de áreas comunes como sanciones.

Se tiene personal que cumple funciones para el condominio, este software debería reducir esfuerzo, costos y tiempo.

Uso de QR para autenticar la visita de extraños por parte de viviendas.

Sistema para ofertar viviendas.

Pagó automático de las despensas mediante pasarelas de pago.

Se tendrá una versión móvil y una versión web la cual será para uso de residentes o vigilantes.

Se tendrá notificaciones le listas negras, comunicados para los residentes, se debe tener confirmación de lectura para seguimiento por parte la administración, se debe tener métodos de pago para expensas, etc.

Se debe generar reportes de las novedades.

Aplicar IA en el software, reconocimiento fácil para el ingreso, reconocimiento de placas, reconocimiento de imágenes para el ingreso y salida de vehículos, si no está registrado el guardia conforma el ingreso del visitante y lo registra.

Se usaría visión inteligente para reconocimiento de acciones inapropiadas como perros sueltos, vehículos mal estacionados, determinar el comportamiento de pago para tratos preferenciales por parte del guardia.

Sugerencia: proponer alguna novedad.

Entrega el 1 de octubre

El backend python con django, frontend react, móvil postgree, base de datos postgree, a nivel de IA cualquier de uso publico, como el de Amazon, Microsoft.

Se puede usar IA para el desarrollo de aplicaciones, el estudiante es responsable de lo que se haga, en la presentación no se usará IA, para que el estudiante conozcas lo que se hizo.

Proceso unificado, UML 2.5, uso de Github, uso de herramientas como trello,

Clase 5 de septiembre

El despliegue será en: AWS, Google cloud y Azure.

La aplicación móvil debe estar alojada en drive.

La documentación debe estar organizada en capítulos.

Se debe describir la fundamentación teoría.

Un capítulo debe hablar sobre los antecedentes con sus respectivas características, hablar de algunas aplicaciones similares con respecto a la administración de condominios.

Un capítulo sobre la descripción del problema.

Un capítulo que hable sobre las aplicaciones de la IA qué se han implementado como machine-learning, de modelo estadístico con sus respectivas fundamentaciones como por ejemplo random forest, un tema esencial puede ser visión artificial, documentar las herramientas de IA que se puede ocupar respecto a la visión artificial, reconocimiento facial, si se ocupa OCR para detectar las placas.

Hablar sobre los lenguajes de desarrollo como python, flutter, etc.

Ocupar sobre el proceso unificado de desarrollo de software, se debe documentar en un capítulo este proceso.

Otro capítulo sobre el uso de UML para generar modelos.

Otro capítulo sobre el proceso de desarrollo de software.

Estudiar las diferencias entre scrum y scrum esta es pregunta cero.

Inicio, desarrollo, transición y prueba son las fases de ciclo.

Y 5 flujos de trabajo, como captura de requisitos.

Documentar tal como se aprendió en proceso unificado de desarrollo de software, se tiene que tener un capítulo que hable de UML 2.5 o superior ver la documentación oficial.

Un capítulo que hable, usar una IA para ayudar en el uso, como un asistente inteligente qué siga el proceso de uso.

Adjuntar bibliográfica, anexos.

El formato debe ser APA.

Documento en 2 formatos: digital o impreso, el pdf debe ser navegable.

Se debe entregar el 1 de octubre máximo a las 8 de la mañana.

Solo un documento por grupo.

Se debe prever como demostrar las funcionalidades.

Todo debe ser en línea.

En la documentación agregar QR para acceder a los repositorios, al recurso para aprender a usar el software, para acceder al software, debe estar disponible hasta el último día de entrega de notas.

Clase 13 de septiembre

Próximo viernes presentación del primer sprint.

Una pregunta de examen puede ser las diferencias entre scrum y proceso unificado de desarrollo de software.

El ciclo de vida del desarrollo de software mediante scrum es iterativo es incremental.

En la planificación se considera que el software está naciendo con 3 fases subsecuentes: sprint backlog, daily scrum, fase incremental y la revisión del sprint.

Mediante ingeniería de requisitos mediante el cliente conoce la problemática que se tiene que resolver mediante el software

Pregunta de examen los aspectos básicos de todo el pdf al detalle como el equipo y las fases.

La coordinación se realizará a diario para incrementar la fase de desarrollo en la que se sacan conclusiones para implementar en el siguiente sprint.

En el sprint backlog estarán los casos de usos planificados para ser desarrollados durante y la realización del presente sprint.

Las 3 c se refiere a la tarjeta de caso de usuario, conversación y confirmación.

En el punto c se especifica el PHU generalmente se usan números de la secuencia de Fibonacci, mientras más alto mayor la urgencia.

Clase 15 de septiembre

Se debe aclarar el registro de asistencia,,

Tomar el caso de estudio entre otros, trabajadores qué o trabajan directamente en oficina por ejemplo vinculado con el GPS.

Clase 26 de septiembre

En 2 semanas presentación del segundo sprint.

Funcionalidad obligatoria para proyectos, a eso se lo denomina características generales qué deben tener todos los proyectos.

1. Debe ser una aplicación o sistema basado en una solución universal para el mercado que se está apuntando, el proyecto no está dirigido a una empresa en particular sino para cualquier entidad que cubra cualquier ámbito sobre el cual está siendo desarrollado, no solo debe funcionar para un caso en específico, eso implica un enfoque basado en SAAS o software como servicio, en el proyecto se debe desarrollar el software enfocado en que el usuario use el software basado en licencia en las instalaciones del cliente, tiene como ventaja que los usuarios o clientes no tienen que

invertir mucho para usar, con el enfoque pagar por lo que se usa. Los posibles clientes tienen que saber los planes que se van a ofrecer qué varían de acuerdo a los planes disponibles y método de pago como plataforma de pago como stripe.

Se crean cuantas de usuario que son enviadas por correo al usuario, como desarrollador no se tiene que programar nada, todo esto se define por elección del usuario.

Se tiene que tener para el usuario un panel de personalización por ejemplo subir un logo de la empresa.

Aspectos técnicos desde un punto de diseño. - la base de datos se debe tener tenant, una mase de datos para múltiples usuarios, por ejemplo una base de datos es como un edificio y un inquilino ocupó unas base de datos, evitando el acceso de un cliente a la cuenta de otro usuario en la base de datos.

El mutitenad es el uso de múltiples usuarios en una mismas base de datos, es más proponenzo a que se acceda a datos de diferentes clientes o acceso cruzado, se requiere el uso de índice compuesto, un dato tendría un ID para diferenciar a que cliente pertenece

2. Un sistema multiusuario. - enfocado para que muchos usuarios utilizan el sistema, en el usuario se asume que no se sabrá que usuario ni cuanto usuarios utilizaran el sistema, solo se contemplara 3 categorías de usuario superadmin solo para los desarrolladores, admin para cada cliente, la categoría será libre, el admin asigna roles para sus usuarios, el administrador lo hará a través de un panel de administración.

El admin tendrá un catálogo de funciones y datos que tendrá acceso su roles.

Sistema de crud de compra y venta de productos:

Código de producto:

Stock:

Costo:

Precio:

Este formulario tendrá acceso por parte de comprador y encargado de almacén.

Al encargado de almacén le doy acceso crud para productos, al vender ver los productos, no podría modificar precio de los productos solo verlo.

No usar el mismo formulario modificado como un copiar y modificar, sino asignar segmentos para los roles, por ejemplo visible pero no editable, acceder por roles asignando esos privilegios por parte del administrador cliente.

El cliente debe creer que el cliente es el único que sabe su contraseña, el desarrollador no debe asignar contraseña nueva al cliente, esta es una política de negocio de todo desarrollador.

3. El sistema debe estar preparado para no ser posible su uso para realizar fraudes, por ejemplo un contador usando el sistema para uso contable, la administración sospecha de fraude, se sospecha de el, se pide al contador lo estados de cuenta, el contador genere un balance general trucho con transacciones fraudulentas, y borra toda la bitácora, un nuevo contador revisa todo nuevo y encuentra el fraude, la responsabilidad será del desarrollador, la solución sería un log inviolable.

Atributos para archivo log y donde guardar.

El log tendría como atributos la fecha y hora, id del usuario, IP del usuario, fecha y hora real obtenido del servidor.

Guardado

Solo debe ser accesible desde el mismo software, no debe ser posible acceder desde otro medio.

4. Reportes la relación puede generar muchos reportes o mucha información, el sistema debe venir con reportes predeterminados pero configurables y personalizados, con pantalla previa por ejemplo, ventas por fechas, por clientes por periodo, no se tiene que mostrar todo como si se tratara de visualizar las tablas de la base de datos, deben ser exportables a pdf, exel o por correo, algunos vienen con planillas HTML para exportar a la web. Se debe generar permitir filtrar por ejemplo todos los productos organizados por categorías, por rango de precios, productos seleccionados, usando consultas por ejemplo, los reportes se deben crear de manera dinámica, en vez que el usuario escriba la consulta SQL, el usuario debe realizar esta consulta de manera intuitiva mediante un menú conocido como QBE.

Se debe poder generar consultas mediante prompts de audio.
Para el sistema debe tener acceso a la base de datos.

Fecha de entrega 8 de octubre

Clase 29 de septiembre

Cliente por suscripción

Asignación por roles en formulario, una bitácora,
Buscar un mecanismo de seguridad para la base de datos,
Política de qué y cuándo recuperar los datos.

Generación de reportes.

Generar reportes dinámicos.

IA con prompts

Uso de tecnología apropiada a la aplicación:

1. Stack tecnológico. - herramientas para el desarrollo de la aplicación. Fundamentar el uso, debe ser responsive, multiplataforma, se debe ejecutar en Android, en la aplicación móvil
2. Uso de hardware apropiado,
3. Despliegue.

Poder realizar backup de manera periódica

Redactar un manual de usuario, una IA, un chatbot para guiar al usuario.

Clase 15 de octubre

Segundo examen parcial.

Desarrollar un sistema de gestión empresarial, de ventas y compras con Ecommerce que involucre el uso de IA, un sistema de ventas para una boutique con funcionalidades para registrar los productos por categoría, características, etc.

Con funcionalidad de venta en efectivo o con venta en línea. Se debe registrar los productos, trabajadores y clientes.

Generar reportes dinámicos de ventas, por categoría, menos vendido, más vendido, etc. Se debe generar reportes a través de una IA, por texto o audio, generar reportes por ejemplo en rango de fecha, generar reportes de cliente, etc. en un pdf o un excel.

Osea un clásico un MVP, para la gestión de una boutique.

Una IA tiene la capacidad de aprender, de predecir el futuro, por tanto se debe generar pronósticos de ventas por ejemplo, estimación de ventas futuras de un determinado producto.

Para esta funcionalidad de usará para empezar machine-learning, correspondiente a la no supervisada, usar random forest para hacer predicciones el cual será entrenado con datos, se entregan los datos y el resultado para que aprenda, no requiere enormes cantidades de datos, se usará un dataset como ser públicos que se puede personalizar para usarlo, usar datos reales, pero el más recomendable será usar datos sintéticos creados de manera artificial, el 80 % será para entrenar al modelo y el 20 % será para pruebas.

Aplicación web y móvil, la versión móvil y ocuparlo para potenciar el proyecto, algo novedoso mi ejemplo un probador de ropa.

Stack tecnológico:

Todo el backend en python, el más recomendable es Spring Boot, para machine-learning se debe usar python, todo puede estar en python pero las funcionalidades de machine-learning en python.

Usar microservicios dará puntos extras.

El frontend en react en web, flutter para móvil.

Base de datos postgresql

PUDS en base a Rumbau, Jackobson, c4 y UML 2.5

Desplegado en aws, azure o Google cloud.

Fecha de entrega: 12 de noviembre,