



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLOGICO
NACIONAL DE MEXICO

Instituto Tecnológico de Pabellón de Arteaga
Departamento de Ciencias Básicas

PROYECTO DE TITULACIÓN

[SYSTEMGOLD]

PARA OBTENER EL TÍTULO DE INGENIERO EN TECNOLOGIAS DE LA INFORMACION Y COMUNICACION

PRESENTA:

LUIS FERNANDO DURON CASTAÑEDA

ASESOR:

JOSE NORBERTO MONDRAGON REYES

Noviembre



Ricardo
2022 Flores
Año de Magón
PRECURSOR DE LA REVOLUCION MEXICANA

2. Agradecimientos.

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo.

Especial reconocimiento merece el interés mostrado por mi trabajo y las sugerencias recibidas de la profesor y amigo MC. Renato Eduardo Reyes González, con la que me encuentro en deuda por el ánimo infundido y la confianza en mí depositada. También me gustaría agradecer la ayuda recibida del profesor Abel Solís Alemán.

Quisiera hacer extensiva mi gratitud a mis compañeros del de la ingeniería en tecnologías de la información y comunicación. Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia y amigos.

A todos ellos, muchas gracias.

3. Resumen.

El presente proyecto de residencias consiste en desarrollar e implementar un sistema para el almacenamiento de alumnos en la escuela Golden English dedicado al personal de dicha empresa, la principal característica de este sistema es la facilidad de guardar información de los alumnos que estén registrados en la empresa gracias a su amigable diseño, resolviendo así necesidades específicas de la empresa y ayudando a la difusión de su trabajo.

El presente proyecto de ingeniería en sistemas implica tres ramas. La primera de estas ramas es la Programación C# siendo esta la base principal del proyecto, se apoyará en la rama de Diseño con Frames y sustentará toda su información en la rama de Base de Datos. Para la realización de estas residencias es necesario apoyarse en una metodología. Apoyada en el lenguaje de modelamiento UML permite el desarrollo de todos los diagramas que servirán como base para la programación.

Las herramientas utilizadas en el desarrollo son herramientas Microsoft, para la programación se utilizó Microsoft Visual Studio .NET 2010 con Framework 4.0, para la integración se implementó en el motor de base de datos SQL Server 2008.

Índice

2. Agradecimientos.....	2
3. Resumen.....	3
4. Índice.....	¡Error! Marcador no definido.
Lista de Figuras.....	6
Lista de Tablas.....	6
5.- Introducción.....	7
6. Descripción de la empresa u organización y del puesto o área del trabajo del estudiante.....	8
7. Problemas a resolver, priorizándolos.....	9
8. Objetivos (General y Específicos).....	9
8.1 OBJETIVO GENERAL.....	9
8.2 OBJETIVOS ESPESIFICOS.....	9
9. Justificación.....	10
10. Marco Teórico (fundamentos teóricos).....	11
10.1 DEFINICIÓN.....	11
10.1.1 ANTECEDENTES.....	12
10.1.2 CARACTERÍSTICAS.....	14
10.1.3 TIPOS DE SISTEMAS.....	15
10.1.4 Base de Datos.....	18
10.2.0 Metodología.....	24
10.2.1 METODOLOGIA AUP.....	24
10.2.2 CICLO DE VIDA DEL PROCESO UNIFICADO ÁGIL.....	25
10.2.3 FASES.....	25
10.2.4 INCREMENTO Y DESARROLLO DE AUP.....	27
10.2.5 PRINCIPIOS DE LA AUP.....	28
11. Procedimiento y descripción de las actividades realizadas.....	29
11.1.1 PROPÓSITO.....	29
11.1.2 ÁMBITO DEL SOFTWARE.....	29
11.1.3 ABREVIATURAS, ACRÓNIMOS Y DEFINICIONES.....	30
11.1.4 PERSPECTIVA DEL SOFTWARE.....	30
11.1.5 FUNCIONES DEL SOFTWARE.....	30

11.2.1 CARACTERÍSTICAS DE LOS USUARIOS.....	31
11.2.2 SUPOSICIONES INICIALES.....	31
11.2.3 PRUEBA DE CAJA BLANCA Y GRAFOS.....	35
Cronograma de actividades.....	38
RESULTADOS.....	39
12. Resultados.....	39
Objetivos.....	39
Resultados Obtenidos.....	39
La presente investigación se propuso originalmente, aunque de manera exploratoria.....	39
CONCLUSIONES.....	40
14. Conclusiones del Proyecto.....	40
15. Competencias desarrolladas y/o aplicadas.....	40
FUENTES DE INFORMACIÓN.....	42
16. Fuentes de información.....	42
(S.C.M, 2018).....	42
Bibliografía.....	42

Lista de Figuras

Figura.1. uso de metodología AUP. -----	17
Figura.2. ciclo de vida AUP. -----	19
Figura.3. Grafo Principal. -----	28
Figura.4. Relacion de entidades-----	20

Lista de Tablas

Requisito Funcional 1. -----	26
Requisito Funcional 2. -----	28
Tabla de resultados. -----	33

GENERALIDADES DEL PROYECTO

5.- Introducción

Los sistemas de información se originan en casi todas las áreas de una empresa y se relacionan con todos los problemas de la empresa. De ahí la definición de que un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

Para tener un buen sistema se deben tomar en cuenta dos aspectos básicos, desde el hardware necesario para que el sistema pueda operarse y el recurso humano que interactúa con el sistema de información.

En el desarrollo de los sistemas de información se puede observar que las aplicaciones son una herramienta y no un instrumento que debe tenerse para utilizar la tecnología de la información, lo que trae como consecuencia la mejora en el desempeño de la organización, e incluye beneficios tanto para sus empleados, clientes y personas que tengan que ver con ella.

Cada departamento de una empresa en cualquier sector requerirá de sistemas de información, desde oficinas gubernamentales, fábricas, o desde la contabilidad hasta la mercadotecnia, los sistemas de información son herramientas clave para el procesamiento de transacciones, toma de decisiones, resolución de problemas y las operaciones de toda organización.

Debido a que los sistemas de información se tienen en todas las empresas de una forma u otra, el desarrollo del presente trabajo se enfocará a lo que la mayoría de éstas tienen:

Recursos humanos.
Almacenar datos.

Existen más áreas de aplicación de los sistemas de información pero todo eso dependerá del tipo de empresa y el giro que tenga, para desarrollar incluso más sistemas de información que los tradicionales o comunes.

6. Descripción de la empresa u organización y del puesto o área del trabajo del estudiante.

Golden English es un corporativo nacional con 27 sucursales en la República Mexicana, dedicadas específicamente a la enseñanza del idioma inglés con Técnicas neurolingüísticas, las cuales son las más modernas que existen en la actualidad para el aprendizaje.

Conociendo las necesidades y las experiencias de nuestros suscriptores el compromiso es real, para que al término de 4 meses puedan hablar un inglés básico fluido y en 12 meses dominar este importante idioma.

Nuestro personal Académico además de ser certificado, tiene un alto sentido de permanencia, pertenencia y responsabilidad, comprometido con nuestra misión; hacer que nuestros suscriptores dominen este idioma

Puesto:

En esta organización el estudiante Luis Fernando Durón Castañeda ocupa el puesto en recepción donde se encuentra laborando y trabajando en su proyecto.

7. Problemas a resolver, priorizándolos.

Actualmente la escuela Golden English presenta un problema para el llenado de alumnos recién inscritos en dicha escuela.

- Llenado de alumnos inscritos.
- Consultas de alumnos inscritos
- Generar información confiable.

8. Objetivos (General y Específicos)

8.1 OBJETIVO GENERAL

Apoyar, automatizar y agilizar la elaboración de documentos para el registro de alumnos que egresen en esta institución “Golden English”.

8.2 OBJETIVOS ESPECÍFICOS

- Desarrollar un sistema para la modernización y eficiencia de los procesos de la institución.
- Generar información confiable y oportuna.
- Disminuir los tiempos de la elaboración de un registro de alumnos.

9. Justificación

Debido a los problemas en la gestión de almacenamiento de alumnos, se requiere de un sistema capaz de administrar adecuadamente los alumnos que puedan inscribirse en la institución

Es necesario modernizar la metodología de negocio en la empresa, para poder agilizar el tiempo y poder almacenar los datos de los alumnos inscritos en dicha institución, para estar al tanto con información actual, esto desencadenará en ahorros en tiempo y dinero.

Además, que con esto el alumno aprenderá y desarrollará habilidades de desarrollo en lenguajes de programación c# y base de datos SQL Server.

MARCO TEÓRICO

10. Marco Teórico (fundamentos teóricos).

10.1 DEFINICIÓN

El estudio de los sistemas de información es un campo multidisciplinario, por lo tanto, no existe alguna perspectiva o teoría que por sí sola predomine, por consiguiente, distintos autores nos dan a conocer distintas definiciones que a continuación se mencionaran:

“Es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común” (Senn, 1992, p.19)

“Es una disposición de componentes integrados entre si cuyo objetivo es satisfacer las necesidades de información de una organización” (Whitten, 2003, p.39)

Whitten (2003) menciona que un sistema de información es:

Es una disposición de personas, actividades, datos, redes y tecnología integrados entre sí con el propósito de apoyar, mejorar las operaciones cotidianas de una empresa, así como satisfacer las necesidades de información las necesidades de información para la resolución de problemas y la toma de decisiones por parte de los directivos de la empresa. (p.39)

“Los sistemas de información son desarrollados con propósitos diferentes dependiendo de las necesidades del negocio” (Kendall, 1997, p.2)

Laudon (1996) define un sistema de información como:

Un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución. Los sistemas de información pueden contener datos acerca de personas, lugares y cosas importantes dentro de la institución y el entorno que la rodea. (p.8)

10.1.1 ANTECEDENTES

Como nos podemos dar cuenta de los sistemas de información abarcan una amplia gama de situaciones que son necesarias para poder comprender lo básico, en la realización de un sistema de información. Para poder realizar esto debemos conocer un poco de su historia y de sus orígenes, para ampliar el conocimiento que se pretende conseguir.

Whitten (año) encontró lo siguiente:

Hay una leyenda que dice que el primer analista de sistemas apareció en escena hace unos 6,000 años durante la construcción de las pirámides de Egipto. Intento idear un modo mejor de construir un producto: una pirámide. Los analistas de sistemas de hoy en día también intentan hallar mejores vías para crear nuevos productos, en este caso sistemas de información. Como tributo a aquel primer analista de sistemas, haremos uso de la pirámide para ilustrar los conceptos y los bloques elementales de los sistemas de información. La pirámide consta de cuatro caras y una base cuadrada, cada una de las caras, más la base, representa diferentes bloques elementales de los sistemas de información y pone de relieve los conceptos y temas que es preciso considerar durante el desarrollo de un sistema de información. (p.38-39)

“Las aplicaciones de sistemas de información tienen su origen en casi todas las áreas de una empresa y están relacionadas con todos los problemas de la organización.” (Senn, 1992, p.60)

“La información en sí no se consideraba como un activo de importancia para la empresa. En la mayoría de las instituciones, la información era considerada como un subproducto y caro resultado de los negocios.” (Laudon, 1996, p.6)

Laudon (1996) menciona que:

Con el transcurso del tiempo, los sistemas han llegado a jugar un papel más importante en la vida de las instituciones. Los primeros sistemas implicaban grandes cambios técnicos relativamente fáciles de alcanzar. Posteriormente, los sistemas implicaban control y comportamiento administrativo. Finalmente, los sistemas influyeron en actividades funcionales centrales relativas a productos, mercados, proveedores y clientes. (p18)

Chiavenato (1992) nos dice que:

La teoría de sistemas se fundamenta en tres premisas básicas:

1. Los sistemas existen dentro de sistemas: cada sistema existe dentro de otro más grande.
2. Los sistemas son abiertos: es consecuencia del anterior. Cada sistema que se examine, excepto el menor o mayor, recibe y descarga algo en los otros sistemas, generalmente en los contiguos. Los sistemas abiertos se caracterizan por un proceso de cambio infinito con su entorno, que son los otros sistemas. Cuando el intercambio cesa, el sistema se desintegra, esto es, pierde sus fuentes de energía.
3. Las funciones de un sistema dependen de su estructura: para los sistemas biológicos y mecánicos esta afirmación es intuitiva. Los tejidos musculares, por ejemplo, se contraen porque están constituidos por una estructura celular que permite contracciones. (p. 43)

Los sistemas de información anteriormente generaban cambios técnicos que afectaban a pocas personas que estaban inmersas dentro de la empresa, a diferencia de los actuales ya que estos involucran a una mayor parte de la institución.

10.1.2 CARACTERÍSTICAS

Senn (1992) señala que:

Para alcanzar sus objetivos, los sistemas interactúan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de las fronteras de los sistemas. Los sistemas que interactúan con su medio ambiente (reciben entradas y producen salidas) se denominan sistemas abiertos. En contraste, aquellos que no interactúan con su medio ambiente se conocen como sistemas cerrados. Todos los sistemas actuales son abiertos. Es así como los sistemas cerrados existen solo como un concepto, aunque muy importante. (p.21)

“Todos los sistemas de información pueden describirse como soluciones institucionales y de administración a los retos del entorno.” (Laudon, 1996, p.12)

“La microcomputadora de escritorio representa la base de un sistema actual de información.” (Laudon, 1996, p.18)

Como podemos darnos cuenta hoy en día la computadora es una herramienta vital para los sistemas de información ya que en la mayoría de los casos estos se encuentran funcionando gracias a la computadora, para que de esta manera los sistemas de información se automaticen.

10.1.3 TIPOS DE SISTEMAS

Los sistemas de información se dividen en:

“SISTEMAS DE NIVEL OPERATIVO: Sistemas de información que hacen el seguimiento de las actividades y las transacciones elementales de la organización.” (Laudon, 1996, p.15)

“SISTEMAS DE NIVEL DE CONOCIMIENTOS: Sistemas de información en los que se apoyan los trabajadores del conocimiento y de la información en una institución.” (Laudon, 1996, p.15)

“SISTEMAS DE NIVEL GERENCIAL: Son sistemas de información en los que se apoya el seguimiento, control y toma de decisiones y las actividades administrativas de los administradores de nivel medio.” (Laudon, 1996, p.15)

“SISTEMA DE NIVEL ESTRATÉGICO: Sistemas de información que apoyan a las actividades de planeación a largo plazo de los niveles de dirección de la institución.” (Laudon, 1996, p.15)

Kendall (1997) divide los sistemas de información en:

Sistemas de procesamiento de transacciones (TPS): Son sistemas de información computarizados desarrollados para procesar gran cantidad de datos para transacciones rutinarias de los negocios, tales como nomina e inventario. Los TPS eliminan el tedio de las transacciones operacionales necesarias y reducen el tiempo que alguna vez se requirió para ejecutarlas manualmente, aunque las personas deben alimentar datos a los sistemas computarizados.

Sistemas de automatización de oficina y sistemas de manejo de conocimiento: Al nivel de conocimiento de la organización hay dos clases de sistemas. Los sistemas

automatizados de oficina (OAS) que dan soporte a los trabajadores de datos, usan la información para analizarla y transformar datos. Los aspectos familiares incluyen procesamiento de palabras, hojas de cálculo, editor de publicaciones, comunicación mediante correo de voz, correo electrónico y videoconferencias. Los sistemas de manejo de conocimiento (KWS) dan soporte a los trabajadores profesionales, tales como científicos, ingenieros y doctores, les ayudan a crear un nuevo conocimiento que contribuya a la organización o a toda la sociedad.

Sistemas de información gerencial (MIS): Estos sistemas no reemplazan a los sistemas de procesamiento de transacciones. Los MIS son sistemas de información computarizada que trabajan debido a la interacción resuelta entre gentes y computadoras. Requieren que las gentes, el software y el hardware trabajen al unísono. Los sistemas de información gerencial producen información que es usada en la toma de decisiones.

·Sistemas de apoyo a decisiones (DSS): Una clase de más alto nivel en los sistemas de información computarizada son los sistemas de apoyo a decisiones (DSS). Es similar al sistema de información gerencial tradicional en que ambos dependen de una base de datos como fuente. Un sistema de apoyo a decisiones se aparta del sistema de información gerencial tradicional en que enfatiza el apoyo a la toma de decisiones en todas sus fases, estos sistemas están más hechos a la medida de la persona o grupo que los usa.

Sistemas expertos e inteligencia artificial (AI): Los sistemas expertos usan los enfoques del razonamiento de la AI para resolver los problemas que les plantean los usuarios de negocios. Los sistemas expertos son un caso muy especial de un sistema de información, cuyo uso ha sido factible para los negocios a partir de la reciente y amplia disponibilidad de hardware y software tal como las microcomputadoras y sistemas expertos. Un sistema experto, también llamado sistema basado en conocimiento captura en forma afectiva y usa el conocimiento de un experto para resolver un problema particular experimentado en una organización.

Arechiga (1983) señala que:

Dentro de los sistemas de información, se encuentran básicamente cuatro diferentes tipos:

SISTEMA MANUAL: En estos sistemas los datos son registrados manualmente mediante el uso del lápiz o pluma sobre documentos. Usualmente, estos documentos son transferidos en forma manual desde un lugar a otro; pudiendo almacenarse temporal y permanentemente en: archiveros. Para manejar archivos se establecen procedimientos manuales que permiten localizar eficientemente los documentos requeridos en un instante dado.

SISTEMA MECÁNICO: En este tipo de sistemas, la recopilación de datos se logra mediante el uso de dispositivos mecánicos, tales como: máquinas de escribir, cajas registradoras, impresoras de cheques. Aquí los documentos son almacenados de la misma manera que los sistemas manuales, pero los datos contenidos en ellos estarán impresos o mecanografiados en lugar de estar escritos a mano.

SISTEMA ELECTROMECAÁNICO: En sistemas de este tipo se emplea una codificación diferente a la escritura normal para manejar información, es decir, la información se simboliza mediante marcas sensibles o caracteres ópticos o magnéticos.

SISTEMA ELECTRÓNICO: Para llevar a cabo las operaciones tales como clasificar, reproducir, calcular o tabular en los sistemas manuales, mecánicos y electromecánicos; se requiere del empleo de varias personas o maquinas.

10.1.4 Base de Datos

10.1.4.1 Conceptos y definiciones

Un conjunto de información relacionada entre sí, almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos que están estructurados y organizados independientemente de su utilización, y que es utilizada para que el usuario con necesidad de información la consulte en tiempo real.

Ventajas de Base de Datos.

1. Independencia de datos y tratamiento. Cambio en datos no implica cambio en programas y viceversa (Menor coste de mantenimiento).
2. Coherencia de resultados. Reduce redundancia.
3. Mejora en la disponibilidad de datos No hay dueño de datos (Pero no son públicos), guardamos descripción.
4. Cumplimiento de ciertas normas. Restricciones de seguridad, accesos (Usuarios a datos), operaciones (Operaciones sobre datos).
5. Otras ventajas: Más eficiente gestión de almacenamiento.

Tipos de Base de Datos.

Por variabilidad de los datos almacenados

- ✓ Estáticos. Solo lectura, son históricos dinámicos.
- ✓ Se modifican con el tiempo.

Por Contenido.

- ✓ Bibliográficos. Información sobre la fuente principal texto completo.
- ✓ Contenidos

Modelo de Administración de Datos.

- ✓ **Jerárquicas:** Forma de árbol invertido, puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos, existe redundancia.
- ✓ **Red:** Este modelo permite la representación de muchos a muchos, de tal forma que cualquier registro dentro de la base de datos puede tener varias ocurrencias superiores a él. El modelo de red evita redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector.

- ✓ **Relacional:** Este modelo se está empleando con más frecuencia en la práctica, debido a las ventajas que ofrece sobre los dos modelos anteriores, entre ellas, el rápido entendimiento por parte de usuarios que no tienen conocimientos profundos sobre Sistemas de Bases de Datos.
- ✓ **Multidimensional:** Similar a la relacional, esta almacena estructura de base de datos
- ✓ **Orientada a objetos:** Relaciona el estado y comportamiento de datos
- ✓ **Documentales:** Indexación a texto completo
- ✓ **Distribuidas:** Almacenadas en varias computadoras
- ✓ **Deductivas:** A través de inferencias (aplicadas al entorno científico)

Nos enfocaremos en la forma relacional que es la más utilizada.

Los DBMS (Sistemas Administradores de Bases de Datos)

El DBMS: es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos, son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, está compuesto por:

- ✓ DDL (Data Definition language): Lenguaje de Definición de Datos. Por medio de este el DBMS identifica las descripciones de los elementos de los esquemas y almacena la descripción del esquema en el catálogo del DBMS. Por medio de este el DBMS especifica el esquema conceptual e interno (Base de datos Almacenada).
- ✓ DML (Data Manipulation language): Lenguaje de Manipulación de Datos. Permite la manipulación de las operaciones de Inserción, Eliminación y Modificación.
- ✓ SQL: Lenguaje de Consulta.

Ejemplos: Oracle, Access, Informix, SQL Server. La capacidad de los DBMS se define por su motor, es decir la cantidad de datos que puede ser capaz de manejar.

Objetivos.

- ✓ Independencia física y lógica de los datos. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- ✓ Redundancia mínima. En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida
- ✓ Integridad de los datos. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados.
- ✓ Acceso concurrente por parte de múltiples usuarios. debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
- ✓ Consultas completas optimizadas. Es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.
- ✓ Seguridad de acceso y auditoría. Deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada.
- ✓ Respaldo y recuperación. Deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- ✓ Acceso a través de lenguaje estándar. Pueda ser accesible con lenguaje común.

Propósito

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos.

Identificación de Entidades.

Alguna cosa acerca de la cual almacenamos datos. Una persona, lugar, cosa o concepto que tiene características de interés para la empresa.

Relación de Entidades.

Identificar que entidades se involucran en los procesos, y para la práctica se entrelazan con una palabra que identifica el proceso y flechas de dirección. Ejemplo:

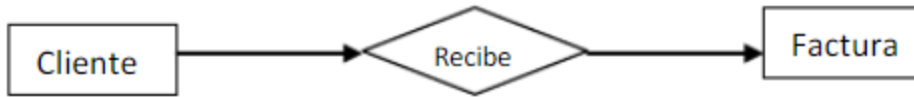


Figura.4. Relación de entidades

Identificación de tablas

Es generar la tabla para cada entidad definiendo los campos o registros que sean necesarios para cada uno de ellos.

Identificación de claves o Keys.

Una **clave primaria** es aquella columna (pueden ser también dos columnas o más) que identifica únicamente a esa fila. La clave primaria es un identificador que va a ser único para cada fila. En una tabla puede que tengamos más de una clave, en tal caso se puede escoger una para ser la clave primaria, las demás claves son las **Claves candidatas**. Además, es la posible clave primaria. Una **clave foránea** es aquella columna que existiendo como dependiente en una tabla, es a su vez clave primaria en otra tabla. Una **clave alternativa** es aquella clave candidata que no ha sido seleccionada como clave primaria, pero que también puede identificar de forma única a una fila dentro de una tabla.

Ejemplo: Si en una tabla clientes definimos el número de documento (id cliente) como clave primaria, el número de seguro social de ese cliente podría ser una clave alternativa. En este caso no se usó como clave primaria porque es posible que no se conozca ese dato en todos los clientes. Una **clave compuesta** es una clave que está compuesta por más de una columna.

- ✓ **Pk** = clave principal
- ✓ **Fk** = clave foránea

- ✓ **Nn** = no nulo
- ✓ **Uk** = única
- ✓ **Ck** = clave de chequeo

Tipo de datos, y longitud.

Esto define la clase de información que va a ser almacenada en cada campo. Puede definirse de varios tipos como texto, numérico, monetario, fecha, etc. También debemos definir la longitud del campo es decir que carga soportaría, por ejemplo definimos el campo **nombre** de tipo **texto** de longitud **30**, esto nos quiere decir que el campo nombre almacenara datos de texto hasta 50 caracteres.

Normalización.

El proceso de normalización de bases de datos

consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. Las bases de datos relacionales se normalizan para:

- ✓ **Evitar la redundancia de los datos.**
- ✓ **Evitar problemas de actualización de los datos en las tablas.**
- ✓ **Proteger la integridad de los datos.**

En el modelo relacional es frecuente llamar Tabla a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

- ✓ **Cada columna debe tener su nombre único.**
- ✓ **No puede haber dos filas iguales. No se permiten los duplicados.**

- ✓ **Todos los datos en una columna deben ser del mismo tipo.**

Seguiremos 4 Formas Normales:

1. Primera Forma Normal.

Busca campos multivaluados es decir que en un mismo campo guarda dos datos distintos, y los separa en campos distintos. Por ejemplo: El campo teléfono se puede dividir en Telefono1 y Teléfono2

2. Según Forma Normal.

Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros. relacionar estas tablas mediante una clave externa.

3. Tercera Forma Normal. Análisis de tabla para distinguir los campos que no están asociados directamente con la clave principal.

4. Forma Normal de Boyce- Codd (FNBC)

Se divide la tabla en dos: **CompVenta_maestro** y **CompVent_Detalle** Donde el criterio para separar la tabla es elegir los campos que se relación en entre si y a su vez sean multivaluados, esto se da mayormente en documentos tales como comprobantes de venta/factura/recibo, etc.

Relación de tablas.

Relacionamos los datos de tres formas diferentes:

- ✓ **1 a 1**
- ✓ **1 a varios**
- ✓ **Varios a varios**

- ✓ **1 a 1**

Al introducir un registro en un campo introducimos simultáneamente uno en otro campo (pero solo uno por vez)

- ✓ **1 a varios**

Permiten a un campo tener relacionado, y puede introducir en un campo varios registros.

- ✓ **Varios a varios**

Establece que varios campos con varios registros, pueden tener asociados varios campos también con varios registros.

10.2.0 Metodología

10.2.1 METODOLOGIA AUP

LA METODOLOGIA AUP es la versión simplificada del Proceso Unificado de Racional (RUP). Podemos describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas. Figura.1.

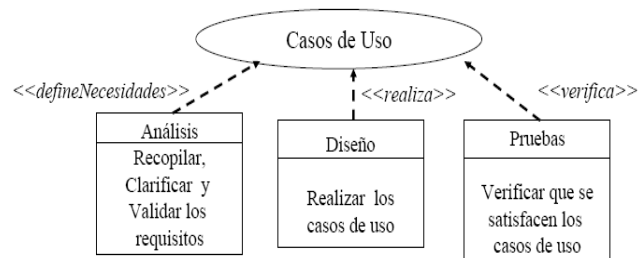


Figura.1. uso de metodología AUP.

LAS CARACTERISTICAS QUE COMPRENDE ESTA METODOLOGIA SON LAS SIGUIENTES

Características. -

Iterativo e Incremental.

- ✓ Descomposición de un proyecto grande en mini-proyectos
- ✓ Cada mini-proyecto es una iteración
- ✓ Las iteraciones deben estar controladas
- ✓ Cada iteración trata un conjunto de casos de uso

Ventajas del enfoque iterativo

- ✓ Detección temprana de riesgos
- ✓ Administración adecuada del cambio
- ✓ Mayor grado de reutilización
- ✓ Mayor experiencia para el grupo de desarrollo

Esta metodología se caracteriza por estar dirigida por casos de uso centro en la arquitectura y por ser ligero y ágil.

10.2.2 CICLO DE VIDA DEL PROCESO UNIFICADO ÁGIL.

Fases:

- * Origen
- * Elaboración
- * Construcción
- * Transición

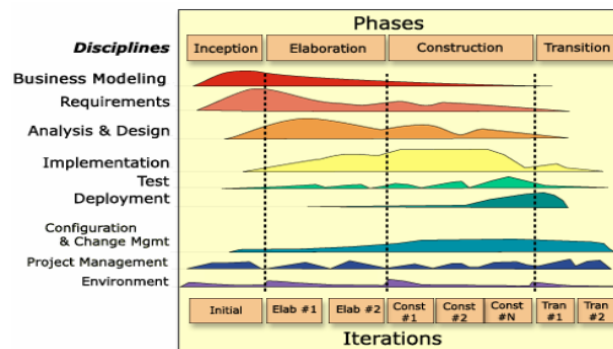


Figura.2. ciclo de vida AUP.

10.2.3 FASES

- ✓ Iniciación
- ✓ Elaboración

- ✓ Construcción
- ✓ Transición

AUP establece cuatro fases establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados: □ **Iniciación (Concepción):** El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.

- ✓ **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- ✓ **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- ✓ **Transición:** el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Las disciplinas se llevan a cabo de manera sistemática, a la definición de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar, y entregar el software de trabajo que responda a las necesidades de sus interlocutores.

Las disciplinas son:

- ✓ **Modelo.** El objetivo de esta disciplina es entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio.
- ✓ **Aplicación.** El objetivo de esta disciplina es transformar su modelo (s) en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.

- ✓ Prueba. El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.
- ✓ Despliegue. El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.
- ✓ Gestión de configuración. El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.
- ✓ Gestión de proyectos. El objetivo de esta disciplina es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc.), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.
- ✓ Entorno. El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario.

10.2.4 INCREMENTO Y DESARROLLO DE AUP.

Los equipos de AUP suelen ofrecer versiones de desarrollo al final de cada iteración en preproducción área (s). Una versión de desarrollo de una aplicación es algo que podrían ser liberados en la producción si se ponen a través de su pre-producción de garantía de calidad (QA), las pruebas y los procesos de despliegue. La primera producción de liberación a menudo toma más tiempo para entregar versiones posteriores. La primera

producción de liberación puede tomar doce meses para entregar la segunda versión de nueve meses, y luego otras liberaciones se entregan cada seis meses. Una de las primeras se centra en cuestiones de despliegue, no sólo permite evitar los problemas, sino que también permite tomar ventaja de sus experiencias durante el desarrollo. Por ejemplo, cuando despliegue un software en su área deberá tomar notas de lo que funciona y lo que no, toma nota de que puede servir como la columna vertebral de su instalación de scripts.

10.2.5 PRINCIPIOS DE LA AUP.

La AUP es ágil, porque está basada en los siguientes principios:

- ✓ El personal sabe lo que está haciendo. La gente no va a leer detallado el proceso de documentación, pero algunos quieren una orientación de alto nivel y / o formación de vez en cuando. La AUP producto proporciona enlaces a muchos de los detalles, si usted está interesado, pero no obliga a aquellos que no lo deseen.
- ✓ Simplicidad. Todo se describe concisamente utilizando un puñado de páginas, no miles de ellos.
- ✓ Agilidad. Ágil arriba el ajuste a los valores y principios de la alianza ágil.
- ✓ Centrarse en actividades de alto valor. La atención se centra en las actividades que se ve que son esenciales para el de desarrollo, no todas las actividades que suceden forman parte del proyecto.
- ✓ Herramienta de la independencia. Usted puede usar cualquier conjunto de herramientas que usted desea con el ágil UP. Lo aconsejable es utilizar las herramientas que son las más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.

- ✓ Adaptación de este producto para satisfacer sus propias necesidades. La AUP producto es de fácil acomodo común a través de cualquier herramienta de edición de HTML. No se necesita comprar una herramienta especial, o tomar un curso, para adaptar la AUP.

DESARROLLO

11. Procedimiento y descripción de las actividades realizadas.

11.1.1 PROPÓSITO

Este documento presenta y especifica el sistema para la empresa Golden English. Cuyos requisitos representan el comportamiento y especificaciones del software, la finalidad del desarrollo de la app es:

- ✓ Autenticación de la base de datos.
- ✓ Almacenar datos dentro de un sistema.
- ✓ Minimizar tiempos y costos.

11.1.2 ÁMBITO DEL SOFTWARE

Este documento está basado en los requisitos y especificaciones para el sistema. En el que se muestra los requisitos funcionales para los usuarios, sobre el uso, función y comunicación de las interfaces en la aplicación.

11.1.3 ABREVIATURAS, ACRÓNIMOS Y DEFINICIONES

- ✓ App: Abrev. de aplicación.
- ✓ Usuario: Es el que da comienzo con la operación. (Usuario/Cliente).
- ✓ Datos: Son los que ingresa el Usuario/Cliente.
- ✓ Tablas: Como las vistas, son de igual manera mostradas como las vistas.
- ✓ ID: identificador (clientes, productos).

11.1.4 PERSPECTIVA DEL SOFTWARE

El software funcionara sobre SQLServer en conexión con Visual Studio. Este software será capaz de mostrar datos dentro de una base de datos, con la seguridad de que los registros estarán bajo altos estándares de seguridad. Dicha información será guardada en el servidor.

11.1.5 FUNCIONES DEL SOFTWARE

El software realizara las operaciones y funciones diarias dentro del servidor de forma rápida, ordenada y segura, el procesamiento de los datos, así como actualizar, eliminar, insertar, mostrar y seleccionar registros en el servidor, tales como:

- ✓ Autenticación de Bases de datos SQL
- ✓ Conexiones a Visual Studio para ejecución de tablas.
- ✓ Registro de datos en el servidor.

11.2.1 CARACTERÍSTICAS DE LOS USUARIOS.

El software será usado por el usuario encargado de la base de datos y usuarios dependientes de él. El servidor permitirá operaciones como ejecución de tablas, consultas en la app. El usuario no necesita tener conocimientos profesionales de software en base de datos, pero si familiarizados con los procedimientos que le permitirán realizar las operaciones dentro del software. El usuario no necesita experiencia en el trabajo de estos sistemas.

11.2.2 SUPOSICIONES INICIALES

El usuario deberá actualizar datos después de cada acción, siempre y cuando sea necesario y así obtener un buen seguimiento.

REQUISITOS GENERALES.

- ✓ Descripción. El estado inicial del software se muestra por pantalla de inicio (Login). Desde esta ventana de inicio el usuario puede elegir entre las distintas opciones:

Mostrar las opciones del menú a elegir.

1) Login

2) Menú Inicio

3) Tabla 1 ... tabla N

- ✓ Entrada: El usuario registra su nombre y contraseña, entra en el menú inicio, selecciona la tabla a que desee entrar, registro y llenado de campos y elegir la consulta que se requiera para esta acción como: consultar, eliminar, insertar e incluso modificar datos en el servidor. Además de opción salir
- ✓ Procesamiento: Ejecución de la opción que el usuario necesite o pida realizar
- ✓ Salida: El servidor muestra un mensaje de que la acción se ha completado.

Requisito Funcional 1

- ✓ Descripción. El menú Login muestra la ventana donde se analizan dos campos donde va el nombre y contraseña del usuario existente.
- ✓ Entrada: Los campos deben ser llenados con nombre y contraseña.
- ✓ Procesamiento: El servidor manda llamar el usuario y comprueba si este existe.
- ✓ Salida: Muestra un mensaje si hay acceso o se denegó a la aplicación.

	Si	No	Tal vez
¿Están todos los formatos de los informes especificados?			X
¿Están especificados los interfaces con otros sistemas de software y hardware externos?			X
¿Están especificados todos los interfaces de comunicación, incluyendo handshaking, chequeo de errores y protocolos de comunicación?		X	
¿Se ha especificado, para todas aquellas operaciones que sea necesario, el tiempo esperado de respuesta desde el punto de vista del usuario?		X	
¿Se han especificado otras consideraciones de temporización, como en tiempo de procesamiento, transferencia de datos y rendimiento del sistema?		X	
¿Se ha especificado todas las tareas que el usuario desea realizar?	X		
¿Especifica cada tarea los datos que se usan en la tarea y los datos resultantes de la tarea?	X		
¿Se ha especificado el nivel de seguridad?			X
¿Se ha especificado la fiabilidad incluyendo las consecuencias de los fallos del software, información vital que ha de protegerse de fallos, detección de errores y recuperación de los mismos?		X	

¿Se ha especificado el equilibrio aceptable entre atributos de calidad contradictorios, por ejemplo, entre robustez y corrección?		X	
¿Se ha especificado el límite de memoria?		X	
¿Se ha especificado el límite de almacenamiento?			X
¿Se ha incluido la definición de éxito? ¿Y si fallo?	X		
¿Se ha especificado la mantenibilidad del sistema, incluyendo la habilidad para responder a?		X	

Requisito Funcional 2

- ✓ Descripción. El cliente selecciona la tabla la cual servirá para trabajar, muestra los campos que deberán ser llenados adecuadamente, y seleccionar la función (opción) con la que el cliente vaya a trabajar.
- ✓ Entrada: Dar clic sobre los campos que serán llenados.
- ✓ Procesamiento: Ejecución de la opción, el servidor manda llamar la tabla
- ✓ Salida: Muestra la tabla con los campos que se llenaron.

	Si	No	Tal vez
¿Están todos los formatos de los informes especificados?	X		
¿Están especificados los interfaces con otros sistemas de software y hardware externos?			X
¿Están especificados todos los interfaces de comunicación, incluyendo handshacking, chequeo de errores y protocolos de comunicación?		X	

¿se ha especificado, para todas aquellas operaciones que sea necesario, el tiempo esperado de respuesta desde el punto de vista del usuario?	X		
¿Se han especificado otras consideraciones de temporización, como en tiempo de procesamiento, transferencia de datos y rendimiento del sistema?	X		
¿Se ha especificados todas las tareas que el usuario desea realizar?	X		
¿Especifica cada tarea los datos que se usan en la tarea y los datos resultantes de la tarea?	X		
¿Se ha especificado el nivel de seguridad?		X	
¿Se ha especificado la fiabilidad incluyendo las consecuencias de los fallos del software, información vital que ha de protegerse de fallos, detección de errores y recuperación de los mismos?			
¿Se ha especificado el equilibrio aceptable entre atributos de calidad contradictorios, por ejemplo, entre robustez y corrección?			X
¿Se ha especificado el límite de memoria?		X	
¿Se ha especificado el límite de almacenamiento?		X	
¿Se ha incluido la definición de éxito? ¿y si fallo?		X	
¿Se ha especificado la mantenibilidad del sistema, incluyendo la habilidad para responder a cambios en el entorno operativo, interfaces con otro software, precisión, rendimiento y otras capacidades adicionales predichas			X

11.2.3 PRUEBA DE CAJA BLANCA Y GRAFOS

GRAFO PRINCIPAL

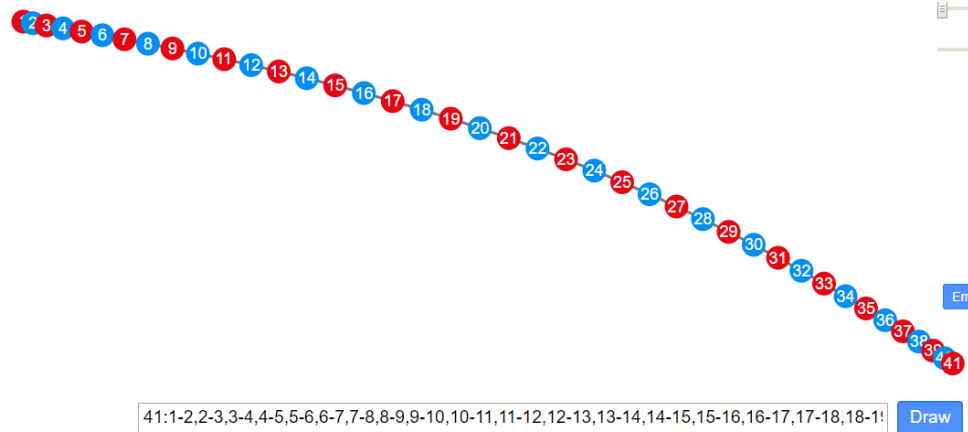


Figura 3. Grafo Principal

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//
using System.Data;
using System.Data.SqlClient;

namespace NativeEnglish
{
    class EscuelaDB
    {
        private SqlConnection conexion = new SqlConnection("Data Source = LAPTOP-
        6AKSSLST; Initial Catalog = NativeEnglish; Integrated Security = true" );
        private DataSet ds;

        public DataTable MostrarDatos() {

            conexion.Open();
            SqlCommand cmd = new SqlCommand("select * from Alumnos", conexion);
            SqlDataAdapter ad = new SqlDataAdapter(cmd);
            ds = new DataSet();
            ad.Fill(ds, "Tabla");
            conexion.Close();
            return ds.Tables["Tabla"];

        }
        public DataTable Buscar(string nombre)
        {

            conexion.Open();
            SqlCommand cmd = new SqlCommand(string.Format("select * from Alumnos where
            NombreAlumno like '{0}%", nombre), conexion);
            SqlDataAdapter ad = new SqlDataAdapter(cmd);
    
```

```
        ds = new DataSet();                                     16
        ad.Fill(ds, "Tabla");                                  17
        conexion.Close();                                     18
        return ds.Tables["Tabla"];                            19
    }

    public bool Insertar(string id, string NombreAlumno, string ApellidoPaterno,
string ApellidoMaterno, string Cantidad) {                  20

        conexion.Open();                                     21

        SqlCommand cmd = new SqlCommand(string.Format("insert into Alumnos values
({0}','{1}','{2}','{3}','{4}']", new string[] {id, NombreAlumno, ApellidoPaterno,
ApellidoMaterno, Cantidad }),conexion);                    22
        int filasafectadas = cmd.ExecuteNonQuery();          23
        conexion.Close();                                    24
        if (filasafectadas > 0) return true;                 26
        else return false;                                   27
    }
    public bool Eliminar(string id)                          28
    {

        conexion.Open();                                     29

        SqlCommand cmd = new SqlCommand(string.Format("delete from Alumnos where ID =
{0}", id), conexion);                                     30
        int filasafectadas = cmd.ExecuteNonQuery();          31
        conexion.Close();                                    32
        if (filasafectadas > 0) return true;                 33
        else return false;                                   34
    }
    public bool Actualizar (string id, string NombreAlumno, string ApellidoPaterno,
string ApellidoMaterno, string Cantidad)                  35
    {

        conexion.Open();                                     36

        SqlCommand cmd = new SqlCommand(string.Format("update Alumnos set
NombreAlumno = '{0}', ApellidoPaterno = '{1}', ApellidoMaterno = '{2}', Cantidad = '{3}'
where ID = {4}",new string[] {NombreAlumno, ApellidoMaterno, ApellidoPaterno, Cantidad,
id }), conexion);                                         37
        int filasafectadas = cmd.ExecuteNonQuery();          38
        conexion.Close();                                    39
        if (filasafectadas > 0) return true;                 40
        else return false;                                   41
    }
}
}
```

Botón Insertar (Código)

```
public bool Insertar(string id, string NombreAlumno, string ApellidoPaterno, string
ApellidoMaterno, string Cantidad) { 1
    conexion.Open(); 2
    SqlCommand cmd = new SqlCommand(string.Format("insert into Alumnos values
({0}','{1}','{2}','{3}','{4}']", new string[] {id, NombreAlumno, ApellidoPaterno,
ApellidoMaterno, Cantidad }),conexion); 3
    int filasafectadas = cmd.ExecuteNonQuery(); 4
    conexion.Close(); 5
    if (filasafectadas > 0) return true; 6
    else return false; 7
}
```

Botón Eliminar (Código)

```
public bool Eliminar(string id) 1
{
    conexion.Open(); 2
    SqlCommand cmd = new SqlCommand(string.Format("delete from Alumnos where ID =
{0}", id), conexion); 3
    int filasafectadas = cmd.ExecuteNonQuery(); 4
    conexion.Close(); 5
    if (filasafectadas > 0) return true; 6
    else return false; 7
}
```

Botón Actualizar (Código)

```
public bool Actualizar (string id, string NombreAlumno, string ApellidoPaterno,
string ApellidoMaterno, string Cantidad) 1
{
    conexion.Open(); 2
    SqlCommand cmd = new SqlCommand(string.Format("update Alumnos set
NombreAlumno = '{0}', ApellidoPaterno = '{1}', ApellidoMaterno = '{2}', Cantidad = '{3}'
where ID = {4}",new string[] {NombreAlumno, ApellidoMaterno, ApellidoPaterno, Cantidad,
id }), conexion); 3
```

```

int filasafectadas = cmd.ExecuteNonQuery();           4
conexion.Close();                                  5
if (filasafectadas > 0) return true;               6
else return false;                                 7
}

```

Cronograma de actividades

Actividades por mes	Enero	Febrero	Marzo	Abril	Mayo	Junio
Instalación de SQL Server 2008 y Visual Studio 2010						
Diseño del Login y Codificación						
Conexión a SQL Server						
Diseño de la interfaz del usuario						
Codificación de los botones (Eliminar, Insertar y Modificar)						
Manejo de prueba y errores						
Entrega de proyecto SystemGold						

RESULTADOS

12. Resultados

Objetivos	Resultados Obtenidos
<p>1. Desarrollar un sistema para la modernización y eficiencia de los procesos de la institución.</p> <p>2. Generar información confiable y oportuna.</p> <p>3. Disminuir los tiempos de la elaboración de un registro de alumnos.</p>	<p>La presente investigación se propuso originalmente, aunque de manera exploratoria.</p> <p>En este sentido, los resultados obtenidos por la investigación permiten señalar que los objetivos se han logrado por cuanto se ha sistematizado la posición relevante respecto al problema que es el registro de alumnos que pudiesen inscribirse en dicha empresa.</p> <p>Para el logro de estos objetivos principales, se han enunciado los enfoques sobre el problema de dicha empresa y su control a la hora de registrar alumnos dentro de la empresa. Además, se logró situar un perfecto orden de datos dentro del sistema SysteGold.</p> <p>Dentro del presente proyecto se encuentra en una de las empresas que cuentan con un sistema similar así dándole mayor potencial a dicha empresa.</p>

CONCLUSIONES

14. Conclusiones del Proyecto

Los sistemas de información son básicos y de apoyo total para la toma de decisiones de cualquier área de la empresa, son una herramienta básica para tener un mejor control de la información.

Aunque no se debe pasar por alto:

Los cambios tecnológicos que se mueven más rápido que los seres humanos o instituciones.

Se necesita el uso de tecnología para simplificar la comunicación y la coordinación.

Los sistemas pueden dar soporte a las ventas y compras de productos en muchos países.

Se tiene con ello también nuevas formas de hacer negocios.

Una cosa es usar la tecnología de la información para diseñar, producir, entregar y mantener nuevos productos y otra cosa es ganar dinero haciendo esto.

También se dan cambios en la organización debido al desarrollo de sistemas más eficaces.

Debido a que los sistemas de información juegan un papel crítico en los negocios, gobierno y en la vida diaria, entonces debemos asegurarnos que sean precisos, confiables y seguros.

COMPETENCIAS DESARROLLADAS

15. Competencias desarrolladas y/o aplicadas.

1. Aplique habilidades de ingeniería para el desarrollo del sistema fortaleciendo la empresa Golden English.

2. Diseña e innova los procesos en base a la necesidad de la empresa.
3. Gestiona eficientemente los recursos de la empresa con visión a futuro con el fin de suministrar los datos redactados.
4. Aplica métodos cuantitativos y cualitativos en el análisis e interpretación de datos y modelado de sistemas en los procesos organizacionales, para la mejora continua atendiendo estándares de calidad mundial.
5. Gestiona sistemas integrales de calidad para la mejora de los procesos, ejerciendo un liderazgo estratégico y un compromiso ético.
6. Aplica las normas legales para la creación y desarrollo del sistema.
7. Utiliza las nuevas tecnologías de información y comunicación en la organización, para optimizar los procesos y la eficaz toma de decisiones.
8. Analiza las variables económicas para facilitar la toma estratégica de decisiones en la empresa.

FUENTES DE INFORMACIÓN

16. Fuentes de información

(S.C.M, 2018)

Bibliografía

S.C.M, E. (29 de 05 de 2018). *monografias.com*. Obtenido de monografias.com:
<http://www.monografias.com/trabajos66/elaboracion-informe-residencias-profesionales/elaboracion-informe-residencias-profesionales2.shtml>

(ASML, 2018)

Bibliografía

ASML, H. C. (30 de 05 de 2018). *es.scribd.com*. Obtenido de es.scribd.com:
<https://es.scribd.com/doc/16155626/proyecto-base-de-datos>

S.C.M, E. (29 de 05 de 2018). *monografias.com*. Obtenido de monografias.com:
<http://www.monografias.com/trabajos66/elaboracion-informe-residencias-profesionales/elaboracion-informe-residencias-profesionales2.shtml>