

Instituto Tecnológico de Pabellón de Arteaga

Tecnologías de la Información y de las Comunicaciones

Desarrollo, Aplicación y Consulta

Tareas

9^{no} Semestre

Maestro: Eduardo Flores González

Cinthia Marleny Huerta Valdivia

Fecha: 19/Septiembre/2018

Contenido

| | |
|--|----|
| Lectura 2 | 2 |
| Herramientas para obtener el mejor rendimiento en marketing | 2 |
| Entregables del proyecto Unidad 1 | 3 |
| Sistema de Información para la administración del gimnasio Animal GYM..... | 3 |
| Lectura 3 | 10 |
| Tendencias de la Industria de Desarrollo de Software | 10 |
| Lectura 4 | 11 |
| 8 señales que indican la necesidad de un consultor | 11 |
| Resumen del libro “Aprendiendo UML en 24 horas” – Parte 1 | 12 |
| Resumen del libro “Aprendiendo UML en 24 horas” – Parte 2..... | 14 |
| Trabajos citados | 16 |

Lectura 2

Herramientas para obtener el mejor rendimiento en marketing

La directora de marketing de IBM Michelle Peluso encargó una plataforma a medida que pone todos sus datos de marketing y visualiza la efectividad de la campaña. Este programa se llama Pearl, el cual agrega datos de numerosos recursos de marketing de terceros y los transforma en cuadros y gráficos. Esto ayuda a la toma de decisiones más informadas al momento de crear una campaña y así nos muestra que el dinero invertido de manera eficiente.

La página Top Sheet, fue diseñada por IBM para mostrar un resumen de la efectividad de cada campaña de marketing la cual utiliza las imágenes de los círculos para mostrar el progreso de cada proyecto: los círculos más grandes representan campañas más exitosas. La información se agrega de una variedad de fuentes externas, incluso de algunas de las herramientas más populares en marketing, incluidas las de startups respaldadas por Venture. Los programas integrados a Pearl son Lotame, Optimizely, HotJar y muy pronto lo estará Sprinklr. [1]

Entregables del proyecto Unidad 1

Sistema de Información para la administración del gimnasio Animal GYM

Identificación del problema

Se ha identificado la falta de interacción con el cliente en el gimnasio Animal Gym, dado que los clientes de éste no tienen manejo de su información personal como, las rutinas que deben de llevar día con día, revisión de sus pagos mensuales, y no tienen una comunicación directa con el nutriólogo. Así mismo el dueño tiene problemas para registrar los pagos, cada cuándo se debe dar mantenimiento a los aparatos y cuantas veces se ha llamado al quiropráctico.

Oportunidades

Este tipo de sistemas no se ha implementado en este gimnasio, sólo se utiliza el sistema de pagos mensuales y con este proyecto se podría mejorar la forma en la que se interactúa con los clientes de una manera más innovadora y así poder ayudarles más en su avance y poder darles motivación.

Objetivo General

Desarrollar un sistema de información el cual exista una comunicación y administración con el cliente y éste tenga la información actualizada, de la misma manera que el dueño y los instructores tengan conocimiento del avance de cada cliente, así como el registro de sus pagos mensuales.

Objetivos específicos

- Diseñar interfaces aplicadas a usuarios, administrador, instructores etc.
- Desarrollar un sistema web con Node.js
- Desarrollar una BD en PostgreSQL
- Sistema compatible con computadoras y celulares
- Crear manual de usuario

Actores

- Administrador: Será el encargado de sistemas y quién tendrá acceso a toda la información de pagos, chequeos con nutriólogo, registro de mantenimiento y avances de los clientes con ayuda de sus instructores.
- Instructores: Encargados de las rutinas individuales de los clientes y checar su avance
- Clientes: Tener a la mano la información de sus pagos mensuales, sus rutinas, citas y avance.
- Mantenimiento: Reparación de los aparatos que sufran algún desperfecto
- Nutriólogo: Encargado de entregar a los clientes una dieta saludable
- Quiropráctico: Encargado de revisar las lesiones de lo clientes

Riesgos

- No podría concluirse por falta de conocimiento de las herramientas y por falta de experiencia
- Podrá no concluir por falta de tiempo
- No podrá terminarse si no se cuenta con el apoyo monetario suficiente
- Podrá no terminarse en caso de que el cliente pierda el interés

Viabilidad del sistema

- El sistema es viable ya que el gym no cuenta con un sistema de este tipo para resolver la problemáticas entre los clientes y el control que tienen los dueños.
- Es viable ya que los encargados lo encuentran útil para cubrir las necesidades

Estudio de Factibilidad

- Organizacional: Se garantiza la elaboración del software, ya que los integrantes del equipo de trabajo lo realizarán personalmente y se entregarán evidencias de su desarrollo
- Técnica: Se cuenta con dos equipos de cómputo para la elaboración y ejecución de este proyecto, tenemos acceso fácil a Internet y apoyo por parte del docente tutor del proyecto.
- Se elaborará con las herramientas mencionadas en el entorno de trabajo porque son muy factibles y actualizadas para el desarrollo de un sistema web
- Financiera: Se cuenta con poca disponibilidad de capital para la realización del proyecto, a lo que se sugiere solicitar apoyo al ITPA en caso de ser necesario.
- Se solicitará el apoyo para la compra de las licencias

Entorno de trabajo

Se utilizarán las siguientes herramientas para la elaboración del proyecto:

- NodeJS
- JavaScript
- Bootstrap
- PostgreSQL

Cronograma de Trabajo

| CRONOGRAMA DE TRABAJO | | | | | |
|-----------------------|--|---------------------------------------|---|---|-----------------------------------|
| MES | ACTIVIDADES A REALIZAR | | | | |
| Agosto | Búsqueda de problemática | Búsqueda de posibles soluciones | Búsqueda de herramientas para solucionar la problemática | Definición de objetivos, área de oportunidad y estudio de factibilidad del proyecto | Definición de actividades futuras |
| Septiembre | Instalación del software necesario | Chequeo de requerimientos del sistema | Diseño de BD | | Diseño de Interfaz |
| Octubre | Utilización de Bootstrap para una mejor visualización y compatibilidad | Conexión de la interfaz con la BD | Pruebas de funcionalidad (Se dejará que una persona tome el papel de un usuario para que pruebe el sistema e identificar fallas y posibles mejoras) | | |
| Noviembre | Mejorar el software | Chequeo de Calidad de software | Realizar manual de usuario | Últimas pruebas del sistema | Entrega del sistema |

Requerimientos funcionales

| | | |
|--|----------------|------------------|
| RF1. Todos los usuarios tendrán un login | | |
| Funcional | Prioridad Alta | Dificultad media |

| | | |
|---|-----------------|------------------|
| RF2. EL administrador será quien los registre | | |
| Funcional | Prioridad Alta | Dificultad media |
| RF3. Las contraseñas pueden cambiarse después del registro (No funcional) | | |
| Funcional | Prioridad | Dificultad |
| RF4. El cliente consultará sus citas | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF5. El cliente tiene consultará sus rutinas | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF6. El cliente tiene consultará sus pagos | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF7. El cliente recibe alerta de mensualidad por vencerse | | |
| Funcional | Prioridad Media | Dificultad Baja |
| RF8. El administrador puede marcar como pagado en efectivo | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF9. El instructor subirá las rutinas al cliente | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF10. El instructor puede consultará el avance del cliente | | |
| Funcional | Prioridad Media | Dificultad Media |
| RF11. El cliente registrará su avance | | |
| Funcional | Prioridad Alta | Dificultad Alta |
| RF12. El cliente puede consultar el avance | | |
| Funcional | Prioridad Media | Dificultad Media |
| RF13. El administrador contactará a mantenimiento | | |
| Funcional | Prioridad Media | Dificultad Media |

| | | |
|--|-----------------|------------------|
| RF14. El cliente podrá hacer citas con el nutriólogo | | |
| Funcional | Prioridad Alta | Dificultad Media |
| FR15. El cliente podrá hacer citas con el quiropráctico | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF16. El administrador consultará el registro de pagos | | |
| Funcional | Prioridad Alta | Dificultad Baja |
| RF17. El administrador consultará las citas de los clientes | | |
| Funcional | Prioridad Media | Dificultad Baja |
| RF18. El administrador consultará los registros de mantenimiento | | |
| Funcional | Prioridad Media | Dificultad Baja |
| RF19. El de mantenimiento consultará las citas de reparación | | |
| Funcional | Prioridad Alta | Dificultad Baja |
| RF20. El nutriólogo consultará las citas | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF21. El nutriólogo podrá ver el avance de las rutinas del cliente | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF22. El quiropráctico consultará las citas | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF23. El quiropráctico consultará las rutinas del cliente | | |
| Funcional | Prioridad Alta | Dificultad Media |
| RF24. | | |
| | | |
| RF25. | | |
| | | |

Requerimientos no funcionales

| | | |
|--|----------------|------------------|
| RNF1. Las sesiones de los usuarios durarán 20 minutos | | |
| No funcional | Prioridad Alta | Dificultad Media |
| RNF2. Los métodos de pago será por PayPal y por efectivo | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF3. Campos validados en HTML5 | | |
| No funcional | Prioridad Alta | Dificultad Media |
| RNF4. Campos validados en JavaScript | | |
| No funcional | Prioridad Alta | Dificultad Media |
| RNF5. Señalar los campos obligatorios | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF6. Los permisos de acceso solo pueden ser cambiados por el Admin. | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF7. Los registros de usuarios deben revisarse | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF8. Manuales de usuario | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF9. Manuales de Instructores | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF10 Manuales para el administrador | | |
| No funcional | Prioridad alta | Dificultad Media |

| | | |
|--|----------------|------------------|
| RNF11. Debe contener mensajes de error y que estén bien explicados | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF10. Tendrá un apartado de ayuda | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF11. Diseño adaptable a laptop, teléfono o tablet | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF12. Usar un framework CSS para el diseño | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF13. El sistema debe estar disponible para acceder a cualquier hora | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF14. El sistema debe proteger los datos de personas externas | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF15. Después de 4 intentos fallidos en login bloquea el usuario | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF16. Para Reiniciar el sistema solo se tendrán 5 seg. | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF17. El sistema será desarrollado en Node js | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF18. El sistema tendrá BD en Postgresql | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF19. El sistema tendrá compatibilidad con navegadores Chrome y Firefox | | |

| | | |
|---|----------------|------------------|
| No funcional | Prioridad alta | Dificultad Media |
| RNF20.El sistema debe cumplir con los objetivos ya establecidos | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF21. Los mensajes del sistema deben estar en español | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF22. | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF23. El sistema debe de ser compatible con diferentes SO | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF24. El sistema tendrá un registro de inicios de sesión | | |
| No funcional | Prioridad alta | Dificultad Media |
| RNF25. | | |
| No funcional | Prioridad alta | Dificultad Media |

Lectura 3

Tendencias de la Industria de Desarrollo de Software

En los últimos años se ha popularizado y el trabajo en las áreas de lenguajes de programación, la interfaz de usuario, programación, ingeniería de software, métodos de análisis y diseño y equipos de desarrollo han evolucionado al paso de los años y han avanzado demasiado en estos últimos años. Por ejemplo, en la interfaz de usuario que hace muchos años se utilizaba estar impresa en papel y hasta los 80's se comenzaron a utilizar los monitores. Ahora ya se utilizan sistemas de ventanas con objetos gráficos manejados mediante el teclado y el ratón.

La industria del software debe ser competitiva en estos tiempos de globalización se debe tener más productividad mediante herramientas de programación con soporte de asistentes, tener interoperabilidad para que las aplicaciones de diversas empresas se deban operar entre sí. La interfaz de usuario debe ser única y usar lenguajes de

programación estándar. Los métodos de desarrollo orientados a objeto deben estar afianzados y el software debe ser eficiente, usable y mantenible.

El desarrollo de software será simplificado en el futuro por el uso de herramientas más poderosas de programación integradas a otras herramientas de software que soporten las demás actividades. [2]

Lectura 4

8 señales que indican la necesidad de un consultor

Los consultores son contratados por tienen enormes ventajas y fortalezas, en empresas de todo tamaño. Las compañías más grandes del mundo, siempre tienen un staff de consultores externos que los apoyan. La mayor ventaja del consultor es poder ver desde afuera lo que sucede en la empresa, esa posibilidad de tener una visión objetiva, más la experiencia acumulada en otras compañías, le permite ver cosas que otros no.

Llevo diez años –de mis 30 como profesional– dedicado totalmente a la consultoría, especializado en reingeniería financiera y en temas de estrategia gerencial. Mi experiencia me dice que la inversión que hace una compañía en una consultoría calificada y bien enfocada, se paga muchas veces y se recupera en un plazo muy corto. La inversión se paga porque al resolver problemas, aumentan la capacidad de las empresas para mejorar sus ganancias o evitar situaciones de crisis.

Se contratan consultores externos por razones que han sido muy buenos servicios de consultor, por la necesidad de diagnosticar problemas y hallar soluciones, la necesidad de una reestructuración completa. Un ejemplo es que, en una empresa, se tuvo que invertir casi \$4 millones de dólares para duplicar su capacidad de producción. Tuvo que hacerse por dos razones:

1) El mercado se estaba expandiendo y el producto de esta empresa estaba en una demanda creciente.

2) El nivel de deuda de la compañía exigía un nivel de ventas más alto para cubrir sus crecientes costos fijos, principalmente el pago de la deuda. El punto de equilibrio financiero había aumentado en casi un 60%.

También se contratan consultores por la necesidad de capital y financiamiento por las reglamentaciones gubernamentales e impuestos. Cuando estas empresas operan en diversos países, este asunto es más complejo. Al abrir en un nuevo país, lo más efectivo es contratar consultores externos conocedores y experimentados en esos nuevos mercados que la empresa quiere operar. La necesidad de elevar el nivel de eficiencia y competencia. Un consultor experimentado y experto puede colaborar con el empresario, grandemente en esto. Las ganancias son enormes, a veces con pequeños; pero precisos cambios en las áreas de la empresa que tienen capacidad de generar más valor.

La necesidad de incrementar sus ventas, la necesidad de ideas frescas y por políticas de la compañía. Es muy indispensable la contratación de un consultor por estas razones y se pueden obtener grandes beneficios.

El consultor funciona como un filtro que evalúa objetivamente las diferentes propuestas.

Hay varias razones adicionales que llevan a las empresas a contratar consultores externos:

- La necesidad de contratar personal temporal.
- La necesidad de capacitar empleados.
- Todos los temas que tienen que ver con los computadores y el procesamiento de datos.

Y ahora, cada vez más, todo lo referente a lograr que las compañías tengan presencia en internet. Este mundo cada día es más virtual y, año con año, las ventas que se hacen por medio de Internet.

Contratar consultores experimentados puede contribuir a mejorar los niveles de eficiencia general de la empresa, aumentar las ganancias y le ayudan al empresario a llevar sus negocios al siguiente nivel de crecimiento. [3]

Resumen del libro "Aprendiendo UML en 24 horas" – Parte 1

El UML es un modelo con el cual se pueden representar de una manera gráfica mediante diagramas que es lo que hará el sistema que se desarrollará, existen varios tipos de diagramas UML como el diagrama de clases, el diagrama de objetos, de casos de uso, de estados, de secuencias, de actividades, colaboraciones, componentes y de distribución. Son varias maneras en las que se puede representar el qué hará el sistema.

Los objetos en los diagramas UML son de mucha importancia ya que contienen propiedades y acciones, la orientación a objetos depende de varios fundamentos importantes como la herencia que se refiere a heredar los atributos y operaciones de su clase y la clase también puede heredar de otra clase. El polimorfismo se refiere a que es una acción que se puede usar en objetos diferentes, pero se usa la acción de distintas maneras. Entre otros fundamentos se encuentran los objetos que son para mostrar una "cara" al exterior ocultando su funcionalidad con una interfaz.

Las clases se representan mediante rectángulos y éste contiene el nombre, los atributos, las operaciones y responsabilidades dividiéndose por líneas. Para tener un mejor control en la clase se le agregan restricciones, las cuales tienen mayor información como las notas adjuntas.

Los diagramas de clases nos ayudan con la interacción con el cliente para que nos pueda dar más información sobre el sistema.

Las relaciones son muy importantes al momento de utilizar las clases de el UML, de esta manera se pueden guiar. La asociación es la conexión fundamental entre las clases, las multiplicidades muestran la relación de cuántos a cuántos se pueden conectar. La herencia es representada por una línea seguida de un triángulo sin rellenar que apunta a la clase principal.

Las dependencias son representadas por una línea discontinua de que une las dos clases de la dependencia.

Las representaciones de las agregaciones en UML son similares a las composiciones. La línea que las une tiene un rombo, en las agregaciones el rombo no está lleno, pero en las composiciones el rombo que lo une si lo está. Un diagrama de contexto de

composición es como un mapa detallado de un mapa mayor, muestra las clases anidadas en el diagrama y representa la forma en que se relaciona con otros objetos del sistema. Una realización es una asociación entre una clase y una interfaz, la cual contiene operaciones que solo algunas clases pueden utilizar. La realización se representa mediante una línea continua con una punta de flecha en forma de triángulo sin rellenar, la cual conecta la clase con la interfaz.

Hay tres niveles de visibilidad que son el protegido (#), donde su funcionalidad se puede extender a las clases secundarias, el privado (-), en donde sus atributos y operaciones solo se pueden usar en esa misma clase y la pública (+) donde se cualquier clase puede utilizar sus funciones. Otro aspecto de los atributos y operaciones es el ámbito, que tiene dos tipos: en el ámbito de instancia, cada objeto de la clase tiene su propio valor mediante un conjunto de objetos y en el ámbito de archivador solo tiene un valor para un atributo u operación.

Los casos de Uso nos sirven para ver la representación gráfica mediante los usuarios potenciales con escenarios donde el actor es una persona, componente, hardware, etc. Y debe dar de resultado algo de valor. Nos muestra de una manera muy simple una secuencia de pasos por cada caso de uso. La entrevista con el usuario es la forma mas sencilla para realizarlos. El objetivo es conocer los candidatos del caso de uso y sus actores.

Los casos de uso nos facilitan la comunicación con los clientes y los analistas del sistema solo usando un diagrama. Se representa mediante un símbolo de un actor con una figura humana y el caso de uso mediante un ovalo, uniéndolos con una línea.

Los diagramas de estados de UML capturan los cambios de estados como de sucesos y tiempo. Un estado se representa mediante un rectángulo de vértices redondeados y una línea continua con punta de flecha representa la transición de estados, puede contener variables y actividades del estado. La transición puede ocurrir al cumplirse una condición o condición de seguridad. El estado puede tener subestados que pueden ser secuenciales o concurrentes.

El estado histórico superficial recuerda solo es subestado principal y un estado histórico profundo recuerda todos los niveles de subestados. Una señal es un objeto y puede crear una jerarquía de herencia de señales. Los desarrolladores deben de conocer el comportamiento de todos los objetos para aplicarlos al software.

Los diagramas de secuencias le agregan a los diagramas UML el tiempo de las interactividades de los objetos. Los objetos se colocan arriba y el tiempo va hacia abajo. Los mensajes son flechas que conectan una línea de vida con otra. La ubicación del mensaje representa en que momento ocurrirá dependiendo el tiempo. Algunos sistemas una operación se puede activar a si misma conociéndose como recursividad y representada con una flecha de activación hacia sí misma con un pequeño rectángulo sobre la activación. Un diagrama de colaboraciones es similar al de secuencias, con la diferencia de que es de acuerdo al espacio. El diagrama de colaboraciones muestra las asociaciones entre los objetos como los mensajes que pasan entre objetos.

Los diagramas de actividades son como los diagramas de flujo mostrando los pasos a seguir, son útiles para las operaciones de los objetos y procesos. Se representan como un rectángulo con esquinas redondeadas. Se puede mostrar una señal representada con un pentágono convexo y la recepción con un cóncavo. Un diagrama de componentes representa un elemento real que es un componente de software, el cual se accede mediante la interfaz, la relación entre componente e interfaz se llama realización. La representación de un componente es mediante un rectángulo con dos

pequeños encima de éste al lado izquierdo. Se puede conectar con la interfaz con una línea discontinua y una flecha con un triángulo sin relleno o con un pequeño círculo en una línea continua, mostrando una relación de realización. [4]

Resumen del libro “Aprendiendo UML en 24 horas” – Parte 2

En UML se utilizan también los diagramas de distribución, lo cuales nos muestran como se ve un sistema físicamente cuando se conjuga. Los sistemas contienen nodos y éstos se representan mediante cubos, donde las conexiones entre ellos se unen por una línea. Hay dos tipos de nodos que son los de procesador y dispositivo, en donde el procesador ejecuta un componente y el dispositivo interactúa con el mundo. Estos diagramas son de utilidad al momento de simular redes como token-ring, ARCnet, thin ethernet e inalámbricas como Ricochet.

El UML también puede aplicarse a situaciones reales y no sólo de sistemas. El UML tiene 4 capas que son los objetos de usuario, modelado, metamodelado y metametamodelado. Donde se analiza un sistema normalmente se trabaja con las primeras dos capas. En la tercera vienen los conceptos del UML y en la cuarta se orientan a la teoría y diseñadores de lenguaje.

Las metodologías son muy importantes para que se puedan estructurar los segmentos y actividades de un proyecto, de lo contrario si no se usara sería un caos. Una metodología que nos da un patrón para el proceso de desarrollo, se llama GRAPPLE (Directivas para el Rápido Desarrollo de Aplicaciones), el cual consta de 5 segmentos: Recopilación de necesidades, Análisis, Diseño, Desarrollo y Distribución.

Se analiza un caso donde se usa el proceso de desarrollo en UML. En el caso, un restaurante decide aplicar la tecnología de cómputo en el restaurante del futuro. El analista debe comprender todo el proceso del desarrollo, recopilar las necesidades y acciones que se requieren para posteriormente usarlas en el modelo GRAPPLE.

La entrevista es muy útil para realizar el análisis del dominio, se usan clases para comprenderlo mejor y agruparlas de manera en que se agruparan en el diagrama. La derivación de objetos compuestos o agregados nos ayudan a organizar el modelo. Se requerirán de más entrevistas para dar cuerpo al modelo, aunque ya en este punto se podrían agregar atributos y operaciones.

El siguiente paso es que el equipo de desarrollo ya tenga una visión mas clara sobre lo que el sistema de cómputo realizará para el restaurante. En este ejemplo se decidió por tener la agilización de la transmisión de información y se buscarán los tipos de tecnología a utilizar. Para ver los requerimientos del sistema, el equipo de desarrollo se reúne con los usuarios potenciales y los expertos de dominio, dando como resultado un diagrama de paquetes donde cada uno representa una sección principal de su funcionalidad.

Los casos de uso son muy útiles para el desarrollo del sistema, sin embargo, no sólo se tiene que hacer esto. El equipo de desarrollo tiene que comprender cada uno de ellos detalladamente para su comprensión. Para analizar los diagramas de casos de uso, se tiene que analizar la descripción y derivar las condiciones previas y resultantes, especificando los pasos. Aparte de los casos de uso se tienen que analizar los componentes del sistema que surgieron a partir de éste. Todo esto es necesario para

que los programadores realicen más fácil su trabajo como en la codificación de los objetos del sistema y la forma en que se comunican entre ellos. Después de esto, el sistema ya está mas cerca de hacer realidad.

Al llegar a la parte del diseño del proyecto, se tiene que enfocar e la interfaz del usuario y la distribución del sistema que salen de los casos de uso. El sistema debería permitir al usuario completar cada caso de uso, por esto es que es muy importante la interfaz. El ingeniero de sistemas se debe orientar a la arquitectura física del sistema. El ingeniero del sistema hace un diagrama de distribución UML que nos muestra los nodos, los componentes y las conexiones entre nodos. Aunque los detalles no aparecen en el proceso GRAPPLE se pueden empezar a pensar en ellos en etapas previas. Al finalizar el modelado del sistema, éste se puede reutilizar en otras nuevas ideas de negocios.

Una clase es parametrizada cuando sus parámetros están desvinculados, al vincularlos se crea una clase. Una colaboración parametrizada nos sirve como un patrón de diseño, la cual es útil en diversos dominios.

El patrón de diseño se ocupa de que los objetos pasen una petición entre ellos hasta que uno pueda manejarla. Permiten que los diseñadores vuelvan a utilizar las soluciones ya probadas, incorporar componentes sólidos en diseño y documentar los sistemas

Los sistemas incrustados son computadoras que se encuentran dentro de un dispositivo como un electrodoméstico. U sistema incrustado podría ser tolerante, ósea que no tiene que cumplir con plazos rigurosos o puede ser estricto, que sí tiene que cumplir los plazos.

Los conceptos importantes de los sistemas incrustados son el tiempo, las interrupciones y subprocesos. Un sistema operativo en tiempo real dirige tráfico entre subprocesos e interrupciones. El programador del tiempo del núcleo determina cuál tarea se realiza a continuación, un núcleo puede ser preferencial o cooperativo.

El modelo UML de puede aplicar en diversas áreas y como para el modelado de negocios. Para modelar una GUI hacemos un diagrama híbrido mostrando las relaciones de espacio de los componentes de la pantalla, mostrando las conexiones y sus casos de uso.

Las reglas de condiciones (if-then) en un sistema experto, son el bloque de construcción de una base de conocimientos. Se sugiere hacer un diagrama para visualizar las reglas y sus relaciones internas. En el diagrama, un cuadro dividido en compartimentos modela a la regla. En una parte está el identificador, en la otra la parte if, otro la parte then en las siguientes dos las reglas relacionadas. Los vínculos de las reglas adyacentes se unen con líneas de conexión entre las partes adecuadas.

Trabajos citados

- [1 B. Peterson, «Business Insider,» 25 Agosto 2018. [En línea]. Available:
] https://www.businessinsider.com/ibm-marketing-pearl-optimizely-sprinklr-2018-8?utm_content=bufferc78b7&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer-tiuk#sprinklr-a-social-media-management-platform-that-lets-marketers-see-the-impact-of-omn. [Último acceso: 31 Agosto 2018].
- [2 M. J. P. Mercado, «Scielo,» 2001. [En línea]. Available:
] http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892001000200009. [Último acceso: 07 09 2018].
- [3 E. N. Montenegro, «Enrique Montenegro,» 31 07 2012. [En línea]. Available:
] <http://www.enriquemontenegro.com/8-senales-que-indican-la-necesidad-de-un-consultor/>. [Último acceso: 07 09 2018].
- [4 J. Schmuller, «Aprendiendo UML en 24 horas,» Pr4entice Hall, p. 404.
]