**Documento de Especificación del Sistema**

**DeliverYApp**

**Indice**

Contenido

[**1.** **Introducción** 6](#_Toc453203830)

[**1.1** **Propósito del Documento** 6](#_Toc453203831)

[**1.2** **Alcance del Producto** 6](#_Toc453203832)

[**1.3** **Definiciones, acrónimos y abreviaturas** 6](#_Toc453203833)

[1.3.1 Del Negocio 6](#_Toc453203834)

[1.3.2 Del Sistema 6](#_Toc453203835)

[1.3.3 De Tecnología 7](#_Toc453203836)

[**1.4** **Referencias** 7](#_Toc453203837)

[**1.5** **Visión General del documento** 7](#_Toc453203838)

[**2.** **Descripción General** 7](#_Toc453203839)

[**2.1** **Perspectiva del producto.** 7](#_Toc453203840)

[**2.2** **Funciones del Producto**. 7](#_Toc453203841)

[**2.3** **Características de los usuarios**: 8](#_Toc453203842)

[**2.4** **Restricciones** 8](#_Toc453203843)

[**2.5** **Suposiciones y Dependencias** 9](#_Toc453203844)

[**3.** **Requisitos Específicos** 9](#_Toc453203845)

[**3.1** **Interfaces Externas** 9](#_Toc453203846)

[**3.2** **Funciones del Sistema** 9](#_Toc453203847)

[**3.3** **Rendimiento del sistema** 9](#_Toc453203848)

[**3.4** **Requerimientos lógicos del sistema de base de datos** 9](#_Toc453203849)

[**3.5** **Propiedades emergentes del Sistema** 9](#_Toc453203850)

[Capítulo I 10](#_Toc453203851)

[Capítulo II 13](#_Toc453203852)

[Modelado de Requerimiento del Sistema 14](#_Toc453203853)

[Descripción Escrita del modelado del Sistema 14](#_Toc453203854)

[Capítulo III 18](#_Toc453203855)

[**Estudio de Viabilidad** 18](#_Toc453203856)

[Capítulo IV 20](#_Toc453203857)

[Calendarización 20](#_Toc453203858)

[Capítulo V 22](#_Toc453203859)

[**COCOMO2** 23](#_Toc453203860)

[Características 23](#_Toc453203861)

[Objetivos para la construcción de COCOMO II 23](#_Toc453203862)

[Formula 24](#_Toc453203863)

[Modelos de COCOMO II 24](#_Toc453203864)

[Ventajas 24](#_Toc453203865)

[Desventajas 24](#_Toc453203866)

[Conclusión 24](#_Toc453203867)

[Capítulo VI 25](#_Toc453203868)

[Plan de riesgos 26](#_Toc453203869)

[Introducción 26](#_Toc453203870)

[Propósito 26](#_Toc453203871)

[Alcance 26](#_Toc453203872)

[Análisis de Riesgos 26](#_Toc453203873)

[Capítulo VII 29](#_Toc453203874)

[Plan de Gestión de Configuración 29](#_Toc453203875)

[Introducción 29](#_Toc453203876)

[Propósito 30](#_Toc453203877)

[Alcance 30](#_Toc453203878)

[Gestión de Configuración 30](#_Toc453203879)

[Herramientas, Entorno e Infraestructura 31](#_Toc453203880)

[Herramientas 31](#_Toc453203881)

[Programa de la Gestión de Configuración 31](#_Toc453203882)

[Identificación de la Configuración 31](#_Toc453203883)

[Elementos de Configuración 31](#_Toc453203884)

[Nomenclatura de Elementos 31](#_Toc453203885)

[Elementos de la Línea Base del Proyecto 37](#_Toc453203886)

[Control de Configuración 37](#_Toc453203887)

[Solicitud de Cambios 37](#_Toc453203888)

[Aprobación de Cambios 37](#_Toc453203889)

[Implementación de Cambios 37](#_Toc453203890)

[Estado de la Configuración 38](#_Toc453203891)

[Informes y Auditorías 38](#_Toc453203892)

[Calendario 38](#_Toc453203893)

[Capacitación y Recursos 38](#_Toc453203894)

[Mantenimiento del Plan de Gestión de la Configuración 38](#_Toc453203895)

[Capítulo VIII 39](#_Toc453203896)

[Descripción de Roles 40](#_Toc453203897)

[Líder del Proyecto: 40](#_Toc453203898)

[Documentador: 40](#_Toc453203899)

[Desarrollador: 40](#_Toc453203900)

[Tester: 41](#_Toc453203901)

[Gestor de Riesgo: 41](#_Toc453203902)

[Gestor de Configuración: 41](#_Toc453203903)

[Capítulo IV 42](#_Toc453203904)

[Propósito 43](#_Toc453203905)

[Referencias 43](#_Toc453203906)

[Gestión 44](#_Toc453203907)

[Organización 44](#_Toc453203908)

[Actividades 44](#_Toc453203909)

[Ciclo de vida del software cubierto por el Plan 44](#_Toc453203910)

[Actividades de calidad a realizarse 44](#_Toc453203911)

[Relaciones entre las actividades de SQA y la planificación 46](#_Toc453203912)

[Responsables 46](#_Toc453203913)

[Documentación 46](#_Toc453203914)

[Propósito 46](#_Toc453203915)

[Documentación mínima requerida 46](#_Toc453203916)

[Especificación de requerimientos del software 47](#_Toc453203917)

[Descripción del diseño del software 48](#_Toc453203918)

[Plan de Verificación & Validación 49](#_Toc453203919)

[Documentación de usuario 49](#_Toc453203920)

[Plan de Gestión de configuración 49](#_Toc453203921)

[Propósito 49](#_Toc453203922)

[Resumen 49](#_Toc453203923)

[Organización, Responsabilidades 50](#_Toc453203924)

[Herramientas, Entorno, e Infraestructura 50](#_Toc453203925)

[Forma de trabajo 50](#_Toc453203926)

[Control de Cambios 50](#_Toc453203927)

[Reportes y Auditorias 51](#_Toc453203928)

[Otros documentos 51](#_Toc453203929)

[Estándares, prácticas, convenciones y métricas 51](#_Toc453203930)

[Objetivos 51](#_Toc453203931)

[Métricas de proceso 53](#_Toc453203932)

[Métricas de proyecto 53](#_Toc453203933)

[Métricas de producto 54](#_Toc453203934)

[Estándar de documentación 54](#_Toc453203935)

[Estándar de verificación y prácticas 55](#_Toc453203936)

[Otros Estándares 55](#_Toc453203937)

[Revisiones y auditorías 55](#_Toc453203938)

[Objetivo 55](#_Toc453203939)

[Requerimientos mínimos 55](#_Toc453203940)

[Revisión de requerimientos 55](#_Toc453203941)

[Revisión de diseño preliminar 55](#_Toc453203942)

[Revisión de diseño crítico 56](#_Toc453203943)

[Auditoría funcional 56](#_Toc453203944)

[Auditoría física 56](#_Toc453203945)

[Auditorías internas al proceso 56](#_Toc453203946)

[Revisiones de gestión 56](#_Toc453203947)

[Revisión del Plan de gestión de configuración 56](#_Toc453203948)

[Revisión Post Mortem 56](#_Toc453203949)

[Agenda 56](#_Toc453203950)

[Otras revisiones 56](#_Toc453203951)

[Revisión de documentación de usuario 56](#_Toc453203952)

[Verificación 56](#_Toc453203953)

[Reporte de problemas y acciones correctivas 56](#_Toc453203954)

[Herramientas, técnicas y metodologías 57](#_Toc453203955)

[Gestión de riesgos 57](#_Toc453203956)

[Formulario de Pedidos y Detección de Cambios 57](#_Toc453203957)

1. **Introducción**
   1. **Propósito del Documento**

Este documento tiene como propósito definir las especificaciones funcionales y no funcionales del sistema para la implementación de una aplicación WEB, capaz de gestionar los pedidos de los restaurantes.

* 1. **Alcance del Producto**

Diseño, desarrollo e implementación del Sistema “DeliverYApp”.

DeliverYApp será un entorno web en el que se podrán registrar las empresas gastronómicas y así de esa forma poder ofrecer el servicio de delivery, dicho sistema permitirá efectuar pedidos del menú seleccionado por el cliente, una vez confirmado el pedido, la recepcionista recibe las notificaciones y se genera un nuevo pedido con estado pendiente, una vez que el menú se encuentra listo para su entrega, se efectuara el envió del pedido y este pasa a estado enviado, y por ultimo una vez cobrado se finaliza el pedido y este pasa a estado pagado.

No contará con la funcionalidad de facturación y contabilidad.

Los objetivos del sistema son minimizar los problemas referentes a la gestión de pedidos de parte del cliente, optimizar la gestión de ventas del restaurante.

El beneficio con que cuenta el sistema es el de agilizar los pedidos de tal forma que no se pierda tiempo con las llamadas.

La meta a alcanzar es mejorar el servicio de delivery y aumentar las ventas.

* 1. **Definiciones, acrónimos y abreviaturas** 
     1. **Del Negocio**

1. **Pedido/s:** Proceso en el que el cliente confirma su pedido del menú.
2. **Cancelación de Pedido:** Proceso en el cual el cliente cancela su pedido.
3. **Recepcionista:** encargado de recibir los pedidos y procesarla.
   * 1. **Del Sistema**
4. **Administrar:** acción de agregar, modificar, eliminar y consultar la información de un determinado objeto o persona.
5. **Administrador del Sistema:** encargado del soporte técnico y operativo del Sistema.
6. **Registrarse:** Proceso en el cualel administrador da de alta a los usuarios del sistema, delegando a cada uno sus respectivos roles.
7. **Usuario:** persona que puede ingresar a la parte privada del Sistema utilizando su usuario y contraseña.
8. **Rol:** conjunto de permiso que pueden asignarse a un usuario.
9. **Cliente:** persona encargada de realizar los pedidos.
10. **Pruebas:** Proceso mediante el cual se realizan pruebas para el perfecto funcionamiento del Sistema.
    * 1. **De Tecnología**
11. **Servidor web:** es un programa que implementa el protocolo HTTP. Este protocolo está diseñado para transferir lo que llamamos hipertextos, pagina Web o páginas HTML.
12. **HTTP:** Protocolo de transferencia de hipertexto (HiperText Transfer Protocol) es el protocolo usado en cada transacción.
13. **Sistema de Gestión de Base de Datos (SGBD):** Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, la recepcionista y las aplicaciones que la utilizan.
14. **Base de Datos:** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.
15. **Aplicación:** es un programa informático para facilitar al usuario la realización de un determinado trabajo.
16. **MySQL:** Motor de bases de datos.
17. **C#:** Lenguaje de programación usado para la creación de aplicación.
18. **Apache:** servidor Web de código abierto.
19. **Dirección IP (Internet Protocol):** número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP.
20. **Balsamiq mockups:** es una aplicación para crear maquetas para interfaces gráficas para usuario. Le permite al diseñador diagramar widgets pre construidos.
21. **ArgoUML:** es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD. Dado que es una aplicación Java, está disponible en cualquier plataforma soportada por Java.
    1. **Referencias**

Especificación de Requisitos según el estándar de IEEE 830 – IEEE Std. 830-1998.

* 1. **Visión General del documento**

En este documento se encontrará las características del sistema, interfaces del sistema, características de los usuarios, descripción de los requerimientos funcionales, no funcionales y del sistema.

1. **Descripción General**
   1. **Perspectiva del producto.**

El sistema DeliverYApp será un producto diseñado para trabajar en entornos web, lo que permitirá su utilización de forma descentralizada, además trabajará de manera independiente por lo tanto no interactuará con otro sistemas.

* 1. **Funciones del Producto**.

El sistema DeliverYApp permitirá realizar las siguientes funciones:

1. **Registro de Empresa (Login - Empresa):** Se le proveerá de un usuario y contraseña para que pueda acceder y tener control de los servicios que provee el sistema.
2. **Gestión de roles:** Se especificará los niveles de accesos del sistema.
3. **Gestión de usuario:** Se crearan los diferentes usuarios del sistema indicando sus roles.
4. **Gestión de menú:** Este contendrá un formulario para el registro de los datos del producto, platos, precios, foto, etc.
5. **Gestión de pedidos:** Este contendrá opciones para manejar los estados de los pedidos.
6. **Gestión de informes**: Este contendrá opción para generar informes estadísticos.
7. **Registro de cliente (Login - cliente):** El cliente deberá registrarse antes de solicitar el pedido.
8. **Elección y confirmación del pedido:** Este contendrá opciones de elección de los diferentes tipos de platos que a su vez deberá ser confirmado.
   1. **Características de los usuarios**:

* **Administrador del sistema:** usuario con gran conocimiento en el manejo del sistema, con una previa capacitación por parte de la entidad.
* **Usuario del sistema:** es el encargado de gestionar los pedidos, con una previa capacitación por parte de la entidad.
* **Cliente:** usuario visitante que solita los pedidos, debe contar con un conocimiento básico sobre el manejo de páginas web.
  1. **Restricciones**
* **Políticas de la empresa:** La aplicación se desarrollara mediante software de licencia abierta por lo tanto no se deberá pagar por el uso del servidor web(apache), sistema de gestión de base de datos (MySQL) y el lenguaje de programación C#, por lo tanto la utilización de estos programas se harán mediantes las políticas establecidas por este tipo de licenciamiento.
* **Limitaciones del hardware:** para esta aplicación será necesaria un computador servidor en el cual se instalara el servidor web apache, MySQl, C# y la aplicación DeliverYApp.
* **Interfaces con otras aplicaciones:** Debido a que el sistema no interactúa con otro sistema y es autónomo no se desarrollaran interfaces con otras aplicaciones. Las conexiones necesarias para la utilización del servidor web, MySQL, C# y un DNS será por medio de la configuración de estos programas.
* **Funciones de control:** El sistema debe controlar los permisos que tiene los usuarios para su accesibilidad de una manera correcta, de tal forma que pueda acceder la información que le corresponde de acuerdo a su rol. Debe tener controles adecuados para la validación de datos.
* **Lenguaje de programación:** Todo el material que se realice para la recepcionista y la aplicación debe estar en lenguaje español.
* **Protocolo de comunicación:** Se usar los protocolos TCP/IP, HTTP.
* **Requisitos de habilidad:** La información correspondiente al pedido como numero de documento, nombre, dirección, teléfono y elección de menú deben estar ajustado a la realidad para evitar inconvenientes y garantizar el pedido.
* **Credibilidad en la aplicación:** El sistema deberá ser sometida a una serie de pruebas para establecer que se encuentra conforme a los requerimiento que se plasman en el documento, en tanto en la consistencia de datos como el rendimiento de la aplicación, tales como tiempo de respuesta.
* **Consideración acerca de la seguridad:** Cada Usuario deberá autenticarse y su acceso verificado por una sola terminal para su respectiva labor de acuerdo de lo que su labor especifica.
  1. **Suposiciones y Dependencias**
* La red interna deberá estar configurada para el manejo de protocolos TCP/IP, HTTP, DNS principalmente todo lo relacionado a desempeño y seguridad.
* Debe realzarse una capacitación adecuada y acorde a lo que cada usuario va a realizar. Su capacitación se hará en el momento que sea necesario y a las personas indicadas.

1. **Requisitos Específicos**
   1. **Interfaces Externas**

El sistema DeliverYApp no tendrá interconexión con otros sistemas informáticos por lo tanto no es necesario la utilización de interfaces específicas para este punto.

* 1. **Funciones del Sistema**
* El cliente primeramente deberá registrarse para realizar su pedido.
* Deberá contar con los módulos citados más arriba.
* Contará con módulos de reportes estadísticos de los pedidos realizados.
  1. **Rendimiento del sistema**
* Necesitará una terminal como mínimo.
* El computador del usuario deberá contar con un navegador web (Google Chrome, Firefox, Internet Explorer, etc.).
* Conexión a internet.
  1. **Requerimientos lógicos del sistema de base de datos**
* El sistema debe permitir la retención de datos para guardar un registro de ellos para posterior reportes.
* El sistema debe tener en cuenta que información solo puede ser eliminada y cual es de gran importancia.
* La infraestructura debe ser capaz de soportar estos procedimientos.
  1. **Propiedades emergentes del Sistema**
* Seguridad: será por medio de un usuario y una contraseña, el encargado de la asignación de tal tarea será el administrador del sistema teniendo en cuenta el nivel de acceso a cada usuario.
* Portabilidad: puesto que funcionara con cualquier plataforma web.

1. **Necesidad de los diagramas UML**

UML (Lenguaje unificado de Modelado) es un conjunto de normas y estándares gráficos respecto a cómo se deben representar los esquemas relativos al software.

Mediante UML se pueden establecer los requerimientos y la estructura necesaria para desarrollar un sistema antes de codificarlo (escribirlo).

UML está compuesto de diferentes diagramas las cuales reflejan las diferentes etapas del desarrollo del sistema. Estos ayudan a detectar las dependencias y dificultades con la que contara el sistema y a reducir el margen de error a la hora de codificar.

# Capítulo I

**Modelado de Requerimiento del Negocio**

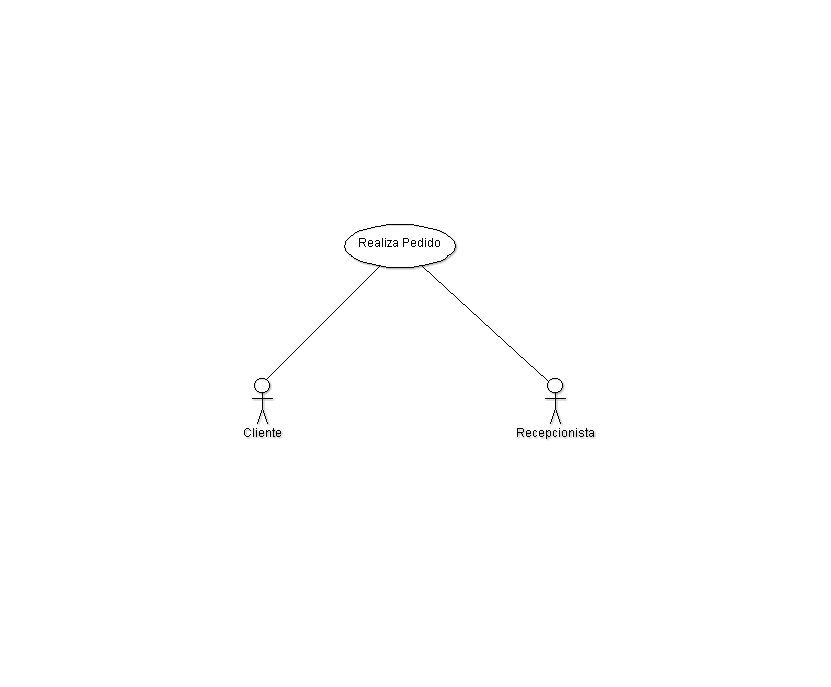
**Descripción Escrita del Modelado de negocio**

El cliente llama al negocio, la recepcionista atiende la llamada y le consulta sobre su necesidad, el cliente si no conoce el menú o lo que le gustaría consumir pregunta sobre el menú vigente, la recepcionista le dicta el menú, posteriormente el cliente elige según su preferencia, la recepcionista pregunta si está satisfecho con su pedido o si le gustaría agregar algo más, el cliente pide algo mas o solo confirma el pedido, la recepcionista solicita los datos personales del cliente como: nombre, número de teléfono y dirección, una vez obtenido los datos la recepcionista le comunica el monto total y pregunta si tiene cambio, el cliente le contesta y finaliza la llamada, el pedido es pasado de la recepción a la cocina en donde se prepara, luego de elaborar completamente el pedido, se le entrega al repartidor para enviarle al cliente su pedido y cobrar.

****

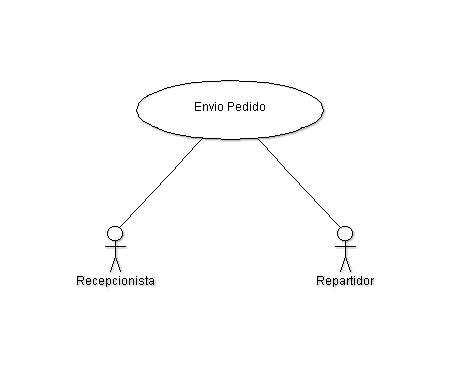
**Tabla de Caso de Uso**

|  |  |
| --- | --- |
| **Autores** | DeliverYapp. |
| **Actores:** | Cliente, Recepcionista, repartidor |
| **Fecha:** | 04-06-2016. |
| **Descripción:** | Proceso de Pedido de Delivery. |
| **Precondiciones** | El cliente debe contar con un teléfono, celular (con crédito) para la realización de llamada. |
| **Flujo normal** | |
| 1. El Cliente llama por teléfono al restaurante 2. La recepcionista contesta la llamada y pregunta que se le ofrece. 3. Si no conoce el menú le pregunta a la recepcionista. Si conoce el menú pasa al paso 6. 4. La recepcionista le dicta el menú. 5. El cliente realiza su pedido. 6. La recepcionista anota el pedido. 7. La recepcionista Pregunta si quiere agregar algo más a su pedido. 8. En caso de que el cliente diga “No” solo confirma su pedido. Si el cliente responde “Si” vuelve al paso 5. 9. La recepcionista solicita los datos personales al cliente(nombre, apellido, dirección, número teléfono) 10. La recepcionista comunica el tiempo estimado, monto total y pregunta si tiene cambio. 11. El cliente le contesta y finaliza la llamada. 12. La recepcionista pasa el pedido a la cocina 13. Una vez terminado el pedido, la recepcionista le entrega al repartidor para llevarle el pedido al cliente. 14. El repartidor entrega el pedido al cliente y cobra. | |
| **Flujo alternativo** | |
| 1. Si la recepcionista no contesta, el cliente vuelve a llamar.   En el caso de que se corte la llamada en cualquiera de los ítems del flujo normal se le devuelve la llamada.  En el caso de que el cliente vuelve a llamar para cancelar el pedido, este se da de baja.  13 En caso de que el pedido no llega, el cliente vuelve a llamar y el recepcionista busca la mejor solución. | |
| **Poscondiciones:** Se completa el servicio de delivery. | |

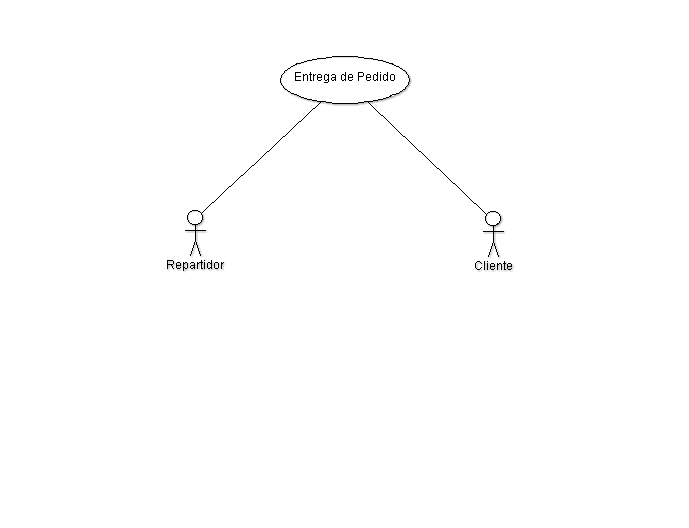
1.1 Realización de pedido

|  |  |
| --- | --- |
| **Autores** | DeliverYapp. |
| **Actores:** | Cliente, Recepcionista |
| **Fecha:** | 02-08-2016. |
| **Descripción:** | El cliente realiza el pedido. |
| **Precondiciones** | El cliente debe contar con un teléfono, celular (con crédito) para realizar la llamada. |
| **Flujo normal** | |
| 1. El cliente llama por teléfono al restaurante 2. La recepcionista contesta la llamada y pregunta que se le ofrece. 3. Si no conoce el menú le pregunta a la recepcionista. Si conoce el menú pasa al paso 6. 4. La recepcionista le dicta el menú. 5. El cliente realiza su pedido. 6. La recepcionista anota el pedido. 7. La recepcionista Pregunta si quiere agregar algo más a su pedido. 8. En caso de que el cliente diga “No” solo confirma su pedido. Si el cliente responde “Si” vuelve al paso 5. 9. La recepcionista solicita los datos personales al cliente(nombre, apellido, dirección, número teléfono) 10. La recepcionista comunica el tiempo estimado, monto total y pregunta si tiene cambio. 11. El cliente le contesta y finaliza la llamada. | |
| **Flujo alternativo** | |
| * 1. Si la recepcionista no contesta, el cliente vuelve a llamar.   2. Si la recepcionista no contesta, el cliente pasa por el local. | |
| **Postcondiciones:** Se generó pedido de delivery. | |

1.2 Envió de pedido



|  |  |
| --- | --- |
| **Autores** | DeliverYapp. |
| **Actores:** | Recepcionista, repartidor |
| **Fecha:** | 02-08-2016. |
| **Descripción:** | Proceso de envió de Delivery. |
| **Precondiciones** | La recepcionista debe contar con un pedido |
| **Flujo normal** | |
| 1. La recepcionista le entrega al repartidor el pedido, el monto a cobrar y la ubicación del cliente.   2 - El repartidor hace la entrega del pedido | |
| **Flujo alternativo** | |
| 1No se realiza la entrega de pedido | |
| **Postcondiciones:** Envio de pedido delivery | |

1.3 Entrega de Pedido

|  |  |
| --- | --- |
| **Autores** | DeliverYapp. |
| **Actores:** | Cliente, repartidor |
| **Fecha:** | 02-08-2016. |
| **Descripción:** | Proceso de entrega de Delivery. |
| **Precondiciones** | El cliente debe haber hecho un pedido. |
| **Flujo normal**   1. El repartidor recibe el pedido y se dirige a la ubicación indicada. 2. El repartidor llega a la ubicación indicada y hace la entrega del pedido al cliente. 3. El cliente le paga al repartidor. 4. El repartidor se retira. | |
| **Flujo alternativo** | |
|  | |
| **Postcondiciones:** Se hizo la entrega del pedido. | |

# Capítulo II

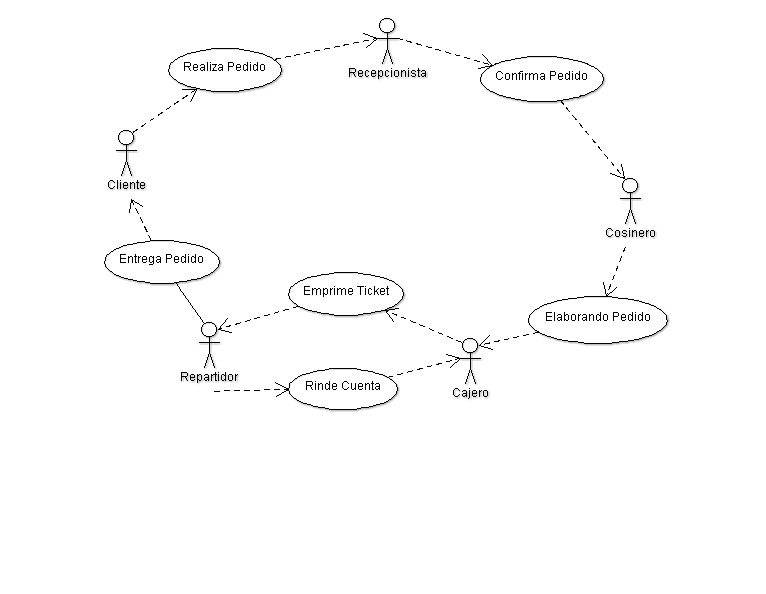
## Modelado de Requerimiento del Sistema

### 

### Descripción Escrita del modelado del Sistema

El cliente desde su navegador ingresa a la aplicación “DeliverYApp”, primeramente se debe registrar (Loguearse) con los datos requeridos por el sistema, luego de eso podrá ver un mensaje de bienvenida y una lista de restaurantes en el cual tendrá la opción de elegir uno de ellos, una vez seleccionado el restaurante se desplegará una interfaz con las opciones del menú donde el cliente por medio de imágenes con sus respectivas descripciones optara por su plato de preferencia. Realizado esto aparecerán los datos personales con la lista de pedido que ha hecho el cliente con sus respectivos precios y el monto total a abonar. Si el cliente está de acuerdo, presiona el botón para confirmar su pedido y se mostrará un mensaje de envío correcto. Luego, la recepcionista recibe el pedido y lo verifica (si los datos son coherentes y correctos), en caso de que no se cumplan los requisitos el sistema notificara al cliente que su pedido fue rechazado y que lo intente nuevamente, al confirmar que el pedido cumple los requisitos, se notificará al cliente que su pedido fue aceptado, este ingresara a la cocina con estado pendiente, culminado el menú solicitado, pasa al cajero y cambia el estado a enviado imprimiendo un ticket con los datos del cliente necesarios para el envío, al retornar el repartidor rinde cuenta de los pedidos entregados el cajero ingresa el pago actualizando el estado a cobrado.

**Grafico**



**Tabla de Caso de Uso**

|  |  |
| --- | --- |
| **Autores** | Grupo DeliverYApp |
| **Actores:** | Cliente, Recepcionista, Repartidor |
| **Fecha:** | 26-05-2016 |
| **Descripción:** | Permite hacer el pedido de su restaurant preferido. |
| **Precondiciones** | El Cliente debe contar con un dispositivo con acceso a internet (celulares, computadoras, etc.) |
| **Flujo normal** | |
| 1. El cliente ingresa a la aplicación de DeliverYApp se registra, en caso de que este registrado se loguea. 2. La aplicación le da la bienvenida y la opción de elegir el restaurante. 3. Una vez que el cliente elige el restaurante tiene las opciones de Menú, Bebidas, Ofertas del día, Combo etc. 4. El cliente elige unas de las opciones y se muestran los detalles (fotos, precios, etc.) 5. El cliente selecciona el menú que desea, una vez que esté satisfecho con el pedido ,presiona la opción de realizar pedido 6. En el Sistema aparecerá la opción de elegir si el pedido realizado se va a enviar o se va a ir a buscar, ya seleccionado una de estas opciones aparece los datos personales y la lista de pedido que el cliente ha seleccionado y el monto total. 7. Si el cliente está de acuerdo, presiona el botón “Confirmar Pedido”, en caso de que no esté de acuerdo, Cancela el pedido. 8. Se muestra un mensaje de confirmación “solicitud de pedido DeliveYApp Enviado “ 9. La recepcionista recibe la notificación de pedido del cliente y verifica los datos, una vez confirmado pasa el pedido a la cocina con el estado pendiente. 10. Culminado el menú solicitado, pasa a la recepcionista y a su vez cambia el estado ha enviado, imprimiendo un ticket con los datos del cliente, para que el repetidor realice el envío. 11. Al retornar, el repartidor rinde cuenta de los pedidos entregados, la recepcionista ingresa el pago actualizando el estado ha cobrado. | |
| **Flujo alternativo** | |
| 10- En caso de que no se haya impreso el ticket, se ingresa a la opción de imprimir el ticket de pedido en cuestión.  Si el cliente se queda sin acceso a internet, no podrá realizar su pedido | |
| **Precondiciones: Se realizó y se entregó el pedido de menú con éxito.** | |

1.1 Asignación de roles

|  |  |
| --- | --- |
| **Nombre** | Asignación de roles |
| **Autores:** | Administrador, Sistema |
| **Fecha:** | 09-06-2016 |
| **Descripción:** | Asigna los roles a los usuarios del sistema. |
| **Precondiciones** | Debe estar registrado |
| **Flujo normal** | |
| 1. El administrador se loguea al sistema. 2. El sistema le da la bienvenida con el menú principal. 3. El administrador ingresa al módulo de empleados, ingresa los datos personales, le asigna uno de los roles y lo guarda. | |
| **Flujo alternativo** | |
| 1. En caso de no contar con el rol de Administrador deberá solicitarlo. | |
| **Poscondición: Se ha asignado un nuevo rol al empleado.** | |

1.2 Creación de menú

|  |  |
| --- | --- |
| **Nombre** | Creación de menú |
| **Autores:** | Usuario, Sistema |
| **Fecha:** | 09-06-2016 |
| **Descripción:** | Se registran los menús ofrecidos por la empresa. |
| **Precondiciones** | Estar registrado al sistema. |
| **Flujo normal** | |
| 1. El usuario se loguea al sistema. 2. Ingresa al módulo de crear menú. 3. Se muestra el formulario con los campos a rellenar. 4. Una vez rellenado el formulario tiene la opción de agregar más. 5. En caso de que no tenga que agregar más tiene la opción de guardar. 6. Si por algún motivo relleno mal tiene la opción de eliminar o cancelar. | |
| **Flujo alternativo** | |
| 1. Si se cancela vuelve al paso 4 | |
| **Precondiciones: Se crea un nuevo menú del restaurante.** | |

# Capítulo III

**Estudio de Viabilidad**

* **Técnica**

Se ha evaluado la opción de escoger el lenguaje de programación de Ruby on Rails (RoR) por el hecho de que la mayoría de los integrantes del grupo poseen un conocimiento básico, dispuestos en aprender e innovar y además porque ofrece una inmensa librerías (gemas) que nos facilitara muchas cosas en el momento de desarrollar el sistema.

* **Económicamente**

El desarrollo no representa un costo económico elevado, teniendo en cuenta que tenemos entorno de desarrollo gratuito y los demás gastos serán absorbidos por los alumnos.

* **Operativamente**

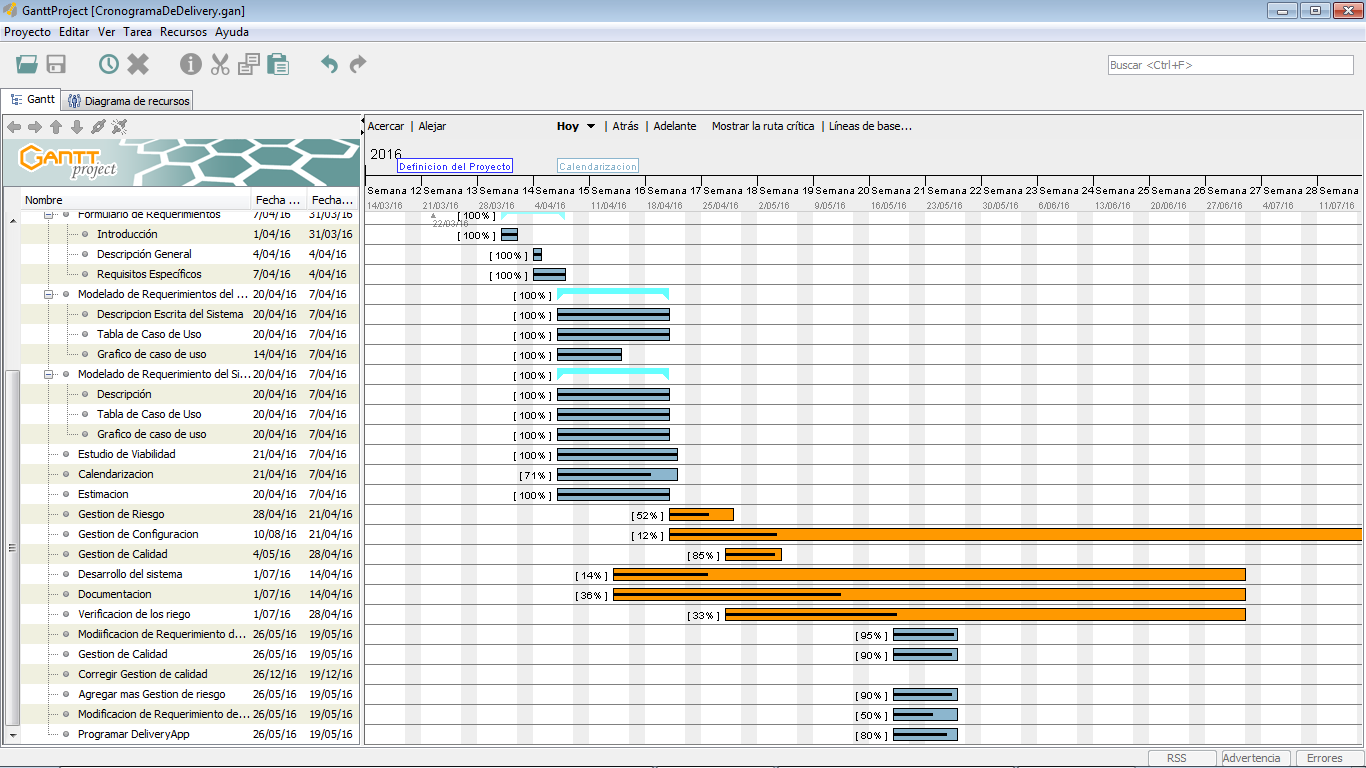
Cuenta con los recursos humanos preparados y la infraestructura necesarias para desarrollar dicho sistema.

De acuerdo al análisis realizado hemos comprobado que el sistema de “Delivery” a desarrollar es viable por el hecho de que les facilita a los usuarios realizar sus pedidos de forma más rápida e intuitiva.

# Capítulo IV

## Calendarización





# Capítulo V

**Estimación de Costos**

**COCOMO2**

El Modelo Constructivo de Costos (COCOMO, por su acrónimo del inglés COnstructive COst MOdel)

COCOMO ll es un modelo (algoritmo) que permite estimar el coste, esfuerzo y tiempo   
cuando se planifica una nueva actividad de desarrollo software.

El CocoMo surgió para medir y calcular el coste y el tiempo de un determinado proyecto basándose fundamentalmente en las líneas de código y algunas constantes.

Surge como una alternativa para incluir componentes de incerteza en las estimaciones, conforme al nivel de información disponible. Este es un modelo paramétrico que establece ecuaciones matemáticas para describir las relaciones entre el tamaño del software - factor primario de costo usualmente representado en términos de puntos de función - y otros factores secundarios que buscan capturar particularidades de producto, proceso, personas y plataforma.

EI primer modelo CocoMo se publicó en 1981 por Barry Boehm. En los 15 años siguientes las técnicas de desarrollo de software cambiaron drásticamente. A raíz de esto surgió el CocoMo II, publicado por primera vez en 1997.

**Características**

* Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.
* Representa el más extenso modelo empírico para la estimación de [software](http://www.ecured.cu/Software).
* Existen herramientas automáticas que estiman costos basados en COCOMO como ser: Costar, COCOMO 81.

**Objetivos para la construcción de COCOMO II**

* Desarrollar un modelo de estimación de costo y cronograma de proyectos de [software](http://www.ecured.cu/Software) que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
* Construir una [base de datos](http://www.ecured.cu/Base_de_datos) de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
* Implementar una herramienta de [software](http://www.ecured.cu/Software) que soportara el modelo.
* Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

**Formula**

E=a(KI)° · m(X) donde:

"E" es Salario/mes (Media).

"a" y "b" son constantes según el modo (Orgánico, Semi libre o Rígido).

"KI" es la cantidad de líneas de código (En miles).

"m(X)" es el multiplicador que depende de 15 atributos constantes.

**Modelos de COCOMO II**

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son:

* Modelo de composición de aplicación. Utilizado durante las primeras etapas de la [Ingeniería del software](http://www.ecured.cu/Ingenier%C3%ADa_de_software), donde el prototipado de las interfaces de usuario, la interacción del sistema y del software, la evaluación del rendimiento, y la evaluación de la madurez de la tecnología son de suma importancia.
* Modelo de fase de diseño previo. Utilizado una vez que se han estabilizado los requisitos y que se ha establecido la arquitectura básica del software.
* Modelo de fase posterior a la arquitectura. Utilizado durante la construcción del software.

### Ventajas

* Es fácil de realizar y de interpretar.
* Tiene pocas variables.
* Se acerca a la realidad en la mayoría de los casos.

**Desventajas**

* No saca resultados fiables en proyectos demasiado pequeños.
* La elección de las variables es muy subjetiva y depende de la persona que realiza el estudio.

**Conclusión**

Cocomo 2 es una buena opción para proyectos medianos/grandes aunque para su realización es necesario conocer cuál va a ser el equipo de desarrolladores, el equipo Informático, el entorno etc.

# Capítulo VI

## Plan de riesgos

### Introducción

En la vida de un proyecto informático pueden provocarse errores por causas humanas, por tardar más en hacer una actividad que creíamos bien estimada, y por no terminar a tiempo un elemento que retrasa una integración con otros elementos, etc. El análisis de riesgo puede adelantarse a algunos problemas, y afirmarnos que si se produce su impacto en los objetivos de proyectos será menor posible.

El Plan de riesgo es una parte del requerimiento del sistema en el cual se plasman los riesgos y posibles soluciones a implementar si estas llegan a ocurrir en el transcurso de la elaboración del proyecto. Se obtiene una lista de los posibles riesgos, luego estas se clasifican y se separan de acuerdo al tipo.

### Propósito

El propósito del Plan de Riesgos es detectar, como dice ya el titulo los futuros riesgos que se podrían presentar durante la elaboración del proyecto y hacer de estas una lista para luego analizar y sacar las posibles soluciones para luego implementarlas si se da ocasión.

### Alcance

El alcance de este documento es fijar un plan de políticas a seguir por parte de los encargados del proyecto, para tener en cuenta ante la aparición de un problema. El proyecto DeliverYapp establece un plan para administrar los productos de trabajo del proyecto, incluyendo tanto los entregables de software como la documentación del proyecto.

**Gestión de Riesgos**

**Identificación de Riesgos**

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Tipo de riesgo** | **Descripción** |
| Rotación de personal | Proyecto y producto | Personal con experiencia abandona el proyecto antes de que finalice |
| Cambios de requisitos | Proyecto y producto | Existencia de más cambios de requerimientos de los previstos inicialmente |
| Retrasos en la especificación | Proyecto y producto | Retrasos en las especificaciones de interfaces esenciales |
| Subestimación del tamaño | Proyecto y producto | El tamaño del requisito (la ERS, del proceso de IR) se ha subestimado |
| Bajo rendimiento de la herramienta CASE | Producto | Las [herramientas CASE](http://www.monografias.com/trabajos14/herramicase/herramicase.shtml) que ayudan al proyecto no tienen el rendimiento y las funcionalidades esperadas |

### Análisis de Riesgos

|  |  |
| --- | --- |
| **Tipo de riesgo** | **Posibles riesgos** |
| Personal | Imposible contratar personal con los conocimientos requeridos. |
| Organizativos | La organización se reestructura y una nueva [administración](http://www.monografias.com/trabajos36/administracion-y-gerencia/administracion-y-gerencia.shtml) se responsabiliza del proyecto. |
| Herramientas | Las distintas [herramientas](http://www.monografias.com/trabajos11/contrest/contrest.shtml) CASE no están disponibles |
| Requerimientos | Cambios de requerimientos que precisan modificaciones en el [diseño](http://www.monografias.com/trabajos13/diseprod/diseprod.shtml). |
| Estimación | El tamaño del [sistema](http://www.monografias.com/trabajos11/teosis/teosis.shtml) a desarrollar está subestimado. |

**Riegos por efectos**

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Probabilidad** | **Efectos** |
| Problemas financieros de la organización reducen el presupuesto del proyecto | baja | Catastrófico |
| Imposible contratar personal con los conocimientos requeridos | alta | Catastrófico |
| Personal clave enfermo o no disponible en momentos críticos | moderada | Serio |
| Cambios de requerimientos que precisan modificaciones en la codificación | moderada | Serio |
| El tiempo requerido para desarrollar el proceso de IR está subestimado | alta | Serio |
| Los clientes no comprenden el impacto de los cambios en los requerimientos | moderada | Tolerable |

**Estrategias por riesgos.**

|  |  |
| --- | --- |
| **Riesgo** | **Estrategia** |
| Problemas financieros de la organización | Preparar un documento breve para la [dirección](http://www.monografias.com/trabajos15/direccion/direccion.shtml) de la [empresa](http://www.monografias.com/trabajos11/empre/empre.shtml) que [muestra](http://www.monografias.com/trabajos11/tebas/tebas.shtml) que el proyecto hace contribuciones muy importantes a las metas del negocio |
| Problemas de reclutamiento | Organizar cursos de [capacitación](http://www.monografias.com/trabajos/adpreclu/adpreclu.shtml) para el personal ya existente, investigar la posibilidad de contratar en otras regiones del país |
| Enfermedad del personal | reorganizar el equipo de tal forma que se solapen el trabajo y los miembros comprendan [el trabajo](http://www.monografias.com/trabajos/fintrabajo/fintrabajo.shtml) de los demás |
| Cambios en los requisitos | Rastrear la información para valorar el impacto de los requerimientos, maximizar la información oculta en ellos |
| Tiempo de IR subestimado | Alertar al cliente de las dificultades potenciales y las posibilidades de retraso |

# Capítulo VII

Plan de Gestión de Configuración

Introducción

Planificación de la configuración:

Se describen las actividades de gestión de configuración de software que deben ser ejecutadas durante el proceso de desarrollo del proyecto. Se definen tanto los proyectos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

**Definición de la Línea Base:**

Informar sobre la verificación realizada sobre los objetos de configuración que componen la versión actual para asegurar que se encuentran en estado consistente en la Línea Base del Proyecto.

**Seguimiento de la Línea Base**

Recibir, registrar y mantener todos los productos recibidos a través de todas sus versiones.

**Control de Cambios:**

Seguimiento de un ciclo de vida de un cambio.

**Informe Final de Configuración:**

Realizar el Informe Final de Configuración, el cual contiene un resumen de las actividades realizadas de SCM y sus resultados a lo largo del proyecto.

**Producir la Versión de Producto a Liberar:**

Realizar el Informe Final de Configuración, el cual contiene un resumen de las actividades realizadas de SCM y sus resultados a lo largo del proyecto.

Propósito

Este documento describe las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Aquí se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

Alcance

El ámbito de este documento es el proyecto DeliverYapp y establece un plan para administrar los productos de trabajo del proyecto, incluyendo tanto los entregables de software como la documentación del proyecto.

Gestión de Configuración

Las responsabilidades son:

* Planificación de Configuración
* Definir la Línea Base
* Seguimiento de la Línea Base
* Control de Cambios
* Realización del Informe Final de Configuración
* Producir la Versión de Producto a Liberar

Herramientas, Entorno e Infraestructura

Herramientas

Durante el proceso de gestión de configuración se utilizará la herramienta Github para el control de versiones del proyecto. Esta gestión se hará mediante la herramienta Zenhub para los documentos y Sublime para el código fuente.

El control de cambios, así como la gestión de defectos, se llevará a cabo mediante la plantilla “Solicitud de Cambio”, permitiendo al equipo de desarrollo ir detallando los defectos encontrados para que la persona responsable pueda corregirlos.

Programa de la Gestión de Configuración

Identificación de la Configuración

Elementos de Configuración

Para este proyecto los elementos de configuración se corresponderán con los entregables definidos en el Modelo de Proceso, aunque no necesariamente todos los entregables deben ser elementos de configuración.

La decisión de cuál de los entregables serán elementos de configuración será tomada por el SCMR, quién deberá tomar en cuenta qué productos serán necesarios cuando se quiera recuperar una versión completa del sistema.

Se debe generar una línea base por iteración en cada Fase, de acuerdo a lo siguiente:

* Los eventos que dan origen a la línea base.
* Los elementos que serán controlados en la línea base.
* Los procedimientos usados para establecer y cambiar la línea base.
* La autorización requerida para aprobar cambios a los documentos de la línea base.

Nomenclatura de Elementos

Se especifica cómo se distinguirán las diferentes versiones de cada elemento.

Para todos los elementos de configuración se les deberá agregar, después del nombre del mismo, información acerca del grupo al que corresponde el elemento y la versión del mismo.

El formato para esta nomenclatura es: **NomenclaturaG**X**v**Y.extensión, donde:

· Nomenclatura es la especificada más abajo para cada elemento.

· X es un número de 1 dígito que identifica al grupo.

· Y indica la versión del elemento de configuración o entregable.

· Extensión indica la extensión del elemento de configuración o entregable.

Para los entregables, se deberá identificar a que Fase e iteración corresponden en forma manual. Se indica la siguiente nomenclatura para cada entregable en el modelo de proceso, según la disciplina.

**Requerimientos:**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| RQACT | Acta de Reunión de Requerimientos |
| RQDRQ | Especificación de Requerimientos |
| RQMOD | Modelo de Casos de Uso |
| RQRSU | Requerimientos Suplementarios |
| RQDVC | Documento de Validación con el Cliente |
| RQPIU | Pautas para Interfaces de Usuario |
| RQRCA | Requerimientos Candidatos |
| RQALS | Alcance del Sistema |
| RQGLO | Glosario |
| RQOOMDO | Modelo de Dominio |
| RQOODRP | Documento de Requerimientos para el Prototipo |
| RQGXNOM | Nomenclatura |

**Diseño:**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| DSMDI | Modelo de Diseño |
| DSARQ | Descripción de la Arquitectura |
| DSOOMDA | Modelo de Datos |
| DSOODDP | Documento de Diseño del Prototipo |

**Implementación:**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| IMEDT | Estándar de Documentación Técnica |
| IMEI | Estándar de Implementación |
| IMPR | Prototipo |
| IMIIN | Informe de Integración |
| IMDT | Documentación técnica |
| IMIVU | Informe de Verificación Unitaria |
| IMOOPII | Plan de Integración de la Iteración |
| IMOOMIM | Modelo de Implementación |
| IMOOEJI | Ejecutable de la Iteración |
| IMOORRP | Reporte de Revisión por Pares |
| IMOOCVU | Clases de la Verificación Unitaria de Módulo |
| IMGXICO | Informe de Consolidación |
| IMGXEST | BC Con Estilos |
| IMGXCON | BC Consolidado |
| IMGXNUC | BC Núcleo |
| IMGXMOD | BC Módulo |

**Verificación:**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| VRPVV | Plan de Verificación y Validación |
| VRDAP | Documento de Evaluación y Ajuste del Plan de V & V |
| VRPVI | Plan de Verificación de la Iteración |
| VRMCP | Modelo de Casos de Prueba |
| VRIVD | Informe de Verificación de Documento |
| VRIVI | Informe de Verificación de Integración |
| VRIVS | Informe de Verificación del Sistema |
| VRRPR | Reportes de Pruebas |
| VREV | Evaluación de la Verificación |
| VRIFV | Informe Final de Verificación |

**Implantación (IP):**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| IPMSU | Materiales para Soporte al Usuario  (Se pueden usar sufijos para identificar cada ítem dentro del material Ej. IPMSUMU para Manual de Usuario) |
| IPMCA | Materiales para Capacitación |
| IPPS | Presentación del Sistema |
| IPPLA | Plan de Implantación |
| IPVPR | Versión del Producto |
| IPOOEDU | Estándar de Documentación de Usuario |
| IPOORFPA | Reporte Final de Pruebas de Aceptación |

**Gestión de Configuración y Control de Cambios (SCM):**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| SCMPLA | Plan de Configuración |
| SCMMAC | Manejo del Ambiente Controlado |
| SCMGC | Gestión de Cambios |
| SCMRV | Registro de Versiones |
| SCMILB | Informe de la Línea Base del Proyecto |
| SCMIF | Informe Final de SCM |

**Gestión de Calidad (SQA):**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| SQAPLA | Plan de Calidad |
| SQADAP | Documento de Evaluación y Ajuste del Plan de Calidad |
| SQARTF | Informe de RTF |
| SQAES | Entrega Semanal de SQA |
| SQAIR | Informe de Revisión de SQA |
| SQADV | Descripción de la Versión |
| SQANV | Notas de la Versión |
| SQAIF | Informe Final de SQA |

**Gestión de Proyecto (GP):**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| GPPLA | Plan de Proyecto |
| GPISP | Informe de Situación del Proyecto |
| GPEM | Estimaciones y Mediciones |
| GPDRI | Documento de Riesgos |
| GPRAC | Registro de Actividades |
| GPIFP | Informe Final de Proyecto |
| GPARE | Acta de la Reunión de Equipo |
| GPPIT | Plan de la Iteración |
| GPPDE | Plan de Desarrollo |
| GPICF | Informe de Conclusiones de la Fase |
| GPPDP | Presentación al Director del Proyecto |
| GPARD | Acta de la Reunión con el Director del Proyecto |
| GPOODAP | Documento de Evaluación y Ajuste al Plan de Proyecto |
| GPIARI | Acta de la Reunión de Integración |

**Comunicación (COM):**

|  |  |
| --- | --- |
| Nomenclatura | Entregable |
| COMDI | Documento Informativo |
| COMENS | Encuesta de Satisfacción del Cliente |
| COMEVS | Evaluación de Satisfacción del Cliente |

Elementos de la Línea Base del Proyecto

N/A

Control de Configuración

El procedimiento que se describe a continuación es el que se utilizará cada vez que se precise introducir un cambio al sistema.

Se entiende por cambio al sistema, las modificaciones que afecten a la línea base del sistema, como pueden ser:

* Cambios en los Requerimientos.
* Cambios en el Diseño.
* Cambios en la Arquitectura.
* Cambios en las herramientas de desarrollo.
* Cambios en la documentación del proyecto.

Solicitud de Cambios

Cuando se realiza la solicitud de un cambio, se actualiza el documento de “Solicitud de cambio” para registrar esta solicitud.

Aprobación de Cambios

Se debe formar el “Comité de Control de Configuración” y determinar su autoridad para la aprobación de cambios.

La composición de este comité puede variar según el tipo de cambio y las líneas de trabajo involucradas en él.

Se sugieren como posibles integrantes:

* Administrador (obligatorio)
* Arquitecto (opcional)
* Analista (opcional)
* Implementador (opcional)
* SCM (obligatorio)
* Cliente (opcional)

Se define un comité de Control de Configuración de nivel superior, compuesto por el Gerente de proyecto, al cual se elevarán las solicitudes de cambios cuya aprobación o desaprobación no se pueda resolver por el primer comité.

Implementación de Cambios

Una vez realizada la evaluación del cambio, se decide en qué momento implementarlo. Esta etapa involucra los procesos necesarios para implementar la solicitud y monitorear el progreso del trabajo.

Además se especificará el momento de liberación del cambio; así como también los responsables de las actividades que involucra el cambio.

Estado de la Configuración

N/A

Informes y Auditorías

Cada cierto tiempo, el gestor de configuración y cambios realizará un informe para el jefe de proyecto con el fin de revisar la evolución de los defectos que se vayan registrando en el mismo.

En este informe (“Informe de Evolución de Defectos”) se detallará los defectos detectados, sus prioridades, responsables de corregirlos, su estado y el procedimiento que se ha seguido o se va a seguir a la hora de resolverlos.

Este nos permitirá obtener una serie de indicadores para determinar la calidad del producto que se está desarrollando, permitiendo al jefe de proyecto informar al cliente sobre este aspecto.

Las auditorías que se llevarán a cabo para comprobar si los cambios se han realizado correctamente serán al final de cada iteración, pero antes de que se cree una línea base. En éstas se revisarán tanto los requisitos funcionales y de rendimiento, como que el producto cumpla con las especificaciones detalladas en las que se define. Tomarán parte en éstas el cliente, el jefe de proyecto y el gestor de configuración.

Calendario

La definición de la línea base será determinada al comienzo de cada semana de la siguiente manera: https://github.com/francben/proyectoDeliberyApp#burndown?milestoneId=x, donde la X corresponde a la semana corriente.

Todavía no se cuenta con una implementación del Control de Cambios.

Capacitación y Recursos

Para el manejo de las versiones se utiliza Git.

Mantenimiento del Plan de Gestión de la Configuración

El responsable de monitorear el Plan de SCM es el responsable de SCM.

Para la frecuencia de las modificaciones no hay un tiempo estipulado, serán las necesarias, siempre limitadas al tiempo de finalización de la iteración.

Este Plan deberá ser revisado al inicio de cada fase, modificado de acuerdo a lo necesario, aprobado y distribuido al equipo de proyecto.

# Capítulo VIII

## Descripción de Roles

## Líder del Proyecto:

**Objetivos**:

* Coordinar el quipo
* Asegurar que todos cumplan con sus trabajos.

Responsabilidad:

* Desarrollo del plan del proyecto.
* Identificación de los requerimientos y el alcance del proyecto.
* Comunicación.
* Administración de los recursos humanos y materiales.
* Control de tiempos.
* Identificación y control de riesgos.
* Administración de los costos/presupuesto, el aseguramiento de la calidad.
* El reporte y evaluación del desempeño del proyecto.
* El líder de proyecto debe mantener su foco en asegurar que el proyecto se termine en el tiempo y presupuesto planeado, y muy frecuentemente con tiempos limitados.

## Documentador:

El objetivo principal del documentador es el de actuar como medio de comunicación entre los miembros del equipo.

* Mantener la información generada durante el proceso de desarrollo con un adecuado procesamiento que permita la calidad en el mantenimiento de la misma.
* Organizar y garantizar el almacenamiento y recuperación de la documentación de los procesos y productos más recientes durante el desarrollo, manteniendo así la información al día.
* Mantener la consistencia en la apariencia y estructura de los documentos, facilitando su almacenamiento, recuperación e intercambio, no permitiendo el almacenamiento de documentos con formatos diferentes al establecido.
* Generar plantillas que permitan una adecuada estructura y uniformidad en los documentos que lo requieran.
* Elaborar, almacenar y permitir la recuperación de la documentación que se genera como parte de la gestión del proyecto.
* Construir documentación de cara al usuario que contempla los aspectos que son de utilidad para los usuarios finales del sistema y proveer una adecuada capacitación.

La documentación es necesaria para informar a todos los miembros del equipo de desarrollo acerca del estado y evolución del proyecto.

## Desarrollador:

Debe convertir las especificaciones del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de software de apoyo a la programación.

* Creación del código ejecutable.
* Creación de la documentación de operación.
* Integración del entorno.
* Conducir revisiones del software.
* Realizar mantenimiento y actualizaciones del sistema existentes, las cuales serán solicitados a través del formulario de requerimiento de desarrollo.
* Realizar pruebas del programa desarrollado con los usuarios solicitantes.

## Tester:

El objetivo principal del tester es el de diseñar test que en forma sistemática, pueda eliminar diferentes casos de errores. Realizando esto con la mínima cantidad de tiempo y esfuerzo.

* Construir y aplicar los planes de prueba unitarios de modulo, de sistema, de aceptación parcial, manteniéndolos actualizados durante el proyecto.
* Velar por la amplitud y la exactitud de todos los documentos del proyecto.
* Velar por el estándar adoptado para el desarrollo
* Velar por la calidad del producto final.

## Gestor de Riesgo:

* Documentos de requisitos de Usuario y Software.
* Plan de administración del proyecto.
* Plan de testeo.
* Fase de diseño arquitectónico.
* Fase de diseño detallado.
* Políticas de control de cambios, control de errores y control de la configuración.
* Documentación.
* Descubrir errores en funciones, lógica e implementación en cualquiera de las representaciones del software.
* Verificar que el software bajo revisión cumple con los requisitos.
* Asegurarse que el software ha sido representado de acuerdo al estándar en uso.
* Alcanzar software que es desarrollado en forma uniforme.
* Hacer el proyecto más manejable.

## Gestor de Configuración:

* Gestionar la planificación, identificación, control, seguimiento y auditoría de todos los elementos de configuración en la base de datos de configuración.
* Desarrollar el plan de gestión de configuración.
* Promover el uso efectivo de la CMDB.( configuration management database)
* Monitorizar y reportar los cambios no autorizados sobre los CIs. (Configuration Items)
* Asegurar la consistencia e integridad de los datos de la CMDB a través de la ejecución de procedimientos de verificación y auditoría.
* Revisar tipos de elementos de configuración, relaciones, atributos y valores asociados, estructura de la base de datos, derechos de acceso.
* Aprobar cambios estructurales en la CMDB.
* Coordinador de configuración.
* Reportar cualquier discrepancia o no conformidad en los CIs al gestor de configuración.
* Participar en la mejora continua del proceso de gestión de configuración.

# Capítulo IV

Plan de SQA

Propósito

Esta sección debe contener el propósito y alcance del Plan de Calidad.

Lista los ítems del software cubiertos por el Plan y el uso que se le dará al software en cuestión.

Se especifica la porción del ciclo de vida del software cubierta por el Plan.

La tarea fundamental del Plan de Calidad es exigir que se cumplan todas las normas y estándares establecidos para asegurar el buen fin del proyecto. Es muy importante lograr el entendimiento de la tarea de SQA por parte de todos y generar un compromiso de aceptación por los errores que se detecten.

El mismo debe presentar un marco básico organizacional que contenga lo siguiente:

* Prácticas de Aseguramiento de la calidad: Adecuadas herramientas de desarrollo, técnicas, métodos y estándares, definidos y disponibles para realizar las revisiones.
* Software para la evaluación del plan de proyecto.
* Evaluación de requerimientos: Si consideramos que los productos de calidad son raramente desarrollados en base a requerimientos que no contemplen las necesidades de sus clientes, los requerimientos iníciales tienen que ser revisados para ver si reflejan fiel y completamente las necesidades del cliente.
* Evaluación del diseño: Se debe verificar que se cumpla con los requerimientos y que siga la metodología establecida.
* Evaluación de la codificación: Controlar que se cumpla con los estándares de codificación y evaluar la correctitud técnica del producto.
* Evaluación de los procesos de integración y pruebas: Controlar que se esté cumpliendo con el Plan de Testing.
* Recolección de métricas.

Referencias

[ANSI/IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance

Documento Plantilla gestión de riesgos

Documento plan de gestión de riesgos

Documento plantilla de revisión de SQA

Documento Informe Final de SQA

Documento plantilla de Revisión técnica formal

SQuaRE, ISO 25000:2005,

Quality management systems – Requirements ISO 9001:2008]

Gestión

Se debe especificar la organización, actividades y responsables.

Organización

Distinguir las estructuras dentro de la organización que tienen influencia y controlan la calidad del software.

Descripción de las dependencias o independencias de las estructuras antes mencionadas con respecto a los responsables del desarrollo del software.

Actividades

Ciclo de vida del software cubierto por el Plan

Esta sección debe contener las actividades más importantes del ciclo de vida del software que cubre el Plan.

Actividades de calidad a realizarse

Las tareas a ser llevadas a cabo deberán reflejar las evaluaciones a realizar, los estándares a seguir, los productos a revisar, los procedimientos para la elaboración de los distintos productos y los procedimientos para informar de los defectos detectados a sus responsables y realizar el seguimiento de los mismos hasta su corrección.

Las actividades que se realizarán son:

Revisar cada producto

Revisar el ajuste al proceso

Realizar Revisión Técnica Formal (RTF)

Asegurar que las desviaciones son documentadas.

Revisar cada producto

En esta actividad se revisan los productos que se definieron como claves para verificar en el Plan de calidad.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en esta revisión. Se revisan los productos contra los estándares, utilizando la checklist definida para el producto.

Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión de SQA, este informe debe ser distribuido a los responsables del producto y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

Revisar el ajuste al proceso

En esta actividad se revisan los productos que de definieron como claves para verificar el cumplimiento de las actividades definidas en el proceso. Con el fin de asegurar la calidad en el producto final del desarrollo, se deben llevar a cabo revisiones sobre los productos durante todo el ciclo de vida del software.

Se debe recoger la información necesaria de cada producto, buscando hacia atrás los productos previos que deberían haberse generado, para poder establecer los criterios de revisión y evaluar si el producto cumple con las especificaciones.

Esta información se obtiene de los siguientes documentos:

Plan del Proyecto

Plan de la iteración

Plan de Verificación Antes de comenzar, se debe verificar en los informes de revisión previos que todas las desviaciones fueron corregidas, si no fuese así, las faltantes se incluyen para ser evaluadas.

Como salida se obtiene el Informe de revisión de ajuste al Proceso, este informe debe ser distribuido a los responsables de las actividades y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

Realizar Revisión Técnica Formal (RTF)

El objetivo de la RTF es descubrir errores en la función, la lógica o la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas. Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el responsable de SQA e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

La duración de la reunión no debe ser mayor a dos horas.

Como salida se obtiene el Informe de RTF.

Asegurar que las desviaciones son documentadas

Las desviaciones encontradas en las actividades y en los productos deben ser documentadas y manejadas de acuerdo a un procedimiento establecido.

Se debe chequear que los responsables de cada plan los modifiquen cada vez que sea necesario, basados en las desviaciones encontradas.

Relaciones entre las actividades de SQA y la planificación

|  |  |
| --- | --- |
| **Actividad** | **Semana** |
| Actividad 1 | Semana cuando se realiza |
| Actividad 2 | Semana cuando se realiza |

Responsables

Identificar los distintos responsables de cada actividad identificada. Dichas actividades son: las revisiones, el análisis causal, el mantener una base de datos de los errores encontrados a lo largo del desarrollo e informes.

Para la puesta en marcha de estas actividades se deberá seguir el siguiente ciclo de prevención:

* Ejecutar una tarea
* Realizar un control de revisiones, para decidir la aceptación o necesidad de corrección de dicha tarea.
* En caso de que en la revisión se presenten errores se realizara un análisis causal para determinar el motivo de estos. Se analiza un determinado error, se establece una hipótesis de su posible causa, se trata de deducir en qué momento se produjo y por qué. Luego se deberá realizar la corrección del mismo y tomar una acción correctiva con el fin de eliminar la causa del problema.
* El resultado del análisis causal es ingresado a una base de datos para mantener un registro y poder obtener métricas.
* Se comienza nuevamente el ciclo ejecutando la tarea.

Documentación

Propósito

Identificación de la documentación relativa a desarrollo, Verificación y Validación, uso y mantenimiento del software.

Establecer como los documentos van a ser revisados para chequear consistencia: se confirman criterio e identificación de las revisiones.

Documentación mínima requerida

Esta busca asegurar que la implementación logrará satisfacer los requerimientos.

Especificación de requerimientos del software

El documento de especificación de requerimientos deberá describir, de forma clara y precisa, cada uno de los requerimientos esenciales del software además de las interfaces externas.

El cliente deberá obtener como resultado del proyecto una especificación adecuada a sus necesidades en el área de alcance del proyecto, de acuerdo al compromiso inicial del trabajo y a los cambios que este haya sufrido a lo largo del proyecto, que cubra aquellos aspectos que se haya acordado detallar con el cliente.

La especificación debe:

Ser completa:

Externa, respecto al alcance acordado.

Internamente, no deben existir elementos sin especificar.

Ser consistente, no puede haber elementos contradictorios.

Ser no ambigua, todo término referido al área de aplicación debe estar definido en un glosario.

Ser verificable, debe ser posible verificar siguiendo un método definido, si el producto final cumple o no con cada requerimiento.

Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.

Incluir requerimientos de calidad del producto a construir.

Los requerimientos de calidad del producto a construir son considerados dentro de atributos específicos del software que tienen incidencia sobre la ‘calidad en el uso’.

Funcionalidad

Adecuación a las necesidades

Precisión de los resultados

Interoperabilidad

Seguridad de los datos

Confiabilidad

Madurez

Tolerancia a faltas

Recuperabilidad

Usabilidad

Comprensible

Aprendible

Operable

Atractivo

Eficiencia

Comportamiento respecto al tiempo (Ver si aplica)

Utilización de recursos

Mantenibilidad

Analizable

Modificable

Estable, no se producen efectos inesperados luego de modificaciones

Verificable

Portabilidad

Adaptable (Ver si aplica)

Instalable

Co-existencia

Reemplazante (Ver si aplica)

Cada uno de estos atributos debe cumplir con las normas y regulaciones aplicables a cada uno.

Descripción del diseño del software

El documento de diseño especifica como el software será construido para satisfacer los requerimientos.

Deberá describir los componentes y subcomponentes del diseño del software, incluyendo interfaces internas. Este documento deberá ser elaborado primero como Preliminar y luego será gradualmente extendido hasta llegar a obtener el Detallado.

El cliente deberá obtener como resultado del proyecto el diseño de un producto de software que cubra aquellos aspectos que se haya acordado con el cliente incorporar al diseño, en función de la importancia que estos presenten y de sus conexiones lógicas.

El diseño debe:

Corresponder a los requerimientos a incorporar:

Todo elemento del diseño debe contribuir a algún requerimiento

La implementación de todo requerimiento a incorporar debe estar contemplada en por lo menos un elemento del diseño.

Ser consistente con la calidad del producto

Plan de Verificación y Validación

El Plan de V y V deberá identificar y describir los métodos a ser utilizados en:

La verificación de que:

a. Los requerimientos descritos en el documento de requerimientos han sido aprobados por una autoridad apropiada.

b. Los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.

c. El diseño expresado en el documento de diseño esta implementado en código.

Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en el documento de requerimientos.

Documentación de usuario

La documentación de usuario debe especificar y describir los datos y entradas de control requeridos, así como la secuencia de entradas, opciones, limitaciones de programa y otros ítems necesarios para la ejecución exitosa del software.

Todos los errores deben ser identificados y las acciones correctivas descritas.

Como resultado del proyecto el cliente obtendrá una documentación para el usuario de acuerdo a los requerimientos específicos del proyecto.

Plan de Gestión de configuración

El Plan de gestión de configuración debe contener métodos para identificar componentes de software, control e implementación de cambios, y registro y reporte del estado de los cambios implementados.

La Gestión de Configuraciones permite controlar el sistema como producto global a lo largo de su creación, obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores durante el mismo, lo que se traduce en un aumento de calidad del proceso de desarrollo y de mejora de la productividad en la organización.

La gestión de configuración facilita además el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo.

Propósito

Controlar la entrega y el cambio de los elementos a través del ciclo de vida del sistema.

Almacenar el estado de los elementos y de las peticiones de cambio.

Resumen

La Gestión de Configuración, en resumen, identifica los elementos de un proyecto de desarrollo de software (especificaciones, requisitos, arquitecturas, código, planes, etc.) proporcionando el control de los elementos identificados y la generación de informes de estado de la configuración, consiguiendo, al mismo tiempo, claridad de gestión, al asignar responsabilidades al personal encargado de las tareas de control a lo largo del ciclo de vida del producto.

Organización, Responsabilidades

Se designará a un integrante del grupo para la administración de gestión de versiones, el cual se encargará de administrar y dar los permisos en el gestor. Pudiendo cualquier integrante solicitarle al grupo algún cambio para que el mismo estudie su necesidad.

Herramientas, Entorno, e Infraestructura

Se utilizara la herramienta de Gestión de Configuraciones (CGS) Subversion y TortoiseSVN. Este maneja ficheros y directorios a lo largo del ciclo de vida del proyecto. Los ficheros se almacenan en un repositorio central, recordando todos los cambios que se hayan realizado, permitiendo a los integrantes del grupo poder recuperar versiones anteriormente guardadas, examinar la historia de cuando y como fueron modificados los datos, quien hizo los mismos y así poder coordinar el trabajo.

Siendo la misma especialmente útil para los documentos revisados frecuentemente, como el código fuente, la documentación, etc., como así también llevar un balance histórico de las diferentes versiones del sistema.

Forma de trabajo

Durante el proceso de gestión de configuración se utilizará la herramienta Subversion para el control de versiones del producto. Cuando algún miembro haga una modificación en el proyecto, deberá acceder al servidor donde está alojada esta aplicación para almacenar la parte modificada en él, teniendo el resto del equipo de desarrollo la última versión actualizada en dicho servidor. Esta gestión de acceso al servidor para la actualización se hará mediante la herramienta Tortoise para los documentos y el plugin Subclise para el código fuente.

Control de Cambios

Se efectúa una solicitud de cambio utilizando el [Formulario de Pedido y Detección de Cambio](#Formulario). Especifica los procedimientos para solicitar un cambio a una línea base y la documentación necesaria.

El mismo contiene:

Nombre y versión del Elemento de Configuración de Software a cambiar.

Nombre del peticionario.

Fecha de petición

Necesidad del cambio

Descripción del cambio pedido

Prioridad

Estado

Fecha del cambio

Reportes y Auditorias

Se realizará las siguientes auditorias:  
 **Auditoria Funcional:** Cuyo objetivo es comprobar que se han completado todas las pruebas necesarias para el / los ECS auditados, y que, teniendo en cuenta los resultados de los tests, se puede afirmar que el / los ECS satisfacen los requisitos que se impusieron sobre él.

**Revisión formal de certificación**: Cuyo objetivo es certificar que el / los ECS se comportan correctamente en su entorno operativo.

Otros documentos

Esta sección puede contener otros documentos que se identifiquen de incidencia en la calidad del producto a desarrollar, por ejemplo:

* Plan de desarrollo
* Plan de proyecto
* Manual de estándares y procedimientos.

Estándares, prácticas, convenciones y métricas

Esta sección deberá cumplir con las siguientes funciones:

Identificar los estándares, prácticas, convenciones y métricas que serán aplicadas.

Indicar como será monitoreado y asegurado el cumplimiento con estos ítems

El IEEE “Standard Glosary of Software Engering Terms” define como métrica: “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”.

Las métricas son una herramienta poderosa y fundamental para el trabajo en SQA. Su aporte fundamental son las medidas preventivas que pueden surgir a raíz de su estudio. Sin duda aportan conclusiones que muchas veces no se aprecian a simple vista y que ayudan a mejorar la eficiencia del grupo de trabajo y la calidad de los productos. Aportan un caudal de información para hacer controles estadísticos de la calidad. Además cabe resaltar que nunca debe dejarse de buscar nuevas métricas de acuerdo a las nuevas variaciones y tendencias de las estadísticas.

Objetivos

Existen dos objetivos importantes que se persiguen dentro del programa de métricas:

Documentar las metas a la hora de establecer un programa de métricas. Esto tiene sentido a la hora de decidir exactamente qué debe lograrse antes de gastar recursos estableciendo un programa de este tipo.

Identificar la información (la métrica) necesaria para lograr estas metas y establecer el marco de referencia de donde puede ser obtenida.

El cometido de los ocho pasos es crear un proceso a través del cual un programa corriente de métrica puede ser utilizado como una herramienta estratégica de gestión.

Métricas de proceso

Se recopilan de todos los proyectos y durante un largo periodo de tiempo

Caracterizados por:

* Control y ejecución del proyecto.
* Medición de tiempos de las fases.

Para este proyecto se trabajará con las siguientes métricas del proceso:

* Costo de remoción de defectos
* Cantidad de código rehusado
* Distribución de esfuerzo por fase
* Efectividad para remover defectos entre fases
* Soporte de herramientas para procesos propuestos

Métricas de proyecto

* Permiten evaluar el estado del proyecto.
* Permiten seguir la pista de los riesgos.

Para este proyecto se trabajará con las siguientes métricas del proyecto:

* Cantidad de puntos de función liberados por unidad de tiempo
* Costo del desarrollo
* Costo del soporte
* Horas trabajadas
* Tiempo (calendario) transcurrido
* Distribución del esfuerzo por fase
* Cambios sobre requerimientos durante el desarrollo
* Cambio sobre requerimientos en operación
* Origen de los cambios sobre requerimientos
* Cronograma Vs Estimado
* Costo sobre valor agregado
* Porcentaje de requerimientos implementados por unidad de tiempo

Métricas de producto

* Se centran en las características del software y no en cómo fue producido.
* También son productos los artefactos, documentos, modelos, y componentes que conforman el software.
* Se miden cosas como el tamaño, la calidad, la totalidad, la volatilidad, y el esfuerzo

Para este proyecto se trabajará con las siguientes métricas del producto:

* Puntos de Caso de Uso
* Puntos de función
* Complejidad de diseño (acoplamiento)
* Complejidad de código
* Métodos por clase
* Profundidad y ancho de jerarquías
* Cantidad de objetos y cantidad de relaciones de colaboración diferentes
* Volativilidad de componentes
* Complejidad de despliegue
* Densidad de defectos
* Tipo y origen de defectos
* Cantidad de problemas reportados
* Tiempo transcurrido entre fallas
* Tiempo esperado para la siguiente falla
* Tiempo requerido para reparar
* SLOC
* Facilidad de aprendizaje de uso

Estándar de documentación

Como estándares de documentación se definirán dos documentos:

* Estándar de documentación técnica
* Estándar de documentación de usuario.

La documentación técnica del producto debe:

* Ser adecuada para que un grupo independiente del de desarrollo pueda encarar el mantenimiento del producto.
* Incluir fuentes, Modelos de Casos de Uso, Objetos de diseño.

Para la escritura de documentos se han definido plantillas para ser utilizadas en la elaboración de entregables.

En estas plantillas se definen:

* Encabezado y pie de página.
* Fuente y tamaño de fuente para estilo normal.
* Fuente y tamaño de fuente para los títulos a utilizar.
* Datos mínimos que se deben incluir: fecha, versión y responsables.

Estándar de verificación y prácticas

Se utilizan las prácticas definidas en el Plan de Verificación y Validación.

Como estándar se utiliza el documento de:Std 1012-1986 IEEE Standard for Software Verification and Validation Plans.

Otros Estándares

En esta sección se deberán definir otros estándares a utilizar.

Revisiones y auditorías

Objetivo

Definición de las revisiones y auditorías técnicas y de gestión que se realizarán.

Especificación de cómo serán llevadas a cabo dichas revisiones y auditorías.

Requerimientos mínimos

Se especifican las revisiones y auditorías que deben realizarse como mínimo, así como la agenda para la realización de las mismas.

Revisión de requerimientos

Esta revisión se realiza para asegurar que se ha cumplido con los requerimientos especificados por el Cliente

Revisión de diseño preliminar

Esta revisión se realiza para asegurar la consistencia y suficiencia técnica del diseño preliminar del software.

Revisión de diseño crítico

Esta revisión se realiza para asegurar la consistencia del diseño detallado con la especificación de requerimientos.

Auditoría funcional

Esta auditoría se realiza previa a la liberación del software, para verificar que todos los requerimientos especificados en el documento de requerimientos fueron cumplidos.

Auditoría física

Esta revisión se realiza para verificar que el software y la documentación son consistentes y están aptos para la liberación.

Auditorías internas al proceso

Estas auditorías sirven para verificar la consistencia: del código versus el documento de diseño, especificaciones de interfase, implementaciones de diseño versus requerimientos funcionales, requerimientos funcionales versus descripciones de testeo.

Revisiones de gestión

Estas revisiones se realizan periódicamente para asegurar la ejecución de todas las actividades identificadas en este Plan. Deben realizarse por una persona ajena al grupo de trabajo.

Revisión del Plan de gestión de configuración

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de gestión de configuración.

Revisión Post Mortem

Esta revisión se realiza al concluir el proyecto para especificar las actividades de desarrollo implementadas durante el proyecto y para proveer recomendaciones.

Agenda

En esta sección se deberá especificar la agenda para las revisiones y auditorías detalladas anteriormente.

Otras revisiones

Revisión de documentación de usuario

Se revisa la completitud, claridad y aplicación de uso.

Verificación

Se debe identificar todas las verificaciones que no fueron identificadas en el Plan de V & V para el software y debe especificar los métodos a ser usados.

Reporte de problemas y acciones correctivas

Esta sección debe incluir: Descripción de las prácticas y procedimientos que se seguirán para el reporte, seguimiento, y resolución de los problemas surgidos en el desarrollo de software; especificar los responsables comprometidos con la implementación de estas acciones correctivas.

Herramientas, técnicas y metodologías

Se deben identificar herramientas de software, técnicas, y metodologías de soporte para las actividades de aseguramiento de calidad.

Gestión de riesgos

Se deben especificar los métodos y procedimientos utilizados para especificar, monitorear, y controlar las áreas de riesgo durante el proyecto.

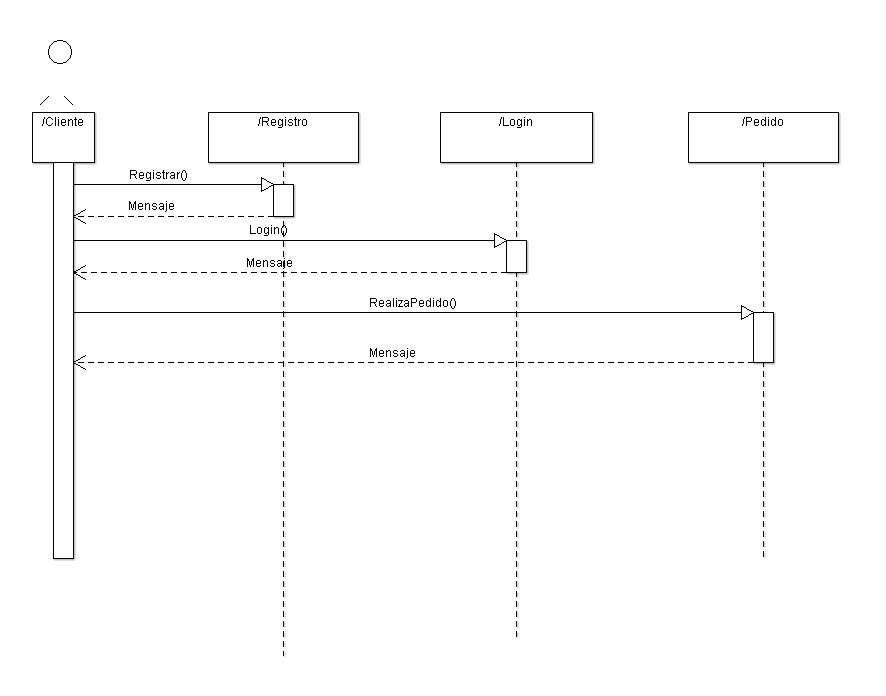
Los riesgos identificados, la estrategia de mitigación, monitoreo y plan de contingencia a ser llevados a cabo, serán descritos en el Documento de Gestión de Riesgos, con lo cual se podrá hacer referencia a él.

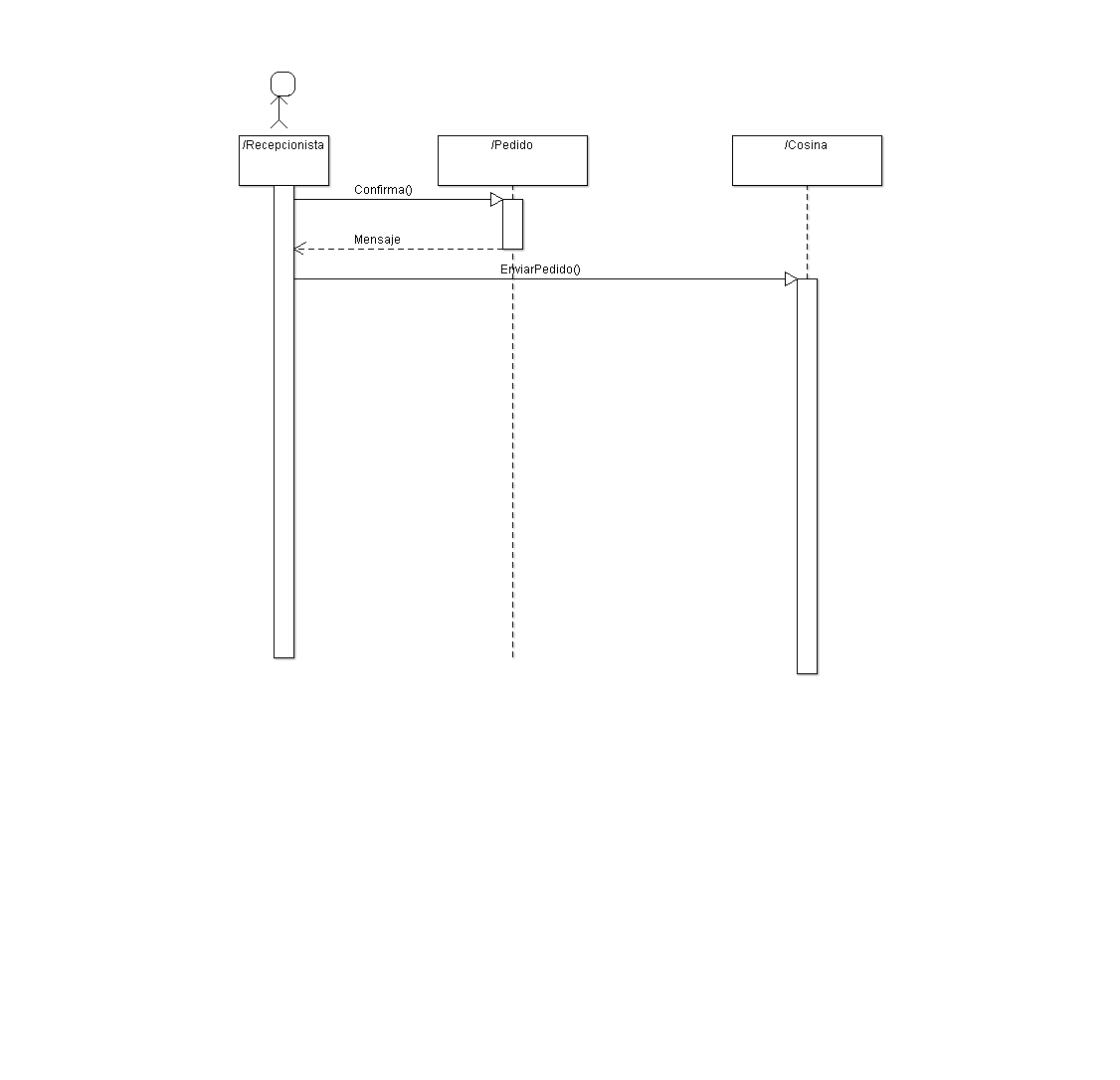
**Anexos**

Formulario de Pedidos y Detección de Cambios

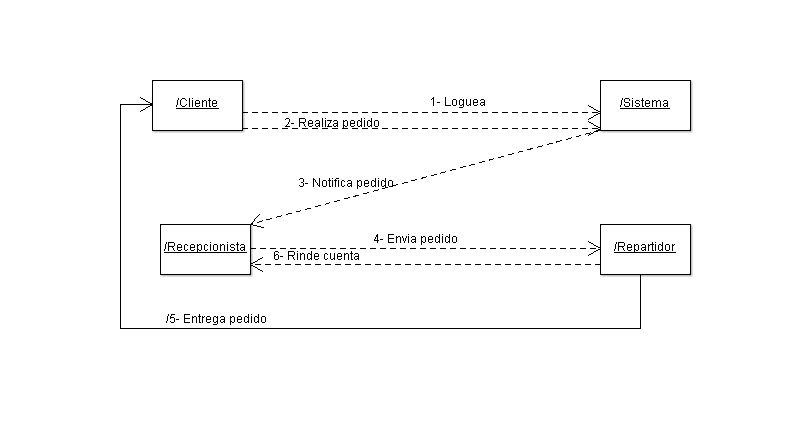
|  |  |
| --- | --- |
| Formulario de Pedidos y Detección de Cambios | |
| Fecha de Petición: |  |
| Nombre y Versión del Elemento |  |
| Nombre del Solicitante: |  |
| Necesidad de Cambio: |  |
| Descripción del cambio pedido: |  |
| Prioridad: |  |
| Estado: |  |
| Fecha del cambio: |  |
| Identificador de la nueva versión: |  |
| Que fue afectado por este cambio |  |

**Diagrama de Secuencia**

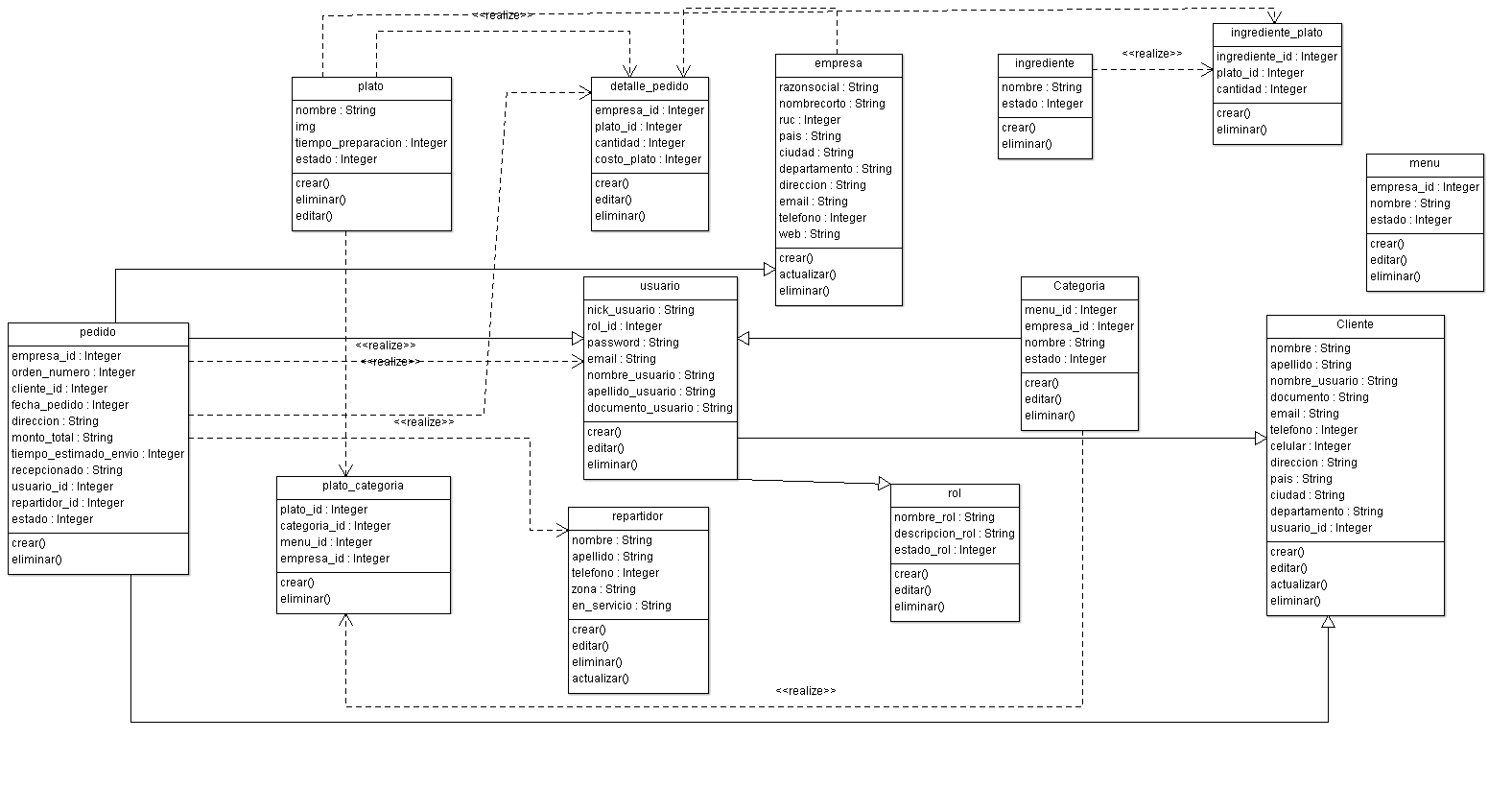
****

****

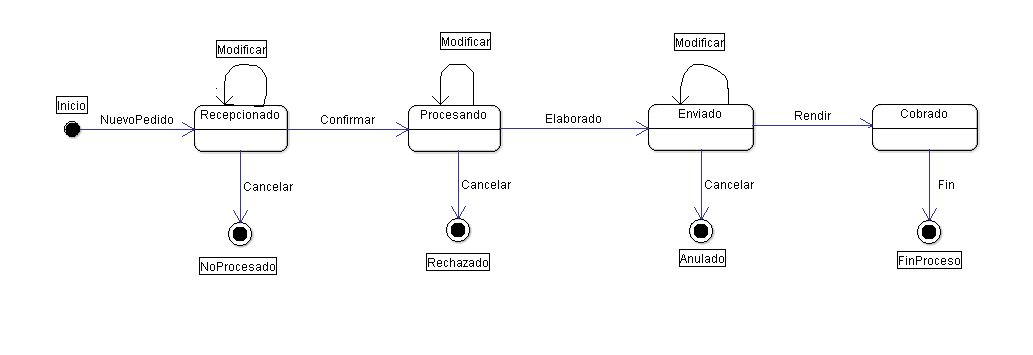
**Diagrama de Colaboración**

****

**Diagrama de Clases**

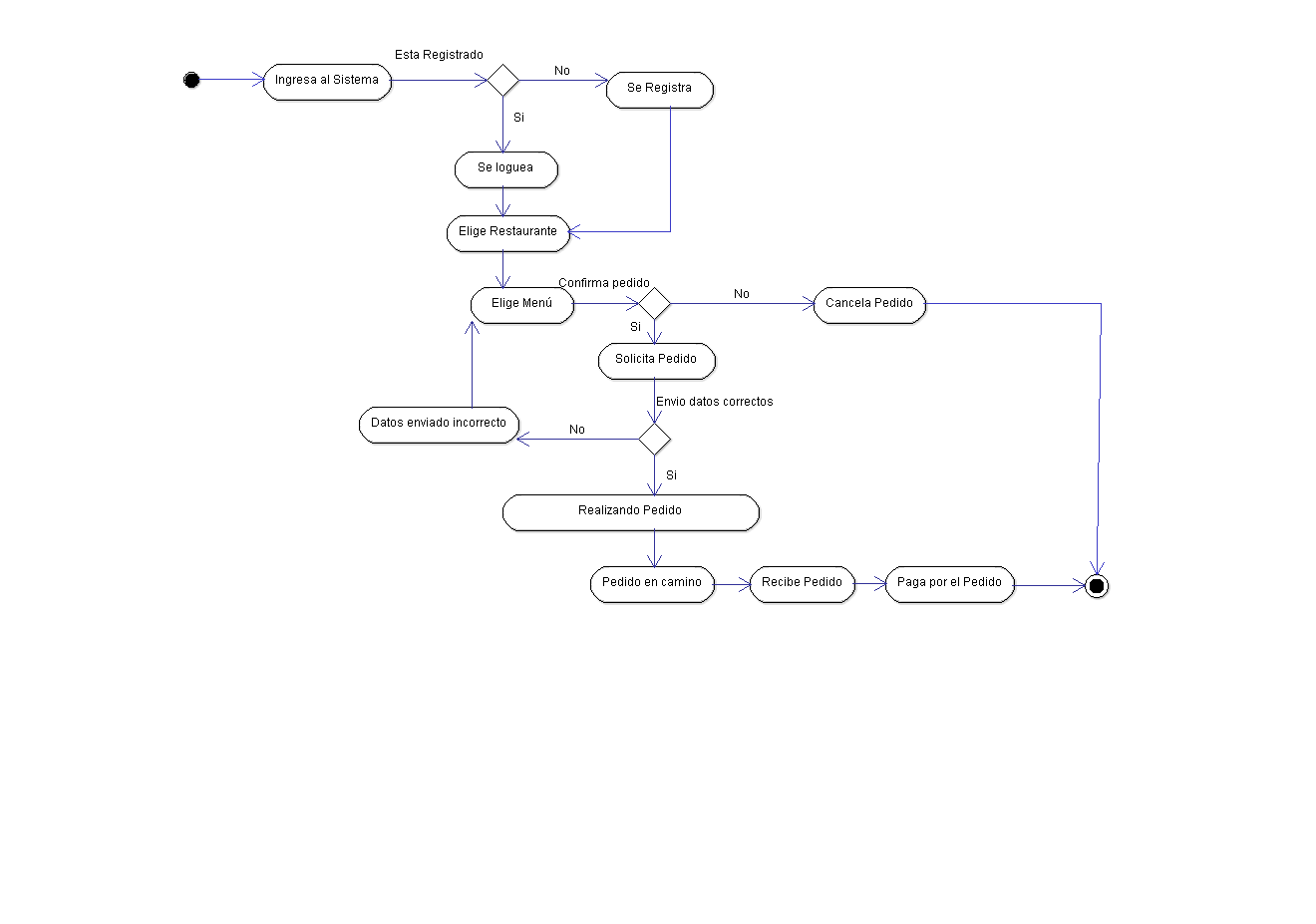
****

**Diagrama de Estados**

****

**Diagrama de Actividades**

Cliente

****

Recepcionista

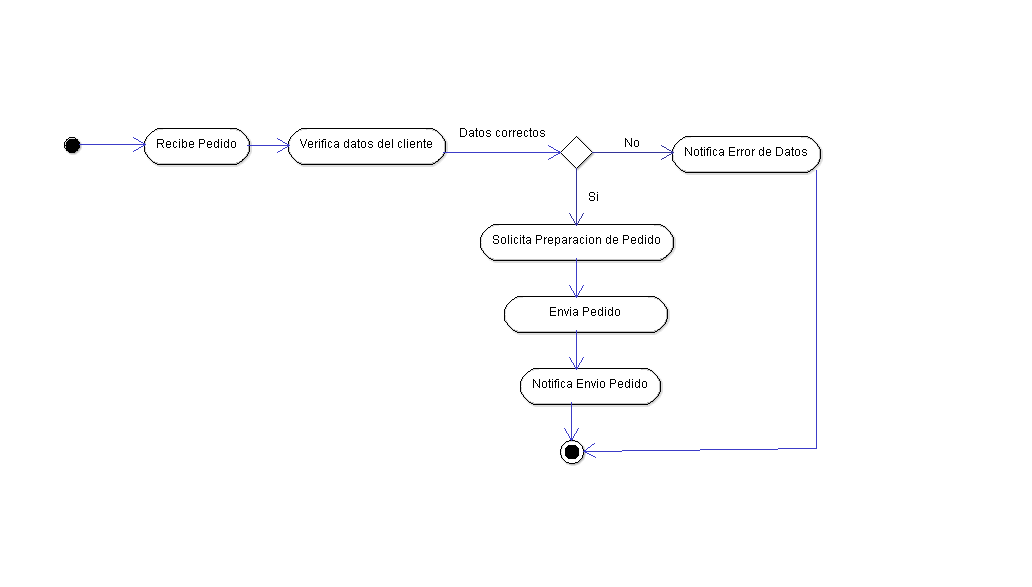
****

Diagrama de componente

