

**Aplicación web/móvil para establecimientos gastronomicos**

**RESTAPP**



April 4, 2013

Ferrero raviola salomón

INGEÑERIA EN SISTEMAS DE INFORMACIÓN

## Introducción

Cumpliendo con el plan de estudio de la carrera Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional, se presenta el siguiente desarrollo correspondiente a la cátedra de Proyecto Final de finalización de carrera.

[Palabras claves]

[Agradecimientos]

[Datos de contacto]

[Índice]

# Capítulo **1**

## Introducción de la Idea

### 1. IDENTIFICACIÓN DEL PROBLEMA

#### 1.1 Situación Actual

La forma en la que generalmente se resuelven las congestiones en establecimientos gastronómicos se basa en la contratación de personal extra, que solamente es aprovechado en periodos de tiempo muy breves donde la concurrencia en el local es elevada. Esta problemática se intensifica debido a la metodología de trabajo del personal, que generalmente consiste en un sistema de notas en lápiz y papel que se ingresan a un sistema centralizado para su procesamiento. Esto significa repetir el trabajo, primero en la mesa y después en el envío del pedido a la cocina. Sin dudas, esta tarea vuelve aún más ineficientes a aquél personal de atención que trabaja en grandes restaurantes donde las mesas se encuentran muy alejadas de la cocina y las distancias físicas son considerables, es decir, restringe al mozo a una cierta cantidad de mesas antes de verse obligado a volver a la cocina a informar a los cocineros sobre ellas. Todas estas cuestiones significan tiempos acumulativos que se reflejan en los tiempos de espera de los clientes y en la calidad de atención general del local.

Además de esto, muchos gerentes no cuentan con información más allá de la parte contable sobre la gestión de su negocio, la falta de información conlleva a una pobre toma de decisiones. Para la mayoría de los dueños de restaurantes, identificar qué platos son los más pedidos, el desempeño de un mozo en particular, en qué época del año se consume cada plato y/o como asignar tiempos de reservas, es una tarea difícil o imposible.

El presente proyecto apunta a crear una solución orientada a asistir la gestión de restaurantes, bares, bodegas y establecimientos gastronómicos del estilo mediante la aplicación de tecnologías de información móviles.

### 2**.** SOLUCIÓN PROPUESTA

Un sistema de Información es, por definición, un conjunto de elementos que interactúan entre sí con un fin común; que permite que la información esté disponible para satisfacer las necesidades en una organización. La problemática planteada, sugiere la necesidad de un sistema que permita la gestión de las actividades diarias de un restaurante. Para saber

#### 2.1 Información de entrada

Antes de poder definir una solución, se debe conocer la información a gestionar así como también el dominio del problema. En particular, se necesita conocer:

* Información de pedidos, las órdenes de cada persona atendida así como también los precios de los Ítems de menú solicitados y la mesa a la que se debe entregar.
* Información de los Ítems de Menú ofrecidos, la oferta del restaurante que será presentada al cliente.
* Información sobre disponibilidad de mesas, para poder avisar a potenciales clientes del estado del restaurante en caso de que deseen realizar una reservación.
* Información básica sobre usuarios del sistema (empleados).

La información provista existe en todo restaurante, se necesitará, sin embargo, un método de recolección de información apropiado que permita almacenarla y resumirla.

#### 2.2 Información de salida

Una vez procesada, para resolver la problemática planteada, el sistema de gestión desarrollado debería arrojar la siguiente información:

* Información cronológica resumida de los pedidos realizados, sus clientes e ítems solicitados.
* Información sobre montos recaudados.
* Información sobre el estado económico del restaurante.
* Información sobre tiempos de desocupación de mesas.

#### 2.3 Presentación del Sistema

El proyecto implica el desarrollo de un software adaptado a las necesidades del cliente que agilice la toma de datos, mejore el desempeño general de los procesos y genere información completa, oportuna y precisa para la toma de decisiones.

La aplicación, de ahora en adelante conocida bajo el nombre *RestApp*, se trata de un sistema software inclinado a restaurantes para terminales móviles (tablets) que se comunica con un servidor central permitiendo a los mozos tomar órdenes y enviarlas vía wireless a los cocineros. A su vez llevará la contabilidad básica del restaurante, calculando cuántas mesas se atendieron, cuánto se recaudó por mesa, quien las atendió, tiempo promedio en que las mesas están ocupadas/desocupadas, que plato se ordenó más, que plato no se ordenó en absoluto, tiempo de espera de los clientes antes de conseguir una mesa, etc.

La aplicación será totalmente personalizable, permitiendo a cada restaurante añadir su propio estilo, logo, menú y precios.

Se incluirá la posibilidad de generar reportes en tiempo real y subirlos a internet para que el dueño o encargado de la sucursal pueda saber que pasa en su local sin necesidad de estar presente. Además brindará la posibilidad de registrar reservas con tiempos precisos utilizando estudios de cola a través de un sitio web.

**Los objetivos de esta plataforma serán**:

* Agilizar los tiempos de espera de clientes en restaurantes (Asignación de tiempos).
* Optimizar la tarea de los mozos evitando el re trabajo.
* Proveer un menor tiempo de respuesta al cliente.
* Gestionar tiempos precisos de reserva de mesas mediante simulación de cola.
* Brindar un soporte a la toma de decisiones centralizando la información relacionada a los eventos.

#### 2.4 Arquitectura Preliminar

Procedemos a subdividir el sistema en una arquitectura basada en tres aplicaciones o plataformas diferentes:

* **Plataforma Mobile:** Toma de órdenes de pedido y confirmación de pagos.
* **Plataforma Web pública:** Interfaz para realización de reservas y consulta de información relacionada al negocio.
* **Plataforma Web Administración:** Gestión de usuarios, visualización de ordenes activas, generación de reportes y cobro a clientes.

La imagen siguiente tiene por objetivo diagramar la arquitectura tentativa del sistema:



**Figura 1:** Arquitectura preliminar del sistema.

[CAMBIAR IMAGEN]

##### 2.4.1 Aplicación Mobile

Una **aplicación móvil** o App es una [aplicación informática](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) diseñada para ser ejecutada en [teléfonos inteligentes](http://es.wikipedia.org/wiki/Tel%C3%A9fonos_inteligentes), [tabletas](http://es.wikipedia.org/wiki/Tableta_(computadora)) y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los [SO móviles](http://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil) como [Android](http://es.wikipedia.org/wiki/Android" \o "Android), [iOS](http://es.wikipedia.org/wiki/IOS_(sistema_operativo)" \o "IOS (sistema operativo)), [BlackBerry OS](http://es.wikipedia.org/wiki/BlackBerry_OS), [Windows Phone](http://es.wikipedia.org/wiki/Windows_Phone), entre otros. Para nuestro caso en particular, seleccionaremos el SO Android, por ser, sin duda alguna, el más utilizado en nuestro país, de acuerdo al sitio de estadísticas globales

TODO: Rest, json, gson, retrofit, dto

[http://gs.statcounter.com](http://gs.statcounter.com/).



Figura 2: Porcentaje de uso de SO móviles en Argentina desde Mayo a Octubre del 2013.

Además de su amplia distribución, Android tiene una gran selección de dispositivos de hardware, lo que le permitirá al dueño del restaurante ajustar su presupuesto según prefiera.

Aplicación móvil será la herramienta mediante la cual el personal de atención recolectará las órdenes de los clientes y las comunicará a los cocineros.

##### 2.4.2 Aplicación Web pública

En la ingeniería de software se denomina **aplicación web** a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

RestApp implementará una aplicación de este tipo para permitir a sus usuarios, en este caso, potenciales clientes del establecimiento gastronómico, realizar reservas, consultar disponibilidad, menú del día, promociones e información varia.

##### 2.4.3 Aplicación Web Administración

Desde la aplicación de administración los usuarios con los privilegios requeridos podrán visualizar y editar mesas activas, gestionar cobros, gestionar usuarios y permisos, ver historial de transacciones, enviar mensajes a los mozos u otros usuarios y generar reportes de estados.

##### 2.4.4 Base de datos

La BD será llevará un registro de usuarios del sistema, junto con sus correspondientes permisos. A su vez, almacenará información sobre Ítems de menú disponibles y un registro histórico ordenes de pedidos. Por defecto, la base de datos puede estar implementada en el mismo ordenador físico (servidor) que esté corriendo la aplicación web, con la posibilidad de exportarla a un servidor físico diferente. La tecnología seleccionada como gestor de base de datos es MySQL.

##### 2.4.5 API

Una API o interfaz de programación de aplicaciones representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación. En nuestro caso, la API se encargará de comunicar la base de datos con las distintas aplicaciones del sistema, el objetivo es abstraer la funcionalidad necesaria para hacer consultas la base de datos y, al mismo tiempo, re utilizar el código para hacerlo más fácil de mantener.



### 3. EMPRESA INVOLUCRADA

El equipo de desarrollo se hará con el conocimiento necesario para el análisis y construcción del sistema mediante la realización de entrevistas con la franquicia local de ‘Betos Lomos’, restaurante ubicado por 9 de julio 2302, San Francisco, (Córdoba). El objetivo es conocer a fondo el modelo de negocio a partir de personas con experiencia en el rubro y poder, de esta forma, adaptar el software lo mayor posible a las necesidades de un local de este tipo.

Cabe aclarar que ‘Betos Lomos’ solo será un soporte externo para el equipo de desarrollo, y que RestApp será personalizable y aplicable en cualquier tipo de establecimiento gastronómico.



### 4**.** ALCANCES Y LIMITACIONES

A través del ciclo de vida del proyecto, se llevarán a cabo las siguientes actividades:

* + Modelado de Negocio
  + Identificación de Requerimientos
  + Análisis
  + Diseño
  + Implementación
  + Pruebas
  + Confección de manual de usuario y sistema.

El resultado del proyecto será RestApp, una aplicación que incluye todos los componentes de software necesarios para su correcto funcionamiento, así como también su respectiva documentación y manuales.

RestApp no será un reemplazo directo para un sistema de gestión financiera o accesoria contable, y solo se limitará a realizar una contabilidad básica para toma de decisiones simples. Su función principal será agilizar los tiempos de atención, facilitar las tareas del personal del establecimiento y llevar un registro de transacciones del negocio.

### 5**.** MODELO DE NEGOCIO

Un modelo de negocio es el mecanismo por el cual un [negocio](http://es.wikipedia.org/wiki/Negocio) busca generar [ingresos](http://es.wikipedia.org/wiki/Ingreso) y [beneficios](http://es.wikipedia.org/wiki/Beneficio_econ%C3%B3mico). Es un resumen de cómo el proyecto planea servir a sus [clientes](http://es.wikipedia.org/wiki/Cliente_(econom%C3%ADa)).

El modelo de negocio describe el modo en que el proyecto creará y capturará la atención del segmento de mercado.

#### 5.1 Oferta de mercado actual

Después de un análisis de la oferta de mercado actual, no comprobamos la existencia de productos sustitutos. Sin embargo, existen soluciones similares. A continuación se detallan los productos en el mercado más conocidos/utilizados en el país, junto con sus funciones principales y una comparativa con el sistema propuesto:

##### Tu Carta Digital: Sistema de presentación al usuario final



 www.tucartadigital.com

TuCartaDigital es un servicio de cartas electrónicas de vinos se adaptan a cualquier tipo de local o negocio, son fácilmente administrables y sus clientes siempre podrán acceder vía web a su bodega estén donde estén y desde cualquier dispositivo.

##### **Comparativa con el sistema propuesto:**

TuCartaDigital ofrece al cliente la posibilidad de ver la oferta completa del restaurante cliente, incluyendo precios y fotografías. RestApp estará orientado a agilizar la gestión y se enfocará en brindar un servicio tanto al restaurante como al cliente. Al utilizar RestApp, el dispositivo móvil estará en manos del mozo o personal de atención, y solo se le entregará al cliente en caso de que desee visualizar la carta. Los objetivos respecto a este producto cambian considerablemente. Tu carta digital, no ofrece un sistema de pagos ni de toma de órdenes.

«Restorando»: Start Up Argentina: Sistema de reservas en restaurantes, gratis y al instante



 www.buenos-aires.restorando.com.ar

Lanzado en 2010 Restorando le ofrece a miles de personas la comodidad de poder reservar su mesa online en tiempo real. Restorando trabaja junto con los restaurantes para hacer más cómodo y efectivo el proceso de reserva y espera.

##### **Comparativa con el sistema propuesto:**

*Restorando* cubre una de los objetivos propuestos en el presente proyecto: Ofrecer un servicio de reservas online e información actualizada del estado del restaurante.

Sin embargo, no cubre las funcionalidades administrativas y de gestión relacionadas con la atención del cliente en el local que plantea RestApp, entre ellas, la posibilidad de generar órdenes.

#### 5.3 Plan de comercialización y características del producto

Debido a que el dominio del problema está bien definido, la aplicación resultante no será construida para un restaurante en particular. En cambio, se creará un producto genérico, totalmente personalizable que podrá ser adquirido por cualquier restaurante, bodega o establecimiento gastronómico.

RestApp será ofrecido como un sistema de paquete cerrado que incluirá los tres módulos definidos previamente (móvil, web publica y web administración) y el software requerido para poder ejecutar dichas aplicaciones. De forma opcional, es posible adquirir el hardware requerido recomendado por RestApp para la implementación del sistema.

### 6**.** ÁREAS DE CONOCIMIENTO INVOLUCRADAS

Aunque el proyecto involucra conocimientos adquiridos en todas las cátedras de la carrera Ingeniería en Sistemas de Información, las materias que, en mayor medida, contribuyeron en la elaboración del este proyecto son las siguientes:

* **Arquitecturas Móviles:** Programación de la aplicación móvil, desarrollo de aplicaciones nativas en el sistema operativo Android.
* **Algoritmos y estructuras de datos:** Fundamentos básicos de la programación, variables y estructura de datos.
* **Gestión de datos, Sistemas de gestión:** Fundamentos básicos del manejo, diseño y elaboración de bases de datos.
* **Construcción de software:** Buenas prácticas de programación.
* **Probabilidades y estadísticas, Simulación**: Teoría de colas.
* **Redes, Redes Avanzadas:** Comunicación de dispositivos, protocolos de red, seguridad.
* **Ingeniería de software, Ingeniería de software Orientada a objetos:** Planificación de proyectos, gestión de riesgos, gestión de configuraciones, metodologías de desarrollo.
* **Análisis de diseño, diseño de sistemas:** Construcción de modelos, diagramas, patrones de diseño.
* **Administración de Recursos:** Estudios de factibilidad.

### 7**.** ENTREGABLES

El siguiente cuadro lista los artefactos que constituirán los entregables al final del desarrollo del proyecto, brindando además una pequeña descripción de los mismos.

|  |  |
| --- | --- |
| Artefacto | Descripción |
| Componentes de software | * Aplicación para dispositivos Android: puede descargarse directamente desde Google Play, el código fuente estará disponible en el repositorio de GitHub. * Aplicación Web publica para que los usuarios accedan desde Internet. El código fuente estará disponible en el repositorio de GitHub. * Aplicación Web Administración para que los administradores configuren y parametricen el sistema. El código fuente estará disponible en el repositorio de GitHub. |
| Manual de usuario | Documento de [comunicación técnica](http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_t%C3%A9cnica) destinado a dar asistencia a las personas que utilizaran la aplicación. Una guía paso a paso para la instalación, uso y mantenimiento. |
| Documento del Proyecto | * Plan de Proyecto * Estudio de Pre Factibilidad * Plan de Gestión de Riesgos * Plan de Gestión de Configuraciones * Plan de Pruebas |
| Modelado | * Casos de Uso * Análisis * Diseño * Implementación * Despliegue |

# Capítulo **2**

## Estudio de Prefactibilidad

### 8. INTRODUCCIÓN

El estudio de prefactibilidad se lleva a cabo con el objetivo de contar con información sobre el proyecto a realizar, mostrando las alternativas disponibles y las condiciones técnicas, legales, operativas y financieras que rodean al proyecto. El siguiente documento hará referencia a todos aquellos aspectos que hacen posible la viabilidad del proyecto seleccionado.

### 9. ESTUDIO ECONÓMICO

El estudio de prefactibilidad económica tiene como objetivo brindar una primera estimación de los recursos financieros necesarios para llevar adelante el proyecto.

Se tendrán en cuenta los recursos utilizados para desarrollo de la solución, los mismos de clasifican en:

Hardware:

|  |  |  |
| --- | --- | --- |
| RECURSO | CANTIDAD | PRECIO |
| Notebook | 2 | $5500 |
| Pc Escritorio | 1 | $3800 |
| Dispositivo Móvil | 2 | $1800 |
| Impresora | 1 | $500 |

Software:

|  |  |  |
| --- | --- | --- |
| RECURSOS | CANTIDAD | PRECIO |
| Microsoft Word 2013 | 2 | $0 (Licencia educativa) |
| Enterprise Architect | 2 | $0 (Licencia educativa) |
| Git/GitHub | 3 | Gratuito |
| MySQL Workbench CE | 3 | Gratuito |
| Eclipse + Android SDK | 2 | Gratuito |
| Microsoft Visual Studio 2012 | 2 | $0 (Licencia educativa) |
| Twitter Bootstrap Framework | 2 | Gratuito |
| MySQL |  | Gratuito |
| Microsoft Windows 8 | 3 | $0 (Licencia educativa) |

### 10. ESTUDIO DE MERCADO

#### 10.1 ¿Qué beneficios le trae al negocio el sistema?

Es de esperarse una reducción considerable en el tiempo de espera en mesa promedio actual en hora pico. En verano, al haber una mayor concurrencia, se estima que este porcentaje sea aún mayor.

### 11. ESTUDIO TÉCNICO

¿Disponemos de los conocimientos y habilidades necesarias para la implementación del proyecto? ¿Se dispone del equipo y herramientas para llevarlo a cabo?

**Disponibilidad de Recursos tecnológicos principales a utilizar:**

**Tablets**: Actualmente podemos encontrar una variedad de fabricantes (Samsung, Apple) y sistemas operativos entre los cuales elegir (iOS, Android, Windows, Linux). Los aspectos que vamos a evaluar son la velocidad y alcance de transmisión de datos (wifi), la flexibilidad y facilidad de uso del SO, y la robustez física del dispositivo, debido al uso excesivo que deberá soportar. Además, Se debe considerar también la duración de la batería y el tiempo de recarga de la tablet.

**Routers inalámbricos**: Cualquier dispositivo Wi-Fi con velocidades superiores a 300Mbps y un alcance de moderado a bueno será suficiente. Preferentemente dos antenas o más.

**ISP / Conexión a internet**: Conexión de banda ancha de 3Mb o superior.

### 12. ESTUDIO LEGAL

Las herramientas a utilizar no suponen un problema legal ya que se utilizará software libre y software con licencia estudiantil.

### 13. ESTUDIO FINANCIERO

Analizamos si disponemos del capital necesario para emprender el proyecto y los costos derivados del mismo.

* **Tablets:** el costo varía según la marca y modelo que se elija, su precio oscila  $2200 y $4000.
* **Conectividad:** Red LAN interna, con salida a internet para generar reportes, routers, cables, fichas, costo $400
* **ISP: $**180



COSTO APROXIMADO TOTAL: $3580

***Conclusión:*** *Las limitaciones económicas no son relevantes y el equipo de desarrollo ya dispone de la mayoría de los artefactos necesarios para llevar a cabo el proyecto.*

# Capítulo **3**

## Plan de Proyecto

### 14. METODOLOGÍA Y CICLO DE VIDA

#### 14.1 RUP: Rational Unified Process

El proyecto hará uso del Proceso Unificado de Rational (RUP) en su desarrollo. RUP es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de [IBM](http://es.wikipedia.org/wiki/IBM). Junto con el Lenguaje Unificado de Modelado [UML](http://es.wikipedia.org/wiki/UML), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

Por afinidad y conocimientos con este proceso, se lo tomará como referencia para llevar a cabo el presente proyecto, con las adaptaciones y personalizaciones que el equipo de trabajo considere necesarias.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura 7.1 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.



Figura 7.1 *RUP*

#### 14.2 Fases

RUP es un modelo en fases que identifica cuatro fases discretas en el proceso de software (Inicio, Elaboración, Construcción y Transición). A diferencia del modelo en cascada, donde las fases se igual con actividades del proceso, las fases de RUP están más estrechamente vinculadas con la empresa que con las preocupaciones técnicas. El siguiente cuadro muestra los hitos y objetivos propuestos para el proyecto *RestApp* así como también una descripción breve de los mismos.

|  |  |  |
| --- | --- | --- |
| Fase | | Descripción |
| Inicio | OBJETIVOS | Durante esta primera fase, se buscará entender el dominio del problema y llegar a un consenso claro entre los integrantes del equipo de desarrollo sobre el producto a desarrollar, especificando las limitaciones que se consideren necesarias. Para ello, se llevarán a cabo entrevistas con restaurantes conocidos de la ciudad y se confeccionará un modelo de requerimientos de alto nivel. Además se especificarán los recursos necesarios para llevar a cabo el proyecto, se preparará el ambiente de trabajo y diseñara una arquitectura preliminar del sistema. |
| HITOS | * Establecer una justificación o caso de negocio para el proyecto * Establecer el ámbito de sistemas y sus limitaciones. * Definir los casos de usos y requerimientos principales. * Definir una o más arquitecturas posibles para el sistema. * Identificar riesgos. * Preparar una planificación preliminar del proyecto y un costo estimado. |
| Elaboración | OBJETIVOS | Durante la fase de elaboración se amplía la comprensión del problema de dominio, se intenta establecer un marco conceptual arquitectónico para el sistema, se diseña el plan de proyecto e identifican y priorizan los riesgos claves. |
| HITOS | * Documento de requerimientos. * Modelo de caso de usos, con la gran mayoría de actores y descripciones de casos de usos definidos. * Descripción de la arquitectura de software. * Diagrama de Clases. * Modelo de Datos. * Diagrama de Secuencia * Diagrama de Actividades * Priorizar los riesgos identificados y crear planes de prevención y contingencia. |
| Construcción | OBJETIVOS | El objetivo principal de esta fase es construir el sistema de software, se concentra en de desarrollo de componentes y características del sistema. En esta fase, se desarrollan los tres módulos del sistema en forma paralela, se prueban sus interfaces y se integran para formar el sistema final. Además, se confecciona un instructivo de uso para el usuario final. |
| HITOS | * Sistema de software completo para poner en funcionamiento. * Instructivo de uso para el usuario final. |
| Transición | OBJETIVOS | En la fase de Transición se traslada el sistema desde el ambiente de desarrollo a un ambiente real. Una vez que la aplicación esté disponible para el usuario final se procede con la instalación en el ambiente de trabajo. También, se capacita en su uso a los usuarios finales y se valida el desempeño del sistema contra sus expectativas. Por último, se evalúa el nivel de calidad del producto final con aquél establecido en la fase de Inicio. |
| HITOS | * Sistema de software probado y con funcionamiento correcto/esperado en un entorno operacional (más documentación asociada). * Comparativa con planeaciones iniciales. * Instructivo de uso refinado. |

#### 14.3 Iteraciones

Cada fase en RUP puede ser dividida en iteraciones. Una iteración es un bucle completo de desarrollo. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes y verifican que los componentes satisfacen los casos de uso. Una vez que la iteración satisface a sus objetivos, el desarrollo continúa con la iteración siguiente.

Este enfoque incremental ofrece ciertas ventajas, entre ellas:

* Los riesgos se mitigan en etapas tempranas
* El cambio es más manejable
* Existe más nivel de reutilización
* La calidad final del producto es mejor.

A continuación se presenta una tabla que describe las iteraciones planificadas para el presente proyecto:

|  |  |  |
| --- | --- | --- |
| FASE | # | OBJETIVOS |
| Inicio | 1 | **ITERACCIÓN I1:**  En esta primera iteración se desarrolla una descripción del producto final a partir de la idea original y se confecciona el análisis de negocio para el producto. La iteración se enfoca en estudiar el dominio de negocio, extraer los requisitos funcionales y no funcionales principales y establecer las restricciones de la solución. Esto implica identificar todos los casos de usos y describir los más importantes.  Interpretar el dominio de negocio significa identificar y evaluar los riesgos del proyecto, estimar los recursos necesarios y confeccionar un plan que especifique fechas y objetivos principales.  **Documentos resultantes:**   * Un documento que detalle una visión general de los requerimientos principales del proyecto, sus características y limitaciones. * Un modelo inicial de casos de uso (Completo en un 10% - 20%) * Una evaluación de riesgos inicial. * Un plan de proyecto, que especifique fases e iteraciones. |
|  | Meta | **Definición de objetivos del ciclo de vida**   * Se logró una comprensión de los requerimientos clara y precisa. * Se logró un acuerdo en el equipo de desarrollo sobre las estimaciones de costo, tiempos, prioridades, riesgos y proceso de desarrollo.   El proyecto puede ser cancelado o re-pensado considerablemente si falla alcanzar esta meta. |
| Elaboración | 3 | ITERACIÓN E1:  Se identifican y especifican en detalle los casos de usos principales y sus respectivas descripciones. Se detallan y priorizan los riegos del proyecto según probabilidad de ocurrencia e impacto y se crean planes de prevención y contingencia. Se confeccionan los casos de usos y un modelo de casos de uso.  **Documentos resultantes:**   * Un modelo de casos de uso (al menos 60% completo), con todos los actores identificados y la mayoría de las descripciones. * Identificación de una arquitectura software preliminar de la solución. * Un documento que prioriza los riesgos del proyecto en función de su probabilidad de ocurrencia y criticidad. * Desarrollo de prototipos independientes para prueba. * Un refinamiento del plan del proyecto que identifique las actividades de cada iteración y el criterio de evaluación. |
| ITERACIÓN E2 :  Se define la arquitectura del software mediante la creación de un diagrama de clases y un modelo de datos. En esta iteración, se crean los diagramas de actividades y secuencia. Se reúne la información necesaria para identificar la tecnología a apropiada para la implementación.  **Documentos resultantes:**   * Un modelo de casos de uso (al menos 80% completo), con todos los actores identificados y la mayoría de las descripciones. * Mapeo de requerimientos a casos de usos. * Diagrama de Clases (al menos 50% completo). * Modelo de Datos (al menos 60% completo). * Diagrama de Secuencia * Diagrama de Actividades |
| ITERACIÓN E3:  Se confeccionan prototipos funcionales independientes de los módulos del sistema más importantes (móvil, web frontend y backend) para familiarizar el equipo de trabajo con la tecnología seleccionada. Se refinan los documentos generados en la fase de elaboración.  **Documentos resultantes:**   * Modelo de casos de uso (al menos %90 completo). * Diagrama de Clases (al menos 90% completo). * Modelo de Datos (al menos 90% completo). * Un documento de riesgos con planes de prevención y contingencia. * Refinamiento de Secuencia * Refinamiento de Actividades |
|  | Meta | Al final de la etapa de elaboración, se analizan los objetivos del sistema de forma detallada así como su dominio, la elección de arquitectura y la resolución de los principales riesgos.  Las respuestas a las siguientes preguntas debería ser afirmativa:   * ¿La visión del producto es estable? * ¿La arquitectura es estable? * ¿Los prototipos independientes ejecutables muestran que los riesgos encontrados pueden ser afrontados de una forma creíble? * ¿El plan para la fase de construcción está lo suficiente mente detallado?   El proyecto pude ser re planificado o considerablemente re estructurado si falla en conseguir esta meta. |
| Construcción | 6 | ITERACCIÓN C1:  Construcción de la base de datos. |
| ITERACCIÓN C2:  Definición de las interfaces de usuarios para las aplicaciones Móviles, web backend y frontend. Pruebas de conectividad con la base de datos, corrección de fallas. |
| ITERACCIÓN C3:  Aplicación Móvil: Creación y autentificación de usuarios.  Aplicación web frontend: Refinamiento de interfaz de usuario, integración de registros de usuario con Facebook  Aplicación web backend: Creación y autentificación de usuarios. Refinamiento de la interfaz. |
| Aplicación Móvil: Codificación de ABM principales y configuraciones de personalización. Refinamiento de interfaz de usuario.  Aplicación web frontend: ABM de reservas.  Aplicación web backend: Codificación de ABM principales  Comunicación con monitor en la cocina. |
| ITERACCIÓN C4:  Calculo de colas, Cobros, comunicación servicio AFIP, impresión de facturas. |
| ITERACCIÓN C5:  Pruebas de interfaces, pruebas de caja negra, blanca … |
| Transición | 1 | ITERACIÓN T1:  El objetivo principal consiste en el "traspaso" del sistema desarrollado a la producción, por lo que es disponible y comprensible para el usuario final.  Creación de video tutoriales para los usuarios finales (administradores y personal de atención)  Pruebas de funcionamiento en situaciones criticas  Comparación con las expectativas de calidad en la fase de inicio.  Evaluación de riesgos e incidentes. |

### 15. RECURSOS, HERRAMIENTAS Y ROLES

La realización del proyecto requerirá de recursos humanos, económicos, físicos, y técnicos. A continuación se listan los recursos disponibles.

#### 15.1 Recursos Humanos

|  |  |
| --- | --- |
| NOMBRE DEL RECURSO | DENOMINACIÓN EN EL DIAGRAMA DE GANT |
| Ferrero, Lucas | LF |
| Raviola, Fernando | FR |
| Salomón, Oscar | OS |

#### 15.2 Roles y Responsabilidades

|  |  |  |
| --- | --- | --- |
| ROL | RESPONSABILIDADES | RECURSO |
| Project Manager | Planeación, control y gestión del plan de proyecto. | OS, FR, LF |
| Arquitecto de software | Decidir las tecnologías a utilizar para el desarrollo de la solución. | LF |
| Analista Funcional | Registrar los requerimientos y confeccionar los casos de uso. | LF |
| Test Manager | Asegurar y documentar las pruebas del sistema. | OS |
| Deployment Manager | Supervisar el progreso del desarrollo de las aplicaciones de software. | OS,FR,LF |
| Configuration Manager | Gestiona las políticas y el ambiente de desarrollo para llevar a cabo una eficiente gestión de configuraciones. | FR |
| Modelador de  Análisis & Diseño | Responsable de los modelos de Análisis (dominio del problema) y de Diseño (dominio de la solución). | OS,FR,LF |
| Administrador de Base de Datos | Responsable de la estructura, los datos y la seguridad del Web Service / Base de datos. | OS |

#### 15.3 Herramientas de Software [COMPLETAR]

|  |  |
| --- | --- |
| HERRAMIENTA | LICENCIA |
| Microsoft Office (Word, Project) | Educativa |
| Enterprise Architect | Educativa |
| GitHub | N/A |
| MySQL Workbench | Community edition |
| Android Studio | Open source |
| Sublime Text | Open source |
| Apache | Servidor web open source |

# Capítulo **4**

## Gestión de Riesgos

### 16. INTRODUCCIÓN

Cualquier proyecto, sin importar su naturaleza, enfrenta factores externos e influencias que inducen algún grado de incertidumbre sobre su costo final, su fecha de finalización y/o sus objetivos. El efecto que esta incertidumbre tiene sobre los objetivos de un proyecto se denomina “riesgo”.

Toda actividad involucra riesgo. La gestión de riesgos de un proyecto se encarga de anticiparse a los mismos mediante un análisis, priorización y creación de planes de prevención y contingencia para aquellos riesgos más relevantes.

El presente proyecto hace uso parcial de la guía ‘*A structured approach to Enterprise Risk Management (ERM) and the requirements of ISO 31000’* para describir el proceso de forma lógica y sistemática.

Esta guía es el resultado del trabajo de un equipo conformado por las principales organizaciones de gestión de riesgos en el Reino Unido – la Asociación de Seguro y Gestión de Riesgo (Association of Insurance and Risk Managers - AIRMIC), la Asociación de Gestión de Riesgos del sector público (Public sector risk management association – ALARM) y el Instituto de gestión de Riesgos (Institute of Risk Management IRM). Para más información:

<http://www.ferma.eu/wp-content/uploads/2011/10/a-structured-approach-to-erm.pdf>

### 17. EVALUCACIÓN DE RIESGOS

La evaluación de riesgo involucra la identificación e priorización de los mismos. Se presenta a continuación una descripción de los riesgos que afectarán al proyecto así como también sus niveles de probabilidad e impacto.

#### 17.1 Identificación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | # | Nombre | Probabilidad | Impacto |  |
| USUARIO | 1 | Resistencia al cambio | Baja | Bajo |  |
| 2 | Curva de aprendizaje elevada | Media | Media |  |
| TECNOLÓGICOS | 3 | Dificultad de aplicación de tecnologías | Media | Alta |  |
| 4 | Incompatibilidad de dispositivos/tecnologías seleccionadas | Baja | Alto |  |
| 11 | Los componentes desarrollados por separados no se pueden relacionar con facilidad | Media | Alto |  |
| PLANEAMIENTO & CONTROL | 5 | Planificación demasiado optimista | Alta | Alto |  |
| 6 | Omisión de tareas criticas | Alta | Alto |  |
| 7 | Cambio de requerimientos | Baja | Medio |  |
| 8 | Modelado inadecuado, incorrecto o incompleto. | Alta | Alto |  |
| EQUIPO DE TRABAJO | 9 | Incumplimiento de los horarios de trabajo establecidos | Media | Alto |  |
| 10 | Aceptación de ofertas laborales (reducción de disponibilidad horaria) | Baja | Alto |  |

17.1 Planes de Prevención y Contingencia

|  |  |
| --- | --- |
| RESISTENCIA AL CAMBIO | |
| # | 1 |
| DESCRIPCIÓN | El usuario no se adapta al nuevo sistema e intenta evitar su uso. Lo encuentra impráctico, lento, difícil de usar, poco intuitivo o simplemente incómodo. |
| IMPACTO | El proyecto sufrirá un retraso causado por la necesidad de entender el punto de vista del cliente, analizar el problema y re-diseñar los aspectos que hacen que el sistema no se adecue a las expectativas de los usuarios. |
| PLAN DE PREVENCIÓN | Se realizaran y analizaran entrevistas con potenciales usuarios y/o conocedores del rubro en **dos** ocasiones diferentes, para asegurarse de que se comprende el dominio del problema en su totalidad así como también definir la mejor forma de encararlo. |
| PLAN DE CONTINGENCIA |  |

|  |  |
| --- | --- |
| CURVA DE APRENDIZAJE ELEVADA | |
| # | 2 |
| DESCRIPCIÓN | Si bien el sistema funciona según lo pactado, el usuario no puede implementarlo a corto plazo sin un extenso instructivo que explique su utilización. Resulta difícil para los principiantes, comprender los aspectos básicos del sistema de software. |
| IMPACTO | El proyecto |
| PLAN DE PREVENCIÓN | Se realizarán instructivos de uso concreto y apuntado a los diferentes usuarios del sistema (Administradores, Personal de atención, Cocineros, etc.) |
| PLAN DE CONTINGENCIA | Se creará un sitio web con video tutoriales sobre el mantenimiento y uso del sistema y se incorporará además una sección de FAQ (preguntas frecuentes). |

|  |  |
| --- | --- |
| DIFICULTAD DE APLICACIÓN DE TECNOLOGÍAS | |
| # | 3 |
| DESCRIPCIÓN | El equipo de trabajo tiene problemas al codificar el sistema o al establecer su estructura física. |
| IMPACTO | El proyecto sufrirá un retraso directamente proporcional del obstáculo a superar. |
| PLAN DE PREVENCIÓN | Durante todo el proyecto se construirán prototipos desechables incrementables con el único objetivo de familiarizarse con las funciones del lenguaje de programación/software a utilizar |
| PLAN DE CONTINGENCIA | Se buscará ayuda profesional referida al inconveniente. |

|  |  |
| --- | --- |
| INCOMPATIBILIDAD DE DISPOSITIVOS / TECNOLOGIA SELECCIONADA | |
| # | 4 |
| DESCRIPCIÓN | La tecnología seleccionada para llevar a cabo el proyecto no es la idea, es difícil comunicar los distintos módulos, surgen problemas de compatibilidad entre los dispositivos. |
| IMPACTO | El sistema de software resultante será inestable, se necesitarán muchas herramientas de terceros o frameworks para solucionar problemas de comunicación. La velocidad final puede verse afectada. |
| PLAN DE PREVENCIÓN | En el estudio de mercado se considerará la tecnología no solo por precio de dispositivo, facilidad de uso y facilidad de implementación sino que también se tendrá en cuenta su alcance. A su vez, la definición de interfaces de comunicación del proyecto será planificada en etapas tempranas para considerar un cambio de herramientas. |
| PLAN DE CONTINGENCIA | Al poder detectar problemas en etapas tempranas, el plan de contingencia será cambiar la tecnología seleccionada, ya con un mejor entendimiento del sistema, elegir una acorde a la solución. |

|  |  |
| --- | --- |
| PLANIFICACIÓN DEMASIADO OPTIMISTA | |
| # | 5 |
| DESCRIPCIÓN | Las duraciones estimadas de las actividades en el plan de proyecto son considerablemente menores a las reales. |
| IMPACTO | El tiempo total del proyecto es mucho mayor al tiempo pactado en la planificación. No se respeta el plan de trabajo por lo que se requieren re planificaciones que conllevan aún más retrasos. |
| PLAN DE PREVENCIÓN | Se añadirá un margen de error a cada estimación en la planificación de aproximadamente un 10% en sentido pesimista para salvaguardar posibles retrasos no contemplados. |
| PLAN DE CONTINGENCIA | Como primer medida, se intentará reasignar los recursos para cada tarea (Cada miembro del equipo realizará tareas independientes para acelerar el proceso). Si esto falla o simplemente no aplica, se procederá a re planificar considerando un margen de error que dependerá del retraso actual y de la naturaleza de la actividad. |

|  |  |
| --- | --- |
| OMISIÓN DE TAREAS CRITICAS | |
| # | 6 |
| DESCRIPCIÓN | No se incluyeron en la planificación tareas necesarias para la finalización del proyecto. Alguna de las tareas incluidas no eran lo suficientemente atómicas y debieron ser subdivididas. |
| IMPACTO | El tiempo total del proyecto es mucho mayor al tiempo pactado en la planificación. |
| PLAN DE PREVENCIÓN | La planificación contemplará la revisión de la misma en el workflow de ‘plan de proyecto’ exigido por la metodología RUP. Se actualizará la línea base en cada iteración con el objetivo de detectar la mayor cantidad de actividades posibles. |
| PLAN DE CONTINGENCIA | COMPLETAR |

|  |  |
| --- | --- |
| CAMBIO DE REQUERIMIENTOS | |
| # | 7 |
| DESCRIPCIÓN | Los requerimientos del sistema de software a realizar cambian. Se agregan nuevas funciones y/o características consideradas de mayor importancia. |
| IMPACTO | Trabajo en casos de usos con requerimientos de baja prioridad, extensión de la duración del proyecto. |
| PLAN DE PREVENCIÓN | Las entrevistas están diseñadas para extraer la mayor cantidad de requerimientos. Se llevaran a cabo tres iteraciones dentro de la fase elaboración para asegurar una base sólida al sistema. |
| PLAN DE CONTINGENCIA | Se analizará la importancia de los nuevos requerimientos y se descartarán requerimientos de menor jerarquía. |

|  |  |
| --- | --- |
| MODELADO INADECUADO, INCORRECTO O INCOMPLETO. | |
| # | 8 |
| DESCRIPCIÓN | El modelado inicial no tiene calidad de diseño lo que destina al proyecto a una calidad de implementación pobre. |
| IMPACTO | Retrasos en el proyecto debido a re-trabajos, el modelado presenta inconvenientes en etapas posteriores. |
| PLAN DE PREVENCIÓN | Mediante las iteraciones del proceso RUP se refinan de manera progresiva cada uno de los modelos que componen el sistema. Revisando y haciendo correcciones según corresponda. |
| PLAN DE CONTINGENCIA | Dependiendo de la fase en la que el proyecto se encuentre cuando se presente el inconveniente se tomarán medidas apropiadas. En la fase de elaboración, se replantean los modelos existentes y se evalúa la posibilidad de replanificar y añadir una nueva iteración. De manifestarse en la fase de construcción, se considerarán los modelos más representativos del sistema y se concentrarán esfuerzos en aplicar correcciones a los mismos, siempre otorgando prioridad a aquellos requerimientos más importantes. |

|  |  |
| --- | --- |
| INCUMPLIMIENTO DE LOS HORARIOS DE TRABAJO ESTABLECIDOS | |
| # | 9 |
| DESCRIPCIÓN | El equipo de trabajo no respeta los horarios establecidos en el calendario de tesis. |
| IMPACTO | Retraso en actividades, dilatación de la duración del proyecto. |
| PLAN DE PREVENCIÓN | Pactar días y horarios convenientes para los integrantes del equipo de desarrollo. Proveer un medio de comunicación que permita informar de faltas y tardanzas. |
| PLAN DE CONTINGENCIA | Establecer nuevas fechas de encuentro que respeten o mejoren las estimaciones de la planificación. Replanificar el proyecto con los nuevos horarios. |

|  |  |
| --- | --- |
| ACEPTACIÓN DE OFERTAS LABORALES (REDUCCIÓN DE DISPONIBILIDAD HORARIA) | |
| # | 10 |
| DESCRIPCIÓN | Uno o más de los integrantes del equipo decide emplearse en un trabajo de 8 horas o más. |
| IMPACTO | Extensión de la duración del proyecto. Reducción de comunicación entre integrantes. |
| PLAN DE PREVENCIÓN | Siendo de interés común para el equipo de trabajo terminar con el proyecto, los integrantes del mismo se comprometen a dedicar un cierto número de horas diarias (véase calendario de planificación) independientemente de sus respectivas actividades personales. |
| PLAN DE CONTINGENCIA | Replanificar teniendo en cuenta las nuevas limitaciones horarias |

|  |  |
| --- | --- |
| LOS COMPONENTES DESARROLLADOS POR SEPARADOS NO SE PUEDEN RELACIONAR CON FACILIDAD | |
| # | 11 |
| DESCRIPCIÓN | Las comunicaciones entre los distintos módulos no funcionan de la forma que se esperaba. |
| IMPACTO | Retraso de actividades, y replanifacion del proyecto debido a investigación para dar solución al problema. |
| PLAN DE PREVENCIÓN | Se dedicará cierta cantidad de tiempo a identificar cuáles son las formas de comunicar los diferentes módulos para que operen de forma correcta y óptima, reutilizando código. Se consultará, además, con profesionales en el tema, en caso de ser necesario. |
| PLAN DE CONTINGENCIA | Se consultara con profesionales locales o mediante la ayuda de Internet una solución al problema. Además, en caso de ser necesario, se replanificara el proyecto. |

#### 17.2 Seguimiento, Control y Resultados de los Riesgos

A continuación se detallan los riesgos que han sido mitigados mediante la ejecución de los planes de contingencia correspondientes, así como también aquellos riesgos que no han ocurrido gracias a los planes de prevención detallados en el punto anterior.

|  |  |  |  |
| --- | --- | --- | --- |
|  | # | Nombre | Estado |
| USUARIO | 1 | Resistencia al cambio | NO OCURRIÓ / MITIGADO |
| 2 | Curva de aprendizaje elevada | NO OCURRIÓ / MITIGADO |
| TECNOLÓGICOS | 3 | Dificultad de aplicación de tecnologías | OCURRIÓ |
| 4 | Incompatibilidad de dispositivos/tecnologías seleccionadas | NO OCURRIÓ / MITIGADO |
| 11 | Los componentes desarrollados por separados no se pueden relacionar con facilidad | OCURRIÓ |
| PLANEAMIENTO & CONTROL | 5 | Planificación demasiado optimista | OCURRIÓ |
| 6 | Omisión de tareas criticas | NO OCURRIÓ / MITIGADO |
| 7 | Cambio de requerimientos | OCURRIÓ |
| 8 | Modelado inadecuado, incorrecto o incompleto. | OCURRIÓ |
| EQUIPO DE TABAJO | 9 | Incumplimiento de los horarios de trabajo establecidos | OCURRIÓ |
| 10 | Aceptación de ofertas laborales (reducción de disponibilidad horaria) | OCURRIÓ |

**TODO: Hacer análisis de los porcentajes de los riesgos ocurridos sobre el total y ver si hacer un análisis, luego de finalizar el proyecto, post mortem.  
Evaluar el riesgo agregado #11.**

# Capítulo **5**

## Gestión de Configuraciones

### 18. INTRODUCCIÓN

Se denomina Gestión de la Configuración al conjunto de procesos destinados a asegurar la calidad de todo producto obtenido durante cualquiera de las etapas del desarrollo de un [Sistema de Información](http://es.wikipedia.org/wiki/Sistema_de_Informaci%C3%B3n) (S.I.), a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo

En resumen, las principales tareas en la gestión de configuraciones son:

* Llevar el control de todos los elementos de configuración de la infraestructura TI con el adecuado nivel de detalle y gestionar dicha información a través de la Base de Datos de Configuración (**CMDB**).
* Proporcionar información precisa sobre la configuración TI a todos los diferentes procesos de gestión.
* Interactuar con las **Gestiones de Incidentes, Problemas, Cambios y Versiones** de manera que estas puedan resolver más eficientemente las incidencias, encontrar rápidamente la causa de los problemas, realizar los cambios necesarios para su resolución y mantener actualizada en todo momento la **CMDB**.
* Monitorizar periódicamente la configuración de los sistemas en el entorno de producción y contrastarla con la almacenada en la **CMDB** para subsanar discrepancias.

A continuación se define una lista con los elementos que se van a administrar durante el proyecto:

* Proyecto Final RestApp (Microsoft Word 2013)
* Planificación (Microsoft Project 2013)
* Proyecto RestApp (Enterprise Architect)
* Manual De usuario (Microsoft Word 2013)
* Base de Datos (MySQL Workbench CE)
* Código fuente de las soluciones

### 19. HERRAMIENTAS PARA LA GESTIÓN DE CONFIGURACIONES

El proceso de versionado será asistido por:





**GitHub** es uno de los sistemas de control de versiones más populares entre los desarrolladores. Un excelente servicio de alojamiento de repositorios de software con este sistema, que lejos de quedarse en esta funcionalidad, ofrece hoy en día un conjunto de características muy útiles para el trabajo en equipo.

#### 19.1 Características de GitHub útiles para el trabajo en equipo

* Un **wiki** la cual opera con Git para el mantenimiento de las distintas versiones de las páginas.
* Un sistema de seguimiento de problemas, que al estilo del clásico sistema de tickets, permiten a los miembros de tu equipo (o a cualquier usuario de GitHub si tu repositorio es público) abrir un ticket detallando un problema que tenga con tu software o una sugerencia que desee hacer al mismo.
* Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero (ej: “Esto es mejor que lo extraigamos a una nueva clase”), y debatir sobre determinados cambios realizados en un commit específico.
* Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

#### 19.2 Trabajando con GitHub

Se trabaja con tres repositorios distintos:

* **Repositorio upstream**: es el repositorio en github de la aplicación con la cual vamos a colaborar. No podremos hacer ningún cambio directamente sobre este repositorio, ya que no tendremos permisos de escritura sobre el mismo. Lo que haremos es crear nuestra propia copia *(fork)* del mismo.
* **Repositorio origin**: es nuestra copia del repositorio alojada en github. A este repositorio enviaremos los cambios que efectuaremos en nuestro repositorio local.
* **Repositorio local**: es el repositorio local que tendremos en nuestra propia estación de trabajo.

Comúnmente realizaremos los cambios trabajando en nuestro repositorio **local**. Cuando hayamos confirmado *(commit)* nuestros cambios localmente, los enviaremos *(push)* a nuestro repositorio **origin** alojado en github. Luego haremos un pedido *(pull request)* al repositorio **upstream** solicitándole que integre los cambios del repositorio **origin**.

PORTADA FIN GESTIÓN

PORTADA COMIENZO PRODUCTO

# Capítulo **6**

## Análisis

### 20. ESPECIFICACIÓN DE REQUERIMIENTOS

Este sección proporciona una completa especificación de requerimientos de software siguiendo el estándar 830 publicado por la IEEE (*IEEE Recommended Practice for Software Requirements Speciﬁcations* - *IEEE Std 830-1998*) con las modificaciones necesarias que lo adecuan al presente trabajo.

El propósito del presente SRS es definir los requerimientos principales tanto funcionales como no funcionales del software a desarrollar, sus interfaces, lo que el sistema realizará, las restricciones bajo las cuales deberá operar y cómo reaccionará a estímulos externos, sirviendo como contrato entre, y futura referencia para, el cliente y el equipo de desarrollo.

El documento está dirigido a las personas que utilicen el sistema, ya sean clientes, usuarios finales o el mismo equipo de desarrollo.

#### 20.1 Definiciones, acrónimos y abreviaturas

* SRS - System Requirements Specification / Especificación de requerimientos de sistemas.
* API - Application Programming Interface / Interfaz para la programación de aplicaciones.
* OS - Operating System - Sistema Operativo
* ABM - Altas, Bajas y modificaciones.
* BD - Base de datos

#### 20.2 Referencias

* Manual del sistema.
* IEEE Recommended Practice for Software Requirements Speciﬁcations (IEEE Std 830-1998)

#### 20.3 Funciones del sistema

Resumen de las principales funciones de la aplicación:

* Registro de nuevas órdenes de cliente.
* Presentación cronológica de los platos a preparar. (En cocina)
* Visualización de los ítems del menú en interfaz móvil (Fotos, Descripción, etc.)
* Gestión de órdenes, personal de atención e Ítems del menú. (ABM)
* Registro de reservas vía web.
* Cálculo de tiempo aproximado desocupación de mesas.
* Generación de reportes de balances, desempeño de personal e insumos consumidos.

#### 20.4 Características del usuario

El sistema distingue tres usuarios diferenciados del sistema. Estos usuarios serán reflejados en el sistema mediante roles.

**Personal de administración**: Solicitará la emisión de reportes y visualizará el estado de las ventas en tiempo real. La instrucción requerida se puede encontrar en el manual del sistema.

**Chef /Cocinero**: Solamente consumirá información de la pantalla de avisos que informa de forma cronológica, qué platos se están solicitando y sus cantidades.

**Personal de atención:** Usará la aplicación con alta frecuencia, principalmente para recolectar las órdenes provenientes de las mesas y notificarlas al sistema central. La instrucción requerida se puede encontrar en el manual del sistema.

**Cliente**: El cliente puede llegar a interactuar directamente con el sistema en el caso de que requiera ver una versión expandida del menú, el mozo puede facilitarle el dispositivo móvil para que recorra fotografías y descripciones de las ofertas del restaurante. No requiere instrucción alguna.

Este tipo de usuario puede, además, interactuar con la aplicación desde su interfaz web, a través de la cual podrá solicitar reservas, consultar el estado del restaurante y su oferta.

#### 20.5 Requisitos

##### 20.5.1 Requisitos Funcionales

**RF 01 – Registrar de Ordenes**

El sistema registrará las órdenes solicitadas por los clientes. El registro será llevado a cabo por el personal de atención al público e incluirá uno o más ítems del menú.

**RF 02 - Gestionar de Ítems de Menú**

El sistema permitirá la administración de los ítems de menú ingresados al sistema por parte del usuario, ya sea ingreso, salida o modificación de la información.

**RF 03 - Registrar de Mesas**

El sistema permitirá registrar las mesas disponibles en el local. La aplicación debe conocer las mesas disponibles para atención y así permitir al personal de atención identificar la mesa atendida.

**RF 04- Gestionar de Reservas**

El sistema registrará las reservas de mesas generadas por clientes

**FR 05 -  Devolver el tiempo estimado de disponibilidad.**

**RF 06 - Gestionar Cobros**

El sistema calculará el importe total de una Orden de compra y registrar el pago.

**RF 07 - Emisión de Facturas**

El sistema permitirá la registración e impresión de facturas correspondientes a cada orden

**RF 08 - Gestionar de Personal de Atención**

El sistema permitirá la creación, modificación y eliminación de personal de atención.

**RF 09 - Notificar a cocineros.**

El sistema enviará Información de Platos a preparar  a la cocina de forma cronológica

**RF 10 – Generar reportes estadísticos.**

El sistema generará gráficas y reportes sobre el desempeño de los clientes, platos solicitados e Ingresos diarios / mensuales / anuales.

##### 19.5.2 Requisitos no funcionales

Un requisito no funcional o atributo de calidad es un requisito que específica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar.

Existen diferentes tipos de requisitos no funcionales que pueden ser clasificados, entre otros, en requisitos no funcionales de:

RENDIMIENTO

**RNF 01 - Realización de consultas a la BD en tiempos menores a 3 segundos.**

Los dispositivos (tablets) deberán obtener información de la base de datos en un tiempo máximo no mayor a 5 segundos, sin importar el tipo de consulta.

**RNF 02 - Adaptación de la página web a varios dispositivos (Responsive Design)**

El diseño del sitio web debe poder ser visualizado desde distintos dispositivos

**RNF 03 - Todas las transacciones deben ser procesadas en menos de 5 segundos con excepción de las transacciones de análisis.**

Cualquier consulta a la base de datos debe ser realizada en menos de 5 segundos, no entran en esta clasificación la generación de reportes de balances mensuales, desempeño anual o mensual de empleados ni el reporte de insumos consumidos.

CONCURRENCIA

**RNF 04 - El sistema deberá soportar hasta 20 usuarios simultáneos.**

Los requisitos mencionados deberían mantenerse inmutados si se mantiene una cantidad de veinte usuarios (Dispositivos móviles) o menos. A partir de esa cantidad se aceptan reducciones en el desempeño del sistema.

INTERFAZ

RNF 05 - Las interfaces de usuarios de los dispositivos móviles deben ser intuitiva y fácil de utilizar

COMPATIBILIDAD

RNF - Con respecto a la aplicación para Android, habrá cierta compatibilidad hacia atrás, ya que hoy en día existen todavía muchos dispositivos corriendo versiones desactualizadas de este sistema operativo. RNF07 La aplicación móvil deberá poder correr sin inconvenientes en cualquier dispositivo Android cuya versión sea superior a la 2.2.3

SEGURIDAD

RNF 06 - Las claves de acceso de los administradores deberán almacenarse en la base de datos de forma cifrada, para evitar que usuarios malintencionados accedan al panel de control y comprometan el funcionamiento habitual del sistema.

PORTABILIDAD

La aplicación web deberá ser independiente del sistema operativo del usuario

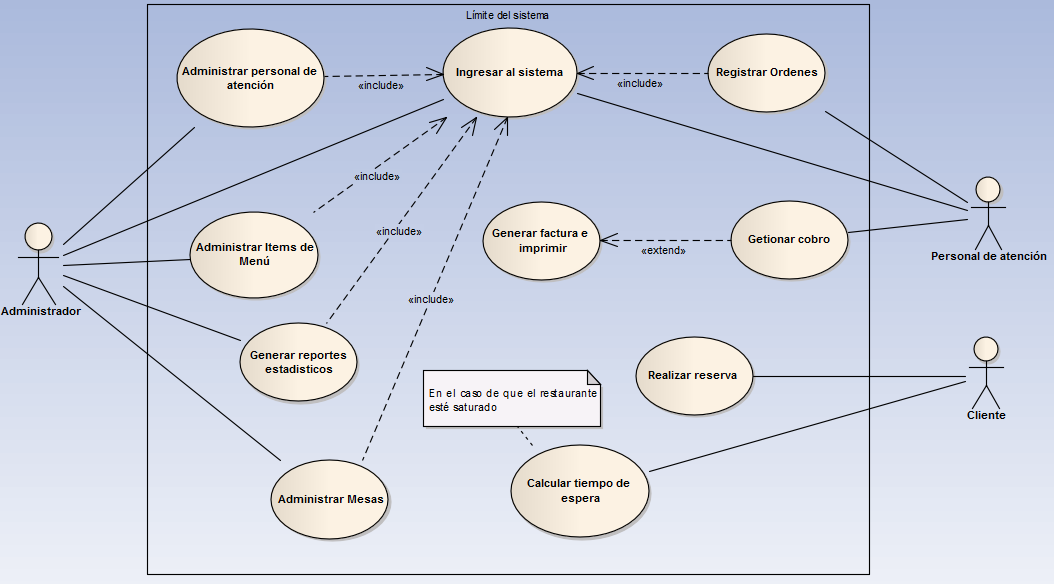
La aplicación web deberá poder ejecutarse sin problema bajo cualquier sistema operativo y en la última versión de los principales navegadores, a saber: Chrome, Firefox, Internet Explorer, Opera y Safari.

### 21. MODELO DE CASOS DE USO

#### 21.1 Diagrama de Caso de usos

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.



#### 21.2 Descripción de Caso de usos

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del caso de uso: Ingresar al sistema | | | #1 |
| Prioridad: | Alta | Media | Baja |
| Complejidad: | Alta | Media | Baja |
| Actor principal: Administrador, Personal de atención. | | Actor secundario: | |
| Tipo de caso de uso: | | Concreto | Abstracto |
| Objetivo: Llevar un control de los usuarios que ingresen a la aplicación. | | | |
| Precondiciones:   1. El usuario debe estar dado de alta en la aplicación. | | | |
| Postcondiciones: | Éxito: El usuario puede hacer uso de la aplicación. | | |
| Fracaso: El usuario no puede ser autentificado. | | |
| Curso normal: | | Curso alternativo: | |
| 1. Se inicia la aplicación. | |  | |
| 1. El usuario ingresa sus datos de logueo. | |  | |
| 1. La aplicación verifica que los tipos de datos ingresados son válidos. | | * 1. Se alerta sobre un campo sin completar o con caracteres no válidos.   2. Vuelve al paso 2. | |
| 1. Se comprueba la existencia del usuario en la base de datos. | | * 1. No se encuentra la combinación de nombre de usuario y contraseña en la base de datos.      1. Vuelve al paso 2.      2. Cancela la operación. Fin del CU. | |
| 1. El usuario ingresa a la aplicación. | |  | |
| 1. Fin del CU. | |  | |
| Observaciones: N/A | | | |
| Caso de uso que incluye: N/A | | | |
| Caso de uso al que extiende: N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del caso de uso: Administrar personal de atención / mesas / ítems de menú. | | | #2 |
| Prioridad: | Alta | Media | Baja |
| Complejidad: | Alta | Media | Baja |
| Actor principal: Administrador | | Actor secundario: | |
| Tipo de caso de uso: | | Concreto | Abstracto |
| Objetivo: crear, eliminar o modificar información sobre personal de atención / mesas / ítems de menú, en el sistema. | | | |
| Precondiciones:   1. El usuario debe estar autentificado en el sistema. 2. El usuario debe tener permisos de administrador. | | | |
| Postcondiciones: | Éxito: se crea, elimina o modifica información sobre personal de atención / mesas / ítems de menú. | | |
| Fracaso:   1. Creación: Si los datos ingresados son inválidos, el alta no se completará y el usuario recibirá un mensaje de error. 2. Eliminación: Si la entidad que se quiere eliminar no existe, la baja no se completará y el usuario recibirá un mensaje de error. 3. Modificación: Si la entidad que se quiere editar no existe, la modificación no se completará y el usuario recibirá un mensaje de error. Si los datos ingresados son inválidos, la modificación no se llevará a cabo y el sistema arrojará un mensaje de error. | | |
| Curso normal: | | Curso alternativo: | |
| Creación:   1. Ir a la opción correspondiente para dar de alta a una nueva entidad. | |  | |
| 1. El administrador ingresa los datos solicitados en el formulario. | |  | |
| 1. La aplicación verifica que los tipos de datos ingresados son válidos. | | * 1. Se alerta sobre un campo sin completar o con caracteres no válidos.   2. Vuelve al paso 2. | |
| 1. Los datos se almacenan en la base de datos del sistema. | |  | |
| 1. Fin del caso de uso. | |  | |
| Modificación:   1. Ir a la opción correspondiente para modificar una entidad ya existente | |  | |
| 1. El sistema traerá los datos de la entidad seleccionada | |  | |
| 1. El administrador modificará los datos necesarios. | |  | |
| 1. La aplicación verifica que los tipos de datos ingresados son válidos. | | * 1. Se alerta sobre un campo sin completar o con caracteres no válidos.   2. Vuelve al paso 3. | |
| 1. Los cambios realizados se guardan en la base de datos. | |  | |
| 1. Fin del CU | |  | |
| Eliminación   1. Dirigirse a la opción correspondiente para dar de baja a una entidad existente. | |  | |
| 1. La entidad seleccionada se elimina de la base de datos. | |  | |
| 1. Fin del caso de uso. | |  | |
| Observaciones: N/A | | | |
| Caso de uso que incluye: Ingresar al sistema | | | |
| Caso de uso al que extiende: N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del caso de uso: Registrar ordenes | | | #3 |
| Prioridad: | Alta | Media | Baja |
| Complejidad: | Alta | Media | Baja |
| Actor principal: Personal de atención. | | Actor secundario: Cliente | |
| Tipo de caso de uso: | | Concreto | Abstracto |
| Objetivo: Listar cantidad de los ítems de menú solicitados por cada cliente en sus respectivas mesas e informar al personal de cocina sobre dichas solicitudes. | | | |
| Precondiciones: El personal de atención a cargo del registro debe estar logeado en el sistema. | | | |
| Postcondiciones: | Éxito: La orden es enviada a cocina. | | |
| Fracaso: La orden no es procesada, el personal de cocina no es informado sobre la solicitud | | |
| Curso normal: | | Curso alternativo: | |
| 1. El personal de atención (PE) selecciona una mesa disponible, elije la opción ‘Nueva orden’ y especifica la cantidad de personas. | | * 1. El personal de atención selecciona una mesa ocupada.   2. Continúa en el paso 2 del curso normal. | |
| 1. El PE asigna múltiples ítems de menú, dependiendo de sus solicitudes. | |  | |
| 1. El PE confirma los pedidos y envía la orden a la cocina. | |  | |
| 1. Fin de caso de uso | |  | |
| Observaciones: N/A | | | |
| Caso de uso que incluye: Ingresar al sistema | | | |
| Caso de uso al que extiende: N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del caso de uso: Gestionar cobro | | | #5 |
| Prioridad: | Alta | Media | Baja |
| Complejidad: | Alta | Media | Baja |
| Actor principal: Personal de atención. | | Actor secundario: Cliente | |
| Tipo de caso de uso: | | Concreto | Abstracto |
| Objetivo: Asentar el cobro y dar por finalizada la atención a clientes | | | |
| Precondiciones:   1. El personal de atención a cargo del registro debe estar logeado en el sistema. 2. Debe existir una orden actualmente asignada a la mesa a cobrar | | | |
| Postcondiciones: | Éxito: Se registra un ingreso | | |
| Fracaso: La orden no es procesada, el personal de cocina no es informado sobre la solicitud | | |
| Curso normal: | | Curso alternativo: | |
| 1. El personal de atención (PE) selecciona la opción ‘Nueva orden’ y especifica número de mesa y cantidad de personas. | | * 1. El personal de atención selecciona una orden existente.   2. Continúa en el paso 2 del curso normal. | |
| 1. Por cada uno de los clientes, el PE asigna múltiples ítems de menú, dependiendo de sus solicitudes. | |  | |
| 1. El PE confirma los pedidos y envía la orden a la cocina. | |  | |
| 1. Fin de caso de uso | |  | |
| Observaciones: N/A | | | |
| Caso de uso que incluye: Ingresar al sistema | | | |
| Caso de uso al que extiende: N/A | | | |

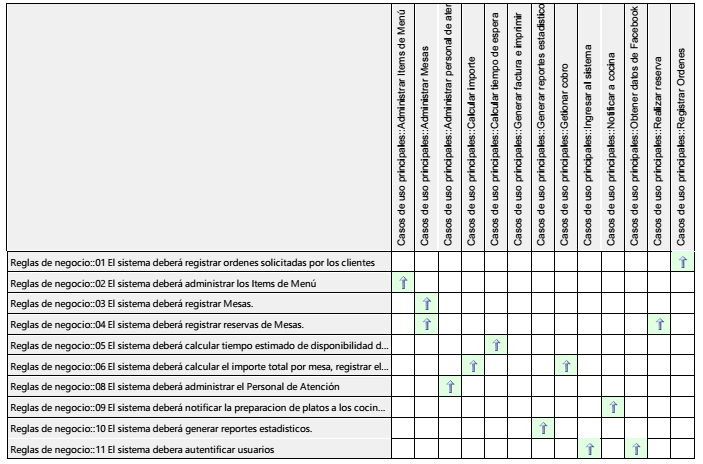
|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del caso de uso: Realizar reserva | | | #6 |
| Prioridad: | Alta | Media | Baja |
| Complejidad: | Alta | Media | Baja |
| Actor principal: Cliente. | | Actor secundario: | |
| Tipo de caso de uso: | | Concreto | Abstracto |
| Objetivo: Reservar un mesa para un día y horario en particular | | | |
| Precondiciones: | | | |
| Postcondiciones: | Éxito: Se guarda en la base de datos una nueva entrada que especifica fecha y horario de la reserva. | | |
| Fracaso: El usuario recibe un mensaje de error indicando que la solicitud no fue procesada. | | |
| Curso normal: | | Curso alternativo: | |
| 1. Un potencial cliente se identifica en la plataforma web de la aplicación mediante la red social Facebook/Google. | |  | |
| 1. Se dirige a la sección de reservas | |  | |
| 1. Selecciona una fecha, horario y cantidad de personas. | |  | |
| 1. La aplicación verifica la disponibilidad en el restaurante para la hora especificada | |  | |
| 1. Se confirma y registra la solicitud de reserva | | * 1. No se encuentra disponibilidad para le fecha, horario o cantidad de personas especificadas.      1. Retorna al paso 3.      2. El usuario cancela la operación. Fin de caso de uso | |
| 1. Fin de caso de uso | |  | |
| Observaciones: N/A | | | |
| Caso de uso que incluye: N/A | | | |
| Caso de uso al que extiende: N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del caso de uso: Generar reportes estadísticos | | | #7 |
| Prioridad: | Alta | Media | Baja |
| Complejidad: | Alta | Media | Baja |
| Actor principal: Administrador | | Actor secundario: | |
| Tipo de caso de uso: | | Concreto | Abstracto |
| Objetivo: Generar información relevante para la toma de decisiones. | | | |
| Precondiciones: El usuario debe estar logeado como administrador | | | |
| Postcondiciones: | Éxito: Una vez autenticado, el usuario puede conocer las estadísticas y descargar / imprimir aquellos reportes que le resulten de interés. | | |
| Fracaso: El administrador recibe un mensaje de error indicando que la solicitud no fue procesada. | | |
| Curso normal: | | Curso alternativo: | |
| 1. El administrador se dirige a la sección ‘Reportes’ | |  | |
| 1. Selecciona el tipo de reporte a generar e indica los parámetros necesarios. | |  | |
| 1. El sistema procesa la solicitud y muestra la información en pantalla | |  | |
| 1. Fin de caso de uso | |  | |
| Observaciones: N/A | | | |
| Caso de uso que incluye: N/A | | | |
| Caso de uso al que extiende: N/A | | | |

#### 21.3 TRAZABILIDAD DE REQUERIMIENTOS Y CASOS DE USO

A continuación, se muestra la matriz de incidencia que representa la trazabilidad bidireccional entre los requerimientos del sistema y los componentes que intervienen en todas las etapas del desarrollo. El objetivo de diagramar la trazabilidad es determinar si todos y cada uno de los requerimientos del sistema son contemplados en solución propuesta. A su vez, permite asegurar que toda funcionalidad posee un requerimiento inicial que le dio origen y la justifica.

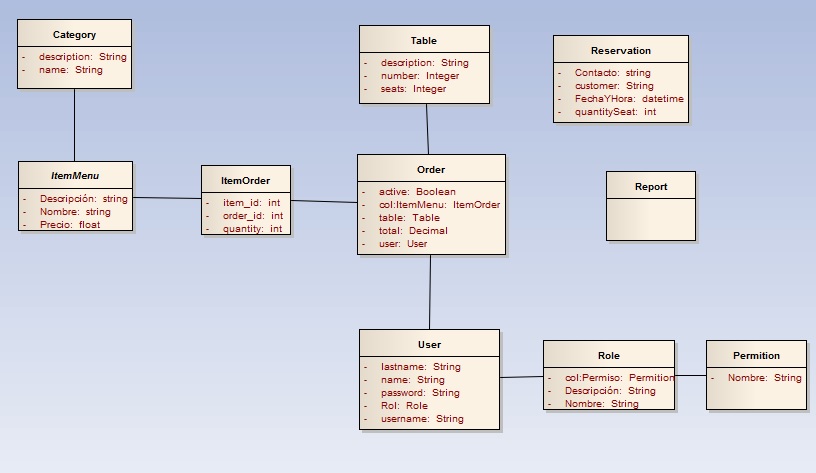
Por otro lado, también es una herramienta útil a la hora de evaluar el impacto de diferentes pedidos de cambio sobre los requerimientos que van surgiendo durante el desarrollo del proyecto. Es decir, ante un cambio en los requerimientos, se busca determinar cómo se verán afectados los tiempos y costos del proyecto a partir del impacto en la arquitectura de la solución, el diseño de los componentes, las tareas asignadas, etc.



### 22. MODELO DE DOMINIO

#### 22.1 Diagrama de Clases de Análisis

A continuación se presenta el diagrama de clases de análisis de la aplicación en desarrollo.

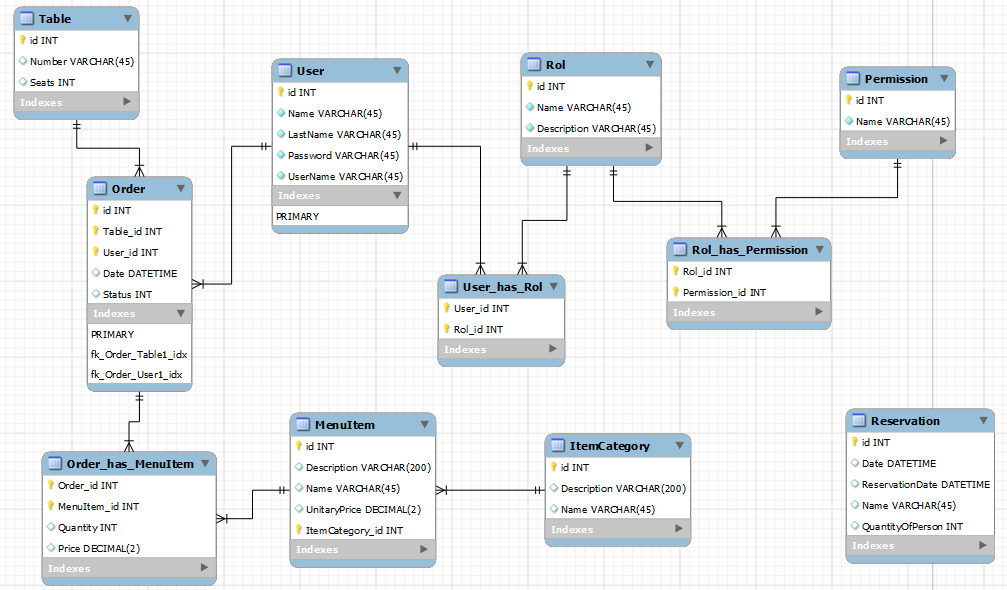


# Capítulo **7**

## Diseño

### 23. DISEÑO DEL SISTEMA

23.1 Modelo de Datos

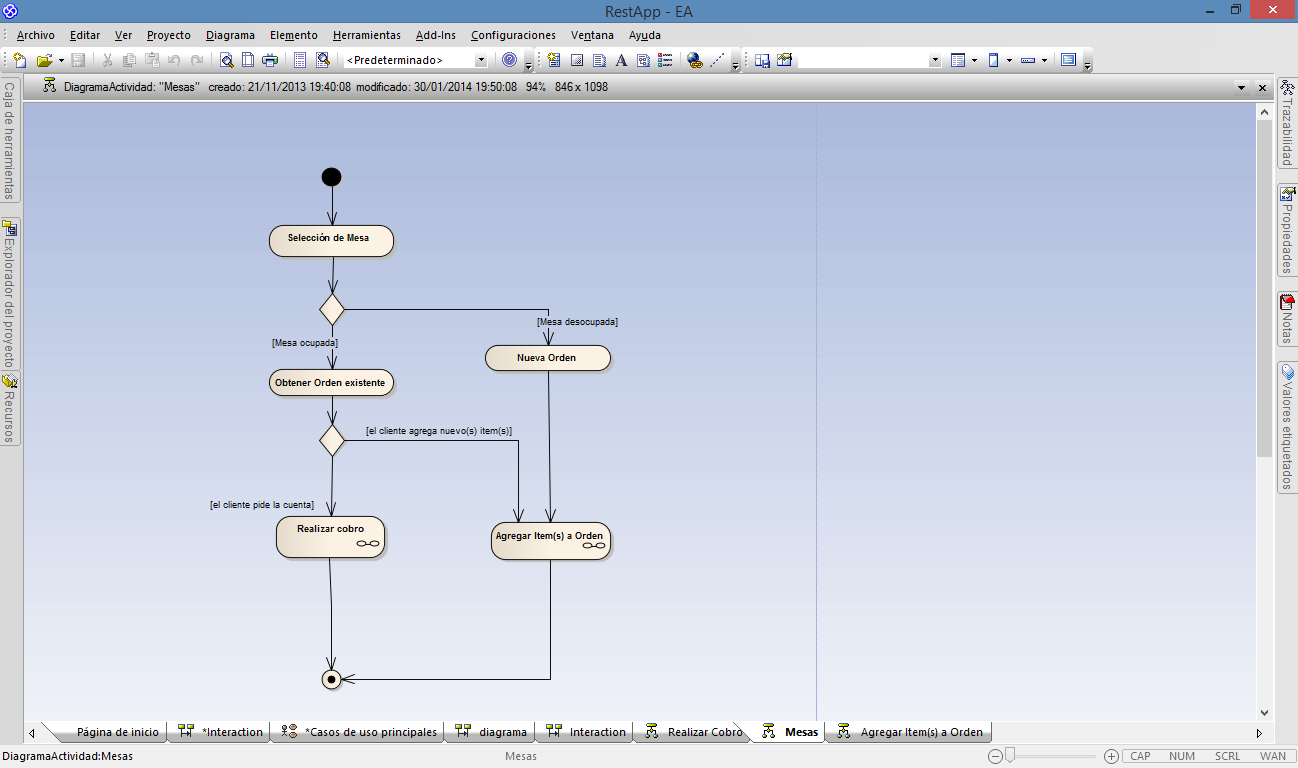
El siguiente es el diagrama Entidad-Relación (DER) de la aplicación:

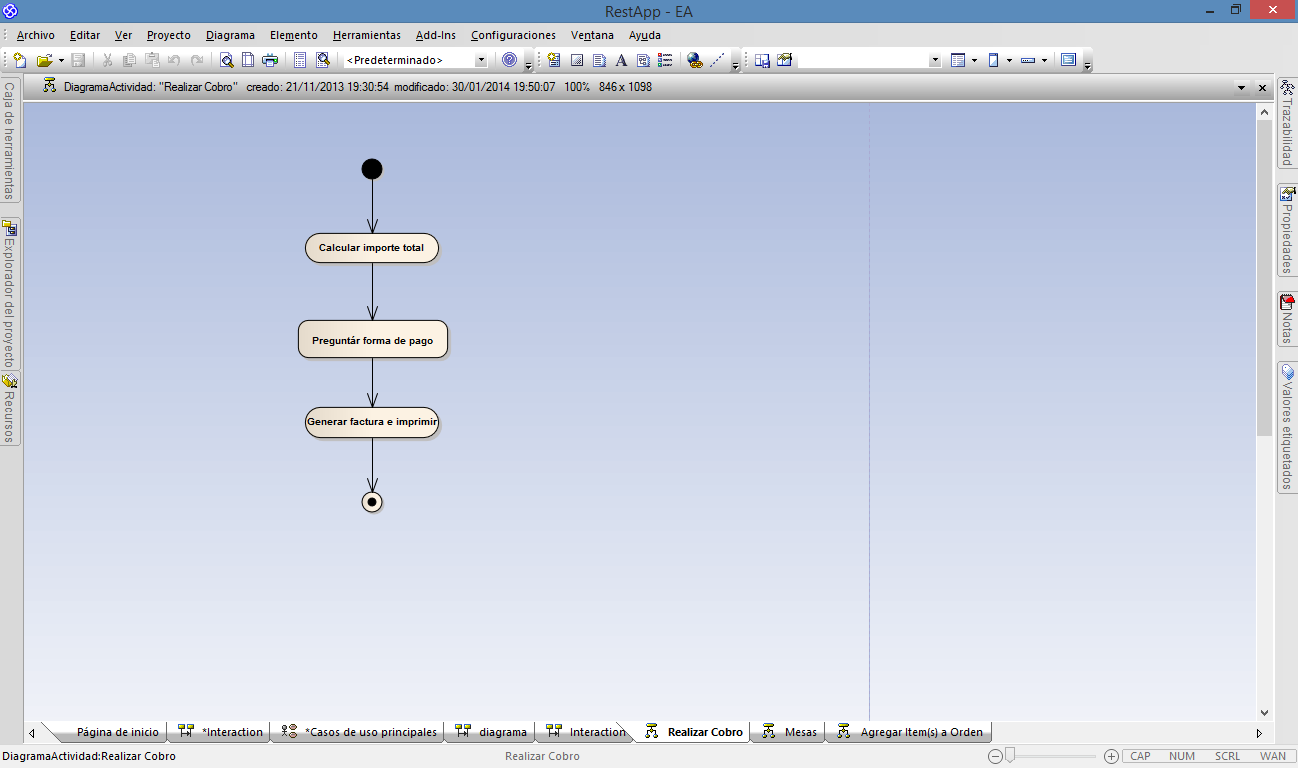
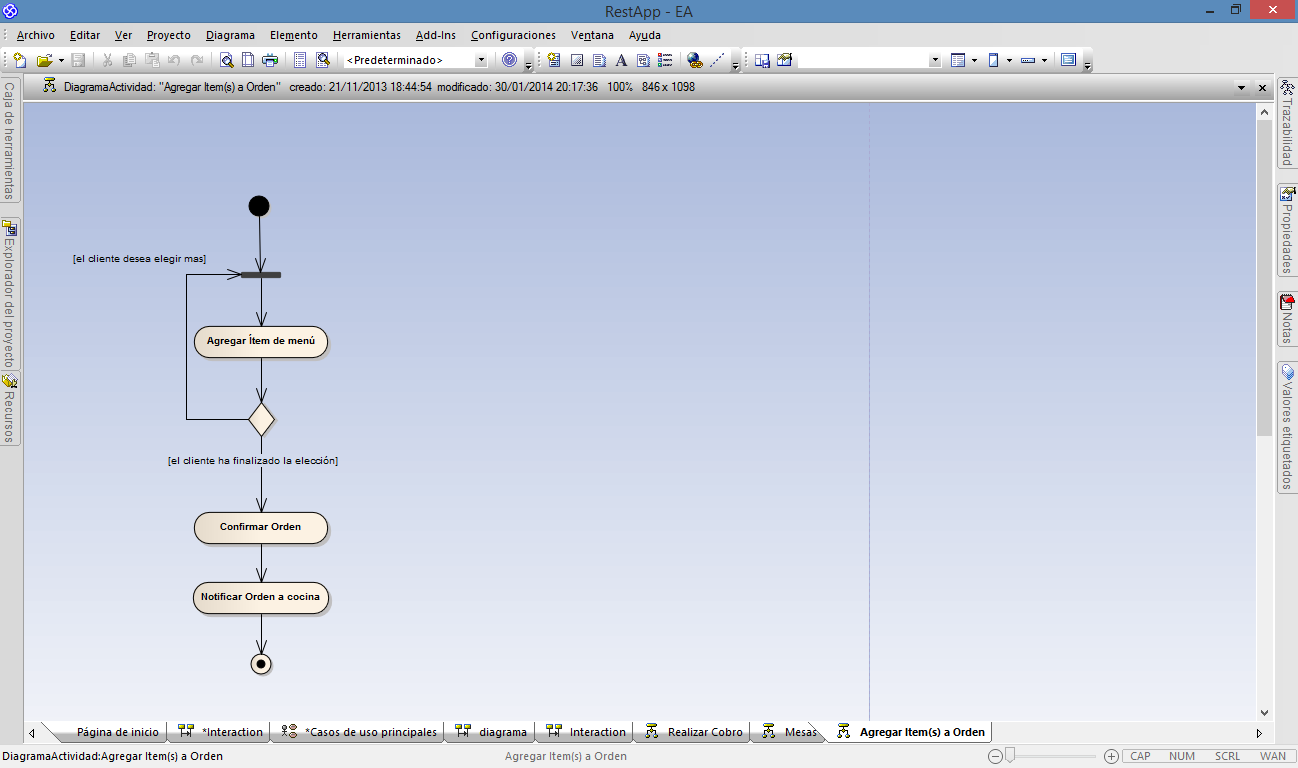
#### 23.2 Diagrama de secuencia

A continuación se incluyen algunos de los diagramas de secuencia correspondientes a los procesos más relevantes de la aplicación RestApp.

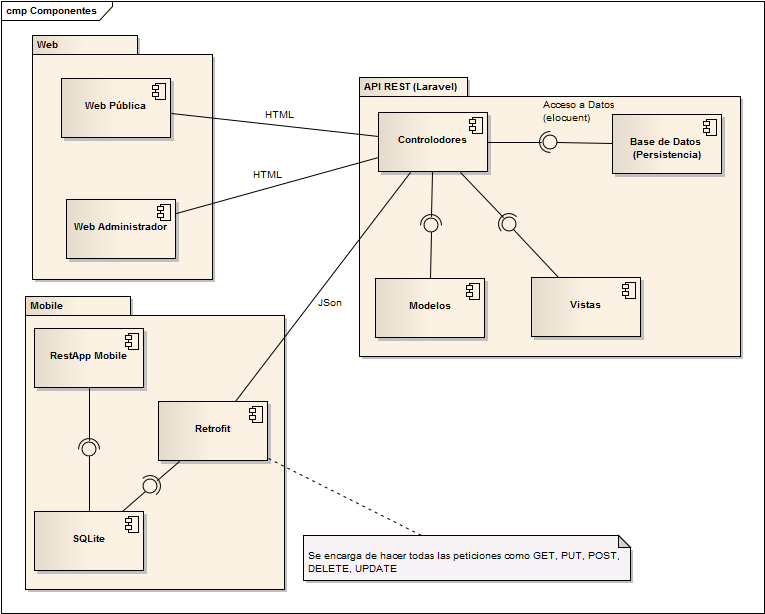
[Ver diagrama en EA (3)]

23.3 Diagrama de Actividad





23.4 Modelo de componentes



# Capítulo **8**

## Implementación

### DETALLES DE LA IMPLEMENTACIÓN

Este capítulo tiene como objetivo brindar detalles de la implementación indicando software, hardware, tecnologías utilizadas, frameworks, librerías y APIs utilizadas para el desarrollo y ejecución de la aplicación.

### REQUISITOS HARDWARE

A continuación se definen los requisitos mínimos para la ejecución de RestApp:

* PC o servidor configurado con apache. (ESPECIFICAR)
* PC con navegador Web, preferiblemente Mozilla Firefox, Google Chrome, Internet Explorer versión 8 o superior.
* Tablets: mínimo una, con sistema operativo Android.
* Conexión a internet o configurar una LAN entre los dispositivos a utilizar.

### BASE DE DATOS

La base de datos que utiliza la aplicación es MySQL.

### APLICACIÓN ANDROID

Android es un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles (smartphones o tablets) con pantalla táctil. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google.

La aplicación para dispositivos Android se desarrolló usando el SDK de , que provee de todas las librerías y herramientas necesarias a la hora Android7 de construir, probar y depurar aplicaciones para Android. El SDK de Android proporciona también un emulador de teléfono, así como documentación y tutoriales con ejemplos muy útiles. Las plataformas de desarrollo soportadas por Android SDK incluyen cualquier distribución moderna de Linux, Max OS X 10.4.9 o posterior, y Windows XP o posterior. El IDE que se usará es Android , que es un entorno de desarrollo oficial desarrollado por Google Studio8 basado en IntelliJ IDEA.