

“Sistema para reservar y comprar menú de restaurantes en línea”

DANIEL FERNÁNDEZ GONZÁLEZ

OCTAVIO VALENCIA VÁSQUEZ

**PROYECTO DE TÍTULO PARA OPTAR AL TÍTULO**

**DE INGENIERO EN COMPUTACIÓN E INFORMÁTICA**

VIÑA DEL MAR – CHILE

AGOSTO 2014

Tabla de contenido

[CAPÍTULO 1 0](#_Toc397346806)

[**1.1.** **Introducción** 5](#_Toc397346807)

[**1.2.** **Definición del problema** 6](#_Toc397346808)

[**1.3.** **Objetivo General acorde al problema** 8](#_Toc397346809)

[CAPITULO 2 9](#_Toc397346810)

[**2.1.** **Solución propuesta** 10](#_Toc397346811)

[**2.2.** **Alcances** 10](#_Toc397346812)

[**2.3.** **Limitaciones** 10](#_Toc397346813)

[**2.4.** **Alternativas de solución** 11](#_Toc397346814)

[CAPITULO 3 13](#_Toc397346815)

[**3.1.** **Estilo arquitectónico** 14](#_Toc397346816)

[**3.2.** **Diseño de alto nivel** 16](#_Toc397346817)

[3.2.1. **Vista Lógica** 18](#_Toc397346818)

[3.2.2. **Vista Despliegue** 19](#_Toc397346819)

[3.2.3. **Vista física** 19](#_Toc397346820)

[3.2.4. **Vista de Procesos** 20](#_Toc397346821)

[3.2.5. **Diseño de la Base de datos** 23](#_Toc397346822)

[**3.3.** **Requerimientos** 4](#_Toc397346823)

[3.3.1. **Requerimientos funcionales** 4](#_Toc397346824)

[3.3.2. **Requerimientos no funcionales** 5](#_Toc397346825)

[3.3.3. **Trazabilidad de requerimientos** 6](#_Toc397346826)

[**3.4.** **Explicación Metodología Seleccionada** 4](#_Toc397346827)

[3.4.1. **SCRUM**: Para la gestión del proyecto 4](#_Toc397346828)

[3.4.2. **Roles de Scrum** 5](#_Toc397346829)

[3.4.3. **Ventajas y desventajas de Scrum** 6](#_Toc397346830)

[3.4.4. **XP (Xtreme Programming):** Para el desarrollo del proyecto. 7](#_Toc397346831)

[3.4.5. **Roles en metodología XP:** 8](#_Toc397346832)

[3.4.6. **Ventajas y desventajas de XP** 8](#_Toc397346833)

[3.4.7. **Combinación de roles entre Scrum y XP** 9](#_Toc397346834)

[**3.5.** **Plan de proyecto (incluir planificación otras métricas)** 4](#_Toc397346835)

[3.5.1. **Planificación de los Sprint** 4](#_Toc397346836)

[3.5.2. **Planificación de las tareas** 4](#_Toc397346837)

[3.5.3. 4](#_Toc397346838)

[**3.6.** **Plan de Pruebas** 5](#_Toc397346839)

[**3.7.** **Plan de Gestión de Configuración** 6](#_Toc397346840)

[**3.8.** **Plan de Gestión de Cambios** 7](#_Toc397346841)

[**3.9.** **Plan de Gestión de Riesgos** 8](#_Toc397346842)

[**3.10.** **Ambiente de desarrollo, pruebas y producción** 9](#_Toc397346843)

[CAPITULO 4 10](#_Toc397346844)

[RESULTADO Y DISCUSIÓN 10](#_Toc397346845)

[Análisis 11](#_Toc397346846)

[Diseño detallado 12](#_Toc397346847)

[Pruebas Desarrollador 13](#_Toc397346848)

[Pruebas de Sistema (funcionales y de rendimiento) 14](#_Toc397346849)

[Pruebas de Aceptación (al menos usabilidad) 15](#_Toc397346850)

[Automatización (si corresponde) y regresión 16](#_Toc397346851)

[Evidencias de Liberación del producto 17](#_Toc397346852)

[Evidencias de Control de versiones 18](#_Toc397346853)

[Evidencias de Control de cambios 19](#_Toc397346854)

[CAPITULO 5 20](#_Toc397346855)

[CONCLUSIONES 20](#_Toc397346856)

[PostMortem 21](#_Toc397346857)

[Problemas Abiertos 22](#_Toc397346858)

[Trabajo Futuro 23](#_Toc397346859)

[4. Bibliografía 24](#_Toc397346860)

INDICE DE TABLAS

**INDICE DE FIGURAS**

No se encuentran elementos de tabla de ilustraciones.

**CAPÍTULO 1**

**Introducción**

* 1. **Introducción**

En muchas ocasiones nuestro tiempo disponible para consumir alimentos es restringido, sobre todo en trabajos en los que se cuenta con solo 45 minutos de colación o tal vez menos; es problemático el hecho de tener que cumplir en sólo unos de minutos la necesidad diaria de almorzar en locales y restaurantes sumamente llenos. El tiempo de espera es largo, ya sea por la espera en que el garzón nos atienda o la demora en la cocina al preparar los alimentos. La entrega tardía de los pedidos nos obliga a digerir nuestros alimentos rápidamente, lo que es perjudicial tanto en factores de estrés y/o alimenticios.

Se necesita lograr optimizar estos procesos y aprovechar de mejor manera las horas que disponemos para comer o compartir en locales, disminuyendo tiempos de espera.

* 1. **Definición del problema**

En la actualidad las personas que asisten a restaurantes, deben lidiar con el problema de disponibilidad de mesa, esperar para su atención por parte de los garzones y entrega del menú solicitado. Además en ocasiones se limitan a entrar a algunos restaurantes por prejuicio de precio, ya que al observar el local por fuera se tiende a creer que porque tiene un diseño sofisticado, sus precios serán elevados.

También está el caso de las personas que trabajan y tienen poco tiempo para almorzar.

Por otra parte los restaurantes requieren constantes estrategias de marketing para generar más ingresos y atraer nueva clientela.



Figura 1.1 Diagrama causa y efecto Ishikawa de problema general

Figura 1 1

Figura 1.1

Figura 1 2:Diagrama causa y efecto Ishikawa de problema general

Figura 1 3

En este proyecto se consideró solucionar las causas de los efectos de los clientes y restaurantes, los cuales son participe de provocar el problema principal:



Figura 1.2

Figura 1.2 Diagrama causa y efecto Ishikawa indicando cuales causas va a solucionar el proyecto.

Identificación de causas del problema:

CA1 : Desconocimiento de ubicación de restaurantes.

CA1.1: Pérdida de tiempo buscando restaurantes.

CA2 : Escases de tiempo para almorzar.

CA2.1: Tiempo de 45 minutos a 1 hora de almuerzo.

CA3 : Poca paciencia en la espera de atención.

CA4 : Indecisión al elegir por falta de información de los restaurantes.

CA5 : Falta de información de ubicación de los restaurantes

CA6 : Falta de información de la disponibilidad de mesas de los restaurantes.

* 1. **Objetivo General acorde al problema**

Disminuir los tiempos a 0 minutos de los clientes desde cuando llegan al restaurant, hasta cuando recibe su menú solicitado.

* 1. **Objetivos Específicos**

OBJ1: Dar información de los restaurantes de la V Región.

OBJ2: Entregar la ubicación de los restaurantes mediante la geolocalización.

OBJ3: Entregar información de los menús de los restaurantes asociados a la aplicación.

OBJ4: Otorgar la disponibilidad de las mesas de un restaurant seleccionado.

OBJ5: Desarrollar un sistema en donde los clientes puedan buscar un restaurant por geolocalización y solicitar un menú en línea.

Matriz de objetivos específicos versus las causas del problema a satisfacer:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Obj. Causas | CA1 | CA1.1 | CA2 | CA2.1 | CA3 | CA4 | CA5 | CA6 |
| OBJ1 |  | **X** |  |  |  | **X** | **X** |  |
| OBJ2 | **X** |  |  |  |  |  |  |  |
| OBJ3 |  |  | **X** | **X** |  |  |  |  |
| OBJ4 |  |  |  | **X** |  |  |  | **X** |
| OBJ5 |  |  | **X** | **X** | **X** |  |  |  |

Tabla 1.1

Tabla 1 1

***¡Error! No se encuentra el origen de la referencia.***

Tabla 1.1 Matriz de objetivos específicos versus causas del problema.

**CAPITULO 2**

**FUNDAMENTACIÓN DEL PROBLEMA**

* 1. **Solución propuesta**

Crear un sistema en el que los usuarios podrán buscar un restaurant por geolocalización, elegirlo, reservar la mesa y comprar su menú en línea para luego ir al restaurant y no deba esperar por la atención y posterior entrega de su menú.

El cliente al realizar la reserva en línea, se le entregará un ticket con la información respectiva a la reserva, además esa información será enviada por el sistema vía email al restaurant para que éste ya sepa que menú reservar, con fecha, hora y datos del cliente.

* 1. **Alcances**

Este proyecto tendrá las siguientes características:

* Mostrar los restaurantes cerca de su posición a través de geolocalización.
* Mostrar disponibilidad de mesas en los restaurantes ingresados al sistema.
* Permitir al usuario/cliente reservar una mesa.
* Mostrar los menús de los restaurantes asociados a la aplicación.
* Permitir al usuario/cliente poder pagar en línea el menú seleccionado.
* Atraer clientes a los restaurantes, a través de esta aplicación.
  1. **Limitaciones**

Cabe destacar que las limitaciones de este proyecto están supeditadas a lo anteriormente expuesto. Por lo que no solucionará los problemas del tiempo que el cliente ocupe ya sea en el traslado hacia el restaurant ni de lo que se demore en digerir su alimento. Además, estará acotado a los restaurantes de la zona de Viña del Mar.

* 1. **Alternativas de solución**

En la actualidad ya existe en el mercado nacional algunas páginas que entrega servicio de reserva a los clientes, las cuales se nombran a continuación:

* [www.restorando.cl](http://www.restorando.cl): Ofrece una variedad de restaurantes de Santiago y V región, con la limitación de que no muestra los menús, las reservas se pueden hacer con cualquier nombre de usuario y no tiene pago en línea.
* [www.atrapalo.cl](http://www.atrapalo.cl): Ofrece una variedad de restaurantes a nivel nacional, con la limitación de que no muestra los menús, las reservas se pueden hacer con cualquier nombre de usuario y no tiene pago en línea.
* [www.resermap.cl](http://www.resermap.cl): Ofrece una variedad de restaurantes de Santiago, con la limitación de que no muestra los menús, las reservas se pueden hacer con cualquier nombre de usuario y no tiene pago en línea.
* <http://www.divertimento.cl/reservas/>: Es un restaurant que para reservar se debe enviar un mail para que le envíen una confirmación de disponibilidad.
  1. **Factibilidades**

Para verificar si este proyecto es viable contemplamos una investigación inicial que consistió en realizar estadísticas y encuestas a los grupos sociales e interesados para ver las necesidades o requerimientos a los cuales deberá ir dirigido:

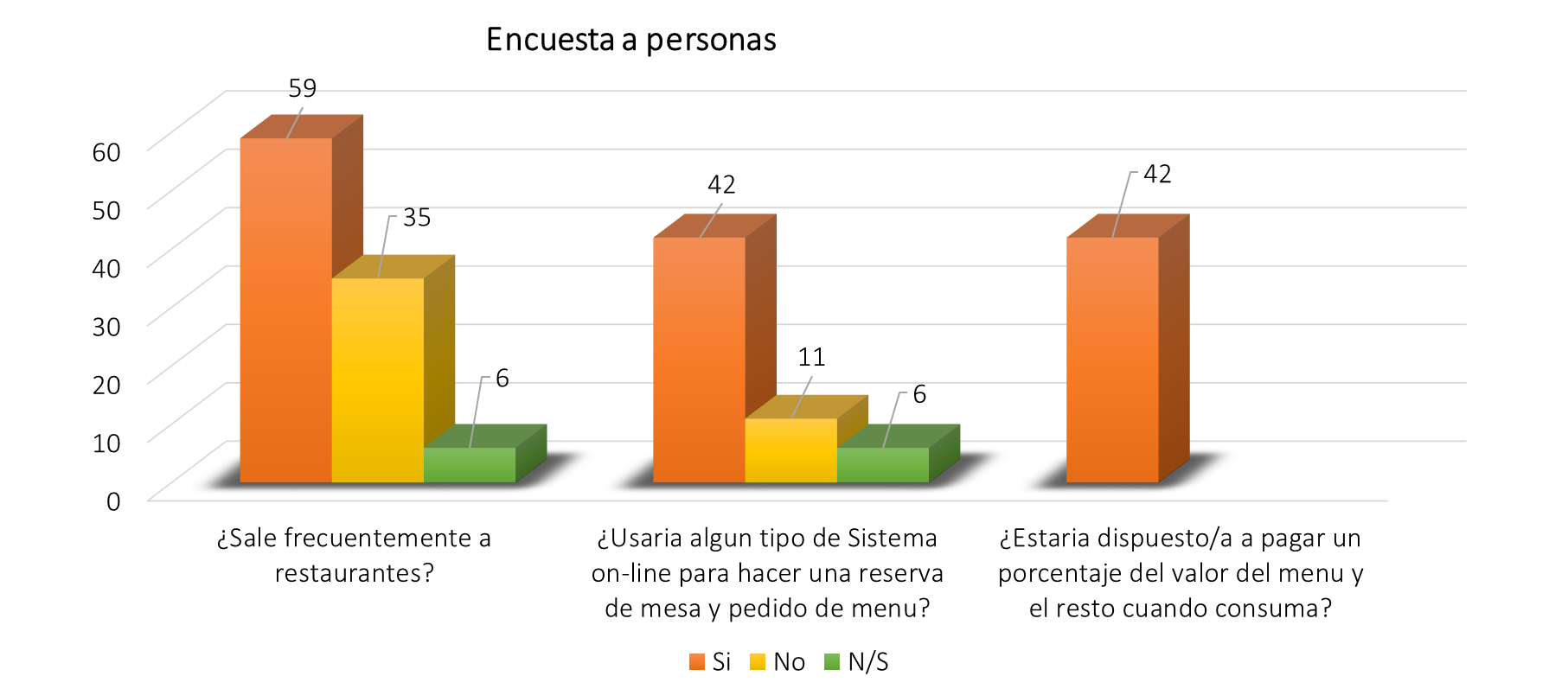


Figura 2.1 Gráfico de encuesta a personas de la ciudad de Viña del Mar.

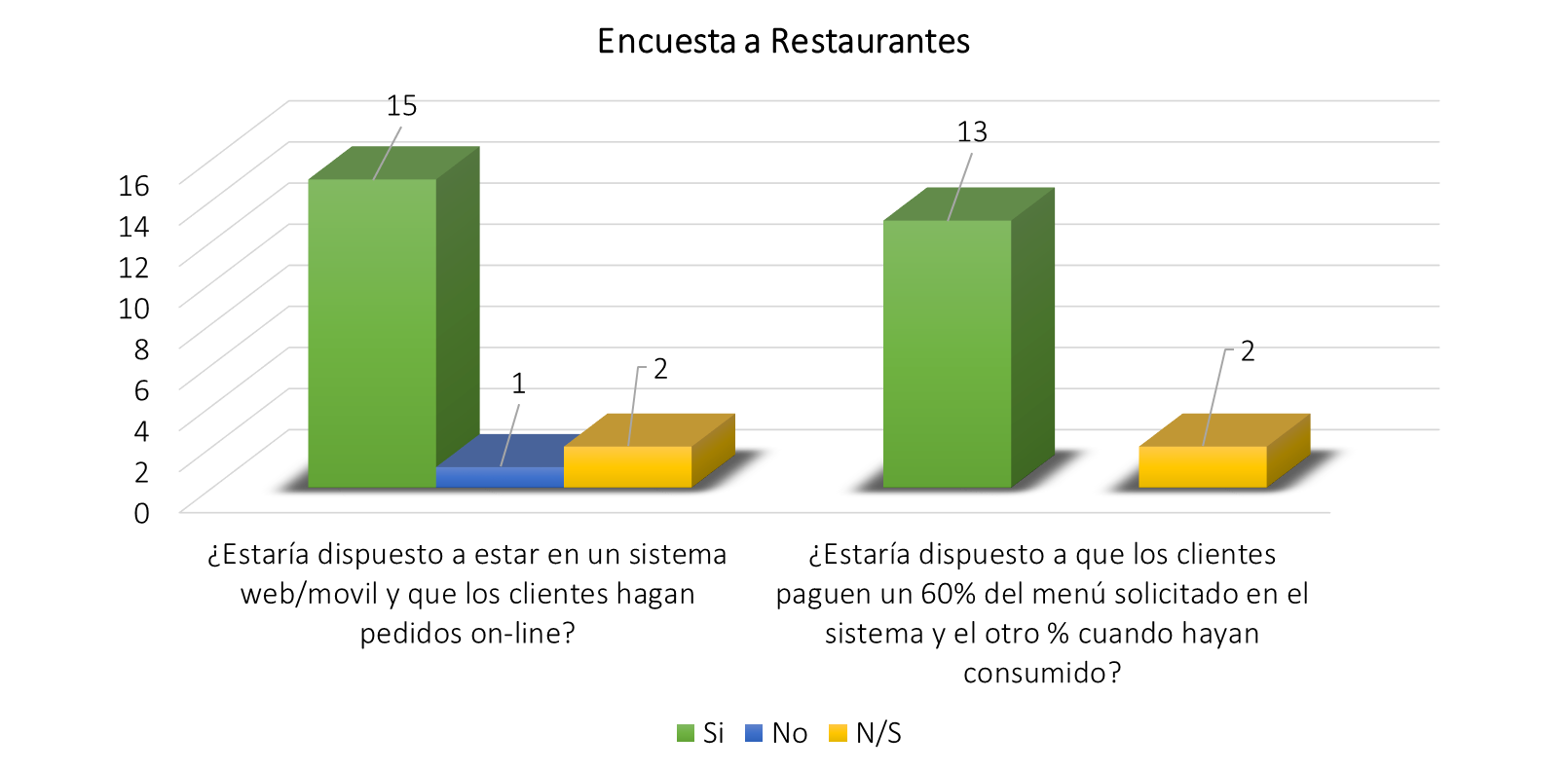


Figura 2.2 Gráfico de encuesta a restaurantes de la ciudad de Viña del Mar.

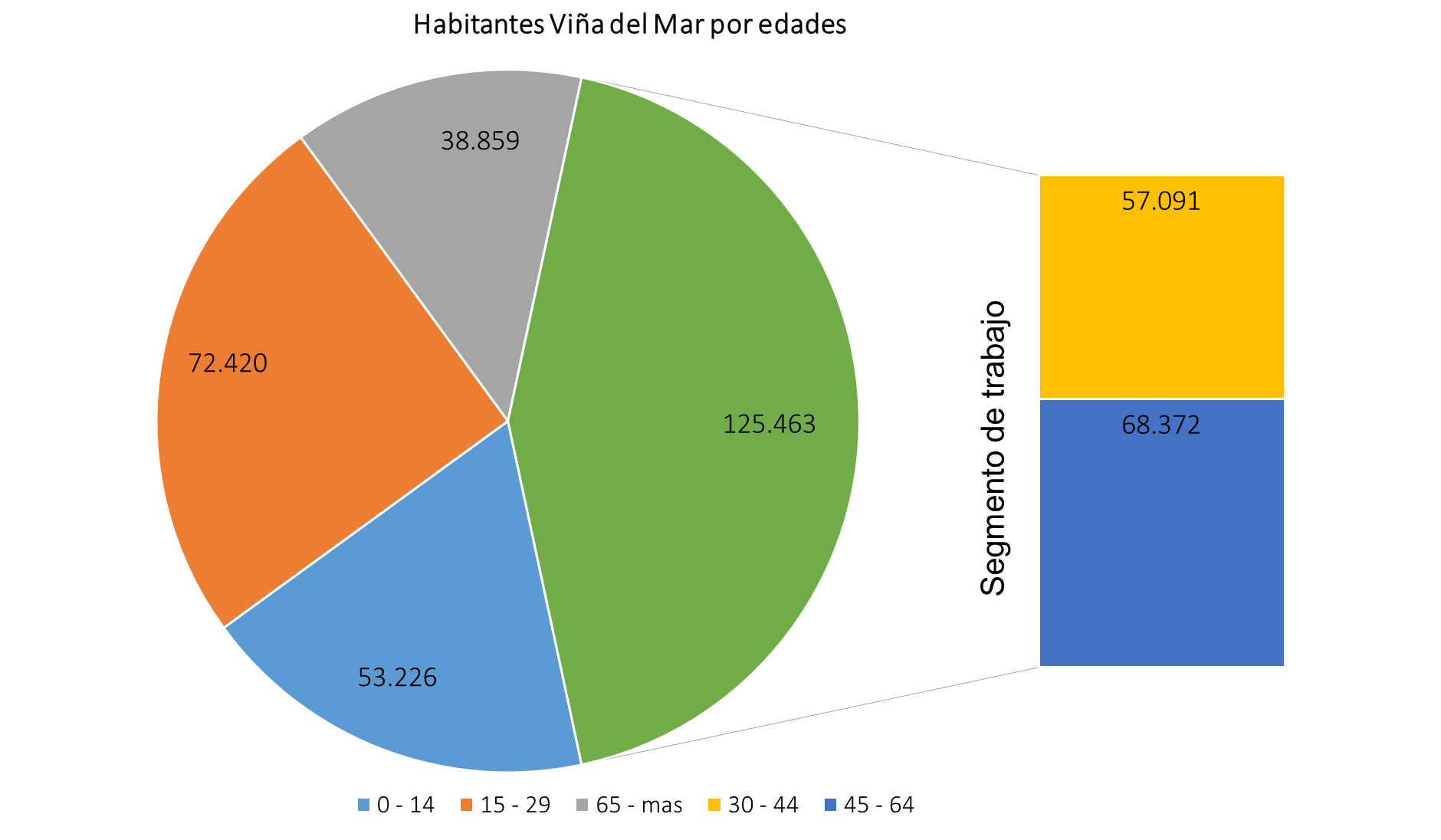


Figura 2.3 Gráfico de la cantidad de habitantes de la ciudad de Viña del Mar.

*Fuente Instituto Nacional de Estadísticas año 2012*

**CAPITULO 3**

**MATERIALES Y METODOS**

* 1. **Estilo arquitectónico**

Mostramos a continuación el esquema de la arquitectura lógica que se ha seguido en el desarrollo del proyecto.

Se trata del patrón MVC (Modelo-Vista-Controlador), una arquitectura orientada a desacoplar la interfaz de usuario (de entrada y presentación) aislándola de la lógica de negocio de la aplicación, y del modelo de datos implementado, permitiendo así que cada una sea desarrollada, probada y mantenida de forma independiente.

Siguiendo este patrón de diseño, la arquitectura del software queda separada en tres capas, según se puede ver en el diagrama.

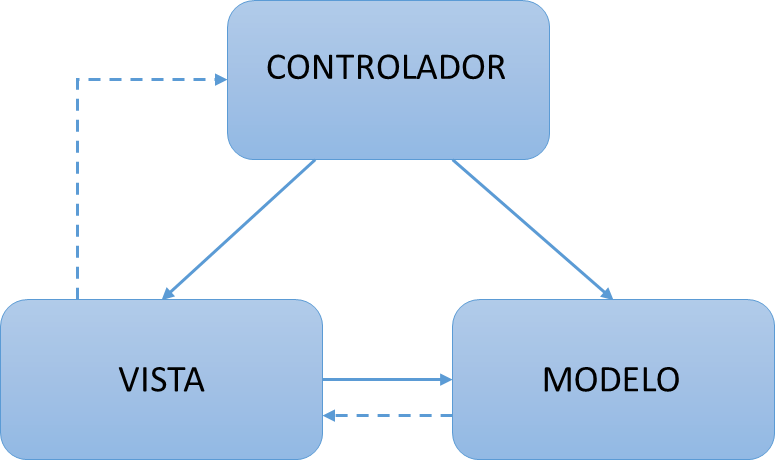
****

Figura 3.4 Diagrama Modelo Vista Controlador.

* **Modelo**

Se trata de la representación específica de la información o modelo de datos con el cual opera el sistema. En este caso el modelo es el encargado de mapear la información almacenada en las distintas clases que representan las entidades, pero es independiente del sistema de almacenamiento que se implementa en una base de datos centralizada con MySQL.

En el modelo también quedan definidas algunas reglas de negocio enfocadas a las validaciones haciendo uso de la técnica de Ruby on rails que es Rspec para la aplicación de TDD.

* **Vista**

Se encarga de la presentación al usuario del modelo en un formato adecuado para que este pueda interactuar, lo que habitualmente se conoce como interfaz de usuario.

La vista recibe los datos del modelo y los presenta al usuario, recogiendo la información introducida por éste y reaccionando, según las funciones y métodos implementados en el controlador, a los distintos eventos lanzados por el usuario.

* **Controlador**

Es el encargado de gestionar tanto la vista como el modelo, respondiendo a los eventos que recibe del usuario, invocando las diferentes peticiones al modelo y proporcionando la vista adecuada como respuesta.

* 1. **Diseño de alto nivel**

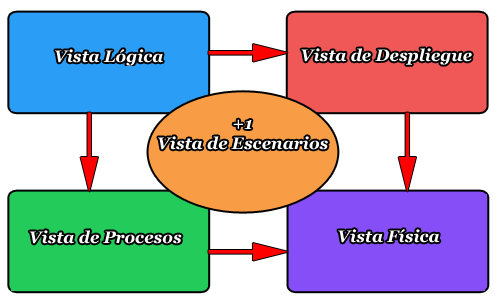
La arquitectura software trata el diseño e implementación de la estructura de alto nivel del software. Es el resultado de ensamblar un cierto número de elementos arquitectónicos para satisfacer la funcionalidad y ejecución de los requisitos del sistema; así como los requisitos no funcionales del mismo: fiabilidad, escalabilidad, portabilidad, disponibilidad, etc. Perry y Wolf (1992) describen una arquitectura software como:

Arquitectura Software = {Elementos, Formas, Fundamento/Restricciones}

Es muy complejo capturar la arquitectura software en un sólo modelo (o diagrama). Para manejar esta complejidad se representan diferentes aspectos y características de la arquitectura en múltiples vistas. Una vista es “una presentación de un modelo, la cual es una descripción completa de un sistema desde una particular perspectiva” (Kruchten, 1995). El modelo más aceptado a la hora de establecer las vistas necesarias para describir una arquitectura software es el modelo 4+1 (Kruchten, 1995).

Este modelo define 4 vistas principales:

* **Vista Lógica** (Logical View), modelo de objetos, clases, entidad – relación, etc.
* **Vista de Proceso** (Process View), modelo de concurrencia y sincronización.
* **Vista de Despliegue** (Development View), organización estática del software en su entorno de desarrollo (librerías, componentes, .ear, .jar, etc.).
* **Vista Física** (Physical View), modelo de correspondencia software - hardware (aspectos de distribución en máquinas, por ejemplo)



* + 1. **Vista Lógica**

**Modelo de clases**

Se presenta, a continuación el modelo o diagrama de clases del proyecto. El diagrama de clases es un tipo de diagrama estático que define la estructura del sistema, mostrando las clases que lo componen y la forma en que se relacionan entre ellas. Las clases presentadas corresponden a la evolución desde el sprint 1 al sprint 4.

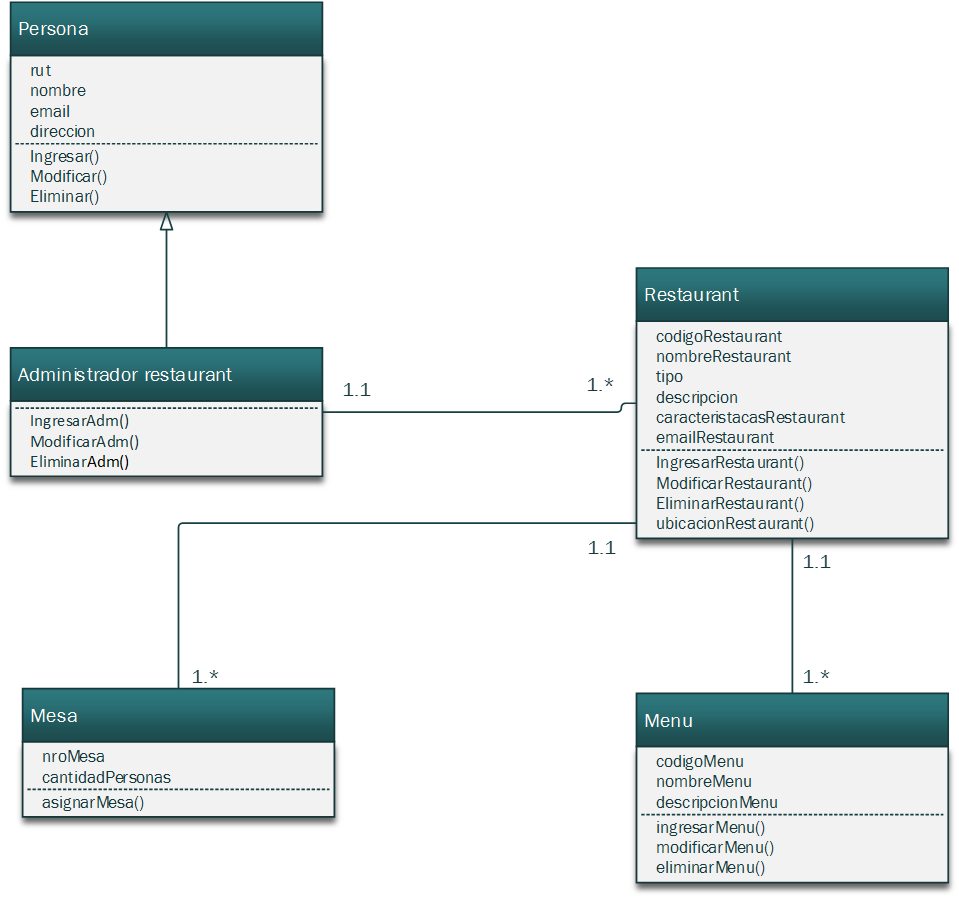


Figura 3.6 Diagrama clases.

* + 1. **Vista Despliegue**

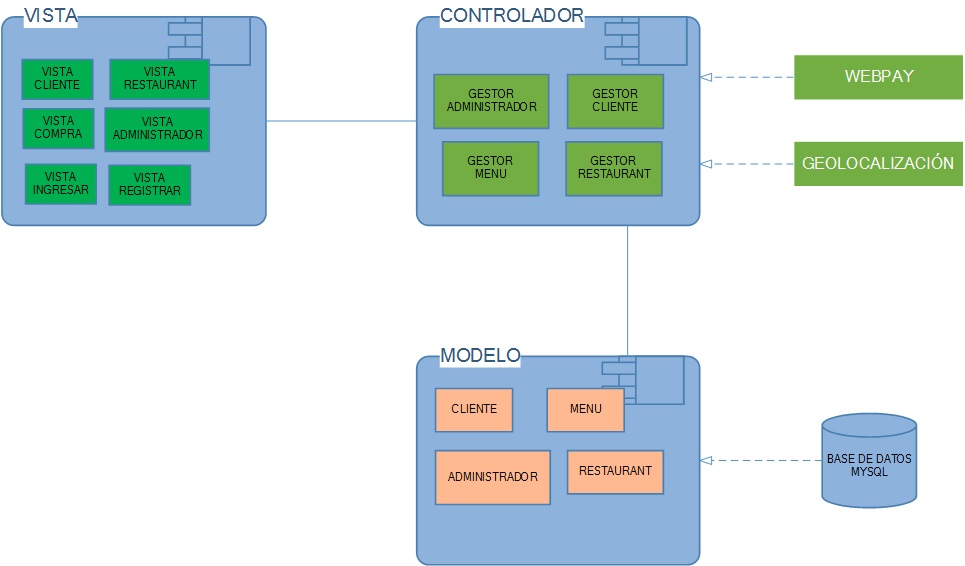


Figura 3.7 Diagrama despliegue

* + 1. **Vista física**

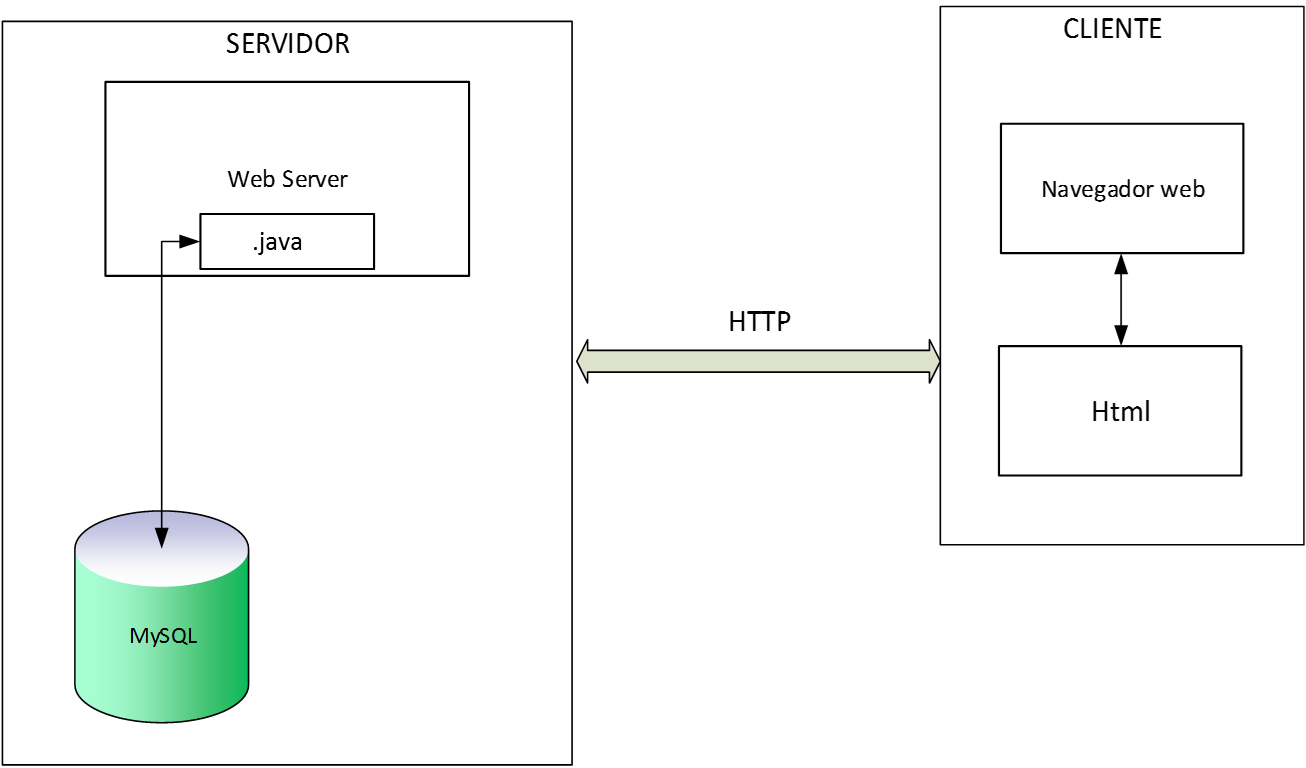


Figura 3.8 Diagrama físico

* + 1. **Vista de Procesos**

**Diagrama de Actividad**



Figura 3.9 Diagrama actividad “Gestionar administrador restaurant”



Figura 3.91 Diagrama actividad “Gestionar restaurant”

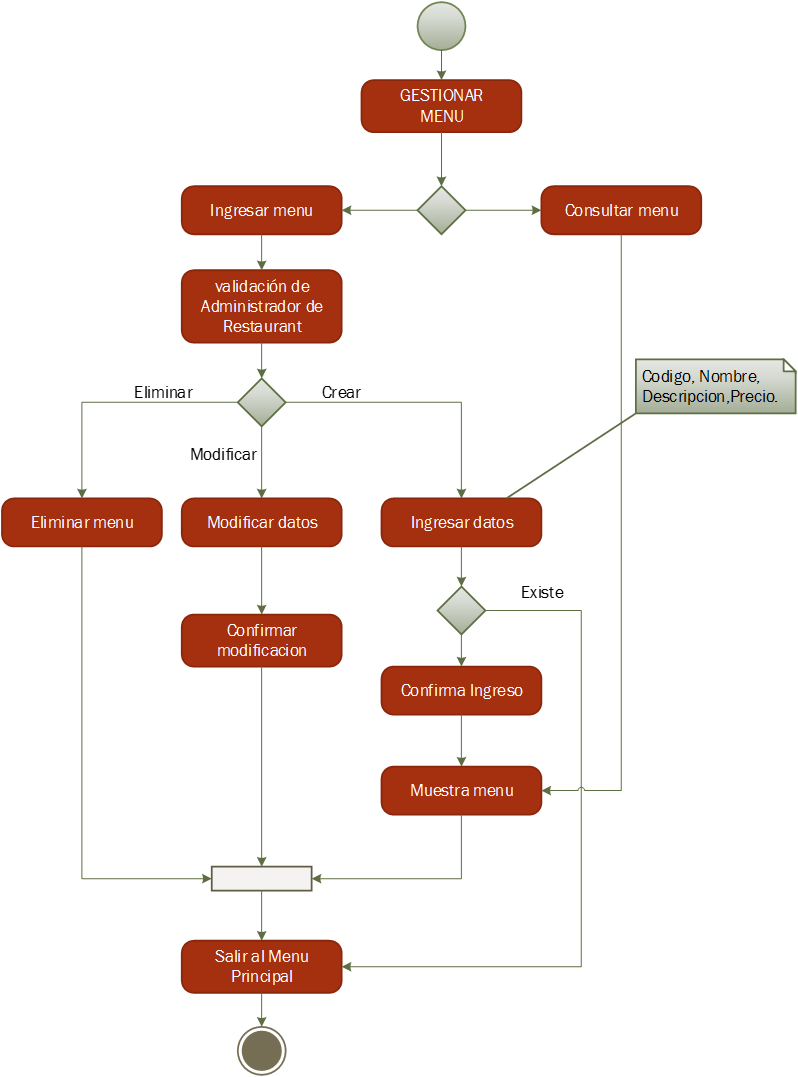


Figura 3.92 Diagrama actividad “Gestionar menú”

* + 1. **Diseño de la Base de datos**

Presentamos, a continuación, el diseño de la persistencia en base de datos con el correspondiente diagrama entidad relación (diagrama E/R), para nuestro modelo de entidades definido previamente.

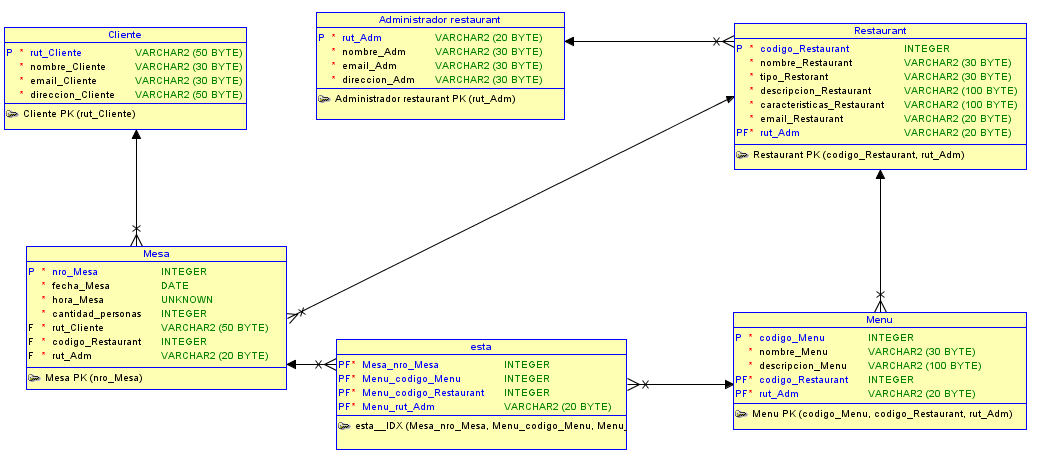


Figura 3.93 Diagrama clases.

* 1. **Requerimientos**
     1. **Requerimientos funcionales**

A continuación se muestran los requerimientos funcionales del sistema, obtenido desde las historias de usuario. Además se puede observar su prioridad, cuales ya están terminados y al sprint que pertenecen.

Tabla 3.1 Product backlog del proyecto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Historia Usuario** | **Descripción** | **Prioridad** | **Estado** | **Sprint** |
| **11** | Como cliente necesito acceder al sistema. | **1** | Terminada | 1 |
| **1** | Como Administrador del restaurant necesito registrarme en la aplicación. | **2** | Terminada | 1 |
| **2** | Como Administrador del restaurant necesito ingresar mi restaurant a la aplicación. | **3** | Terminada | 1 |
| **18** | Como administrador del restaurant necesito modificar mis datos | **4** | Terminada | 1 |
| **19** | Como administrador del restaurant necesito eliminar mi registro | **4** | Terminada | 1 |
| **3** | Como administrador del restaurant necesito modificar los datos de mi restaurant. | **5** | Terminada | 2 |
| **4** | Como administrador del restaurant necesito eliminar mi restaurant. | **5** | Terminada | 2 |
| **17** | Como cliente necesito acceder al sistema para buscar un restaurant en mi smartphone. | **6** | Terminada | 3 |
| **13** | Como cliente necesito reservar una mesa con fecha y hora determinada en el restaurant elegido en línea. | **7** | En proceso | 4 |
| **14** | Como cliente necesito seleccionar el menú del restaurant reservado, para cancelarlo. | **7** | En proceso | 4 |
| **15** | Como cliente necesito pagar menú en línea. | **7** | En proceso | 4 |
| **12** | Como cliente necesito registrarme en el sistema para poder acceder a las reservas. | **8** | Pendiente |  |
| **16** | Como cliente necesito comentar sobre un restaurant o del sistema. | **9** | Pendiente |  |
| **8** | Como Administrador del restaurant necesito ingresar ofertas o datos de promociones de mi restaurant. | **10** | Pendiente |  |
| **9** | Como Administrador del restaurant necesito modificar ofertas o datos de promociones de mi restaurant. | **10** | Pendiente |  |
| **10** | Como Administrador del restaurant necesito eliminar ofertas o datos de promociones de mi restaurant. | **10** | Pendiente |  |
| **20** | Como cliente necesito modificar mis datos. | **11** | Pendiente |  |
| **5** | Como Administrador del restaurant necesito ingresar los menús de mi restaurant. | **12** | Pendiente |  |
| **6** | Como administrador del restaurant necesito modificar los datos de menús en mi restaurant. | **12** | Pendiente |  |
| **7** | Como administrador del restaurant necesito eliminar los menús de mi restaurant. | **12** | Pendiente |  |

* + 1. **Requerimientos no funcionales**

|  |  |  |
| --- | --- | --- |
| **REQUISITOS NO FUNCIONALES** | | |
| **RNF1** | USABILIDAD | Sera realizada la interfaz de usuario General (Dispositivo móvil): Sera desplegada la información de los restaurantes por georeferencia a través de un dispositivo móvil con Sistema Operativo Android a los usuarios. |
| **RNF2** | USABILIDAD | Debe ser fácil de usar. Con ayudas e interfaces intuitivas |
| **RNF3** | SEGURIDAD | El ingreso al sistema estará restringido bajo contraseñas cifradas y usuarios personalizados |
| **RNF4** | PORTABILIDAD | El sistema debe brindar comodidad al usuario y a otras áreas que trabajan o necesitan de esta aplicación. |
| **RNF5** | RENDIMIENTO | El sistema debe soportar el manejo de gran cantidad de información durante su proceso |
| **RNF6** | DESEMPEÑO | El sistema no presentara problemas para su manejo o implementación |
| **RNF7** | DISPONIBILIDAD | La aplicación móvil estará soportada para versiones de Android 2.3.3 - 4.4.2 |
| **RNF8** | INTERFAZ | Se programara la interfaz del dispositivo móvil para el administrador del restaurant, donde las opciones de visualización sean visibles a los clientes. |

Tabla 3.2 Requerimientos no funcionales del proyecto.

* + 1. **Trazabilidad de requerimientos**



Tabla 3.3 Trazabilidad de objetivos específicos vs requerimientos.

* 1. **Explicación Metodología Seleccionada**
     1. **SCRUM**: Para la gestión del proyecto

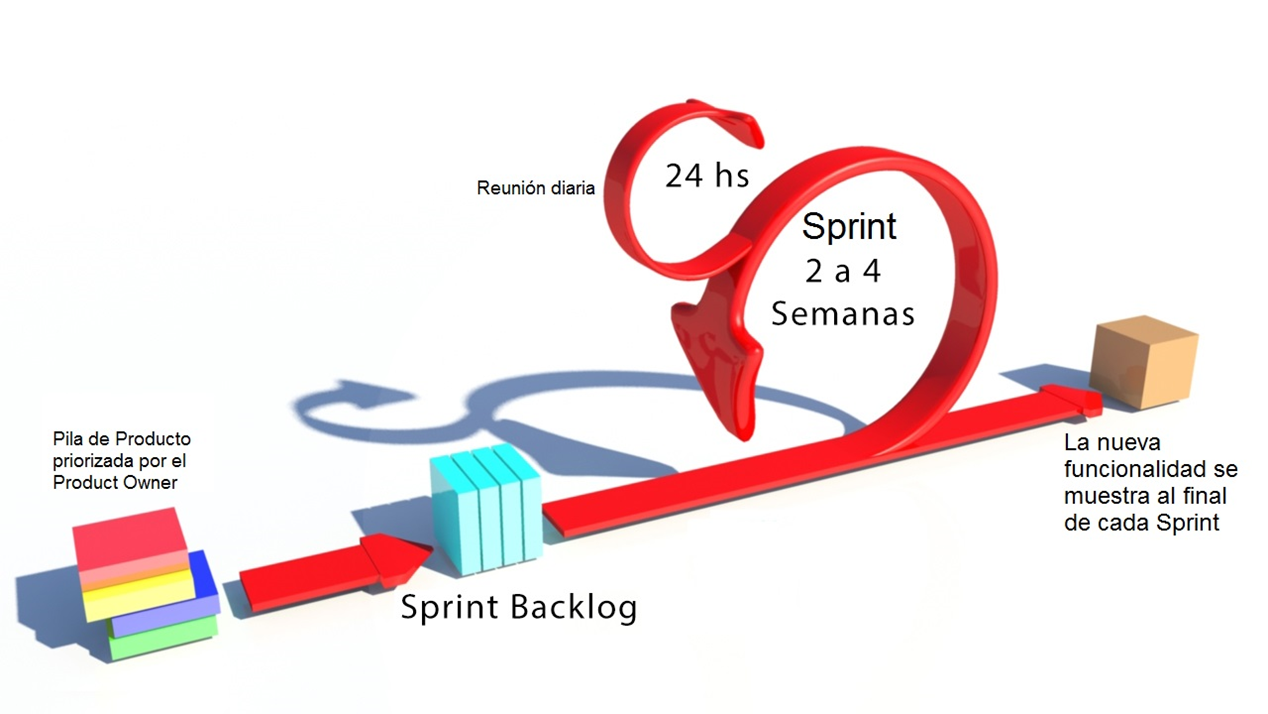


Figura 3.1 Diagrama de desarrollo de Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos, como es nuestro equipo, ya que trabajamos de manera integrada y comprometida con cualquier tipo de proyecto.

En esta metodología se realizan entregas parciales del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales.

Debido a estas características, esta metodología se ajusta adecuadamente a nuestro proyecto, ya que debemos ir mostrando productos funcionales en corto tiempo para ir analizando si vamos cumpliendo con las expectativas del cliente y/o usuario, hasta llegar al producto final.

* + 1. **Roles de Scrum**

En Scrum hay 3 roles principales y uno auxiliar los cuales son:

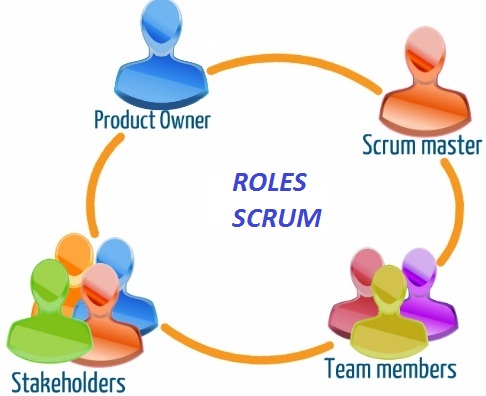


Figura 3.2 Roles en metodología Scrum.

Roles principales

**Product owner:** Representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

**ScrumMaster:** Su trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint, se asegura de que el proceso Scrum se utiliza como es debido y es el que hace que las reglas se cumplan.

**Team members (*Equipo de desarrollo*):** Tienen la responsabilidad de entregar el producto. Ellos cuentan las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc.).

Rol auxiliar

**Stakeholders:** Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones del sprint.

Roles en el proyecto SRCR:

* **Product owner:** Oscar Pinto.
* **ScrumMaster:** Daniel Fernández.
* **Equipo de desarrollo:** Daniel Fernández / Octavio Valencia.
* **Cliente:** Clientes de restaurant.
  + 1. **Ventajas y desventajas de Scrum**

|  |  |
| --- | --- |
| ***Ventajas***   * *Se obtiene software lo más rápido posible y este cumple con los requerimientos más importantes.* * *Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.* * *Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.* * *Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.* * *Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.* * *Permite producir software de una forma consistente, sostenida y competitiva.* * *Las reuniones se dedican a inconvenientes recientes, evitando el estancamiento* | ***Desventajas***   * *Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.* * *Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas.* |

Tabla 3.1 Ventajas y desventajas de la metodología Scrum.

* + 1. **XP (Xtreme Programming):** Para el desarrollo del proyecto.

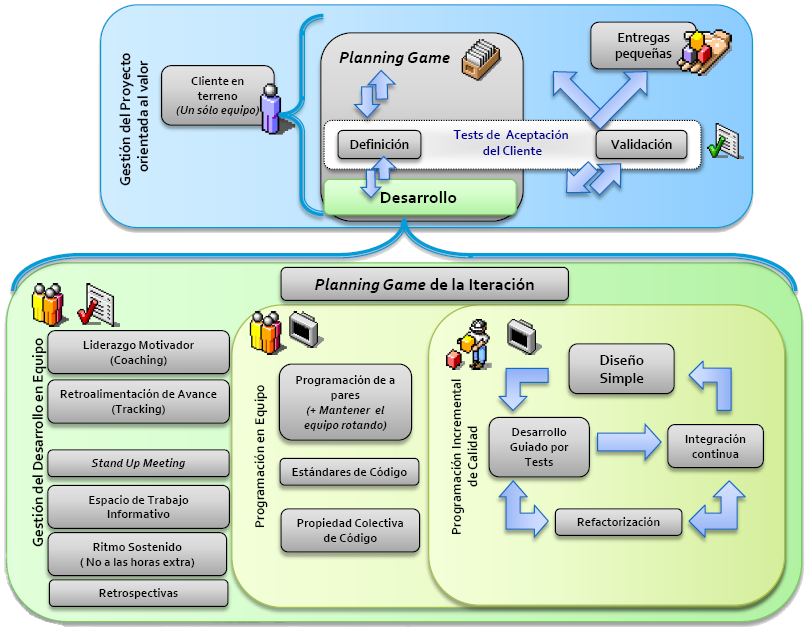


Figura 3.3 Diagrama de desarrollo de metodología XP.

La metodología XP [8] se ajusta a la de gestión (SCRUM), ya que tiene las siguientes características:

* Porque la programación está basada en los deseos del cliente.
* El equipo lo conforman el jefe de proyecto (Scrum manager), los desarrolladores y el cliente(o Product Owner).
* Programación organizada en comunicación con todo el equipo.
* La experiencia indica que tiene menor tasa de errores.
* Se pueden realizar pruebas continuas.
* Programación de a pares con estándares de código.
  + 1. **Roles en metodología XP:**



Figura 3.4 Roles en metodología XP.

* + 1. **Ventajas y desventajas de XP**

|  |  |
| --- | --- |
| ***Ventajas***   * *Programación organizada.* * *Menor taza de errores.* * *Satisfacción del programador.* * *Solución de errores de programas.* * *Versiones nuevas.* * *Implementa una forma de trabajo donde  se adapte fácilmente a las circunstancias* | ***Desventajas***   * *Es recomendable emplearlo solo en proyectos a corto plazo.* * *Altas comisiones en caso de fallar.* * *Imposible prever todo antes de programar.* |

* + 1. **Combinación de roles entre Scrum y XP**



* 1. **Plan de proyecto (incluir planificación otras métricas)**
     1. **Planificación de los Sprint**

Los sprint establecidos para el proyecto tendrán una duración de 3 semanas con un máximo de 4. Al iniciar nuestro proyecto se estimaron 5 sprint, pero con el avance del proyecto se han aumentado a 7, debido a riesgos ocurridos y por cambios en las prioridades por parte del Product Owner.

Sprint 1:

|  |  |  |
| --- | --- | --- |
| **HU** | **Descripción** | **Estado** |
| **11** | Como cliente necesito acceder al sistema. | Terminada |
| **1** | Como Administrador del restaurant necesito registrarme en la aplicación. | Terminada |
| **2** | Como Administrador del restaurant necesito ingresar mi restaurant a la aplicación. | Terminada |
| **18** | Como administrador del restaurant necesito modificar mis datos | Terminada |
| **19** | Como administrador del restaurant necesito eliminar mi registro | Terminada |

Sprint 2:

|  |  |  |
| --- | --- | --- |
| **HU** | **Descripción** | **Estado** |
| **3** | Como administrador del restaurant necesito modificar los datos de mi restaurant. | Terminada |
| **4** | Como administrador del restaurant necesito eliminar mi restaurant. | Terminada |

Sprint 3:

|  |  |  |
| --- | --- | --- |
| **HU** | **Descripción** | **Estado** |
| **17** | Como cliente necesito acceder al sistema para buscar un restaurant en mi smartphone | Terminada |

Sprint 4:

|  |  |  |
| --- | --- | --- |
| **HU** | **Descripción** | **Estado** |
| **13** | Como cliente necesito reservar una mesa con fecha y hora determinada en el restaurant elegido en línea. | En proceso |
| **14** | Como cliente necesito seleccionar el menú del restaurant reservado, para cancelarlo. | En proceso |
| **15** | Como cliente necesito pagar menú en línea. | En proceso |

Sprint 5:

|  |  |  |
| --- | --- | --- |
| **HU** | **Descripción** | **Estado** |
| **12** | Como cliente necesito registrarme en el sistema para poder acceder a las reservas | Pendiente |
| **20** | Como cliente necesito modificar mis datos | Pendiente |
| **5** | Como Administrador del restaurant necesito ingresar los menús de mi restaurant. | Pendiente |

Sprint 6:

|  |  |  |
| --- | --- | --- |
| **HU** | **Descripción** | **Estado** |
| **6** | Como administrador del restaurant necesito modificar los datos de menús en mi restaurant. | Pendiente |
| **7** | Como administrador del restaurant necesito eliminar los menús de mi restaurant. | Pendiente |

Sprint 7:

|  |  |  |
| --- | --- | --- |
| **HU** | **Descripción** | **Estado** |
| **16** | Como cliente necesito comentar sobre un restaurant o del sistema. | Pendiente |
| **8** | Como Administrador del restaurant necesito ingresar ofertas o datos de promociones de mi restaurant. | Pendiente |
| **9** | Como Administrador del restaurant necesito modificar ofertas o datos de promociones de mi restaurant. | Pendiente |
| **10** | Como Administrador del restaurant necesito eliminar ofertas o datos de promociones de mi restaurant. | Pendiente |

Tabla 3.3 Listado de historias de usuario por sprint.



Tabla 3.4 Planificación de los sprint del proyecto completo.

* + 1. **Planificación de las tareas**

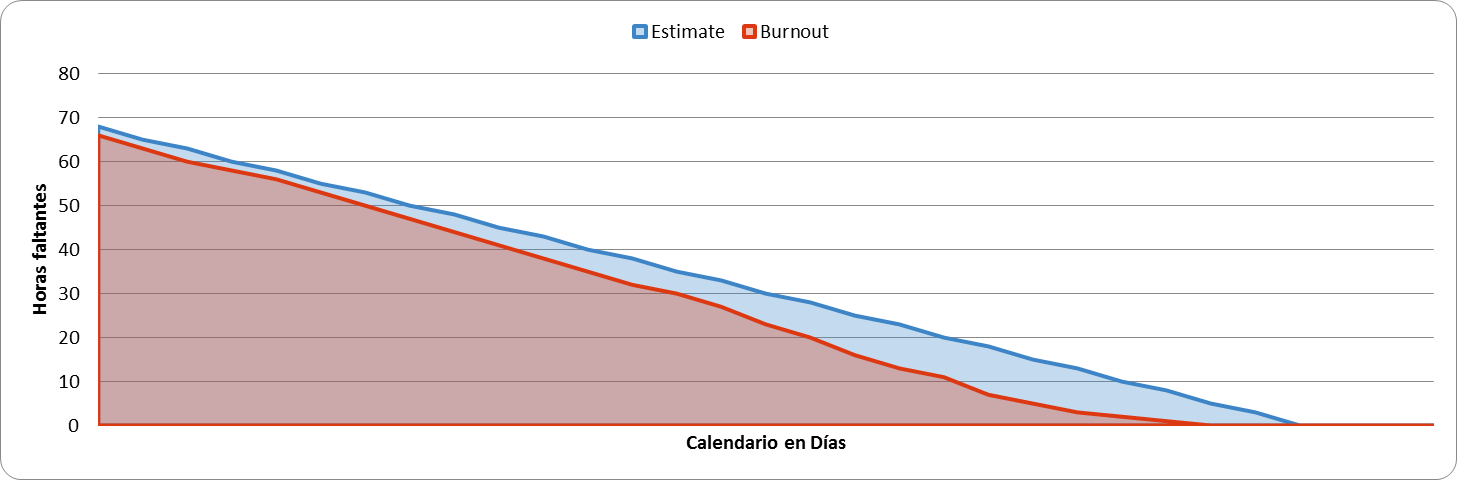
Para el desarrollo de las tareas del proyecto, realizamos el siguiente seguimiento de acuerdo a cada Sprint:

**SPRINT 1**



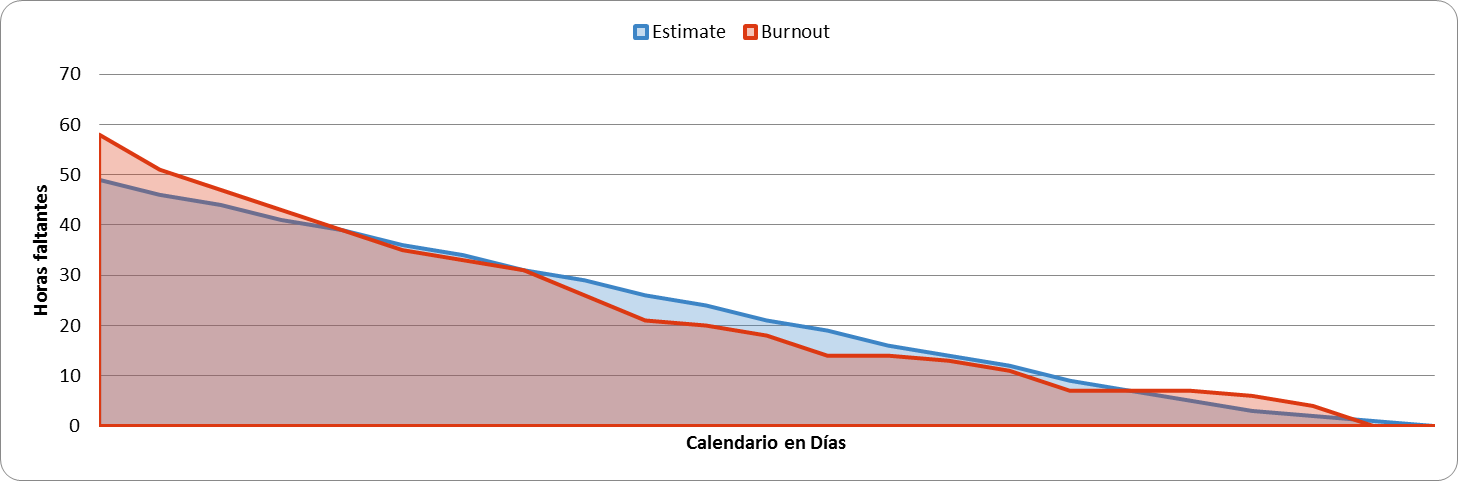
**SPRINT 2**



****

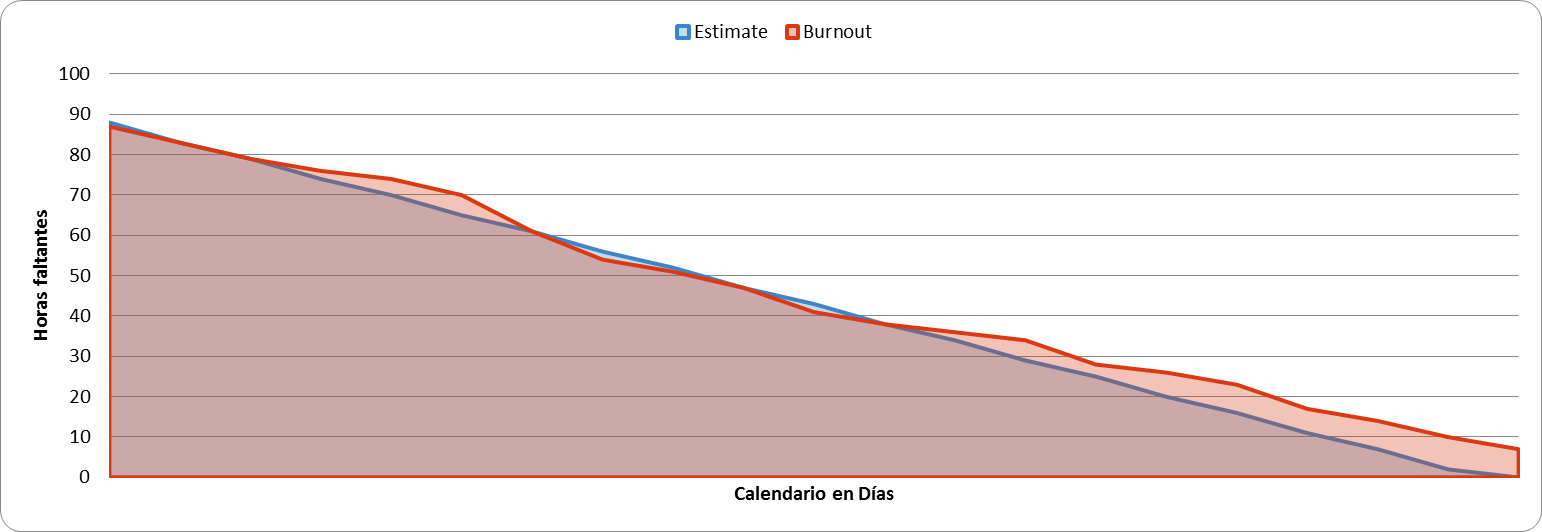
**SPRINT 3**



****

**SPRINT 4**



****

* 2. **Plan de Pruebas**
     1. **Propósito del Plan**

El propósito de este plan es planificar, estructurar y documentar la planificación de las pruebas de aceptación del sistema a realizar, así como la estrategia a utilizar para su ejecución.

* + 1. **Alcance**

Luego de finalizar las pruebas de sistema, el programa se encuentra completamente ensamblado, y se han encontrado y corregido los errores entre los módulos, métodos, clases y objetos. En este punto se comienza con la etapa de las pruebas de validación de requerimientos más conocida como pruebas de aceptación. Éstas se enfocan en las acciones que realiza el usuario además de las salidas del sistema que puedan ser reconocidas por él; dichas acciones y salidas engloban las expectativas del usuario, y están definidas en las especificaciones de los requerimientos del software.

Las pruebas de aceptación, se realizan a los requerimientos funcionales, y a los no-funcionales como facilidad de uso, recuperación, eficiencia, entre otros; y se pretende lograr: corrección, vale decir, carencia de ambigüedad; completitud, es decir, especificación completa y clara del problema; y por último pero no menos importante, consistencia, quiere decir, que no haya requisitos contradictorios.

El plan que a continuación se detalla pretende dar una visión general sobre las actividades a realizar; sobre las pruebas consideradas; además de una explicación global que se consideró para la realización de los documentos a entregar, ya que darán una mayor información relacionada a la evaluación y reportes de este tipo de pruebas.

* + 1. **Requerimientos de Pruebas**
       1. **Introducción**

Este capítulo documenta los requerimientos de prueba durante la Fase de Pruebas del Sistema e Integración para el Sistema para reservar y comprar menú de restaurantes en línea (FlashMenu).

* + - 1. **Filosofía de la prueba**

1. *Generalidades*

El objetivo principal de las pruebas unitarias del sistema será el de establecer un nivel de confianza que nos permitirá asegurar la aceptación del sistema por los usuarios (Product Owner) en las posteriores pruebas de aceptación.

Se probará que la aplicación cumpla con los requerimientos de alto nivel que fueron especificados previamente, verificando que se cumple satisfactoriamente con las funcionalidades y características necesarias para que los usuarios satisfagan estos.

1. *Áreas funcionales*

Esta sección describe las áreas funcionales generales que deberán ser probadas como parte de la fase de pruebas del sistema.

* Funcionalidad especificada en las historias de usuario.
* Manejo de los datos y transacciones involucradas en las funcionalidades del punto anterior.
* Rendimiento al ejecutar las funcionalidades del primer punto.

1. *Categorías de resultados de prueba*

Esta sección describe las categorías que pueden ser asignadas los resultados de prueba en un Caso de Prueba.

**Éxito:** El resultado de la prueba es conforme al resultado esperado.

**Aceptable:** El resultado de la prueba indica que el sistema difiere de la especificación aceptada pero es aceptable, no son necesarios cambios en la aplicación, pero requiriendo un cambio en la Especificación Funcional.

**Tolerable:** El resultado de la prueba es incorrecto, la aplicación en prueba trabaja y podría ser aceptada, pero la falla deberá ser rectificada en el periodo de tiempo acordado.

**Intolerable:** El resultado de la prueba es incorrecto, y la falla debe ser corregida antes de concluir la fase de prueba.

**Error:** El resultado de la prueba observado es correcto, pero el resultado esperado de acuerdo a los scripts de prueba son incorrectos.

* + - 1. **Entorno de la prueba**

1. *Generalidades*

En esta sección se da una breve descripción del entorno de prueba:

* Las pruebas se realizarán principalmente en el laboratorio de Desarrollo de Programas 2 (V201).
* El número de computadoras con que se contará será de acuerdo al tipo de prueba, entre una y cuatro computadoras.
* En los laboratorios se cuenta con acceso al Servidor y definición de Datos del Sistema de Gestión de Club de Tenis

1. *Hardware*

|  |  |
| --- | --- |
| **Recursos del sistema** | |
| **Recurso** | **Nombre/Tipo/Número de serie** |
| Servidor | Proporcionado por el laboratorio de informática |
| PC’s Desarrollo de pruebas – 4 | Proporcionadas por el laboratorio de informática |

1. *Software*

En las PCs de prueba deberán estar instaladas las siguientes aplicaciones de software:

* Sistema Operativo Windows 7 o versiones posteriores
* JAVA2 v1.5.1 o mayor
* Eclipse SDK v3.2 o mayor
* Base de Datos MySQL

1. *Datos de prueba*

Se desarrollarán y especificarán conjuntos de datos de prueba, tomando las muestras necesarias para la ejecución de las pruebas, de manera que se verifique que cumple con diversos tipos de datos.

* + - 1. **Identificación de la prueba**

1. Scripts de prueba

Cada caso de prueba individual deberá tener un script que describa los pasos y los resultados esperados de cada prueba individual. En particular un script contiene la siguiente información:

* Identificador de la prueba.
* Descripción del objetivo de la prueba.
* Descripción del estado de la aplicación antes de la prueba o pre-condiciones de la misma.
* Pasos precisos y no ambiguos para ejecutar la prueba.
* Descripción de los resultados esperados.

1. Reporte de resultados

Los resultados de la prueba son registrados en un formulario de Registro de Resultados de Prueba, el cual contiene la siguiente información:

* Nombre y versión de la aplicación a prueba.
* Fase de Prueba.
* Fecha de Prueba.
* Identificador único de prueba.
* Hora de ejecución de cada Caso de Prueba.
* Resultado observado durante la prueba.
* Categoría de resultado de prueba. (ver punto 3.6.3.2.c)
* Descripción del error.
* Firma del ejecutor y del observador de la prueba.

1. Criterios de aceptación

Esta sección documenta la frecuencia de las categorías de los resultados de prueba (ver sección 3.6.3.2.c) que son consideradas para aceptar la aplicación y pasar con éxito la fase de prueba. Identificamos los siguientes criterios los cuales deben ser evaluados progresivamente.

* Requerimientos de Prueba: ¿Todos los requerimientos del sistema han sido probados?
* Pruebas Cubiertas: ¿Todas las partes del software han sido probadas, incluyendo manejo de errores?
* Medida de Casos de Prueba: ¿Cuántos Casos de Prueba han sido planeados, diseñados, implementados, ejecutados y pasaron con éxito o falla?
* Defectos detectados en Casos de Prueba: Es importante tener un ratio de los defectos encontrados en los casos de prueba, y de los defectos corregidos y mantenidos.

1. Errores de prueba

Esta sección específica los procesos para alcanzar la corrección de los errores observados y registrados durante la prueba.

Para cada error observado que requiera corrección de la aplicación o de la especificación de funcionalidades, el líder del equipo de prueba y el líder de desarrollo y sus respectivos equipos, deben de estar de acuerdo en lo siguiente:

* El ámbito de trabajo adicional y escalas de tiempo para la corrección.
* El Caso de Prueba requerido para ser re-ejecutado después de la corrección.
* Dada una falla, el principal responsable de realizar la corrección es el que se encargó de desarrollar dicho componente.
* Establecer prioridades de acuerdo a una serie de fallas.

1. Documentación de la prueba

Esta sección describe los documentos que deben ser generados durante la actividad de prueba. Estos documentos son los siguientes:

* Scripts de pruebas y Casos de Prueba.
* Resultados de Pruebas siguiendo el formato especificado.
* Reporte consolidado de pruebas por módulo.
* Certificado de prueba para formalizar el hecho de que la aplicación en prueba ha pasado la prueba con éxito.

La lista que se muestra a continuación identifica los requerimientos especificados en las Historias de usuario que se probarán.

1. Pruebas Funcionales

* Revisar la implementación de que:
* cliente acceda al sistema.
* Administrador del restaurant se registre en la aplicación.
* Administrador del restaurant ingrese su restaurant a la aplicación.
* Administrador del restaurant modifique sus datos.
* Administrador del restaurant elimine su registro.
* Administrador del restaurant modifique los datos de su restaurant.
* Administrador del restaurant elimine su restaurant.
* Cliente acceda al sistema para buscar un restaurant en su smartphone.
* Cliente reserve una mesa con fecha y hora determinada en el restaurant elegido en línea.
* Cliente seleccione el menú del restaurant reservado, para cancelarlo.

1. Pruebas de integración

Se realizarán de manera implícita al realizar las pruebas del caso de uso.

1. Pruebas del caso de uso

Se verificará la correcta implementación de los flujos básicos y alternativos de todos los casos de uso a implementar en la iteración.

1. Casos de Prueba

|  |  |
| --- | --- |
| **Pruebas de Aceptación PA01** | |
| **Proyecto:** | FlashMenu |
| **Nº HU que prueba:** | **11** |
| **Titulo Historia de Usuario que prueba:** | Como cliente necesito acceder al sistema para buscar un restaurant en línea |
| **Criterio de aceptación:** | **Poder buscar un restaurant** |
| **Contexto** | Se buscan restaurantes de acuerdo a la posición geográfica del cliente |
| **Evento** | Cuando presiona Buscar restaurantes cercanos |
| **Resultado/Comportamiento esperado** | El sistema muestra los restaurantes cercanos a la posición geográfica del cliente. |

|  |  |
| --- | --- |
| **Pruebas de Aceptación PA02** | |
| **Proyecto:** | FlashMenu |
| **Nº HU que prueba:** | **11** |
| **Titulo Historia de Usuario que prueba:** | Como cliente necesito acceder al sistema para buscar un restaurant en línea |
| **Criterio de aceptación:** | **Al buscar restaurant no se encuentra ninguno** |
| **Contexto** | Al buscar restaurantes en línea no se encuentra ninguno disponible |
| **Evento** | Cuando presiona Buscar restaurantes cercanos |
| **Resultado/Comportamiento esperado** | El sistema muestra un mensaje que no hay ninguno cerca de su ubicación. |

|  |  |
| --- | --- |
| **Pruebas de Aceptación PA03** | |
| **Proyecto:** | FlashMenu |
| **Nº HU que prueba:** | **11** |
| **Titulo Historia de Usuario que prueba:** | Como cliente necesito acceder al sistema para buscar un restaurant en línea |
| **Criterio de aceptación:** | **Al buscar restaurant no muestra la posición geográfica propia.** |
| **Contexto** | Al tratar de buscar restaurantes en línea no logra mostrar la propia ubicación. |
| **Evento** | Cuando presiona Buscar restaurantes cercanos |
| **Resultado/Comportamiento esperado** | El sistema muestra un mensaje que verifique la activación de su sistema de posicionamiento global. |

|  |  |
| --- | --- |
| **Pruebas de Aceptación PA04** | |
| **Proyecto:** | FlashMenu |
| **Nº HU que prueba:** | **1** |
| **Titulo Historia de Usuario que prueba:** | Como Administrador del restaurant necesito registrarme en la aplicación. |
| **Criterio de aceptación:** | **Registrar un administrador de restaurant en el sistema de manera correcta.** |
| **Contexto** | Se registra un administrador del restaurant en el sistema. |
| **Evento** | Cuando se presiona Registrar Administrador de restaurant. |
| **Resultado/Comportamiento esperado** | Se solicitan los datos y muestra mensaje de administrador registrado, entregando un usuario y password. |

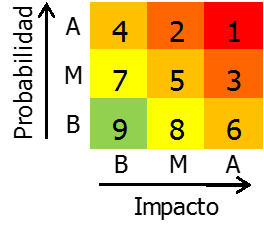
|  |  |
| --- | --- |
| **Pruebas de Aceptación PA05** | |
| **Proyecto:** | FlashMenu |
| **Nº HU que prueba:** | **1** |
| **Titulo Historia de Usuario que prueba:** | Como Administrador del restaurant necesito registrarme en la aplicación. |
| **Criterio de aceptación:** | **Registrar administrador de restaurant con datos erróneos o faltantes.** |
| **Contexto** | Cuando se está registrando un nuevo administrador de restaurant y no se llenan todos los campos necesarios. |
| **Evento** | Cuando se presiona Registrar Administrador de restaurant. |
| **Resultado/Comportamiento esperado** | Se revisan los campos, se muestra un mensaje de error y se indican los campos con error. |

|  |  |
| --- | --- |
| **Pruebas de Aceptación PA06** | |
| **Proyecto:** | FlashMenu |
| **Nº HU que prueba:** | **1** |
| **Titulo Historia de Usuario que prueba:** | Como Administrador del restaurant necesito registrarme en la aplicación. |
| **Criterio de aceptación:** | **Registrar administrador de restaurant existente.** |
| **Contexto** | Cuando se está registrando un nuevo administrador de restaurant y los datos ingresados ya existen. |
| **Evento** | Cuando se presiona Registrar Administrador de restaurant. |
| **Resultado/Comportamiento esperado** | Se revisan los campos, se muestra un mensaje de error y se indica que los datos del administrador ingresado ya existen. |

* 1. **Plan de Gestión de Configuración**
  2. **Plan de Gestión de Cambios**
  4. **Plan de Gestión de Riesgos**
     1. **Matriz de riesgos del proyecto**



Para realizar la matriz de riesgos se utilizó la siguiente tabla de probabilidad versus impacto:

****

* + 1. **Evaluación del nivel de riesgo general del proyecto**

Las posibles causas de riesgo están divididas en cinco áreas: *Adquisición Suministro, Desarrollo, Soporte y Organizacionales*. Las cuales se van a evaluar a continuación:











Cuadro resultante de los valores anteriores:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Valores de riesgo** | | | | | |  | **NIVEL DE RIESGO GENERAL** |
| ÁREA | Adquisición | Suministro | Desarrollo | Soporte | Organización | TOTAL |  |
| Total | 14 | 23 | 52 | 10 | 56 | 155 |  | 155 |
| Máximo | 72 | 49 | 90 | 45 | 88 | 344 |  | 344 |
| Porcentual | 19,44 | 46,94 | 57,78 | 22,22 | 63,64 | 210,02 |  | 45 |
|  |  |  |  |  |  |  |  | **MEDIO** |
| Nivel de riesgo | BAJO | MEDIO | MEDIO | BAJO | MEDIO | ALTO |  |

Información disponible en www.navegapolis.net/files/blog/evaluacion\_**riesgos**\_agilidad.xls

* 1. **Ambiente de desarrollo, pruebas y producción**

**CAPITULO 4**

**RESULTADO Y DISCUSIÓN**

Análisis

Diseño detallado

Pruebas Desarrollador

## Pruebas de Sistema (funcionales y de rendimiento)

Pruebas de Aceptación (al menos usabilidad)

Automatización (si corresponde) y regresión

Evidencias de Liberación del producto

Evidencias de Control de versiones

Evidencias de Control de cambios

**CAPITULO 5**

**CONCLUSIONES**

PostMortem

Problemas Abiertos

Trabajo Futuro

1. **Bibliografía**

Libros consultados:

* [1] Javier Garzas Parra. *Como sobrevivir... A la planificación de un proyecto ágil.*
* [2] Schwaber, K. (1995). *SCRUM Development Process*. Burlington: OOPSLA 95.
* [3] Juan Palacio, Claudia Ruata (2011). *Scrum Manager Gestión de Proyectos*. Rev.1.4.0 Enero-2011
* [4] Juan Palacio (2008). *Flexibilidad con Scrum*. Edición Octubre-2008 / Noviembre-2007
* [5] Henrik Kniberg, Prólogos de Jeff Sutherland y Mike Cohn (2007). *Scrum y XP desde las trincheras*
* [6] Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde.(2009). *INFORMACIÓN BÁSICA DE SCRUM (THE SCRUM PRIMER)*. Versión 1.1
* [7] Eric Ries, *El Método Lean Startup* (2012)
* [8] Beck, K.. “Extreme Programming Explained. Embrace Change”, Pearson Education, 1999. Traducido al español como: “Una explicación de la programación extrema. Aceptar el cambio”, Addison Wesley, 2000
* [9] Kruchten P. Architectural Blueprints—The “4+1” View Model of Software Architecture. IEEE Software, November 1995, 12 (6)