**A N E P Instituto Tecnológico Superior**

CONSEJO DE EDUCACION  **F. Arias – L. Balparda**

TECNICO PROFESIONALGral. Flores 3591 esq.

**(Universidad del Trabajo** Bvar. José Batlle y Ordoñez

**Del Uruguay)** Montevideo, Uruguay

Santiago Illarze

Martina Karszensztejn

Patricio Suárez

Emanuel González

3° IA - ITS

Proyecto

Índice

1. Introducción

1.1. Objetivo

1.2. Alcance

1.3. Limitaciones

2. Integrantes del grupo

3. Reglas de trabajo en equipo

4. Ciclo de Vida

4.1. Modelos

4.2. Justificación del ciclo de vida elegido

5. Diagrama de Gantt

6. Actas de reunión

7. Normativa ISO/IEC 9126

7.1. Disponibilidad

7.2. Eficiencia

7.3. Modificabilidad

7.4. Usabilidad

7.5. Portabilidad

7.6. Seguridad

8. Bibliografía

9.Glosario

1. **Introducción**

Este documento es un informe para la planificación a futuro y detallado de las características desarrolladas de disponibilidad, eficiencia, modificabilidad, usabilidad, portabilidad y seguridad del software creado para el hotel canino Guau.

* 1. **Objetivo**

El propósito de este documento es clarificar las soluciones desarrolladas en la creación del software. Con esto detallamos la calidad de nuestro software.

* 1. **Alcance**

El alcance de este documento es el encargado de la sección administrativa gerencial de la empresa cliente.

* 1. **Limitaciones**

Cent OS 7

Windows 7,8,8.1,10.

MySql Server 5.6

.NET Framework 4.6

1. **Integrantes del grupo**

|  |  |
| --- | --- |
| Nombre | Patricio Suarez |
| Rol | Líder de equipo |
| Categoría profesional | Analista de datos |
| Responsabilidades | Análisis y mantenimiento de la base de datos |
| Información de contacto | patriciorsuarez@hotmail.com |
| Aprobación |  |

|  |  |
| --- | --- |
| Nombre | Emanuel González |
| Rol | Programador |
| Categoría profesional | Programador |
| Responsabilidades | Programación del servidor |
| Información de contacto | spyroddz@gmail.com |
| Aprobación |  |

|  |  |
| --- | --- |
| Nombre | Santiago Illarze |
| Rol | Diseñador grafico |
| Categoría profesional | Diseñador grafico |
| Responsabilidades | Diseño y justificación de la interfaz visual de usuario |
| Información de contacto | Illarzesantiago@gmail.com |
| Aprobación |  |

|  |  |
| --- | --- |
| Nombre | Martina Karszensztejn |
| Rol | Programador |
| Categoría profesional | Programador |
| Responsabilidades | Programación y tester del programa |
| Información de contacto | martinakarszensztejn@tutanota.com |
| Aprobación |  |

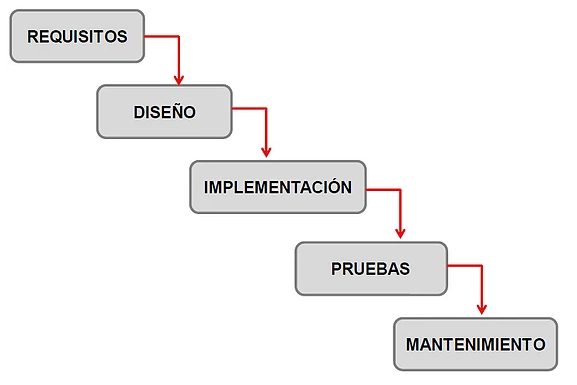
1. **Reglas de trabajo en equipo**

* Respeto en la diferencia de opiniones.
* Toma de decisiones democráticas.
* Apoyo a iniciativas personales.
* Compromiso con las responsabilidades tomadas.
* Distribución de tareas equitativamente.

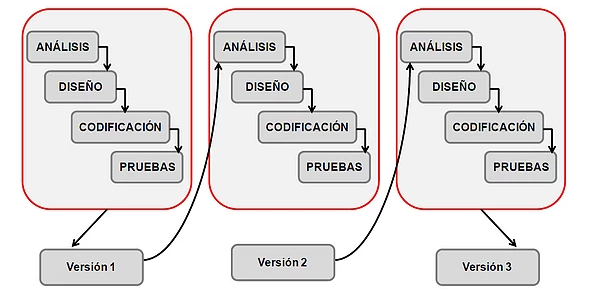
1. **Ciclo de vida**

El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere).

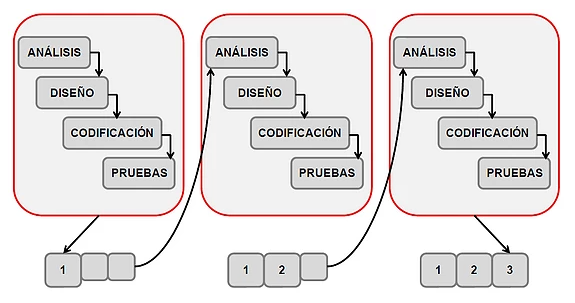
* 1. **Modelos**
     1. **Modelo Cascada**

Es un enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

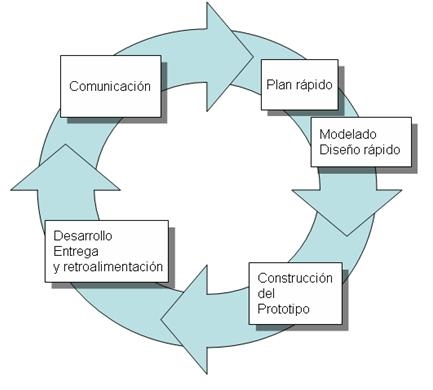
* + 1. **Modelo Iterativo**

Es un modelo derivado del ciclo de vida en cascada. Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de recogida de requisitos.

* + 1. **Modelo Incremental**

El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.

* + 1. **Modelo Evolutivo**

El desarrollo evolutivo consta del desarrollo de una versión inicial que luego de exponerse se va refinando de acuerdo de los comentarios o nuevos requerimientos por parte del cliente o del usuario final. Las fases de especificación, desarrollo y validación se entrelazan en vez de separarse.

* 1. **Elección del modelo**

Elegimos el modelo evolutivo ya que nos pareció el más adecuado según nuestra visión de cómo se debería quedar un programa y como esta forma ayuda a dar un producto final más refinado.

Descripción de como usamos el proceso evolutivo:

Leído los requerimientos por parte de la empresa Guau Hotel Canino y sus necesidades ante la creación de una aplicación para poder manejar los horarios de transporte, reservas, clientes junto a sus datos, mascotas y habitaciones se les realizó un M.E.R base y se creó la primera base de datos, junto al primer prototipo de programa, con un diseño básico.

Se procedió a releer los requisitos por parte de la empresa y se rehízo un M.E.R más acotado para que la funcionalidad del programa sea más fluida. A su vez también se retoco el diseño, dándole una mejor experiencia para el usuario y se agregaron ventanas para los otros empleados (oficinistas y recepcionistas). También se necesita la creación de un presupuesto por costo, instalación y mantenimiento del programa y cámaras que sean instaladas.

A futuro se tiene pensado, incluir más ventanas de ser necesario, mejor codificación para un mejor rendimiento, nuevo diseño de ventas, mejorar la base de datos para evitar futuros errores. Todo esto sería para el segundo prototipo, para entregar y testear. de ser necesario se crearía otro prototipo en base al segundo que cumpla lo mayor posible los requerimientos por parte de la empresa.

1. **Esquema GANTT**

[Enlace al GANTT](file:///C:\Users\Santi_14\Documents\Proyecto\Gantt_Majime_Segunda_Entrega.pdf)

1. **Actas de reunión**
2. **Norma ISO/EIC 912**

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software.

* 1. **Disponibilidad**

Garantizamos una disponibilidad del 95% del sistema de las horas de uso.

En caso de fallas del sistema en cuanto a software, se ofrecen versiones estables de respaldo listas para equiparse.

En caso de fallas del sistema a nivel físico, se tendrá una línea de emergencia con un técnico en la otra línea. De ser necesario este podría coordinar una visita para recuperar la funcionalidad con los datos respaldados previamente. También aplica en caso de fallas con los terminales en uso que fueron equipados por nosotros.

Utilizamos versiones de software que a pesar de no ser las ultimas, si son las más estables.

* 1. **Eficiencia**

El tiempo de respuesta estimado por consulta será de hasta 2 segundos, pero la mayoría de estas serán realizadas en 1 o menos, dependiendo el uso del servidor.

Con el hardware proporcionado por nosotros, garantizamos una funcionabilidad de hasta 30 usuarios si se eligió el presupuesto más bajo, y de 50 en caso de que el presupuesto elegido sea el mayor, realizando estas consultas simultáneamente.

* 1. **Modificabilidad**

El software esta escrito de forma tal que, este sea fácil de modificar, agregando o quitando funcionalidades de este. Cualquier cambio que sea necesario dependiendo del tamaño, si es pequeño puede llevar hasta 1 semana, incluyendo planificación y testeo del mismo. Cambios grandes que involucren la creación de nuevas pestañas, cambios en la base de datos, etc; pueden llevar de 1 semana a 1 mes.

Los cambios de hardware solicitados pueden demorarse hasta 2 semanas si se requiere el cambio del motor de bases de datos o el sistema operativo servidor, junto a él armado o modificación de los dispositivos físicos. En caso de que solo se necesiten cambios en el hardware, estos tendrán una demora máxima de una semana.

* 1. **Usabilidad**

La interfaz del Software está hecha con la intención de tener un uso fácil para usuario. Las interfaces están hechas con la intención de que el usuario sepa siempre en qué sección del programa se realiza tal tarea. La aplicación contara con un manual de usuario para el rápido aprendizaje del mismo, conteniendo ilustraciones para fácil guía de este. A su vez la curva de aprendizaje es notoriamente baja.  
El diseño de este cuenta con un fondo amigable a la vista del usuario para su comodidad, este dispone de botones con íconos que guían al usuario a las diferentes secciones del programa.

* 1. **Portabilidad**

El sistema cuenta con una portabilidad eficaz para la rápida instalación de este. Se podría transferir este programa a una empresa que tuviera similares requerimientos. En caso de que se cambie el hardware utilizado no se encontrarían problemas mientras este cumpla con los requisitos mínimos.

Se debería adaptar el programa en caso de que se cambie el sistema operativo del servidor o el motor de base de datos, pero esto podría ser realizado en un plazo inferior a 15 días hábiles. No se soporta un cambio en el lenguaje de programación, pero si se soporta una actualización a una versión más moderna de este.

* 1. **Seguridad**
     1. **Identificación**

El sistema de registro solicitara una identificación del usuario, la cual es la CI del mismo, después de esto se le solicitara una contraseña, otros datos personales, y el identificador del rango, el cual es una clave previamente dicha por el superior para garantizar la seguridad de los permisos.

* + 1. **Autentificación**

El usuario requerirá ingresar su cedula y contraseña la cual ya estará guardada en el servidor de base de datos de la empresa. La contraseña que él ingrese se mostrará censurada.

* + 1. **Autorización**

Se cuenta con protecciones para evitar que personal no autorizado utilice funcionalidades a las que no tiene permitido acceder.

* + 1. **Confidencialidad**

Las credenciales serán almacenadas bajo un algoritmo de encriptación secreto, para evitar posibles fugas de datos. Ningún usuario a excepción de un gerente podrá ver los datos personales de los usuarios registrados.

* + 1. **Análisis de patrones de acceso**

Se contará con un sistema que guarda registro de los intentos fallidos al momento de iniciar sesión en el programa, guardando la dirección IP del equipo donde se realizó. Después de excesivos intentos fallidos, se le negara el acceso de forma temporal, para evitar hackeos por fuerza bruta. Además, se guardará el registro de aquel usuario que realice un número de peticiones fuera de lo normal.

* + 1. **Recuperación del Sistema**

Todos los datos estarán respaldados localmente y en la nube por lo que se estará preparado para una rápida recuperación de los datos en caso de pérdida, corrupción u otro inconveniente con los datos.

* + 1. **Auditoría**

Además de los registros guardados por un uso anormal, se guardará un registro más básico de todas las acciones realizadas en el programa, el autor, y el momento.

1. **Bibliografía**

Ciclo de vida de desarrollo de software. Extraído de <https://www.efectodigital.online/post/2018/04/23/ciclo-de-vida-de-desarrollo-de-software>

1. Principio del formulario