**Plan de Desarrollo de Software**

Proyecto:

Propuesta de desarrollo de un sistema de información y gestión de servicios de restaurantes

Revisión 1.0

Pendiente logo

**Contenido**

**1 Introducción 3**

1.1 Propósito 3

1.2 Alcance 3

1.3 Resumen 4

**2. Vista general del proyecto 5**

2.1 Propósito, alcance y objetivos 5

2.1.1 Propósito 5

2.1.2 Alcance 5

2.1.3 Objetivos 5

2.2 Suposiciones y restricciones 6

2.3 Entregables del proyecto 6

2.4 Evolución del plan de desarrollo de software 8

**3. Organización del proyecto 9**

3.1 Participantes en el proyecto 9

3.2 Roles y responsabilidades 9

**4. Gestión del proyecto 11**

4.1 Estimaciones del proyecto 11

4.2 Plan del proyecto 11

4.2.1 Plan de las fases 11

4.2.2 Calendario del proyecto 12

4.3 Seguimiento y control del proyecto 16

**5. Referencias 17**

**1. Introducción**

Este documento provee una visión general del plan de desarrollo del software “EasyOrder”, así como la organización de los miembros y la función de cada uno, para obtener un sistema optimo y eficaz.

El enfoque de desarrollo propuesto es el “eXtream Programming” o “Programación Extrema” (XP); de acuerdo a las características del proyecto, se seleccionan los roles, las actividades a realizar y los documentos entregables que serán generados.

* 1. **Propósito**

El propósito del plan de desarrollo de software es proporcionar la información necesaria para el control del proyecto, y tener un estimado del tiempo oportuno de entrega del mismo.

* 1. **Alcance**

Se desarrollará los requisitos del producto desde la perspectiva de quienes interactuaran con el sistema para una mejor acogida y facilidad de uso.

Se realizará el análisis en base a los requerimientos del cliente y a la información sustraída gracias a los métodos de recolección de datos tales como:

* La encuesta
* La observación
* La consulta de expertos

Se realizarán las primeras versiones de los casos de uso y sus correspondientes diagramas, el modelo conceptual de la solución propuesta, los requerimientos funcionales y no funcionales que encontrarán en el documento SRS.

* 1. **Resumen**

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

* Vista general del proyecto: proporciona una descripción del propósito, el alcance y los objetivos previstos para este software, las suposiciones y restricciones, la documentación entregable y la evolución del plan de desarrollo de software.
* Organización del proyecto: proporciona la información de los participantes del proyecto, sus roles y responsabilidades.
* Gestión del proceso: proporciona las estimaciones del proyecto, el plan del proyecto y su seguimiento y control.

**2. Vista general del proyecto**

**2.1 Propósito alcance y objetivos**

La información que a continuación se incluye ha sido extraída de los métodos de recolección de datos y las reuniones con el grupo de trabajo de desarrollo de software.

**2.1.1 Propósito**

En este documento se han estructurado los requerimientos funcionales y no funcionales del sistema de información **easyorder** que se dispone para mejorar la atención al cliente en restaurantes. Su objetivo es definir cada requisito funcional que se va a implementar en el sistema de información; además de todos los recursos de infraestructura utilizados en el desarrollo e implementación de este aplicativo, obteniendo así el control e integración de dichos recursos y formando una base sólida en el desarrollo del proyecto.

**2.1.2 Alcance**

## Diseño, desarrollo e implantación de un Sistema de Información que gestiona los pedidos de clientes en restaurantes, además de servir de apoyo gerencial.

**Easyorder** es un aplicativo WEB, permitirá a cualquier restaurante que lo implemente, consultar y administrar las solicitudes hechas por sus clientes, respecto a la solicitud de sus servicios, además que integra funcionalidades de gestión de empleados, módulos de inventario y consultas que busca solventar necesidades administrativas.

**2.1.3 Objetivos**

* Administrar solicitudes de servicios de los clientes.
* Llevar un seguimiento de la solicitud del cliente.
* Easyorder será accesible al cliente instruyéndolo intuitivamente en el uso del mismo al realizar la solicitud.
* Administrar las tareas de los empleados del restaurante respecto a la solicitud del cliente.
* Llevar un registro de las actividades de los usuarios del sistema.
* Llevar un registro de inventario.
* Administrar la disposición de servicios (platos y promociones) del restaurante

Gestión de calificación del servicio.

**2.2 Suposiciones y restricciones**

* El Cliente debe contar con un dispositivo móvil para ingresar al sistema.
* Para ingresar al sistema el Cliente debe contar con una conexión a internet.

**2.3 Entregables del proyecto**

Los documentos descritos a continuación según la programación extrema podrán modificarse con forme el avance y desarrollo del software lo requiera, concluyendo así que las versiones finales de los documentos estarán en su totalidad a la finalización del proyecto.

* **Plan de desarrollo de software: e**s el presente documento.
* **Documento srs:** el estándar ieee 830-1998 para el srs(en inglés) o ers (especificación de requerimientos de software) es un conjunto de recomendaciones para la especificación de los requerimiento o requisitos de software el cual tiene como producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas.
* **Modelo de casos de uso:** el modelo de casos de uso presenta las funciones del sistema y los actores que hacen uso de ellas. se representa mediante diagramas de casos de uso.
* **Especificaciones de casos de uso:** para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. también, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un diagrama de actividad.
* **Prototipos de interfaces de usuario:** se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. sólo los de este último tipo serán entregados al final de la fase de elaboración, los otros serán desechados. asimismo, este artefacto, será desechado en la fase de construcción en la medida que los resultados de las iteraciones vayan desarrollando el producto final.
* **Modelo de análisis y diseño:** este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.
* **Modelo de datos:** previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. para expresar este modelo se utiliza un diagrama de clases (donde se utiliza un profile uml para modelado de datos, para conseguir la representación de tablas, claves, etc.).
* **Modelo de implementación:** este modelo es una colección de componentes y los subsistemas que los contienen. estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (este modelo es sólo una versión preliminar al final de la fase de elaboración, posteriormente tiene bastante refinamiento).
* **Casos de prueba:** cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. estos casos de prueba son aplicados como pruebas de regresión en cada iteración. cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.
* **Lista de riesgos:** este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

**2.4 Evolución del plan de desarrollo de software**

El plan de desarrollo de software se revisará y refinará antes del comienzo de cada iteración.

**3. Organización del proyecto**

**3.1 Participantes en el software**

* Anderson Rodríguez
* Diego Murillo
* Marvy Martínez
* Moisés Urueña
* Nancy Rojas

**3.2 Roles y responsabilidades**

A continuación, se describen las responsabilidades de los roles del equipo de desarrollo de software.

|  |  |
| --- | --- |
| **Puesto** | **Responsabilidad** |
| Programador | Responsable sobre el código, el diseño (refactorización, simplicidad), la integridad del sistema (pruebas). |
| Tracker | Recoge, analiza y publica información sobre la marcha del proyecto sin afectar demasiado el proceso, supervisa el cumplimiento de la estimaciones en cada iteración, Informa sobre la marcha de la iteración en curso, controla la marcha de las pruebas funcionales, de los errores reportados, de las responsabilidades aceptadas y de las prueba añadidas por los errores encontrados. |
| Coach | Experto en XP, responsable del proceso en su conjunto, identifica desviaciones y reclama atención sobre las mismas, guía el grupo de forma indirecta, interviene directamente si es necesario, ataja rápidamente el problema. |
| Encargado de pruebas | Ejecuta las pruebas funcionales y publica los resultados. |

**4. Gestión del proceso**

**4.1 Estimaciones del proyecto**

El presupuesto del proyecto no se ha establecido y los recursos involucrados se encuentran en el documento SRS.

**4.2 Plan del proyecto**

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

**4.2.1 Plan de las fases**

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase.

|  |  |  |
| --- | --- | --- |
| **Fase** | **Nro. Iteraciones** | **Duración** |
| Fase de inicio | 5 | 5 Semanas |
| Fase de elaboración | 6 | 6 Semanas |
| Fase de construcción | 24 | 24 Semanas |
| Fase de transición |  |  |

Los hitos que marcan el final de cada fase se describen en la siguiente tabla

|  |  |
| --- | --- |
| **Descripción** | **Hito** |
| Fase de inicio | En esta fase de desarrollaran los requisitos del producto desde las perspectiva del usuario, los cuales serán establecidos en la documentación entregable (ver “2.3 Entregables del proyecto”) |
| Fase de elaboración | En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura teniendo en cuenta los requisitos del producto, se identificaran y especificaran los casos de uso que serán implementados en las primera versión del software. |
| Fase de construcción | Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el modelo de análisis y diseño. |
| Fase de transición | En esta fase se dará por terminada la preparación del software y se marcara su fin con la entrega de la documentación entregable. |

**4.2.2 Calendario del proyecto**

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

|  |  |  |  |
| --- | --- | --- | --- |
| **Disciplinas / documentos generados o modificados durante la Fase de Inicio** | **Comienzo** | **Aprobación** | |
| **Modelado del Negocio** |  |  | |
| Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio | Semana 1  08/02 – 13/02 |  | |
| **Requisitos** |  |  | |
| Documento SRS | Semana 2  15/02 – 20/02 |  | |
| Modelo de Casos de Uso | Semana 3  22/02 – 27/02 |  | |
| Especificación de Casos de Uso | Semana 4  29/02 – 05/03 |  | |
| Especificaciones Adicionales | Semana 5  07/03 – 12/03 |  | |
| **Análisis/Diseño** |  |  | |
| Modelo de Análisis/Diseño |  | siguiente fase | |
| Modelo de Datos |  | siguiente fase | |
| **Implementación** |  |  | |
| Prototipos de Interfaces de Usuario |  | siguiente fase | |
| Modelo de Implementación |  | siguiente fase | |
| **Despliegue** |  |  | |
| Modelo de Despliegue |  | siguiente fase | |
| **Gestión de Cambios y Configuración** | Durante todo el proyecto | | |
| **Gestión del proyecto** |  | |  |
| Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones | Semana 2  15/02 – 20/02 | |  |
| **Ambiente** | Durante todo el proyecto | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Disciplinas / documentos generados o modificados durante la Fase de Elaboración** | **Comienzo** | **Aprobación** | |
| **Modelado del Negocio** |  |  | |
| Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio | Semana 1  08/02 – 13/02 |  | |
| **Requisitos** |  |  | |
| Documento SRS | Semana 2  15/02 – 20/02 |  | |
| Modelo de Casos de Uso | Semana 3  22/02 – 27/02 |  | |
| Especificación de Casos de Uso | Semana 4  29/02 – 05/03 |  | |
| Especificaciones Adicionales | Semana 5  07/03 – 12/03 |  | |
| **Análisis/Diseño** |  |  | |
| Modelo de Análisis/Diseño | Semana 6  14/03 – 19/03 | Revisar en cada iteración | |
| Modelo de Datos | Semana 7  28/03 – 02/04 | Revisar en cada iteración | |
| **Implementación** |  |  | |
| Prototipos de Interfaces de Usuario | Semana 8-9  04/04 – 16/04 | Revisar en cada iteración | |
| Modelo de Implementación | Semana 10-11  18/04 – 30/04 | Revisar en cada iteración | |
| **Despliegue** |  |  | |
| Modelo de Despliegue |  | Revisar en cada iteración | |
| **Gestión de Cambios y Configuración** | Durante todo el proyecto | | |
| **Gestión del proyecto** |  | |  |
| Plan de Desarrollo del Software en su versión 2.0 y planes de las Iteraciones | Semana 2  15/02 – 20/02 | | Revisar en cada iteración |
| **Ambiente** | Durante todo el proyecto | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Disciplinas / documentos generados o modificados durante la Fase de Construcción** | **Comienzo** | **Aprobación** | |
| **Modelado del Negocio** |  |  | |
| Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio | Semana 1  08/02 – 13/02 |  | |
| **Requisitos** |  |  | |
| Documento SRS | Semana 2  15/02 – 20/02 |  | |
| Modelo de Casos de Uso | Semana 3  22/02 – 27/02 |  | |
| Especificación de Casos de Uso | Semana 4  29/02 – 05/03 |  | |
| Especificaciones Adicionales | Semana 5  07/03 – 12/03 |  | |
| **Análisis/Diseño** |  |  | |
| Modelo de Análisis/Diseño | Semana 6  14/03 – 19/03 | Revisar en cada iteración | |
| Modelo de Datos | Semana 7  28/03 – 02/04 | Revisar en cada iteración | |
| **Implementación** |  |  | |
| Prototipos de Interfaces de Usuario | Semana 8-9  04/04 – 16/04 | Revisar en cada iteración | |
| Modelo de Implementación | Semanas 10-11  18/04 – 30/04  Semanas 12-32  02/05 – 17/09 | Revisar en cada iteración | |
| **Despliegue** |  |  | |
| Modelo de Despliegue | Semanas 33-36  19/09 – 15/10 | Revisar en cada iteración | |
| **Gestión de Cambios y Configuración** | Durante todo el proyecto | | |
| **Gestión del proyecto** |  | |  |
| Plan de Desarrollo del Software en su versión 3.0 y planes de las Iteraciones | Semana 2  15/02 – 20/02 | | Revisar en cada iteración |
| **Ambiente** | Durante todo el proyecto | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Disciplinas / documentos generados o modificados durante la Fase de Transición** | **Comienzo** | **Aprobación** | |
| **Modelado del Negocio** |  |  | |
| Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio | Semana 1  08/02 – 13/02 |  | |
| **Requisitos** |  |  | |
| Documento SRS | Semana 2  15/02 – 20/02 |  | |
| Modelo de Casos de Uso | Semana 3  22/02 – 27/02 |  | |
| Especificación de Casos de Uso | Semana 4  29/02 – 05/03 |  | |
| Especificaciones Adicionales | Semana 5  07/03 – 12/03 |  | |
| **Análisis/Diseño** |  |  | |
| Modelo de Análisis/Diseño | Semana 6  14/03 – 19/03 | Revisar en cada iteración | |
| Modelo de Datos | Semana 7  28/03 – 02/04 | Revisar en cada iteración | |
| **Implementación** |  |  | |
| Prototipos de Interfaces de Usuario | Semana 8-9  04/04 – 16/04 | Revisar en cada iteración | |
| Modelo de Implementación | Semanas 10-11  18/04 – 30/04  Semanas 12-32  02/05 – 17/09 | Revisar en cada iteración | |
| **Despliegue** |  |  | |
| Modelo de Despliegue | Semanas 33-36  19/09 – 15/10  Semanas 37-39  24/10 – 05/11 | Revisar en cada iteración | |
| **Gestión de Cambios y Configuración** | Durante todo el proyecto | | |
| **Gestión del proyecto** |  | |  |
| Plan de Desarrollo del Software en su versión 4.0 y planes de las Iteraciones | Semana 2  15/02 – 20/02 | | Revisar en cada iteración |
| **Ambiente** | Durante todo el proyecto | | |

**4.3 Seguimiento y control del proyecto**

**Gestión de requisitos**

Los requisitos del sistema son especificados en documento SRS, los requisitos serán modificados si el desarrollo del sistema lo requiere.

**Control de plazos**

El calendario del proyecto llevara un seguimiento con forme avance el desarrollo del sistema.

**Control de calidad**

Los defectos detectados llevaran un control y documentación, también de la solución de cada uno de estos para asegurar la calidad del sistema.

**Gestión de riesgos**

A partir de la fase de inicio se llevará una lista de riesgos potenciales y su posible solución, estos serán reevaluados antes de cada iteración.

**Gestión de configuración**

Se realizará una gestión de configuración para llevar un registro de los avances de cada iteración; estos se publicaran para que todos los integrantes del equipo de desarrollo tenga acceso a él.

**Referencias**

Propuesta de desarrollo de un sistema de información y localización geográfica de sitios de interés en la ciudad de Neiva, Plan de desarrollo de software versión 1.0.

SRS / ERS Especificación de requerimientos de software, http://www.icesi.edu.co/departamentos/tecnologias\_informacion\_comunicaciones/proyectos/lisa/home/analisis/srs/srs

Roles de la metodología XP, https://sites.google.com/site/xpmetodologia/marco-teorico/roles

Arc\_2574, https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0ahUKEwixidHF5fXLAhUD2R4KHRRaASYQFghCMAM&url=http%3A%2F%2Faulati.unipamplona.edu.co%2Funipamplona1%2Fhermesoft%2FaulaTI%2Fmaterial%2Fmedios%2Farc\_2574.doc&usg=AFQjCNFYInurdnnZFzox0sD0JBOngtt-qQ&sig2=PQVagym8dIbb\_5QBp1Oc1Q&bvm=bv.118443451,d.dmo&cad=rja