**Documento de requisitos**

**De**

**“Sistema de Control y registro de tiempos de repartidores”**

Fecha 10/03/2015

Versión 0.1

Integrantes Vilchis Ontiveros

Del equipo Juan Manuel

Armenta Bojórquez

Alejandro

Melgar Mejía

Arturo Christopher

Valenzuela Moreno

Saúl Gerardo

Prefacio

Este es el Documento de Requisitos del proyecto de control de tiempos de repartición de piezas en la empresa Autored, llamado “Sistema de Control y registro de tiempos de repartidores”, el cual tiene como propósito el desarrollo de un sistema software capaz de asignar un tiempo predefinido a cada viaje del repartidor junto con los detalles del mismo y guardar un registro para tomar decisiones en base a la información recopilada

**Alcance del documento**

El Documento de Requisitos es la base de todo el desarrollo futuro de

*“Sistema de Control y registro de tiempos de repartidores”*. Describe los siguientes aspectos del sistema:

Propósito, contexto, requisitos funcionales, requisitos de pruebas, requisitos

de ambiente, arquitectura del sistema y riesgos del proyecto.

**Documentos relacionados**

Documento de Inicio de Proyecto, proyecto “Estacionamiento Inteligente”,

Versión 0.1.

**Autores**

Vilchis Ontiveros Juan Manuel, Armenta Bojórquez Alejandro, Melgar Mejia Christopher, Valenzuela Moreno Saúl Gerardo

**Lectores**

Este documento está dirigido principalmente a los desarrolladores del

proyecto, pero es de interés de todos los interesados en el mismo.

Historia del Documento

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Explicación del cambio | Autor |
| 0.1 | 0.1 10/03/2015 | Primer Borrador | Vilchis Ontiveros Juan Manuel, Armenta Bojorquez Alejandro, Melgar Megia Christopher, Valenzuela Moreno Saul Gerardo |

**1. Introducción**

En esta introducción se describe brevemente el contexto, objetivo y alcance del proyecto a desarrollar, así como la documentación relativa al mismo.

1. Propósito

El sistema a desarrollar busca generar un control de tiempo de entrega de los repartidores en base a una asignación automática de tiempo calculado en base al destino del viaje y el registro de todos los parámetros del viaje realizado.

2. Alcance

El alcance de este proyecto se enfoca al cálculo de duración de tiempos por viaje en base a su(s) destinos y el registro de dichos viajes.

3. Contexto

El programa se desarrollara usando Visual Basic en el entorno de trabajo de Visual Studio 2012. Con códigos ya anteriormente probados sobre la manipulación de datos y la persistencia usando MySQL.

4. Referencias

**2. requisitos del sistema**  
1. Requisitos funcionales

Rf1. El software debe de ejecutarse al momento en que un repartidor salga a una vuelta o mandado de la empresa.  
  
Rf2 El software mediante la huella digital del repartidor debe de iniciar el conteo del tiempo hacia su destino también para parar el conteo de tiempo deberá de proporcionar la huella.  
  
Rf3 Al final del mes o quincena el software deberá ser capaz de poder generar un reporte con sus vueltas y destinos generados por cada repartidor, también como calcular el dinero ganado por las vueltas.

1.1. Requisitos no funcionales

Rnf1. El software puede contener errores al momento de iniciar el contador y crear distorsión en los tiempos de llegada.

Rnf2. Podría haber errores al momento de leer la huella digital del repartidor y al no tener otra opción disponible para marcar los tiempos habría costos para la empresa.

Rnf3. Solo usuarios registrados para acceder al sistema pueden manipular los reportes de ganancias generados para cada repartidor.  
  
  
**2.** Requisitos de interfaz

La tabla 1 muestra la lista de eventos externos a los que el sistema responde.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Evento | Descripción | Indicador | Parámetros | Respuesta |
| Evento externo 1 | El sistema debe detectar cuando un repartidor sale a la ruta. | El repartidor al salir a la ruta. | El parámetro será el repartidor que debe salir. | Inicio del procesamiento |
| Evento externo 2 | El sistema debe detectar las vueltas seleccionadas y registrarlas en un su base de datos. | El software será quien inicie el evento, al detectar la huella digital del repartidor. | El parámetro será la huella digital del repartidor. | El software al detectar la huella digital será registrado en la base de datos |
| Evento 3 | El sistema debe calcular las vueltas y el tiempo que se encuentra una sucursal de otra. | El software es el que inicie este evento al detectar por segunda vez la huella digital del repartidor. | El parámetro será la huella digital del repartidor. | El sistema entregara las vueltas realizadas en un día, semana, quincena o mes, como el tiempo y los lugares concurridos como según se requiera en la empresa. |

|  |  |  |
| --- | --- | --- |
| Respuestas | Descripción | Parámetros |
| Respuesta al evento 1 | Se inicia el software mediante la captura de la huella digital de cada repartidor. | Detecta la huella digital de cada repartidor. |
| Respuesta al evento 2 | El sistema detecta y almacena en una base de datos al repartidor que sale a realizar su recorrido. | Las huellas de los repartidores |
| Respuesta al evento 3 | El software calcula el tiempo y las vueltas realizadas al día, mes o quincena de cada repartidor. | El número de vueltas realizadas por repartidor se encuentran almacenadas en la base de datos de registros. |

3. Manejo de excepciones

Excepción 1: En caso de que el sistema no funcione se tendrá que reniciar el equipo y volver a ejecutar el software.  
  
Excepción 2: En caso de que el sistema no detecte la huella del repartidor saliente.

Excepción 3: En caso de que el sistema no lleve el conteo del tiempo y de las vueltas.  
  
  
La siguiente tabla muestra las condiciones de excepción que se pueden dar en el sistema en relación con eventos externos e internos.

|  |  |  |
| --- | --- | --- |
| Excepción | Descripción | Eventos o situación en que ocurre |
| 1 | Error del sistema | Puede ocurrir por falta de memoria o problemas en el código. |
| 2 | No detecte la huella del repartidor | Puede ocurrir que el repartidor no este registrado o que el aparato este averiado. |
| 3 | No lleve el conteo del tiempo y de las vueltas del repartidor | Puede ocurrir por falta de un cálculo no bien echo dentro del código del sistema. |

**4. Requisitos de prueba.**  
  
Son la clases de pruebas que se harán sobre el sistema para determinar que se cumplen los requisitos funcionales.

**RP1.**  
\* Probar que el sistema se ejecute al momento de salir el repartidor.  
\* Probar que el software funcione correctamente cuando salga diferentes repartidores.  
\* probar que el software se mantenga activo y estable en el transcurso del día.

**RP2.**  
\* Probar que el software registre correctamente las huellas digitales.  
\* Probar que el software detecte correctamente las huellas digitales de cada repartidor.  
\* Probar correctamente el registro de cada repartidor.  
 **RP3**  
\* Detectar si el repartidor coloco su huella digital para parar el tiempo.  
\* El software rescate correctamente los datos de la base de datos.  
\* Calcular el tiempo y las vueltas de cada repartidor.

**5. Matriz de Requisitos Funcionales y de Pruebas**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RP1** | **RP2** | **RP3** |
| **Rf1.**  El software debe de ejecutarse al momento en que un repartidor salga a una vuelta o mandado de la empresa. | **x** |  |  |
| **Rf2.**  El software mediante la huella digital del repartidor debe de iniciar el conteo del tiempo hacia su destino también para parar el conteo de tiempo deberá de proporcionar la huella. |  | **x** |  |
| **Rf3.**  Al final del mes o quincena el software deberá ser capaz de poder generar un reporte con sus vueltas y destinos generados por cada repartidor, también como calcular el dinero ganado por las vueltas. |  |  | **x** |

Casos de Uso

Sistema que controla el área de repartos en Autored Refaccionarias

|  |  |
| --- | --- |
| Responsabilidades del actor | Responsabilidad del sistema |
| Registrar al repartidor |  |
| Seleccionar el destino | Registra el destino |
|  | Calcula el tiempo del destino |
| Agrega uno o más destinos en curso. | Recalcula el tiempo del destino más el que ya estaba en curso. |
| Finaliza el viaje | Calcula el total de los destinos de cada viaje. |
|  |  |
|  |  |

Registrar al repartidor.  
El actor se encarga de registrar cada repartidor en su sucursal correspondiente.  
  
Seleccionar el destino.  
El actor es el que selecciona el destino a donde mandara a la entrega de la refacción.  
  
Registra el destino.  
El sistema se encarga de registrar el destino de cada repartidor para saber a donde fue y localizarlo.

Agrega uno o más destino en curso.

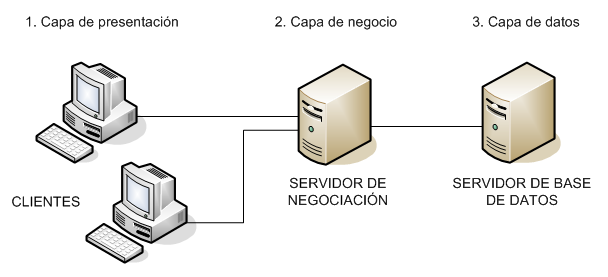
Se presenta la opción de que el repartidor se encuentre en un destino y se le solicite atender otro destino. Por ejemplo, el repartidor salió de la sucursal de victoria a la sucursal de Sanalona, entonces llega un cliente a Victoria y le solicita una pieza que se encuentra en lombardo, la sucursal victoria tiene que hablar al repartidor para informarle que vaya a Lombardo por la refacción del cliente aquí ya se originó otro destino pero el actor es el que manipula esta opción.  
  
Recalcula el tiempo del destino más el que ya estaba en curso.  
El sistema es el encargado en llevar la sumatoria del total de los destinos de cada repartidor.  
  
Finaliza el viaje.  
El actor es el que se encarga de finalizar el viaje cuando el repartidor llega a su origen.  
Calcula el total de los destinos de cada viaje.  
El sistema es el encargado en hacer una sumatoria total por día, por semana, por quincena o por mes según sea requerido en la empresa.

Arquitectura del sistema

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación seria el modelo de interconexión de sistemas abiertos.

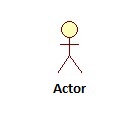
Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.



Capas:

1. **Capa de presentación (Clientes):** la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.
2. **Capa de negocio (Servidor de negociación):** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
3. **Capa de datos (Servidor de base de datos):** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Casos de uso.



Sistema

|  |
| --- |
| Se abre el sistema |
| Se registra cada repartidor en su sucursal correspondiente |
| Se selecciona el destino a mandar la entrega |
| Registra el destino de cada repartidor |
| Calcula y registra el tiempo al destino a llegar |
| El repartidor se dirige a su destino |
| Le indica al repartidor a dirigirse a otra sucursal |
| Recalcula y registra el tiempo del primer destino más el tiempo del nuevo destino a llegar |
| Se finaliza el viaje al llegar al punto de origen |
| Calcula el total de los destinos de cada viaje. |
| Se encarga de hacer la sumatoria total por dia, semana, quincena o mes |