



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

TRATAMIENTO DE INFORMACIÓN A TRAVES DE PDA

Alumno: Maria Georgieva Caneva

Tutor: Jesús Villadangos Alonso

Pamplona, Junio de 2010

Agradezco a mi tutor, Jesus, por haberme ofrecido este proyecto para finalizar mis estudios y por su interés, esfuerzo, rapidez a la hora de contestar todos los correos, aguante y ayuda para que todo salga bien.

Agradezco a mis compañeros de universidad y sobre todo a mis compañeros de laboratorio especialmente a Leti, por toda la ayuda mostrada a lo largo de la carrera y por supuesto, por aguantarme después de haber pasado tantas horas juntas y de seguir siendo las mejores amigas para todo..

Agradezco a toda mi familia y a mi novio Rumen , por preocuparse por mí en los momentos de agobio (sobre todo estos últimos días), por estar siempre pendientes de saber qué tal estoy y por supuesto, por estar siempre a mi lado.

Agradezco a toda la gente, amigos, conocidos... que me aguantan tal y como soy, alegre y muy divertida, y que por supuesto también me aguantan en mis momentos malos, tristes...

Gracias a todos

INDICE

CAPÍTULO 1: INTRODUCCIÓN	5
1.1 INTRODUCCIÓN	5
1.2 OBJETIVOS	5
1.3 DESCRIPCIÓN DEL PROBLEMA	5
1.4 REQUISITOS	7
1.4.1. Requisitos Generales	7
1.4.2. Requisitos de hardware y de sistema	7
1.4.3. Requisitos de la base de datos	9
1.5. LENGUAJE DE PROGRAMACIÓN	9
CAPÍTULO 2: ANÁLISIS DEL PROBLEMA	11
2.1. DIAGRAMA DE CASOS DE USO	11
2.1.1. Caso de uso general	11
2.1.2. Caso de Uso recogida información de PDA	12
2.1.3. Caso de Uso para la Empresa	14
CAPÍTULO 3: DISEÑO	17
3.1. DIAGRAMA DE CLASES Y DE SECUENCIA	17
3.1.1 Diagrama De Secuencia	17
3.1.2 Diagrama De Clases	18
3.2 DIAGRAMA ENTIDAD-RELACIÓN	18
3.2.1. Modelo entidad /relación de la BD	19
3.2.1.1. Esquema Entidad/Relación	19
3.2.1.2 Analisis de entidades y atributos	19
3.2.1.3. Análisis de las Relaciones	20
3.2.1.4 Paso a Tablas	21
CAPÍTULO 4: IMPLEMENTACIÓN	23
4.1 Visual C# y SQL Server Compact 3.5	23
4.2 Organización del código de la Aplicación	23
4.2.1. Clases	23
CAPÍTULO 5: TEORIAS PARA EL CÁLCULO DE DISTANCIAS ENTRE CADENAS	31

CAPÍTULO 6: RESULTADOS OBTENIDOS DE LOS ESTUDIOS	32
CAPÍTULO 7 : SIMULACIÓN	51
CAPÍTULO 8 : CONCLUSIONES	54
CAPÍTULO 9: LINEAS FUTURAS	55
CAPÍTULO 10 : BIBLIOGRAFÍA	56
ANEXO	58

CAPÍTULO 1: INTRODUCCIÓN

1.1 INTRODUCCIÓN



En los últimos años las industrias han optado por automatizar todos los trabajos que antes realizaban manualmente y en papel, para poder obtener un rendimiento mayor de los profesionales y que los resultados sean de más fácil manejo. Para ello la existencia de las PDAs ha sido de gran ayuda, ya que son aparatos electrónicos que ocupan poco lugar y abarcan un gran espectro de funciones.

1.2 OBJETIVOS

El objetivo del proyecto es obtener el análisis, diseño e implementación de una aplicación con la cual una empresa será capaz de conocer el flujo de trabajo realizado por sus trabajadores gracias a la información almacenada en una PDA. Ya que estos dispositivos son usados para almacenar información que puede ser consultada a cualquier hora y en cualquier lugar.

Para realizar la aplicación se va a simular la sincronización del dispositivo PDA, por medio de una Base de Datos implementada a través de SQL Server Compact 3.5 agregada a un proyecto de Visual Studio C#.

1.3 DESCRIPCIÓN DEL PROBLEMA

Las empresas normalmente tienen establecidos procedimientos de trabajo para las tareas habituales. Por ejemplo, considerando el caso de una empresa de mantenimiento, éstas tienen documentado las actividades a realizar para las revisiones periódicas.

En nuestro caso estamos considerando que la empresa tiene dichas actividades identificadas. También consideramos que el trabajador realiza las actividades previstas equipado con una PDA que automáticamente va a recoger información de estas actividades durante su desarrollo. En este caso, estamos asumiendo que el sistema en revisión dispone de aparatos que van notificando la posición del trabajador o bien la cercanía de las diferentes herramientas, etc. Es decir, mientras el operario va realizando los procedimientos de mantenimiento, la PDA va recibiendo notificaciones que almacena de forma automática.

A partir de las suposiciones anteriores, el problema se centra en resolver de forma automática qué procesos de mantenimiento han sido realizados por el operario a partir de las notificaciones recogidas por la PDA durante el desarrollo de las diferentes actividades de mantenimiento que ha realizado el operario.

Es importante destacar que nos interesa que sea de forma automática para evitar que el trabajador centre parte de su actividad en registrar las actividades, ya que esto puede retrasar en gran medida el desarrollo de las mismas.

Por medio de un estudio por parte de la empresa sobre los datos almacenados para el flujo de trabajo de los operarios, se podrá estimar qué procesos impuestos por la organización han sido realizados el operario. Es decir, que conociendo el flujo de trabajo del operario, la empresa estará capacitada para saber con certeza en cuanto ha cumplido su trabajador las actividades impuestas por ésta.

Además, a través de la PDA el operador a la hora de realizar un trabajo no tendrá que saber qué notificaciones tiene que seguir en cada momento ya que el dispositivo, le va a informar del flujo de trabajo. Los pasos que sigue el operario se van a ir almacenando en la PDA del operario y estos datos van a ser los datos de donde salen los datos para realizar el estudio del trabajo del operador.

Más adelante en la memoria se van a explicar detalladamente cada uno de los pasos que hay que seguir para poder alcanzar el correcto funcionamiento de la aplicación. A nivel de usuario (que serán los operadores y los encargados trabajadores de la empresa, que analizarán la información de la PDA) será un programa al cual se accederá a través del dispositivo, con sus pantallas, botones, ayudas, etc. Pero a nivel de analista/programador se analizará toda la información muy detalladamente para que todo el sistema funcione con precisión y sea fácil de usar para cualquier usuario.

Para ello, se comentará brevemente los pasos que se han seguido a lo largo del proyecto, así como las características más importantes sobre la funcionalidad de la herramienta:

- En la primera iteración la aplicación tiene que ser capaz de resolver los siguientes apartados:
 - recogida de datos
 - simular sincronización de la PDA
 - almacenar tiempo dedicado a cada notificación
 - insertar notificaciones en BD
- En la segunda iteración lo que se debe de realizar es:
 - el detectar por la empresa a través de la información almacenada en la PDA que actividad se ha realizado de verdad por el operador.
 - por medio de métodos de estudio la empresa será capaz de tener la certeza a través de distancia y % de cambios Levenshtein, notificaciones claves, de detectar las actividades verdaderamente realizadas por el operario.

1.4 REQUISITOS

En este apartado se mostrarán los requisitos a partir de los cuales se ha hecho el análisis y diseño de la aplicación software. Se explicarán los requisitos generales y a continuación los requisitos por categorías.

1.4.1. Requisitos Generales

El funcionamiento que se puede obtener con esta aplicación, tiene los siguientes puntos importantes:

Aplicación en la cual mediante un dispositivo (PDA), vamos a ir recibiendo las notificaciones que va realizando un operario cualquiera que sea su área de trabajo. Algunas de ellas pueden recibirse correctamente o incorrectamente. Por lo que debemos analizar el caso de fallos en la recepción de notificaciones. Estos fallos se van a modelar mediante el uso de valores aleatorios.

La aplicación debe tomar como entrada las notificaciones recibidas durante las operaciones de mantenimiento y dar como resultado los procesos/actividades que se consideran realizados. Este resultado se debe analizar para el caso de fallos en las notificaciones y se deben considerar diferentes alternativas o criterios para la toma de la decisión.

La aplicación debe considerar que en un proceso/actividad puede haber notificaciones que tienen que existir (relevantes) para poder asegurar la ejecución del proceso/actividad.

La aplicación tiene que proporcionar el grado de realización de la tarea propuesta como resultado del análisis de las notificaciones.

1.4.2. Requisitos de hardware y de sistema

Para instalar el Visual Estudio 2008, se requiere un ordenador con al menos las siguientes características:

Los Sistemas operativos soportados son: Windows XP Service Pack 2, Windows Server 2003 Service Pack 1, Windows Server 2003 R2, Windows Vista y Windows Server 2008

Requerimientos de Hardware: mínimo 1.6 GHz CPU, 384 MB RAM, resolución 1024 X 768 y disco duro a 5400 RPM.

A su vez, se requiere software de gestor de base de datos: que es Microsoft SQL Server 2008 Compact 3.5, siendo la base de datos local para esta aplicación, debido a que permite crear bases de datos compactas que se pueden implementar en equipos de escritorio, Smart Device como puede ser una PDA o el Tablet PC.

Finalmente, se requiere disponer de unas librerías, instaladas en el Visual C# , las que son:

`using System`

El espacio de nombres System, que contiene clases fundamentales y clases base que definen tipos de datos de referencia y de valor de uso frecuente, eventos y controladores de eventos, interfaces, atributos y excepciones de procesamiento.

`using System.Collections.Generic`

El espacio de nombres System.Collections.Generic, que contiene interfaces y clases que definen colecciones genéricas, permitiendo que los usuarios creen colecciones con establecimiento inflexible de tipos para proporcionar una mayor seguridad de tipos y un rendimiento mejor que los de las colecciones no genéricas con establecimiento inflexible de tipos.

Este espacio de nombres se necesita para poder utilizar correctamente la clase Dictionary, que representa una colección de claves y valores.

`using System.Linq`

El espacio de nombres System.Linq proporciona clases e interfaces que admiten consultas que utilizan Language-Integrated Query (LINQ).

`using System.Text;`

El espacio de nombres System.Text contiene clases que representan codificaciones de caracteres Unicode y UTF-8; clases base abstractas para la conversión de bloques de caracteres en bloques de bytes y viceversa; y una clase auxiliar que manipula y da formato a los objetos String sin necesidad de crear instancias intermedias de String.

`using System.Data;`

El espacio de nombres System.Data proporciona acceso a las clases que representan la arquitectura de ADO.NET. ADO.NET le permite generar componentes que administran eficazmente los datos de varios orígenes de datos.

El eje de la arquitectura ADO.NET es la clase DataSet. Cada DataSet puede contener varios objetos DataTable, y cada DataTable contiene datos de un solo origen, como SQL Server.

`using System.Data.SqlClient;`

El espacio de nombres System.Data.SqlClient es el proveedor de datos de .NET Framework para SQL Server.

El proveedor de datos de .NET Framework para SQL Server describe una colección de clases utilizadas para obtener acceso a una base de datos de SQL Server en el espacio administrado. Al utilizar SqlDataAdapter, se puede rellenar un objeto DataSet residente en memoria, que sirve para consultar y actualizar la base de datos.[7]

Cada proveedor de datos .Net Framework tiene una clase DataAdapter correspondiente que se utiliza como puente entre un origen de datos y un objeto DataSet.

`using System.Data.SqlServerCe; [6]`

El espacio de nombres System.Data.SqlServerCe es el proveedor de datos de .NET Compact Framework para SQL Server Mobile. El proveedor de datos de .NET Compact Framework para SQL Server Mobile es una colección de clases que se utiliza en el entorno de desarrollo administrado para proporcionar acceso a los dispositivos inteligentes a las bases de datos de SQL Server Mobile. Con System.Data.SqlServerCe, se pueden crear y administrar bases de datos de SQL Server Mobile en un dispositivo inteligente, así como establecer conexiones a las bases de datos de SQL Server.

1.4.3. Requisitos de la base de datos

Microsoft SQL Server 2008 Compact 3.5 es la base de datos local para esta aplicación, debido a que permite crear bases de datos compactas que se pueden implementar en equipos de escritorio, Smart Device como puede ser una PDA o el Tablet PC.

Para el correcto funcionamiento del sistema, tendrán que estar insertados previamente datos referentes a las actividades: notificaciones y su orden de realización en cada actividad, así como el detalle de saber que notificaciones es clave o no en cualquiera de las actividades.

1.5. LENGUAJE DE PROGRAMACIÓN

C# es un lenguaje orientado a objetos con seguridad en el tratamiento de tipos, que es a la vez sencillo y potente y permite a los programadores crear una gran variedad de aplicaciones. Combinado con .NET Framework, Visual C# permite la creación de aplicaciones para Windows, servicios Web, herramientas de base de datos, componentes, controles y mucho más. [3],[2], [B],[A]

Para el desarrollo de este proyecto se usará Visual Studio 2008 en su versión gratuita Express Edition. Dentro de este paquete hay varios apartados pero únicamente se utilizará Visual C# 2008 Express que permite la creación de aplicaciones de consola que admiten la entrada de datos y muestran resultados en la consola de la línea de comandos.[10]

Además, las aplicaciones de consola también resultan muy útiles para programas que requieren poca o ninguna interacción con el usuario que sería el caso de esta aplicación, ya que se pretende que el operario y los encargados del estudio estadístico interactúen lo menos posible con el dispositivo PDA.

A continuación se muestran los componentes de Visual Studio 2008 Express Edition empleados en el desarrollo de la herramienta.

Clase

Las clases son un tipo de datos muy eficaz. Como las estructuras, las clases definen los datos y el comportamiento del tipo de datos. Los programadores pueden crear objetos que son instancias de una clase. A diferencia de las estructuras, las clases admiten herencia, que es una parte fundamental de la programación orientada a objetos.

Método

Los métodos son un bloque de código que contiene una serie de instrucciones. En C#, cada instrucción se ejecuta en el contexto de un método. Se declaran en una clase o estructura especificando el nivel de acceso, el valor devuelto, el nombre del método y los parámetros de método. Los parámetros de método se incluyen entre paréntesis y separados por comas. Los paréntesis vacíos indican que el método no requiere ningún parámetro.

Struct

Un tipo Struct es un tipo de variable que se suele utilizar para encapsular pequeños grupos de variables relacionadas

Dictionary [13]

Representa una colección de claves y valores. La clase genérica Dictionary proporciona una asignación de un conjunto de claves a un conjunto de valores. Cada elemento que se agrega al diccionario está compuesto de un valor y su clave asociada. Recuperar un valor utilizando su clave es muy rápido, cerca de $O(1)$, porque la clase Dictionary se implementa como una tabla hash.

```
//la clase Dictionary representa una coleccion de claves y valores
Dictionary<string, double> idlist = operativo.GetIdList(idactividadvalue);
//foreach requiere el tipo de cada elemento de la coleccion
//como el diccionario es una coleccion de claves y valores
//el tipo del elemento es una estructura KeyValuePair del tipo de la clave
//y del tipo del valor
foreach (KeyValuePair<string, double> item in idlist)
{
    Console.WriteLine(string.Format("{0}- {1}", item.Key, item.Value));
}
```

DataSet[6]

Es una caché de memoria interna de datos recuperados de un origen de datos, representa un componente fundamental de la arquitectura de ADO.NET. DataSet está compuesto por una colección de objetos DataTable que se pueden relacionar entre ellos mediante objetos DataRelation. También se puede imponer la integridad de los datos de DataSet mediante los objetos UniqueConstraint y ForeignKeyConstraint. Para obtener información más detallada sobre el trabajo con objetos DataSet, vea Utilizar DataSets en ADO.NET. [9]

```
DataSet ds = new DataSet();
adp.Fill(ds);
Console.WriteLine("IDACT          ID");

foreach (DataRow row in ds.Tables[0].Rows)
{
    Console.WriteLine(string.Format("{0}          {1}", row["IDACT"], row["ID"]));
}
```

CAPÍTULO 2: ANÁLISIS DEL PROBLEMA

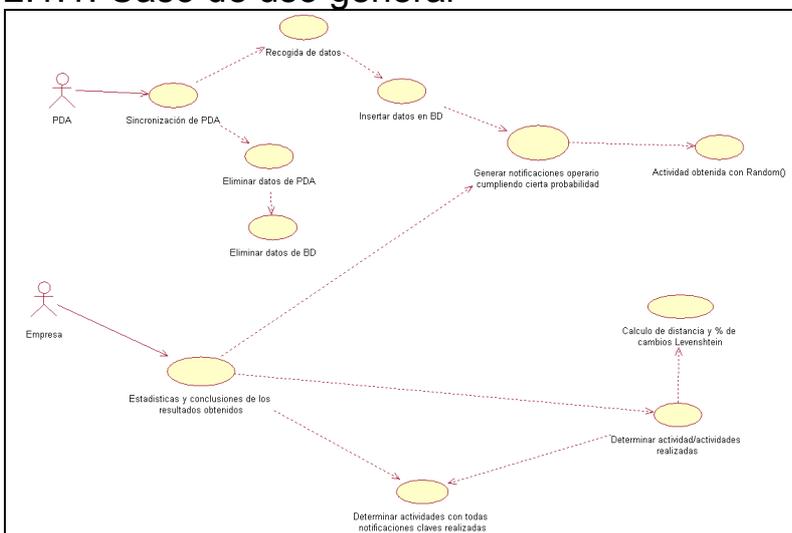
Una vez expuesta la introducción al problema, donde se ha indicado que es lo que se desea desarrollar e implementar, habrá que centrarse en el análisis.

Para ello se realizará un minucioso estudio sobre todos los casos de uso que aparezcan a lo largo del problema. Los casos de uso son descripciones de la funcionalidad asociada al sistema, pero con la ventaja de que son totalmente independientes a las siguientes fases del desarrollo (concretamente, independiente de la fase de implementación). Es decir, utilizar los diagramas de casos de uso en el análisis del problema no va unido a la implementación del proceso, precisamente el análisis de casos de uso es una representación del lenguaje natural entendible por todo tipo de usuario, donde se hace una descripción detallada de todas las funcionalidades de la herramienta software que se quiere diseñar.

2.1. DIAGRAMA DE CASOS DE USO

Tras haber recogido todos los requisitos en el apartado anterior y tras haber realizado una entrevista con el hipotético usuario de la herramienta, se comenzará a mostrar los diagramas de casos de uso que hemos obtenido. Los diagramas se mostrarán desde el caso más abstracto, iterando e incrementando, hasta llegar al diagrama que directamente se implementará en el lenguaje correspondiente, tal y como se formaliza en el proceso unificado.[C]

2.1.1. Caso de uso general



- **Actores:**

Los actores del caso de uso son la PDA (su sincronización) y el encargado del estudio estadístico de la empresa.

- **Descripción:**

Por medio del dispositivo PDA al operario se le indica cual va a ser su flujo de trabajo, que tendrá que ser conocido por la empresa para poder tener la certeza por medio de un estudio amplio donde se tendrán en cuenta diferentes criterios para la toma de decisión, de qué

trabajos ha completado dicho operario y en todo caso si no los ha completado, saber en cuanto se ha aproximado a estas actividades.

- **Precondiciones:**

Simulando el funcionamiento de la PDA, equivaldría a poder obtener una actividad de las que están almacenadas en la Base de Datos por medio de la función Random () conseguir sus notificaciones que cumplen una probabilidad que no es constante y considéralas como el flujo de trabajo del operario.

- **Poscondiciones:**

Según la opción que seleccione el usuario podrá:

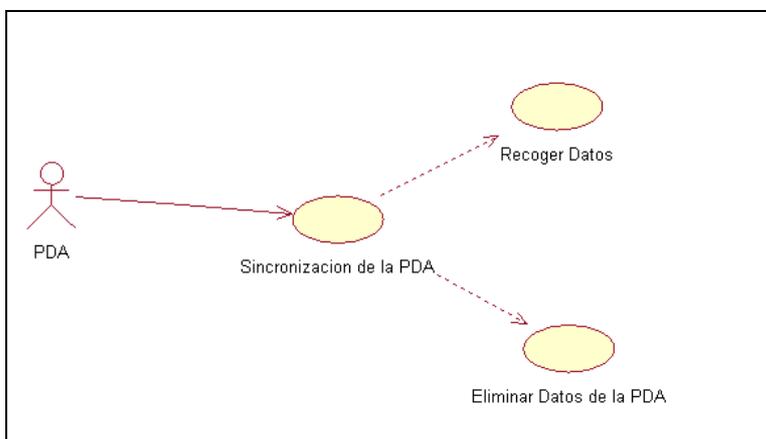
1. Conocer el flujo de trabajo del operador
2. Conocer las actividades realizadas por el operador, las completadas y las realizadas parcialmente, así como la distancia que existe entre unas y otras.

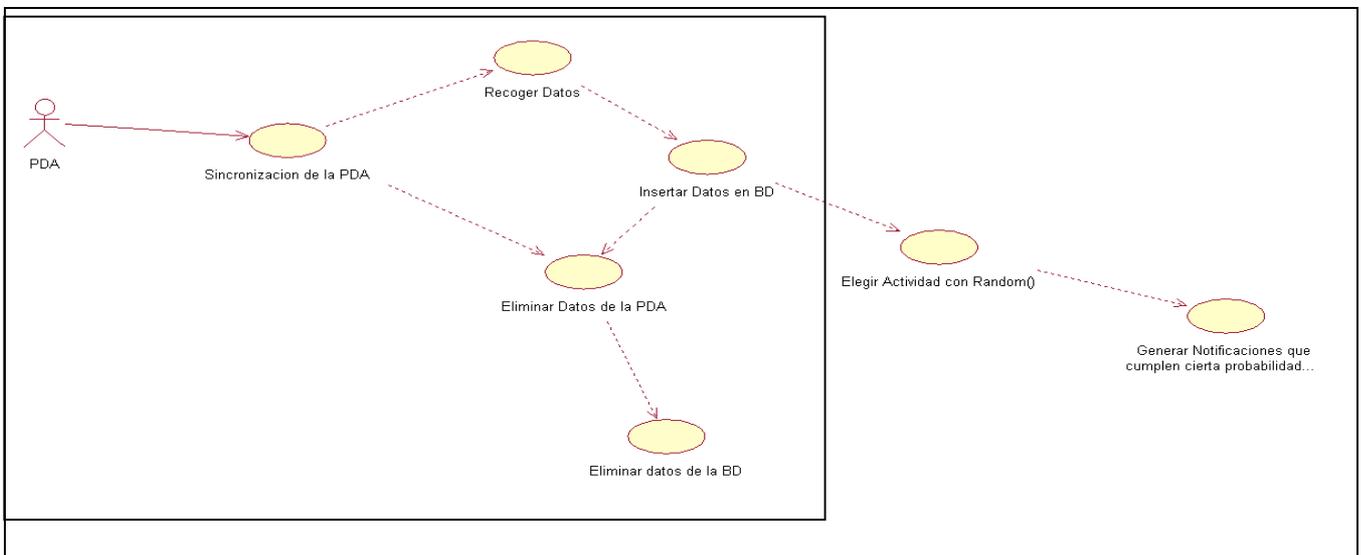
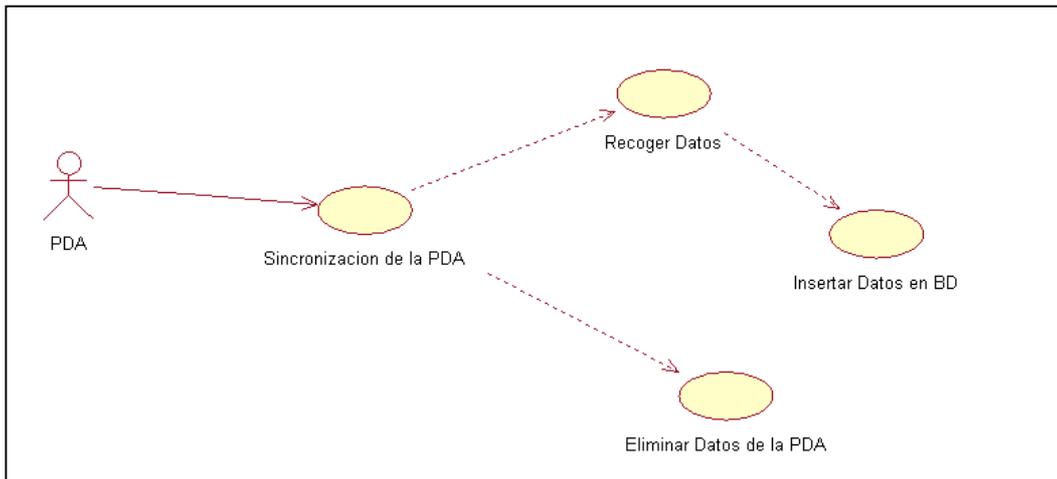
- **Flujos alternativos:**

Condiciones que pueden ocurrir que provoquen que el sistema no funcione correctamente:

- 1 Que no se haya producido una sincronización eficiente.
- 2 Que no se hayan insertado las notificaciones correctamente en la BD
3. No obtener el flujo de trabajo del operario
- 4 Perder la información necesaria para la realización del estudio final.

2.1.2. Caso de Uso recogida información de PDA





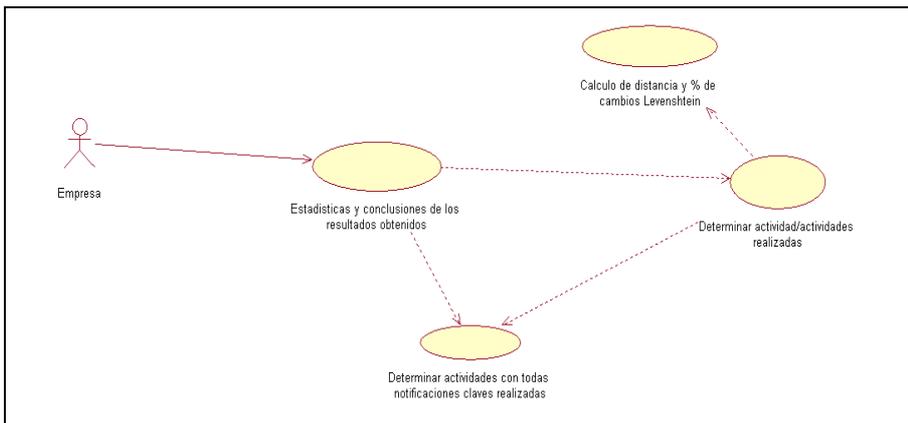
• **Actores:**

El actor del caso de uso es la PDA que para realizar un trabajo dado, recibe notificaciones a través de una conexión por medio de infrarrojos, con lo cual simulando el funcionamiento de la PDA se van insertando datos en la BD a través de un Random y así simulamos el recibiendo de notificaciones. Se simula la

situación en la que en el dispositivo se almacenan las notificaciones que se han ido realizando por un operario, posteriormente este flujo de trabajo se almacena en la BD y ya se dispone de información para tras realizar la sincronización de la PDA en la empresa, poder realizar el estudio.

- **Descripción:**
Usando el método Random () lo que conseguimos es obtener notificaciones con una cierta probabilidad de haber sido recibidas correctamente y se van insertando en la BD junto con el tiempo para la realización de cada una de las notificaciones y así se va construyendo el flujo de trabajo del operador.
- **Precondiciones:**
Poner en marcha la PDA en la ubicación donde va a realizar el operario su trabajo. Que se produzca correctamente la sincronización de la PDA
- **Poscondiciones:**
Según la opción que seleccione el usuario podrá:
 3. Ver Notificaciones necesarias
 4. Que los datos hayan sido recibidos correctamente y que se pueda obtener de la PDA, el flujo de trabajo realizado por el operador.
 5. Simulando el funcionamiento del dispositivo , que se hayan insertado las notificaciones con sus tiempo respectivos correctamente para que se pueda obtener una información cierta del trabajo del operario.
- **Flujos alternativos:**
Condiciones que pueden ocurrir que provoquen que el sistema no funcione correctamente:
 - 1 Que no se haya producido una sincronización eficiente.
 - 2 Que no se hayan insertado las notificaciones correctamente en la BD
 - 3 Que no se haya realizado correctamente la generación de tiempos y notificaciones a partir del Random ()

2.1.3. Caso de Uso para la Empresa



- **Actores:**

El actor del caso de uso es la empresa que es la encargada de estudiar el flujo de trabajo realizado por el operario.

Y de decidir por medio de varios métodos de comparación, a cual de las actividades impuestas por la organización se acerca el flujo de trabajo realizado por el operario.

- **Descripción:**

La empresa recoge la información insertada en la BD, a través del programa de la PDA. Se le presentan las notificaciones recibidas por el operario junto con el tiempo que le ha dedicado a cada una de ellas.

Lo primero que hace es comparar a través del algoritmo de Levenshtein con cual de las actividades impuestas por la organización se parece mas este flujo de trabajo, calculando la distancia y % de cambios.

Busca notificaciones claves, para poder obtener un criterio de realización del conjunto de notificaciones claves impuestas para cada actividad y determinar que porcentaje de ellas se han dado en el flujo de trabajo del operario. Es decir que el objetivo de la empresa es poder encontrar unas similitudes entre el flujo del trabajo del operario y las actividades que verdaderamente se han impuesto para realizar por los trabajadores.

- **Precondiciones:**

La eficiente inserción de los datos en la BD (la correcta sincronización de la PDA)

Tener bien definidas las actividades impuestas y sus tiempos medios de realización

Tener bien definidas todas las notificaciones clave para toda actividad, para poder después realizar el estudio debido .

- **Poscondiciones:**

Tener la certeza de decidir a través de la información obtenida de la PDA y los estudios realizados que actividades se han realizado realmente por el operador (a

partir de su flujo de trabajo).

- **Flujos alternativos:**

No obtener correctamente de la BD el flujo de trabajo realizado por el operador.

No poder realizar el estudio.

No poder estudiar la distancia entre las actividades y no poder calcular el porcentaje de cambios presente.

No obtener ninguna notificación clave para cierta actividad

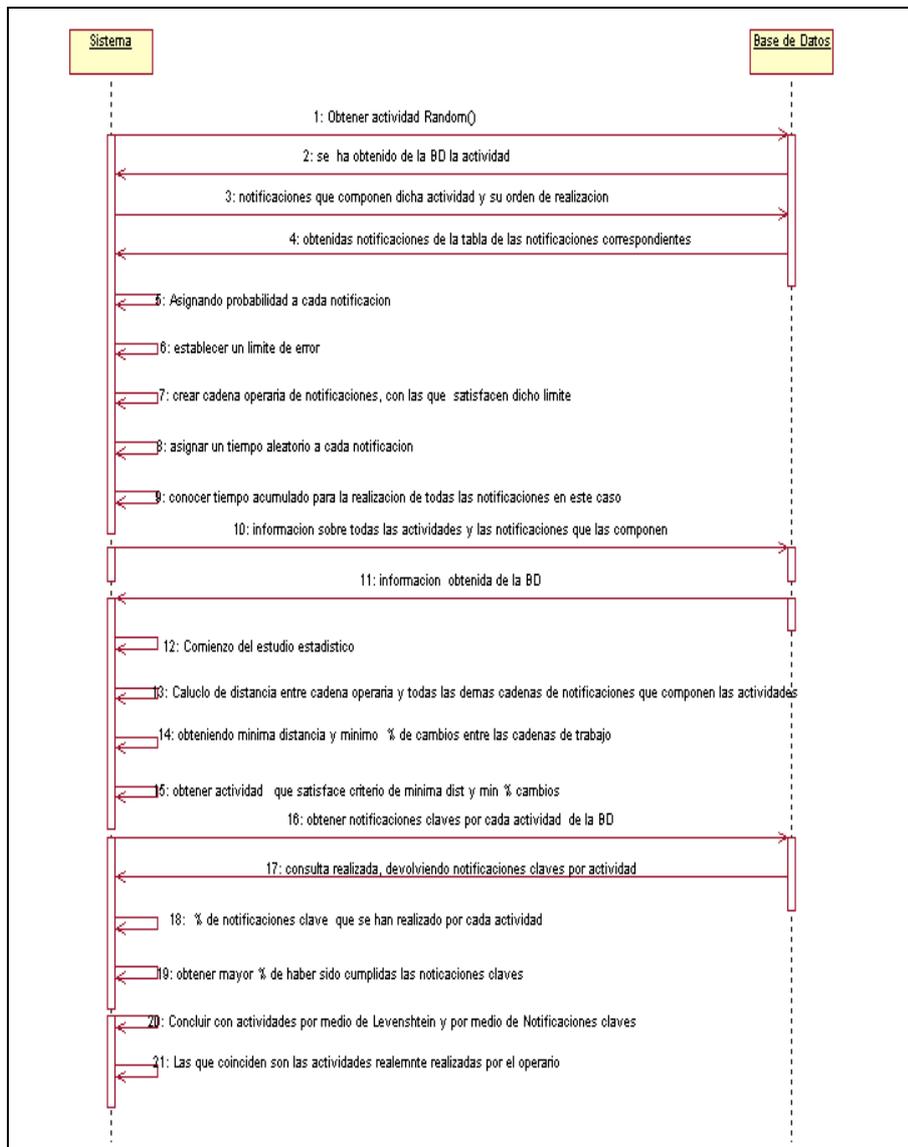
CAPÍTULO 3: DISEÑO

3.1. DIAGRAMA DE CLASES Y DE SECUENCIA

3.1.1 Diagrama De Secuencia[C]

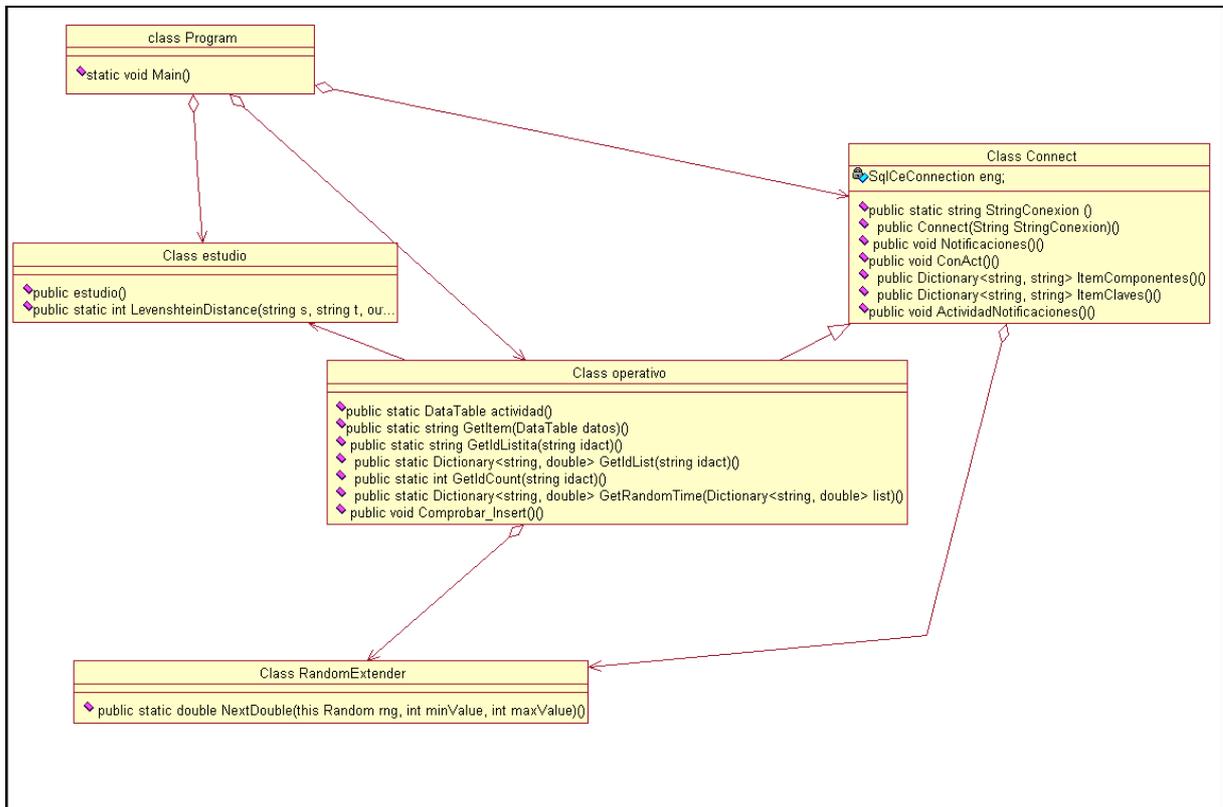
Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar.

Un diagrama de secuencia representa una forma de indicar el período durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento.



3.1.2 Diagrama De Clase[C]

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.



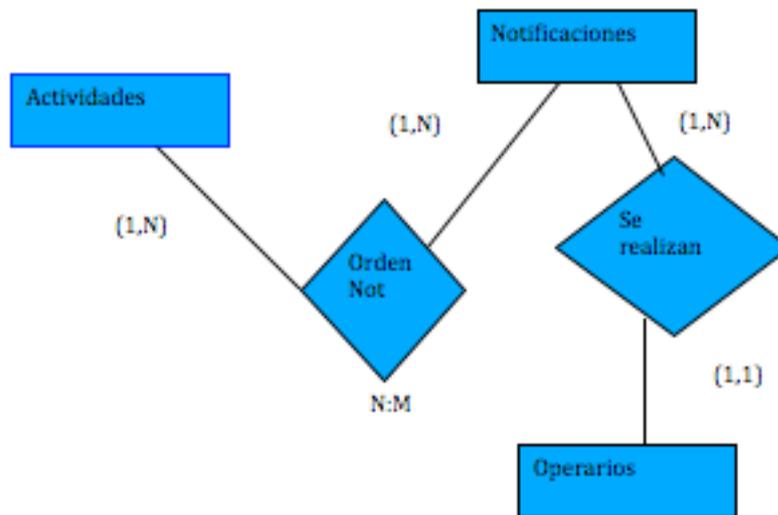
3.2 DIAGRAMA ENTIDAD-RELACIÓN

Información de la Base de Datos[11]

Para la realización de la herramienta se ha utilizado una base de datos que almacena toda la información referente a las actividades, notificaciones, notificaciones claves, un orden de realización de estas y como información final el flujo de trabajo compuesto por notificaciones y tiempos para un operario dado. Para construirla, se ha diseñado previamente el diagrama Entidad-Relación de la misma, como podemos ver en la figura

3.2.1. Modelo entidad /relación de la BD

3.2.1.1. Esquema Entidad/Relación



3.2.1.2 Analisis de entidades y atributos

Actividades	Notificaciones	Operarios
IDACT	ID	notificación
DESC	DESC	Tiempo

Actividades:

Con esta entidad lo que se pretende representar son las actividades que hay impuestos por la empresa y una descripción que define en que consisten cada una de ellas.

Atributos:

- IDACT - código para poder identificar las actividades

- DESC -una breve descripción de cada actividad.

La clave primaria de esta entidad es IDACT

Notificaciones:

Con esta entidad lo que se pretende representar son las notificaciones que pueden existir en el ámbito de la empresa y una breve descripción de cada una de ellas.

Atributos:

- ID - código para poder identificar las notificaciones
- DESC - una breve descripción de cada notificación

La clave primaria de esta entidad es ID

Operarios:

Con esta entidad lo que se pretende representar es el flujo de trabajo del operario. Lo que interesa conocer es las notificaciones que ha realizado y el tiempo invertido en cada una de ellas.

Atributos:

- notificación - código para poder identificar las notificaciones
- tiempo -un numero que representa el tiempo

Clave primaria formada por notificación y tiempo.

3.2.1.3. Análisis de las Relaciones

Actividades-notificaciones(N:M)

Una actividad esta compuesta por una o más notificaciones. Así como una notificación la pueden contener una o más actividades.

Con lo cual obtenemos la relación OrdenNot en la que dispondremos de la siguiente información:

Atributos:

- ID - clave primaria de la entidad Notificaciones, para conocer la Notificación., FK

- IDACT – clave primaria de la entidad Actividades, para conocer la Actividad, FK
- orden – el orden impuesto para la realización de las notificaciones en una Actividad.
- clave - tener conocimiento de si una cada una de las notificaciones que componen las actividad es tiene la prioridad de ser considerada notificación clave para esta actividad o no.

3.2.1.4Paso a Tablas

Actividades:

<u>IDACT</u>	DESC
PK	

Notificaciones:

<u>ID</u>	DESC
PK	

OrdenNot: ID es una FK de Notificaciones, IDACT es una FK de Actividades

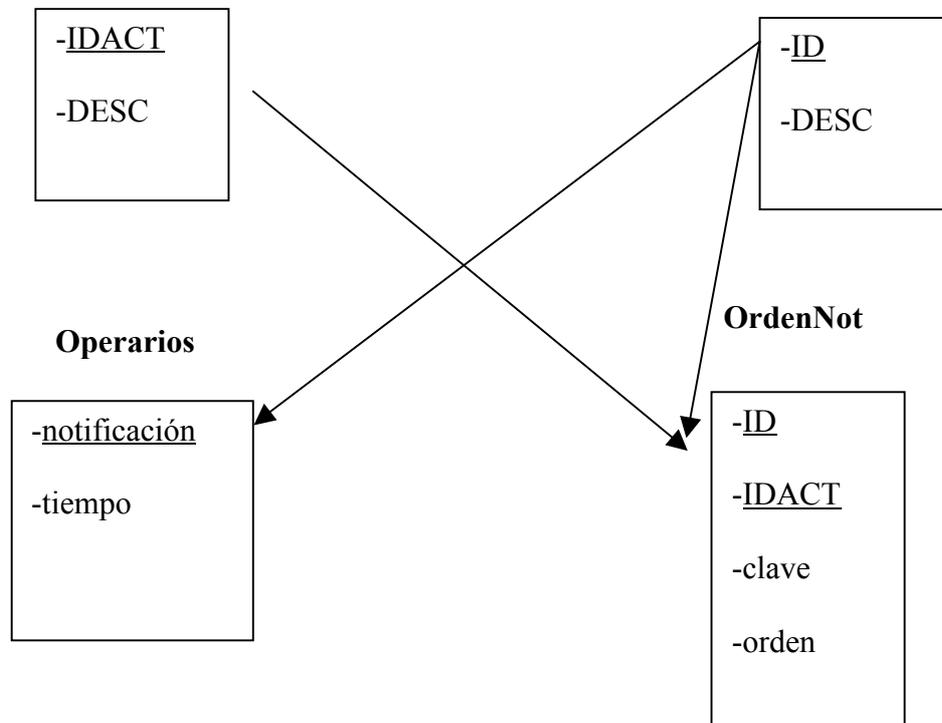
<u>ID</u>	<u>IDACT</u>	orden	Clave
PK	PK		

Operarios: notificación es una FK de Notificaciones

<u>Notificación</u>	<u>Tiempo</u>
PK	PK

ACTIVIDADES

NOTIFICACIONES



CAPÍTULO 4: IMPLEMENTACIÓN

4.1 Visual C# y SQL Server Compact 3.5

Tal y como se ha comentado en el apartado anterior sobre los requisitos, para la implementación de la herramienta, se ha utilizado el entorno de programación Microsoft Visual Studio, concretamente la versión Microsoft Visual Studio 2008 Express Edition, edición gratuita. [12],[7]

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows que soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET.

En cuanto al lenguaje de programación, se ha elegido Visual Basic C#, orientado a objetos y considerado como una evolución de Visual Basic. El principal motivo de esta decisión ha sido por lo completa que es ésta herramienta y porque se ha querido trabajar para aprender algo sobre ésta herramienta que tanto se pide en el mercado de trabajo hoy en día.

4.2 Organización del código de la Aplicación

4.2.1. Clases

Class Program

```
class Program
```

Es la clase en la que se encuentra el Main, es decir que es la clase en la que se invocan todos los métodos, pero también allí se realizan algunos tratamientos, que están o no referidos a alguno de los métodos de las demás clases.

```
static void Main(string[] args)
```

Los pasos siguientes son los que están efectuados por medio del Main

- Se comprueba la correcta conexión a la BD
- Obtenemos todas las actividades existentes y su descripción.
- Obtenemos todas las notificaciones que puedan existir y sus descripciones

-Se obtienen todas las actividades y las notificaciones que las componen según el orden de haber sido realizadas.

-La actividad seleccionada con el Random() [5]

-Las notificaciones que requiere dicha actividad

-Asignamos un numero aleatorio a cada notificación de las que se han obtenido anteriormente por la actividad seleccionada, que a partir de ahora serán las notificaciones con las que se trabajara para crear el flujo de trabajo verdaderamente realizado.

-Marcamos un limite que será el fijado para considerar el correcto funcionamiento del dispositivo PDA, las notificaciones cuyo numero aleatorio asignado sea mayor o igual que el limite puesto, serán las notificaciones que simularan el hecho de haber sido recibidas correctamente y marcaran el flujo de trabajo verdadero del operario. Las demás que tengan un numero inferior al limite, se consideraran como notificaciones con las que se ha obtenido algún problema a la hora de haber sido recibidos y con lo cual , el operario no podrá elaborarlas.

-Asignamos un numero aleatorio, que será el tiempo dedicado para la realización de una notificaciones a todas las notificaciones correspondientes a la actividad obtenida por el, teniendo en cuenta que para las que se ha obtenido un numero aleatorio de probabilidad mayor que el limite, tendrán un valor mas grande, que las que no .

-Se acumula el tiempo dedicado a todas las notificaciones, para poder así obtener cuanto se ha tardado en realizar desde la ultima hasta la primera notificación, es decir poder obtener el tiempo que ha dedicado el operario en este trabajo.

-Se realiza la inserción de las notificaciones acompañadas por el tiempo para cada una de ellas en la BD, para poder así tener almacenado el flujo de trabajo del operario.

-Se crea la cadena operaria, para poder manejar con mas facilidad las notificaciones que ha realizado el operario.

-Se crean cadenas para todas las notificaciones por las que están compuestas las actividades(de la información almacenada en la BD)

-Aplicando el algoritmo de Calculo de distancia de Levenshtein [4],[14] estamos realizando la primera parte del estudio, es decir obtenemos la distancia y el porcentaje de cambios entre la cadena operaria y todas las demás cadenas(formadas por las notificaciones que deben de componer las actividades)

-Se crea una estructura en la que se almacena cada cadena para cada actividad y los valores obtenidos de distancia y de porcentaje de cambios respectivos con la cadena operaria.

```
struct tipoRegistro
{
    public string cadena_operativa;
    public int distancia_cadenas;
    public double porcentaje_cambios;
}
```

- Con los valores obtenidos se calculan los valores mínimos de distancia y de porcentaje de cambios así sabremos a que actividad se parece el trabajo realizado por el operario. Por lo contrario también podremos saber a que actividad se parece lo mínimo.

-En la segunda parte del estudio hay que considerar las notificaciones marcadas como claves, es decir de gran importancia para poder considerar como completadas todas las actividades.

- Se forman cadenas con las notificaciones claves de cada actividad.

- Por medio de una función que determina que % de notificaciones clave se han realizado para cada tarea, obtenemos comparando la cadena operaria con las notificaciones claves de cada actividad, el número de apariciones de notificaciones claves para cada cadena y el porcentaje de realización.

-De esta forma obtenemos la actividad o actividades para la cuales se han completado, comparando con la cadena operaria el mayor numero de notificaciones claves y podemos tener la certeza de que el trabajo que ha realizado el operario se corresponde con esta actividad/actividades.

En esta estructura lo que se almacena es el porcentaje de las notificaciones claves realizadas por cada actividad dada.

```
struct ANP
{
    public string Act;
    public string NotiClave;
    public decimal Porcenataje_Real;
}
```

-Teniendo en cuenta los dos estudios realizados según los criterios mencionados, las actividades que hemos obtenido por medio del criterio del cumplimiento de las notificaciones claves, se consultan los valores obtenidos para estas actividades por medio del algoritmo de Levenshtein, se buscan los menores y así se obtiene la actividad que realmente ha realizado el operario en su flujo de trabajo.

Class Random Extender

```
public static class RandomExtender
```

Métodos:

```
public static double NextDouble(this Random rng, int minValue, int maxValue)
```

Se necesita la clase Next Double para poder obtener valores aleatorios con decimales entre un rango establecido.

Class Connect

```
class Connect
```

Esta es la clase en la que se realiza la conexión a la Base de Datos y también, se obtiene información de las actividades que existen, de las notificaciones que las componen, del orden de realización que se cumple para las notificaciones y por ultimo la presencia de notificaciones claves o no en todas las actividades

Cadena de Conexión a la BD: [8] [11] [12]

```
public static string StringConexion = ("DataSource=C:\\Documents and Settings\\MGeorgieva\\Misdocumentos\\VisualStudio2008\\Projects\\proyecto\\proyecto\\base1.sdf;Persist Security Info=False;");
```

Métodos:

```
public void Notificaciones()
```

Se realiza una consulta a la BD para obtener todas las notificaciones que hay.

```
public void ConAct()
```

Consultando a la BD, se obtienen todas las actividades existentes que estan almacenadas en la BD. [6]

```
public Dictionary<string, string> ItemComponentes()
```

Se obtienen las notificaciones que componen cada actividad y se forman cadenas de caracteres (notificaciones) con las que mas adelantes se podrá trabajar.

```
public Dictionary<string, string> ItemClaves()
```

Este método es utilizado para poder obtener las notificaciones claves de cada actividad, entendiendo por clave las notificaciones imprescindibles de realizar para poder tratar a una actividad como realizada.

```
public void ActividadNotificaciones()
```

Obtenemos información sobre cada actividad y las notificaciones por las que están compuestas, teniendo en cuenta el orden de realización de estas.

Así se podrá saber cual es el flujo de trabajo necesario para que una actividad se considere como realizada totalmente.

A la hora de realizar el estudio se necesita esta información y por esta razón se obtiene por medio de una consulta a la BD.

Class operativo

```
class operativo : Connect
```

Esta clase hereda de la Clase Connect, ya que en esta se realiza la conexión a la Base de Datos de la cual la Clase Operativo también hace uso.

Lo que se consigue con esta clase es la detección de una actividad de las almacenadas en la BD, utilizando Random (), de allí se obtienen también las notificaciones que la componen y se por medio de unas funciones, se le asignan números aleatorios a cada una de ellas, que representarían el tiempo que emplea el operario en realizar cada una de ellas.

Por ultimo lo que también se obtiene con esta clase, es la comprobación de la correcta inserción de los datos que componen el flujo de trabajo realizado por el operario.

Métodos:

```
public static DataTable actividad()
```

Selecciona todas las actividades que hay almacenadas en la BD y las guarda en un DataTable para poder trabajar con ellas en los siguientes métodos.
Ya que para poder seleccionar una de las actividades por medio de Random () tenemos que conocer todas las actividades que están almacenadas en la BD.

```
public static string GetItem(DataTable datos)
```

Por medio de Random () [5] se elige una de las actividades, de las que disponemos. De este modo lo que verdaderamente se hace, es que se empieza a simular la sincronización del dispositivo PDA. Se escoge una actividad solo para poder obtener mas tarde las notificaciones que la componen, para poder formar la cadena operativa del trabajador. Es decir para poder simular de estas notificaciones el flujo de trabajo que se ha realizado verdaderamente.

Para conseguir valores mas variados se pueden utilizar estas semillas: DateTime.Now.Ticks o DateTime.Now.Millisecond

```
public static string GetIdListita(string idact)
```

Con este método lo que se consigue son las notificaciones por las que esta compuesta, la actividad que ha sido seleccionado por el Random().

```
public static Dictionary<string, double> GetIdList(string idact)
```

[13]

Se construye un diccionario con las notificaciones de la actividad elegida y un numero aleatorio comprendido entre 0-1 , con el cual se va a simular el correcto funcionamiento del dispositivo PDA, es decir que se marcara un numero como limite y de allí podremos ir obteniendo una serie de notificaciones realmente hechas, consideradas de haber sido recibidas correctamente.

```
public static int GetIdCount(string idact)
```

Con este método se consigue el número de notificaciones que componen la actividad elegida. Puede ser de utilidad, a la hora de querer saber de que longitud es el conjunto de notificaciones que componen una actividad dada.

```
public static Dictionary<string, double> GetRandomTime(Dictionary<string, double> list)
```

Con este método lo que se realiza es la asignación a cada notificación realizada por el operario, sin tener en cuenta el error de haber sido recibidas correctamente impuesto, es decir a todas las notificaciones que obtenemos tras saber cual es la actividad resultante de la

utilización del Random().[5]

Entonces aquí se le asigna un tiempo a cada notificación, que será el tiempo dedicado para la realización. Si que se impone un criterio relacionado con el correcto recibimiento, simulando el funcionamiento del dispositivo PDA y es que:

```
if (item.Value > (double)0.5)
    valrandom = random.NextDouble(5, 10);
else
    valrandom = random.NextDouble(0, 5);
```

```
public void Comprobar_Insert ()
```

Método por medio del cual consultamos si hemos insertado correctamente las notificaciones que ha realizado el operario y el tiempo que ha dedicado a cada una de ellas.

Class estudio

Es la clase en la que comienza la parte del estudio de la aplicación. Se describe el algoritmo de cálculo de distancia y % de cambios entre dos cadenas, llamado Algoritmo de Levenshtein.

Métodos:

```
public static int LevenshteinDistance(string s, string t, out double porcentaje)
```

Se le considera Distancia de Levenshtein a la distancia de edición, o distancia entre palabras, al número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra.

Se entiende por operación, bien una inserción, eliminación o la sustitución de un carácter. Esta distancia recibe ese nombre en honor al científico ruso Vladimir Levenshtein, quien se ocupara de esta distancia en 1965. Es útil en programas que determinan en cuánto son similares dos cadenas de caracteres, como es el caso de los correctores de ortografía.

En esta aplicación, es muy útil este Algoritmo, ya que se necesita para poder estudiar en cuanto son diferentes la cadena de trabajo realizado por el operario y las cadenas del flujo de trabajo compuestas por las notificaciones que deberían de realizar para completar todas las actividades existentes.

Ya que de esta manera, se podrá tener certeza por medio del flujo de trabajo del operario, el echo de saber que actividades ha realizado completamente o parcialmente, o puedo darse el caso en que no haya completado ninguna actividad.

Pero todas estas conclusiones que se obtienen tras el estudio, se explicaran mas adelante.

Estructuras creadas para el acceso correcto a los datos necesitados

```
struct tipoRegistro
{
    public string cadena_operativa;
    public int distancia_cadenas;
    public double porcentaje_cambios;
}
```

```
struct tipoActi
{
    public string cadena_op;
    public string act_corresp;
}
```

```
struct ANP
{
    public string Act;
    public string NotiClave;
    public decimal Porcenataje_Real;
```

CAPÍTULO 5: TEORIAS PARA EL CÁLCULO DE DISTANCIAS ENTRE CADENAS

Distancia de Levenshtein

La "Distancia Levenshtein" o "Distancia de Edición", como también se le conoce a esta métrica, es el número de eliminaciones, inserciones o sustituciones requeridas para transformar una cadena fuente "x" en una cadena objetivo "z", toma este nombre del científico ruso Vladimir Levenshtein, quien ideó el algoritmo en 1965 [4]. Desde entonces y mayormente en la actualidad, este algoritmo ha sido utilizado en el desarrollo de varias aplicaciones, tales como:

1. Sistemas para la revisión de faltas ortográficas automatizada en textos.
2. Sistemas de reconocimiento de voz.
3. Sistemas para el análisis de ADN.
4. Sistemas para la detección de plagios.

Se le considera una generalización de la distancia de Hamming, que se usa para cadenas de la misma longitud y que solo considera como operación la sustitución. Hay otras generalizaciones de la distancia de Levenshtein, como la distancia de Damerau-Levenshtein, que consideran el intercambio de dos caracteres como una operación. Estas generalizaciones serán detalladas más adelante.

Distancia de Hamming

Se denomina Distancia de Hamming, a la efectividad de los códigos de bloque y depende de la diferencia entre una palabra de código válida y otra. Cuanto mayor sea esta diferencia, menor es la posibilidad de que un código válido se transforme en otro código válido por una serie de errores. Si dos palabras de código difieren en una distancia d , se necesitan d errores para convertir una en la otra.

La distancia de Hamming se denomina así gracias a su inventor Richard Hamming, profesor de la Universidad de Nebraska, que fue el que introdujo el término para establecer una métrica capaz de establecer un código para la detección y auto-corrección de códigos. Se emplea en la transmisión de información digitalizada para contar el número de desvíos en cadenas de igual longitud y estimar el error, por esto se denomina a veces como distancia de señal.

Distancia de Damerau-Levenshtein o distancia de edición [1]

Se denomina Distancia de edición al número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra. Se entiende por operación, bien una inserción, eliminación, sustitución o transposición de dos caracteres. Lo que la distingue de la distancia de Levenshtein es que ésta última cuenta como una sola operación de edición a cualquiera de las tres primeras, pero cuenta la transposición como dos operaciones de edición.

CAPITULO 6: RESULTADOS OBTENIDOS DE LOS ESTUDIOS

Se simula la generación de notificaciones que formarían el flujo de trabajo de un operario. Con la aplicación podremos generar notificaciones primero escogiendo una actividad de las dadas, después obtener las notificaciones por las que esta compuesta y asignarles un numero aleatorio que va a representar la probabilidad que tienen de ser recibidas correctamente, es decir que el operario podrá ejercerlas ya que las ha obtenido correctamente.

De esta manera lo que podemos conseguir es una gran variación de conjuntos de notificaciones, para los que vamos a analizar como influye a la hora de decidir que actividad ha realizado el operario.

Hay que realizar un análisis sobre los resultados obtenidos teniendo en cuenta los siguientes criterios, para poder determinar la actividad o actividades realizadas según el conjunto de notificaciones que determinan el flujo de trabajo del operario.

- Longitud del conjunto de notificaciones realizadas por el operario
- Error consentido para la simulación del correcto funcionamiento del dispositivo por el cual se recibiría el conjunto de notificaciones.
- Notificaciones consideradas como claves, es decir de gran importancia para la correcta realización de una actividad.
- La variación de los resultados obtenidos según la longitud del conjunto de las notificaciones claves.
- Resultados obtenidos según el Algoritmo de Levenshtein, calculando la distancia y el % de cambios que hay que realizar entre dos conjuntos de notificaciones para que sena idénticos.
- El flujo de trabajo del operario, dependerá mucho también del numero aleatorio asignado a cada notificación, ya que al estar usando Random para la generación de los números aleatorios que son la probabilidad que tienen cada una de las notificaciones de haber sido recibidas correctamente.
- Entonces según el error de funcionamiento que consideremos es muy probable que con un mismo conjunto de notificaciones de partida, al final se estén realizando distintas actividades, por esta razón.

Cuanto mayor sea el límite impuesto para el correcto recibimiento de los datos se obtienen los siguientes resultados:

- menor % de cumplimiento de notificaciones claves para todas las actividades

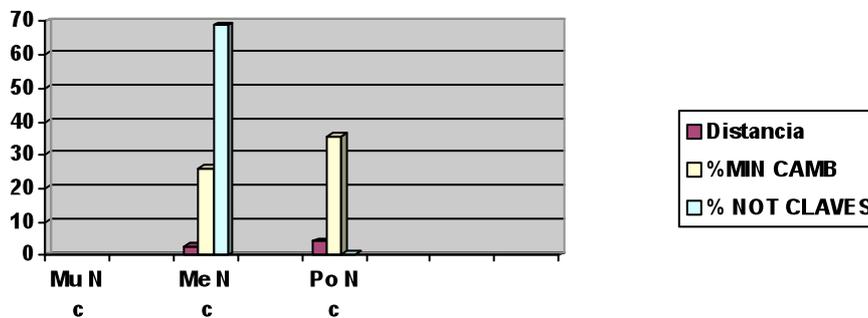
- mayor distancia y mayor % de cambios entre la cadena operaria de notificaciones y las que realmente componen las actividades.

- menos numero de actividades obtenidas como resultado de haber sido realizadas por el operario con su flujo de trabajo.

****Estudios Realizados Según el error producido en el funcionamiento de la pda.***

Actividad de longitud Corta Comparada con Actividades de longitud Corta

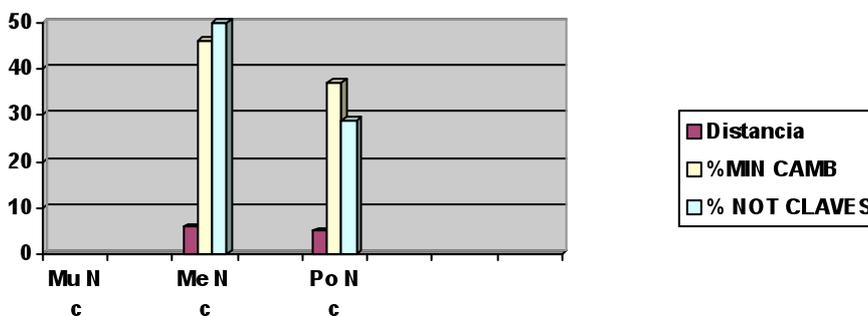
Error <=0.10



Con un error menor que 0.10 partiendo de actividades cortas con una cantidad media o pequeña de notificaciones consideradas como claves, cada vez mas se dejan de detectar las actividades cortas con cantidad escasa de notificaciones y se consideran completadas las actividades cortas con una cantidad media de notificaciones aunque no todas al completo, es decir que de la mayoría se realiza solo una parte de la actividad.

Y respecto a las actividades cortas cuya cantidad de notificaciones claves es considerada como media, con este error se da la situación de no detectar completamente ni la actividad realizada por el operario.

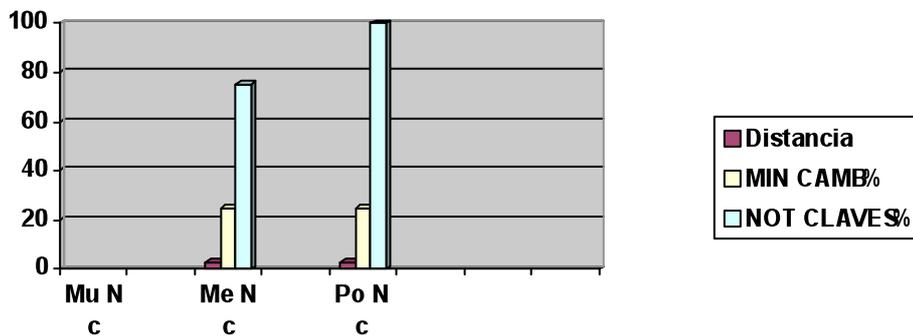
Error <= 0.05



Teniendo un error menor que 0.05 lo que se puede comprobar es que se consiguen detectar muchas mas actividades de longitud corta que tienen una cantidad media de notificaciones claves, que las actividades corta con un numero pequeño de notificaciones claves.

Se da el caso en el que se parte de una actividad corta con un numero pequeño de notificaciones claves, pero la que es detectada aunque no completándose con totalidad es una actividad corta con un numero medio de notificaciones consideradas como claves. Esto ocurre ya que al tener un error elevado cada vez hay mas probabilidad de que las notificaciones claves consideradas de longitud pequeña no se hayan dado para la actividad corta, entonces al no estar justo estas notificaciones claves, dichas actividades no se considerarían como detectadas.

Error <=0.01



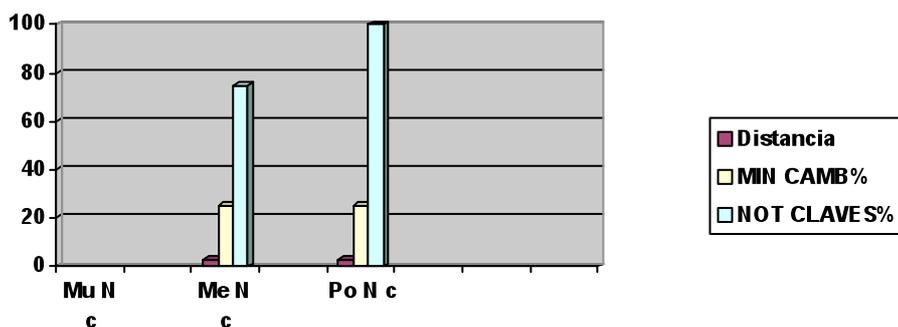
Al tener errores bajos 0.01 los resultados obtenidos son similares al caso de no tener errores. Ya que se detectan correctamente todas las actividades consideradas de longitud corta y de tener poca cantidad de notificaciones claves, consideradas como imprescindibles para la realización de dichas actividades.

Aparte se puede apreciar como se pueden detectar como realizadas también las actividades cortas cuyas notificaciones claves sean consideradas de cantidad media.

De esta manera se detectan como realizadas otras actividades además de la actividad propia que ha ido a realizar el operario.

También hay que considerar el caso en el que puede haber unos pequeños cambios, que indican que a pesar de tener errores muy bajos, es importante que dichos cambios no sean sobre las notificaciones consideradas como relevantes en concreto para las actividades cortas con cantidad media y corta de notificaciones claves. Ya que al no estar justo estas notificaciones claves, dichas actividades no se considerarían como detectadas.

Error=0



Si no hay error se detectan correctamente todas las actividades consideradas de longitud corta y de tener poca cantidad de notificaciones claves, consideradas como imprescindibles para la realización de dichas actividades. Aparte se puede apreciar como se pueden detectar como realizadas también las actividades cortas cuyas notificaciones claves sean consideradas de cantidad media.

De esta manera se detectan como realizadas otras actividades además de la actividad propia que ha ido a realizar el operario.

Si la actividad realizada por el operario es de longitud corta y tiene notificaciones claves de cantidad corta, se puede dar el caso de que la única actividad detectada sea esta, ya que puede suceder de no tener notificaciones claves en común entre actividad corta de pocas notificaciones claves y actividad corta de cantidad media de notificaciones claves

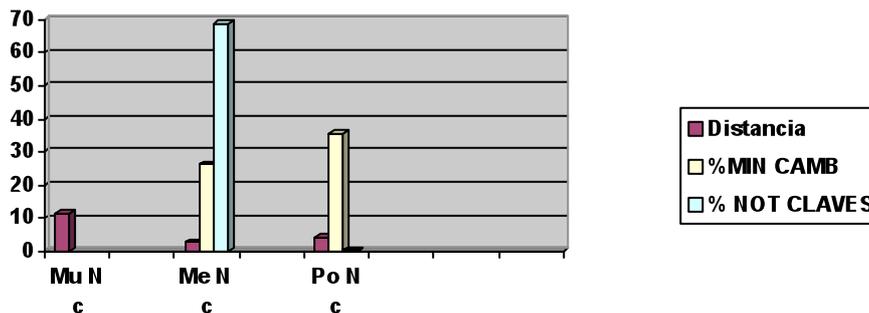
Al comparar actividades de longitud corta con actividades de longitud corta se obtienen distancias y % de cambios muy pequeños según el Algoritmo de Levenshtein, pero a medida que aumenta el error de recibimiento de las notificaciones estos valores suben.

Con errores muy pequeños se completan mas las actividades con pocas notificaciones claves, pero al aumentar el error se produce cambio y es que se consideran completadas mas las actividades consideradas de tener notificaciones claves medias que las de pocas.

Esto se debe a que al aumentar el error, se puede dar el caso de que ningunas de las notificaciones claves establecidas para las actividades consideradas de tener pocas notificaciones claves se haya dado.

Actividad corta comparada con actividades de longitud media

Error ≤ 0.10



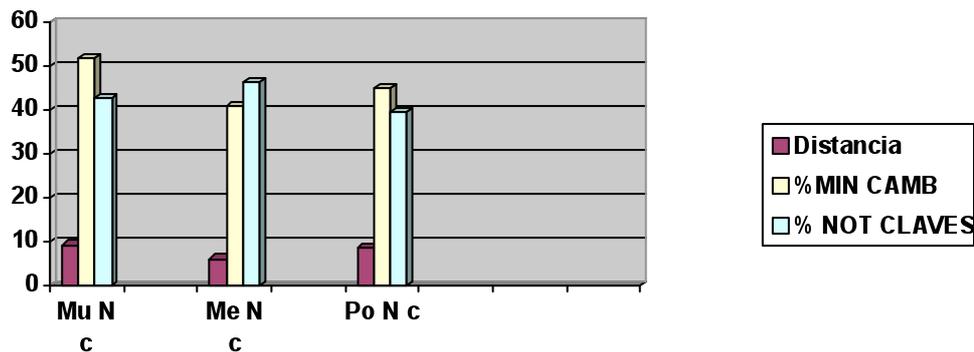
Teniendo un porcentaje menor que 0.10 considerándolo un error alto y sabiendo que la actividad realizada por el operario es de longitud corta y con una cantidad media o escasa de notificaciones claves, lo que se comprueba es que se detectan también actividades de longitud media, pero entre ellas las que mas se consideran completadas aunque no totalmente son las que tienen pocas notificaciones y muchas.

En este caso partiendo de las actividades almacenadas en la BD y sus respectivos detalles de notificaciones que las componen y notificaciones claves para considerarlas como imprescindibles, se da la situación que las actividades de longitud media con una cantidad de notificaciones claves media, no son detectados ya que las notificaciones claves para estas

actividades son muy distintas a las notificaciones que componen las actividades de longitud corta de las que se parte.

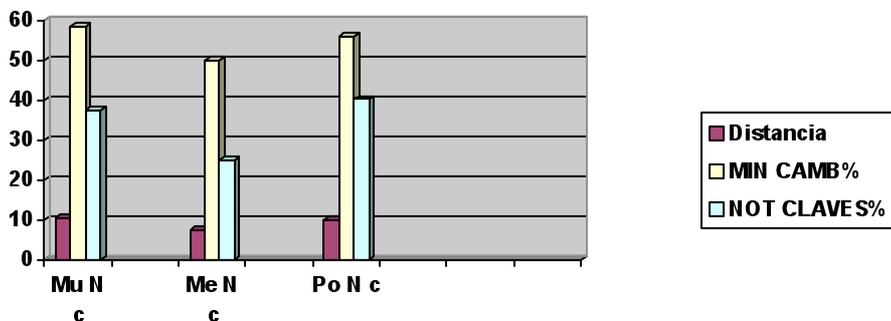
Así se comprueba que si unas actividades no tienen en común notificaciones, no hay ninguna manera de poder detectar unas partiendo de las otras.

Error ≤ 0.05



Con probabilidad de ocurrir mayor que 0.05 partiendo de una actividad cuya cantidad de notificaciones es de longitud corta, comparándola con actividades de longitud media, lo que se observa es que se detectan mayoritariamente aquellas actividades consideradas medias con un número medio de notificaciones claves, seguidas de las actividades medias con muchas notificaciones claves y por último se detectarían las actividades medias con poca cantidad de notificaciones claves. Ya que al tener un error ≥ 0.05 es muy probable que de las notificaciones claves consideradas pocas no se haya dado ninguna realizando la comparación con la actividad de longitud corta de la que se parte.

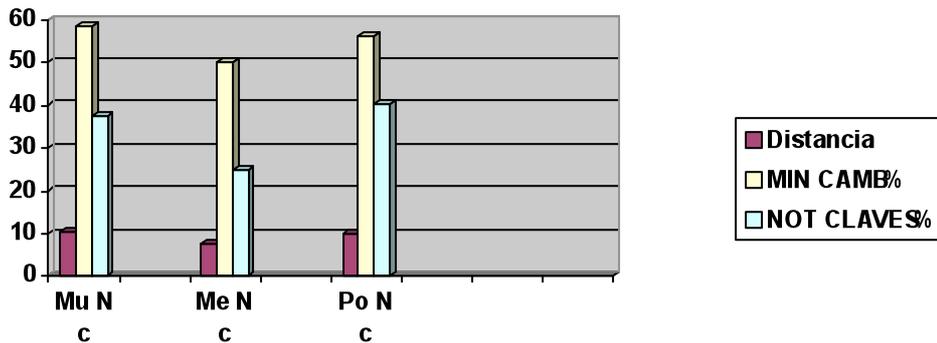
Error ≤ 0.01



Con un error cercano al 0 lo que se observa es que si se parte de actividades de longitud corta y han sido recibidas casi todas las notificaciones que las componen, entonces se detectarían aunque no al 100 % las notificaciones claves de las actividades medias con una cantidad pequeña de notificaciones claves.

Se observa también que muy se detectan muy pocas de las actividades medias cuya cantidad de notificaciones claves es de cantidad media, ya que puede que pase que estas notificaciones no esten en las notificaciones de la actividad corta de partida.

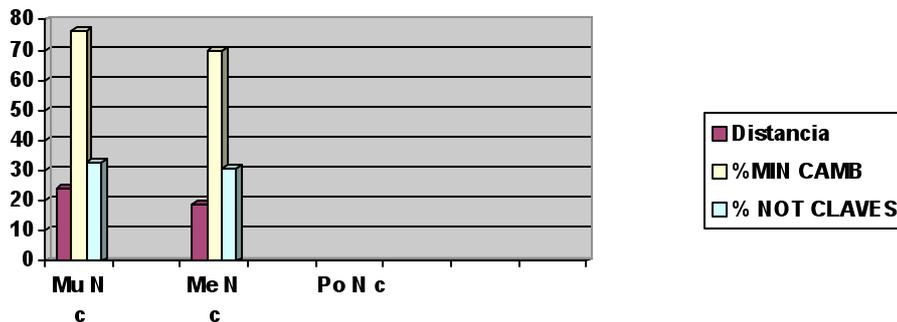
Error=0



Sin ningún error siendo la actividad de longitud corta se obtienen resultados muy similares a los obtenidos con un error ≥ 0.01 . se detectarían las mismas actividades.

Actividad corta comparada con actividades de longitud Larga

Error ≤ 0.10

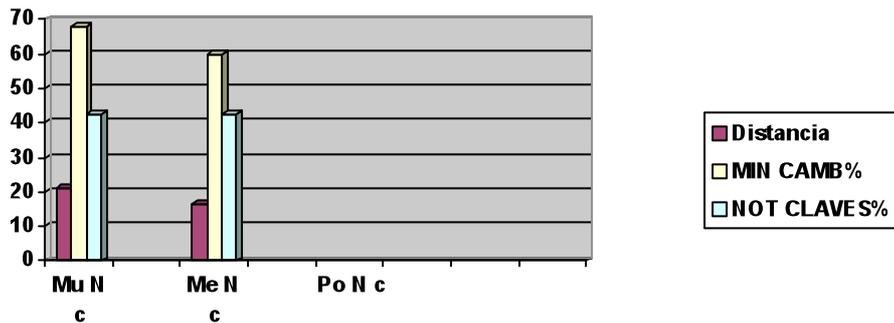


Partiendo de una actividad realizada por el operario con un error 0.10, lo que se comprueba de este grafico es que no se puede considerar como detectada completamente ninguna actividad de longitud larga.

Esto se debe a que las actividades de longitud larga están compuestas por muchas notificaciones y a la vez se consideran completadas si se ha dado un numero amplio o medio de notificaciones claves y partiendo de actividades cortas esto no puede realizarse.

Como mucho puede suceder que algunas de las notificaciones claves de las actividades largas se hayan realizado en las actividades cortas y así se obtendrá como detectada alguna parte de las actividades largas.

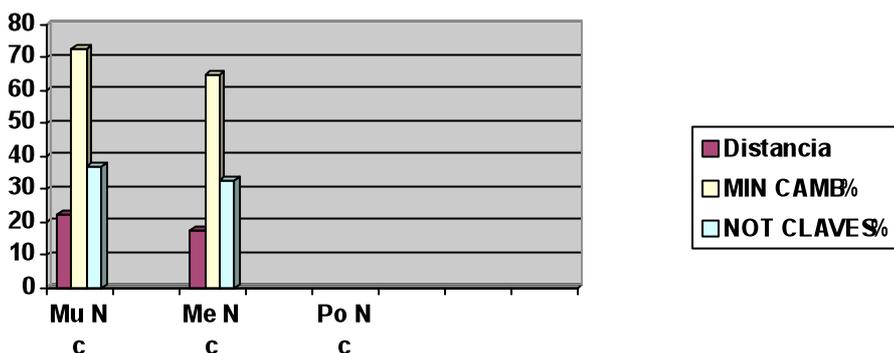
Error ≤ 0.05



Con una actividad corta , siendo el error ≤ 0.05 lo que se puede obtener son actividades de longitud larga pero no se logra completarlas ya que se ve que las que tienen una cantidad media de notificaciones claves se detectarían con más probabilidad que las de muchas notificaciones claves.

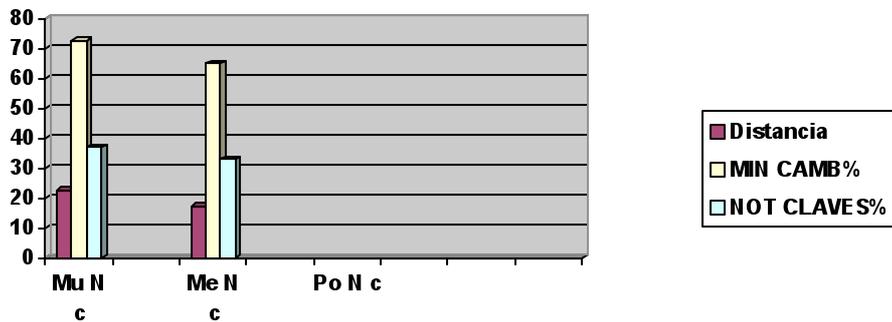
Pero en ninguno de los casos serían detectadas totalmente.

Error ≤ 0.01



Con un error cercano al 0 , lo que se observa es que con una actividad corta, no se puede completar actividad de longitud larga con cantidad media o mucha de notificaciones claves. Ya que la actividad larga dispone de más notificaciones claves, que las notificaciones que componen la actividad de partida corta.

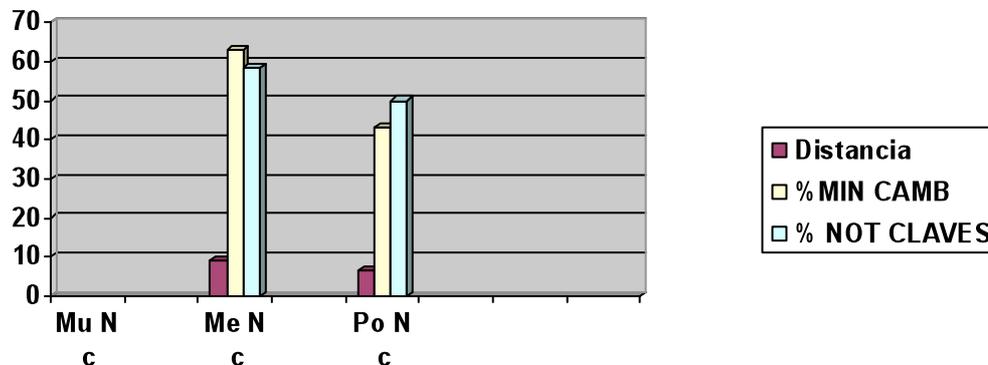
Error=0



Con un error 0 , se reciben correctamente todas las notificaciones de la actividad corta de partida y lo que se observa es que con una actividad corta, no se puede completar actividad de longitud larga con cantidad media o mucha de notificaciones claves. Ya que la actividad larga dispone de mas notificaciones claves, que las notificaciones que componen la actividad de partida corta.

Actividad Media comparada con actividades de longitud Corta

Error <=0.10



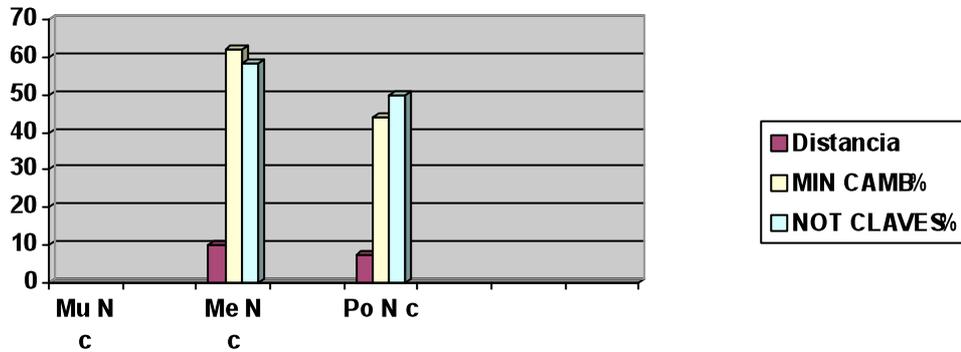
Con un error elevado como es el 0.10 partiendo de una actividad de longitud media, lo que comprobamos es que no se detectan al completo actividades cortas con cantidad media y escasa de notificaciones claves.

Se puede apreciar que se consiguen completar mas aunque no con totalidad las actividades con cantidad media de notificaciones claves, que las de cantidad escasa, pero las distancias marcan decisión contraria. Ya que hay más distancia entre las actividades cortas y las actividades medias con medias notificaciones claves, que la distancia con las actividades con notificaciones claves escasas.

Esto ocurre ya que al tener un error alto, las actividades cortas que están compuestas por cantidad escasa de notificaciones claves, tienen dichas notificaciones claves distintas a las notificaciones que componen las actividades medias dadas.

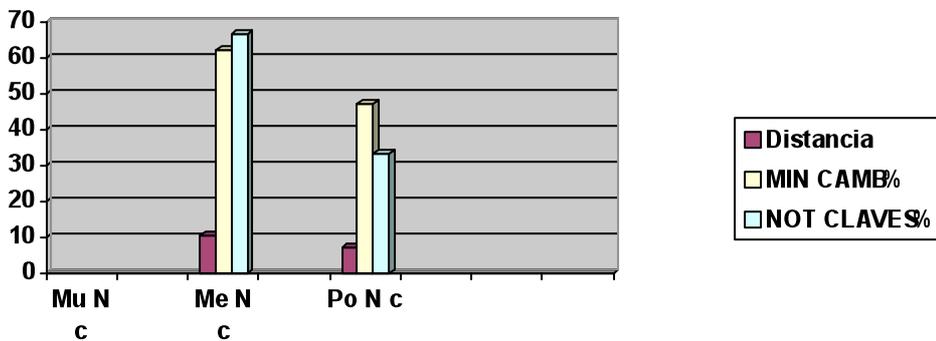
Es un rasgo muy importante a la hora de detectar alguna otra actividad aparte de la realizada por el operario, ya que por muy pequeño que sea el error, al no tener notificaciones en común no se puede completar otra actividad aparte

Error ≤ 0.05



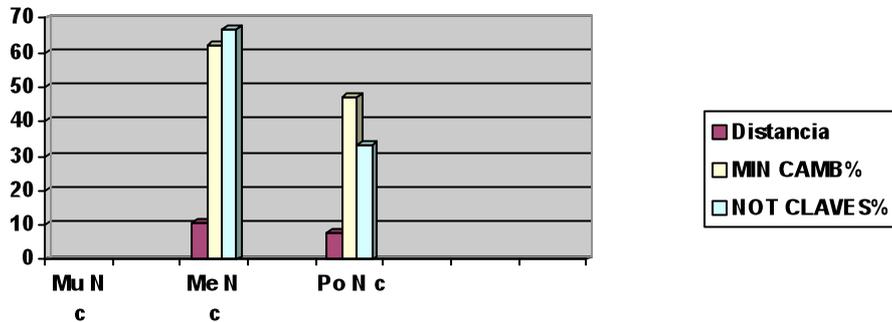
Con un error $\leq 0,05$ partiendo de una actividad de longitud media, se puede conseguir las actividades de longitud corta aunque no al completo, se puede observar que se detectan mejor las actividades cortas con una cantidad media de notificaciones claves, que las que tienen una cantidad pequeña de notificaciones claves. Aunque la distancia entre la actividad media de partida y la actividad corta con media cantidad de notificaciones claves, es mayor que la distancia con las de pocas notificaciones claves.

Error ≤ 0.01



Con un error muy cercano al 0, se observa que se han recibido correctamente casi todas las notificaciones que componen la actividad de longitud media de partida y así se consiguen completar al 70% las actividades de longitud corta con medias notificaciones claves y con un 30 % las actividades de longitud corta con cantidad media de notificaciones claves.

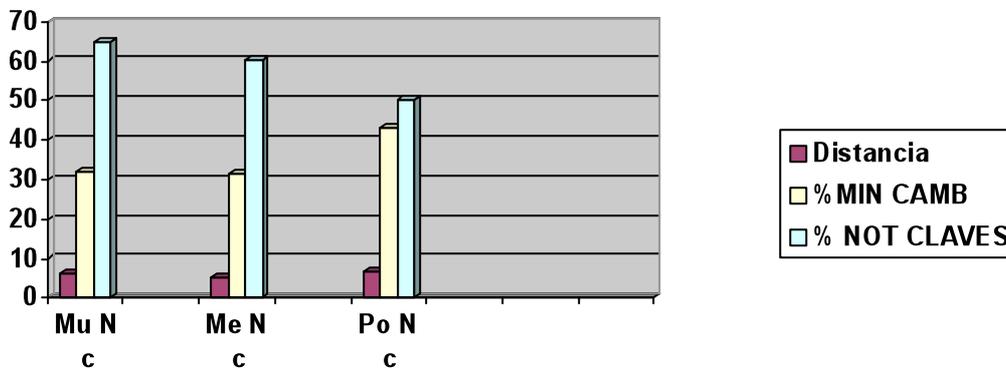
Error=0



Con un error = 0, se reciben correctamente todas las notificaciones de la actividad media de partida, pero no se consigue detectar completamente las actividades cortas con cantidad media, ni con cantidad poca de notificaciones claves.

Actividad Media comparada con actividades de longitud Media

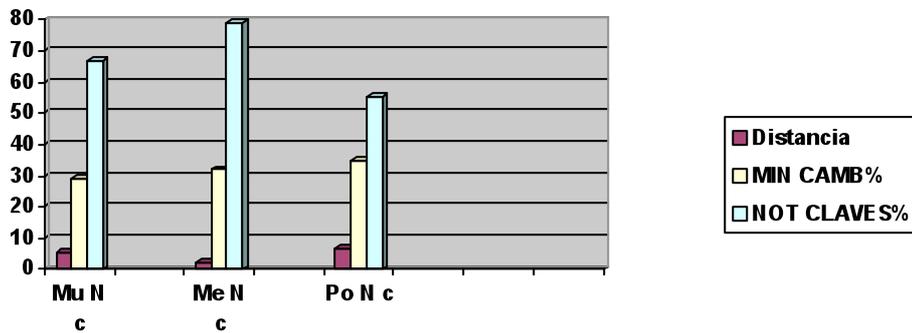
Error <=0.10



Partiendo de actividades de longitud media, con el error al 0.10 se aprecia que las actividades medias mas completadas están compuestas por una cantidad amplia o media de notificaciones claves..

Ya que al tener un porcentaje amplio, puede ser que para las actividades medias de pocas notificaciones claves, no se haya dado ninguna de la notificaciones , que tiene marcadas como claves, entonces por eso las mas completadas aunque tampoco totalmente son las de muchas y medias notificaciones claves, ya que allí al tener mas notificaciones claves , algunas de ellas se han podido dar.

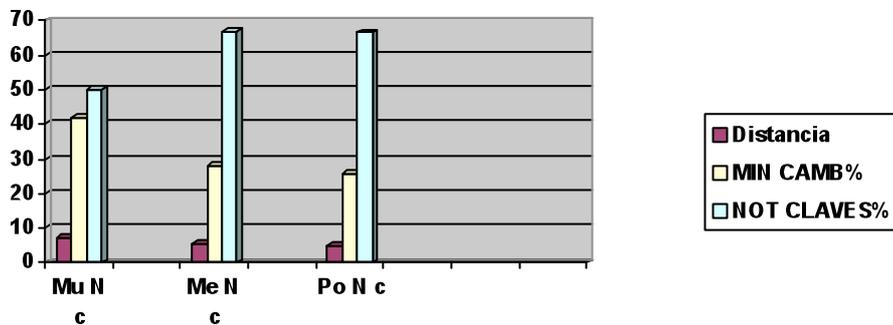
Error ≤ 0.05



Con un error ≤ 0.05 partiendo de una actividad de longitud media, se pueden detectar aunque no totalmente las actividades de longitud media cuyas cantidades de notificaciones claves son muchas, pocas o medias.

Se observa que se detectan con mas porcentaje las de una cantidad media de notificaciones claves, ya que las de pocas notificaciones claves puede ser que justo falten las notificaciones claves que las componen para considerarlas realizadas.

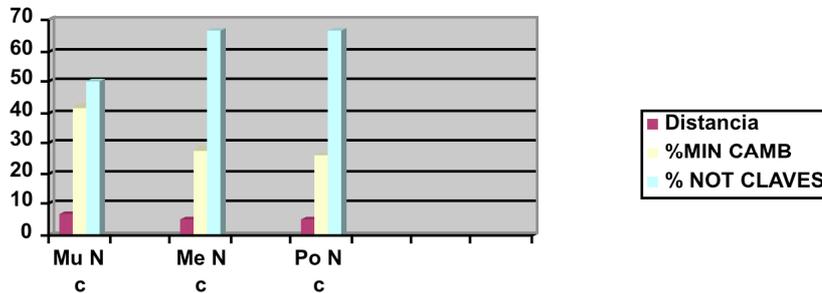
Error ≤ 0.01



Con error cercano al 0, partiendo de una actividad de longitud media, lo que observamos es que se pueden detectar las que mas las actividades de longitud media con cantidad pequeña de notificaciones claves, seguidas por las de cantidad media de notificaciones y por ultimo las de muchas notificaciones claves, aunque ninguna se consigue totalmente. Ya que si la actividad de longitud media de partida es de pocas notificaciones claves es difícil que con ella se complete la actividad de longitud media de muchas notificaciones claves.

Las distancias se ve que son muy pequeñas ya que estamos comparando actividades medias con actividades medias.

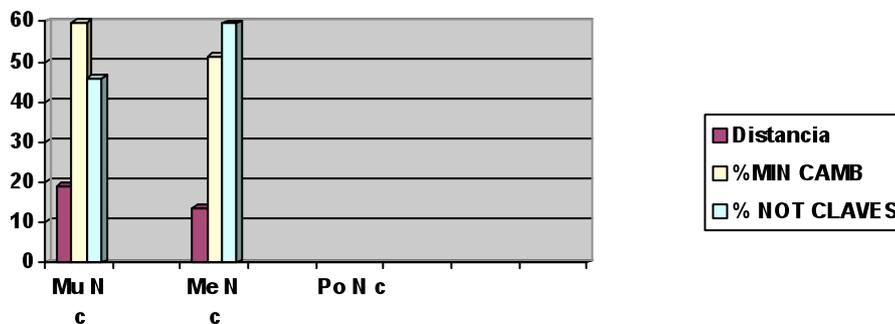
Error=0



Los resultados obtenidos son muy parecidos a los resultados con un error = 0.01, esto es normal .ya que depende tambien cuantas actividades medias se ha simulado sin han sido mas de cantidad de notificaciones claves pequeña, o media o de muchas. Por eso se observa que no se completan las actividades medias totalmente.

Actividad Media comparada con actividades de longitud Larga

Error <=0.10

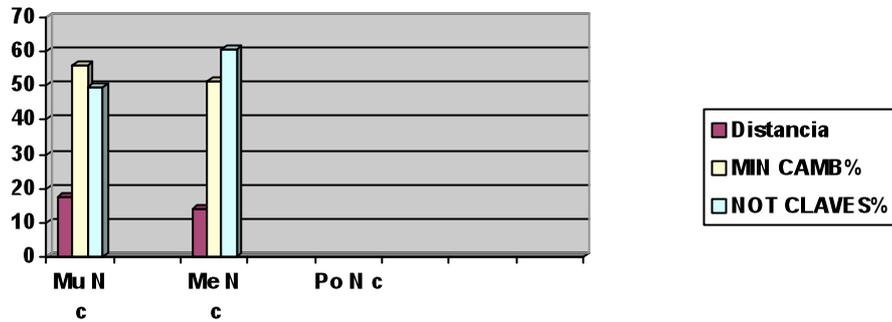


Se comprueba que partiendo de una actividad de longitud media no se han podido completar las actividades de longitud larga cuyas notificaciones claves son de cantidad media, esto es posible dependiendo de las notificaciones claves que componen estas actividades, ya que si están contenidas en la actividad de longitud media , teniendo en cuenta que el error tambien es alto. Pero se observa que es posible.

Las actividades de longitud larga con notificaciones claves de mucha cantidad no se pueden completar partiendo de actividad media. Ya que posiblemente estan compuestas por muchas notificaciones que ni siquiera componen la actividad media de partida.

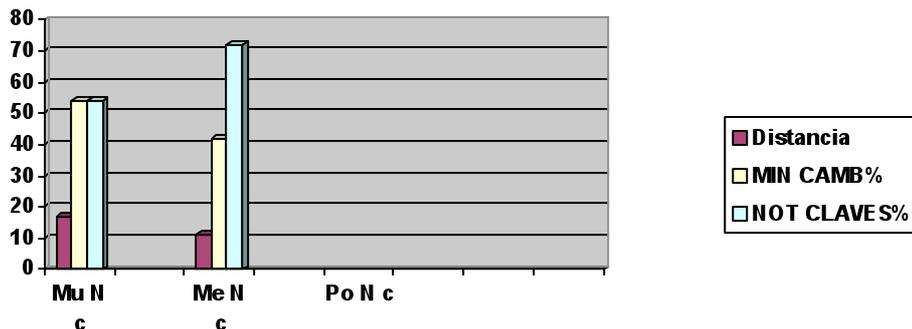
Las distancias son medianas, ya que se estan comparando cadenas de longitudes diferentes y esto es normal.

Error ≤ 0.05



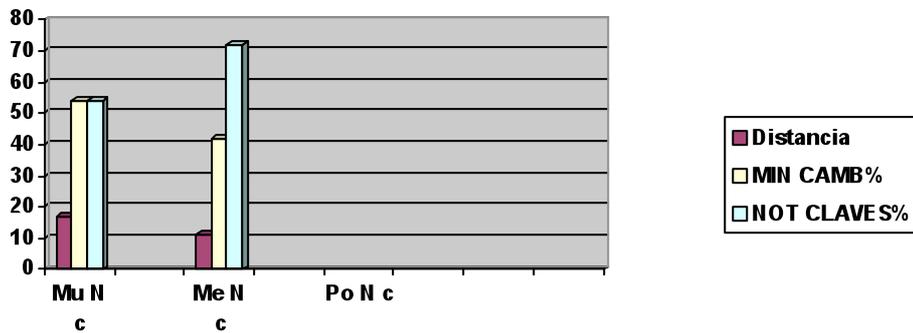
Partiendo de una actividad de longitud media, con un error bastante considerado como es el de la probabilidad de recibir correctamente las notificaciones ≤ 0.05 se comprueba que no es posible detectar con totalidad las actividades de longitud media, aunque se alcanza un 60 % de ellas, en el caso de las que estan compuestas por una cantidad media de notificaciones claves.

Error ≤ 0.01



Con un error muy cercano al 0, nos damos cuenta que partiendo de una actividad media , no podemos detectar y completar con totalidad las actividades de longitud larga, lo máximo que se puede conseguir es completar hasta un 70 % las actividades de longitud larga con media cantidad de notificaciones claves y esto es normal ya que las actividades de longitud media estan compuestas por menos actividades que las de longitud larga, con lo cual lo máximo a lo que nos podemos acercar es a las de cantidad media y pequeña de dichas actividades largas. Tambien se consigue detectar un 50 % de las actividades largas con muchas notificaciones claves, pero al no superar el 50 % estas actividades no se pueden considerar terminadas correctamente.

Error=0

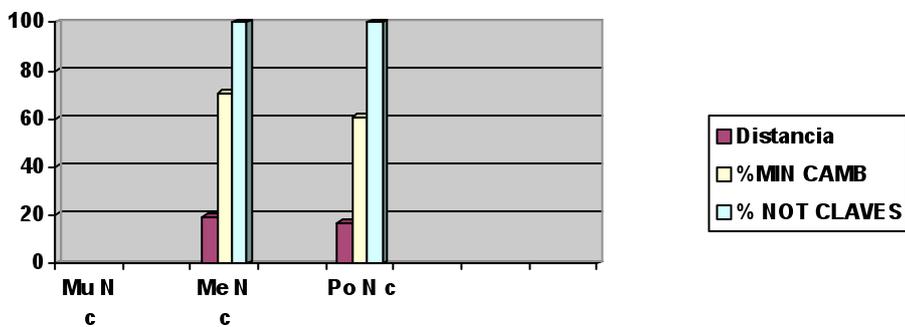


Sin la existencia de ningún error, lo que se observa es que se obtienen resultados muy parecidos a los del error ≤ 0.01

Es decir no conseguimos completar ninguna actividad larga, pero se aproximan las de cantidad media y baja de notificaciones claves. O tal vez las de cantidad corta de notificaciones claves, podrían detectarse completamente dependiendo de las notificaciones claves de las que esta compuesta.

Actividad Larga comparada con actividades de longitud Corta

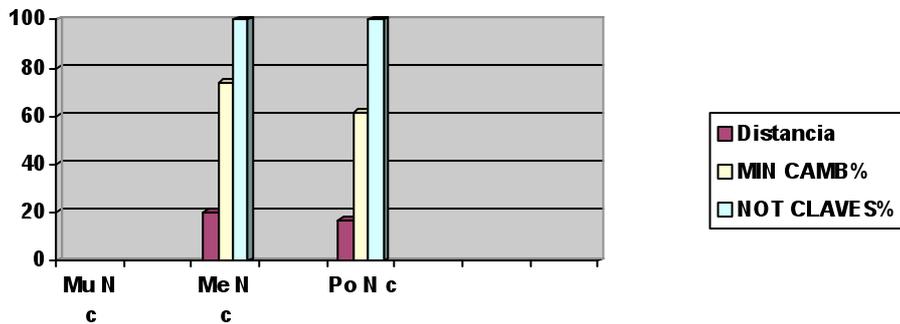
Error ≤ 0.10



Partiendo de una actividad larga de partida, con un error alto ,se aprecia que las actividades de longitud corta por muy alto que sea el error, quedan detectadas completamente. Esto es normal ya que las actividades cortas estan compuestas por menos notificaciones claves , que las actividades largas.

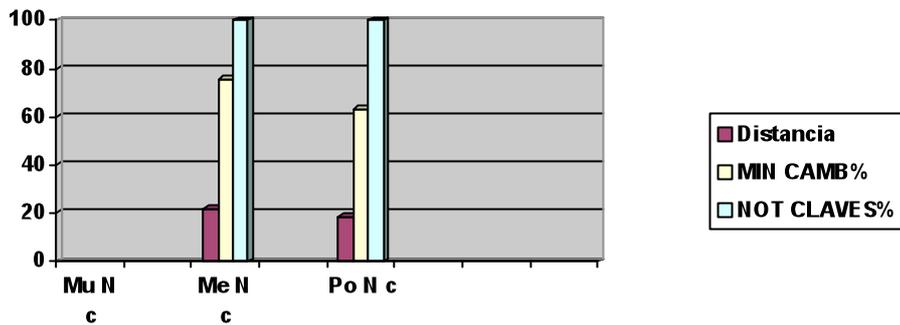
Las distancias y los porcentajes de cambios son altos, ya que estamos comparando cadenas muy largas con cadenas cortas, pero aun y todo a pesar de estos se consiguen completar las actividades cortas.

Error ≤ 0.05



En este caso se observa que los resultados obtenidos son parecidos a los de con un error ≤ 0.10 . Las distancias obtenidas y los cambios de porcentaje también tienen valores similares y se ve que también se consiguen completar las actividades de longitud corta con una cantidad pequeña o media de notificaciones claves.

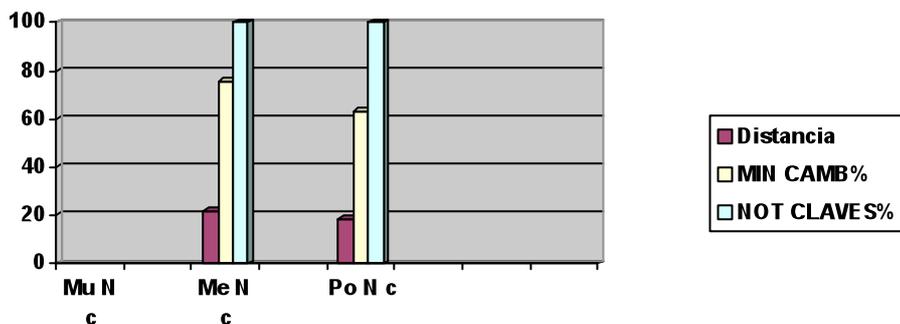
Error ≤ 0.01



Con un error < 0.01 lo que se observa es que también se completan las actividades de longitud corta, pero las distancias crecen ya que la cadena de la longitud larga se hace mayor, cuando el porcentaje de error es más pequeño.

Y también se observa que los cambios de porcentaje han subido de valor.

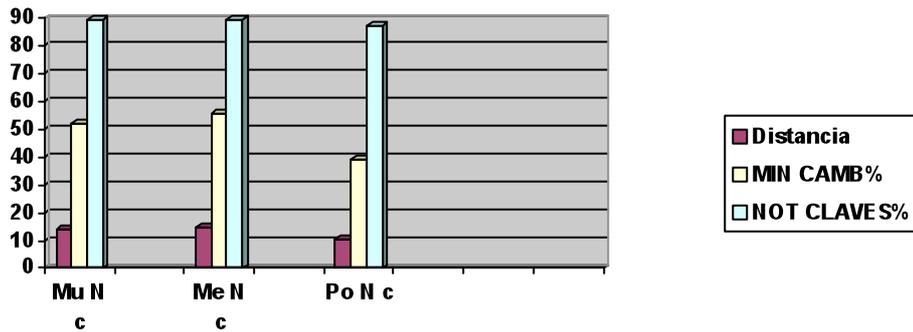
Error=0



Resultados obtenidos muy parecidos a los de con un error ≤ 0.01 , se consiguen todas las actividades de longitud corta y se pueden completar con totalidad.

Actividad Larga comparada con actividades de longitud Media

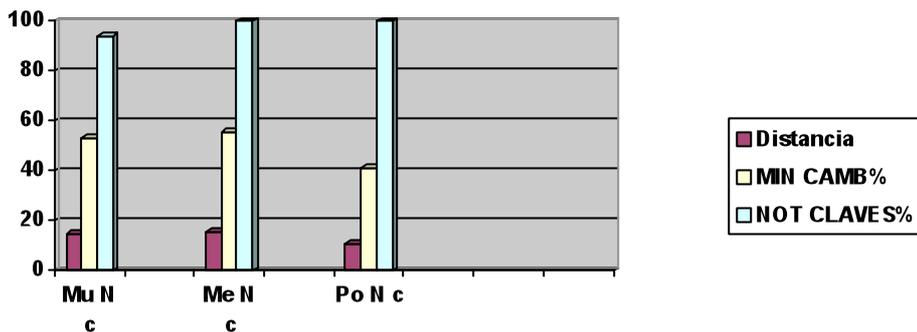
Error ≤ 0.10



Con un error ≤ 0.10 , partiendo de una actividad larga, se observa que se pueden obtener actividades de longitud media aunque no con totalidad, lo máximo se acercan a un 90%, que también es de gran consideración. Ya que eso también dependerá de cuáles son las notificaciones que componen dichas actividades.

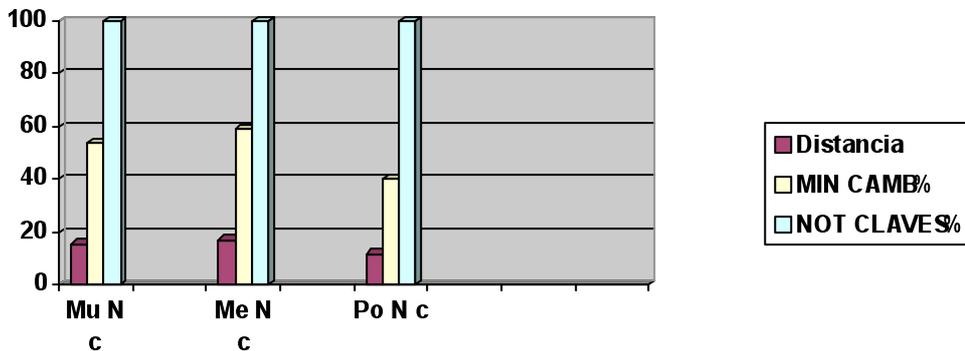
Se observa que las que menos se obtienen son las actividades medias con una cantidad pequeña de notificaciones claves, porque igual justo estas notificaciones no están contenidas en la actividad larga de partida.

Error ≤ 0.05



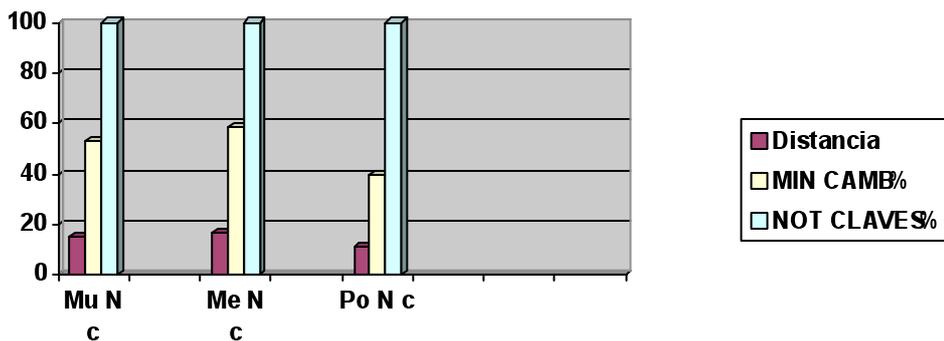
Con un error ≤ 0.05 observamos que partiendo de una actividad de longitud larga, se han podido completar las actividades de longitud media que están compuestas por una cantidad pequeña y media de notificaciones claves, pero las de mucha cantidad no se han podido completar, ya que son cadenas más largas parecidas a las actividades de longitud larga y al existir un error, se observa que faltan notificaciones claves para poder considerarlas completadas.

Error <=0.01



Con un error tal y como es 0.01 apreciamos que partiendo de actividades de longitud larga, se han podido completar las actividades de longitud media totalmente. Ya que al haberse recibido correctamente casi o todas las notificaciones claves de la actividad de longitud larga, existe mas posibilidad de completar las actividades de la longitud media.

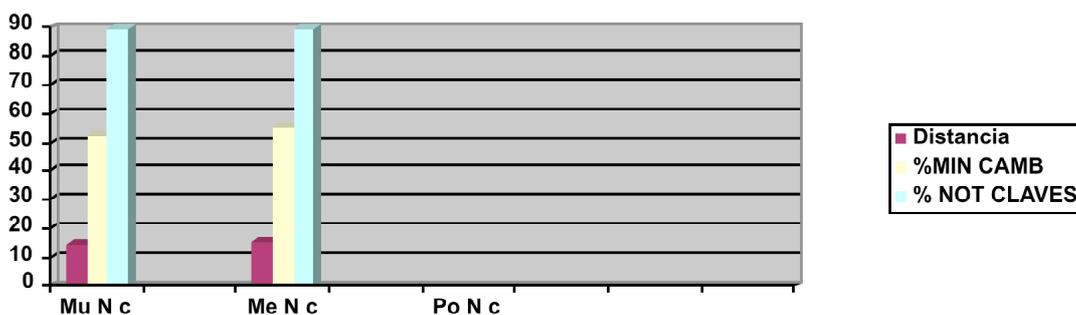
Error=0



Sin la existencia de ningún error se observa que partiendo de una actividad de longitud larga, se han podido detectar las actividades de longitud media de muchas, medias o pocas notificaciones claves .

Actividad Larga comparada con actividades de longitud Larga

Error <=0.10

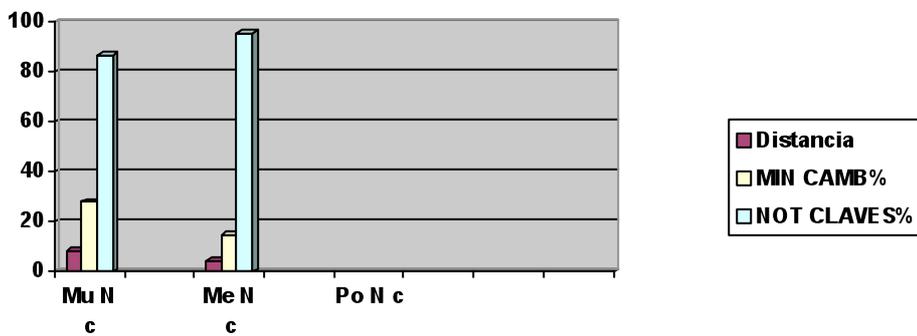


Al tener un error ≤ 0.10 se observa que partiendo de una actividad larga, no es posible detectar todas las actividades larga, con un 90 % se aproxima la detección de las actividades largas cuya cantidad de notificaciones claves es media, y por debajo de este porcentaje estan las actividades largas con una cantidad larga de notificaciones claves.

En este caso es muy importante el error, ya que al ser alto muchas de las notificaciones claves que forman parte de las actividades se han podido no poder detectar.

Se observa que la distancia y los porcentajes de cambio no son muy altos, al estar comparando cadenas prácticamente de la misma longitud.

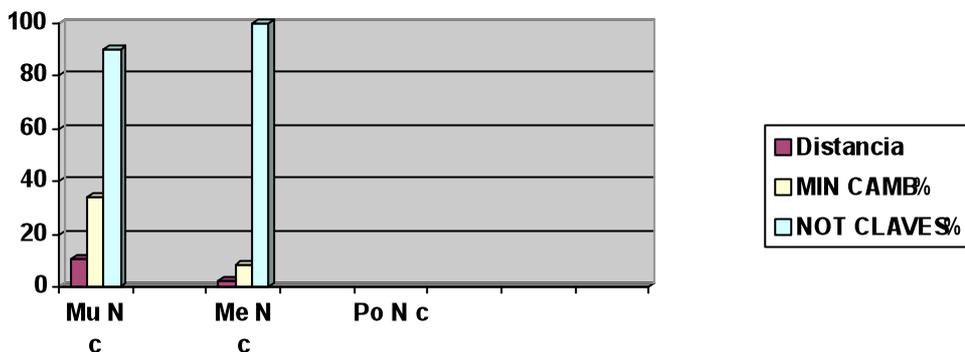
Error ≤ 0.05



Con un error ≤ 0.05 se observa que al estar comparando actividades de longitud larga con actividades de longitud larga, se pueden detectar con mas de 90 % pero no totalmente las actividades de longitud larga con cantidad media de notificaciones claves, por consiguiente las de pocas notificaciones se tendrían mas probabilidad de haber sido completadas.

Las actividades largas con muchas notificaciones claves no quedarían detectadas totalmente ya que se aproximan a un 90 %.

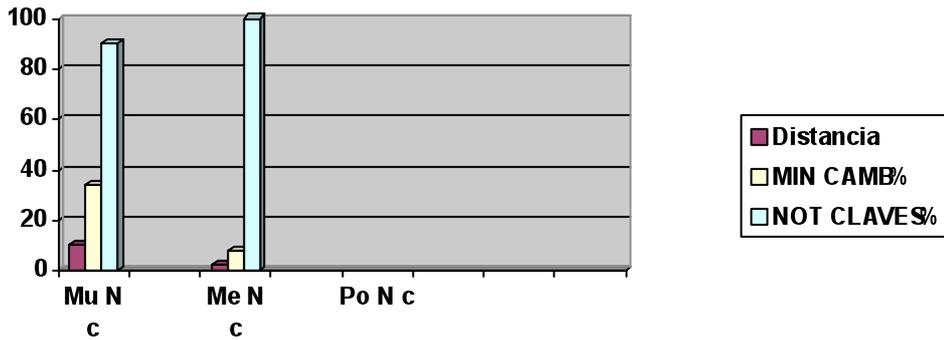
Error ≤ 0.01



Con un error ≤ 0.01 partiendo de actividades largas se pueden obtener, las actividades largas que tienen una cantidad de notificaciones claves media y pequeña al 100 %, pero las de cantidades largas de notificaciones claves no se detectarían totalmente, ya que esto tambien

depende mucho si en la simulación de la recogida de actividades, se han recogido mas actividades largas con muchas o con medias cantidades de notificaciones claves.

Error=0



Sin ningun error presente, partiendo de actividades de longitud larga , se detectan al 100 % las actividades de longitud larga con una cantidad media y pequeña de notificaciones claves, pero no las de una cantidad larga de notificaciones claves, aunque el % de cumplimiento tambien se acerca a la totalidad.

CAPÍTULO 7: SIMULACIÓN

Capturas de pantalla, salida de la aplicación:

```
*****imprimiendo actividad sacada por el random*****
01
las notificaciones requeridas son:
N1,N2,N3,N4,N5,N6,N7,N8,N9,
-----
*****asignando a cada notificacion de las del operario un numero aleato
rio*****
-----
N1- 0,22
N2- 0
N3- 0,48
N4- 0,57
N5- 0,14
N6- 0,27
N7- 0,67
N8- 0,73
N9- 0,66
-----
*****las notificaciones con una probabilidad mayor/igual que el error inpuet
o son:
-----
N1- 0,22
N3- 0,48
N4- 0,57
N6- 0,27
N7- 0,67
N8- 0,73
N9- 0,66
```

```
el tiempo total dedicado desde 1º hasta ultima notificacion es :
35,68
```

```
*****creando cadena para las notificaciones que se deberian hacer *****:
-----
notificaciones que deberian hacerse para la actividad dada:
N1,N2,N3,N4,N5,N6,N7,N8,N9,
-----
*****creando cadena para las notificaciones que se han recibido correctanmete****
-----
cadena de notificaciones realmente hechas:
N1,N3,N4,N6,N7,N8,N9,
```


CAPÍTULO 8 : CONCLUSIONES

La realización de este proyecto fin de carrera ha consistido en obtener el análisis, diseño e implementación de una aplicación con la cual una empresa será capaz de conocer el flujo de trabajo realizado por sus trabajadores gracias a la información almacenada en una PDA. Ya que estos dispositivos son usados para almacenar información que puede ser consultada a cualquier hora y en cualquier lugar.

En el desarrollo de este proyecto se han aprendido las siguientes partes:

El lenguaje Visual C# que se ha utilizado para la realización de la aplicación.

La utilización de un sistema gestor de datos como es Microsoft SQL Server 2008 Compact 3.5, que tampoco se ha visto anteriormente en la carrera.

Desarrollar 3 partes importantes de un proyecto como son el análisis, el diseño y la implementación.

En resumen, se puede decir que el desarrollo de este proyecto ha ayudado notablemente en la formación sobre nuevos lenguajes de programación que, hoy en día, son de los más requeridos en el mundo del mercado de empleo.

CAPÍTULO 9: LINEAS FUTURAS

Se podría dar la posibilidad de utilizar el análisis obtenido tras el estudio de la detección de las actividades para:

- Detectar fallos de sensores. Es un tema muy tratado hoy en día y con el estudio de detección de actividades a partir de otras se podría aplicar a este tema, tras tener establecidos unos criterios para considerar el funcionamiento correcto de los sensores e ir obteniendo los fallos que se pueden producir.

- Utilizar otro tipo de métricas además de Levenshtein, como serían la Distancia de Hamming o la Distancia de Damerau-Levenshtein, que son otros tipos de algoritmos utilizados para el cálculo de distancia entre cadenas, pero en el caso del cálculo de la Distancia de Hamming, la podríamos utilizar solo entre cadenas de la misma longitud pero compuestas por diferentes notificaciones.

Es decir que la podríamos aplicar entre cadenas de longitud media, larga o corta. Podría ser nos de gran utilidad.

- Utilizar diferentes métricas y hacer una votación entre ellas para tratar de ajustar mejor la selección de la actividad.

- Posibilidad de utilizar más actividades para realizar el estudio, así obtendremos unos resultados de los que podremos estar seguros que no pueden tener variación ninguna. Ya que cuanto más actividades son tratadas en el estudio, mayor seguridad para poder concluir.

CAPÍTULO 10 : BIBLIOGRAFÍA

 [A] INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS CON VISUAL BASIC .NET ,Clark Dan ,Editorial: Anaya Multimedia

 [B] VISUAL C# .NET ,Francisco Charte ,Editorial: Anaya

 [C] INGENIERIA DEL SOFTWARE, UN ENFOQUE PRÁCTICO, *Pressman R.*, Editorial: McGraw-Hill

 [1] Distancia de Damerau-Levenshtein-Wikipedia,la enciclopedia libre
<http://es.wikipedia.org/wiki/Distancia_de_Damerau-Levenshtein>

 [2] Curso de Iniciación a C#
<<http://www.elguille.info/NEt/cursocsharprik/index.htm>>

 [3] C# Station <<http://www.csharp-station.com/Tutorial.aspx>>

 [4] Distance Between Strings
<http://www.cut-the-knot.org/do_you_know/Strings.shtml>

 [5] Random (Clase)(System)
<<http://msdn.microsoft.com/es-es/library/system.random.aspx>>

 [6] System.Data.SqlServerCe<[http://msdn.microsoft.com/es-es/library/system.data.sqlserverce\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/system.data.sqlserverce(VS.80).aspx)>

 [7] Sql Server Ce<http://es.wikipedia.org/wiki/SqlServer_ce>

 [8] SqlCeConnection
<[http://msdn.microsoft.com/es-es/library/system.data.sqlserverce.sqlceconnection\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/system.data.sqlserverce.sqlceconnection(VS.80).aspx)>

 [9] SqlCeDataAdapter
<[http://msdn.microsoft.com/es-es/library/system.data.sqlserverce.sqlcedataadapter\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/system.data.sqlserverce.sqlcedataadapter(VS.80).aspx)>

 [10] Desarrolladores C#<<http://msdn.microsoft.com/es-es/vcsharp/default.aspx>>

- 🔗 [11]Aprendiendo a usar Sql <<http://temas-dotnet.blogspot.com/2009/03/aprendiendo-usar-sql-server-ce-35-sp1.html>>
- 🔗 [12]Agregar una BD< [http://msdn.microsoft.com/es-es/library/aa983322\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/aa983322(VS.80).aspx) >
- 🔗 [13]ClaseDictionary<[http://msdn.microsoft.com/es-es/library/xfhwa508\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/xfhwa508(VS.80).aspx)>
- 🔗 [14]Distancia deLevenshtein<http://es.wikipedia.org/wiki/Distancia_de_Levenshtein>

ANEXO

Notificaciones claves consideradas por actividad:

A1 → N1, N3, N5, N7, N8
A2 → N5, N9
A3 → N8, N6, N1, N5, N3
A4 → N2
A5 → N2, N8
A6 → N7, N5, N4
A7 → N7, N9
A8 → N8, N3

Notificaciones ordenadas que componen cada actividad:

A1 compuesta por: N1, N2, N3, N4, N5, N6, N7, N8,
A2 compuesta por: N5, N3, N7, N4, N9, N2,
A3 compuesta por: N8, N6, N1, N2, N5, N3,
A4 compuesta por: N2, N4, N6, N8,
A5 compuesta por: N1, N2, N3, N4, N7, N8,
A6 compuesta por: N8, N7, N6, N5, N4,
A7 compuesta por: N1, N3, N5, N7, N9,
A8 compuesto por: N9, N8, N3, N2, N1,

Se realizara un análisis sobre los resultados que se obtienen para los distintos conjuntos posibles de haber sido realizados como flujo de trabajo del operario, según la variación del error impuesto como limite del correcto funcionamiento del dispositivo.

Error =0

Es decir son recibidos correctamente todas las notificaciones, no se ha producido ningún error.

Veremos como afecta esto a las notificaciones de cada actividad en particular.

ACTIVIDAD A1 obtenida con Random

Conjunto de notificaciones generadas para el operario: N1,N2,N3,N4,N5,N6,N7,N8,

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

N1- 0,7

- N2- 0,1
- N3- 0,3
- N4- 0,9
- N5- 0,1
- N6- 0,7
- N7- 0,3
- N8- 0,4

Calculo de distancia con el algoritmo de Levenshtein:

- La distancia con A1 es 0 y su porcentaje de cambios es 0
- La distancia con A2 es 11 y su porcentaje de cambios es 0,4583333333333333
- La distancia con A3 es 11 y su porcentaje de cambios es 0,4583333333333333
- La distancia con A4 es 12 y su porcentaje de cambios es 0,5
- La distancia con A5 es 6 y su porcentaje de cambios es 0,25
- La distancia con A6 es 13 y su porcentaje de cambios es 0,5416666666666667
- La distancia con A7 es 10 y su porcentaje de cambios es 0,4166666666666667
- La distancia con A8 es 13 y su porcentaje de cambios es 0,5416666666666667

El valor mínimo de distancia es: 0

El valor máximo de distancia es: 13

El valor mínimo de % de cambios es: 0

El valor máximo de % de cambios es: 0,5416666666666667

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0
La actividad a la que corresponde dicha cadena es la: A1

Estudio para el cumplimiento de notificaciones claves:

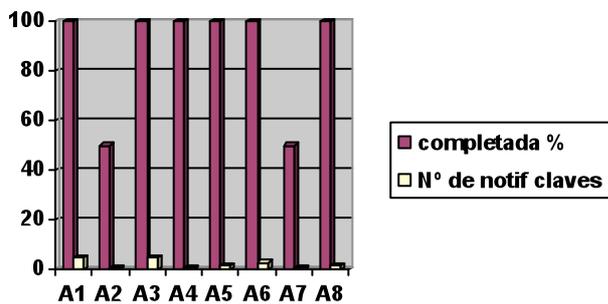
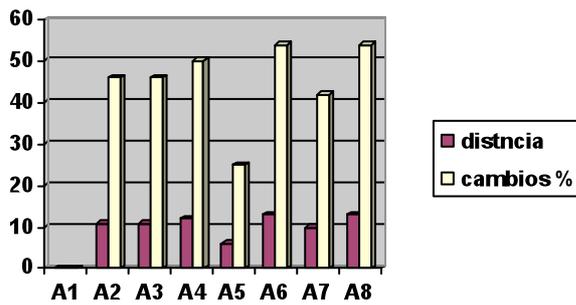
- PorcentajeA1:100 5
- PorcentajeA2:50 1
- PorcentajeA3:100 5
- PorcentajeA4:100 1
- PorcentajeA5:100 2
- PorcentajeA6:100 3
- PorcentajeA7:50 1
- PorcentajeA8:100 2

El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES: 100

La actividad a la que corresponde dicho porcentaje es:
A1 A3 A4 A5 A6 A8

Toma de decisión:

POR NOTIFICACIONES CLAVES A1, A3, A4, A5, A6, A8
POR DISTANCIA DE LEVENSHTEIN A1
ACTIVIDAD REALIZADA POR EL OPERARIO A1



ACTIVIDAD A2 obtenida con Random

Conjunto de notificaciones generadas para el operario: N5, N3, N7, N4, N9, N2,

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

- N5- 1
- N3- 0,3
- N7- 0,7
- N4- 0,8
- N9- 0,2

N2- 0,6

Calculo de distancia con el algoritmo de Levenshtein:

La distancia con A1 es 11 y su porcentaje de cambios es 0,4583333333333333
La distancia con A2 es 0 y su porcentaje de cambios es 0
La distancia con A3 es 6 y su porcentaje de cambios es 0,3333333333333333
La distancia con A4 es 9 y su porcentaje de cambios es 0,5
La distancia con A5 es 5 y su porcentaje de cambios es 0,2777777777777778
La distancia con A6 es 7 y su porcentaje de cambios es 0,3888888888888889
La distancia con A7 es 6 y su porcentaje de cambios es 0,3333333333333333
La distancia con A8 es 8 y su porcentaje de cambios es 0,4444444444444444

El valor mínimo de distancia es: 0

El valor máximo de distancia es: 11

El valor mínimo de % de cambios es: 0

El valor máximo de % de cambios es: 0,5

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0

La actividad a la que corresponde dicha cadena es la: A2

Estudio para el cumplimiento de notificaciones claves:

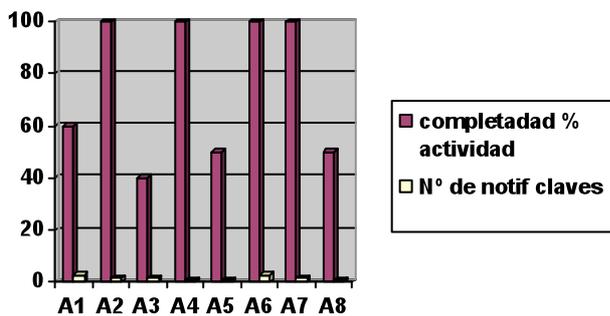
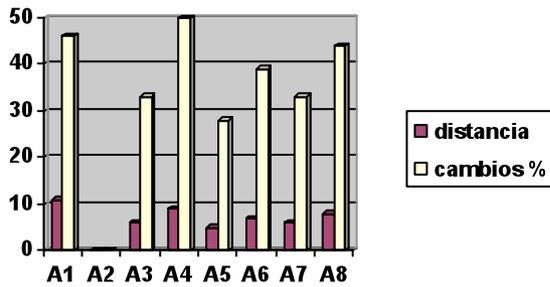
PorcentajeA1:60 3
PorcentajeA2:100 2
PorcentajeA3:40 2
PorcentajeA4:100 1
PorcentajeA5:50 1
PorcentajeA6:100 3
PorcentajeA7:100 2
PorcentajeA8:50 1

El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES : 100

La actividad a la que corresponde dicho porcentaje es: A2 A4 A6 A7

Toma de decisión:

POR NOTIFICACIONES CLAVES A2, A4, A6, A7
POR DISTANCIA DE LEVENSHTTEIN A2
ACTIVIDAD REALIZADA POR EL OPERARIO A2



ACTIVIDAD A3 obtenida con Random

Conjunto de notificaciones generadas para el operario: N8, N6, N1, N2, N5, N3,

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

- N8- 0,9
- N6- 1
- N1- 0,7
- N2- 0,3
- N5- 0,3
- N3- 0,4

Calculo de distancia con el algoritmo de Levenshtein:

- La distancia con A1 es 11 y su porcentaje de cambios es 0,4583333333333333
- La distancia con A2 es 6 y su porcentaje de cambios es 0,3333333333333333
- La distancia con A3 es 0 y su porcentaje de cambios es 0
- La distancia con A4 es 10 y su porcentaje de cambios es 0,5555555555555556
- La distancia con A5 es 6 y su porcentaje de cambios es 0,3333333333333333
- La distancia con A6 es 6 y su porcentaje de cambios es 0,3333333333333333
- La distancia con A7 es 8 y su porcentaje de cambios es 0,4444444444444444

La distancia con A8 es 7 y su porcentaje de cambios es 0,388888888888889

El valor mínimo de distancia es: 0

El valor máximo de distancia es: 11

El valor mínimo de % de cambios es: 0

El valor máximo de % de cambios es: 0,555555555555556

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0

La actividad a la que corresponde dicha cadena es la: A3

Estudio para el cumplimiento de notificaciones claves:

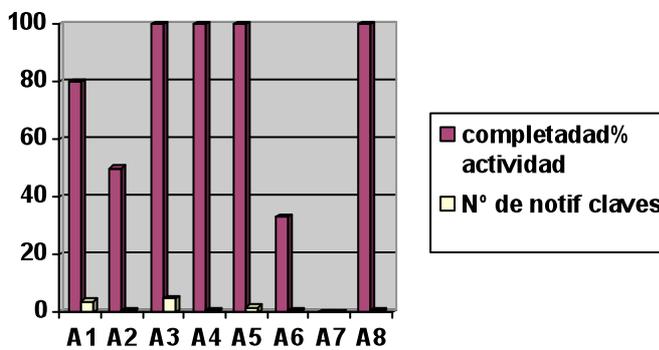
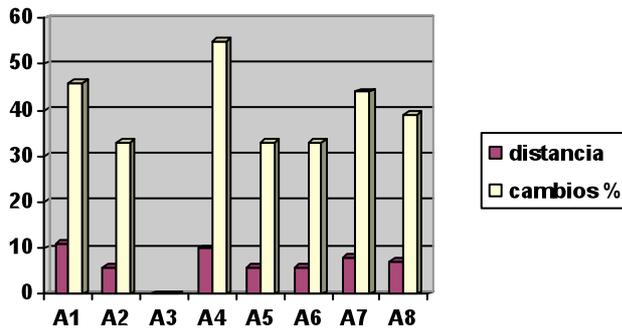
PorcentajeA1:80 4
PorcentajeA2:50 1
PorcentajeA3:100 5
PorcentajeA4:100 1
PorcentajeA5:100 2
PorcentajeA6:33 1
PorcentajeA7:0 0
PorcentajeA8:100 2

El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES :
100

La actividad a la que corresponde dicho porcentaje es: A2 A4 A6 A7

Toma de decisión:

POR NOTIFICACIONES CLAVES A3, A4, A5, A8
POR DISTANCIA DE LEVENSHTTEIN A3
ACTIVIDAD REALIZADA POR EL OPERARIO A3



ACTIVIDAD A4 obtenida con Random

Conjunto de notificaciones generadas para el operario: N2, N4, N6, N8,

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

- N2- 0,1
- N4- 0,7
- N6- 0,9
- N8- 0,6

Calculo de distancia con el algoritmo de Levenshtein:

- La distancia con A1 es 12 y su porcentaje de cambios es 0,5
- La distancia con A2 es 9 y su porcentaje de cambios es 0,5
- La distancia con A3 es 10 y su porcentaje de cambios es 0,5555555555555556
- La distancia con A4 es 0 y su porcentaje de cambios es 0
- La distancia con A5 es 7 y su porcentaje de cambios es 0,3888888888888889
- La distancia con A6 es 6 y su porcentaje de cambios es 0,4
- La distancia con A7 es 7 y su porcentaje de cambios es 0,4666666666666667

La distancia con A8 es 7 y su porcentaje de cambios es 0,4666666666666667

El valor mínimo de distancia es: 0

El valor máximo de distancia es: 12

El valor mínimo de % de cambios es: 0

El valor máximo de % de cambios es: 0,5555555555555556

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0

La actividad a la que corresponde dicha cadena es la: A4

Estudio para el cumplimiento de notificaciones claves:

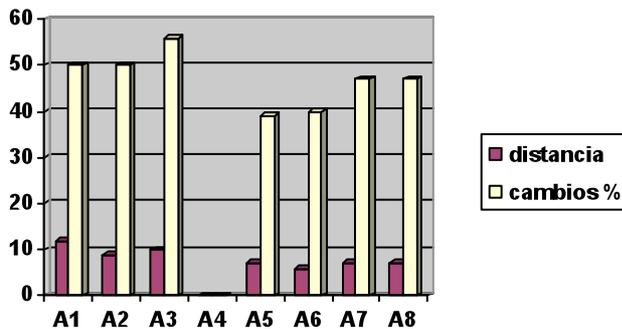
- PorcentajeA1:20 1
- PorcentajeA2:0 0
- PorcentajeA3:40 2
- PorcentajeA4:100 1
- PorcentajeA5:100 2
- PorcentajeA6:33 1
- PorcentajeA7:0 0
- PorcentajeA8:50 1

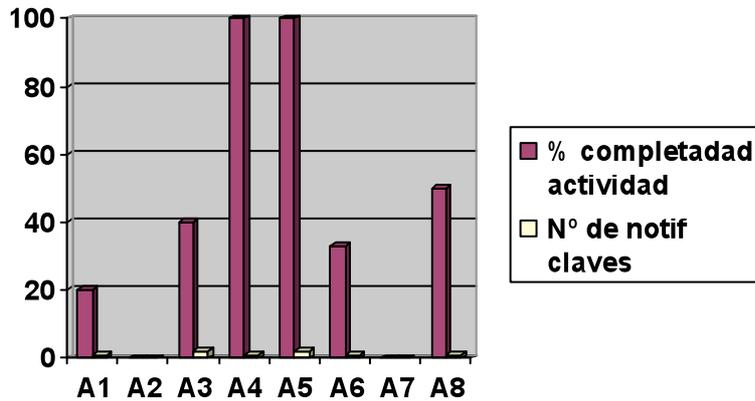
El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES: 100

La actividad a la que corresponde dicho porcentaje es: A2 A4 A6 A7

Toma de decisión:

POR NOTIFICACIONES CLAVES A4, A5
 POR DISTANCIA DE LEVENSHTTEIN A4
 ACTIVIDAD REALIZADA POR EL OPERARIO A4





ACTIVIDAD A5 obtenida con Random

Conjunto de notificaciones generadas para el operario: N1, N2, N3, N4, N7, N8,

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

- N1- 0,3
- N2- 0,5
- N3- 0,5
- N4- 0,8
- N7- 0,7
- N8- 0,7

Calculo de distancia con el algoritmo de Levenshtein:

- La distancia con A1 es 6 y su porcentaje de cambios es 0,25
- La distancia con A2 es 5 y su porcentaje de cambios es 0,2777777777777778
- La distancia con A3 es 6 y su porcentaje de cambios es 0,3333333333333333
- La distancia con A4 es 7 y su porcentaje de cambios es 0,3888888888888889
- La distancia con A5 es 0 y su porcentaje de cambios es 0
- La distancia con A6 es 8 y su porcentaje de cambios es 0,4444444444444444
- La distancia con A7 es 5 y su porcentaje de cambios es 0,2777777777777778
- La distancia con A8 es 7 y su porcentaje de cambios es 0,3888888888888889

El valor mínimo de distancia es: 0

El valor máximo de distancia es: 8

El valor mínimo de % de cambios es: 0

El valor máximo de % de cambios es: 0,4444444444444444

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0

La actividad a la que corresponde dicha cadena es la: A5

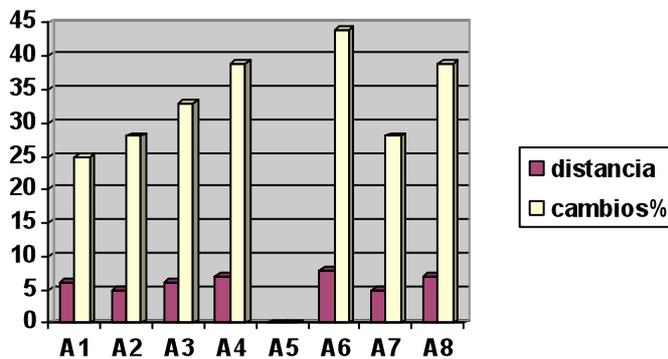
Estudio para el cumplimiento de notificaciones claves:

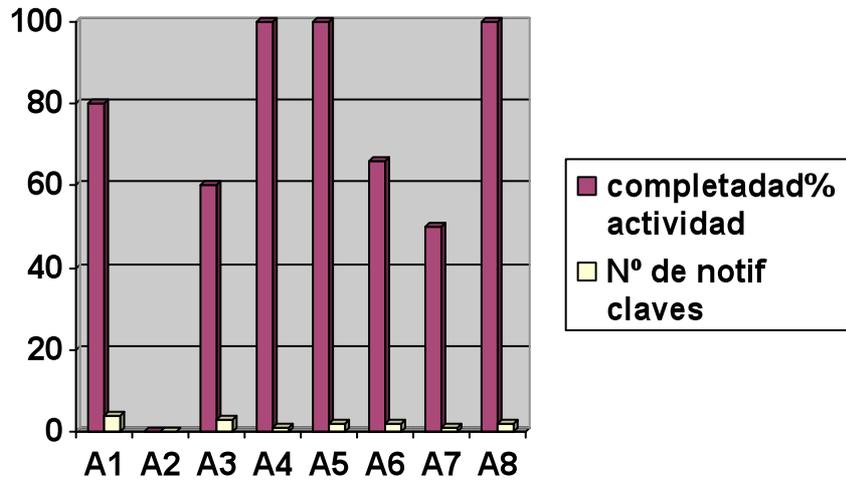
- PorcentajeA1:80 4
- PorcentajeA2:0 0
- PorcentajeA3:60 3
- PorcentajeA4:100 1
- PorcentajeA5:100 2
- PorcentajeA6:66 2
- PorcentajeA7:50 1
- PorcentajeA8:100 2

El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES: 100
La actividad a la que corresponde dicho porcentaje es: A4 A5 A8

Toma de decisión:

POR NOTIFICACIONES CLAVES A4, A5, A8
POR DISTANCIA DE LEVENSHTTEIN A5
ACTIVIDAD REALIZADA POR EL OPERARIO A5





ACTIVIDAD A6 obtenida con Random

Conjunto de notificaciones generadas para el operario: N8, N7, N6, N5, N4,

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

- N8- 0,4
- N7- 0,1
- N6- 0,3
- N5- 0,7
- N4- 0,3

Calculo de distancia con el algoritmo de Levenshtein:

- La distancia con A1 es 13 y su porcentaje de cambios es 0,541666666666667
- La distancia con A2 es 7 y su porcentaje de cambios es 0,388888888888889
- La distancia con A3 es 6 y su porcentaje de cambios es 0,333333333333333
- La distancia con A4 es 6 y su porcentaje de cambios es 0,4
- La distancia con A5 es 8 y su porcentaje de cambios es 0,444444444444444
- La distancia con A6 es 0 y su porcentaje de cambios es 0
- La distancia con A7 es 5 y su porcentaje de cambios es 0,333333333333333
- La distancia con A8 es 5 y su porcentaje de cambios es 0,333333333333333

El valor mínimo de distancia es: 0
El valor máximo de distancia es: 13
El valor mínimo de % de cambios es: 0
El valor máximo de % de cambios es: 0,541666666666667

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0

La actividad a la que corresponde dicha cadena es la: A6

Estudio para el cumplimiento de notificaciones claves:

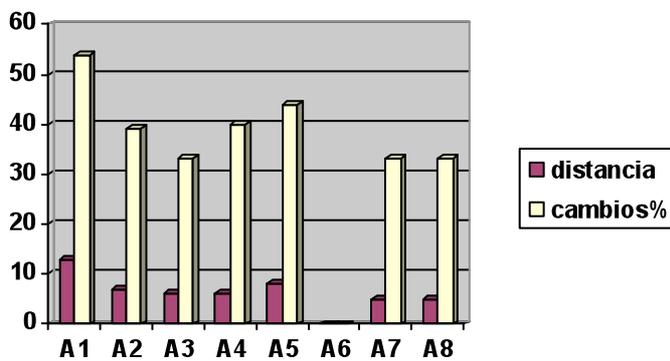
PorcentajeA1:60 3
PorcentajeA2:50 1
PorcentajeA3:60 3
PorcentajeA4:0 0
PorcentajeA5:50 1
PorcentajeA6:100 3
PorcentajeA7:50 1
PorcentajeA8:50 1

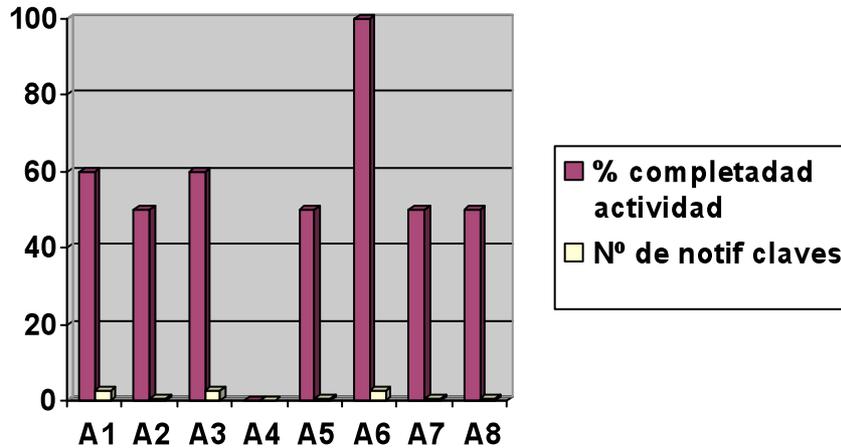
El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES :100

La actividad a la que corresponde dicho porcentaje es: A6

Toma de decisión:

**POR NOTIFICACIONES CLAVES A6
POR DISTANCIA DE LEVENSHTTEIN A6
ACTIVIDAD REALIZADA POR EL OPERARIO A6**





ACTIVIDAD A7 obtenida con Random

Conjunto de notificaciones generadas para el operario: N1, N3, N5, N7, N9

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

- N1- 0,5
- N3- 0,5
- N5- 0,2
- N7- 0,1
- N9- 0,1

Calculo de distancia con el algoritmo de Levenshtein:

- La distancia con A1 es 10 y su porcentaje de cambios es 0,416666666666667
- La distancia con A2 es 6 y su porcentaje de cambios es 0,333333333333333
- La distancia con A3 es 8 y su porcentaje de cambios es 0,444444444444444
- La distancia con A4 es 7 y su porcentaje de cambios es 0,466666666666667
- La distancia con A5 es 5 y su porcentaje de cambios es 0,277777777777778
- La distancia con A6 es 5 y su porcentaje de cambios es 0,333333333333333
- La distancia con A7 es 0 y su porcentaje de cambios es 0
- La distancia con A8 es 5 y su porcentaje de cambios es 0,333333333333333

- El valor mínimo de distancia es: 0
- El valor máximo de distancia es: 10
- El valor mínimo de % de cambios es: 0
- El valor máximo de % de cambios es: 0,466666666666667

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0

La actividad a la que corresponde dicha cadena es la: A7

Estudio para el cumplimiento de notificaciones claves:

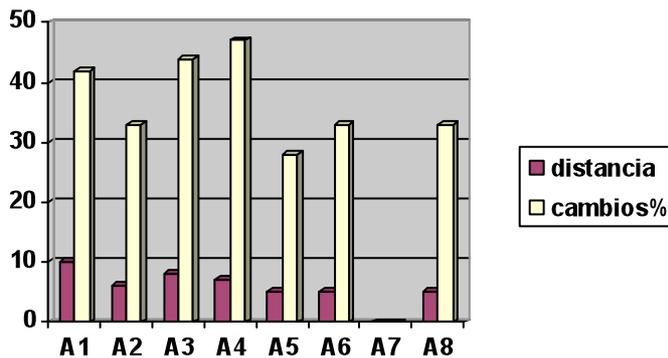
PorcentajeA1: 80 4
PorcentajeA2:100 2
PorcentajeA3:60 3
PorcentajeA4:0 0
PorcentajeA5:0 0
PorcentajeA6:66 2
PorcentajeA7:100 2
PorcentajeA8:50 1

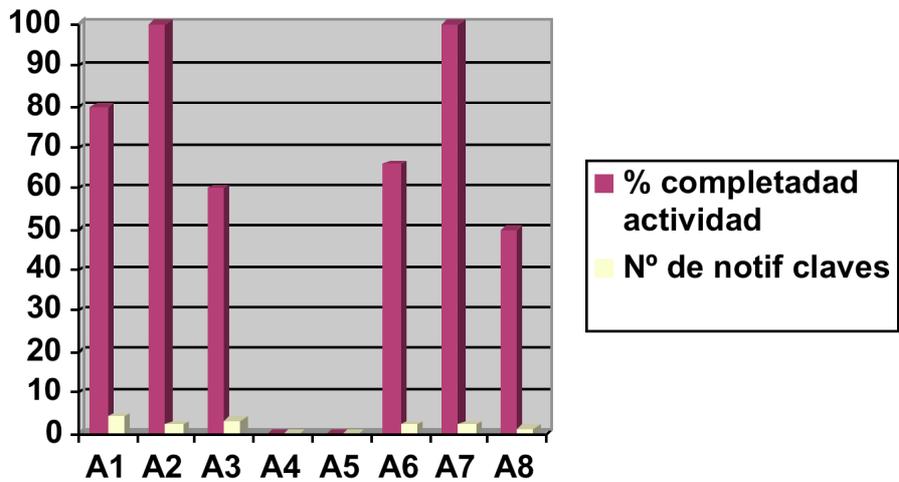
El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES: 100

La actividad a la que corresponde dicho porcentaje es: A2 A7

Toma de decisión:

POR NOTIFICACIONES CLAVES A2 A7
POR DISTANCIA DE LEVENSHTTEIN A7
ACTIVIDAD REALIZADA POR EL OPERARIO A7





ACTIVIDAD A8 obtenida con Random

Conjunto de notificaciones generadas para el operario: N9, N8, N3, N2, N1,

Valores aleatorios correspondientes de haber ocurrido cada una de las notificaciones, es decir la probabilidad que tienen de haber sido correctamente recibidas:

- N9- 0,5
- N8- 0,6
- N3- 0,5
- N2- 0,6
- N1- 0,9

Calculo de distancia con el algoritmo de Levenshtein:

- La distancia con A1 es 13 y su porcentaje de cambios es 0,541666666666667
- La distancia con A2 es 8 y su porcentaje de cambios es 0,444444444444444
- La distancia con A3 es 7 y su porcentaje de cambios es 0,388888888888889
- La distancia con A4 es 7 y su porcentaje de cambios es 0,466666666666667
- La distancia con A5 es 7 y su porcentaje de cambios es 0,388888888888889
- La distancia con A6 es 5 y su porcentaje de cambios es 0,333333333333333
- La distancia con A7 es 5 y su porcentaje de cambios es 0,333333333333333
- La distancia con A8 es 0 y su porcentaje de cambios es 0

El valor mínimo de distancia es: 0

El valor máximo de distancia es: 13
El valor mínimo de % de cambios es: 0
El valor máximo de % de cambios es: 0,541666666666667

Consideramos valores aceptados _ mínimos: Distancia: 0 Porcentaje de cambios: 0

La actividad a la que corresponde dicha cadena es la: A8

Estudio para el cumplimiento de notificaciones claves:

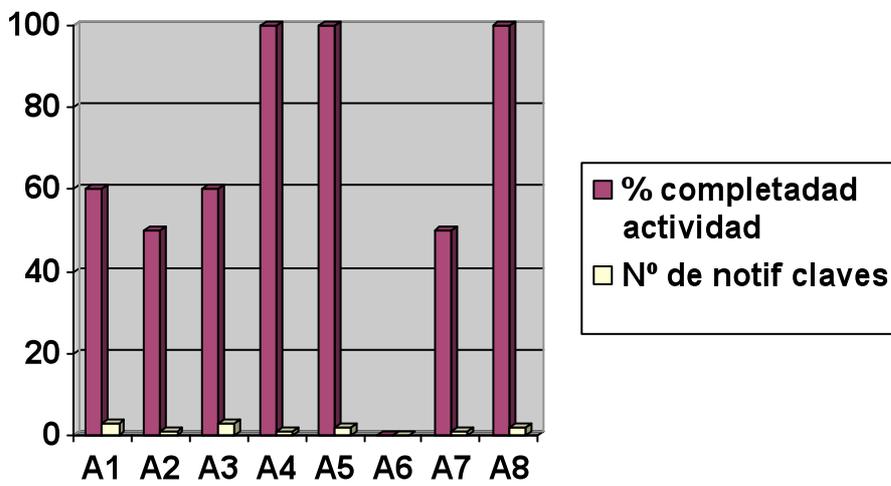
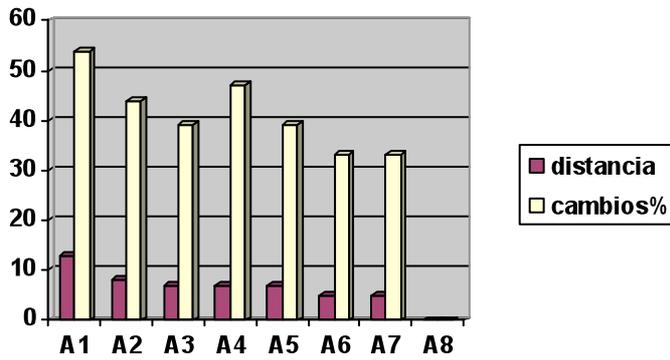
PorcentajeA1: 80 4
PorcentajeA2:50 1
PorcentajeA3:60 3
PorcentajeA4:100 1
PorcentajeA5:100 2
PorcentajeA6:0 0
PorcentajeA7:50 1
PorcentajeA8:100 2

El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES: 100

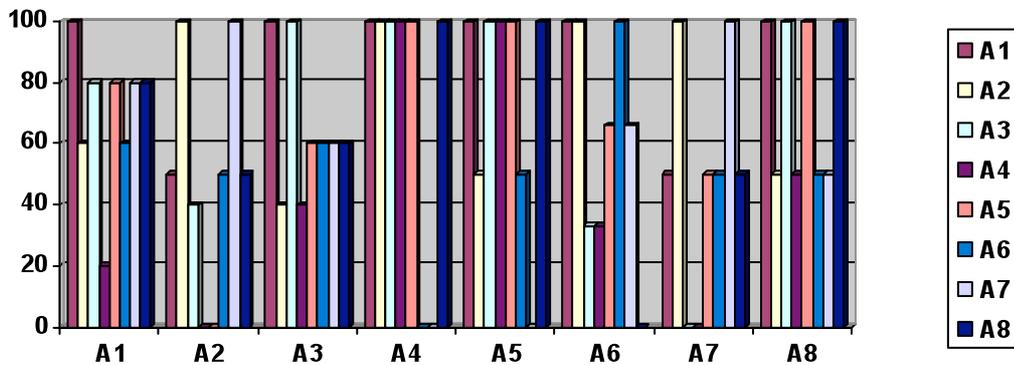
La actividad a la que corresponde dicho porcentaje es: A4 A5 A8

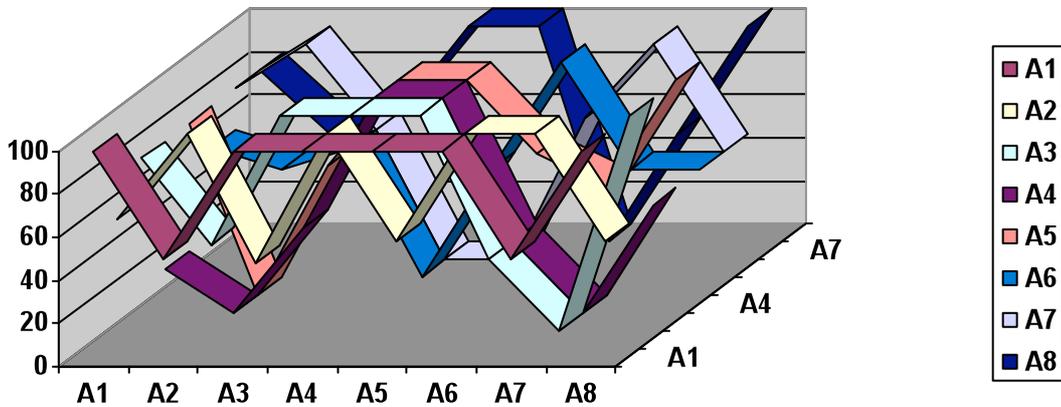
Toma de decisión:

POR NOTIFICACIONES CLAVES A4 A5 A8
POR DISTANCIA DE LEVENSHTein A8
ACTIVIDAD REALIZADA POR EL OPERARIO A8



Graficas para poder observar la visión global del cumplimiento de las notificaciones claves de cada actividad.





Conclusiones obtenidas de estas pruebas, aplicando este criterio de error:

- Cuando a una actividad se le haya marcado un número mínimo de notificaciones claves, tiene mayor probabilidad por haber sido completada, comparando el conjunto de notificaciones claves por cada actividad con el conjunto de notificaciones realizadas por el operario.

- Si el conjunto de notificaciones realizadas por el operario es amplio, es mayor la probabilidad de haber realizado varias actividades, ya que cuanto mayor sea el flujo de trabajo del operario, más notificaciones claves se habrán realizado y estudiando para que actividad, que notificaciones son claves comprobamos las actividades que ha completado el operario según el criterio de cumplimiento de notificaciones claves.

- Comprobamos que por ejemplo las notificaciones que componen la actividad 1, son 8. Con lo cual este conjunto de notificaciones, comparado con las notificaciones de las actividades con 4, 5 y 6 notificaciones, podemos comprobar que según el criterio de notificaciones claves, con el conjunto de 8 notificaciones se completan muchas más actividades, que con las otras.

- Según el criterio de cálculo de distancia entre el flujo de trabajo del operario, es decir el conjunto de notificaciones y los conjuntos de notificaciones que componen las actividades que hay, podemos observar que existe una distancia y un % de cambios nulos, cuando se comparan dos conjuntos idénticos.

Por lo contrario cuando los conjuntos de notificaciones que se comparan cada vez se diferencian más, se obtiene una mayor distancia y un mayor porcentaje de cambios.

Como ya se ha mostrado en la gráfica de la figura podemos observar que, considerando el error = 0 y las notificaciones claves establecidas previamente obtenemos los siguientes resultados:

Conjunto de notificaciones operarias	Actividades completadas según notificaciones claves
N1,N2,N3,N4,N5,N6,N7,N8	A1,A3,A4,A5,A6,A7
N5,N3,N7,N4,N9,N2	A2,A4,A6,A7
N8,N6,N1,N2,N5,N3	A3,A4,A5,A8
N2,N4,N6,N8	A4,A5
N1,N2,N3,N4,N7,N8	A4,A5,A8
N8,N7,N6,N5,N4	A6
N1,N3,N5,N7,N9	A2,A7
N9,N8,N3,N2,N1	A4,A5,A6

Error =0,2

ACTIVIDAD A3 obtenida con Random

Conjunto de notificaciones generadas para el operario: N8,N6,N1,N2,N5,N3

Asignando una probabilidad de haber sido recibidas correctamente las notificaciones:

- N8- 0,8
- N6- 0,6
- N1- 0,2
- N2- 0,1
- N5- 0,7
- N3- 0,1

Notificaciones con probabilidad ≥ 0.2

- N8- 0,8
- N6- 0,6
- N1- 0,2
- N5- 0,7

Criterio de Calculo de distancia entre dos cadenas Levenshtein:

- La distancia con A1 es 15 y su porcentaje de cambios es 0,625
- La distancia con A2 es 10 y su porcentaje de cambios es 0,5555555555555556
- La distancia con A3 es 6 y su porcentaje de cambios es 0,3333333333333333
- La distancia con A4 es 4 y su porcentaje de cambios es 0,3333333333333333
- La distancia con A5 es 10 y su porcentaje de cambios es 0,5555555555555556

La distancia con A6 es 5 y su porcentaje de cambios es 0,3333333333333333
La distancia con A7 es 7 y su porcentaje de cambios es 0,4666666666666667
La distancia con A8 es 6 y su porcentaje de cambios es 0,4

el valor minimo de distancia es:4
el valor maximo de distancia es:15
el valor minimo de % de cambios es:0,3333333333333333
el valor maximo de % de cambios es:0,625
valores aceptados minimos: Distancia:4 Porcentaje de cambios:0,3333333333333333
la cadena de trabajo que se corresponde con la realizada por el operario es:
N2,N4,N6,N8,

La actividad a la que corresponde dicha cadena es la :A4

Criterio de cumplimiento de notificaciones claves:

PorcentajeA1:60 3
PorcentajeA2:50 1
PorcentajeA3:80 4
PorcentajeA4:0 0
PorcentajeA5:50 1
PorcentajeA6:33 1
PorcentajeA7:0 0
PorcentajeA8:50 1

El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES :80
La actividad a la que corresponde dicho porcentaje es: A3

Observamos que aplicando los dos criterios obtenemos resultados diferentes.

ACTIVIDAD A8 obtenida con Random

Conjunto de notificaciones generadas para el operario: N9,N8,N3,N2,N1

N9- 0,5
N8- 0,9
N3- 0,2
N2- 0
N1- 0,1

Notificaciones que superan el error 0.2

N9- 0,5

N8- 0,9

N3- 0,2

Criterio de Levenshtein:

La distancia con A1 es 17 y su porcentaje de cambios es 0,7083333333333333

La distancia con A2 es 12 y su porcentaje de cambios es 0,6666666666666667

La distancia con A3 es 11 y su porcentaje de cambios es 0,6111111111111111

La distancia con A4 es 6 y su porcentaje de cambios es 0,5

La distancia con A5 es 11 y su porcentaje de cambios es 0,6111111111111111

La distancia con A6 es 9 y su porcentaje de cambios es 0,6

La distancia con A7 es 9 y su porcentaje de cambios es 0,6

La distancia con A8 es 6 y su porcentaje de cambios es 0,4

el valor minimo de distancia es:6

el valor maximo de distancia es:17

el valor minimo de % de cambios es:0,4

el valor maximo de % de cambios es:0,7083333333333333

valores aceptados_minimos: Distancia:6 Porcentaje de cambios:0,4

La cadena de trabajo que se corresponde con la realizada por el operario es:

N9,N8,N3,N2,N1,

La actividad a la que corresponde dicha cadena es la :A8

Criterio de cumplimiento de notificaciones claves:

PorcentajeA1:40 2

PorcentajeA2:50 1

PorcentajeA3:40 2

PorcentajeA4:0 0

PorcentajeA5:50 1

PorcentajeA6:0 0

PorcentajeA7:50 1

PorcentajeA8:100 2

el valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES es: 100

la actividad a la que corresponde dicho porcentaje es : A8

ACTIVIDAD A7 obtenida con Random

Conjunto de notificaciones generadas para el operario: N1,N3,N5,N7,N9,

- N1- 0,3
- N3- 0,6
- N5- 0
- N7- 0,5
- N9- 0,7

Notificaciones cuya probabilidad supera el error considerado:

- N1- 0,3
- N3- 0,6
- N7- 0,5
- N9- 0,7

Criterio de Calculo de distancia entre dos cadenas Levenshtein:

- La distancia con A1 es 13 y su porcentaje de cambios es 0,541666666666667
- La distancia con A2 es 7 y su porcentaje de cambios es 0,388888888888889
- La distancia con A3 es 9 y su porcentaje de cambios es 0,5
- La distancia con A4 es 4 y su porcentaje de cambios es 0,333333333333333
- La distancia con A5 es 7 y su porcentaje de cambios es 0,388888888888889
- La distancia con A6 es 7 y su porcentaje de cambios es 0,466666666666667
- La distancia con A7 es 3 y su porcentaje de cambios es 0,2
- La distancia con A8 es 6 y su porcentaje de cambios es 0,4

- el valor minimo de distancia es:3
- el valor maximo de distancia es:13
- el valor minimo de % de cambios es:0,2
- el valor maximo de % de cambios es:0,541666666666667

valores aceptados_minimos: Distancia:3 Porcentaje de cambios:0,2
 la cadena de trabajo que se corresponde con la realizada por el operario es:
 N1,N3,N5,N7,N9,
 la actividad a la que corresponde dicha cadena es la :A7

Criterio de cumplimiento de notificaciones claves:

PorcentajeA1:60 3

PorcentajeA2:50 1
PorcentajeA3:40 2
PorcentajeA4:0 0
PorcentajeA5:0 0
PorcentajeA6:33 1
PorcentajeA7:100 2
PorcentajeA8:50 1

el valor maximo DE PORCENTAJE DE CUMPLIMIENTO ES es:100
la actividad a la que corresponde dicho porcentaje es :A7

ACTIVIDAD A5 obtenida con Random

Conjunto de notificaciones generadas para el operario: N1,N2,N3,N4,N7,N8

N1- 0,1
N2- 0,5
N3- 0,9
N4- 0,7
N7- 0,2
N8- 0

las notificaciones con una probabilidad mayor/igual que 0.2 son:

N2- 0,5
N3- 0,9
N4- 0,7
N7- 0,2

Criterio de Calculo de distancia entre dos cadenas Levenshtein:

La distancia con A1 es 12 y su porcentaje de cambios es 0,5

La distancia con A2 es 8 y su porcentaje de cambios es 0,4444444444444444

La distancia con A3 es 10 y su porcentaje de cambios es 0,5555555555555556

La distancia con A4 es 3 y su porcentaje de cambios es 0,25

La distancia con A5 es 6 y su porcentaje de cambios es 0,3333333333333333

La distancia con A6 es 7 y su porcentaje de cambios es 0,4666666666666667

La distancia con A7 es 5 y su porcentaje de cambios es 0,3333333333333333

La distancia con A8 es 6 y su porcentaje de cambios es 0,4

El valor mínimo de distancia es: 3

El valor máximo de distancia es: 12

El valor mínimo de % de cambios es: 0,25

El valor máximo de % de cambios es: 0,5555555555555556

Valores aceptados _ mínimos: Distancia:3 Porcentaje de cambios:0,25

La cadena de trabajo que se corresponde con la realizada por el operario es:

N2,N4,N6,N8,

La actividad a la que corresponde dicha cadena es la: A4

Criterio de cumplimiento de notificaciones claves:

PorcentajeA1:40 2

PorcentajeA2:0 0

PorcentajeA3:20 1

PorcentajeA4:100 1

PorcentajeA5:50 1

PorcentajeA6:66 2

PorcentajeA7:50 1

PorcentajeA8:50 1

el valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES es:100

la actividad a la que corresponde dicho porcentaje es :A4

ACTIVIDAD A1 obtenida con Random

Conjunto de notificaciones generadas para el operario: N1,N2,N3,N4,N5,N6,N7,N8,

N1- 0,8

N2- 0,9

N3- 0,1

N4- 1

N5- 0,1

N6- 0,1

N7- 0,5

N8- 0,4

las notificaciones con una probabilidad mayor/igual que 0.2 son:

- N1- 0,8
- N2- 0,9
- N4- 1
- N7- 0,5
- N8- 0,4

Criterio de Calculo de distancia entre dos cadenas Levenshtein:

- La distancia con A1 es 9 y su porcentaje de cambios es 0,375
- La distancia con A2 es 7 y su porcentaje de cambios es 0,388888888888889
- La distancia con A3 es 8 y su porcentaje de cambios es 0,444444444444444
- La distancia con A4 es 4 y su porcentaje de cambios es 0,266666666666667
- La distancia con A5 es 3 y su porcentaje de cambios es 0,166666666666667
- La distancia con A6 es 5 y su porcentaje de cambios es 0,333333333333333
- La distancia con A7 es 3 y su porcentaje de cambios es 0,2
- La distancia con A8 es 5 y su porcentaje de cambios es 0,333333333333333

- el valor minimo de distancia es:3
- el valor maximo de distancia es:9
- el valor minimo de % de cambios es:0,166666666666667
- el valor maximo de % de cambios es:0,444444444444444

valores aceptados_minimos:Distancia:3 Porcentaje de cambios:0,166666666666667
 la cadena de trabajo que se corresponde con la realizada por el operario es:
 N1,N2,N3,N4,N7,N8,

la actividad a la que corresponde dicha cadena es la :A5

Criterio de cumplimiento de notificaciones claves:

- PorcentajeA1:60 3
- PorcentajeA2:0 0
- PorcentajeA3:40 2
- PorcentajeA4:100 1
- PorcentajeA5:100 2
- PorcentajeA6:66 2
- PorcentajeA7:50 1
- PorcentajeA8:50 1

el valor maximo DE PORCENTAJE DE CUMPLIMIENTO ES es:100
la actividad a la que corresponde dicho porcentaje es :A4 A5

ACTIVIDAD A6 obtenida con Random

Conjunto de notificaciones generadas para el operario: N8,N7,N6,N5,N4,

- N8- 0,2
- N7- 0,2
- N6- 0
- N5- 0,3
- N4- 0,9

las notificaciones con una probabilidad mayor/igual que 0.2 son:

- N8- 0,2
- N7- 0,2
- N5- 0,3
- N4- 0,9

Criterio de Calculo de distancia entre dos cadenas Levenshtein:

- La distancia con A1 es 15 y su porcentaje de cambios es 0,625
- La distancia con A2 es 9 y su porcentaje de cambios es 0,5
- La distancia con A3 es 8 y su porcentaje de cambios es 0,4444444444444444
- La distancia con A4 es 4 y su porcentaje de cambios es 0,3333333333333333
- La distancia con A5 es 9 y su porcentaje de cambios es 0,5
- La distancia con A6 es 3 y su porcentaje de cambios es 0,2
- La distancia con A7 es 6 y su porcentaje de cambios es 0,4
- La distancia con A8 es 6 y su porcentaje de cambios es 0,4

- el valor mínimo de distancia es:3
- el valor máximo de distancia es:15
- el valor mínimo de % de cambios es:0,2
- el valor máximo de % de cambios es:0,625

valores aceptados_minimos:Distancia:3 Porcentaje de cambios:0,2
la cadena de trabajo que se corresponde con la realizada por el operario es:
N8,N7,N6,N5,N4,
la actividad a la que corresponde dicha cadena es la :A6

Criterio de cumplimiento de notificaciones claves:

- PorcentajeA1:60 3
- PorcentajeA2:50 1
- PorcentajeA3:40 2
- PorcentajeA4:0 0
- PorcentajeA5:50 1
- PorcentajeA6:100 3
- PorcentajeA7:50 1
- PorcentajeA8:50 1

el valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES :100

la actividad a la que corresponde dicho porcentaje es :A6

ACTIVIDAD A4 obtenida con Random

Conjunto de notificaciones generadas para el operario: N2,N4,N6,N8,

- N2- 1
- N4- 0,8
- N6- 0,1
- N8- 0,1

las notificaciones con una probabilidad mayor/igual que 0.2 son:

- N2- 1
- N4- 0,8

Criterio de Calculo de distancia entre dos cadenas Levenshtein:

- La distancia con A1 es 18 y su porcentaje de cambios es 0,75
- La distancia con A2 es 13 y su porcentaje de cambios es 0,7222222222222222
- La distancia con A3 es 13 y su porcentaje de cambios es 0,7222222222222222
- La distancia con A4 es 6 y su porcentaje de cambios es 0,5
- La distancia con A5 es 12 y su porcentaje de cambios es 0,6666666666666667
- La distancia con A6 es 10 y su porcentaje de cambios es 0,6666666666666667
- La distancia con A7 es 11 y su porcentaje de cambios es 0,7333333333333333
- La distancia con A8 es 10 y su porcentaje de cambios es 0,6666666666666667

el valor mínimo de distancia es:6

el valor máximo de distancia es:18
el valor mínimo de % de cambios es:0,5
el valor máximo de % de cambios es:0,75

valores aceptados _ mínimos: Distancia:6 Porcentaje de cambios:0,5
la cadena de trabajo que se corresponde con la realizada por el operario es:
N2,N4,N6,N8,

la actividad a la que corresponde dicha cadena es la :A4

Criterio de cumplimiento de notificaciones claves:

PorcentajeA1:0 0
PorcentajeA2:0 0
PorcentajeA3:0 0
PorcentajeA4:100 1
PorcentajeA5:50 1
PorcentajeA6:33 1
PorcentajeA7:0 0
PorcentajeA8:0 0

el valor maximo DE PORCENTAJE DE CUMPLIMIENTO ES es:100
la actividad a la que corresponde dicho porcentaje es :A4

ACTIVIDAD A2 obtenida con Random

Conjunto de notificaciones generadas para el operario: N5,N3,N7,N4,N9,N2,

N5- 0,7
N3- 0,8
N7- 0,6
N4- 0,9
N9- 0,5
N2- 0,2

las notificaciones con una probabilidad mayor/igual que 0.2 son:

N5- 0,7
N3- 0,8
N7- 0,6

N4- 0,9
N9- 0,5
N2- 0,2

Criterio de Calculo de distancia entre dos cadenas Levenshtein:

La distancia con A1 es 11 y su porcentaje de cambios es 0,4583333333333333
La distancia con A2 es 0 y su porcentaje de cambios es 0
La distancia con A3 es 6 y su porcentaje de cambios es 0,3333333333333333
La distancia con A4 es 9 y su porcentaje de cambios es 0,5
La distancia con A5 es 5 y su porcentaje de cambios es 0,2777777777777778
La distancia con A6 es 7 y su porcentaje de cambios es 0,3888888888888889
La distancia con A7 es 6 y su porcentaje de cambios es 0,3333333333333333
La distancia con A8 es 8 y su porcentaje de cambios es 0,4444444444444444

el valor mínimo de distancia es: 0
el valor máximo de distancia es: 11
el valor mínimo de % de cambios es: 0
el valor máximo de % de cambios es: 0,5

Valores aceptados _ mínimos: Distancia:0 Porcentaje de cambios: 0
La cadena de trabajo que se corresponde con la realizada por el operario es:
N5,N3,N7,N4,N9,N2,

La actividad a la que corresponde dicha cadena es la :A2

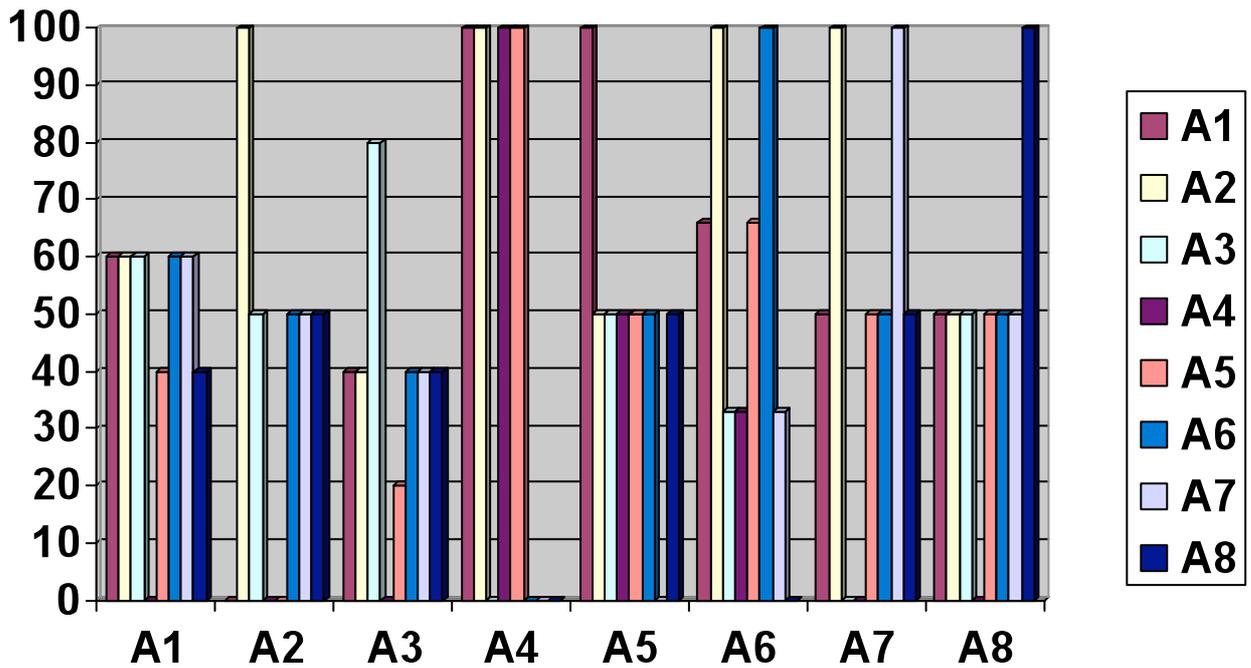
Criterio de cumplimiento de notificaciones claves:

PorcentajeA1:60 3
PorcentajeA2:100 2
PorcentajeA3:40 2
PorcentajeA4:100 1
PorcentajeA5:50 1
PorcentajeA6:100 3
PorcentajeA7:100 2
PorcentajeA8:50 1

El valor máximo DE PORCENTAJE DE CUMPLIMIENTO ES es: 100

La actividad a la que corresponde dicho porcentaje es: A2 A4 A6 A7

Este grafico muestra para cada cadena obtenida a partir de las 8 actividades conseguidas por el Random, que actividades han sido completadas según el criterio de cumplimiento de las notificaciones claves establecidas para cada una de ellas.



Alumno: Maria Georgieva Caneva

Tutor: Jesus Alonso Villadangos

TRATAMIENTO DE INFORMACIÓN A TRAVÉS DE PDA



INTRODUCCIÓN

- Problema:
 - Detectar procesos realizados por el operario a partir de notificaciones recogidas que componen actividades
 - Resolver de forma automática

INTRODUCCIÓN

- Empresa → Actividades y notificaciones que las componen identificadas
- Trabajador → realiza actividades previstas equipado con PDA
 - Recogida automática de información durante desarrollo

FUNCIONALIDAD DE LA HERRAMIENTA

- Recogida de datos
- Simular sincronización de la PDA
- Insertar notificaciones en BD
- Detectar por la empresa con información almacenada en PDA las actividades realizadas por el operador.

FUNCIONALIDAD DE LA HERRAMIENTA

- Métodos de estudio :
 - Distancia y % de cambios Levenshtein
 - % Cumplimiento de notificaciones claves

REQUISITOS

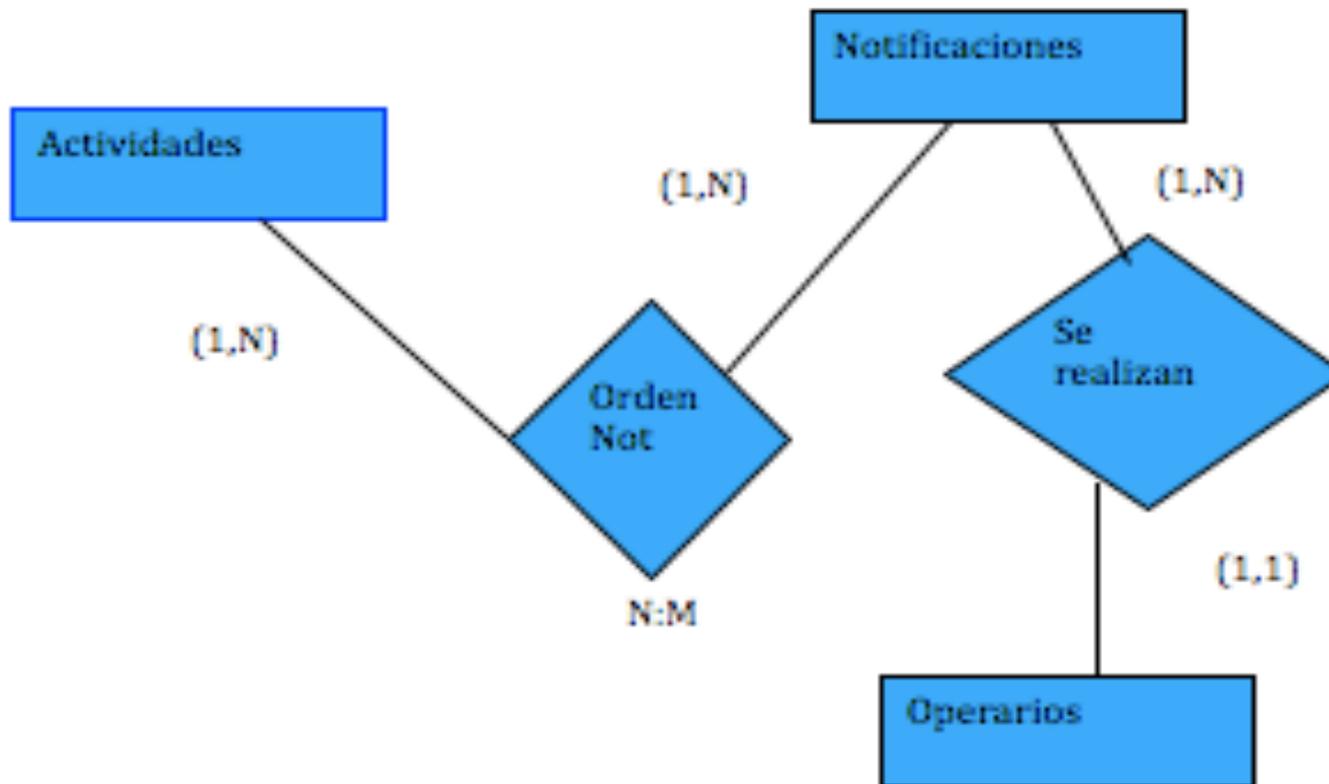
- Analizar el caso de fallos en la recepción de notificaciones
- Asegurar ejecución del proceso → Existencia o no de notificaciones relevantes
- Resultado: grado de realización de actividad propuesta como resultado tras análisis de notificaciones.

HERRAMIENTAS

- Microsoft Visual Studio 2008 Express Edition
 - Microsoft Visual C#
 - SQL Server Compact 3.5



Diagrama Entidad-Relación



TEORIAS CÁLCULO DE DISTANCIAS ENTRE CADENAS

- Distancia de Levenshtein
 - Número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra
 - Operación: inserción, eliminación o sustitución de un carácter

RESULTADOS

```
*****Conexion realizada con exito*****
```

```
*****obteniendo datos de la Base de datos*****
```

```
*****vemos las notificaciones que componen cada actividad*****
```

```
*****las actividades que hay*****
```

```
seleccionando filas de la tabla actividades
```

```
*****imprimiendo actividad sacada por el random*****
```

```
A1
```

```
las notificaciones requeridas son:
```

```
N1,N2,N3,N4,N5,N6,N7,N8,N9,
```

RESULTADOS

```
*****imprimiendo actividad sacada por el random*****  
  
A1  
  
las notificaciones requeridas son:  
N1,N2,N3,N4,N5,N6,N7,N8,N9,  
  
-----  
*****asignando a cada notificacion de las del operario un numero aleato  
rio*****  
  
-----  
N1- 0,22  
N2- 0  
N3- 0,48  
N4- 0,57  
N5- 0,14  
N6- 0,27  
N7- 0,67  
N8- 0,73  
N9- 0,66  
  
-----  
*****las notificaciones con una probabilidad mayor/igual que el error impuest  
o son:  
  
-----  
N1- 0,22  
N3- 0,48  
N4- 0,57  
N6- 0,27  
N7- 0,67  
N8- 0,73  
N9- 0,66
```

RESULTADOS

```
-----  
*****acumulando el tiempo de las notificaciones que verdaderamente realizadas:  
-----
```

```
N1- 3,78  
N2- 1,59  
N3- 1,33  
N4- 8,31  
N5- 2,1  
N6- 2,11  
N7- 8,18  
N8- 6,76  
N9- 5,3
```

```
para la realizacion de la notificacion :N1 se ha requerido un tiempo de:3,78  
para la realizacion de la notificacion :N2 se ha requerido un tiempo de:5,37  
para la realizacion de la notificacion :N3 se ha requerido un tiempo de:6,7  
para la realizacion de la notificacion :N4 se ha requerido un tiempo de:15,01  
para la realizacion de la notificacion :N5 se ha requerido un tiempo de:17,11  
para la realizacion de la notificacion :N6 se ha requerido un tiempo de:19,22  
para la realizacion de la notificacion :N7 se ha requerido un tiempo de:27,4  
para la realizacion de la notificacion :N8 se ha requerido un tiempo de:34,16  
para la realizacion de la notificacion :N9 se ha requerido un tiempo de:39,46
```

```
el tiempo total dedicado desde 1º hasta ultima notificacion es :  
35,68
```

RESULTADOS

```
-----  
***creando cadena para las notificaciones que se deberian hacer ****:
```

```
notificaciones que deberian hacerse para la actividad dada:  
N1,N2,N3,N4,N5,N6,N7,N8,N9,
```

```
-----  
***creando cadena para las notificaciones que se han recibido correctamnete****
```

```
-----  
cadena de notificaciones realmente hechas:  
N1,N3,N4,N6,N7,N8,N9,
```

RESULTADOS

parte del estudio y conclusiones de los datos obtenidos

La distancia con A1 es 6 y su porcentaje de cambios es 0,222222222222222

La distancia con A2 es 8 y su porcentaje de cambios es 0,380952380952381

La distancia con A3 es 16 y su porcentaje de cambios es 0,516129032258065

La distancia con A4 es 10 y su porcentaje de cambios es 0,476190476190476

La distancia con A5 es 3 y su porcentaje de cambios es 0,142857142857143

La distancia con A6 es 10 y su porcentaje de cambios es 0,476190476190476

La distancia con A7 es 7 y su porcentaje de cambios es 0,333333333333333

La distancia con A8 es 14 y su porcentaje de cambios es 0,666666666666667

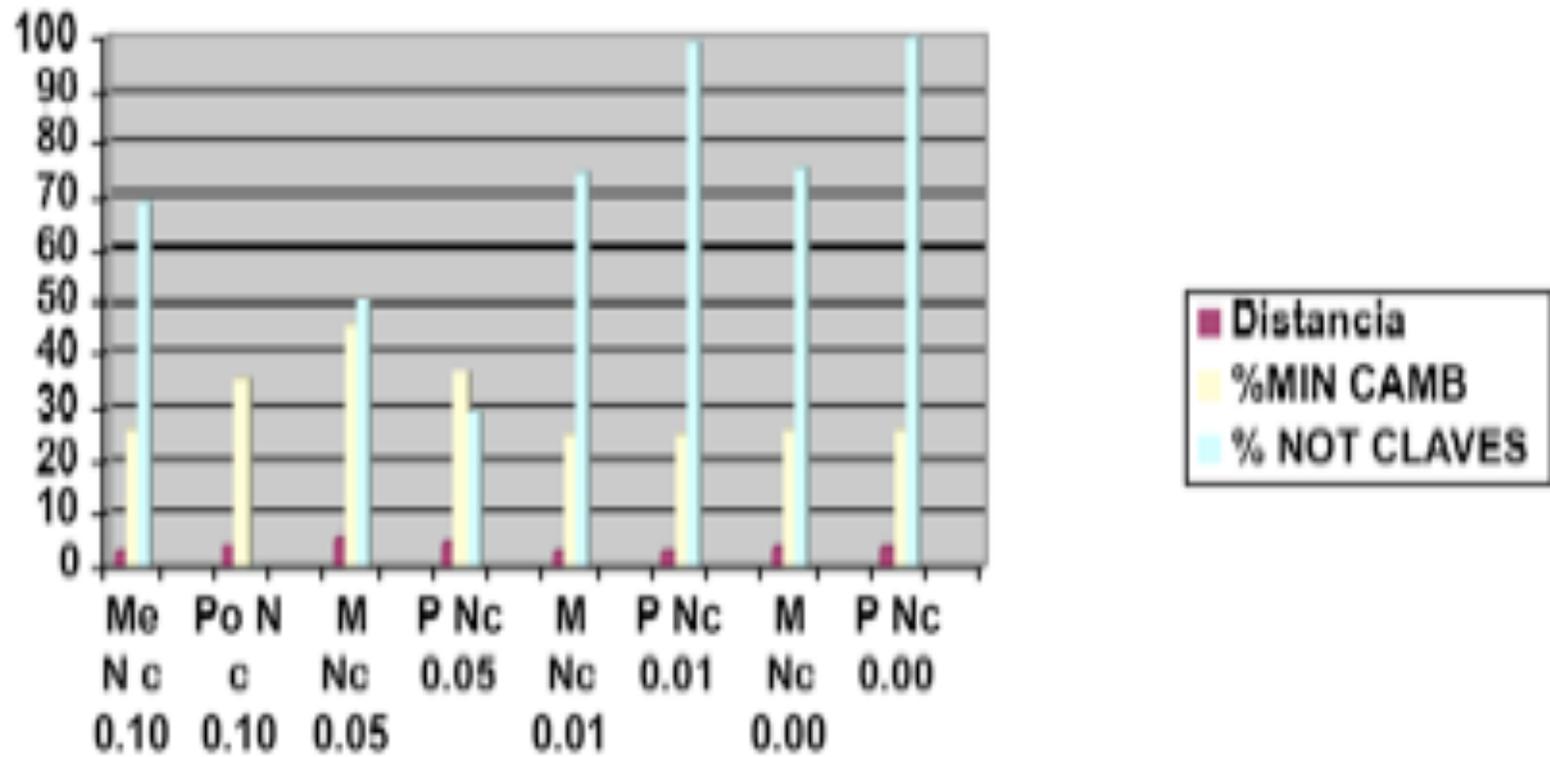
RESULTADOS

```
vemos las notificaciones claves de cada actividad
IDACT      ID
A1         N1
A1         N3
A1         N5
A1         N7
A1         N8
A1         N9
A2         N5
A2         N3
A2         N4
A2         N9
A3         N8
A3         N9
A3         N6
A3         N1
A3         N5
A3         N3
A3         N10
A3         N4
A4         N2
A5         N2
A5         N8
A5         N9
A6         N7
A6         N6
A6         N5
A6         N4
A7         N1
A7         N7
A7         N9
A8         N8
A8         N3
```

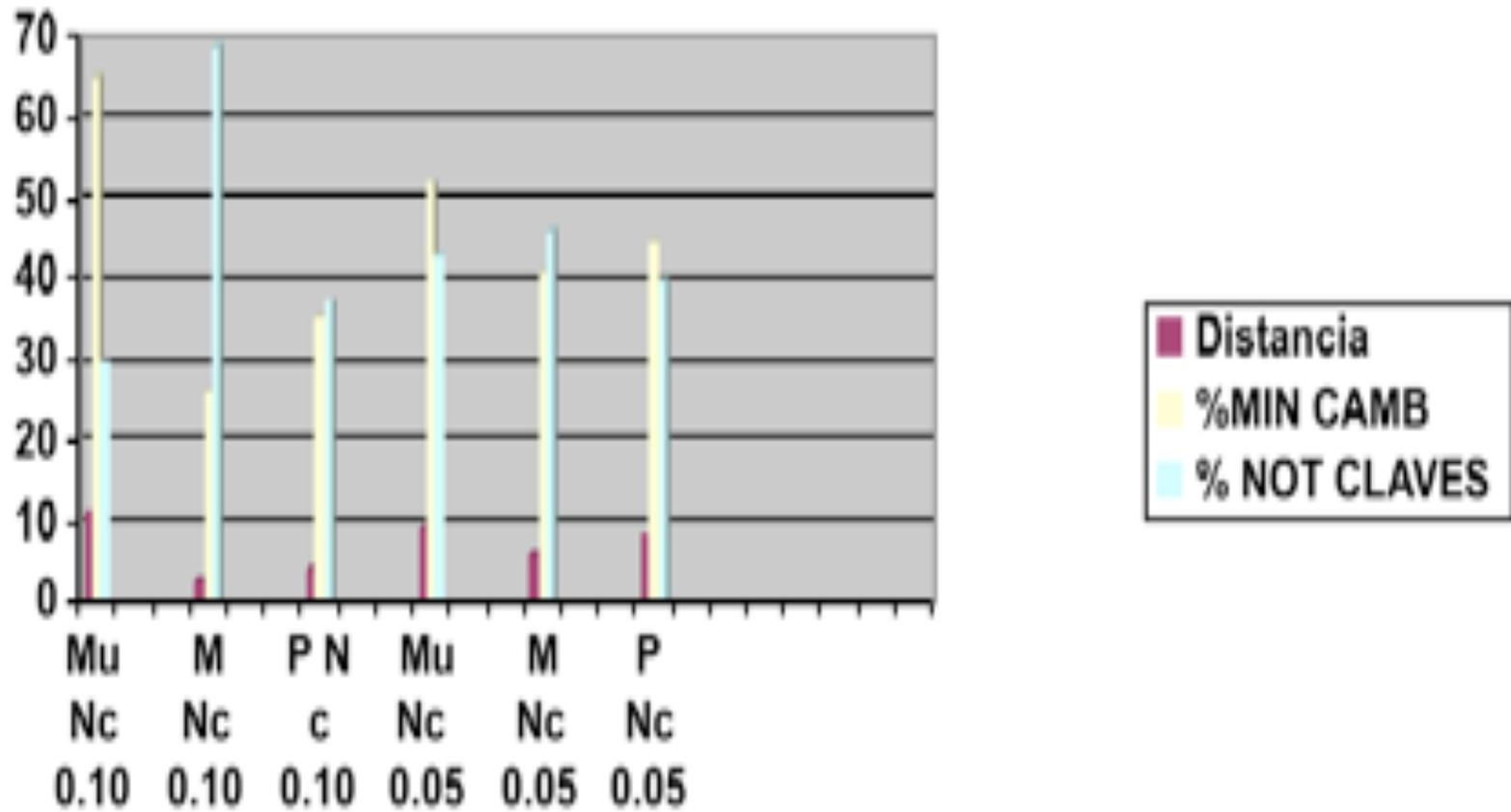

ESTUDIO DE RESULTADOS

- Longitud conjunto de notificaciones realizadas por el operario
- Error para la simulación del correcto funcionamiento del dispositivo
- Notificaciones claves, de gran importancia para la correcta realización de una actividad.

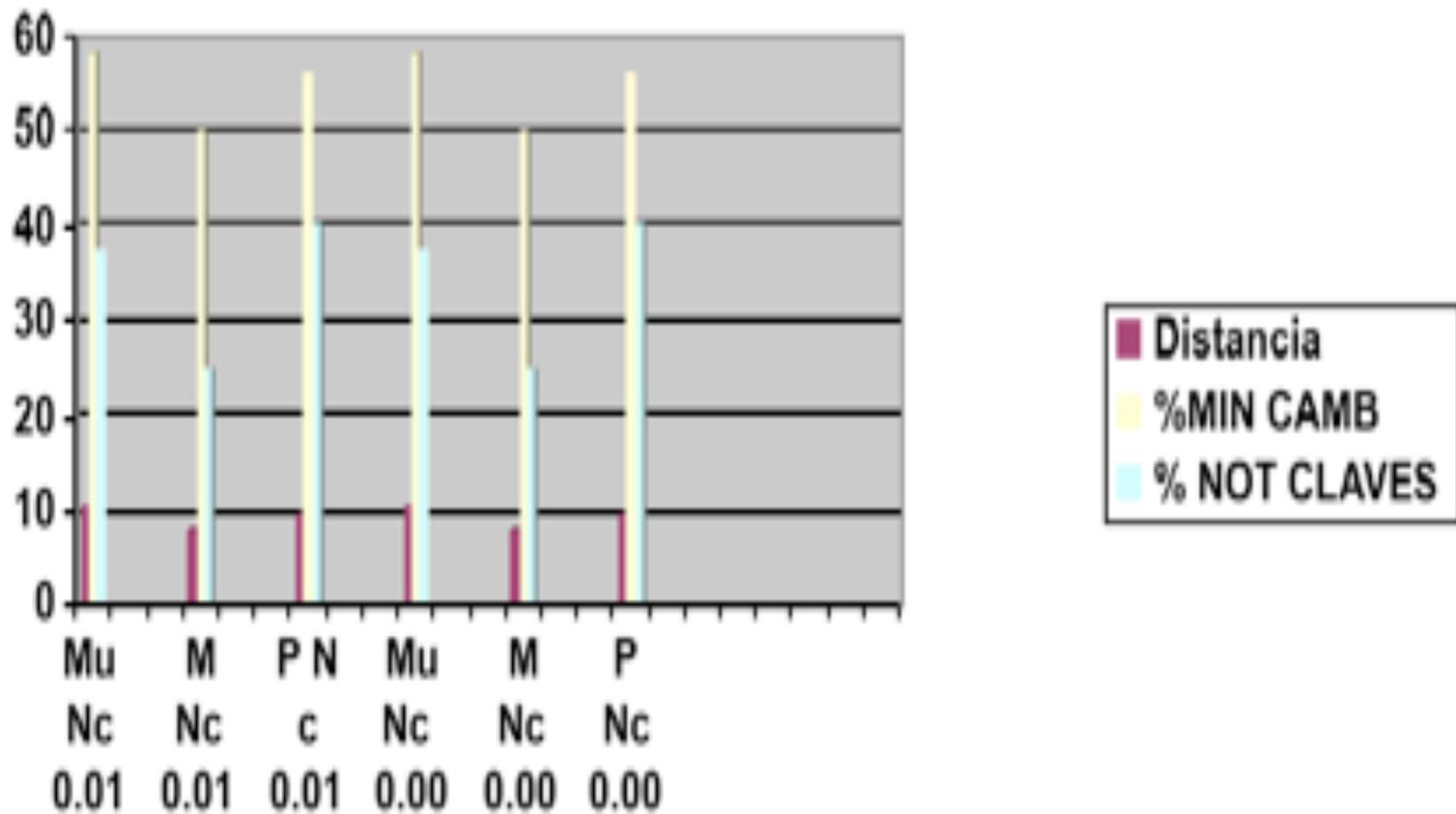
Corta - Corta



Corta - Media



Corta - Larga



CONCLUSIONES

- Mayor límite correcto recibimiento datos:
 - menor % cumplimiento de notificaciones claves
 - mayor distancia y mayor % cambios
 - menos numero de actividades detectadas a partir del trabajo realizado por el operario.

CONCLUSIONES

- Si actividades no tienen notificaciones comunes , no hay manera de poder detectar unas partiendo de las otras.
- Muy probable, con mismo conjunto de notificaciones de partida → se estén realizando distintas actividades, según el error

CONCLUSIONES

- Realización de aplicación comercial
- Análisis, diseño e implementación de aplicación compleja
- Aprendizaje de plataforma Visual Studio

LINEAS FUTURAS

- Utilizar otro tipo de metricas:
 - Distancia de Hamming
 - Distancia de Damerau –Levenshtein
- Utilizar diferentes metricas:
 - Votación
 - Ajustar mejor seleccion de actividad

LINEAS FUTURAS

- Posibilidad de detectar fallos de sensores
- Posibilidad de utilizar más actividades para realizar el estudio

RUEGOS Y PREGUNTAS

APLICACIÓN

GRACIAS POR SU ATENCIÓN