Nombre del equipo

Código del estudiante

Descripción

Manual de diseño, creación y uso   
del sistema de registro de ingresos y egresos

INGRESOS VS EGRESOS

Sistema de Registro

**SISTEMA DE REGISTRO DE INGRESOS VS EGRESOS**

**Julian David Giraldo.**

**Oscar Hernando Mesa G.**

**TOPICOS AVANZADOS DE INGENIERIA DE SOFTWARE**

**INTELIGENCIA DE NEGOCIOS**

**DOCENTE**

**JOSÉ MAURICIO JARAMILLO GOMEZ**

**ESPECIALISTA EN GERENCIA DE PROYECTOS**

**EN INTELIGENCIA DE NEGOCIOS**

**POLITECNICO COLOMBIANO JAIME ISAZA CADAVID**

**FACULTAD DE INGENIERIA INFORMATICA**

**SEDE MEDELLIN**

**2014**

# INTRODUCCIÓN

**En nuestra actualidad, nuestra sociedad le demanda a la ingeniería de software aplicativos del uso cotidiano que permitan solucionar problemas de la vida cotidiana. No es una sorpresa para nadie, que año tras año se ha ido en una mejora continua en el aspecto del desarrollo del software, donde, no solo se espera que una aplicación sea simplemente una herramienta que nos permita almacenar una serie de datos e información en cualquier motor de almacenamiento, si no que nos permita organizar y operar sobre esta información con el fin de tomar decisión que aporten ganancia a toda las actividades cotidianas.**

**El proyecto que se desarrollo en este trabajo tiene como finalidad abstraer la información de ingresos y egresos de cualquier persona y con base a esto se pueda operar esta información y se puedan encontrar algunos patrones que puedan evidenciar algunas características en la forma como se esta captando y gastando el dinero. Con base a lo anterior se espera que una persona este en la facultad de tomar decisiones que le permitan mejorar su juicio en el momento que invierta su dinero.**

**Invitamos al lector a que día a día se interese más en los temas que tienen como objetivo mejorar la calidad de vida del individuo, como lo pueden ser las herramientas de software que apunten a que podamos tomar decisiones que afecten de un modo positivo nuestras vidas.**

CONTENIDO

[INTRODUCCIÓN 2](#_Toc381520117)

[REQUERIMIENTOS 4](#_Toc381520118)

[1. REQUERIMIENTO 1 4](#_Toc381520119)

[1.1. Diseño de modelo entidad relación 4](#_Toc381520120)

[1.1.1. Modelo físico 4](#_Toc381520121)

[1.1.2. Modelo lógico 4](#_Toc381520122)

[1.2. Scripts de creación del modelo 4](#_Toc381520123)

[1.3. Diccionario de datos 4](#_Toc381520124)

[1.4. Scripts de inserción de datos 4](#_Toc381520125)

[2. REQUERIMIENTO 2 4](#_Toc381520126)

[3. REQUERIMIENTO 3 4](#_Toc381520127)

[4. REQUERIMIENTO 4 5](#_Toc381520128)

[5. REQUERIMIENTO 5 5](#_Toc381520129)

[BASE DE DATOS 6](#_Toc381520130)

[MODELO ENTIDAD RELACIÓN 6](#_Toc381520131)

[DIAGRAMA 6](#_Toc381520132)

[SCRIPTS DE CREACIÓN DE ESQUEMA 6](#_Toc381520133)

[DICCIONARIO DE DATOS 6](#_Toc381520134)

[SCRIPTS INSERT DE DATOS 6](#_Toc381520135)

[DISEÑO SISTEMA 7](#_Toc381520136)

[DISEÑO DE PANTALLAS 7](#_Toc381520137)

[MANUAL DE USO 7](#_Toc381520138)

[CONCLUSIONES 8](#_Toc381520139)

[ANEXOS 9](#_Toc381520140)

# REQUERIMIENTOS

## REQUERIMIENTO 1

Se requiere crear un sistema que permita registrar los ingresos (entradas de dinero) y los egresos (salidas de dinero) de las personas.

Este sistema se solicita con el fin de poder controlar las finanzas de un equipo de trabajo.

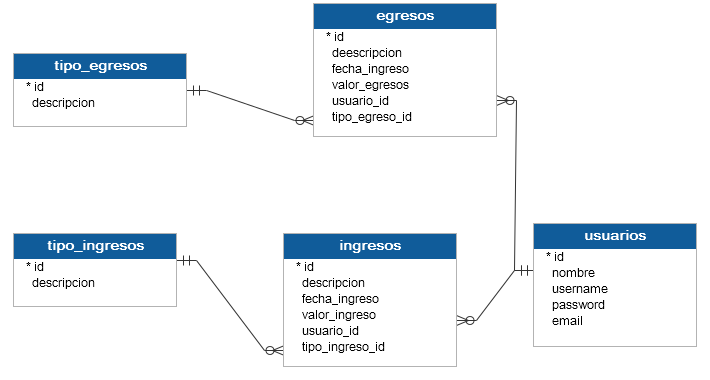
Para esto se deben almacenar la información en una base de datos (Oracle, sql server, my sql, postgres, entre otros).

Debe estar normalizada aplicando como mínimo hasta la cuarta forma normal.

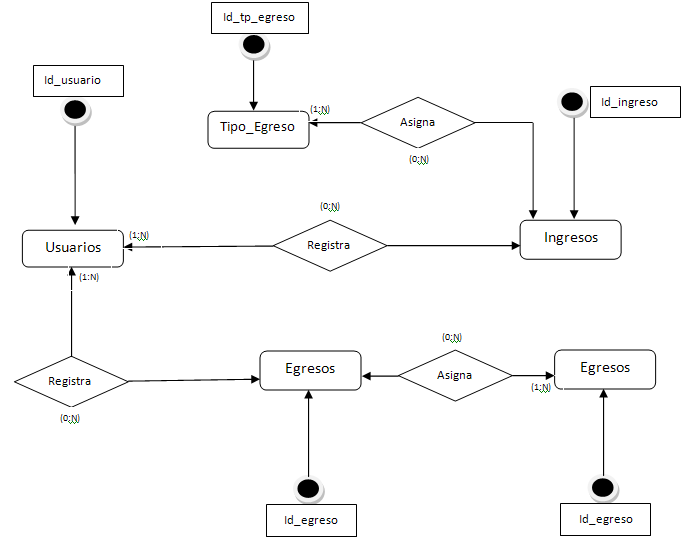
Esta base de datos debe tener lo siguiente:

## Diseño de modelo entidad relación

## Modelo físico



## Modelo lógico



## Scripts de creación del modelo

CREATE DATABASE IngresoEgresos;

USE IngresoEgresos;

DROP TABLE IF EXISTS `egresos`;

DROP TABLE IF EXISTS `ingresos`;

DROP TABLE IF EXISTS `tipo\_egresos`;

DROP TABLE IF EXISTS `tipo\_ingresos`;

DROP TABLE IF EXISTS `usuarios`;

DROP TABLE IF EXISTS `tipo\_egresos`;

CREATE TABLE IF NOT EXISTS `tipo\_egresos` (

`id` int(5) NOT NULL AUTO\_INCREMENT,

`descripcion` varchar(100) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=1 ;

DROP TABLE IF EXISTS `tipo\_ingresos`;

CREATE TABLE IF NOT EXISTS `tipo\_ingresos` (

`id` int(5) NOT NULL AUTO\_INCREMENT,

`descripcion` varchar(100) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `egresos` (

`id` int(5) NOT NULL AUTO\_INCREMENT,

`deescripcion` varchar(300) COLLATE utf8\_unicode\_ci NOT NULL,

`fecha\_egreso` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`valor\_egresos` double NOT NULL,

`usuario\_id` int(5) NOT NULL,

`tipo\_egreso\_id` int(5) NOT NULL,

PRIMARY KEY (`id`),

KEY(`usuario\_id`),

KEY(`tipo\_egreso\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=1 ;

DROP TABLE IF EXISTS `ingresos`;

CREATE TABLE IF NOT EXISTS `ingresos` (

`id` int(5) NOT NULL AUTO\_INCREMENT,

`descripcion` varchar(300) COLLATE utf8\_unicode\_ci NOT NULL,

`fecha\_ingreso` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`valor\_ingreso` double NOT NULL,

`usuario\_id` int(5) NOT NULL,

`tipo\_ingreso\_id` int(5) NOT NULL,

PRIMARY KEY (`id`),

KEY(`usuario\_id`),

KEY(`tipo\_ingreso\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=1 ;

DROP TABLE IF EXISTS `usuarios`;

CREATE TABLE IF NOT EXISTS `usuarios` (

`id` int(5) NOT NULL AUTO\_INCREMENT,

`nombre` varchar(100) COLLATE utf8\_unicode\_ci NOT NULL,

`username` varchar(60) COLLATE utf8\_unicode\_ci NOT NULL,

`password` varchar(50) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`id`),

UNIQUE(`email`),

UNIQUE(`username`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=1 ;

ALTER TABLE `egresos`

ADD CONSTRAINT `egresos\_ibfk\_2` FOREIGN KEY (`tipo\_egreso\_id`) REFERENCES `tipo\_egresos` (`id`),

ADD CONSTRAINT `egresos\_ibfk\_1` FOREIGN KEY (`usuario\_id`) REFERENCES `usuarios` (`id`);

ALTER TABLE `ingresos`

ADD CONSTRAINT `ingresos\_ibfk\_2` FOREIGN KEY (`tipo\_ingreso\_id`) REFERENCES `tipo\_ingresos` (`id`),

ADD CONSTRAINT `ingresos\_ibfk\_1` FOREIGN KEY (`usuario\_id`) REFERENCES `usuarios` (`id`);

## Diccionario de datos

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Restricciones | Tipo de campo | Observaciones. |
| Tabla usuarios | | | |
| id | PK | INT(5) | Llave primario de un usuario, con el que se podrá identificar de otro, el cual se genera automáticamente.. |
| nombre | Requerido | VARCHAR(100) | Nombre completo de la persona, en el cual podrá incluir nombre y apellidos o solo su nombre. |
| username | Requerido, único | VARCHAR(60) | Nombre de usuario, con el cual ingresara a la aplicación y debe ser único por persona. |
| password | Requerido | VARCHAR(50) | Contraseña del usuario, el sistema lo encriptará y almacenara en SHA1. |
| email | Requerido, único | VARCHAR(80) | Correo del usuario. |
| Tabla tipo egresos | | | |
| id | PK | INT(5) | Llave primaria del engreso, el cual se genera automáticamente.. |
| descripcion | Requerido | VARCHAR(100) | Descripción del egreso. |
| Tabla tipo ingresos | | | |
| id | PK | INT(5) | Llave primaria del ingreso, el cual se genera automáticamente.. |
| descripcion | Requerido | VARCHAR(100) | Descripción del egreso. |
| Tabla ingresos | | | |
| Id | PK | INT(5) | Identificador único de los ingresos, el cual se genera automáticamente. |
| descripcion |  | VARCHAR(300) | Una breve descripción del nuevo ingreso que capto el usuario, el cual no será obligatorio ingresar. |
| fecha\_ingreso | Requerido | timestamp | Fecha en la cual capto este nuevo ingreso, por defecto se toma la fecha actual. |
| valor\_ingresos | Requerido | DOUBLE | Valor del ingreso en pesos colombianos. |
| usuario\_id | FK, requerido | INT(5) | Identificador del usuario que está registrando el ingreso. Para realizar esta acción se debe estar logueado, se toma como identificador el que se encuentra en sesión. |
| tipo\_ingreso\_id | FK, requerido | INT(5) | Identificador del tipo de ingreso que se está registrando. |
| Tabla egresos | | | |
| Id | PK | INT(5) |  |
| descripcion |  | VARCHAR(300) | Descripción del egreso, el cual no es obligatorio diligenciar. |
| fecha\_egreso | Requerido | timestamp | Fecha en la cual se realizo el gasto del dinero. |
| valor\_egresos | Requerido | DOUBLE | Valor del egreso que se va a registrar, validando que el que se esté registrando no supere el saldo actual. |
| usuario\_id | FK, requerido | INT(5) | Identificador del usuario que está registrando el egreso. Para realizar esta acción se debe estar logueado, se toma como identificador el que se encuentra en sesión. |
| tipo\_egreso\_id | FK, requerido | INT(5) | Identificador del tipo de ingreso que se está registrando. |

## Scripts de inserción de datos

## REQUERIMIENTO 2

Para el registro de los datos se debe controlar las transacciones a que persona pertenecen por lo tanto es importante y obligatorio seleccionar la persona. Se puede generar de dos maneras, la primera utilizando una pantalla de registro de usuario, la segunda obligando a seleccionar la persona en el momento de insertar la transacción.

## REQUERIMIENTO 3

Los ingresos y los egresos deben estar tipificados con el fin de poder clasificarlos y seleccionarlos en el momento del registro de la transacción.

A continuación unos ejemplos de tipificaciones:

|  |  |
| --- | --- |
| **TIPOS DE INGRESOS** | **TIPOS DE EGRESOS** |
| Salario | Alimentación |
| Prestación de servicios | Transporte |
| Mesada | Entretenimiento |
| Arriendo | Servicios públicos |
| Tarjeta de crédito | Prestamos |

Esta lista anterior son solo alguno de los tipos.

## REQUERIMIENTO 4

Los formularios deben ser amigables y fáciles de usar para el registro de los datos, por lo tanto el manual debe indicar completamente la manera de usar el sistema.

## REQUERIMIENTO 5

Los días que no se tengan ingresos o egresos no deben ser registrados en la base de datos.

## REQUERIMIENTO 6

Los ingresos y los egresos no pueden ser negativos, esto debe controlarse por el sistema.

# BASE DE DATOS

## MODELO ENTIDAD RELACIÓN

### DIAGRAMA

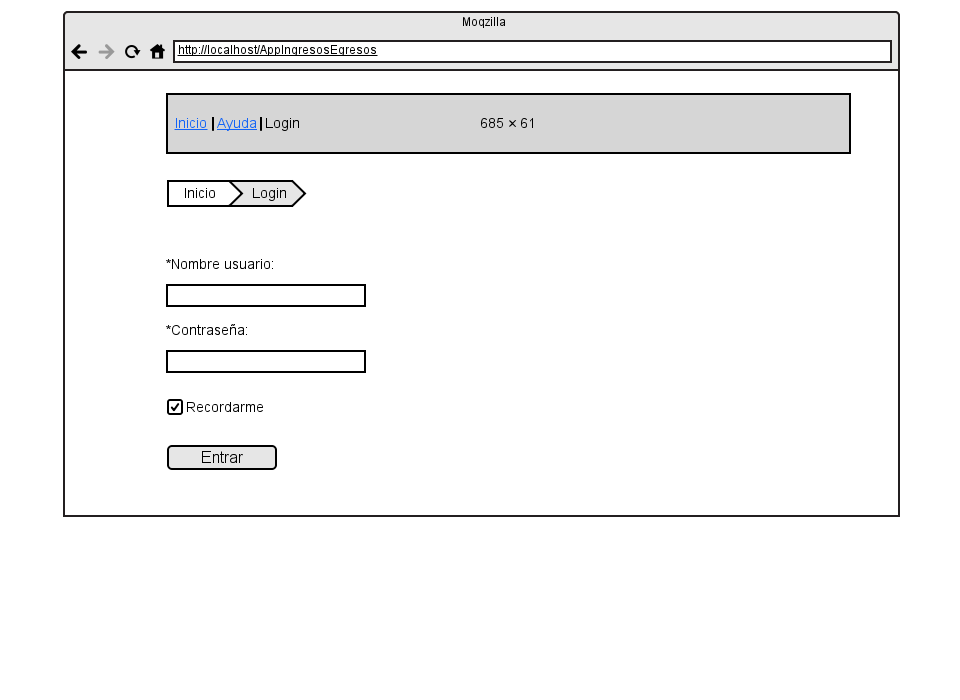
### SCRIPTS DE CREACIÓN DE ESQUEMA

### DICCIONARIO DE DATOS

### SCRIPTS INSERT DE DATOS

# DISEÑO SISTEMA

## DISEÑO DE PANTALLAS



## MANUAL DE USO

# CONCLUSIONES

* Se logro realizar un análisis a un problema de la vida cotidiana, desarrollando una herramienta con la cual se interactuó y se aprovecho para registrar una información con la cual se espera sacar más información en el futuro.
* Se lograron emplear algunas herramientas que sirvieron como soporte para el trabajo en equipo y facilitaron el desarrollo del proyecto.

# ANEXOS