

# Sistema de donaciones para organización Obras del Espíritu Santo

Kevin Céspedes Monge, Mario Molina Medal,

José Retana Pereira, Felipe Villalobos Ulate

*Ingeniería en Sistemas de*

*computación, U. Fidelitas,*

*Heredia/San José, Costa Rica.*

[kcespedes00989@ufide.ac.cr](mailto:kcespedes00989@ufide.ac.cr)

[mmolina00702@ufide.ac.cr](mailto:mmolina00702@ufide.ac.cr)

[jretana80675@ufide.ac.cr](mailto:jretana80675@ufide.ac.cr)

[fvillalobos50829@ufide.ac.cr](mailto:fvillalobos50829@ufide.ac.cr)

**Abstract:** As we all know, today the need for databases has increased over time due to a high growth of small and medium-sized companies and institutions that seek to develop, so they choose to acquire their own to benefit and ease of data management. The following project seeks to precisely solve this need, thus developing a system with a database that can facilitate the administration of these within an institution that specializes in donations.

**Keywords:** Database Languages, C#, HTML, Donations, SQL, SQLdeveloper, Visual Studio.

**Resumen:** Como bien sabemos, hoy en día la necesidad de las bases de datos se ha incrementado con el paso del tiempo debido a un alto crecimiento de pequeñas y medianas empresas e instituciones que buscan desarrollarse, por lo que estas optan por adquirir una propia para beneficio y facilidad de manejo de datos. El siguiente proyecto busca solventar justamente esta necesidad, desarrollando así un sistema con una base de datos que pueda facilitar la administración de estos dentro de una institución que se especializa en donaciones.

**Palabras clave:** Lenguajes de Bases de datos, C#, HTML, Donaciones, SQL, SQLdeveloper.

## I. INTRODUCCIÓN

El presente proyecto busca la realización de un sistema que le permita a los empleados o usuarios finales de la institución en la que se basa este sistema de donaciones, el poder manejar los datos de esta. Con esto nos referimos a que la aplicación se basará en un CRUD donde los usuarios permitidos puedan interactuar con estos datos con las siguientes acciones básicas, la cuales serían; la creación de los datos y su lectura, además de la actualización y eliminación de estos si lo llegan a requerir. Todo esto con el fin de crear un sistema que permita el correcto almacenamiento de los datos no solo de las donaciones, si no también de quienes hacen las donaciones y quienes la recibirán, además de otra información sensible que necesite ser almacenada.

Para la implementación de dicho sistema justamente se ve la necesidad de la creación de una base de datos para un sistema de donaciones la cual será desarrollada en el entorno de SQLdeveloper, también se busca implementar la arquitectura basada en MVC (Modelo Vista Controlador por sus siglas) donde, para términos de vista, se estará utilizando HTML como lenguaje de etiqueta y la ayuda de Bootstrap para una mejor interfaz. En termino de Backend o del lado modelo y controlado se busca la utilización de java, la justificación del por qué escogimos este lenguajes es debido a la experiencia de los involucrados en el progreso de este proyecto, ya que esta es mayor en este lenguaje sobre cualquier otro, por lo que creemos que es una mejor y más fiable opción a la hora de desarrollar este sistema que nos planteamos, además que este mismo cuenta con una gran variedad de ventajas que nos pueden ofrecer una mejor y más cómoda experiencia a la hora de codificar dicho sistema, a su vez trayendo facilidad a todos los integrantes por su enorme versatilidad a la hora de desarrollar aplicaciones web, hablando en términos tanto de Frontend como de Backend.

Para términos de base de datos nos estaremos basando además en un Estándar de diseño de estas. Y esto es importante debido a que; [1] Un estándar de datos es la definición clara y completa de la expectativa de los elementos descritos en un conjunto de datos: como cuando en una hoja de Excel tenemos la primera fila con la serie de nombres de columnas con una descripción de los campos a recibir y su contenido. Por el lado técnico, esto permite tener claridad sobre la información disponible en una solución, y por el lado práctico también permite tener una idea de los objetivos de información que posee esta

herramienta.

Por lo que creemos que, para este proyecto, el definir un Estándar de diseño para la base de datos es de importancia, por lo que estaremos desarrollando uno en el mismo.

## II. OBJETIVO GENERAL Y ESPECIFICO.

### A. Objetivo General.

- Desarrollar un sistema para una base de datos que facilite el manejo de estos en una institución que se especializa en donaciones.

### B. Objetivos específicos.

- Implementación de una base de datos con las tablas necesarias para la interacción de esta con el sistema mediante el lenguaje de SQL en el entorno de SQLdeveloper.
- Definir los requerimientos funcionales a través de un CRUD que se basará en las tablas o la información que será administrada a la base de datos.
- Crear mediante el lenguaje de C# todo el funcionamiento o Backend que tendrá dicha sistema.
- Diseñar una interfaz agradable para los usuarios finales de este sistema con ayuda del lenguaje de etiqueta; HTML y el framework; Bootstrap.

## III. DESARROLLO DE CONTENIDOS

### A. Estándar de diseño de la base de datos.

Definición del estándar de diseño para este proyecto:

- En el caso del nombre de la base de datos iniciara con el prefijo “**DB**” para hacer diferencia de que es el nombre de la propia db y no de una tabla, entidad, etc.
- Para las tablas se utilizará el prefijo “**TB**” en mayúscula seguido de la primera letra del nombre de la tabla en mayúscula y las demás en minúscula.
- Los atributos iniciaran con el prefijo “**AT**” seguido del nombre del atributo, iniciando con la primera letra de la palabra en mayúscula.
- Las tablespaces iniciaran con el prefijo “**TBS**” si no son temporales y si son tablespaces temporales se usará “**TBST**” después del prefijo se inicia el nombre de la tablespace con la primera letra de la palabra en mayúscula.
- Los índices llevarán el prefijo “**IND**” al inicio y seguidamente se iniciará con el nombre del índice con la primera letra en mayúscula.
- Si una palabra tiene “**n**” se sustituye esa letra por una “**n**”.
- Los nombres deben escribirse en plural como ejemplos tenemos productos.

### B. Base de datos.

La idea de la base de datos se basa en un sistema de donaciones por lo que vemos necesario la creación de las tablas de las siguientes entidades;

Usuarios: con esto podremos definir la importancia o definir una jerarquía en quienes están almacenados en el sistema y podremos tener un control de estos en

todo momento. Donaciones: Quizá la tabla más importante, ya que en esta se encuentra la mayoría de la información más destacable de la base de datos, Sedes: con esta tabla podremos identificar las diferentes sedes o establecimientos que tiene esta institución. Roles: En esta tabla se almacena el tipo de rol que puede tener un usuario, ya sea Administrador o Empleado y de esta manera poder asignarle los permisos en el sistema. Correos: En esta tabla se almacena la información referente a los correos de los usuarios. Teléfonos: En esta tabla se almacenan los teléfonos de los usuarios junto con su referente información. Usernames: En esta tabla se almacenan los usernames con su respectivo rol y el dueño de este haciendo referencia a la tabla usuarios. Tipo Donación: Aquí se almacena el tipo de donación que es posible realizar, ya sea en Dinero o en Especies. Lo que nos deja con la estructura del diagrama entidad relación adjuntado en el anexo 1.

#### C. Requerimientos Funcionales.

Para tener una base y tener una idea de lo que se busca lograr en este proyecto, se buscar la implementación de un CRUD para el manejo de la información de la base de datos, por lo que buscamos identificar los requerimientos funcionales que serán necesarios.

- El sistema deberá permitir insertar, leer, actualizar USUARIOS.
- El sistema deberá permitir insertar, leer, actualizar DONACIONES.
- El sistema deberá permitir insertar, leer, actualizar SEDES.

#### D. Requerimientos no funcionales.

- El sistema debe contar con una interfaz gráfica amigable para el usuario.
- El sistema debe ser fácil de usar para el usuario.
- El sistema debe dejar logearse fácilmente a los usuarios

#### E. Siguiendo contenidos

### IV. CONCLUSIONES

Se concluye de la presente investigación que muchas veces de la manera que se piensan las cosas en un principio al final siempre terminan variando por mejores ideas adaptadas al conocimiento que se adquiere con el paso del tiempo y con las lecciones aprendidas gracias a la investigación y la materia vista en clase. En el presente caso se dio un cambio en la estructura de las tablas en donde se tenía estipulado utilizar tablas para los donantes y donatarios, lo que se reemplazó por una sola tabla llamada donaciones la cual se encarga de almacenar esos datos así no se tenían tablas que fueran innecesarias y que fueran a dar conflicto conforme el avance del proyecto. Eso se puede ver en los requerimientos que se fijaron y la dirección que se le dio para que la información se manejara de manera más óptima y eficiente.

Además se obtuvo bastante conocimiento acerca de lo que es el lenguaje PL/SQL a la hora de realizar los procesos necesarios en la base, junto con un amplio conocimiento obtenido en el lenguaje de C# el cual fue utilizado como herramienta para mostrar los datos de la base producto de los cambios que se fueron dando como se mencionaba anteriormente ya que en un principio se estipuló la utilización de Java como el medio de muestra de los datos pero gracias a conocimientos aprendidos en el proceso se optó por C# utilizando el programa Visual Studio Community como una alternativa más adecuada para nuestro proyecto.

### V. RECOMENDACIONES

Algunas recomendaciones importantes serían tener una buena comunicación entre los integrantes del grupo ya que eso fue un aspecto bastante importante de el presente trabajo, Además de la buena elección de un lenguaje de programación el cual sea fácil de conectarse con la base de datos ya que eso en nuestra experiencia hizo mucha diferencia, ser bastante cuidadoso con la creación de tablas y atributos, para que haya un buen modelo relacional el cual no de conflictos a la hora de realizar procedimientos o consultas necesarias para el buen funcionamiento del sistema, además de la utilización de procedimientos PL/SQL para facilitar la funcionalidad y manejo de información, junto con el uso de INNER JOINS para lograr unir información necesaria en una tabla la cual utiliza una relación con otra.

## V.BIBLIOGRAFIA

[1] Alanis, R. "Estándares de datos". Cívica Digital. Agosto 8, 2018.  
<https://themewagon.com/themes/free-bootstrap-5-real-estate-website-template-property/>

Codingraph. Login en Oracle con C# en Visual Studio 2017. (5 de mayo de 2018). Accedido el 7 de diciembre de 2022. [Video en línea]. Disponible:  
<https://www.youtube.com/watch?v=N9uX-0CnLu4>

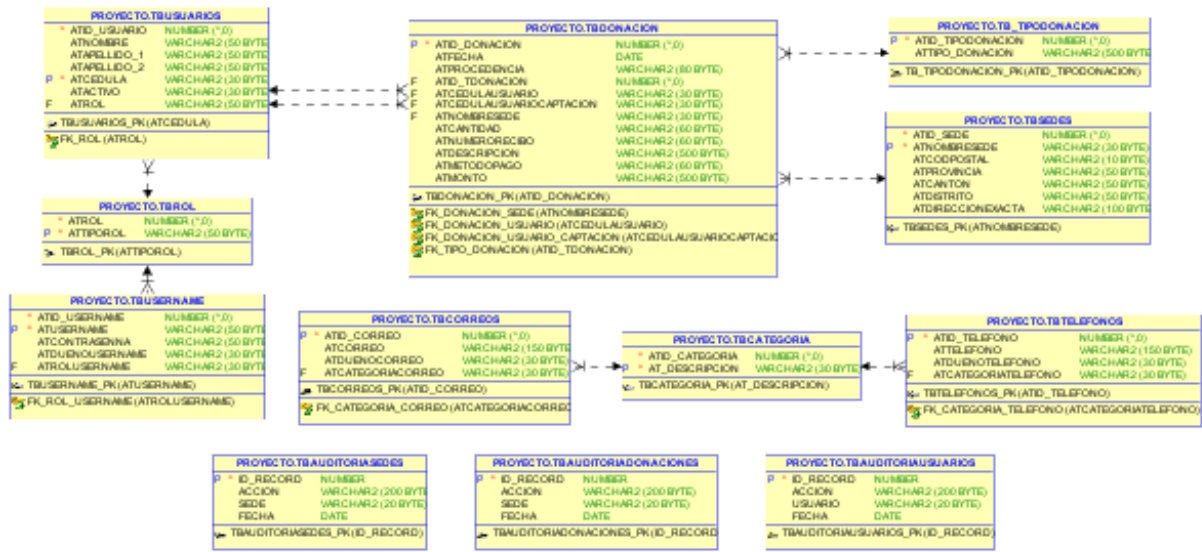
Codingraph. Diccionario de Datos en SQL Developer. (17 de marzo de 2021). Accedido el 7 de diciembre de 2022. [Video en línea]. Disponible:  
<https://www.youtube.com/watch?v=c0j5sRRGGQs>

Codingraph. Insertar datos en C# | Oracle. (15 de abril de 2017). Accedido el 7 de diciembre de 2022. [Video en línea]. Disponible:  
<https://www.youtube.com/watch?v=lv2kOudTdqA>

Consultia. Centrar la posición de la Ventana. (1 de abril de 2014). Accedido el 7 de diciembre de 2022. [Video en línea]. Disponible:  
<https://www.youtube.com/watch?v=ci0HcEN30WY>

## VI.ANEXOS

Diagrama:



Link del Repositorio de GitHub:

[Enlace de GitHub](#)

Script de la Base de Datos:

[Enlace de Script](#)

Diccionario de Datos:

[Enlace de Diccionario de Datos](#)