

 <b>CORPORACIÓN UNIVERSITARIA RAFAEL NÚÑEZ</b> <small>PARA QUE TU DESARROLLO CONTÍNE SU MARCHA</small>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>1/02/2017</b>
		<b>Página</b>	<b>1 de 28</b>

**INSTRUCTIVO:** el siguiente formato es para ser registrado en este, el Documento Consolidado de PAT Colectivo que da evidencia del ejercicio investigativo desarrollado por el colectivo (docentes y estudiantes) del nivel de formación (semestre o año). En esta consideración el documento consolidado de PAT Colectivo, debe contener:

## Portada

### 1. Ficha de Identificación

<b>Facultad: Ingeniería de sistemas</b>		<b>Colectivo Docente</b>	<b>Asignatura</b>
<b>Programa: Tecnología en sistemas de información y de software.</b>			
<b>Semestre: III y IV</b>	<b>Periodo académico: IIP-18</b>	1. Luis A Blanquicett 2. Mario Corpas 3. 4. 5. 6.	1. Física 2. matemáticas 3. programación de base de datos 4. programación III 5. Soporte de software 6. Estructura de datos I 7. programación Gráfica. 8. Electiva tecnológica II.
<b>Docente Orientador del seminario</b>			
Luis Alfredo Blanquicett Mario Corpas			
<b>Título del PAT Colectivo</b>			
Sistemas de información empresarial			
<b>Núcleo Problemático</b>			
¿Cómo desarrollar aplicaciones bajo el modelo de prototipo, aplicando el PSP cada una de sus fases de forma adecuada?			
<b>Línea de Investigación</b>			
<ul style="list-style-type: none"> <li>• Aplicación donde se evidencie el método de desarrollo (POO y procedimental)</li> <li>• Base de datos evidenciando la técnica de recolección de la información para el diseño y desarrollo.</li> <li>• Modelamiento y sistematización de los problemas relacionados con las TIC</li> <li>• Desarrollar un sistema información de acuerdo a la necesidad del sector productivo.</li> </ul>			



**FORMATO REGISTRO DOCUMENTO  
CONSOLIDADO PAT COLECTIVO**

<b>Cod-Doc</b>	<b>FT-IV-015</b>
<b>Versión</b>	<b>1</b>
<b>Fecha</b>	<b>16/01/2017</b>
<b>Página</b>	<b>2 de 28</b>

--

## 2. Informe del Proyecto Académico de Trabajo Colectivo (PAT Colectivo)

### Descripción del Problema

La producción de software es un factor fundamental para fortalecer la economía regional y proyectarse a nivel internacional. Para lograr este objetivo es necesario articular la educación tecnológica y profesional con estándares reconocidos a nivel mundial, que garanticen el buen desarrollo de un producto de software donde el porcentaje de falla y error se lo mas mínimo. Pero para que el profesional del futuro en tecnología y desarrollo de software sea exitoso, debe desarrollar unas habilidades desde los primeros semestre de su profesión y no esperarar a semestres mas avanzados para exigirle y enseñarles procesos de desarrollo de calidad

### Justificación

El PSP entrega a los ingenieros un marco de referencia de disciplina personal para mejorar su trabajo y realizarlo con alta calidad, con el propósito de ayudarlos a aprender y practicar aquellos métodos para producir software que son más efectivos para ellos. Entendiendo como principio fundamental que con un proceso de calidad los productos derivados de éste serán también de calidad.

En la mayoría de las profesiones el trabajo competente requiere el uso de prácticas establecidas, planes y procedimientos que traen orden y eficiencia a cualquier trabajo y permite a los trabajadores concentrarse en producir productos de la más alta calidad.

Precisamente el Personal Software Process o Proceso Software Personal, puede ser usado por futuros desarrolladores de software como guía para un enfoque disciplinado y estructurado en el desarrollo de software. Esta metodología permite se aplicado en los primeros semestres y experiencia de los estudiantes en las asignaturas de programación y desarrollo, el cual obtienen habilidades y

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>3 de 28</b>

competencias sobre el procedimiento adecuado del desarrollo de un producto de software para PYMES.

## 2.1. OBJETIVO GENERAL

Desarrollar un sistema electoral estudiantil, utilizando proceso personal del software PSP.

## 2.2. OBJETIVOS ESPECÍFICO

- Indagar sobre los niveles y fases que posee el PSP (proceso personal del software)
- Investigar cómo diseñar un sistema electoral estudiantil.
- Escoger una institución educativa y conocer sus requerimientos para el desarrollo e implementación de este sistema.
- Socializar los resultados obtenidos a las partes interesadas sobre las necesidades del negocio referente a las tecnologías evidenciadas.
- Comprender y apropiarse los procesos del desarrollo de software personal PSP y sus fases para el desarrollo del producto de software

## Marco Teórico o Referente Teórico

La disciplina del PSP provee un marco estructurado para desarrollar habilidades personales y métodos que se necesitarán más adelante para ir forjando al ingeniero de software. Es importante que la calidad del software desarrollado abarque hasta el más mínimo detalle, por muy pequeño que éste sea, ya que si no se hace así, puede dañar el sistema entero.

El Personal Software Process, PSP, indica a los profesionales de software cómo:

- Administrar la calidad de los proyectos de desarrollo de software, el cual le permite al estudiante planear y controlar los Tiempo de trabajo para el desarrollo del producto y entrega.
- Reducir defectos en los productos, por medio de la planeación y entrega de

modulos mas pequeños se evitan reducir los defectos de cada producto.

•Estimar y planear el trabajo, cada desarrollador debe planear sus horas de trabajo para la entrega de cada modulo asignado y el tiempo de destina a las pruebas individuales.

### Principios del PSP

El diseño de PSP se basa en los siguientes principios de planeación y de calidad según HUMPHREY

- Cada ingeniero es esencialmente diferente; para ser más precisos, los ingenieros deben planear su trabajo y basar sus planes en sus propios datos personales.
- Para mejorar constantemente su funcionamiento, los ingenieros deben utilizar personalmente procesos bien definidos y medidos.
- Para desarrollar productos de calidad, los ingenieros deben sentirse personalmente comprometidos con la calidad de sus productos.
- Cuesta menos encontrar y arreglar errores en la etapa inicial del proyecto que encontrarlos en las etapas subsecuentes.
- Es más eficiente prevenir defectos que encontrarlos y arreglarlos.
- La manera correcta de hacer las cosas es siempre la manera más rápida y más barata de hacer un trabajo.

Para hacer un trabajo de ingeniería de software de la manera correcta, los ingenieros deben planear de la mejor manera su trabajo antes de comenzarlo y deben utilizar un proceso bien definido para realizar de la mejor manera la planeación del trabajo.

NIVEL	NOMBRE	ACTIVIDAD
PSPO	Medición personal	Registro de tiempo Registro de defectos
PSP0.1	Registro de defectos	Patrón de tipos de defectos Patrón de codificación Medida de tamaño Propuesta de mejoramiento de procesos
PSP1	Planeamiento Personal	Estimación de tamaño Informe de pruebas Planeamiento de tareas Cronogramas
PSP2	Gerenciamiento de la calidad personal	Revisiones de código Revisiones de proyecto Patrones de proyecto
PSP3	Proceso Personal Cíclico	Desarrollo Cíclico

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>5 de 28</b>

Tabla 1: Niveles de Mejoramiento PSP Fuente: (Humphrey, 2005)

### Que son las tecnologías de la información

Además de los programas y los datos, los usuarios necesitan también una explicación de cómo usar el programa. Documentos como manuales de usuario y de operación son necesarios para permitir a los usuarios operar con el sistema. Los documentos también son requeridos por las personas encargadas de mantener el software para entender el interior del software y modificarlo, en el caso en que sea necesario.

### Software

Algunas definiciones de **software**:

IEEE Std. 610 define el software como “programas, procedimientos y documentación y datos asociados, relacionados con la operación de un sistema informático”

Según el Webster’s New Collegiate Dictionary (1975), “software es un conjunto de programas, procedimientos y documentación relacionada asociados con un sistema, especialmente un sistema informático”. El software se puede definir como el conjunto de **tres componentes**:

Programas (instrucciones): este componente proporciona la funcionalidad deseada y el rendimiento cuando se ejecute.

Datos: este componente incluye los datos necesarios para manejar y probar

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>6 de 28</b>

programas y las estructuras requeridas para mantener y manipular estos datos.

- Documentos: este componente describe la operación y uso del programa.



Figura No 1. Componentes del software. Fuente: Inteco

Componentes del software.

Es importante contar con una definición exhaustiva del software ya que de otra manera se podrían olvidar algunos componentes. Una percepción común es que el software sólo consiste en programas. Sin embargo, los programas no son los únicos componentes del software.

### Programas

Los programas son conjuntos de instrucciones que proporcionan la funcionalidad deseada cuando son ejecutadas por el ordenador. Están escritos usando lenguajes específicos que los ordenadores pueden leer y ejecutar, tales como lenguaje ensamblador, Basic, FORTRAN, COBOL, C Los programas también pueden ser generados usando generadores de programas.

### Datos

Los programas proporcionan la funcionalidad requerida manipulando datos. Usan datos para ejercer el control apropiado en lo que hacen. El mantenimiento y las pruebas de los programas también necesitan datos. El diseño del programa asume la disponibilidad de las estructuras de datos tales como bases de datos y archivos que contienen datos.

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>7 de 28</b>

## Documentos

Además de los programas y los datos, los usuarios necesitan también una explicación de cómo usar el programa.

Documentos como manuales de usuario y de operación son necesarios para permitir a los usuarios operar con el sistema.

Los documentos también son requeridos por las personas encargadas de mantener el software para entender el interior del software y modificarlo, en el caso en que sea necesario.

## Diseño de Base de Datos

Según el diccionario de la Real Academia Española (2009), una base de datos es un conjunto de datos organizado de tal modo que permita obtener con rapidez diversos tipos de información. En la actualidad, las bases de datos son el principal medio de almacenamiento sobre el cual se apoyan los sistemas de información para hacer caminar al mundo. Por esta razón es suma importancia que las bases de datos sean diseñadas de la manera más óptima posible. Existen varios conceptos relacionados a bases de datos. Una tabla es el concepto fundamental en bases de datos. “Una tabla nos permite mostrar la información de una forma compacta. La tabla tiene una columna para cada atributo de los objetos o entidades descritas. Cada entrada de la tabla es una fila que contiene los valores para cada atributo. Las filas se pueden considerar como registros, y las columnas como campos de registro.” (Mayne y Wood, 1985).

## Orientación a objetos

La orientación a objetos se define como una metodología de diseño de *software* que modela las características de objetos reales o abstractos por medio del uso de clases y objetos. La orientación a objetos involucra muchos conceptos, pero es importante conocer los conceptos básicos necesarios, estos conceptos son

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>8 de 28</b>

explicados a continuación.

## **Objeto**

El objeto, es el concepto principal sobre el cual se fundamenta la tecnología orientada a objetos. Según el Diccionario de la Real Academia Española, un objeto es “todo lo que puede ser materia de conocimiento o sensibilidad de parte del sujeto, incluso este mismo.” Según literatura de orientación a objetos, se menciona que “un objeto tiene identidad, estado y comportamiento.” (Booch, 1990). En el mundo real podemos encontrar muchos ejemplos que cumplen con esta definición, algunos de ellos son: una persona, un automóvil, un perro, un edificio, etcétera.

## **Atributos**

Unos atributos corresponden a una característica de un objeto. Según el Diccionario de la Real Academia Española, un atributo es “cada una de las cualidades o propiedades de un ser”. En el mundo real cada objeto tiene un conjunto de características. Por ejemplo, en el caso un automóvil, sus características o atributos corresponderían al modelo, el color, la marca, el número de placa, tamaño, etcétera.

## **Acciones**

Además de un conjunto de atributos, un objeto tiene comportamiento, el cual es expresado mediante un conjunto de acciones. “Una acción es una determinada actividad que puede afectar los atributos del objeto” (De Champeaux, 1994) En el caso concreto del automóvil, entre las posibles acciones podrían estar: acelerar, frenar y virar.

## **Clase**

Según De Champeaux (1994), una clase puede entenderse como una familia de



**FORMATO REGISTRO DOCUMENTO  
CONSOLIDADO PAT COLECTIVO**

<b>Cod-Doc</b>	<b>FT-IV-015</b>
<b>Versión</b>	<b>1</b>
<b>Fecha</b>	<b>16/01/2017</b>
<b>Página</b>	<b>9 de 28</b>

objetos que tienen algo en común, ó más precisamente hablando, una clase es todo aquello que el conjunto de objetos tiene en común. De esta manera, una clase es caracterizada por su conjunto de atributos definidos y acciones; esta colección de atributos y acciones no cambia.

Así, podemos tener una clase Automóvil con sus atributos (modelo, el color, la marca, el número de placa, tamaño) y acciones (acelerar, frenar, virar); y podemos tener múltiples objetos o instancias de esta clase, por ejemplo, podemos tener un automóvil color verde marca Toyota ó un automóvil azul marca Ferrari, ambos con la capacidad de acelerar, frenar y virar.

### **Antecedentes de la investigación**

Los antecedentes a utilizar para la elaboración de nuestro sistema electoral estudiantil serán artículos e informes ya antes plantados por instituciones la cual nos permitirá guiarnos en nuestro proceso. es decir que nos servirán de guía en la elaboración del programa:

Modelado de un sistema de voto electrónico Revista Digital Universitaria UNAM 2014 Universidad autónoma de México.

Análisis - diseño estructurado del sistema de votación electrónico vía web e-vote (2011) universidad mayor de san Andrés de ciencias puras y naturales carrera de informática.

Para poder realizar y entregar productos de calidad es necesario seguir unos lineamientos que ayuden a lograr la meta con gran éxito, lineamientos que se deben seguir al pie de la letra que generan en el creador o desarrollador. lineamientos como los del PSP que son aplicados en proyectos y verificado en investigaciones como:

### **Estudio del Proceso Software Personal (PSP) en un entorno académico**

“PSP también ayuda a los ingenieros a mejorar la calidad de sus productos reduciendo las tasas de inyección de defectos. Diversas compañías quienes han aplicado los principios del PSP han mostrado un aumento de calidad en sus productos software, así como una reducción en los tiempos de desarrollo (Hayes y Over, 1997; Ferguson et al., 1997; Kelly y Culleton, 1999).”

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>10 de 28</b>

donde su conclusión fue la siguiente: “Los resultados de este estudio indican una ligera mejoría con respecto a las estimaciones de tamaño y esfuerzo. Estos resultados probablemente se deban a que los estudiantes no cuentan con suficiente experiencia codificando programas informáticos. Referente a la calidad del producto, los defectos removidos en la actividad de verificación se redujeron en un factor de 6.6. Se observa que el uso de actividades preventivas como son las revisiones de diseño y de código demuestran ser efectivas.”

Gómez, O. S., Aguilera, A. A., Gómez, G. E., & Aguilar, R. A. (2014). Estudio del Proceso Software Personal (PSP) en un entorno académico. ReCIBE. Revista electrónica de Computación, Informática Biomédica y Electrónica, (2).

otros estudios que son antecedentes del uso y aplicación del PSP también es:

### **Desarrollo de un sistema de gestión de competencias aplicando métricas de software y proceso unificado**

que consistió en: “Se ha desarrollado un sistema de información para la gestión de competencias de la Universidad Autónoma de Manizales UAM®, el cual permite administrar las competencias que los estudiantes desarrollan en su proceso de formación, y a los docentes les permite valorar dichas competencias, realizar seguimientos y consultas mediante diferentes criterios. Para el desarrollo de este sistema se utilizó como base la metodología del Proceso Unificado y los lineamientos de PSP0.1”

Donde el PSP jugó un papel muy importante en cuanto a la estimación del tamaño y tiempo del desarrollo del software como lo expresan así:

“ Al soportar el desarrollo en la metodología de Proceso Unificado se definieron tres iteraciones al inicio del proyecto, y las fases desarrolladas fueron: inicio, elaboración e implementación. No se realiza fase de transición ya que se desarrolló un prototipo funcional. Para cada iteración se han llevado a cabo las actividades propias del ciclo de vida del desarrollo del software, estas son: análisis, diseño, implementación o codificación, pruebas y **postmortem, esta última es propia de PSP y consiste en dar un estimado en cantidad de líneas de código para comparar con la cantidad real obtenida, en la suma de los tiempos totales invertidos en cada una de las actividades de desarrollo con porcentajes a la fecha.** Dentro de cada una de estas actividades se **han aplicado los formatos de PSP0.1**, el cual brinda la posibilidad de medir tamaño del software por cantidad de líneas de código,

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>11 de 28</b>

medición de tiempo neto invertido en cada una de las actividades y registro de defectos, para ello se ha utilizado un programa llamado Process Dashboard.”

Ceballosa, A. D., & Osoriob, A. M. G. Desarrollo de un sistema de gestión de competencias aplicando métricas de software y proceso unificado. otro antecedente es:

### **Entregar software de alta calidad.**

“Al utilizar en su trabajo los métodos y conceptos del PSP, los individuos de casi cualquier campo técnico pueden mejorar sus habilidades de valoración y planificación, adquirir compromisos realizables, gestionar la calidad de su trabajo y reducir el número de 4 PSPSM, Personal Software ProcessSM , TSPSM y Team Software ProcessSM son marcas de servicio de Carnegie Mellon University. 8 Entregar software de alta calidad Cómo cumplir de manera constante los compromisos de calendario, sin sacrificar la calidad defectos de sus productos. La capacitación en PSP forja competencias individuales para generar trabajo de alta calidad.”

Baez, J., & de NEORIS, C. D. T. (2010). Entregar software de alta calidad.

Se tiene como antecedente un artículo que como punto principal es el crear una industria de software en su país (cuba) y que para lograr eso, plantea que se debe seguir unos buenos procesos y por ellos piensa y habla del PSP(pero aplicándolo a un grupo de trabajo). cuyo título es: **Creando un profesional con disciplina en el proceso de desarrollo de software.**

“Para desarrollar la industria cubana de software es preciso contar con procesos bien definidos y un personal competente, entrenado en una disciplina personal y de trabajo en equipo. El proceso de software personal (PSP) contribuye a alcanzar disciplina en el trabajo individual. Aquí se propone un conjunto de consideraciones para introducir las prácticas de PSP y el desarrollo de roles de forma paulatina e incremental con el objetivo de crear las bases para introducir métodos disciplinados de trabajo en equipo y cumplir con los requerimientos de la industria.”

André Ampuero, M., & López Trujillo, Y. (2006). Creando un profesional con disciplina en el proceso de desarrollo de software. Ingeniería Industrial, 27(1).



**FORMATO REGISTRO DOCUMENTO  
CONSOLIDADO PAT COLECTIVO**

<b>Cod-Doc</b>	<b>FT-IV-015</b>
<b>Versión</b>	<b>1</b>
<b>Fecha</b>	<b>16/01/2017</b>
<b>Página</b>	<b>12 de 28</b>

## **Marco conceptual.**

**Formularios:** proveen un conveniente y consistente marco de trabajo para recolectar y retener datos. Especifican los datos requeridos y donde estos deben ser registrados.

**Manejo de defectos:** En el PSP la calidad del producto es medida en términos de "defectos", en donde un defecto es cualquier cosa dentro del programa que deba ser cambiada para permitir que sea propiamente desarrollado, mejorado o usado. Los defectos pueden estar en el código, diseño, requisitos, especificaciones o en la documentación

**Administración del Rendimiento:** El rendimiento es la principal medida de calidad del PSP. El rendimiento total del proceso es el porcentaje de defectos encontrados y corregidos antes que los desarrolladores comiencen a compilar y probar sus programas.

**Controlar el costo de la calidad:** Para gestionar la calidad del proceso, el PSP usa tres mediciones de costo de calidad:

**Costos de estimación:** tiempo del desarrollo utilizado en el diseño y revisión del código.

**Costos de fallas:** tiempo utilizado en compilar y probar

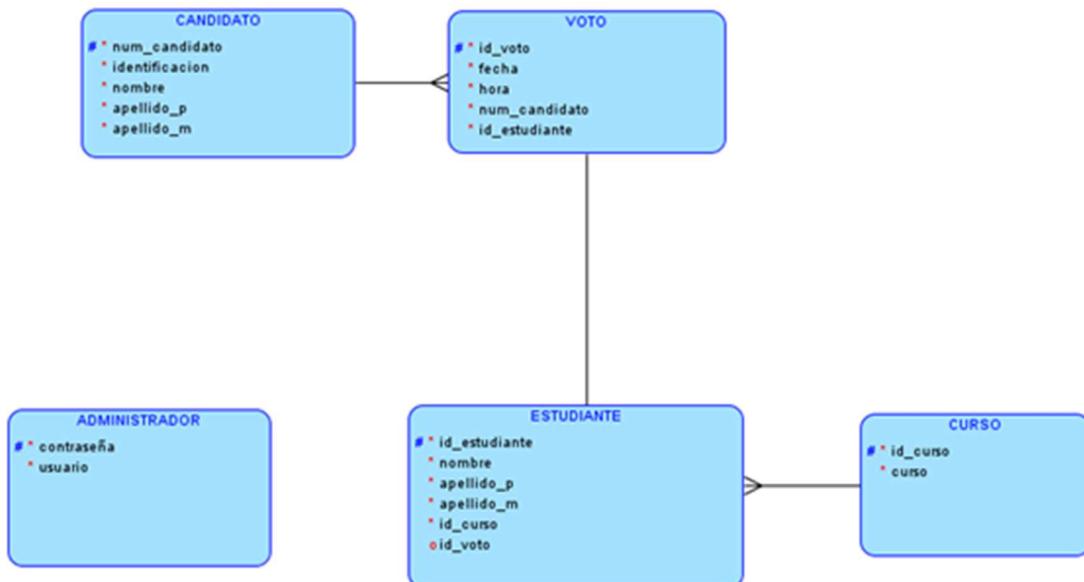
**Costos de prevención:** tiempo utilizado en la prevención de defectos antes que éstos ocurran.

Salinas, E., Cerpa, N., & Rojas, P. (2011). Arquitectura orientada a servicios para software de apoyo para el proceso personal de software. *Ingeniare. Revista chilena de ingeniería*, 19(1), 40-52.

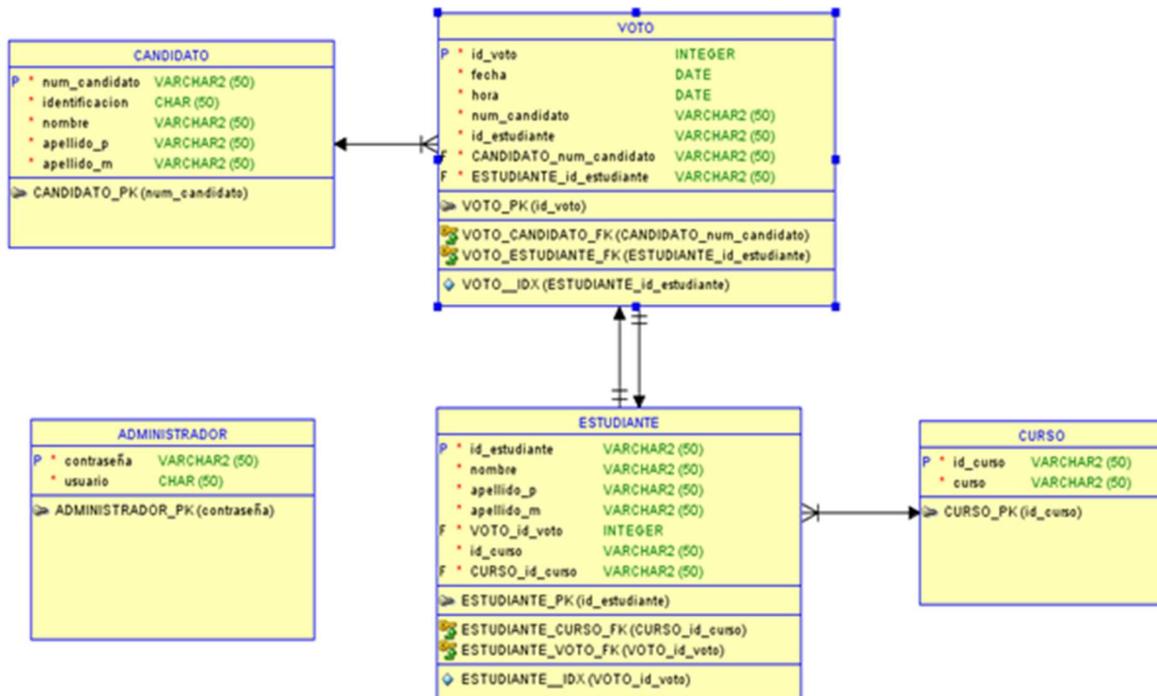
*Desarrollo de un sistema electoral estudiantil, utilizando proceso personal del software*

## DISEÑO DE LA BASE DE DATOS

### Modelo Lógico



### Modelo Relacional



Los dos diseños anteriores (lógico y relacional) de la base de datos del sistema electoral estudiantil consta de cinco tablas que se relacionan entre sí: la tabla CANDIDATO, la tabla VOTO, la tabla ESTUDIANTE y la tabla CURSO.

**Tabla Candidato:** en esta tabla se recoge los datos de los aspirantes a personero estudiantil. Los campos que se encuentran son: Identificación que es el número de su documento de identidad y este campo es obligatorio (not null) y es de tipo varchar 50, también es de tipo único, es decir irrepetible. el campo del nombre es de tipo varchar 50. los campos para los apellidos son obligatorios y de tipo varchar 50. la llave primaria de esta tabla es el número representativo del candidato ante los demás estudiantes.

**Tabla Voto:** esta tabla consta de cinco campos: el id voto que es de tipo numérico que identifica cada voto (es la llave primaria de la tabla) y en esta tabla cada voto está relacionado con la llave primaria de la tabla del candidato. Los

 <b>CORPORACIÓN UNIVERSITARIA</b> <b>RAFAEL NÚÑEZ</b> <small>PARA QUE TU DESARROLLO CONTINUE SU MARCHA</small>	<b>FORMATO REGISTRO DOCUMENTO</b> <b>CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>15 de 28</b>

campos que se encuentran son: fecha de tipo date, la hora (time), el número del candidato y el id del estudiante que efectuó el voto.

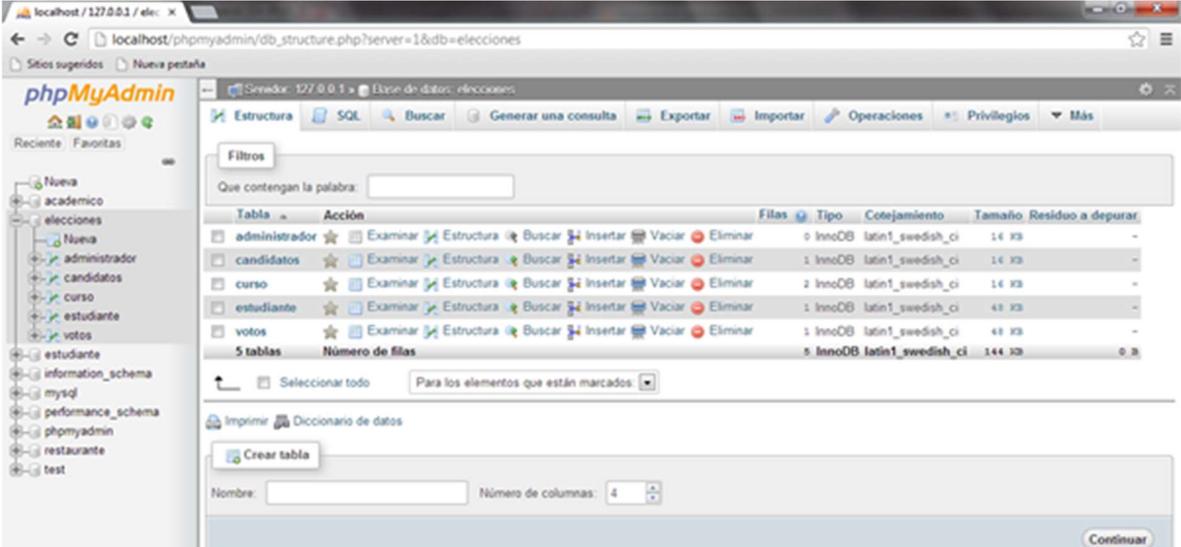
**Tabla Estudiante:** esta tabla está compuesta de cinco campos que son el id\_estudiante que es el número de identificación de cada estudiante, nombre que es de tipo varchar al igual que los campos de apellido y el último campo es el del curso al que pertenece dicho estudiante.

**Tabla Curso:** solo consta de dos campos el id\_curso que hace referencia al código de cada curso el cual es de tipo varchar 50 al igual que el campo curso que es el nombre del curso.

**Tabla administrador:** Esta tabla consta de dos campos usuario y contraseña los cuales los dos son de tipo varchar 50 la llave primaria de esta tabla es la contraseña. El campo usuario es único.

La relación existente entre la tabla candidato y la tabla voto es de uno a muchos por eso la llave primaria del candidato pasa como llave foránea a la tabla de voto. La relación de voto a estudiante es de uno a uno y la del curso a estudiante es de uno a muchos esto hace que la llave primaria de la tabla curso pase a la tabla estudiante como llave foránea.

## Construcción de la BD



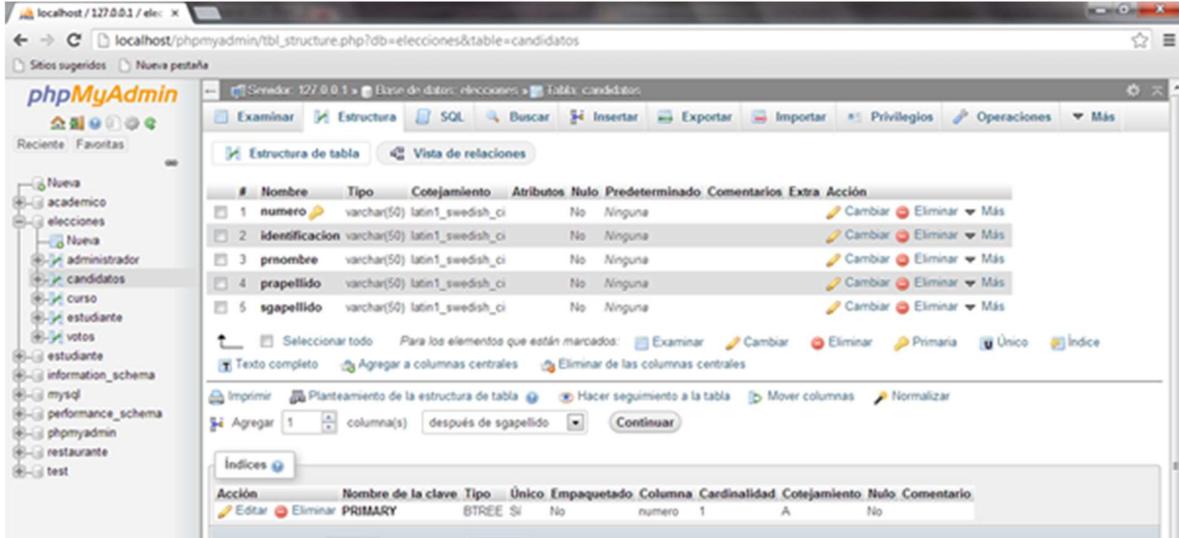
The screenshot shows the phpMyAdmin interface for a database named 'elecciones'. The left sidebar shows a tree view of databases, with 'elecciones' selected. The main area displays a table structure for the 'elecciones' database. The table list includes:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
administrador	Examinar	0	InnoDB	latin1_swedish_ci	14 KB	-
candidatos	Examinar	1	InnoDB	latin1_swedish_ci	14 KB	-
curso	Examinar	2	InnoDB	latin1_swedish_ci	14 KB	-
estudiante	Examinar	1	InnoDB	latin1_swedish_ci	48 KB	-
votos	Examinar	1	InnoDB	latin1_swedish_ci	48 KB	-
<b>5 tablas</b>	<b>Número de filas</b>	<b>5</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>144 KB</b>	<b>0 B</b>

At the bottom of the interface, there is a 'Crear tabla' (Create table) section with a 'Nombre' (Name) field and a 'Número de columnas' (Number of columns) dropdown set to 4.

La base de datos tiene por nombre elecciones como se puede observar al lado izquierdo de la ventana. La cual muestra las cinco tablas que contiene. Candidatos, Votos, administrador, estudiante y curso.

**Candidato:**

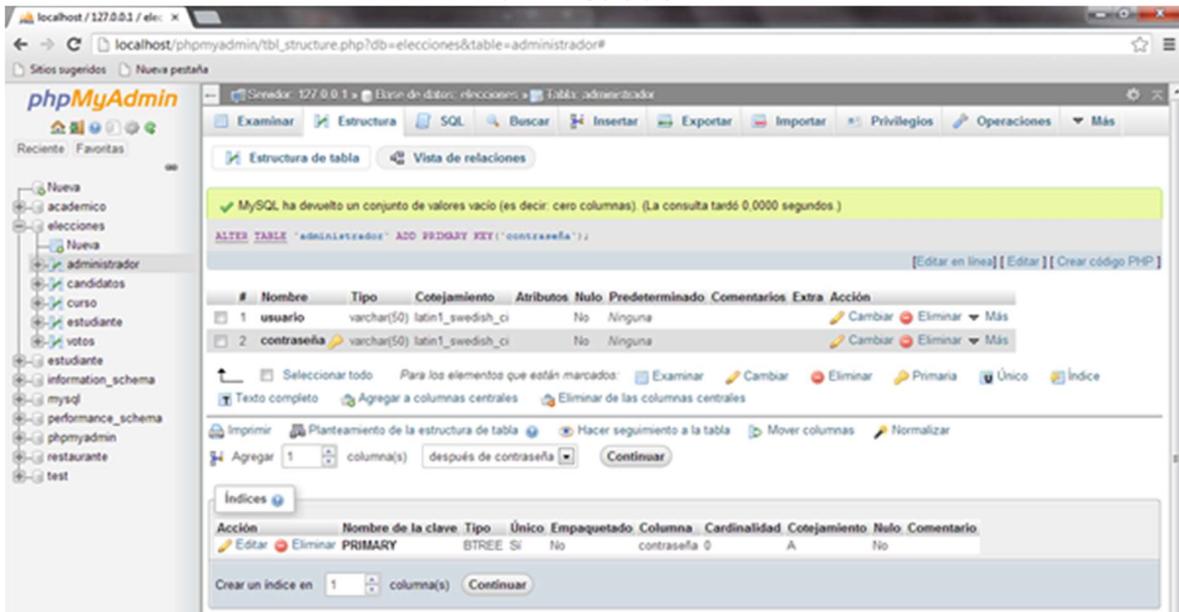


The screenshot shows the phpMyAdmin interface for the 'candidatos' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	numero	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
2	identificacion	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
3	prnombre	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
4	prapellido	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
5	sgapellido	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más

The primary key is 'numero'.

**Administrador**

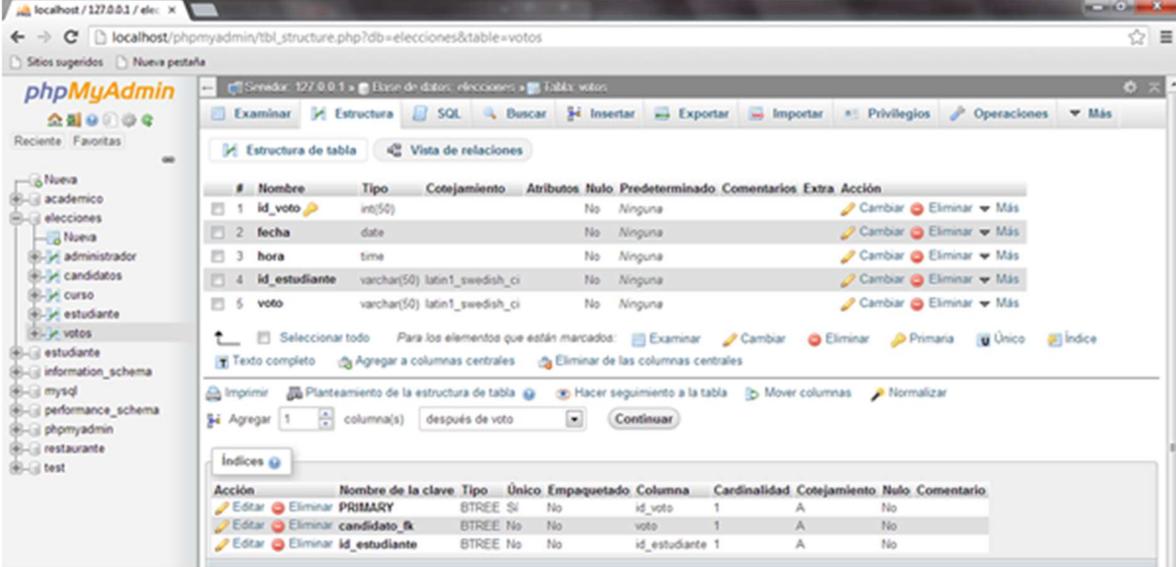


The screenshot shows the phpMyAdmin interface for the 'administrador' table. A message indicates that the table is empty. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	usuario	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
2	contraseña	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más

The primary key is 'contraseña'.

## Votos



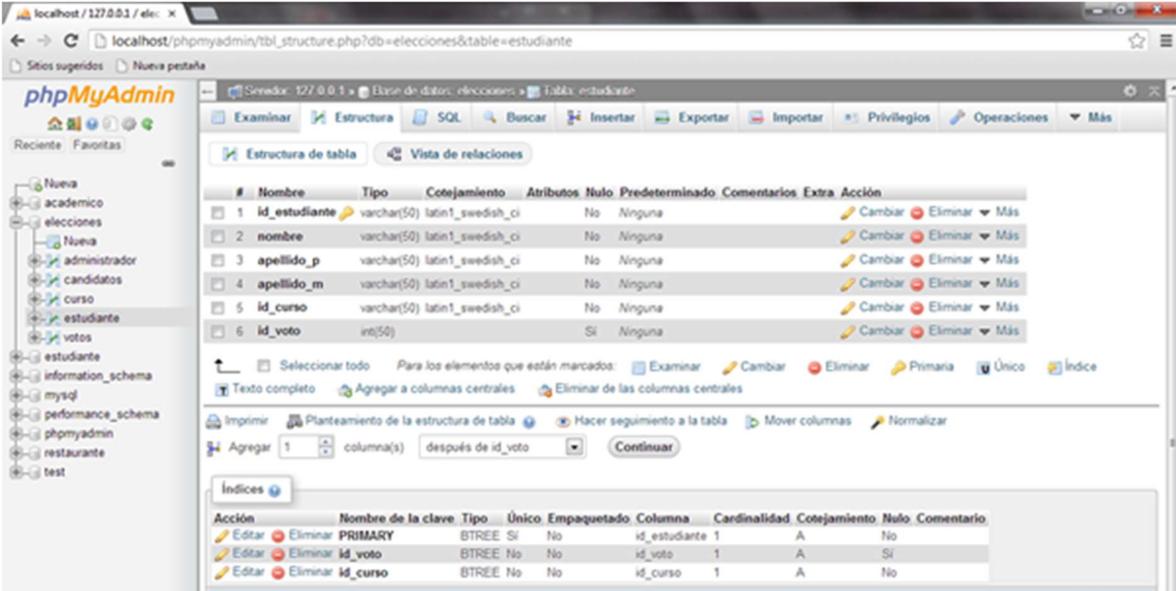
The screenshot shows the phpMyAdmin interface for the 'votos' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id_voto	int(50)			No	Ninguna			Cambiar Eliminar Más
2	fecha	date			No	Ninguna			Cambiar Eliminar Más
3	hora	time			No	Ninguna			Cambiar Eliminar Más
4	id_estudiante	varchar(50) latin1_swedish_ci			No	Ninguna			Cambiar Eliminar Más
5	voto	varchar(50) latin1_swedish_ci			No	Ninguna			Cambiar Eliminar Más

Indices:

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
Editar Eliminar	PRIMARY	BTREE	Si	No	id_voto	1	A	No	
Editar Eliminar	candidato_fk	BTREE	No	No	voto	1	A	No	
Editar Eliminar	id_estudiante	BTREE	No	No	id_estudiante	1	A	No	

## Estudiante



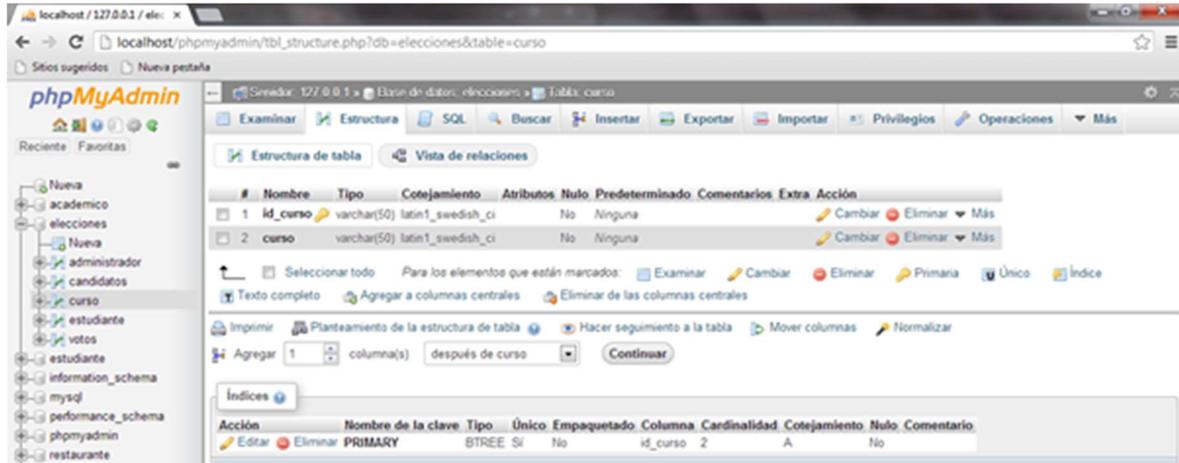
The screenshot shows the phpMyAdmin interface for the 'estudiante' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id_estudiante	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
2	nombre	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
3	apellido_p	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
4	apellido_m	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
5	id_curso	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
6	id_voto	int(50)			Si	Ninguna			Cambiar Eliminar Más

Indices:

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
Editar Eliminar	PRIMARY	BTREE	Si	No	id_estudiante	1	A	No	
Editar Eliminar	id_voto	BTREE	No	No	id_voto	1	A	Si	
Editar Eliminar	id_curso	BTREE	No	No	id_curso	1	A	No	

## Curso



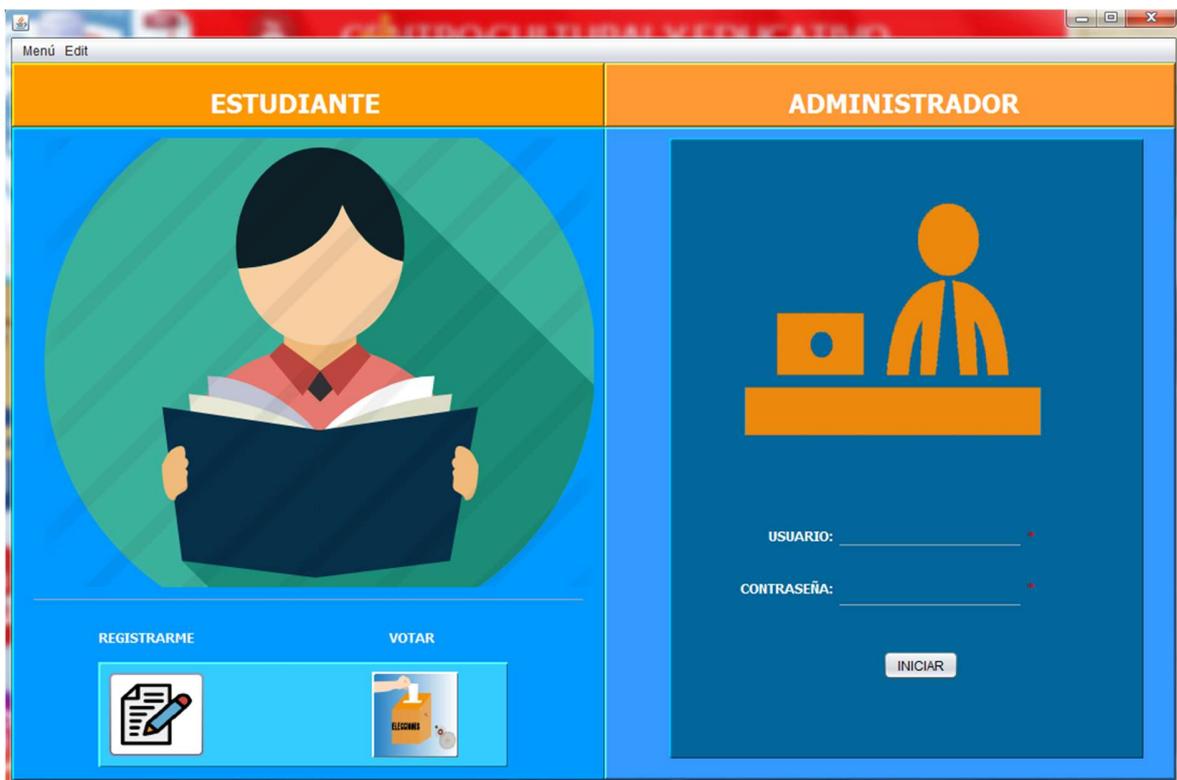
La Herramienta que utilizaremos para desarrollar la aplicación es netbeans bajo el lenguaje de programación java porque primeramente java es el único lenguaje de programación que hasta ahora conocemos y en el que nos hemos desarrollado desde que iniciamos con la temática de la programación y nos parece muy bien el trabajar con un lenguaje orientado a objetos el cual se siente más organizado y comprensible y el entorno de programación (netbeans) que utilizaremos lo sentimos muy claro y práctico a la hora de programar.

el gestor de base de datos del que haremos uso es Phpmyadmin Mysql porque la interfaz que brinda phpmyadmin es muy comoda y practica. ya que nos facilita la elaboración de bases de datos, sus tablas, sus campos y sus constraint sin necesidad de desgastarse tanto para escribirlo a código. es una gran ayuda porque nos ayuda a optimizar el tiempo.

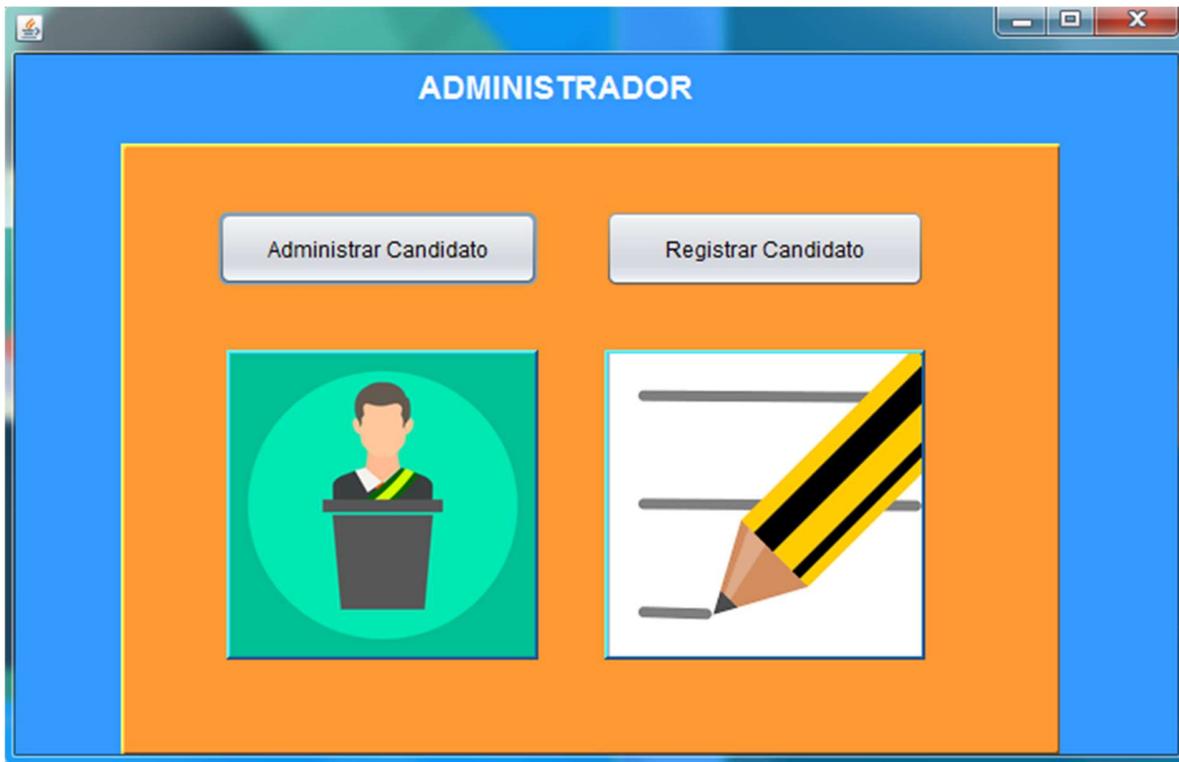
## MANUAL

Esta es la ventana de inicio y como se puede ver está dividida en dos una para el estudiante que puede hacer dos acciones: Registrarse y votar. La otra sección es la del administrador donde el administrador debe ingresar su usuario y contraseña para poder entrar y realizar sus labores que son: registrar a un candidato y administrar al candidato que consiste es cambiar ciertos datos del candidato (si llegara a haber errores con los datos del candidato), y también el administrador tendrá la autoridad de eliminar al candidato

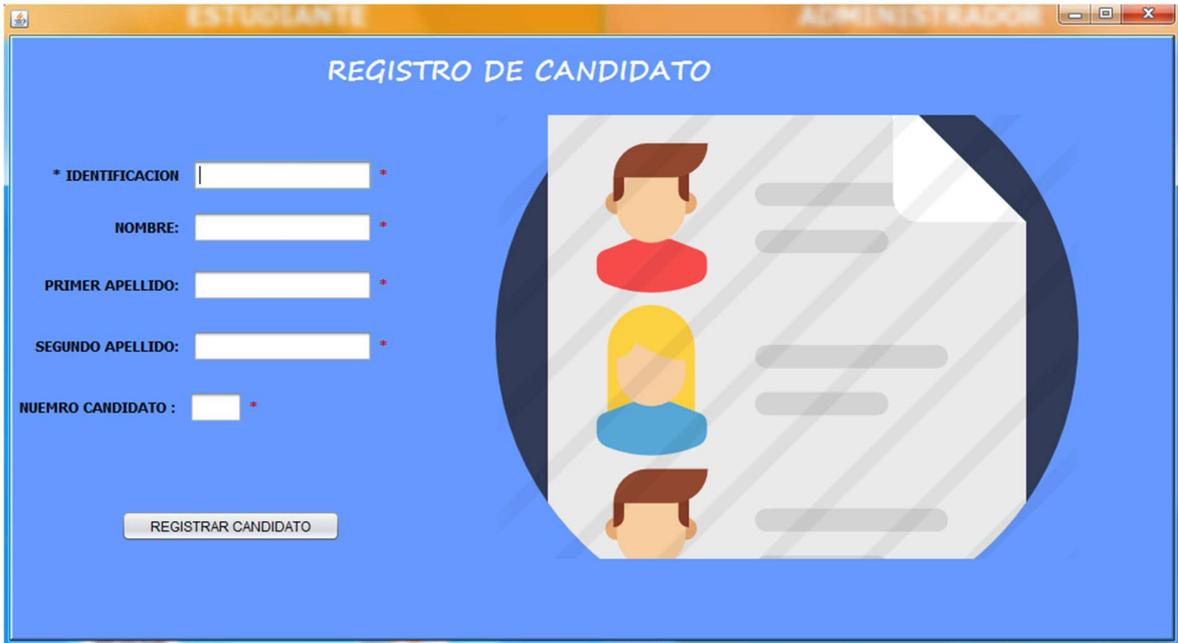
Para indicar cuales campos son obligatorios los hemos marcados o denotados con asteriscos como se puede notar en los campos de usuario y contraseña.



Este es la zona del administrador cuando haya iniciado sesión como se ve tiene dos acciones hasta el momento uno registrar a un nuevo candidato o aspirante a ser personero estudiantil y la otra de poder administrarlo.



Esta es la ventana del registro del candidato como podemos ver todos los campos en estos registros son obligatorios todos tiene el asterisco que los identifican.



ESTUDIANTE ADMINISTRADOR

### REGISTRO DE CANDIDATO

\* IDENTIFICACION  \*

NOMBRE:  \*

PRIMER APELLIDO:  \*

SEGUNDO APELLIDO:  \*

NUMERO CANDIDATO :  \*

REGISTRAR CANDIDATO

En esta ventana el administrador podrá buscar al candidato que desee y podrá modificarlo y también eliminarlo. Para poder buscar a un candidato se deberá obligatoriamente ingresar la identificación del candidato (estudiante) para poder buscarlo, pero al buscarlo los campos nombre, primer apellido y segundo apellido estarán inhabilitados para ingresar texto solo mostraran los datos que se encuentren. Pero si el administrador desea cambiar actualizar algún dato deberá marcar la caja inferior (checkbox) para poder habilitar esos campos y poder ingresar nuevos datos y proceder a darle en el botón actualizar.



**ADMINISTRACION DE CANDIDATO**

IDENTIFICACION:

NOMBRE:

PRIMER APELLIDO:

SEGUNDO APELLIDO:

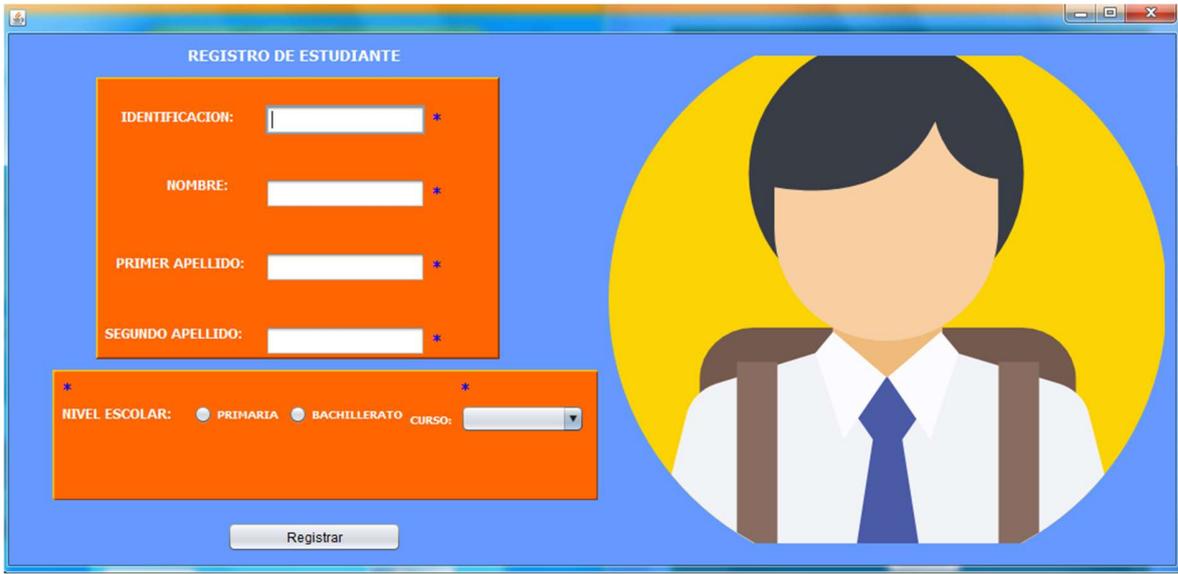
#TARJETON:

---

**opciones**

habilitar campos para actualizar datos

Esta es la ventana con la que el estudiante (votante) se encontrará cuando en el inicio le dé en el botón registrar (ya que solo podrá tener derecho al voto si se encuentra registrado) en esta ventana el estudiante que desee votar se registrará llenando cada campo si dejar ninguno en blanco ya que son obligatorios por eso tienen todos un asterisco.



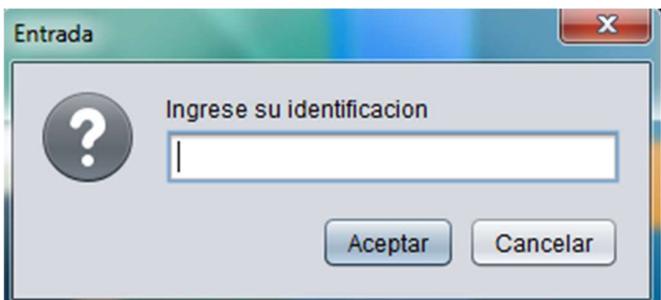
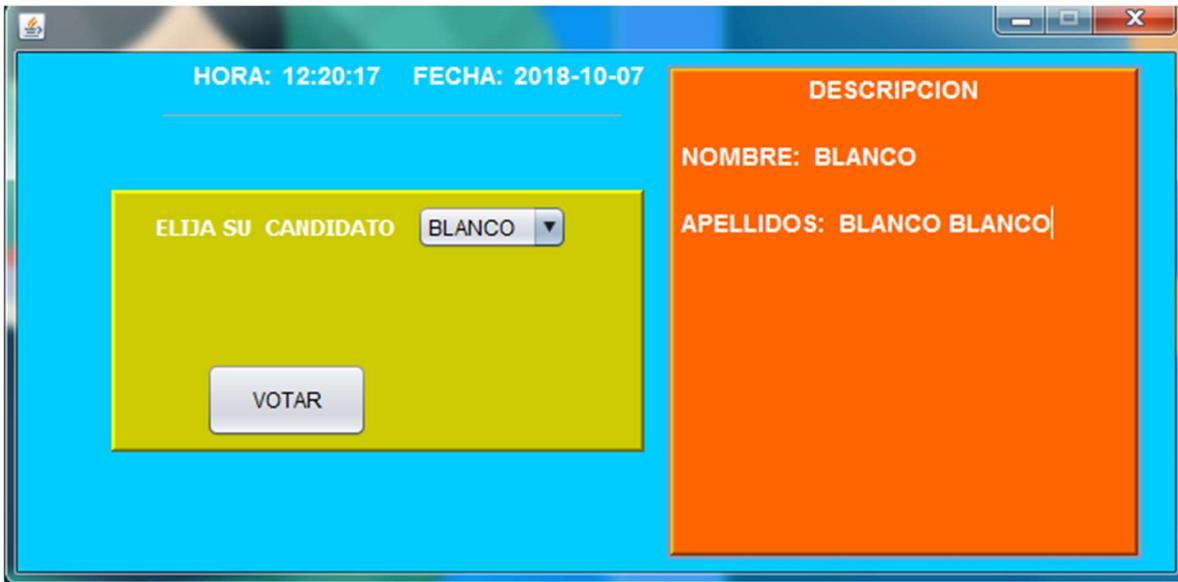
The screenshot shows a web application window titled "REGISTRO DE ESTUDIANTE". The form is set against a blue background and contains several input fields and a button. On the right side of the form is a large circular placeholder for a student's profile picture, featuring a stylized illustration of a person with dark hair, wearing a white shirt, a blue tie, and brown suspenders. The form fields are as follows:

- IDENTIFICACION:** A text input field with a red asterisk to its right.
- NOMBRE:** A text input field with a red asterisk to its right.
- PRIMER APELLIDO:** A text input field with a red asterisk to its right.
- SEGUNDO APELLIDO:** A text input field with a red asterisk to its right.
- NIVEL ESCOLAR:** A section with two radio buttons labeled "PRIMARIA" and "BACHILLERATO".
- CURSO:** A dropdown menu with a red asterisk to its right.

At the bottom center of the form is a button labeled "Registrar".

 <p>CORPORACIÓN UNIVERSITARIA <b>RAFAEL NÚÑEZ</b> PARA QUE TU DESARROLLO CONTINUE SU MARCHA</p>	<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
		<b>Versión</b>	<b>1</b>
		<b>Fecha</b>	<b>16/01/2017</b>
		<b>Página</b>	<b>25 de 28</b>

Cuando el estudiante deba votar tendrá que darle en el botón votar y continuación le aparecerá una ventanita pidiendo su identificación para validar dos cosas: primero si se encuentra registrado para votar el estudiante y detectar si ese estudiante ya realizó un voto o todavía no lo ha hecho. En caso de que el estudiante no se encuentre aparecerá un mensaje diciéndole al estudiante que no está registrado y por eso no podrá votar y si encuentra que el estudiante está registrado y ya realizó un voto el programa no lo dejara votar. Pero si no si el estudiante se encuentra registrado y no ha hecho voto entonces el programa le permitirá entrar a la zona de votación para que realice su votación.

La votación es sencilla el estudiante cuando entre solo tendrá que elegir su candidato como se muestra en la anterior imagen y proceder a darle en votar. La lista de candidatos se desplegará en el combobox y como se puede ver aparece primero con la opción de voto en blanco y en la zona derecha una simple descripción del candidato escogió.

La ventana de resultado:



**TOTAL DE VOTOS POR CANDIDATOS**

numero	prnombre	N° votos
BLANCO	BLANCO	1
C2	carolina	5
C1	javier	4
C4	juan	2
C3	Maria	1
C5	miguel	2

**TOTAL DE VOTOS**

**15**

**HISTORIAL DE VOTOS**

id_voto	fecha	hora	num_candidato
1	2018-11-15	09:29:24	C1
2	2018-11-15	09:37:28	C1
3	2018-11-15	01:15:23	BLANCO
4	2018-11-15	01:15:47	C1
5	2018-11-15	01:16:17	C2
6	2018-11-15	01:16:32	C3
7	2018-11-15	01:16:53	C4
8	2018-11-15	01:17:08	C5
9	2018-11-15	01:17:58	C1

**GANADOR**

**carolina**



**FORMATO REGISTRO DOCUMENTO  
CONSOLIDADO PAT COLECTIVO**

<b>Cod-Doc</b>	<b>FT-IV-015</b>
<b>Versión</b>	<b>1</b>
<b>Fecha</b>	<b>16/01/2017</b>
<b>Página</b>	<b>27 de 28</b>

## **BIBLIOGRAFÍA**

Modelado de un sistema de voto electrónico Revista Digital Universitaria UNAM  
2014 Universidad autónoma de México recuperado de:  
<http://www.revista.unam.mx/vol.15/num4/art24/#>

análisis - diseño estructurado del sistema de votación electrónico vía web e-vote  
(2011) universidad mayor de san Andrés de ciencias puras y naturales carrera de  
informática recuperado de :  
<http://inf162-grupo9-12011.blogspot.com/p/analisis-y-diseno-estructurado.html>

Salinas, E., Cerpa, N., & Rojas, P. (2011). Arquitectura orientada a servicios para  
software de apoyo para el proceso personal de software. Ingeniare. Revista  
chilena de ingeniería, 19(1), 40-52.

Gómez, O. S., Aguilera, A. A., Gómez, G. E., & Aguilar, R. A. (2014). Estudio del  
Proceso Software Personal (PSP) en un entorno académico. ReCIBE. Revista  
electrónica de Computación, Informática Biomédica y Electrónica, (2).

Ceballosa, A. D., & Osorio, A. M. G. Desarrollo de un sistema de gestión de  
competencias aplicando métricas de software y proceso unificado.

Baez, J., & de NEORIS, C. D. T. (2010). Entregar software de alta calidad.

André Ampuero, M., & López Trujillo, Y. (2006). Creando un profesional con  
disciplina en el proceso de desarrollo de software. Ingeniería Industrial, 27(1).



<b>FORMATO REGISTRO DOCUMENTO CONSOLIDADO PAT COLECTIVO</b>	<b>Cod-Doc</b>	<b>FT-IV-015</b>
	<b>Versión</b>	<b>1</b>
	<b>Fecha</b>	<b>16/01/2017</b>
	<b>Página</b>	<b>28 de 28</b>