

**TiTulo**

**Sofware de calificaciones estudiantiles**

**INTEGRANTE:**

* Miller vargas mola (**87223**).

**PRESENTADO A**:

**ING**. John Carlos Arrieta Arrieta.

**INGENIRA EN SISTEMAS TERCER SEMESTRE**

**CARTAGENA DE INDIAS D, T Y C**

**Marzo - 13 - 2022**

Contenido

[INTRODUCCIÓN 4](#_Toc1)

[OBJETIVO 5](#_Toc2)

[Planteamiento del problema 6](#_Toc3)

[JUSTIFICACIÓN 7](#_Toc4)

**INTRODUCCIÓN**

En este documento se explicará todo acerca del software que se realizará durante el proyecto, estaremos creando un software para una Universidad de chile cumpliendo los requerimientos funcionales y no funcionales necesarios para su total cumpliento, de esta manera crear buenas prácticas en el desarrollo y diseño de software, este proyecto tiene como fin promover el conocimiento individual y aumentar el interés al estudio y la responsabilidad que abarca cada punto de un proyecto, para ello se necesitan buenas bases para poder llevar acabo dicho proyecto implementado en tercer semestres de ingeniería de sistemas de la universidad del sinu

**OBJETIVO**

Al finalizar todo el software de escritorio como el de web se espera cumplir con éxito todos los requerimientos funcionales y no funcionales planteados, y tener un software completo,escalable,seguro y agradable al usuario para cumplir dicha meta se necesita implementar todos los conocimientos aprendidos hasta el momento tanto los visto en clase como lo de forma autónoma, también se busca aumentar la lógica individual y generar independencias entre cada estudiante para un mayor manejo de situaciones problemas en el ambiente laboral y estudiantil, también se estimula la responsabilidad individual de cada estudiante para aumentar su compromiso ante cualquier problema o proyecto que a de asumir

**Planteamiento del problema**

Los profesores de la asignatura de Computación II de la Universidad de Chile deciden crear una base de datos que contenga la información de los resultados de los controles realizadas a los alumnos. Para realizar el diseño se sabe que:

* Los alumnos están definidos por su n° de matrícula, nombre y el grupo al que asisten a clase. Dichos alumnos realizan dos tipos de controles a lo largo del curso académico:
* Controles escritos: cada alumno realiza varios a lo largo del curso, y se definen por el n° de control, el n° de preguntas de que consta y la fecha de realización (la misma para todos los alumnos que realizan el mismo control). Evidentemente, es importante almacenar la nota de cada alumno por control.
* Prácticas: se realiza un número indeterminado de ellas durante el curso académico, algunas serán en grupo y otras individuales. Se definen por un código de práctica, título y el grado de dificultad. En este caso los alumnos pueden examinarse de cualquier práctica cuando lo deseen, debiéndose almacenar la fecha y nota obtenida.
* En cuanto a los profesores, únicamente interesa conocer (además de sus datos personales: DNI y nombre), quien es el qué ha diseñado cada práctica, sabiendo que en el diseño de una práctica puede colaborar más de uno, y que un profesor puede diseñar más de una práctica. Interesa, además, la fecha en que ha sido diseñada cada práctica por el profesor correspondiente.

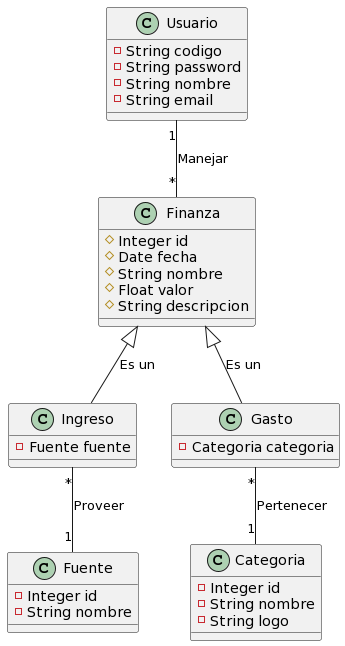
**JUSTIFICACIÓN**

Los profesores de la asignatura de Computación II de la Universidad de Chile necesitan crear una base de datos que contenga la información de los resultados de los controles realizadas a los alumnos de y se les hace algo tedioso llenarlos de manera escrita ya que con el tiempo se deterioran por dicho material por eso se busca la  manera más segura y óptima en el uso de estos datos,se requiere aumentar  el uso de los datos de manera oportuna sin amenazar dicha existencia y poder acceder a ellos de manera más óptima y fácil, esto resuelve los registros guardados en manuscritos, su fragilidad y falta de seguridad, se pretende aumentar la vida útil  de dichos registros y sin exponer su integridad también se aumenta su accesibilidad es necesario tener en cuenta todos los requerimientos dados por el cliente para finalidad de dicho software

METODOLOGÍA

Análisis de Requerimientos

* Metodología de software Tradicional
  + **Requerimientos Funciones**
  + **Requerimientos No Funcionales**
  + **Diagramas de casos de Uso**
  + Descripción descripción o especificación de casos de uso
* Metodología Ágil
  + **Historias de Usuario**
  + **Moockup - Prototipos de IU**
* Modelo Conceptual
  + **Diagrama de clases Conceptual**

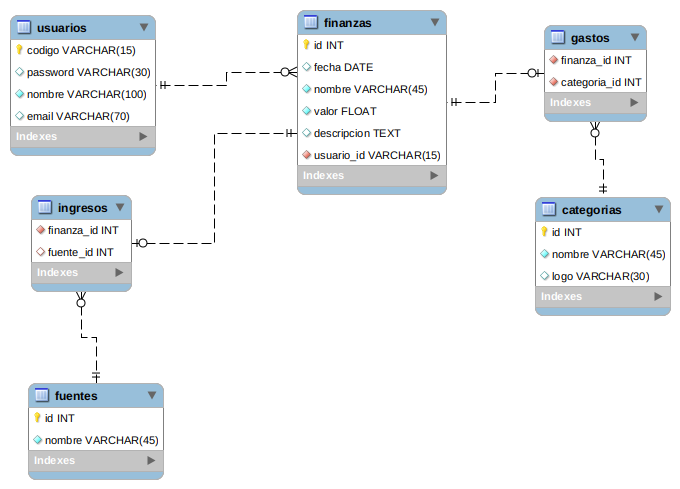


|  |
| --- |
| class Usuario{  - String codigo  - String password  - String nombre  - String email }  class Categoria{  - Integer id  - String nombre  - String logo }  class Finanza{  # Integer id  # Date fecha  # String nombre  # Float valor  # String descripcion }  class Fuente{  - Integer id  - String nombre }  class Ingreso{  - Fuente fuente  }  class Gasto{  - Categoria categoria }  Finanza <|-- Ingreso: Es un Finanza <|-- Gasto : Es un Ingreso "\*"--"1" Fuente: Proveer Gasto "\*"--"1" Categoria: Pertenecer Usuario "1"--"\*" Finanza: Manejar |

* Transformación del Diagrama de clases al lenguaje de programación.
  + ~~Diagrama de Entidad Relación de DB~~
  + ~~Diccionario de Datos~~
  + Modelo de Procesos (Investigar - Diagrama de Actividades o D
    - Diagramas de Procesos
    - Diagramas de Actividades

Diseño y desarrollo de Software

* **Modelo Relacional de BD**



* **Diagrama de clases de Diseño (Modelo de Robustez)**
* Diagrama de Secuencias (investigar)
* Diagrama de Componentes (investigar)
* Diagrama de Colaboración (Investigar)

Implementación y despliegue de software

* Diagrama de Despliegue (Investigar)
* Diagrama de Redes

Pruebas de software

* Pruebas Unitarias
* Pruebas de caja blanca
* Pruebas de caja negra
* Pruebas de estrés
* Pruebas de integración

Mantenimiento

* Mantenimiento preventivo
* Mantenimiento Correctivo
* Mantenimiento Adaptativo