

Universidad de San Carlos de Guatemala.

Facultad de Ingeniería.

Escuela de Ciencias.

PROYECTO 1:
MANUAL TECNICO

Nombre: Harold Steven Catún Guevara

Carné: 202145353

Docente: Ing. Neftalí Calderón

Sección: P

Fecha de la realización: -----

Fecha de entrega: 16/06/2023

CONTENIDO.

- 1. Pantalla de Inicio.....Error! Bookmark not defined.**
- 2. Registro -Empleado-Error! Bookmark not defined.**
- 3. Registro -Articulo-Error! Bookmark not defined.**
- 4. Registro -Cliente-Error! Bookmark not defined.**
- 5. Registro -Proveedor-.....Error! Bookmark not defined.**
- 6. Ingreso -Facturas-Error! Bookmark not defined.**

1. Descripción de la solución.

Se requiere desarrollar una aplicación de escritorio para la gestión de los cursos impartidos en la carrera. Como programador, se ha contratado para brindar una solución al problema. El objetivo principal de la solución es facilitar la administración de cursos, profesores, alumnos y actividades relacionadas.

El sistema permitirá el acceso a diferentes tipos de usuarios, incluyendo un único usuario administrador y profesores. Las funcionalidades del sistema se diferenciarán según el tipo de usuario.

El Administrador será responsable de cargar profesores y cursos en el sistema. Esta funcionalidad permitirá mantener actualizada la información académica y garantizar un correcto registro de los recursos educativos disponibles. La plataforma tendrá capacidad para almacenar hasta 50 profesores y 300 alumnos.

Por su parte, los Profesores podrán ser asignados a los cursos existentes en el sistema.

Dentro de cada curso, podrán gestionar la información de los alumnos, crear actividades educativas y generar reportes sobre el progreso de dichas actividades. Esto brindará a los profesores las herramientas necesarias para llevar a cabo sus clases de manera eficiente.

La aplicación de escritorio proporcionará una interfaz intuitiva y fácil de usar para que los usuarios puedan interactuar con el sistema de forma cómoda y eficaz. Además, se implementarán funcionalidades adicionales, como la capacidad de exportar listados y generar informes en formatos específicos, como HTML.

2. Requerimientos.

Software

- ☐ NetbeansIDE 12.6
- ☐ JDK 17
- ☐ Windows 7.

Hardware

- ☐ Una computadora, ya sea laptop o de escritorio.

3. Diccionario de clases.

Clase Actividad:

Atributos:

Nombre (string): El nombre de la actividad.

Descripción (string): La descripción de la actividad.

Ponderación (string): El peso o ponderación de la actividad en relación al curso.

Clase Alumno:

Atributos:

Código (string): El código del alumno.

Nombre (string): El nombre del alumno.

Apellido (string): El apellido del alumno.

Correo(string): El correo del alumno.

Genero(string): El genero del alumno.

Clase Curso:

Atributos:

Código (string): El código del curso.

Nombre (string): El nombre del curso.

Créditos (String): El número de créditos del curso.

Alumnos(string): El numero de alumnos asignados.

Profesor (string): El objeto Profesor asignado al curso.

Clase Profesor:

Atributos:

Codigo(string): El código del profesor.

Nombre (string): El nombre del profesor.

Apellido (string): El apellido del profesor.

Correo (string): El correo electrónico del profesor.

Contraseña(string): La contraseña del profesor.

Género (string): El género del profesor.

4. Diccionario de Métodos:

vActualizarCurso

public vActualizarCurso(): inicializa el JFrame y lo hace visible.

private void ActualizarBtnActionPerformed(java.awt.event.ActionEvent evt):

actualiza los datos del curso seleccionado en base a los nuevos datos ingresados.

vActualizarProfesor:

public vActualizarProfesor(): inicializa el JFrame y lo hace visible.

vAdminProfesores:

public vAdminProfesores(): inicializa el JFrame y lo hace visible.

private void bCrearProfActionPerformed(java.awt.event.ActionEvent evt): dirige al JFrame vCrearProfesor.

private void bEliminarActionPerformed(java.awt.event.ActionEvent evt): elimina al Profesor seleccionado.

private void bActualizarActionPerformed(java.awt.event.ActionEvent evt): dirige al JFrame vActualizarProfesor.

private void bCargaMActionPerformed(java.awt.event.ActionEvent evt): hace una carga masiva de java.

private void bHTMLActionPerformed(java.awt.event.ActionEvent evt): crea un archive HTML con los datos ingresados de profesores.

private void bCursonActionPerformed(java.awt.event.ActionEvent evt): llama al JFrame vAdministrarCursos.

public void actualizarListadoProfesores(): actualiza los constantemente el listado de profesores en la tabla.

vAdministrarCursos:

public vAdminProfesores(): inicializa el JFrame y lo hace visible.

private void bCrearProfActionPerformed(java.awt.event.ActionEvent evt): dirige al JFrame vCrearProfesor.

private void bEliminarActionPerformed(java.awt.event.ActionEvent evt): elimina al Profesor seleccionado.

private void bActualizarActionPerformed(java.awt.event.ActionEvent evt): dirige al JFrame vActualizarProfesor.

private void bCargaMActionPerformed(java.awt.event.ActionEvent evt): hace una carga masiva de java.

private void bHTMLActionPerformed(java.awt.event.ActionEvent evt): crea un archive HTML con los datos ingresados de profesores.

private void bCursonActionPerformed(java.awt.event.ActionEvent evt): llama al JFrame vAdministrarCursos.

public void actualizarListadoProfesores(): actualiza los constantemente el listado de profesores en la tabla.

vCrearCurso:

public vCrearCurso(): inicializa el JFrame y lo hace visible.

private void CrearBtnActionPerformed(java.awt.event.ActionEvent evt): crea un nuevo curso y lo agrega al arreglo.

vCrearProfesor:

public vCrearProfesor(): inicializa el JFrame y lo hace visible.

private void CrearBtnActionPerformed(java.awt.event.ActionEvent evt): crea un nuevo profesor y lo agrega al arreglo.

vCursos:

public vCursos(): inicializa el JFrame y lo hace visible.

private void bCargaAlumnosActionPerformed(java.awt.event.ActionEvent evt):

hace una carga masiva alumnos.

public void actualizarListadoAlumnos(): actualiza continuamente la tabla de alumnos.

vLogin:

private void bIngresarActionPerformed(java.awt.event.ActionEvent evt): verifica que se ingresen los datos correctos, y redirige dependiendo de que se ingrese.

public vLogin(): inicializa el JFrame y hace que sea visible el método.

vModulo: public vModulo(): inicializa el JFrame y lo hace visible.

private void ActualizarBtnActionPerformed(java.awt.event.ActionEvent evt):

llama al JFrame vActualizarProfesores.