



Universidad Veracruzana

Especificación de Requerimientos de Software

Proyecto: Tequila IDE

Control de versiones

| Fecha | Versión | Descripción | Autor |
|------------|---------|--|--|
| 23/09/2017 | 1.0 | Se inicio la especificación de requerimientos | Cámara Árciga Miguel Alejandro García Cruz Alan Yoset |
| 07/01/2018 | 1.1 | Se completaron las descripciones de casos de uso | Cámara Árciga Miguel Alejandro García Cruz Alan Yoset |

Tabla de contenido

| | |
|--|----|
| Introducción | 4 |
| Propósito | 4 |
| Alcance | 4 |
| Personal Involucrado | 4 |
| Definiciones, acrónimos y abreviaturas | 5 |
| Referencias | 6 |
| Descripción General | 6 |
| Perspectiva del producto | 6 |
| Funcionalidad del producto | 7 |
| Características de los usuarios | 7 |
| Restricciones | 9 |
| Suposiciones y dependencias | 9 |
| Evolución previsible del sistema | 9 |
| Requisitos comunes de las interfaces | 9 |
| Interfaces de usuario | 9 |
| Interfaces de hardware | 10 |
| Interfaces de software | 10 |
| Interfaces de comunicación | 10 |
| Requisitos Funcionales | 10 |
| Usuario | 10 |
| CU01 - Registrar usuario | 10 |
| CU02 - Iniciar sesión | 11 |
| CU03 - Compilar proyecto | 12 |
| CU13 - Editar biografía | 13 |
| CU04 - Ejecutar proyecto | 14 |
| CU18 - Editar archivo | 15 |
| CU15 - Agregar archivo | 15 |
| CU14 - Consultar proyectos | 16 |
| Colaborador Principal | 17 |
| CU09 - Agregar colaborador | 17 |
| CU10 - Eliminar colaborador | 18 |
| CU05 - Crear proyecto | 18 |
| CU12 - Eliminar proyecto | 19 |

| | |
|--|----|
| Colaborador Invitado | 20 |
| CU11 - Abandonar proyecto | 20 |
| Requisitos no funcionales | 21 |
| Requisitos de rendimiento | 21 |
| Seguridad | 21 |
| Fiabilidad | 22 |
| Disponibilidad | 22 |
| Mantenibilidad | 23 |
| Portabilidad | 23 |
| Apéndices | 24 |
| Diagrama de clases | 24 |
| Modelo de datos | 24 |
| Diagrama de despliegue | 25 |
| Justificación para el uso de frameworks en el proyecto | 25 |
| Node.js | 25 |
| JFoenix | 25 |
| Conclusiones | 26 |
| Miguel Alejandro Cámara Árciga | 26 |
| Conclusión | 26 |
| Lecciones aprendidas | 26 |
| Alan Yoset García Cruz | 26 |
| Conclusión | 26 |
| Lecciones aprendidas | 27 |

Introducción

Propósito

Este documento es una Especificación de Requerimientos de Software (ERS) para el desarrollo del proyecto Tequila IDE. Aquí se definirán las especificaciones funcionales y no funcionales del sistema. Este documento está enfocado principalmente para su uso por el equipo de desarrollo como punto de referencia en la primera versión.

Alcance

- El producto por desarrollar se clasifica como software de aplicación específica. Deberá funcionar en un entorno colaborativo en línea, que permita a los usuarios trabajar sobre el mismo proyecto en tiempo real.
- El sistema recibe el nombre de Tequila IDE y tendrá que proporcionar, entre otras, interfaces para la creación y modificación de proyectos de software, compilación, ejecución y eliminación de los mismos.
- A través de Tequila IDE no será posible ejercer depuración de software ni trabajar de manera local, todos los proyectos serán cargados desde y hacia el servidor.
- Tequila IDE representa un gran beneficio tanto para empresas como para equipos de desarrollo independiente que, en respuesta a su necesidad de trabajo colaborativo, utilizan herramientas para el control de versiones que no satisfacen sus requerimientos en tiempo y forma, que además no son tan intuitivas ni eficientes.

Personal Involucrado

| | |
|---------------------------------|---|
| Nombre: | Miguel Alejandro Cámara Árciga |
| Rol: | Analista, Diseñador, Programador, Tester |
| Categoría Profesional: | Estudiante de la Lic. en Ingeniería de Software |
| Información de contacto: | arcamsoft@gmail.com |

| | |
|---------------------------------|---|
| Nombre: | Alan Yoset García Cruz |
| Rol: | Analista, Diseñador, Programador, Tester |
| Categoría Profesional: | Estudiante de la Lic. en Ingeniería de Software |
| Información de contacto: | alancrux_@hotmail.com |

Definiciones, acrónimos y abreviaturas

De la disciplina:

- IDE: Entorno de desarrollo integrado. Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.
- Codificación: Etapa del ciclo de vida de software que consiste en traducir algoritmos previamente diseñados a un lenguaje de programación específico.
- Compilación: Es el proceso por el cual se traducen las instrucciones escritas en un determinado lenguaje de programación a lenguaje máquina.
- Ejecución: es el proceso mediante el cual una computadora lleva a cabo las instrucciones de un programa informático.
- Servidor (software): Un servidor es una aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.
- Servidor (hardware): es una máquina física integrada en una red informática en la que, además del sistema operativo, funcionan uno o varios servidores basados en software.
- Base de datos: es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.
- Interfaz gráfica: conjunto de elementos de la pantalla que permiten al usuario realizar acciones sobre el sistema.
- Plataforma: es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible.
- Lenguaje (de programación): es aquella estructura que, con una cierta base sintáctica y semántica, imparte distintas instrucciones a un programa de computadora.
- Programador: Un programador es aquella persona que escribe, depura y mantiene el código fuente de un programa informático.
- Desarrollador (software): es un especialista en informática que es capaz de concebir y elaborar sistemas informáticos (paquetes de software), así como de implementarlos y ponerlos a punto, utilizando uno o varios lenguajes de programación.
- Freelance: persona que trabaja de forma independiente o se dedica a realizar trabajos de manera autónoma que le permitan desenvolverse en su profesión o en aquellas áreas que pueden ser más lucrativas y son orientadas a terceros que requieren de servicios específicos.
- Intellisense: Es una de varias herramientas que permiten completar código inteligente o completar textos inteligentes en diferentes plataformas.
- Chat: Es uno de los métodos de comunicación digital surgido con las nuevas tecnologías. Consiste en la conversación simultánea entre dos o más personas conectadas a la red.

De la aplicación:

- Usuario: Toda persona con acceso a las funciones y recursos de la aplicación.
- Colaborador: Toda persona que, en su condición de usuario, tenga acceso a recursos compartidos (proyectos) por otro usuario.
- Trabajo colaborativo: es la conformación de un grupo de sujetos homogéneos (con conocimientos similares en el tema), donde no surge un líder como en un trabajo de grupo normal, por el contrario, el liderazgo, así como la responsabilidad, es compartido por todos los integrantes.
- Proyecto: Conjunto de archivos de clase, cabeceras, recursos u otros que en combinación forman una aplicación.
- En línea: hace referencia al estado activo de conectividad en internet. Se opone al término fuera de línea (en inglés offline), que indica un estado de desconexión.
- Tiempo real (síncrono): Que se ejecuta a la par (en vivo o sobre la marcha) sin que el usuario experimente un retraso notable.
- Sincronización (de archivos): es el proceso de asegurarse de que dos o más ubicaciones contengan las mismas versiones de los archivos.

Referencias

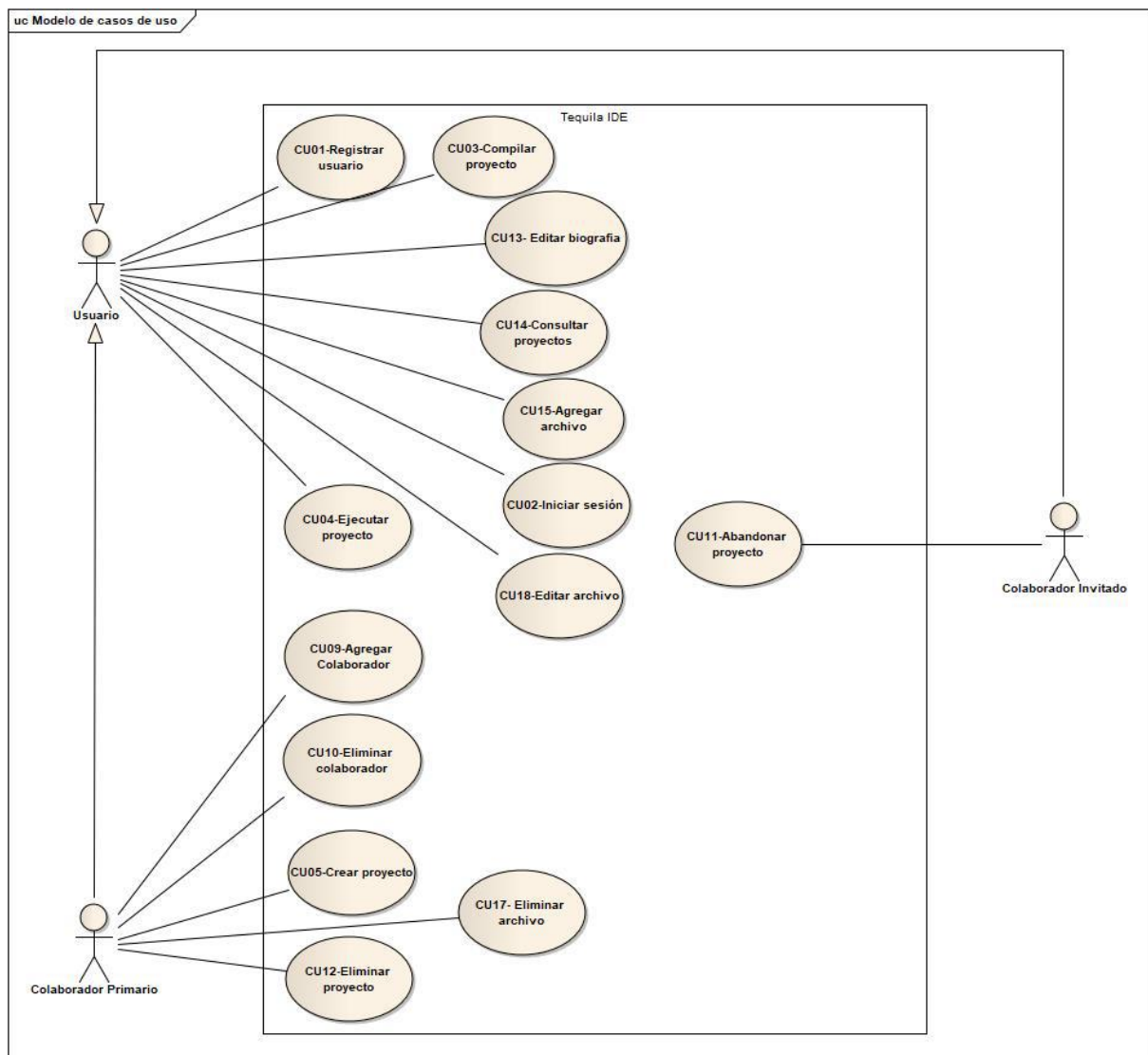
Estándar IEEE 830

Descripción General

Perspectiva del producto

La principal característica del sistema será su capacidad para soportar trabajo colaborativo en línea de varios usuarios de manera síncrona combinado con las herramientas propias de un entorno de desarrollo integrado (IDE), permitiendo a los usuarios ver, modificar, crear, compilar y ejecutar código sobre una misma plataforma. La utilización del sistema pretende crear un ambiente de trabajo en equipo ameno y productivo además de proporcionar una opción a una herramienta que no está disponible en el mercado actual.

Funcionalidad del producto



Características de los usuarios

La aplicación está destinada a ser usada por estudiantes, profesionistas, empleados o trabajadores independientes que, en su condición de programadores, requieran de un entorno de desarrollo integrado, que además de las funciones clásicas (editar, compilar y ejecutar) provea una interfaz amigable y eficiente para el trabajo colaborativo en tiempo real.

| | |
|------------------|---|
| Tipo de usuario: | Estudiante |
| Formación: | Universitario o técnico de carreras afines al desarrollo de software. |

| | |
|--------------|--|
| Habilidades: | <ul style="list-style-type: none"> • Lógico matemática • Conocimientos básicos de hardware y software • Aptitud para el manejo de computadoras • Capacidad de análisis |
| Actividades: | <ul style="list-style-type: none"> • Prácticas de desarrollo • Proyectos escolares individuales • Proyectos escolares grupales • Proyectos de autoaprendizaje |

| | |
|------------------|--|
| Tipo de usuario: | Empleado de una empresa de desarrollo |
| Formación: | Universitario o técnico de carreras afines al desarrollo de software. |
| Habilidades: | <ul style="list-style-type: none"> • Análisis de requerimientos • Diseño de soluciones de software • Construcción de software • Amplio manejo de tecnologías para construcción de software • Análisis y ejecución de pruebas • Evaluación de software • Mantenimiento de software |
| Actividades: | <ul style="list-style-type: none"> • Proyectos de software grupales con fines lucrativos o empresariales • Proyectos de software grupales con fines no lucrativos ni empresariales • Proyectos de software individuales con fines de autoaprendizaje |

| | |
|------------------|--|
| Tipo de usuario: | Trabajador independiente (freelance) |
| Formación: | Universitario, técnico o autodidacta con conocimientos afines al desarrollo de software |
| Habilidades: | <ul style="list-style-type: none"> • Diseño de algoritmos • Construcción de software • Análisis de soluciones |
| Actividades: | <ul style="list-style-type: none"> • Proyectos independientes con fines lucrativos • Proyectos independientes con fines de autoaprendizaje |

Restricciones

- Lenguaje de programación utilizado Java
- Plataforma de diseño JavaFX
- Base de datos MySQL
- Interfaz relativamente fácil de usar para soportar la carga de tareas de un IDE
- El sistema se diseñará siguiendo una arquitectura cliente-servidor
- Internacionalización en 3 idiomas (Español, Inglés y Alemán)
- Límite de colaboradores permitidos por proyecto: 5 usuarios

Suposiciones y dependencias

- Se asume que los usuarios de la aplicación tienen acceso a una conexión a internet con un ancho de banda admisible para satisfacer las necesidades de sincronización en tiempo real.
- Se asume que los usuarios utilizan la aplicación en un ordenador que cumple los requisitos mínimos para su ejecución, sin comprometer las características de la misma. (véase interfaces de hardware).

Evolución previsible del sistema

Identificación de características a incorporar en futuras versiones:

- Tener un chat propio de la aplicación
- Soporte para más lenguajes de programación
- Soporte para Intellisense
- Desarrollo de más temas visuales para la configuración del IDE
- Incorporación de una herramienta para el control de versiones integrada en el sistema
- Incorporación de un tutorial para introducir a nuevos usuarios a las funcionalidades del sistema

Requisitos comunes de las interfaces

Interfaces de usuario

La interfaz de usuario consistirá en un conjunto de ventanas interactivas que abarcan, entre otros elementos, barras de menú, botones, explorador de archivos y áreas de texto. A través de los cuales se podrán guardar, desplegar y modificar los archivos del proyecto solicitado.

Las pantallas de interfaz correspondientes a cada función del sistema podrán ser encontradas en la sección de requisitos funcionales.

Interfaces de hardware

Será necesario disponer de equipos de cómputo en buen estado con las siguientes especificaciones:

- Adaptador de red
- Procesador de 1.66GHz o superior
- Memoria mínima de 256 Mb
- Mouse
- Teclado
- Monitor

Interfaces de software

- El sistema cuenta con una base de datos propia y es independiente de cualquier sistema de software actualmente existente.

Interfaces de comunicación

La comunicación en red es una parte esencial de nuestro sistema. Para la comunicación entre el sistema él y el servidor remoto y viceversa se utilizará el protocolo TCP.

Requisitos Funcionales

Usuario

CU01 - Registrar usuario

| | |
|------------------------|---|
| ID: | CU01 |
| Nombre: | Registrar usuario |
| Descripción: | Permite al usuario crear una cuenta que le proporcionará, además de un identificador único en proyectos colaborativos, acceder al conjunto de funciones del sistema. |
| Precondiciones: | |
| Flujo Normal: | <ol style="list-style-type: none">1. El sistema muestra la ventana GUISignUp que contiene los siguientes elementos:<ol style="list-style-type: none">a. Cuadros de texto para ingresar nombre completo, alias, correo electrónico, contraseña y confirmación de la contraseña.b. Los botones "crear cuenta" e "iniciar sesión".2. El usuario ingresa un alias de su preferencia, nombre completo, correo, contraseña y confirmación de la contraseña.3. El sistema valida los campos, si los campos están correctos y no hay campos vacíos habilita el botón "Crear cuenta". |

| | |
|--------------------------|---|
| | <ol style="list-style-type: none"> 4. El usuario selecciona el botón "Crear cuenta" 5. El sistema genera el registro del nuevo usuario y lo guarda en la base de datos. Al finalizar muestra un cuadro de diálogo con el mensaje "Usuario registrado exitosamente". 6. Termina el caso de uso |
| Flujo Alternativo | <ol style="list-style-type: none"> 2.A.1. El usuario da clic en el botón "iniciar sesión". 2.A.2. Se extiende al CU02 iniciar sesión 2.B.1. El usuario introduce un alias que ya ha sido registrado previamente en el sistema. 2.B.2. El sistema muestra el mensaje "alias ya existente" 2.C.1. El usuario ingresa un correo que ya ha sido registrado previamente en el sistema. 2.C.2. El sistema muestra el mensaje "email ya existente" 2.D.1. El usuario ingresa una contraseña diferente en el campo "confirmar contraseña" a la ingresada en el campo "contraseña" 2.D.2. El sistema muestra el mensaje "las claves no coinciden" 2.E.1. El usuario ingresa una clave sin un dígito 2.E.2. El sistema muestra el mensaje "debes introducir al menos un dígito" 2.F.1. El usuario introduce caracteres no permitidos en los campos nombre o alias 2.F.2. El sistema muestra el mensaje "símbolos inválidos" |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | El usuario debe quedar registrado en el sistema |

CU02 - Iniciar sesión

| | |
|------------------------|--|
| ID: | CU02 |
| Nombre: | Iniciar sesión |
| Descripción: | Permite a un usuario previamente registrado, acceder mediante usuario y contraseña, a su cuenta del sistema para interactuar con las opciones disponibles. |
| Precondiciones: | Que el usuario esté registrado en el sistema |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El sistema muestra la ventana GUILogIn con los siguientes elementos: |

| | |
|-------------------------|---|
| | <ol style="list-style-type: none"> a. los botones "English" y "Deutsche" b. los campos "Alias" y "Contraseña" c. Los botones "Ingresar" y "¿No tienes cuenta? Regístrate" <ol style="list-style-type: none"> 2. El usuario ingresa su nombre de usuario y contraseña y selecciona el botón "Ingresar" 3. El sistema valida los datos ingresados, y abre la ventana GUIFileExplorer 4. Fin de caso de uso |
| Flujo Alterno | <p>2.A.1. El usuario deja algún campo vacío 2.A.2. El sistema muestra el mensaje "debes llenar todos los campos"</p> <p>2.B.1. El usuario introduce un alias no existente 2.B.2. El sistema muestra el mensaje "El usuario no existe"</p> <p>2.C.1. El usuario introduce una contraseña errónea 2.C.2. El sistema muestra el mensaje "Clave incorrecta"</p> <p>2.D.1. El usuario selecciona el botón "English" 2.D.2. El sistema recarga todos los textos de la interfaz gráfica en el idioma inglés</p> <p>2.E.1. El usuario selecciona el botón "Deutsche" 2.E.2. El sistema recarga todos los textos de la interfaz gráfica en el idioma alemán</p> <p>2.F.1. El usuario selecciona el botón "¿No tienes cuenta? Regístrate" 2.F.2. El caso de uso extiende a CU01 Registrar usuario</p> |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | El usuario ingresado al sistema |

CU03 - Compilar proyecto

| | |
|------------------------|--|
| ID: | CU03 |
| Nombre: | Compilar proyecto |
| Descripción: | Permite al usuario traducir el código fuente escrito en el proyecto a lenguaje máquina (o híbrido según sea el caso), dando como resultado una versión ejecutable. |
| Precondiciones: | <ul style="list-style-type: none"> • El proyecto debe estar registrado en la base de datos del sistema. |

| | |
|--------------------------|---|
| | <ul style="list-style-type: none"> • Deben existir archivos de clase, cabecera u otros posibles de compilar asociados al proyecto en la base de datos del sistema. |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El usuario de clic en el botón "compilar proyecto". 2. El sistema recupera de la base de datos los archivos correspondientes al proyecto y ejecuta el proceso de compilación correspondiente al lenguaje de programación sin encontrar errores. Almacena los archivos compilados en el servidor y muestra la ventana GUIConsole que con el mensaje "Compilación exitosa, código de salida 0". 3. Termina el caso de uso. |
| Flujo Alternativo | <p>2.A.1. El sistema ejecuta el proceso de compilación correspondiente al lenguaje del proyecto actual, pero encuentra errores, por lo tanto, muestra la ventana GUIConsole con los mensajes de salida del compilador.</p> <p>2.A.2. Termina el caso de uso</p> |
| Excepciones: | <p>1.- El sistema no se puede conectar a la BD, por lo tanto, muestra el mensaje "Error, intente más tarde"</p> |
| Postcondiciones: | <ul style="list-style-type: none"> • La versión ejecutable del proyecto debe quedar registrada en la base de datos del sistema. |

CU13 - Editar biografía

| | |
|--------------------------|---|
| ID: | CU13 |
| Nombre: | Editar biografía |
| Descripción: | El usuario puede editar su biografía |
| Precondiciones: | Que el usuario este registrado en el sistema |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El sistema muestra la ventana GUIProfile con los siguientes elementos: <ol style="list-style-type: none"> a. los campos alias, correo electrónico y biografía b. los botones "Aceptar" y "Cancelar" 2. El usuario modifica el campo biografía 3. El sistema reconoce el cambio y habilita el botón "Aceptar" 4. El usuario selecciona el botón "Aceptar" 5. El sistema actualiza el usuario con el campo biografía en la base de datos y muestra la alerta con el mensaje "biografía actualizada correctamente" 6. Termina caso de uso |
| Flujo Alternativo | <p>2.A.1. El usuario selecciona el botón "Cancelar"</p> <p>2.A.2. Termina caso de uso</p> |

| | |
|-------------------------|---|
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | La biografía del usuario actualizada en el sistema |

CU04 - Ejecutar proyecto

| | |
|--------------------------|---|
| ID: | CU04 |
| Nombre: | Ejecutar proyecto |
| Descripción: | Permite al usuario tomar la versión ejecutable previamente traducida por el compilador del sistema (de código fuente a máquina) e iniciar la ejecución del programa. |
| Precondiciones: | <ul style="list-style-type: none"> • Debe existir una versión ejecutable del proyecto actual registrada en el servidor del sistema. |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El usuario da clic en el botón "ejecutar proyecto". 2. El sistema muestra la ventana GUIRunProject que contiene: <ol style="list-style-type: none"> a. Un combo para seleccionar la clase principal del proyecto. b. Un checkbox para indicar si el programa requiere parámetros. c. Un cuadro de texto para los argumentos (inicialmente deshabilitado). d. El botón ejecutar programa. 3. El usuario selecciona del combo la clase principal del proyecto y marca el checkbox para indicar que se requieren argumentos. 4. El sistema habilita el cuadro de texto para los argumentos. 5. El usuario agrega los argumentos necesarios y da clic en el botón ejecutar programa. 6. El sistema lleva a cabo y sin errores el proceso de ejecución de los archivos del proyecto previamente compilados que se encuentran en el servidor del sistema con los argumentos proporcionados por el usuario. Luego muestra la ventana GUIConsole con los resultados de la ejecución. 7. Termina el caso de uso. |
| Flujo Alternativo | <p>6.A.1. La ejecución del programa produce excepciones que no generan la finalización.</p> <p>6.A.2. El sistema muestra en la ventana GUIConsole el mensaje de error correspondiente a la excepción producida.</p> |
| Excepciones: | 6.- El sistema no se puede conectar a la BD, por lo tanto, muestra el mensaje "Error, intente más tarde". |
| Postcondiciones: | |

CU18 - Editar archivo

| | |
|--------------------------|--|
| ID: | CU18 |
| Nombre: | Editar archivo |
| Descripción: | Permite al usuario editar el contenido de un archivo y guardarlo en el sistema |
| Precondiciones: | Que el archivo ya este creado |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El sistema muestra la ventana GUITab con el contenido del archivo seleccionado 2. El usuario ingresa texto en el area de la pestaña 3. El sistema guarda el texto en la base de datos, posteriormente emite a todos los usuarios conectados a ese mismo archivo el texto. 4. Termina caso de uso |
| Flujo Alternativo | |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | El contenido del archivo guardado en la base de datos. |

CU15 - Agregar archivo

| | |
|--------------------------|--|
| ID: | CU15 |
| Nombre: | Agregar archivo |
| Descripción: | Permite al usuario agregar un archivo al proyecto seleccionado |
| Precondiciones: | Que haya al menos un proyecto registrado en el sistema |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El usuario selecciona la opción "Agregar archivo" del menu "Archivo" 2. El sistema muestra una alerta con los siguientes elementos: <ol style="list-style-type: none"> a. el campo de la clase b. los botones "Aceptar" y "Cancelar" 3. El usuario llena el campo de la clase y selecciona el botón "Aceptar" 4. El sistema valida el campos, posteriormente guarda el archivo en la base de datos y abre la ventana GUITab 5. Termina caso de uso |
| Flujo Alternativo | <ol style="list-style-type: none"> 2.A.1. El usuario selecciona el botón "Cancelar" 2.A.2. Termina caso de uso |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el |

| | |
|-------------------------|--|
| | mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | El contenido del archivo guardado en la base de datos. |

CU14 - Consultar proyectos

| | |
|--------------------------|---|
| ID: | CU14 |
| Nombre: | Consultar proyectos |
| Descripción: | Permite al usuario consultar los proyectos que se encuentren relacionados a el en el sistema |
| Precondiciones: | Que el usuario tenga proyectos registrados en el sistema |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El sistema muestra la ventana GUIFileExplorer con los siguientes elementos: <ol style="list-style-type: none"> a. un campo de búsqueda para los proyectos b. un área con los proyectos recuperados pertenecientes al usuario c. los botones "Todos", "Mis proyectos" y "Proyectos compartidos" 2. El usuario selecciona el botón "Todos" 3. El sistema muestra en el área de proyectos todos los proyectos recuperados del usuario y los proyectos en los cuales el usuario es colaborador secundario 4. Termina caso de uso |
| Flujo Alternativo | <p>2.A.1. El usuario selecciona el botón "Proyecto compartidos"</p> <p>2.A.2. El sistema muestra en el area de proyectos todos los proyectos en los que el usuario sea colaborador secundario</p> <p>2.B.1. El usuario ingresa un criterio de busqueda en el campo de busqueda</p> <p>2.B.2. El sistema busca en los proyectos uno cuyo nombre coincida con el criterio ingresado por el usuario, si encuentra una coincidencia muestra en el area de proyectos el proyecto el resultado, si no la encuentra, muestra los proyectos pertenecientes al usuario en su lugar.</p> <p>2.C.1. El usuario selecciona un proyecto</p> <p>2.C.2. El sistema abre la ventana GUIEditor</p> |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | Los proyectos del usuario consultados |

Colaborador Principal

CU09 - Agregar colaborador

| | |
|------------------------|---|
| ID: | CU09 |
| Nombre: | Agregar colaborador |
| Descripción: | Permite al colaborador principal agregar a un usuario registrado en el sistema al proyecto actual. |
| Precondiciones: | El número de colaboradores actuales en el proyecto no supera 5 El colaborador está registrado en el sistema El colaborador a agregar debe ser distinto al colaborador principal. |
| Flujo Normal: | <ol style="list-style-type: none">1. El sistema muestra la ventana GUIAddColaborator que contiene:<ol style="list-style-type: none">a. Un cuadro de búsqueda.b. Un area paa el resultado de la busquedac. Los botones "Agregar" y "Cancelar"2. El colaborador principal escribe en el cuadro de búsqueda el alias del colaborador que desea agregar al proyecto3. El sistema consulta de la base de datos y muestra en la lista de colaboradores las coincidencias de usuarios con el criterio introducido por el colaborador principal.4. El colaborador principal selecciona el botón "Agregar"5. El sistema verifica que el número de colaboradores actuales del proyecto no excede los 5 y que el colaborador a añadir no sea el mismo que el colaborador principal, si es asi añade como colaborador al usuario seleccionado. Luego muestra el mensaje "colaboración guardada".6. Termina el caso de uso. |
| Flujo Alterno | <p>2.A.1. El colaborador principal ingresa un alias que no se encuentra registrado en el sistema</p> <p>2.A.2. El sistema muestra el mensaje "no hay concidencias"</p> <p>4.A.1. El colaborador principal da clic en "Agregar" cuando ya hay 5 colaboradores en el proyecto</p> <p>4.A.2. El sistema muestra el mensaje "No puedes agregar más de 5 colaboradores"</p> <p>4.B.1. El colaborador principal da clic en "Agregar" a su mismo usuario</p> <p>4.B.2. El sistema muestra el mensaje "No puedes agregararte a ti mismo"</p> <p>5.A.3. Termina el caso de uso</p> |

| | |
|-------------------------|---|
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | El usuario agregado como colaborador debe quedar asociado al proyecto en el sistema |

CU10 - Eliminar colaborador

| | |
|--------------------------|--|
| ID: | CU10 |
| Nombre: | Eliminar colaborador |
| Descripción: | Permite al colaborador principal eliminar del proyecto a un colaborador. |
| Precondiciones: | Que haya colaboradores secundarios en el proyecto actual |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El usuario selecciona un colaborador secundario de la barra "Colaboradores" 2. El sistema despliega otro menú con las opciones "Información" y "Eliminar colaborador" 3. El usuario selecciona el botón "Eliminar colaborador" 4. El sistema elimina al colaborador secundario, posteriormente retira su icono de la barra "Colaboradores" y muestra la alerta "El colaborador ha sido eliminado". En caso de que el colaborador secundario eliminado se encuentre en dicho proyecto, el sistema le mandara una alerta con el mensaje "Has sido eliminado de este proyecto". 5. El usuario selecciona aceptar 6. Termina caso de uso |
| Flujo Alternativo | |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | El colaborador secundario eliminado del proyecto |

CU05 - Crear proyecto

| | |
|----------------|----------------|
| ID: | CU05 |
| Nombre: | Crear proyecto |

| | |
|--------------------------|--|
| Descripción: | Permite al colaborador principal crear un nuevo proyecto. |
| Precondiciones: | |
| Flujo Normal: | <ol style="list-style-type: none"> 1. El sistema muestra la ventana GUINewProject que contiene los siguientes elementos: <ol style="list-style-type: none"> a. El campo nombre b. Un combobox con 3 lenguajes de programación a seleccionar: Java, C++ y Python. c. Los botones "Crear" y "Cancelar" 2. El colaborador principal escribe el nombre del proyecto. 3. El sistema verifica que el nombre del proyecto no coincida con otro proyecto del colaborador principal y habilita el botón crear. 4. El colaborador principal selecciona un lenguaje de programación y da clic en el botón crear. 5. El sistema genera el nuevo proyecto y lo almacena en la base de datos, luego genera un cuadro de diálogo que solicita el nombre de la primera clase/archivo del proyecto. 6. El colaborador principal escribe el nombre de la primera clase/archivo del proyecto y da clic en aceptar. y muestra la ventana GUIAmbienteDeTrabajo. 7. Termina el caso de uso. |
| Flujo Alternativo | <p>5.A.1. El colaborador principal da clic en el botón cancelar. 5.A.2. Termina el caso de uso.</p> <p>7.A.1. El colaborador principal da clic en el botón cancelar. 7.A.2. Termina el caso de uso</p> <p>7.B.1. El colaborador principal da clic en el botón crear sin haber introducido el nombre del proyecto o la ubicación. 7.B.2. El sistema detecta que los campos están vacíos y muestra el mensaje "Se requiere el nombre y ubicación del proyecto"</p> <p>7.C.1. El colaborador principal da clic en el botón atrás 7.C.2. Regreso al flujo normal en paso 2.</p> |
| Excepciones: | 8.- El sistema no se puede conectar a la BD, muestra el mensaje "Error, intente más tarde" |
| Postcondiciones: | <ul style="list-style-type: none"> • El proyecto debe quedar registrado en la base de datos del sistema |

CU12 - Eliminar proyecto

| | |
|------------|------|
| ID: | CU12 |
|------------|------|

| | |
|--------------------------|--|
| Nombre: | Eliminar proyecto |
| Descripción: | Permite al colaborador principal eliminar un proyecto del sistema |
| Precondiciones: | <ul style="list-style-type: none"> El proyecto debe estar registrado en la base de datos del sistema |
| Flujo Normal: | <ol style="list-style-type: none"> El colaborador principal da clic en el menú "eliminar proyecto". El sistema despliega un cuadro de diálogo con el mensaje "deseas eliminar este proyecto" y los botones "aceptar" y "cancelar". El colaborador principal da clic en el botón "aceptar". El sistema elimina de la base de datos el proyecto actual y muestra el mensaje "El proyecto se ha eliminado exitosamente". Termina el caso de uso. |
| Flujo Alternativo | <p>3.A.1. El colaborador principal da clic en el botón "cancelar".</p> <p>3.A.2. Termina el caso de uso.</p> |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |
| Postcondiciones: | El proyecto debe quedar eliminado de la base de datos del sistema |

Colaborador Invitado

CU11 - Abandonar proyecto

| | |
|--------------------------|---|
| ID: | CU11 |
| Nombre: | Abandonar proyecto |
| Descripción: | Permite al colaborador secundario eliminar sus derechos de ingreso al proyecto. |
| Precondiciones: | Que el colaborador secundario esté en el proyecto |
| Flujo Normal: | <ol style="list-style-type: none"> El usuario selecciona el menú "Archivo" y selecciona "Abandonar proyecto" El sistema muestra la alerta "Deseas abandonar el proyecto?" El usuario selecciona aceptar El sistema elimina el acceso del colaborador secundario al proyecto |
| Flujo Alternativo | <p>3.A.1.- El usuario selecciona el botón cancelar</p> <p>3.A.2.- El sistema cierra la alerta</p> |
| Excepciones: | El sistema no se puede conectar al servidor, por lo tanto, muestra el mensaje "se ha perdido la conexión al servidor" |

| | |
|-------------------------|---------------------------------------|
| Postcondiciones: | La colaboración eliminada del sistema |
|-------------------------|---------------------------------------|

Requisitos no funcionales

Requisitos de rendimiento

| | |
|--------------|---|
| ID: | RC-08 |
| Nombre: | Cantidad de usuarios en tiempo real |
| Descripción: | El sistema debe dar soporte a por lo menos 5 usuarios conectados simultáneamente trabajando en el mismo o diferente proyecto. |

| | |
|--------------|--|
| ID: | RC-09 |
| Nombre: | Cantidad de usuarios registrados |
| Descripción: | El sistema debe dar soporte a por lo menos 100 usuarios registrados en la base de datos. |

| | |
|--------------|---|
| ID: | RC-10 |
| Nombre: | Cantidad de proyectos por usuario |
| Descripción: | El sistema debe dar soporte a por lo menos 2 proyectos por cada usuario registrado en la base de datos. Dando un total mínimo de 200 proyectos. |

Seguridad

| | |
|--------------|---|
| ID: | RC-01 |
| Nombre: | Seguridad de comunicación |
| Descripción: | Las comunicaciones entre cliente y servidor deben estar encriptadas |

| | |
|---------|-------------------|
| ID: | RC-02 |
| Nombre: | Acceso al sistema |

| | |
|--------------|---|
| Descripción: | El acceso al sistema debe ser controlado dependiendo del usuario del que se trate |
|--------------|---|

| | |
|--------------|--|
| ID: | RC-03 |
| Nombre: | Seguridad en la elección de nombre de usuario |
| Descripción: | Cuando un usuario quiera registrarse en el sistema, este debe confirmar que el nombre de usuario a registrar no esté en el sistema y si lo está pedir otro nombre de usuario |

| | |
|--------------|--|
| ID: | RC-04 |
| Nombre: | 3 intentos máximo para ingresar al sistema |
| Descripción: | Si un usuario introduce su nombre de usuario correctamente pero no su password en 3 ocasiones, el sistema bloqueará dicho usuario por un determinado periodo de tiempo |

Fiabilidad

| | |
|--------------|--|
| ID: | RC-11 |
| Nombre: | Almacenamiento de proyectos |
| Descripción: | El sistema debe contar con la capacidad de almacenar de manera fidedigna los proyectos generados por cada usuario. |

| | |
|--------------|---|
| ID: | RC-12 |
| Nombre: | Consulta de proyectos |
| Descripción: | El sistema debe contar con la capacidad de consultar los proyectos generados por cada usuario con integridad, sin alteraciones a los datos. |

Disponibilidad

| | |
|--------------|---|
| ID: | RC-05 |
| Nombre: | El sistema deberá estar siempre disponible |
| Descripción: | El sistema deberá presentar el mínimo de interrupciones |

Mantenibilidad

| | |
|--------------|---|
| ID: | RC-13 |
| Nombre: | Documentación |
| Descripción: | El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible |

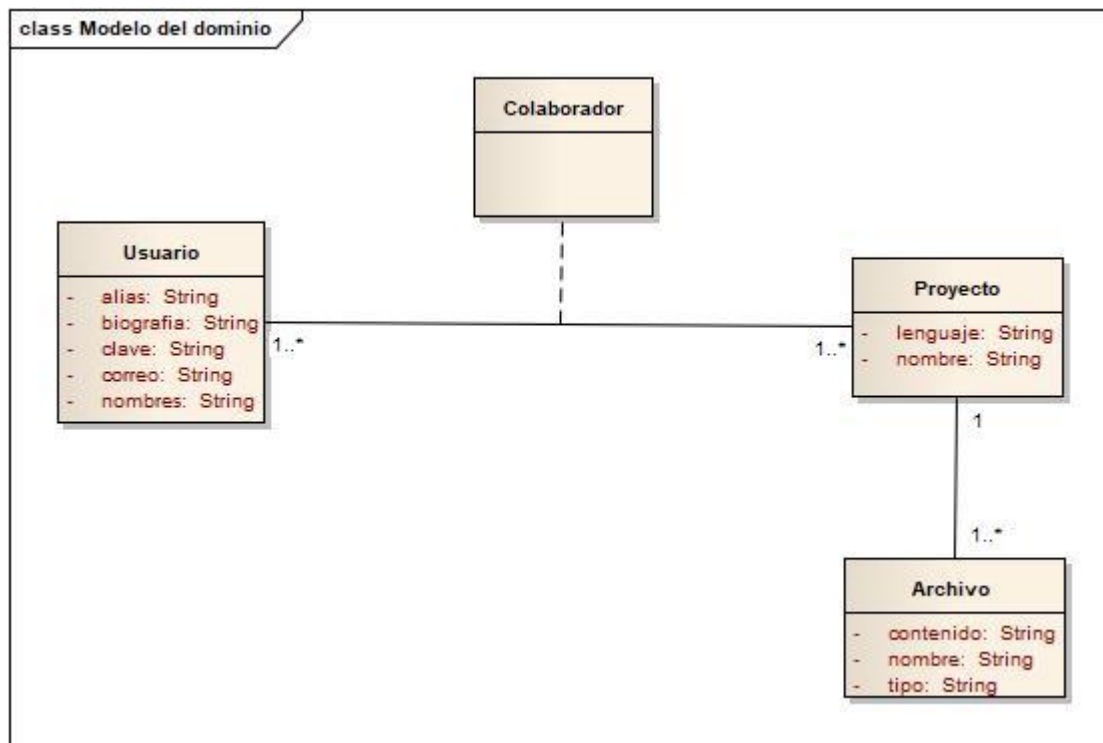
| | |
|--------------|---|
| ID: | RC-14 |
| Nombre: | Guía de estilos |
| Descripción: | El sistema debe disponer de una guía de estilos bien documentada. El código debe apegarse a dicha guía y seguir (aún si no son mencionadas) buenas prácticas de programación. |

Portabilidad

| | |
|--------------|--|
| ID: | RC-07 |
| Nombre: | Software de sistema |
| Descripción: | La aplicación se usará en el sistema operativo Windows |

Apéndices

Diagrama de clases



Modelo de datos

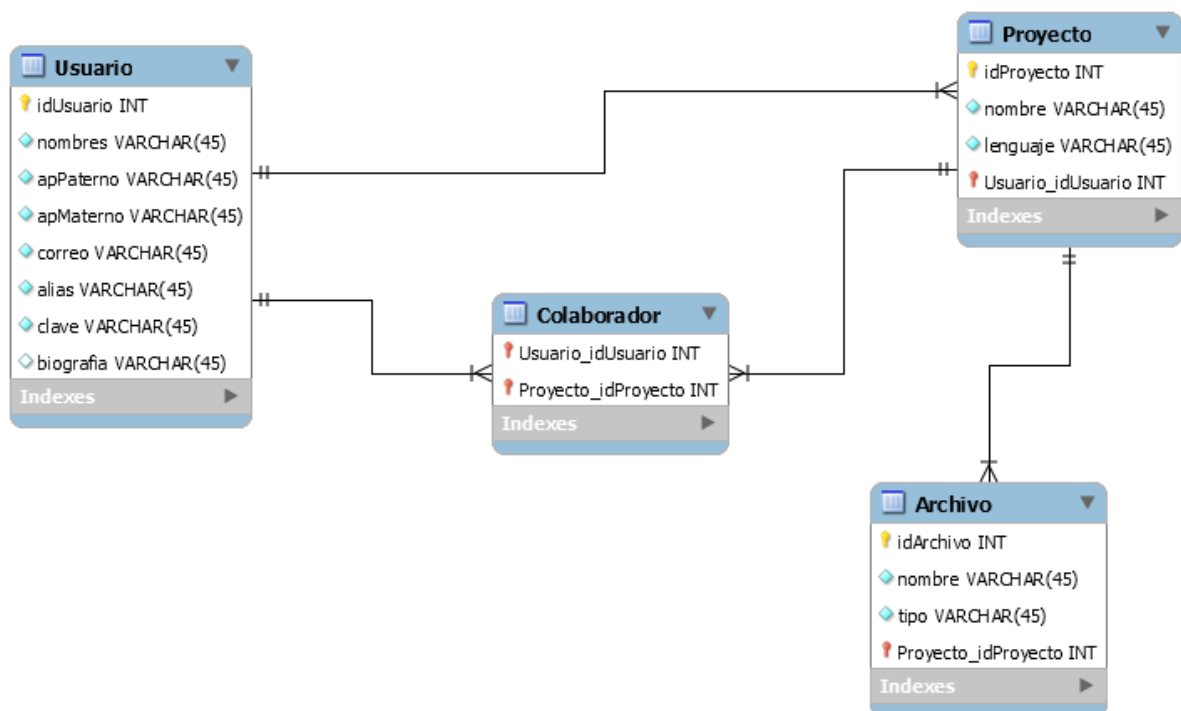
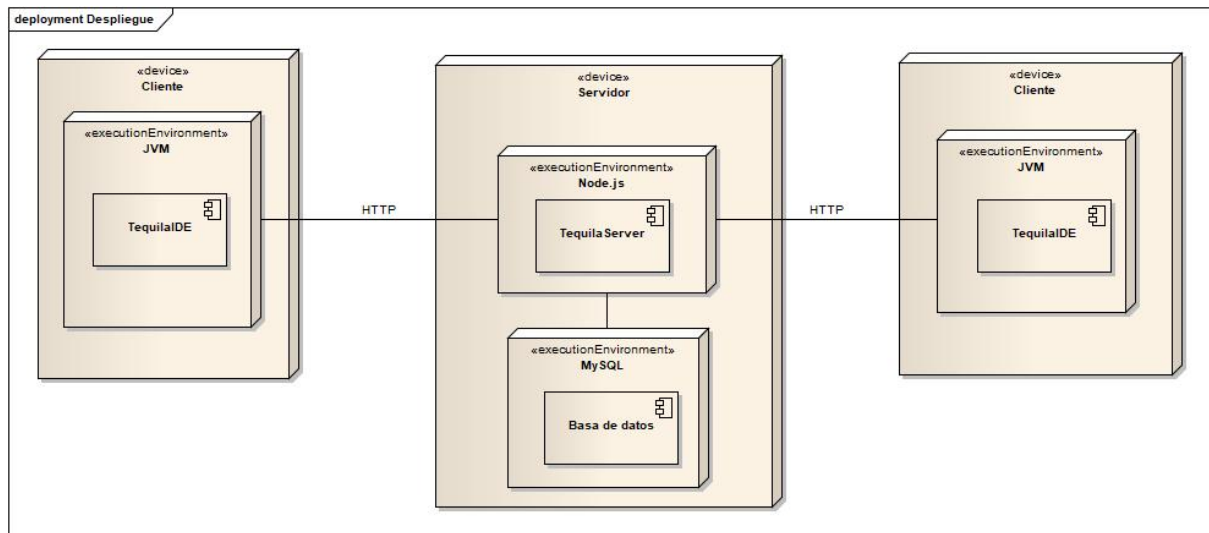


Diagrama de despliegue



Justificación para el uso de frameworks en el proyecto

Node.js / Socket.io

Se decidió utilizar como plataforma de servidor del proyecto Node.js debido a su naturaleza orientada a eventos. Como se presenta en el ERS el proyecto TequilaIDE es de carácter colaborativo y para lograr una interfaz de esta naturaleza y cuyo desempeño sea óptimo para los usuarios finales se necesita de un servidor que maneje eventos en tiempo real de una manera eficiente, y eso es todo lo que encontramos en dicho entorno. Node en conjunto con Socket.io nos permitió manejar eventos simples como conexión y desconexión de una manera que otros frameworks no nos habrían permitido y con una gran simpleza de código, por ejemplo, cuando un usuario se conecta a un proyecto, el servidor reacciona ante este evento y emite una notificación a todos los colaboradores del sistema acerca del suceso, lo que permite una gran versatilidad al codificador para manejar sus respuestas en el código ante el evento. Todo esto, al final se traduce en una mejora en la implementación de los casos de uso establecidos en el ERS.

JFoenix

Se optó por utilizar este componente que implementa el Material de Diseño de Google utilizando componentes de JavaFX debido a nuestro deseo de agregar una interfaz amigable y sofisticada con el usuario. JFoenix nos proporcionó componentes gráficos nativos de JavaFX pero con mayores opciones de personalización para crear una pantalla más agradable a la vista.

Conclusiones

Miguel Alejandro Cámara Árciga

Conclusión

Este proyecto me enseñó a ser precavido con mis habilidades y a tener una mejor medida personal acerca de mis capacidades y hasta donde puedo llegar. Sinceramente no esperaba la complejidad de este sistema, esto debido a que no supe pronosticar las implicaciones de su desarrollo y no supe organizarme acorde, todo esto se vio reflejado en el retraso presentado a la hora de la entrega del mismo. Sin embargo, habiendo presentado dichos argumentos, me siento orgulloso del trabajo final, siento que a pesar de haber administrado mal mis recursos en el desarrollo de este proyecto, mis habilidades son suficientes para esta experiencia educativa.

Lecciones aprendidas

Con respecto a la impartición de este curso, me parece importante recalcar que la dinámica de aprendizaje fue algo que no esperaba, algo inusual para las materias del área disciplinar que hemos estado llevando desde el tercer semestre, sin embargo, fue una experiencia que me agradó bastante, fue una manera diferente de aprender. La mayoría de los temas vistos en clase debían ser investigados en primera instancia por nosotros, y después el profesor nos daba una introducción al tema, fue algo que me agrado pues ya cuando llegaba la explicación del maestro, nosotros ya teníamos una idea de lo que estaba hablando y la explicación era más útil.

Una de las lecciones más difíciles que he aprendido en este curso (aunque he de decir no es exclusivo de este) es que la clave para entregar un proyecto en tiempo y forma y que cumpla con las características requeridas, es la administración del tiempo, la administración es la clave de todo, es más importante me atrevería a decir, que las habilidades técnicas del programador. Es algo que he visto mucho, personas con muchas habilidades y mucha capacidad, pero cuya capacidad de administración es deficiente, tienden a fallar mucho en las materias, cuando otras personas (como yo) que no poseemos tales capacidades obtenemos lo suficiente para aprobar. Se que he dado muchas vueltas con respecto a la administración del tiempo como una lección aprendida en esta conclusión, y es con un propósito, las huellas que ha dejado la entrega final de este curso han sido profundas y con toda sinceridad, espero no repetir esta situación, al menos no debido a una mala organización o mala administración de tiempo.

Alan Yoset García Cruz

Conclusión

Sobre la elaboración de este proyecto (TequilaIDE), un entorno de desarrollo integrado colaborativo, puedo decir que uno de los retos principales a abordar fue el aprendizaje de diferentes tecnologías, el choque en seco de la completa ignorancia de las herramientas, frameworks y plataformas necesarias contra un limitado periodo de entrega de apenas unos meses. Por motivos de diversa índole, desde un cronograma de actividades mal ejecutado,

hasta los días consumidos por otros proyectos el tiempo requerido para finalizar este programa se vio sumamente afectado.

El trabajo de investigación fue exhaustivo y, aunque fuera de tiempo, se alcanzaron las metas deseadas. A pesar de todo, estoy satisfecho con el producto pues cumple con lo estipulado y me aportó nuevos saberes que en un futuro me permitirán abordar proyectos de una manera más eficiente.

Lecciones aprendidas

Respecto a las lecciones aprendidas en este curso podría dividir las en dos categorías, las lecciones aprendidas a nivel personal:

Trabajo en equipo. Aunque desde el inicio de la carrera se ha trabajado en equipo, para esta materia la carga total se repartió únicamente entre dos personas, lo que me permitió aprender más de mi compañero y, de la cantidad de tareas que implica un proyecto de este tipo.

Perder el miedo a aprender. Siempre fue un placer preguntar mis dudas al profesor de esta clase, no sólo porque las respondía con buen ánimo, sino porque la respuesta misma era justo lo que esperaba de un buen académico, algo claro y conciso.

Lecciones aprendidas a nivel académico:

Frameworks es lo primero que me viene a la mente, seguido de sockets, entornos de desarrollo integrado y análisis estático de código.

Frameworks, las aplicaciones semi completas que utilizamos para el desarrollo de grandes sistemas que nos evitan la necesidad de "reinventar la rueda". Sockets, mecanismos que nos permiten comunicar aplicaciones en red (demasiado variados como para explicarlos a detalle), entornos de desarrollo integrado, aplicaciones que reúnen todos los programas que un programador necesita para generar otras aplicaciones. Análisis estático de código, estrategia de revisión de programas utilizada por grandes empresas para garantizar el ahorro de recursos en el desarrollo de software, así como la calidad de los productos.