



Universidad Veracruzana

Sistema para la gestión de Equipo de Cómputo: Obelisco

Experiencia Educativa: Desarrollo de Software
Licenciatura en Ingeniería de Software

Equipo:

Luis Eduardo Aguirre Fuentes

Itzel Martínez Hernández

Díaz Hernández Héctor Joaquín

Sergio Hernández Gómez

Carlos de Jesús Kubota Pérez

Ronaldo Jaime Hernández

Arturo Alejandro Reyes García

José Antonio Sánchez Urieta

Profesor: Irwing A. Ibáñez Castillo

Abril – 2020

Índice

1. Introducción	1
1.1 Objetivos y antecedentes	1
1.2 Metodología	1
1.3 Plan de trabajo	2
2. Tecnologías y herramientas usadas.....	3
2.1 Tecnologías.....	3
2.1.1 C#	3
2.1.2 MySQL	3
2.1.5 Sockets	3
2.2 Herramientas.....	4
2.2.1 Visual Studio.....	4
2.2.2 MySQL Workbench	4
2.2.3 GitHub	4
2.2.4 Adobe XD.....	4
2.2.5 LucidChart	4
2.2.6 Enterprise Architect.....	4
2.2.7 Trello	5
3. Especificación de requisitos	5
3.1 Captura de requisitos	5
3.2 Requisitos funcionales.....	5
3.3 Requisitos no funcionales.....	9
4. Diseño del sistema	9
4.1 Diseño arquitectónico	9
4.2 Diseño detallado	10
5. Pruebas	12
6. Implementación	12
6.1 Aplicación Cliente maestro/Servidor	12
6.2 Aplicación Cliente esclavo	17
6.3 Protocolo de comunicación	20
6.4 Gestión de la configuración	21
7. Métricas	21
7.1 Métricas Orientadas a la Función	21

7.2 Análisis de Puntos Función	22
7.2.1 Factor de Ajuste	22
7.2.2 PFA (Puntos de Función Ajustado).....	23
7.3 Métricas Orientadas al Tamaño.....	24
7.3.1 COCOMO: Modelo constructivo de costes	24
7.3.2 COCOMO: Modelo Básico.....	24
7.3.2 Estimación de Esfuerzo.....	26
8. Conclusiones y trabajos futuros	27
8.1 Conclusiones	27
8.2 Trabajos futuros	28
Bibliografía	29

1. Introducción

En este apartado se explicará y expondrá brevemente el contexto en el cual se ha realizado el presente trabajo como proyecto de la experiencia educativa de Desarrollo de Software, sus objetivos y metodología usada. De igual manera se presentará el plan de trabajo seguido y un estimado del presupuesto requerido.

1.1 Objetivos y antecedentes

Este proyecto es una propuesta para la mejora del sistema usado en el área de cómputo de la Universidad Veracruzana campus Coatzacoalcos, el cual, tiene como objetivo diseñar e implementar un sistema informático para la gestión y administración de los equipos de cómputo usados por los estudiantes, maestros y personal académico.

Actualmente dichos equipos se gestionan con un sistema hecho en Visual Basic, que muestra los equipos disponibles, así como las aulas y los usuarios registrados en el sistema y con su respectivo saldo disponible, todo de manera manual realizado por un encargado, lo que conlleva a un trabajo laborioso y lento.

Como una alternativa, se pensó en simplificar y automatizar la asignación de computadoras y el manejo de ellas sin la necesidad de que un moderador intervenga en ese proceso. De esta forma los clientes (Estudiantes, Maestros y Personal académico) puedan operar con los equipos rápida y eficazmente.

Por lo tanto, además del sistema para los clientes, será necesaria de la implementación de una base de datos para almacenar la información de los usuarios, así como del historial de uso de los equipos para que los responsables tengan disponible dicha información de manera fácil y rápida.

1.2 Metodología

En este proyecto se ha optado por seguir una metodología de desarrollo ágil debido al tiempo de desarrollo dado y porque el cliente fue accesible en ese aspecto.

Al contar con un equipo de desarrollo pequeño (8 personas), se manejó la metodología de desarrollo Scrum, dicho proceso, el cual se puede observar gráficamente en la figura 1, explica brevemente a continuación.



Ilustración 1. Diagrama de trabajo de metodología ágil

En primer lugar, se obtuvieron los requisitos iniciales del sistema a través de una reunión con el cliente, en los cuales se explicaron las funcionalidades deseadas para el sistema y otros aspectos importantes del mismo. A continuación, se ordenaron los requisitos y se seleccionaron las funciones prioritarias para la primera versión del producto. Se comenzó el desarrollo del producto, primero el modelado del software, así como sus diseños e implementándolos. Antes de su despliegue, se realizaron métricas para la estimación de costo para dar una idea al cliente de la complejidad del producto final. Se realizaron pruebas unitarias por parte de los desarrolladores y, una vez superadas, se construyó un desplegable del producto para la prueba y validación por parte del cliente.

Este proceso se realizó en repetidas ocasiones antes de la entrega de este plan y se continuará realizando hasta finalizar el proyecto.

1.3 Plan de trabajo

Al utilizar una metodología ágil en iteraciones, el proyecto completo no debería tener una duración fija. Sin embargo, el proyecto se estimaron tres iteraciones, lo cual supondrá un total de 30 días de trabajo. Por cada iteración extra que posteriormente se desee añadir, se estimaron los días de trabajo ya que esto depende de la cantidad y complejidad de la nueva funcionalidad añadir. Y así. Hasta completar el sistema. En la siguiente figura se puede apreciar el desglose de tareas para las iteraciones especificados para el proyecto.

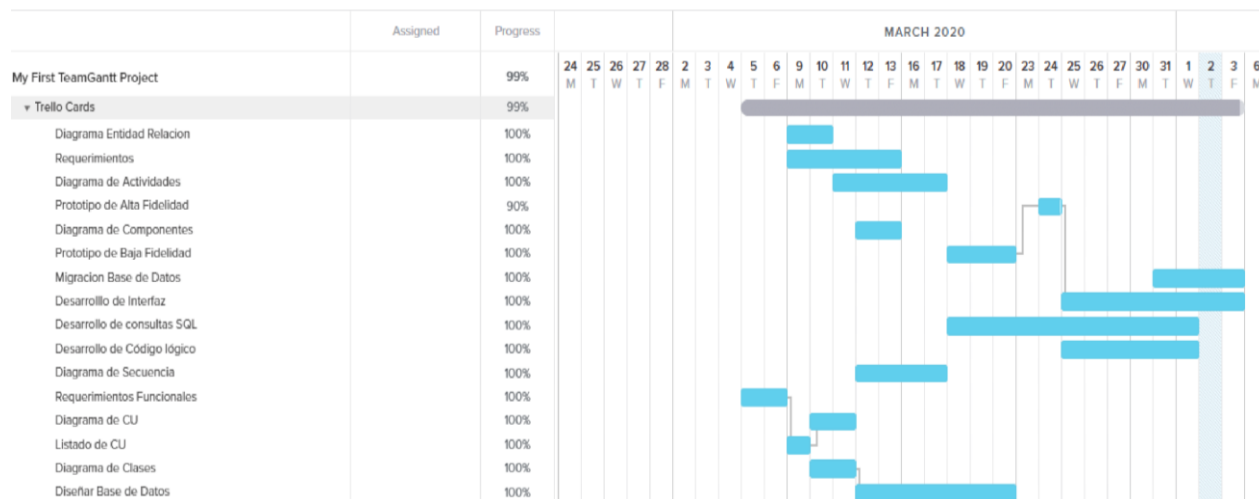


Ilustración 2. Diagrama de Gantt que recoge el plan de trabajo

Respecto al presupuesto, este trabajo se ha realizado con fines académicos por lo que la aportación económica es meramente informativa.

2. Tecnologías y herramientas usadas

En esta sección se describirán las tecnologías y herramientas usadas a lo largo de todo el proyecto.

2.1 Tecnologías

2.1.1 C#

C# o C Sharp es un lenguaje de programación orientado a objetos, desarrollado por Microsoft. Su sintaxis deriva de C/C++, y posee mucha similitud con Java. (Delgado, 2017)

2.1.2 MySQL

MySQL es un sistema gestor de bases de datos relacional, multihilo y multiusuario. Es muy conocido y utilizado debido a las ventajas que aporta, entre las que destacan ser open source, instalarse en multitud de sistemas operativos y permitir una fácil integración con algunos de los lenguajes de programación más utilizados como puede ser Java. (B., 2019)

2.1.5 Sockets

Un socket representa un canal de comunicación entre dos programas, los cuales pueden estar situados en distinto entorno de ejecución, siempre y cuando se tenga comunicación entre ellos. Un socket se define con la dirección IP y el puerto de los programas de ambos extremos. (techopedia, 2011)

2.2 Herramientas

2.2.1 Visual Studio

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. (EcuRed, s.f.)

2.2.2 MySQL Workbench

MySQL Workbench permite diseñar, modelar, generar y administrar bases de datos visualmente. Incluye todo lo que un modelador de datos necesita para crear modelos complejos de ER, ingeniería directa e inversa, y también ofrece características clave para realizar tareas de documentación y gestión de cambios difíciles que normalmente requieren mucho tiempo y esfuerzo. (Amoedo, s.f.)

2.2.3 GitHub

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. (Wikipedia, 2020)

2.2.4 Adobe XD

Adobe XD, o Adobe Experience Design, es una herramienta de edición de gráficos que funciona para crear interfaces de páginas web y de aplicaciones. Permite al diseñador enfocarse en la experiencia del usuario al navegar, con un rango mínimo de error y en el menor tiempo posible. (Molera, 2019)

2.2.5 LucidChart

Lucidchart es una herramienta de diagramación basada en la web, que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas, esquemas de sitios web, diseños UML, mapas mentales, prototipos de software y muchos otros tipos de diagrama. (Lucidchart, s.f.)

2.2.6 Enterprise Architect

Es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. (Pillaca, s.f.)

2.2.7 Trello

Trello es una de las aplicaciones de gestión de proyectos más utilizadas tanto a nivel profesional como personal. Destaca su disposición en forma de tableros virtuales como una especie de 'post-its', o notas donde puedes ir creándolos y organizándolos por prioridades, fechas y con un amplio campo de anotaciones. (ExpertosNegociosOnline, 2019)

3. Especificación de requisitos

Esta sección recoge los requisitos tanto funcionales como no funcionales del sistema a desarrollar.

3.1 Captura de requisitos

Para capturar los requisitos del sistema, se ha optado por realizar una entrevista o reunión, al ser el cliente una persona y tres personas encargadas de tomar dichos requerimientos. Después se procedió a redactar y validar estos requisitos con el cliente previo a su implementación. En futuras reuniones de seguimiento, muy breves, se matizaban y aclaraban ciertos requerimientos que se apuntaban e incorporaban en la siguiente iteración.

La tabla final y enlistada de los requisitos funcionales y no funcionales del sistema a desarrollar se especifica a continuación.

3.2 Requisitos funcionales

Los requisitos funcionales del sistema indican como debe comportarse y responder el mismo. Para contextualizar y que se pueda entender de forma más sencilla el funcionamiento de los casos de uso, a continuación, se procederá a explicar en lenguaje natural el funcionamiento de los mismos.

- **Iniciar sesión:** El sistema permite a los usuarios Consumidores, Administradores y Moderadores, ingresar mediante una matrícula o Núm. De personal y una contraseña.
- **Cerrar sesión:** Para dejar de usar el equipo, es necesario cerrar la sesión previamente abierta. Incluso este caso de uso se activará en caso de que se bloquee el equipo de manera remota o por el exceso de tiempo.
- **Registrarse:** El sistema permite a los usuarios registrarse manualmente para que su usuario quede almacenado en la Base de Datos y pueda ingresar más adelante.
- **Usar computadora:** Los equipos permanecerán bloqueados hasta que un cliente inicie sesión en ellos y si tienen saldo suficiente podrá optar por usar el equipo o imprimir.

- **Imprimir:** El sistema da la ventaja de desbloquear el equipo por 5 minutos para realizar pequeñas operaciones y enviar documentos para imprimir (que no sean a color).
- **Consultar saldo:** Los usuarios registrados poseen de saldo disponible para realizar operaciones como usar el equipo o imprimir, este saldo se muestra cuando el cliente inicia sesión.
- **Agregar saldo:** El usuario moderador es el encargado de realizar esta tarea, pues requiere que los clientes hayan realizado el pago en ventanilla. El moderador confirma la acción y agrega saldo a la cuenta del cliente.
- **Consultar Folio:** El moderador puede consultar el folio de pago que se realizan. Esta información es meramente informativa y para llevar un registro de todos los pagos realizados.
- **Consultar Historial:** El sistema permite a los administradores y moderadores ver el historial de uso de un equipo, mostrando el usuario que la utilizó, la fecha, con hora, de inicio y fin.
- **Consultar equipo:** El sistema da la ventaja de visualizar el estado de los equipos, este puede ser Disponible, Ocupado, o En espera.
- **Asignar equipo:** Si se desea se puede asignar un equipo a usuario de manera manual, esto solo con equipos en estado Disponible. Por lo general esta acción se hará de manera automática.
- **Bloquear equipo:** Un usuario de tipo Moderador o Administrador tiene la posibilidad de bloquear (deshabilitar su uso) de un equipo de manera remota.
- **Validar Consumidor:** Completado el registro de una cuenta de tipo consumidor (Estudiante, Maestro o personal académico) un Moderador puede confirmar el registro de manera manual y presencial para que consumidor pueda usar el equipo.
- **Agregar equipo:** Un administrador tiene la posibilidad de agregar un equipo al sistema, con su nombre, modelo y descripción de este. Se generará un identificador que deberá ser ingresado en el equipo en cuestión para confirmar el registro.
- **Deshabilitar cuenta:** El sistema permite dar de baja o deshabilitar una cuenta de un consumidor ingresando solo su matrícula o Núm. de personal. Y así denegar el acceso de ese consumidor al sistema. La cuenta de consumidor solo tiene de 6 años de vida, pasado ese tiempo, se deshabilitan de manera automática.

El conjunto de requisitos funcionales que el sistema debe cumplir se recoge en una tabla, clasificadas por el Actor (quien realiza la acción), un Identificador y el nombre dicho Caso de Uso.

Listado de Casos de Uso		
Actores	Identificador	Caso de Uso
Consumidor	CU01	Iniciar Sesión
	CU02	Cerrar Sesión
	CU03	Registrarse
	CU04	Usar computadora
	CU05	Imprimir
	CU06	Consultar Saldo
Moderador	CU07	Agregar Saldo
	CU08	Consultar Folio
	CU09	Consultar Historial
	CU10	Consultar Equipo (PC)
	CU11	Asignar equipo
	CU12	Bloquear equipo
	CU13	Validar consumidor
Administrador	CU14	Agregar Equipo (PC)
	CU15	Deshabilitar Cuenta

Tabla 1. Listado de Casos de Uso del sistema

Actor	ACTOR 1. Consumidor
Descripción	Es el actor de usuario más bajo, por lo que sus privilegios son limitados. Es el usuario dedicado para los clientes del sistema.
Responsabilidades	<ul style="list-style-type: none"> ● Iniciar Sesión ● Cerrar Sesión ● Registrarse ● Usar computadora ● Imprimir ● Consultar Saldo
Fuentes	Luis Eduardo Aguirre Fuentes

Tabla 2. Descripción del Actor Consumidor y sus responsabilidades

Actor	ACTOR 2. Moderador
Descripción	Es el actor de usuario que posee privilegios para editar datos de consumidores y manipular los equipos registrados. Hereda las responsabilidades del ACTOR 1 (Consumidor).
Responsabilidades	<ul style="list-style-type: none"> ● Agregar Saldo ● Consultar Folio ● Consultar Historial ● Consultar Equipo (PC) ● Asignar ● Bloquear ● Validar
Fuentes	Luis Eduardo Aguirre Fuentes

Tabla 3. Descripción del Actor Moderador y sus responsabilidades

Actor	ACTOR 3. Administrador
Descripción	Es el actor de usuario más alto con privilegios para administrar el sistema en su totalidad. Hereda las responsabilidades de ACTOR 1 (Consumidor) y ACTOR 2 (Moderador).
Responsabilidades	<ul style="list-style-type: none"> ● Agregar Equipo (PC) ● Deshabilitar

Fuentes	Luis Eduardo Aguirre Fuentes
----------------	------------------------------

Tabla 4. Descripción del actor Administrador y sus responsabilidades

3.3 Requisitos no funcionales

Los requisitos no funcionales imponen restricciones de diseño e implementación relativas al rendimiento, disponibilidad y seguridad, entre otros aspectos.

Los requisitos no funcionales del sistema a desarrollar se recogen en la siguiente tabla.

Identificador	Descripción
RNF-01	El sistema debe soportar una carga de usuarios concurrentes de hasta 30 usuarios
RNF-02	El sistema debe ser fácilmente mantenible
RNF-03	Las interfaces a usuario a desarrollar deben ser intuitivas y de fácil aprendizaje para los usuarios finales.

Tabla 5. Requisitos no funcionales del sistema

4. Diseño del sistema

Basados en los requisitos capturados previamente, se debe proceder al diseño de los modelos de software que ayuden a su construcción.

En este apartado se mostrará el diseño arquitectónico y el diseño detallado especificados para este sistema.

4.1 Diseño arquitectónico

El diseño arquitectónico consiste en la aplicación de patrones y decisiones sobre como estructurar el código para cumplir los requisitos del sistema, así como facilitar el entendimiento y simplicidad del mismo, para futuros mantenimientos.

Al desarrollarse dos partes de un producto de software (cliente de escritorio [maestro] y cliente de escritorio [esclavo]), cada uno tendrá su propia arquitectura.

El cliente de escritorio maestro fungirá como servidor local y se diseñará siguiendo una arquitectura en tres capas, en las que se distinguen:

- **Acceso a datos:** Conecta la aplicación con la base de datos y realiza operaciones sobre ella
- **Operaciones:** Procesa las llamadas a las interfaces y ejecuta la lógica de operaciones necesarias en cada caso para dar soporte a las peticiones.

- **Presentación:** Interfaces necesarias para la administración del sistema

El cliente de escritorio esclavo se configura en 2 capas, que son:

- **Conexión:** Contiene la representación de las interfaces del servidor y la implementación de las comunicaciones de red con el mismo.
- **Presentación:** Contiene las interfaces de usuario y los mensajes que se muestran al usuario al realizar acciones sobre elementos del mismo (pulsar un botón, alertas en pantalla, etc.).

4.2 Diseño detallado

Una vez que la arquitectura esté hecha, se procede a especificar el diagrama de despliegue y el modelo de datos del sistema.

En breve, se muestra el diagrama de despliegue del sistema (Figura 3).

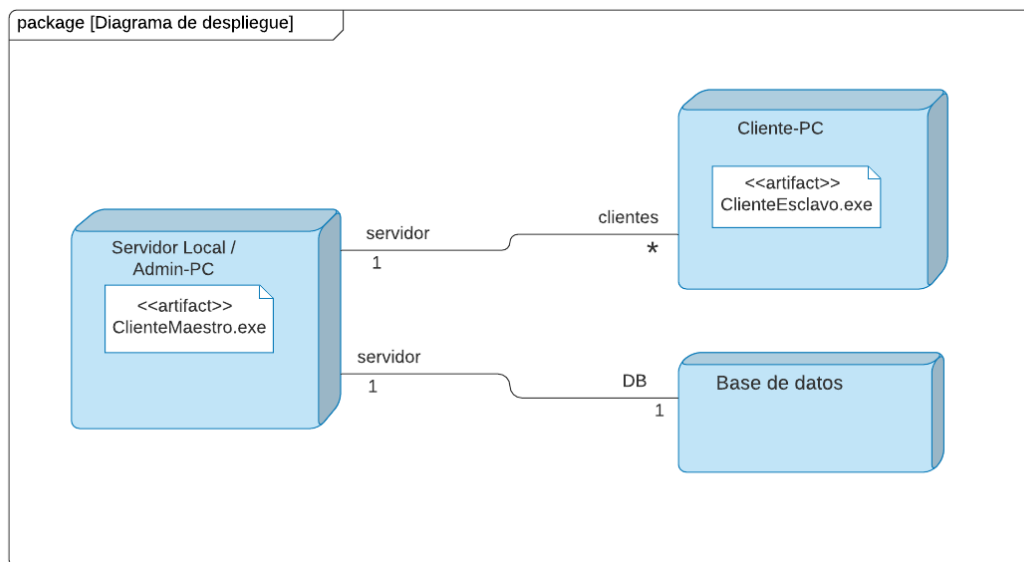


Ilustración 3. Diagrama de despliegue del sistema

En este diagrama se observa tres módulos:

- El primero, el Servidor Local/Admin-PC, alojará la aplicación servidora, que da soporte a la otra aplicación esclava. Así como, el equipo Administrador en el cual se gestionará los datos de los consumidores. Se dispondrá de únicamente un servidor local.
- El segundo modulo corresponde al equipo cliente. Se trata de un PC en el cual se ejecutará el exe correspondiente a la aplicación para participar como consultores de la base de datos y es la que usaran los consumidores

- El tercer modulo corresponde a la base de datos, esto porque se implementará una base de datos externa a la que el equipo responsable maneja y es donde se tendrá los registros de las transacciones y de los usuarios registrados.

Para el diagrama de base de datos para el almacenamiento de los datos está especificado en la siguiente figura (Figura 4).

Se ha diseñado de forma que sea lo más flexible posible a la hora de incluir nuevos usuarios (Consumidores) y dar soporte a los tipos de consultas necesarias. Para el caso de los usuarios nuevos, solo será necesario crear un nuevo registro en la tabla de usuarios y después validar el registro de manera manual por el moderador.

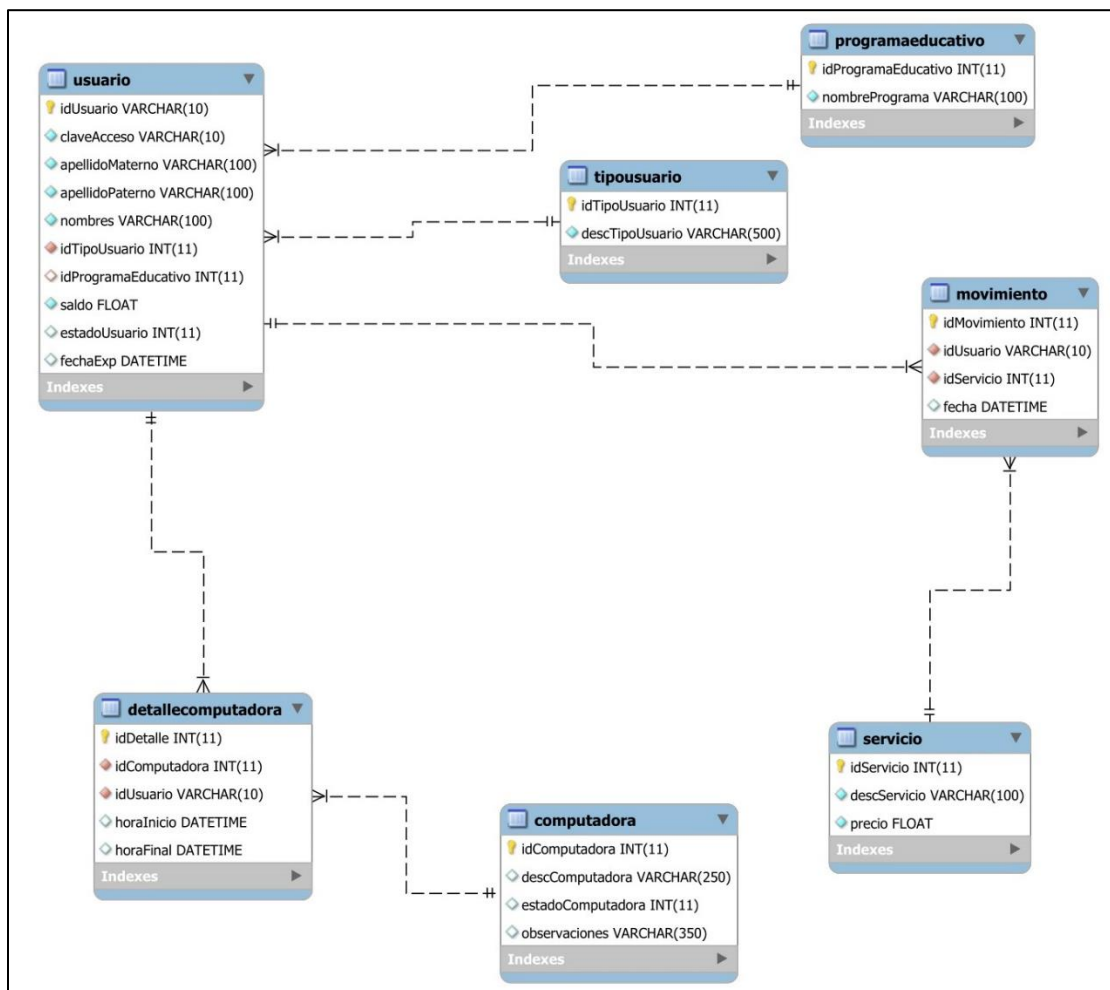


Ilustración 4. Modelo de datos del sistema

Para el modelo de clases del sistema, se dispone de un modelo parecido, con muy pocas variaciones, pero manteniendo la cohesión, y agregando clases nuevas como el “Detalle Computadora” que evita la cardinalidad muchos a muchos y hacer el sistema funcioné mejor.

5. Pruebas

Las pruebas unitarias realizadas nos sirvieron para probar cada parte del código y comprobar la funcionalidad. Se comprobó errores que fueron resueltos de forma rápida y no entorpeció nuestro proceso ágil de desarrollo.

Cada componente del sistema fue probado, manejando diferentes situaciones con diferentes métodos esperando un resultado final. Se escriben los datos de entrada que servirán para probar y esperando el resultado que nos arroje podemos corroborar si es el resultado final esperado. Se hicieron observaciones para ayudar a solucionar cualquier tipo de situaciones.

Los archivos de pruebas se agregaron en la carpeta *Pruebas de Software* de este proyecto.

6. Implementación

Una vez capturadas los requisitos y especificado el diseño, es turno de la implementación de los componentes de softwares requeridos.

Un punto a aclarar es que, este proyecto formaba parte de una evaluación para una experiencia educativa, sin embargo, por motivos ajenos a la universidad, no se ha podido implementar el sistema en el área de producción, es decir, entregárselo al cliente.

Nuestra solución temporal fue implementar de forma remota, alojamos la base de datos en un sitio web dedicado a realizar pruebas de implementación (db4free.net).

Por lo tanto, se abordarán los subtemas de la implementación como una hipótesis y en futuro actualizar estos datos, una vez entregado el producto final al cliente.

6.1 Aplicación Cliente maestro/Servidor

En el caso de codificación de la parte Servidor, como se explico anteriormente en el apartado 4.1, se divide en 3 capas. Como el cliente nos permitió utilizar el equipo Cliente Maestro como servidor, se dispondrá de un servidor local, además, como se dispone de una red local, la conexión y la comunicación de datos hacia el Servidor será de esa manera.

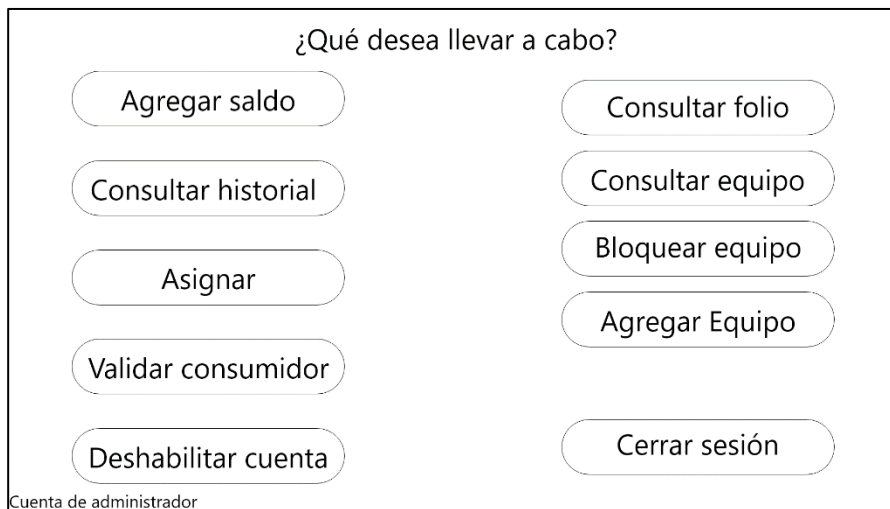


Ilustración 5. Prototipo de Interfaz de vista Administrador

Para la implementación del cliente maestro (Administrador) se ha utilizado C# y la librería Forms para el diseño de interfaces gráficas.

En donde la capa de presentación contiene las interfaces gráficas que el usuario visualizará al utilizar la aplicación, mientras que la capa de conexión contiene la implementación de las operaciones utilizando la comunicación de sockets.

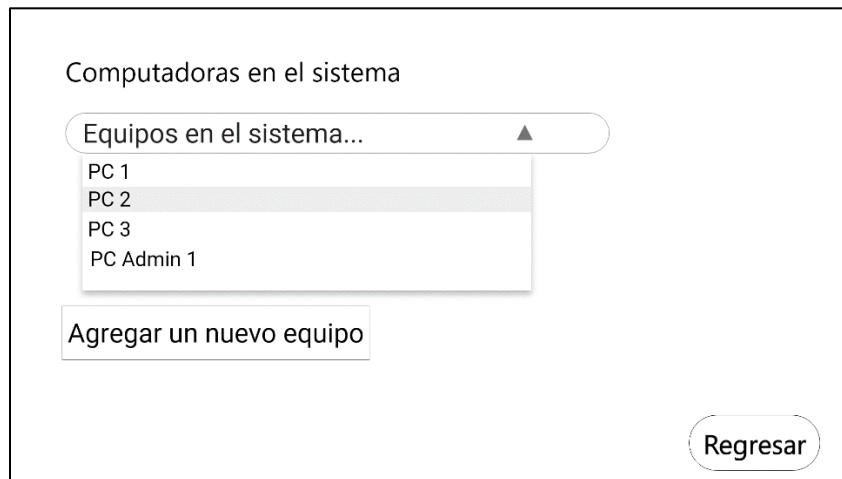
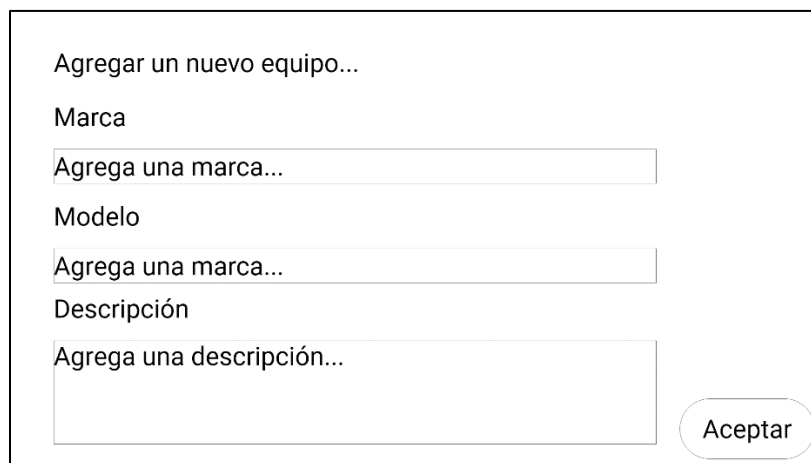


Ilustración 6. Vista de la acción Agregar equipo

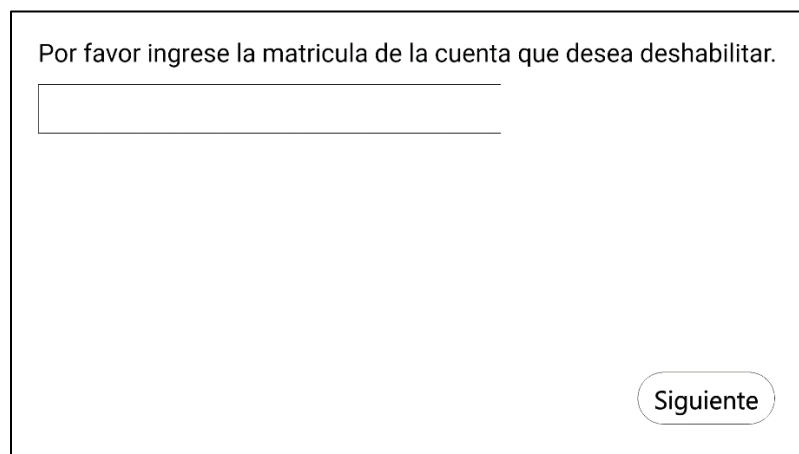
Si selecciona la opción de “Agregar un equipo” se le mostrará un menú como la ilustración 6. En donde se pondrá ver los equipos previamente registrados en el sistema e incluso agregar uno nuevo, con solo hacer click en la opción “Agregar un nuevo equipo”. El sistema pedirá una confirmación.



Formulario de registro de equipo. El formulario está encerrado en un recuadro rectangular. En la parte superior, el título "Agregar un nuevo equipo..." está en un color gris oscuro. Debajo del título, hay tres campos de entrada de texto, cada uno con un encabezado en gris: "Marca", "Modelo" y "Descripción". Los campos de texto tienen un fondo blanco y un borde gris. El campo de "Descripción" es más grande que los otros dos. En la parte inferior derecha del recuadro, hay un botón redondeado con el texto "Aceptar" en un color gris oscuro.

Ilustración 7. Vista de formulario de registro de equipo

El sistema mostrará un formulario como en la imagen 7, en donde solicita al Administrador que ingrese los datos de registro para agregar un nuevo equipo a la base de datos del sistema. Llenado todos los campos, se confirmará la acción con un mensaje de éxito en pantalla.



Formulario de deshabilitación de cuenta. El formulario está encerrado en un recuadro rectangular. En la parte superior, el texto "Por favor ingrese la matricula de la cuenta que desea deshabilitar." está en un color gris oscuro. Debajo del texto, hay un campo de entrada de texto con un fondo blanco y un borde gris. En la parte inferior derecha del recuadro, hay un botón redondeado con el texto "Siguiente" en un color gris oscuro.

Ilustración 8. Vista de la acción deshabilitar cuenta

Si el administrador desea deshabilitar una cuenta, deberá seleccionar dicha opción en el menú, acto seguido, el sistema mostrará un campo de entrada solicitando la matricula o número de personal de la cuenta a la que desea inutilizar.

Datos de la matricula

Nombre

Juan

Matric

S170

Progra

Ingenieria de software

¿Estas seguro que deseas deshabilitar esta cuenta?

SI NO

Siguiente

Ilustración 9. Vista formulario de deshabilitar cuenta

Ingresada la matricula o número de personal, y de hacer click al botón “Siguiente” el sistema buscará en el base de datos la cuenta con la matricula correspondiente, y mostrará los datos en pantalla, como su nombre, matrícula y programa educativo (en el caso de ser estudiante). Para proceder con el bloqueo de la cuenta, solo tiene que presionar “Siguiente” y confirmar la acción. Si todo salió bien, el sistema alertará con mensaje de éxito en pantalla que la acción se ha realizado correctamente.

Ingrese los datos solicitados.

Nombre:

Ingrese su nombre...

Matricula:

Ingrese su matricula...

Programa educativo:

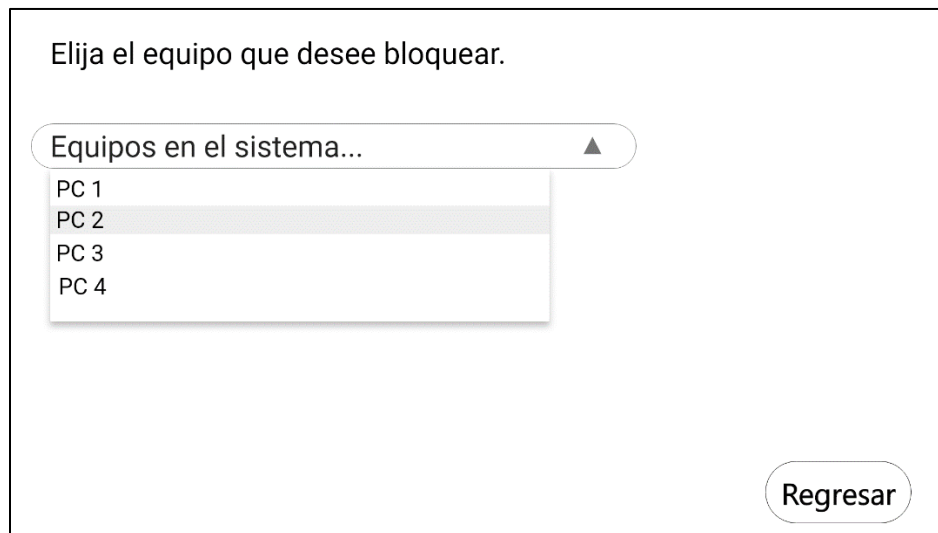
Ingrese su programa educativo...

Monto del saldo solicitado en \$

Siguiente

Ilustración 10. Vista del formulario para agregar saldo

Cuando un Moderador desee agregar saldo a una cuenta, deberá seleccionar la opción de “Agregar saldo” y el sistema mostrará un formulario en pantalla en donde deberá llenar matricula o número de personal de la cuenta a la cual se le sumará el monto indicado. Para continuar solo deberá hacer click en “Siguiente” y el sistema pedirá una confirmación para realizar la carga.



Elija el equipo que desee bloquear.

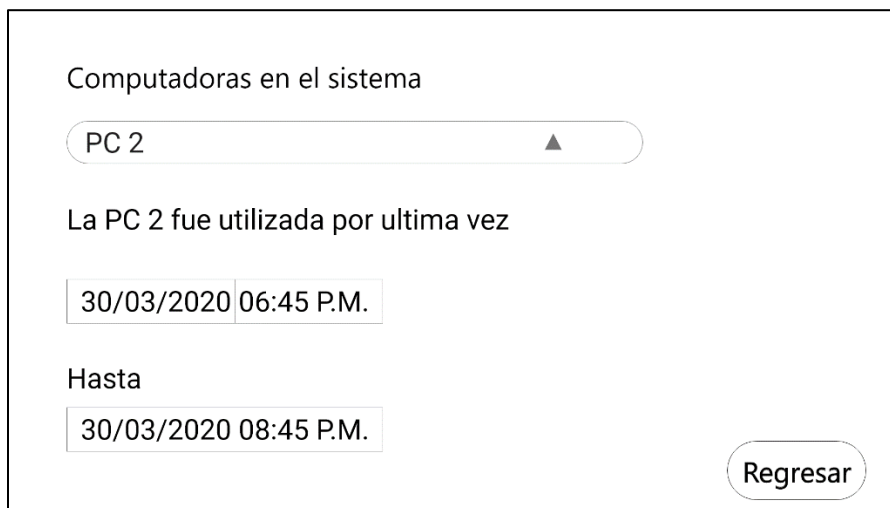
Equipos en el sistema... ▲

- PC 1
- PC 2
- PC 3
- PC 4

Regresar

Ilustración 11. Vista de acción para bloquear equipo

Cuando se selecciona la opción de “Bloquear equipo” el sistema mostrará un menú desplegable en donde el Moderador deberá ingresar, ya sea en texto o buscar en la lista de los equipos registrados previamente en el sistema, seleccionarlo y proceder a bloquearlo remotamente con presionar un solo botón. Solo el administrador podrá desbloquearlo.



Computadoras en el sistema

PC 2 ▲

La PC 2 fue utilizada por ultima vez

30/03/2020 06:45 P.M.

Hasta

30/03/2020 08:45 P.M.

Regresar

Ilustración 12. Vista de consulta de historial de uso

La opción de Consultar historial, el moderador deberá ingresar el identificador del equipo que desee consultar su historial de uso. Al ingresarlo deberá proceder a seleccionar la opción “Siguiete” y el sistema mostrará en pantalla todos los registros que se tengan en la base de datos de uso con esa computadora (fecha y hora) de inicio a fin.

Datos del consumidor:

Nombre:
Juan Pérez

Matricula:
S1701694

Programa educativo:
Ingenieria de Software

Validar usuario:
En espera... ▲

En espera...
Valido

Ilustración 13. Vista de validación de consumidor

Para poder realizar esta acción se requiere previamente que el consumidor haya realizado su registro individualmente (puede ver esta función más adelante en el subtema 5.2). El moderador solo deberá ingresar la matricula del usuario y verificar que los datos estén correctos y coincidan con la persona física. Acto siguiente, podrá cambiar el estado de “Validar usuario” de “En espera” a “Válido” y confirmar la acción.

6.2 Aplicación Cliente esclavo

La aplicación cliente se implementado de igual forma con C#, y con la librería Forms para el desarrollo de interfaces gráficas. Como se ha seguido la arquitectura de dos capas, siendo la de conexión, para hacer consultas al servidor y la de Presentación, que son las interfaces de usuario con las que el cliente interactuará.

A diferencia del cliente Maestro, está aplicación será instalada en los equipos funcionar como un programa de arranque, por lo que para acceder al equipo primero debe de abrir el programa y mediante un Inicio de sesión el usuario podrá acceder al sistema y así realizar las operaciones que quiera.

¿Qué desea llevar a cabo?

Iniciar sesión

Registrarse

Usar computadora

Imprimir

Consultar saldo

This is a general interface prototype. It features a central menu with five options, each enclosed in a rounded rectangular button. The options are: 'Iniciar sesión', 'Registrarse', 'Usar computadora', 'Imprimir', and 'Consultar saldo'. The entire menu is contained within a larger rounded rectangular frame.

Ilustración 14. Prototipo de interfaz de vista general

Esta vista es general, es decir, se encuentra en el cliente maestro y esclavo, y la principal para ingresar al sistema. Para esto, el usuario deberá seleccionar la opción de “Iniciar Sesión” y llenar los campos requeridos.

Ingrese sus datos.

Matricula:

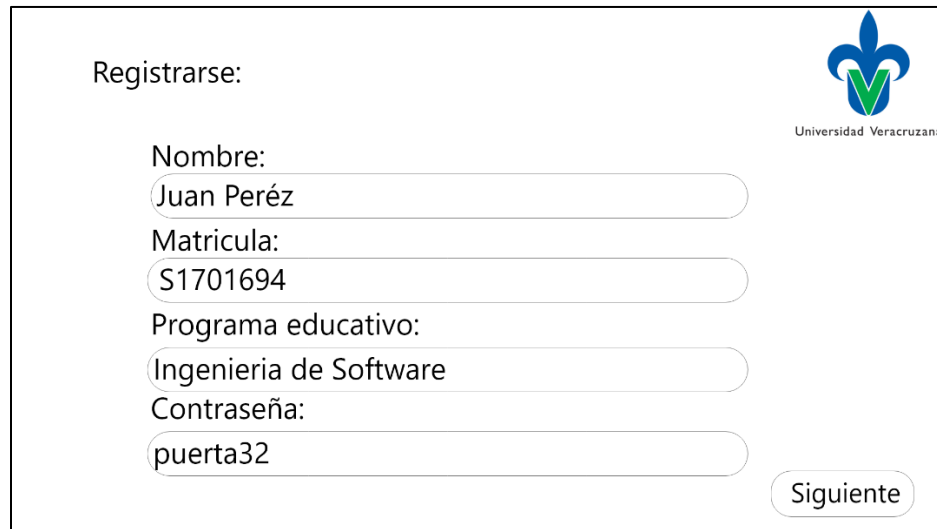
Contraseña:

Next


This is a login view prototype. It contains a title 'Ingrese sus datos.' followed by two labels, 'Matricula:' and 'Contraseña:', each with a corresponding text input field. A 'Next' button is located at the bottom right of the form area.

Ilustración 15. Vista del login del sistema

Para acceder al sistema, el usuario deberá ingresar su Matricula o Número de personal, junto con su contraseña, que estableció al momento de realizar su registro, en el campo correspondiente. El sistema determinará automáticamente si la cuenta es de tipo Consumidor o tipo Administrador/Moderador, y lo redireccionará automáticamente a la vista correspondiente.



Registrarse:


Universidad Veracruzana

Nombre:

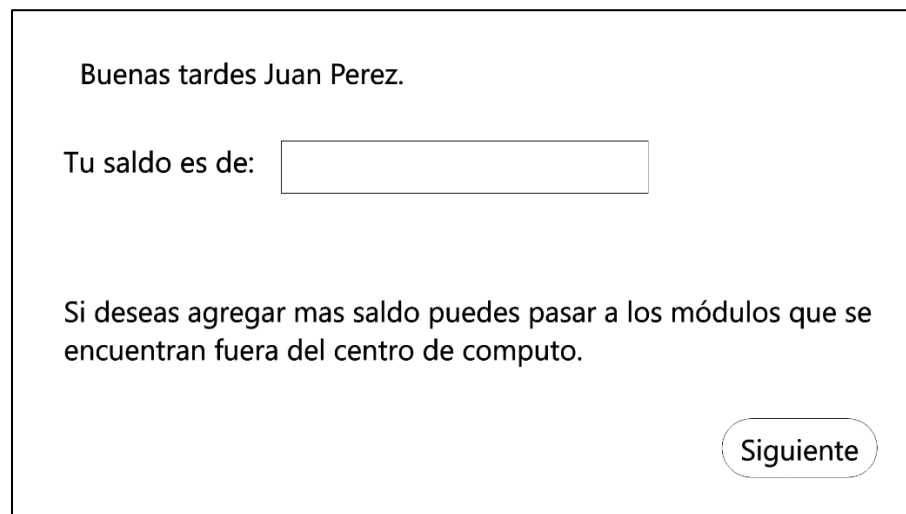
Matricula:

Programa educativo:

Contraseña:

Ilustración 16. Vista prototipo del registro de cliente

En el caso de que el usuario requiera de una cuenta para acceder al sistema, deberá en primer lugar registrarse en él. Seleccionando la opción “Registrarse” el sistema mostrará un formulario de registro que requiere de los datos del usuario necesarios para crear una cuenta y guardarla como registro en la base de datos.



Buenas tardes Juan Perez.

Tu saldo es de:

Si deseas agregar mas saldo puedes pasar a los módulos que se encuentran fuera del centro de computo.

Ilustración 17. Vista de la acción consultar saldo

Con la opción de consultar saldo, el sistema mostrará en pantalla el saldo disponible que tenga la cuenta en sesión. Para agregar saldo se requiere de realizar un pago representativo y acudir con los encargados del área de cómputo, el cual confirmará el pago y procederá a cargar saldo a la cuenta de dicho usuario.

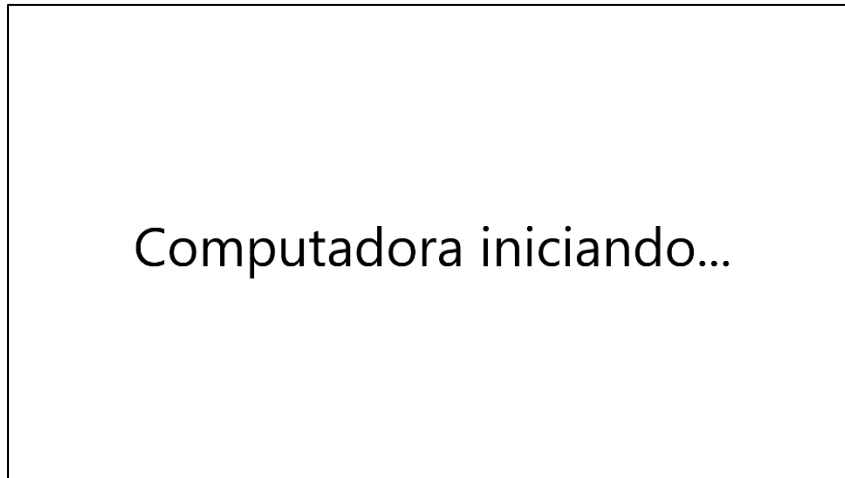


Ilustración 18. Vista prototipo al acceder a un equipo con saldo suficiente

Cuando el cliente consumidor desee “Ocupar la computadora”, solo tiene que seleccionar dicha opción, y si tiene saldo suficiente, el sistema se minimizará para que el usuario pueda ocupar el equipo libremente por 2 horas. Terminado ese lapso de tiempo, se alertará al usuario que su tiempo se ha agotado.

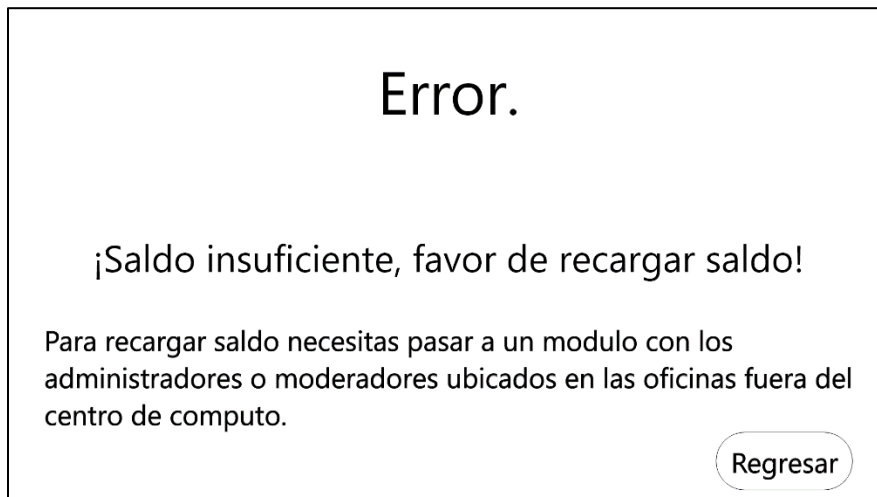


Ilustración 19. Vista de error de saldo insuficiente

El sistema alertará al usuario en caso de que seleccione la opción de “Usar computadora” y su cuenta no posea de saldo suficiente, denegando la acción y permitiendo regresar al menú principal.

6.3 Protocolo de comunicación

La comunicación entre el cliente esclavo y maestro se pretende implementar usando *sockets*, ya que, es un método de comunicación básico y de bajo nivel que está presente en la mayoría de los lenguajes de programación.

Para llevar a cabo la comunicación y enviar los tipos de datos, se ha especificado un protocolo de comunicación específico. Según el puerto al que se conecte la aplicación, accederá a las funciones de usuario consumidor (puerto 12302) o de administrador (puerto 12301). En ambos casos, los pasos son comunes.

Lo que se envía es un código de verificación que permite que el sistema maestro acepte un equipo nuevo en el sistema, este código se genera automáticamente al momento de registrar un equipo nuevo. Una vez teniendo el código, el servidor permite la conexión con el equipo y autoriza la modificación de las tablas de la base de datos del servidor local (Cliente maestro).

6.4 Gestión de la configuración

La gestión de la configuración es una tarea importante para llevar a cabo la implementación de software ya que proporciona de un control versiones, lo que conlleva la posibilidad de volver a anteriores versiones estables, en caso de ser necesario. También facilita el desarrollo entre un equipo de programadores.

Por eso, todo software que se ha desarrollado se encuentra disponible en GitHub, parte Administrador (<https://github.com/kubota54/webf>) y parte Cliente (https://github.com/eduardo1aguirre/OBELISCO_CLIENTE), el cual su uso fue de bastante ayuda y comodidad para el control de versiones del producto en estado de desarrollo temprana.

7. Métricas

En esta sección, se mencionarán las métricas empleadas para obtener una estimación de complejidad y costo del proyecto a desarrollar.

7.1 Métricas Orientadas a la Función

Los puntos de función son una técnica de medición del tamaño funcional del software, desde el punto de vista del cliente.

El objetivo del proceso de medición es ser una medida consistente (que dos profesionales de un mismo proyecto lleguen al mismo resultado) y simple para minimizar el esfuerzo de la medición.

- Estima el esfuerzo en un proyecto informático (HH).
- Estima la duración de un proyecto de software (en meses)
- Estima el costo del proyecto informático.

Tipo/ Complejidad	Baja	Media	Alta
(EI) Entrada externa	3 PF	4 PF	6 PF
(EO) Salida externa	4 PF	5 PF	7 PF
(EQ) Consulta externa	3 PF	4 PF	6 PF

(ILF) Archivo lógico interno	7 PF	10 PF	15 PF
(EIF) Archivo de interfaz externo	5 PF	7 PF	10 PF

Tabla 6. Valores estándar propuestos por la IFPUG, International Function Point Users Group

7.2 Análisis de Puntos Función

Para aplicar las métricas orientadas a la función se tendrá que enlistar todos los requerimientos funcionales del sistema, en donde se le indicará un Tipo de función y una complejidad como se ve en la Tabla 6 en el subtema anterior.

1. El sistema posee de un apartado para iniciar sesión [**EI ALTA 6PF**]
2. El sistema posee un apartado de registro de usuario [**EI ALTA 6PF**]
3. El sistema muestra los datos del cliente (matrícula, nombre, saldo) [**EO MEDIA 5PF**]
4. El sistema muestra el tiempo de uso de la computadora [**EI MEDIA 4PF**]
5. El sistema permite usar la computadora por mínimo dos horas. [**EI MEDIA 4PF**]
6. El sistema posee un apartado para cerrar sesión. [**EQ ALTA 6PF**]
7. El sistema deberá permitir bloquear y desbloquear la computadora por el administrador y moderador en cualquier momento. [**EI ALTA 6PF**]
8. El sistema deberá guardar los registros de hora, fecha y datos de quienes usen las computadoras. [**EI ALTA 6PF**]
9. El sistema podrá bloquear a las personas que le están dando un mal uso a un equipo. [**EI MEDIA 4PF**]
10. El sistema deberá descontar el saldo automáticamente, según el servicio que se

Ilustración 20. Lista de capacidades funcionales del sistema

La lista completa se encuentra en el documento *Aplicando Métricas Orientadas a Tamaño*, en la carpeta de *Documentación* del repositorio del presente documento. Teniendo identificados todos los puntos función del sistema, podemos calcular los Puntos de Función Sin Ajustar.

Tipo/ Complejidad	Baja	Media	Alta	Subtotal
(EI) Entrada externa	3 PF	3 x 4 PF	6 x 6 PF	48
(EO) Salida externa	4 PF	2 x 5 PF	7 PF	10
(EQ) Consulta externa	3 PF	4 PF	3 x 6 PF	18
(ILF) Archivo lógico interno	7 PF	10 PF	1 x 15 PF	15
(EIF) Archivo de interfaz externo	5 PF	7 PF	1 x 10 PF	10
			PFSA	101

Tabla 7. Ponderando el valor de Punto Función Sin Ajustar

7.2.1 Factor de Ajuste

El factor de ajuste está basado en 14 características generales del sistema (General System Characteristics ó GSC's) que evalúan la funcionalidad general de la aplicación que se está midiendo. Cada característica tiene

asociada una serie de cuestiones o preguntas acerca de la misma, cuya respuesta ayuda a determinar su grado de importancia dentro del sistema en función de una escala que va de cero (sin influencia) a cinco (esencial). (Rodríguez, 1999)

Nº de Factor	Característica del sistema	Nivel de influencia 0..5
1	Comunicación de datos	3
2	Procesamiento distribuido	5
3	Performance (desempeño)	1
4	Configuración del equipamiento	1
5	Volumen de transacciones	0
6	Entrada de datos on-line	3
7	Interfase con el usuario	2
8	Actualización on-line	1
9	Procesamiento complejo	1
10	Reusabilidad	1
11	Facilidad de implementación	0
12	Facilidad de operación	1
13	Múltiples locales	2
14	Facilidad de cambios	1
Nivel de influencia		22

Tabla 8. Las 14 características generales del sistema para el cálculo de punto función

Para tener un nivel de influencia más preciso se utilizó el siguiente documento como guía para determinar el nivel de influencia de cada característica (<https://docplayer.es/12389562-Determinacion-del-nivel-de-influencia.html>), en donde se desglosaba cada característica como una pregunta y el nivel se determinaba con ciertas afirmaciones que cumpliera el sistema. Sumamos el subtotal obtenido para calcular el *Nivel de influencia* como se muestra en la tabla 8.

7.2.2 PFA (Puntos de Función Ajustado)

$$PFA = PFSA * [0.65 + (0.01 * \text{factor de ajuste})]$$

$$PFA = 101 * [0.65 + (0.01 * 22)]$$

$$PFA = 101 * 0.87$$

$$PFA = \underline{87.87}$$

Ilustración 21. Fórmula para calcular los PFA

Teniendo los *Puntos de Función Sin Ajustar* y el *Factor de Ajuste*, podemos calcular los Puntos de Función Ajustado. Como se podrá observar en la fórmula se utilizan dos constantes y dos variables descritas anteriormente. Con ese valor calculado tenemos la posibilidad de conocer las linead de

código estimadas para el proyecto, y más adelante ocuparla para realizar estimación de horas/hombre, costos y duración en meses del proyecto

7.3 Métricas Orientadas al Tamaño

Son utilizadas para la normalización de las medidas de calidad o productividad considerando el tamaño del software que se ha producido. Tales como el esfuerzo humano, coste, productividad o documentación. De esta manera se puede evaluar los beneficios derivados de nuevos métodos y herramientas, así como justificar el uso de nuevas herramientas o capacitaciones para la formación del equipo de desarrollo.

7.3.1 COCOMO: Modelo constructivo de costes

El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés Constructive Cost Model) es un modelo matemático de base empírica utilizado para estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado (heysshell, 2010). Para este proyecto abordaremos el modelo *básico*.

7.3.2 COCOMO: Modelo Básico

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

Modo	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semi - Orgánico	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.33

Tabla 9. Tabla de constantes del Modelo Orgánico

Estos valores son para las fórmulas:

- Personas necesarias por mes para realizar el proyecto: $(MM) = a \cdot (Kl^b)$
- Tiempo de desarrollo del proyecto: $(TDEV) = c \cdot (MM^d)$
- Personas necesarias para realizar el proyecto: $(CosteH) = MM/TDEV$
- Costo total del proyecto: $(CosteM) = CosteH \cdot \text{Salario medio entre los programadores y analistas.}$

Las **LOC** (Líneas de código) por punto función en C# es de **58**, por lo que:

$$LOC = 87.87 * 58 = \underline{5,096} \text{ líneas de código}$$

Ilustración 22. Calculando las líneas de código aproximadas

Recordemos que los puntos de función ajustado los obtuvimos en el apartado 6.2.2 y es el valor que se multiplica por 58, dándonos como resultado un valor aproximado de lo que sería la cantidad de líneas de código necesarias para realizar el proyecto.

- Personas necesarias por mes para llevar adelante el proyecto:

$$(MM) = a * (Kl^b) // Kl \text{ son las líneas de código en miles}$$

$$(MM) = 2.4 * (5.09^{1.05})$$

$$(MM) = 2.4 * 5.5214$$

$$(MM) = \underline{13.2514} \text{ persona/mes}$$

Ilustración 23. Fórmula para calcular personas necesarias por mes

Las constantes a y b son remplazadas por los valores descritos en la tabla 9 del modelo orgánico, y la variable Kl son los miles de líneas de código del proyecto, el cual ya calculamos y se muestra en la ilustración 22 de este apartado. Solo sustituimos valores y así obtenemos el estimado de personas necesarias por mes para llevar adelante el proyecto.

- Tiempo de desarrollo del proyecto:

$$(TDEV) = c * (MM^d)$$

$$(TDEV) = 2.5 * (13.2514^{0.38})$$

$$(TDEV) = 2.5 * 2.9969$$

$$(TDEV) = \underline{6.6742} \text{ meses}$$

Ilustración 24. Fórmula para calcular el tiempo de desarrollo

Al igual que en la primera fórmula, las constantes c y d son obtenidas de la tabla 9, la variable MM (personas/mes) es el resultado que se obtuvo en la fórmula anterior, por lo que solo sustituimos valores para así calcular los meses necesarios para desarrollar el proyecto.

- Personas necesarias para realizar el proyecto:

$$(CosteH) = MM/TDEV$$

$$(CosteH) = 13.2514 / 6.6742$$

$$(CosteH) = 1.98 = 2 \text{ personas}$$

Ilustración 25. Fórmula de personas necesarias

En este caso requerimos de haber calculado las variables MM (persona/mes) y TDEV (tiempo de desarrollo en meses) para así realizar una división y obtener el número de personas requeridas para realizar el proyecto.

- Costo total del proyecto:

$$(CosteM) = CosteH * \text{Salario medio entre los programadores y analistas.}$$

$$(CosteM) = 2 * 92.31 \$ \text{ hora}$$

$$(CosteM) = 184.62 \$ \text{ hora}$$

Ilustración 26. Fórmula para obtener costo total

En este caso solo se multiplicó por dos (costeH) el promedio del salario de un programador en México y así calcular el CosteM (total) del proyecto en horas.

7.3.2 Estimación de Esfuerzo

El establecimiento de métodos que permitan determinar y, posteriormente, alcanzar estos objetivos de una forma lo más real y exacta posible ha sido un factor cada vez más importante para la Ingeniería Informática en su conjunto. La estimación de esfuerzo pretende llevar a cabo el cumplimiento de los plazos de entrega dentro de costos establecidos manteniendo niveles de calidad; así como poder realizar un seguimiento y control de la evolución de los proyectos.

Lenguaje	Horas PF Promedio	Líneas de Código por PF
Ensamblador	25	300
COBOL	15	100
Lenguajes 4ta Gen	8	58

Tabla 10. Tabla de horas promedio por puntos función

Como ya se ha mencionado con anterioridad en este documento el lenguaje utilizado para realizar el proyecto es C#, por lo tanto, entra en el área de lenguajes de 4ta generación, entonces podemos utilizar el promedio de horas para realizar el último cálculo para la estimación de esfuerzo.

$$H/H = PFA * \text{Horas PF promedio}$$

$$H/H = 87.87 * 8$$

$$H/H = 703 \text{ Horas Hombre}$$

Ilustración 27. Fórmula para calcular las horas hombre

La variable PFA son los puntos de función ajustado, el cual ya fue calculado en el apartado 6.2.2 y las horas promedio por punto función las obtenemos en la tabla 10, y ahora solo realizar una simple multiplicación con ambos valores y así obtener las horas hombre.

- Por lo tanto = **703** * 184.62 = 129,787.86 \$

Ilustración 28. Cálculo final para coste total del proyecto

Si recordamos en la imagen 26, obtuvimos el valor de CosteM, que es el costo del proyecto en horas, por lo que, ya teniendo las horas hombres realizamos una multiplicación del CosteM por las Horas/Hombre. El resultado de esa operación demostrará una aproximación del costo total del proyecto en pesos mexicanos.

8. Conclusiones y trabajos futuros

En la presente sección del documento, se presentarán las conclusiones obtenidas tras la realización del proyecto, así como las tareas pendientes de desarrollo hasta la entrega definitiva del mismo.

8.1 Conclusiones

Antes de introducirnos de lleno en el mundo del desarrollo de software, pensábamos que las aplicaciones eran creadas solamente escribiendo código. Al entrar a la Licenciatura en Ingeniería de Software, comprendimos que es un proceso mucho más detallado y delicado. Aprendimos más de la parte de la documentación, ya que nos propusimos a lograr un diseño de software casi perfecto. Cada miembro de este equipo tiene sus fortalezas, las cuales fueron utilizadas de forma correcta, y siempre exigiendo más por parte de cada uno. Los tiempos no nos fueron suficientes, se atravesaron muchos obstáculos en el camino, nos enfocamos cada uno en lo que es bueno y

logramos trabajar todo el diseño de software la manera más precisa que pudimos.

Como también en lo personal trabajamos con compañeros que nunca había trabajado, por lo cual me ayudó a no solo desarrollar nuevos conocimientos sino también a entablar nuevas relaciones, esto porque la metodología, Scrum, sugiere diversas reuniones durante el desarrollo. En el proceso también aprendimos que mantener la comunicación es fundamental para llevar a cabo cualquier proyecto. Ya que debido a los problemas derivados por la contingencia que se vive actualmente se tuvieron que buscar distintas técnicas para estar en comunicación.

Trabajar desde casa es posible, sin embargo, nosotros no estábamos tan familiarizados con esta metodología que requiere de mucho contacto, tanto con el equipo de desarrollo como con el cliente, por tal motivo, nos fuimos adaptando en este corto periodo de un mes.

Como conclusión de este proyecto, lo más básico, pero no por ello menos importante que he aprendido, es que la realización de un proyecto no es algo trivial, que pueden surgir muchos contratiempos inesperados, y la capacidad para analizar y superar dichos contratiempos es una habilidad que un buen jefe de proyectos debe poseer, además de establecer e imponer muy buena organización y planificación.

8.2 Trabajos futuros

Como ya se ha mencionado brevemente en el apartado anterior, se han cumplido todos los requisitos iniciales propuestos para este, pero aún quedan algunos requisitos que no fueron implementados, por cuestión de tiempo y porque el cliente fue accesible en la entrega del producto y abierto a futuras mejoras del sistema. Algunos requisitos destacados para su futura implementación son los siguientes:

- Realizar impresiones desde el equipo cliente
- Implementar una base de datos con fotografía de los clientes
- Generar un Boucher de pago para realizar recarga de saldo

Bibliografía

- Amoedo, D. (s.f.). *Ubuntu*. Obtenido de <https://ubunlog.com/mysql-workbench-bases-datos/>
- B., G. (13 de Mayo de 2019). *Hostinger*. Obtenido de <https://www.hostinger.mx/tutoriales/que-es-mysql/>
- Delgado, D. O. (29 de Marzo de 2017). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-c-introduccion/>
- EcuRed. (s.f.). Obtenido de https://www.ecured.cu/Microsoft_Visual_Studio
- ExpertosNegociosOnline. (29 de Noviembre de 2019). Obtenido de ExpertosNegociosOnline.com: <https://www.expertosnegociosonline.com/que-es-trello-para-que-sirve/>
- heyshell. (19 de Octubre de 2010). *Blogger.com*. Obtenido de <http://heysellopez.blogspot.com/2010/10/metrica-orientada-al-tamano-cocomo.html>
- Lucidchart. (s.f.). *Lucidchart*. Obtenido de <https://www.lucidchart.com/pages/es>
- Molera, L. (24 de Mayo de 2019). *Hubspot*. Obtenido de <https://blog.hubspot.es/marketing/para-que-sirve-adobe-xd>
- Pillaca, R. (s.f.). Obtenido de <http://www.techeraperu.com/pdf/Enterprise%20Architect%20%2012.pdf>
- Rodríguez, F. S. (Mayo de 1999). Obtenido de <https://www.infor.uva.es/~manso/calidad/PFA-CLM-2011>
- techopedia. (18 de Agosto de 2011). Obtenido de <https://www.techopedia.com/definition/16208/socket>
- Wikipedia. (30 de Mayo de 2020). Obtenido de <https://es.wikipedia.org/wiki/GitHub>